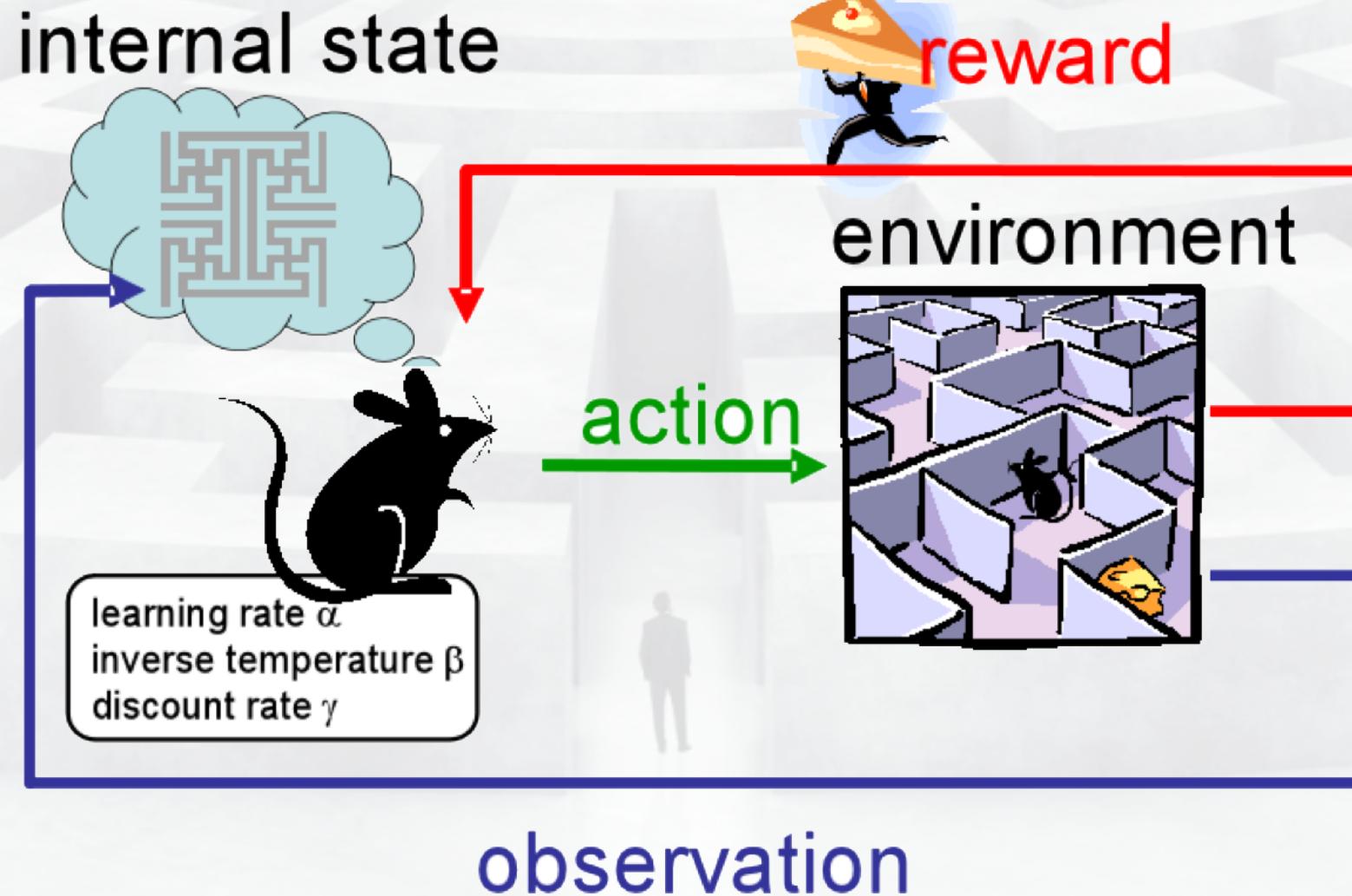


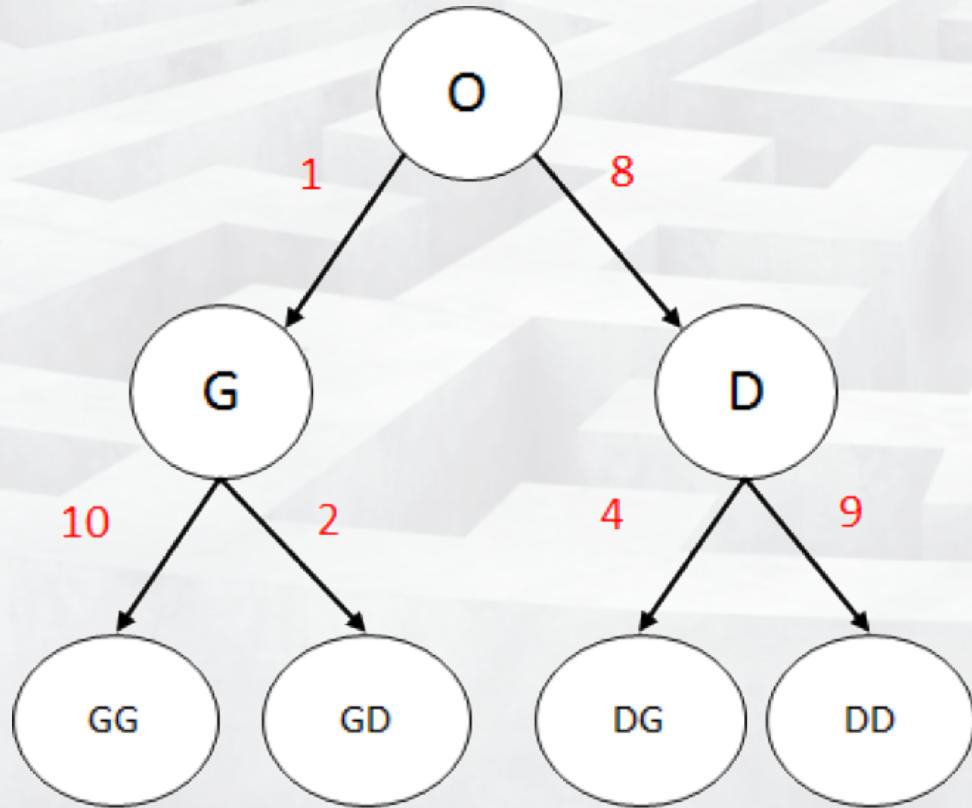
Apprentissage par renforcement

Méthode acteur-critique



I. Environnement et modèle

- On cherche à modéliser l'apprentissage de la topologie d'un labyrinthe contenant des récompenses:



Le labyrinthe

Le labyrinthe comprend:

- Les états (O, G, D,...)
- Les récompenses
(1,8,...)

Pour apprendre la topologie le programme à besoin de:

- Assigner une valeur $v(e)$ à l'état e
- Assigner une valeur $q(a,e)$ à l'action a effectuée à partir de l'état e
- Enregistrer la valeur de la récompense $r(a,e)$ donnée par l'action a à partir de l'état e.
- Assigner une probabilité $\pi(a,e)$ d'effectuer l'action a à partir de l'état e.

I. Environnement et modèle

- À chaque itération du programme est effectué une action.
- À chaque itération le programme met à jour les valeurs $V(e)$, $q(a, e)$, $r(a, e)$ et

$$\widehat{V}(e) \leftarrow \widehat{V}(e) + \eta [r(a, e) + \gamma \widehat{V}(e') - \widehat{V}(e)]$$

$$q(a, e) \leftarrow q(a, e) + \tilde{\eta} [r(a, e) + \gamma \widehat{V}(e') - \widehat{V}(e)]$$

$$\pi(a, e) = \frac{\exp[\beta \cdot q(a, e)]}{\sum_{a'} \exp[\beta \cdot q(a', e)]}$$

Ces valeurs dépendent à la fois des états futurs (e') et des actions effectuées pour mettre à jour la probabilité $\pi(a, e)$.

Les paramètres $\beta, \eta, \tilde{\eta}$ et γ influent le comportement de l'algorithme.

η et $\tilde{\eta}$ sont des degrés d'apprentissage, γ est la projection vers le futur et β est la balance exploration/exploitation

II. Programmation du modèle acteur-critique

1) Modèle simple à 2 étages

- $[V, PI] = \text{ActorCritic2} (\eta, \eta_2, \gamma, \beta, N, i, PI, V, R)$

Elle prend en entrée les paramètres du modèle, le nombre d'itérations, le nombre d'expérience, et les vecteurs V et PI qu'elle met à jour à l'aide des formules du modèle.

- $[PImoy, Vmoy] = \text{Moyennes2} (PI, V, N, dimension)$

Transforme les matrices 3D V et PI en matrices 2D Vmoy et PImoy en effectuant la moyenne sur le nombre d'expériences (dimension)

- $\text{GraphiquesMoyennes2} (N, PImoy, Vmoy, R)$

Représente les valeurs des états O, G et D en fonction des itérations

Représente les probabilités d'aller à gauche pour les états O, G et D.

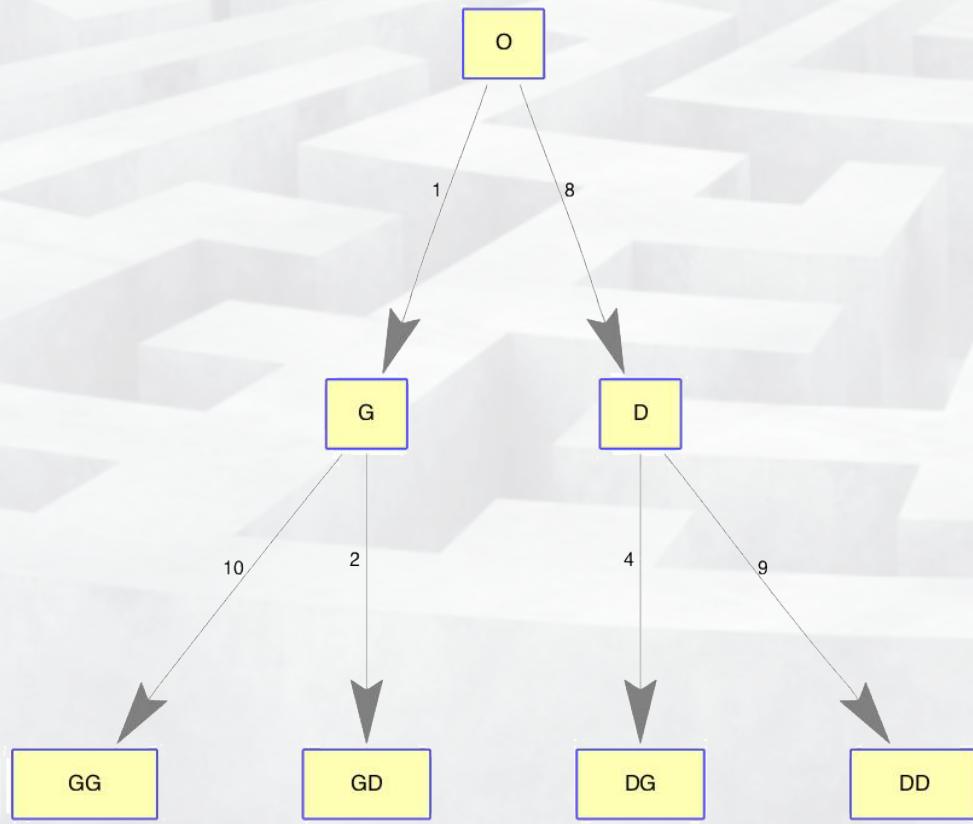
Représente l'arbre du modèle avec les états et les récompenses.

- PPActorCritic2

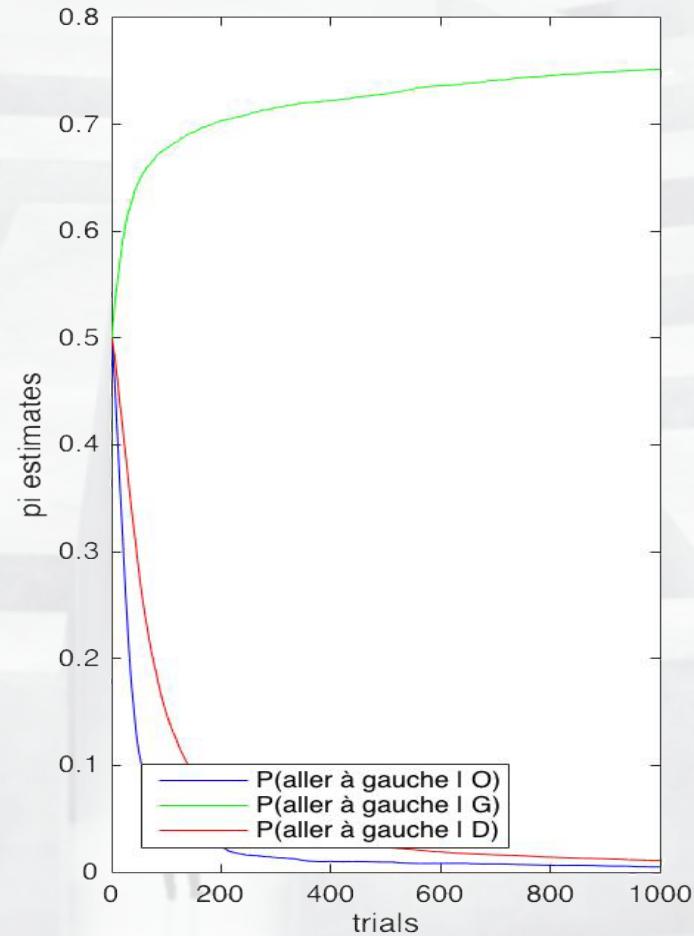
Programme principal où l'on peut faire varier les divers paramètres du modèle, ainsi que le nombre d'expériences.

II. Programmation du modèle acteur-critique

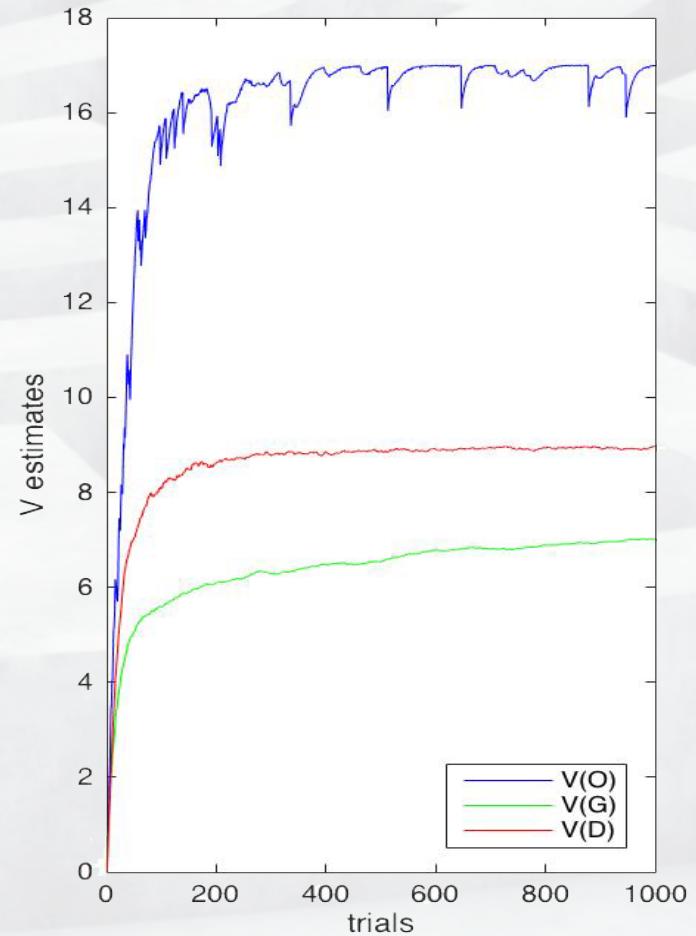
1) Modèle simple à 2 étages



Arbre états-récompenses



Graphes valeurs-probabilités en fonction des itérations



II. Programmation du modèle acteur-critique

2) Modèle à 3 étages

- $[V, PI] = \text{MiseAJourD}(R, Q, V, PI, point, Npoint, i, k, \eta, \eta_2, \gamma, \beta)$

Contient les formules de mise à jour dans le cas où la souris est allée à droite. Idem pour gauche.

- $[V, PI] = \text{MaintenirValeurs} (V, PI, i, point, k)$

Maintient les valeurs de V et PI de l'étape précédente dans le cas où la souris n'est pas passée et donc n'a pas provoqué de mise à jour.

- $[V, PI] = \text{ActorCriticFunction} (\eta, \eta_2, \gamma, \beta, N, i, PI, V, R)$

Comme ActorCritic 2, mais allégée grâce aux fonctions de mise à jour

- $\text{GraphiquesMoyennes3} (N, PI_{moy}, V_{moy}, PI_{SuperMoy}, V_{SuperMoy}, R)$

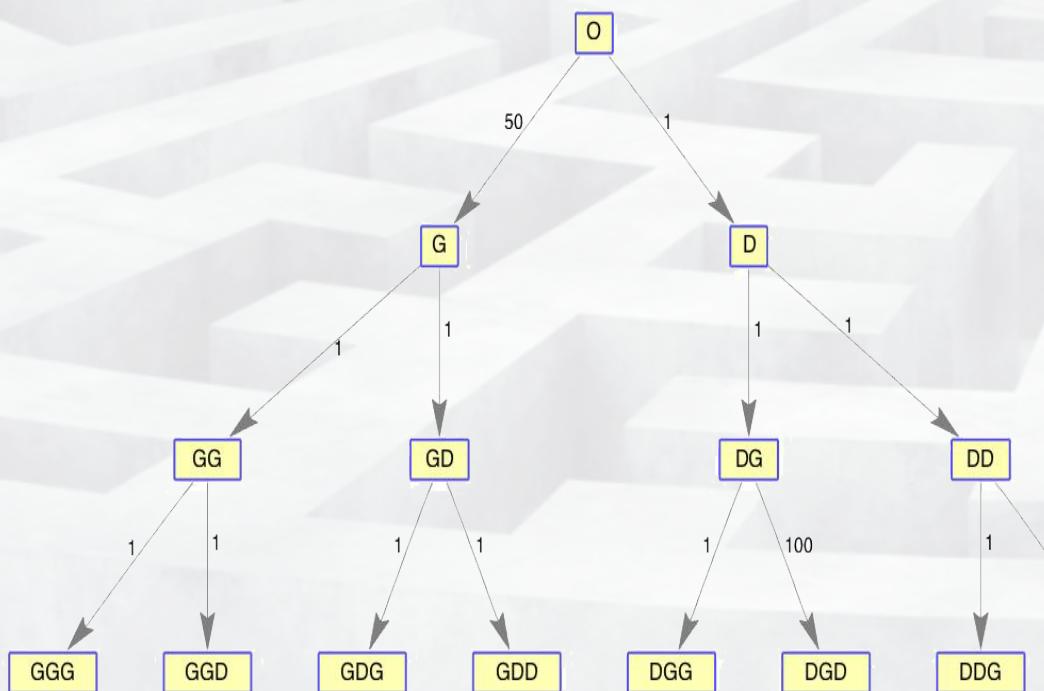
Comme pour le modèle simple, mais avec en plus l'arbre « valeurs probabilités » qui contient les valeurs asymptotiques de valeurs et des probabilités pour chaque état.

- $[PI_{SuperMoy}, V_{SuperMoy}] = \text{SuperMoyennes} (PI_{moy}, V_{moy}, N, seuil)$

Calcule les valeurs asymptotiques de V et PI .

II. Programmation du modèle acteur-critique

2) Modèle à 3 étages



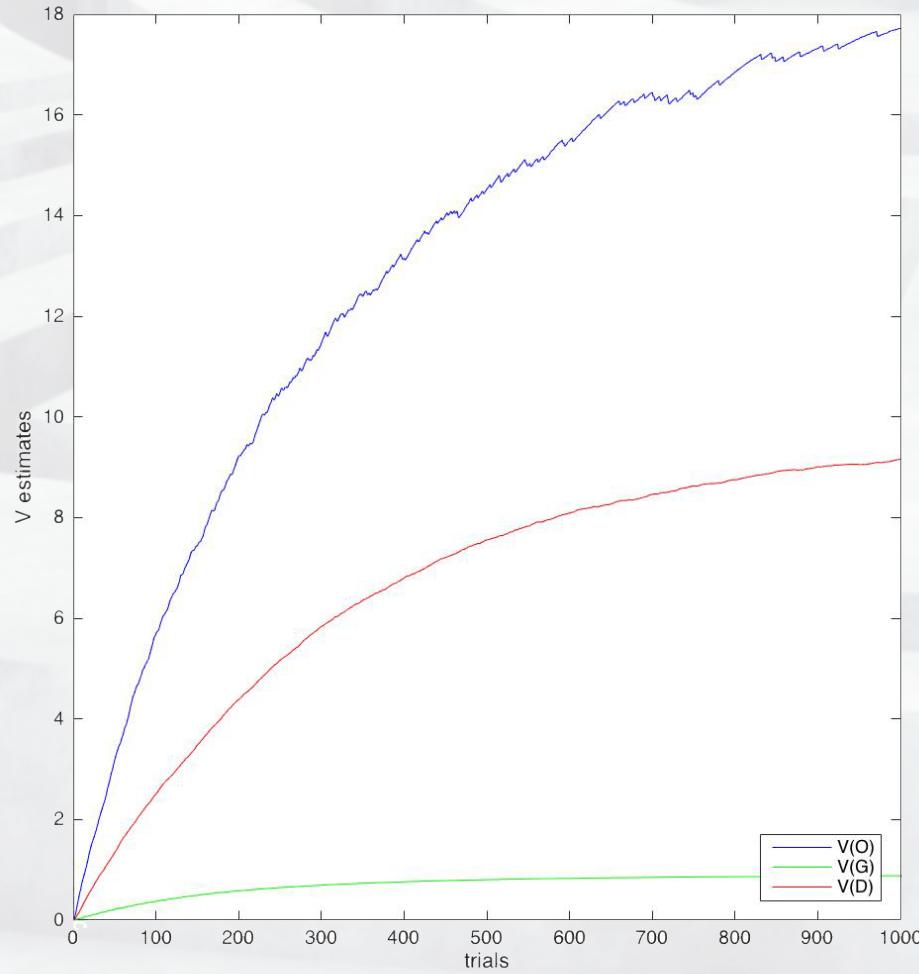
Arbre états-récompenses



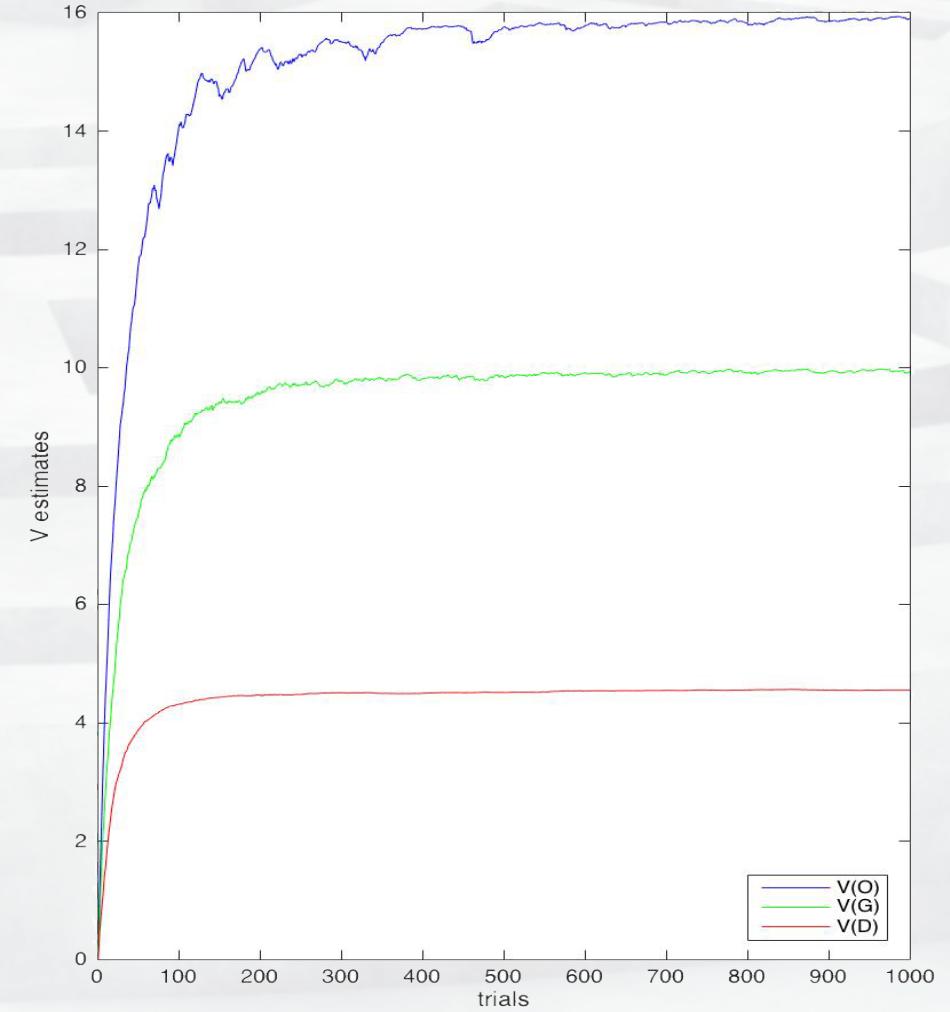
Arbre valeurs-probabilités

III. Influence des paramètres sur le modèle

1) Les taux d'apprentissages

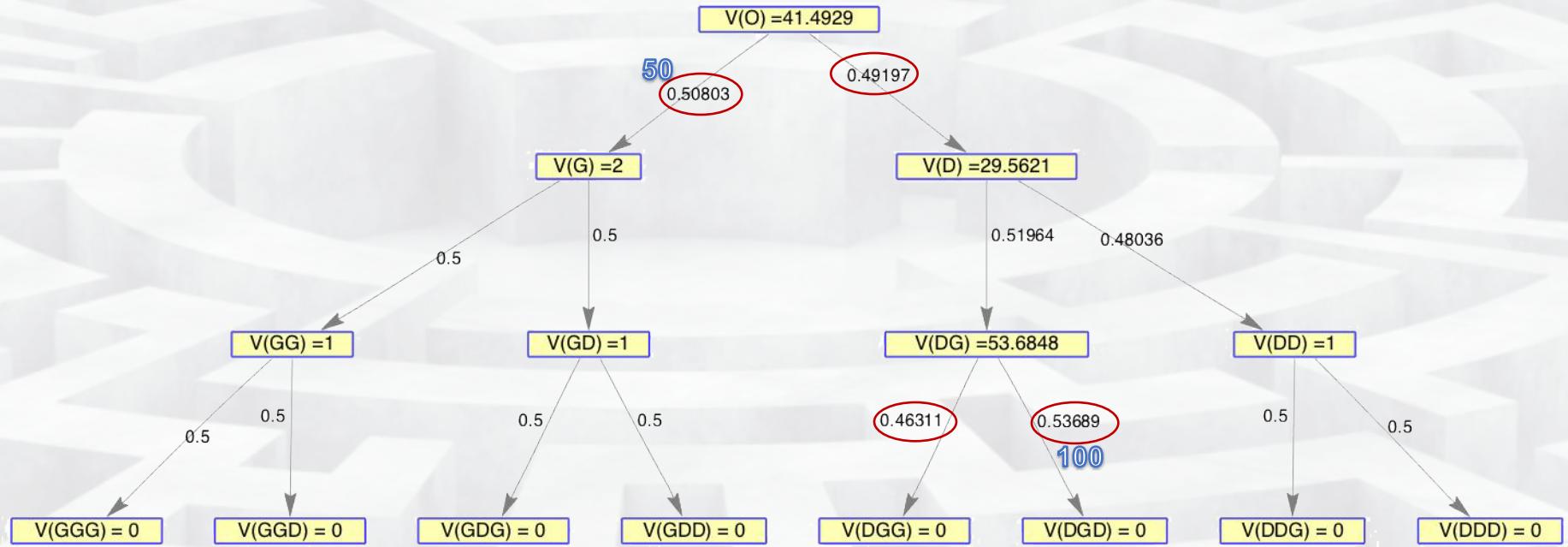


$$\eta = \tilde{\eta} = 0.01$$

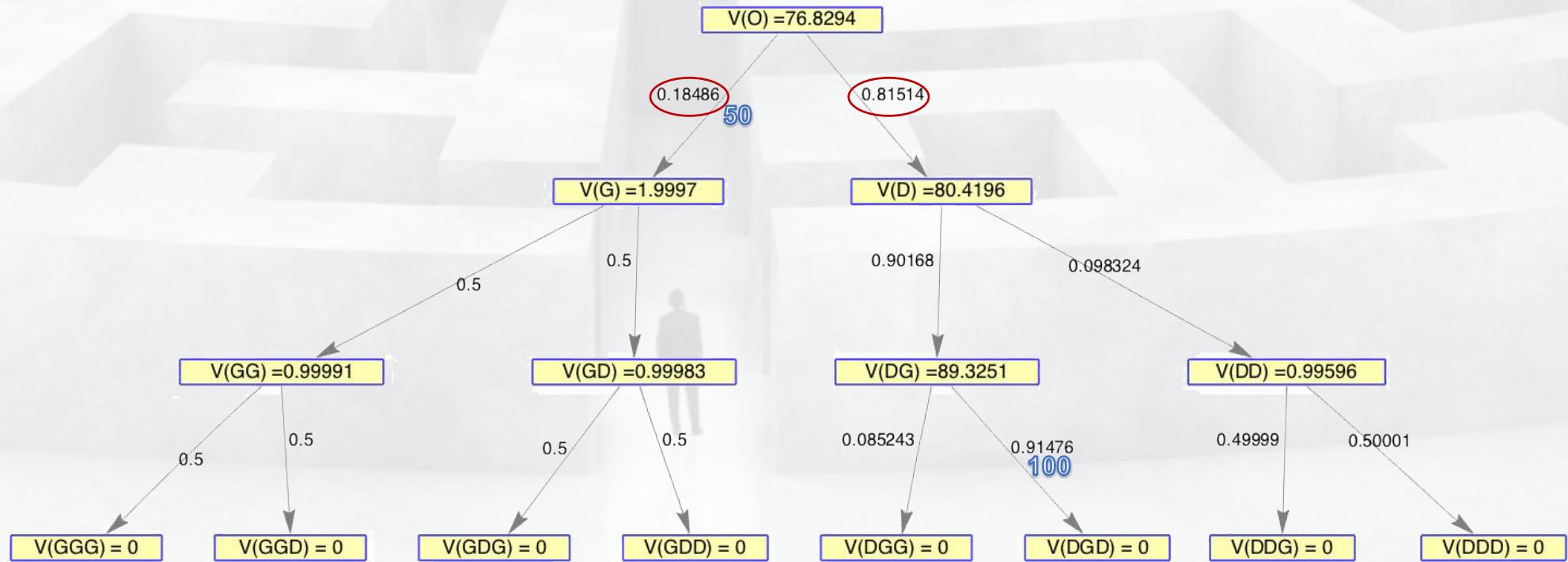


$$\eta = \tilde{\eta} = 0.1$$

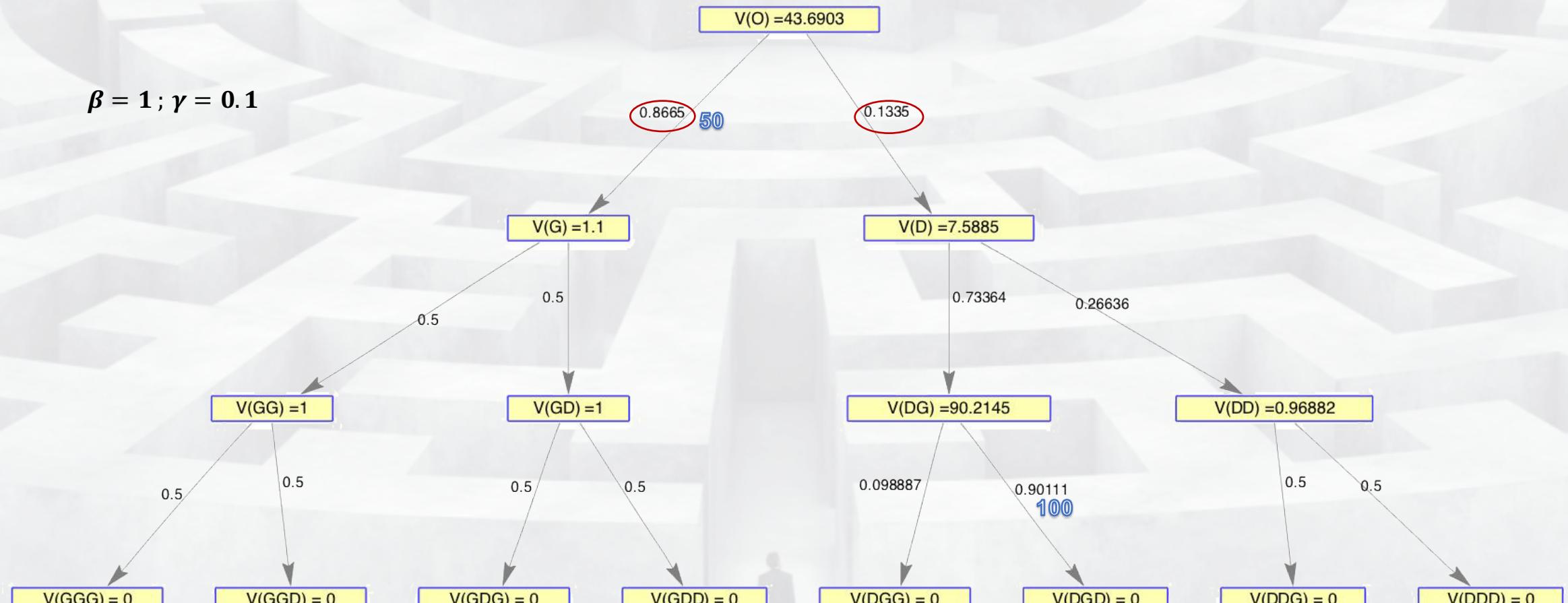
$\beta = 0.01 ; \gamma = 1$



$\beta = 1 ; \gamma = 1$

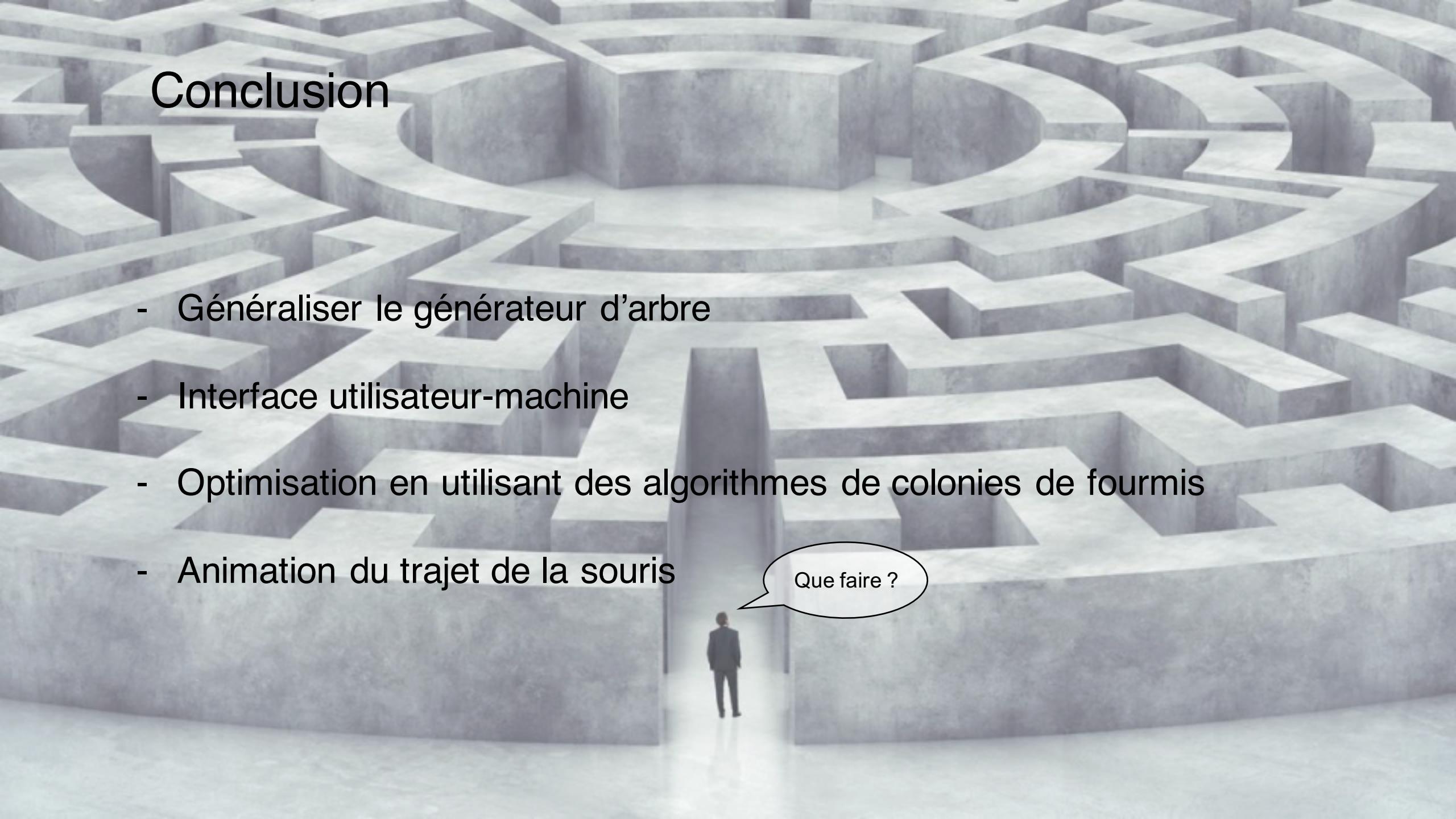


$$\beta = 1; \gamma = 0.1$$



Conclusion

- Généraliser le générateur d'arbre
- Interface utilisateur-machine
- Optimisation en utilisant des algorithmes de colonies de fourmis
- Animation du trajet de la souris



Que faire ?



J'ai trouvé mon fromage...
Et vous ?

