Patrol avoider

Résolution d'un jeu stochastique par des MDPs augmentés

CHÂTEL Romain

MOUADDIB Abdel-Illah

27 novembre 2019

Sommaire

Jeux stochastiques

- Extension de la théorie des jeux à l'incertain
- Prend l'incertitude en compte (environnement, capteurs, effecteurs ...)
- Un ou plusieurs agents avec des buts
- Fonction de gains
- Maximiser le gain immédiat, ET le gain espéré futur
- Actions stochastiques, information incomplète/imparfaite

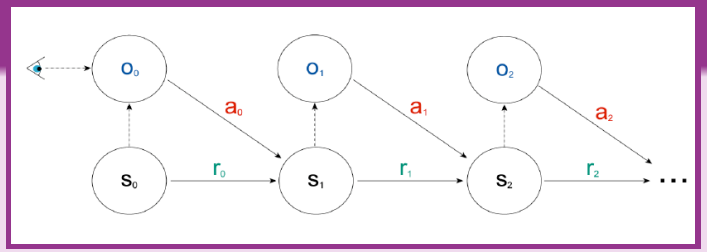
Observabilité totale

- Jeux stochastiques à un seul agent
- Un espace d'états S, propriété markovienne
- Un espace d'actions A
- Un modèle de transitions $T : S \times A \times S \mapsto [0, 1] \Rightarrow Pr(s'|s, a)$
- Une fonction de gains/d'utilité $U : S \times A \mapsto \mathbb{R}$
  $\Rightarrow U(s, a) = R(s) - Cost(s, a)$
- Les agents décident en appliquant une politique $\pi : S \mapsto A$
- Equation d'optimalité de Bellman :

$$v^{\pi *}(s) = \max_{a \in A}[U(s, a) + \gamma \sum_{s' \in S} T(s, a, s')v^{\pi *}(s')]$$

- Résolution par *Value Iteration*, *Policy Iteration*, *Linear programming* ...

Observabilité partielle



Extension à l'incertide sur l'état de l'agent

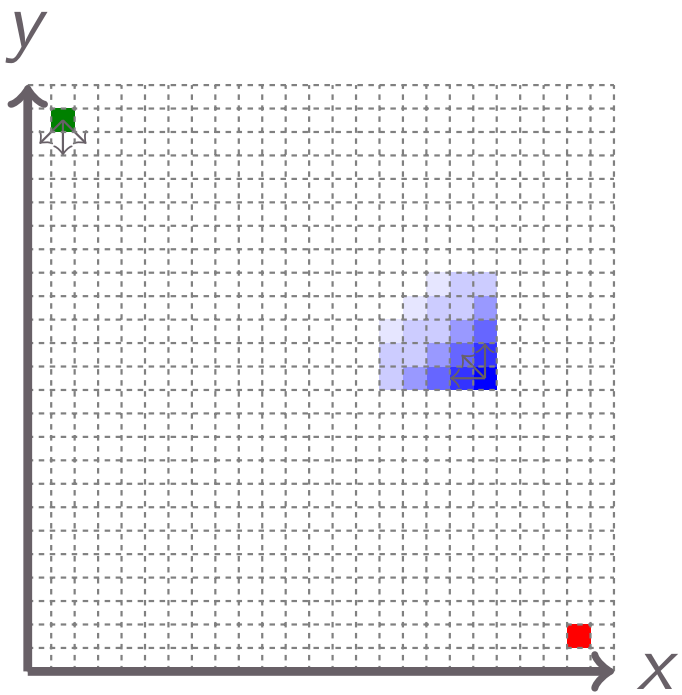
- Un espace d'observations Ω
- Un modèle d'observations $TO : \Omega \times S \times A \mapsto [0, 1] \Rightarrow Pr(o|s', a)$
- Inférer l'état courant à partir des observations
  $\Rightarrow$ Perte de la propriété markovienne
- Transformation en BMDP, espace de croyances sur les états
  $B = [0, 1]^{|S|}$
- Fonction de mise à jour des croyances
  $\zeta : B \times A \times B \Rightarrow Pr(b'|b, a) = \sum_{o \in \Omega} Pr(b'|b, a, o) Pr(o|b, a)$
- Résolution difficile

Sommaire

Éviter un drone de surveillance

Jeu stochastique : (grille stochastique)

- Une cible
- Un drone intrus
- Un drone de surveillance
  - Un champ de perception
  - Plusieurs comportements possibles
    $k_1 \rightarrow \ldots \rightarrow k_2 \rightarrow \ldots \rightarrow k_1 \rightarrow \ldots$
    k_1 : « parcours en zigzags »
    k_2 : « couvrir la cible »



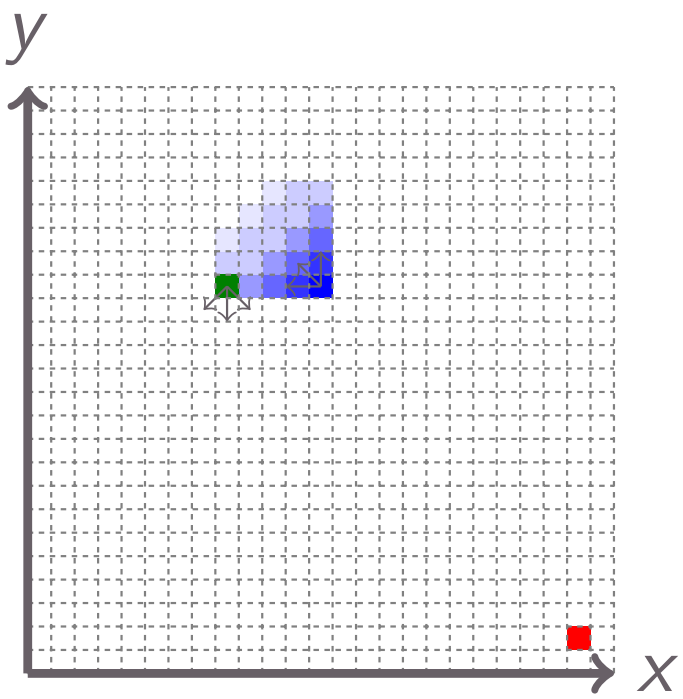
But :

- Compromis entre atteindre la cible et prendre le minimum de risque
- Graal : observabilité partielle sur l'état et le comportement de l'adversaire (satellite)

Éviter un drone de surveillance

Jeu stochastique : (grille stochastique)

- Une cible
- Un drone intrus
- Un drone de surveillance
  - Un champ de perception
  - Plusieurs comportements possibles
    $k_1 \rightarrow \ldots \rightarrow k_2 \rightarrow \ldots \rightarrow k_1 \rightarrow \ldots$
    k_1 : « parcours en zigzags »
    k_2 : « couvrir la cible »



But :

- Compromis entre atteindre la cible et prendre le minimum de risque
- Graal : observabilité partielle sur l'état et le comportement de l'adversaire (satellite)

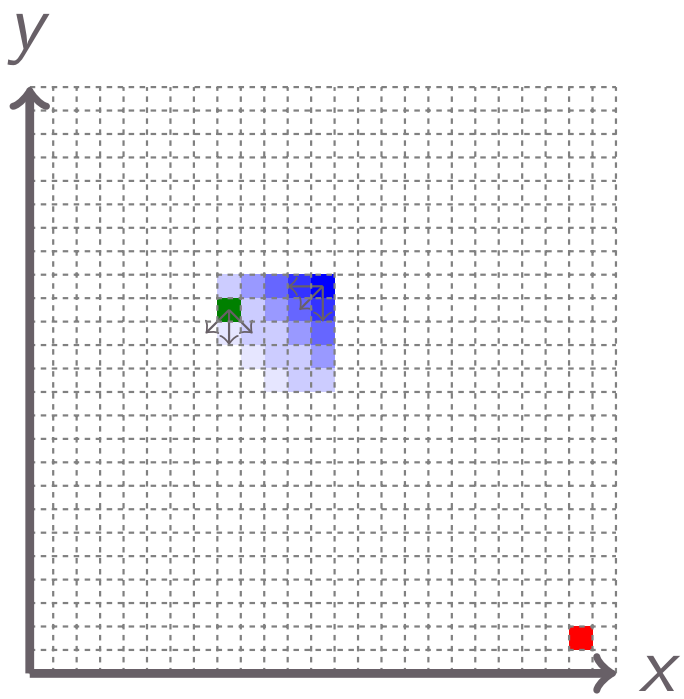
Éviter un drone de surveillance

Jeu stochastique : (grille stochastique)

- Une cible
- Un drone intrus
- Un drone de surveillance
  - Un champ de perception
  - Plusieurs comportements possibles
    $k_1 \rightarrow \ldots \rightarrow k_2 \rightarrow \ldots \rightarrow k_1 \rightarrow \ldots$
    k_1 : « parcours en zigzags »
    k_2 : « couvrir la cible »

But :

- Compromis entre atteindre la cible et prendre le minimum de risque
- Graal : observabilité partielle sur l'état et le comportement de l'adversaire (satellite)



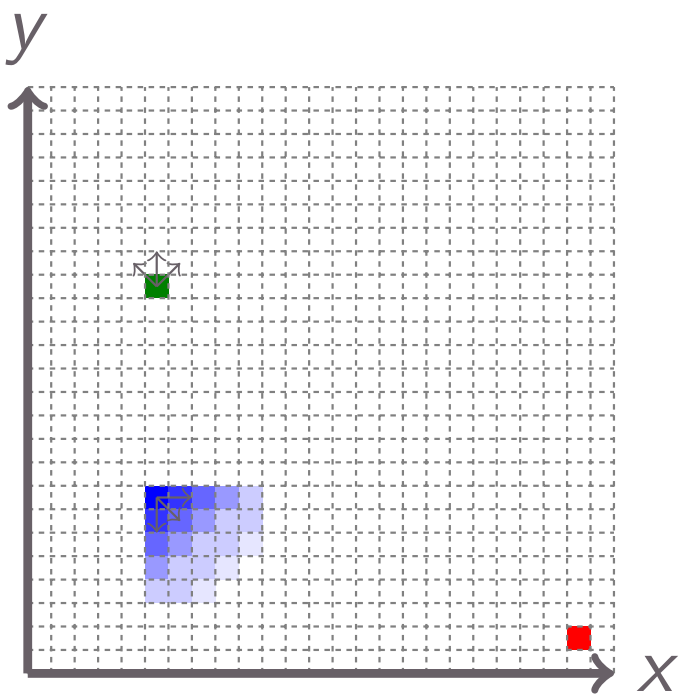
Éviter un drone de surveillance

Jeu stochastique : (grille stochastique)

- Une cible
- Un drone intrus
- Un drone de surveillance
  - Un champ de perception
  - Plusieurs comportements possibles
    $k_1 \rightarrow \ldots \rightarrow k_2 \rightarrow \ldots \rightarrow k_1 \rightarrow \ldots$
    k_1 : « parcours en zigzags »
    k_2 : « couvrir la cible »

But :

- Compromis entre atteindre la cible et prendre le minimum de risque
- Graal : observabilité partielle sur l'état et le comportement de l'adversaire (satellite)



Un jeu de sécurité

Jeux de sécurité

- Une équipe en défense qui protège des cibles (allocation de ressources)
- Une équipe en attaque qui doit choisir quelle cible attaquer
- Résolution par jeu de Stackelberg
- Network Security Game (difficile)

Intérêt

- Modélisation du point de vue de l'attaquant
- Restriction à un ensemble de comportements donnés pour la défense (non-optimale)
- Dynamique comportementale complexe

Un jeu de sécurité

MDP et Optimalité : Vorobeychik and Singh (2012)

« Pour tout jeu de Stackelberg stochastique à somme générale et à facteur d'atténuation, si le meneur suit une politique markovienne stationnaire alors il existe une politique déterministe markovienne et stationnaire qui est une politique best-response pour le suiveur. »

Contraintes

- Utilisation de MDP γ-pondérés
- Le patrouilleur doit suivre une politique markovienne stationnaire

3 Problèmes

1. Observabilité totale
2. Observabilité partielle sur la politique du patrouilleur
3. Observabilité partielle sur l'état du patrouilleur

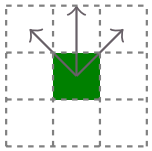
Sommaire

Notations

Symbole	Description	Domaine
X	Ensemble des lignes de la grille	$\{1, \ldots, R\}$
Y	Ensemble des colonnes de la grille	$\{1, \ldots, C\}$
O	Orientation des drones	$\{N, NE, E, SE, S, SW, W, NW\}$
S^I	Espace d'états de l'intrus	$X \times Y \times O$
S^P	Espace d'états du patrouilleur	$X \times Y \times O \times \ldots$
i	État de l'intrus	S^I
$i\|_{XY}$	État de l'intrus projeté sur XY	$X \times Y$
p	État du patrouilleur	S^P
Π^P	Politiques possibles du patrouilleur	$\{k_1, \ldots, k_N\}$
k	Politique courante du patrouilleur	Π^P
$\mathcal{T}^P$	Modèle de transition du patrouilleur	$S^P \times A \times S^P \mapsto [0, 1]$
τ	Position de la cible	$X \times Y$

Actions

- Modèle commun $A \Rightarrow$ simplicité
- Avancer d'une case dans une des 3 directions possibles : {GO-UP-LEFT, GO-STRAIGHT, GO-UP-RIGHT}



Équivalent à :

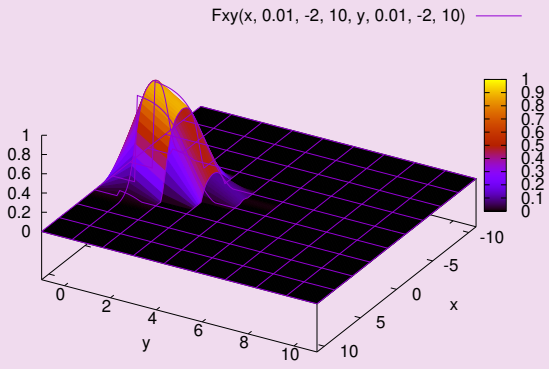
Orientation	Actions possibles
NORTH	{GO-NORTH, GO-NORTH-EAST, GO-NORTH-WEST}
NORTH-EAST	{GO-NORTH-EAST, GO-EAST, GO-NORTH}
EAST	{GO-EAST, GO-SOUTH-EAST, GO-NORTH-EAST}
SOUTH-EAST	{GO-SOUTH-EAST, GO-SOUTH, GO-EAST}
SOUTH	{GO-SOUTH, GO-SOUTH-WEST, GO-SOUTH-EAST}
SOUTH-WEST	{GO-SOUTH-WEST, GO-WEST , GO-SOUTH}
WEST	{GO-WEST, GO-NORTH-WEST, GO-SOUTH-WEST}
NORTH-WEST	{GO-NORTH-WEST, GO-NORTH, GO-WEST}

Risque vs Récompense

Risque

Fonction $D^P : S^I \times S^P \mapsto [0 \dots 1]$:

$$D^P_{ip} = Pr(\text{DETECTED} = T | i, p) = \frac{\mathcal{F}^P_{pi|XY}}{\sum_i \mathcal{F}^P_{pi|XY}}$$



Récompense

$$R^I_i = \begin{cases} 1 & \text{si } i|_{XY} == \tau, \\ 0 & \text{sinon.} \end{cases}$$

Modèle à observabilité totale

Principe

1. Agréger les différentes politiques du patrouilleur en formant un modèle de transitions global T^P
2. Gérer les transitions entre les comportements via un modèle de transitons inter-politiques θ^P

Modèle de transition agrégé

$T^P : S^P \times \Pi^P \times S^P \mapsto [0, 1]$
$T^P_{p'kp} = \mathcal{T}^P_{p'k_pp} = Pr(p'|p, k_p)$

Modèle de transition inter-politique

$\theta^P : \Pi^P \times \Pi^P \times S^I \times S^P$
$\theta^P_{k'kip} = Pr(k'|k, i, p)$

Modèle à observabilité totale

Fonction d'utilité

Compromis entre prise de risque et atteindre la cible :

$$U^I_{ip} = \beta R^I_i + (1-\beta)(1-D^P_{ip})$$

MDP augmenté résultant

Un MDP augmenté $\langle S, A, T^I, T^P, \theta^P, U^I \rangle$

- $S : S^I \times S^P \times \Pi^P$
- $T^I : S^I \times A \times S^I \mapsto [0,1]$

$$V^*_{ipk} = U^I_{ip} + \gamma \max_a \sum_{i'p'k'} T^I_{i'ai} T^P_{p'kp} \theta^P_{k'kip} V^*_{i'p'k'}.$$

Modèle à observabilité totale

Programme linéaire primal

$$\text{Minimize} \sum_{i,p,k} \mu_{ipk} V_{ipk}$$

Subject to

$$V_{ipk} \geq U^I_{ip} + \gamma \sum_{i',p',k'} T^I_{iai'} T^P_{pkp'} \theta_{k'kip} V_{i'p'k'}, \forall (a,i,p,k) \in A \times S^I \times S^P \times \Pi^P$$

$$\text{With} \sum_{i,p,k} \mu_{ipk} = 1$$

Programme linéaire dual

$$\text{Maximize} \sum_{i,p,k} \sum_{a} U^I_{ip} Q_{ipka}$$

Subject to

$$\sum_{a} Q_{i'p'k'a} - \gamma \sum_{ipk} \sum_{a} T^I_{iai'} T^P_{pkp'} \theta_{k'kip} Q_{ipka} = \mu_{i'p'k'}, \forall (i',p',k') \in S^I \times S^P \times \Pi^P$$

$$\text{With} \sum_{i,p,k} \mu_{ipk} = 1$$

Seconde version : Pseudo-minmax

Principe

1. On reprend les 2 principes précédents.
2. On construit une fonction de valeur pour le patrouilleur en considérant que le but du patrouilleur est de mettre en danger l'intrus. $\Rightarrow$ Surestimation du risque.
3. On retranche cette fonction de valeur à celle de l'intrus. $\Rightarrow$ pseudo-minmax.

Le patrouilleur

- Fonction de récompense $R^P_{pi} = D^P_{ip}$
- Fonction de valeur :

$$V^{P*}_{pik} = R^P_{pi} + \gamma \min_{a \in A} \sum_{i'p'k'} T^I_{i'ai}, T^P_{p'kp} \theta^P_{k'kip} V^{P*}_{p'i'k'}$$

Seconde version : Pseudo-minmax

MDP augmenté résultant

Un MDP augmenté $\langle S, A, T^I, T^P, \theta^P, U^I, V^{P*} \rangle$

- $U^I_{ip} = R^I_i$

$$V^*_{ipk} = \beta \Big[U^I_{ip} + \gamma \max_a \sum_{i'p'k'} T^I_{i'ai} T^P_{p'kp} \theta^P_{k'kip} V^*_{i'p'k'} \Big] - (1-\beta) V^{P*}_{pik}$$

Programme Linéaire

Adaptation du précédent.

Question

Nash-optimalité ?

Observabilité partielle sur la politique du patrouilleur

Principe

1. On maintient une croyance sur la politique suivie par la défense
2. On considère que les couples (i, p) sont des observations sur la politique utilisée par le patrouilleur.

Belief MDP

Un BMDP augmenté $\langle S, A, T^I, T^P, \theta^P, U^I, \zeta_\pi \rangle$

- $S : S^I \times S^P \times B^P_\pi$
- $b'_\pi = \zeta_\pi(i, p, b_\pi)$
  $b'_\pi(k') = Pr(k'|i, p, b_\pi) = \sum_k b_\pi(k)\theta^P_{k'kip}$

$$V^*_{ipb_\pi} = U^I_{ip} + \gamma \max_a \sum_{i'p} T^I_{i'ai} \left[\sum_k T^P_{p'kp} b_\pi(k) \right] V^*_{i'p'\zeta_\pi(i,p,b_\pi)}.$$

Observabilité partielle sur l'état du patrouilleur

Principe

1. On maintient une seconde croyance sur l'état du patrouilleur
2. Un oracle (satellite ?) donne des observations sur l'état du patrouilleur

Belief MDP

Un BMDP augmenté $\langle S, A, T^I, T^P, \theta^P, U^I, \Omega, TO, \zeta_\pi, \zeta_s \rangle$

- $S : S^I \times B_s^P \times B_\pi^P$
- $U : U^I_{ib_s} = \beta \left[1 - \sum_p b_s(p) D^P_{ip} \right] + (1-\beta) R^I_i$
- $\Omega : X \times Y \times O$
- $TO : \Omega \times S^P \times \Pi^P \Rightarrow Pr(o|p', k)$
- $\zeta_s : \Omega \times B_s^P \times B_\pi^P \Rightarrow Pr(p'|o, b_s, b_\pi), \forall p'$

$$V^*_{ib_s b_\pi} = U^I_{ib_s} + \gamma \max_a \sum_{i'} T^I_{i'ai} \sum_{op'k} TO_{op'k} \sum_p T^P_{p'kp} b_\pi(k) b_s(p) V^*_{i'\zeta_s(o,b_s,b_\pi)\zeta_\pi(i,p,b_\pi)}.$$

Observabilité partielle sur l'état du patrouilleur

Algorithme 1 : Adaptation de l'algorithme *POMCP*

Donnees : i, h $\mathcal{G}$, $\mathcal{T}$, $\pi_{rollout}$, γ, ϵ, c
Resultat : $\pi^* = Search(i, h)$

definir *Search*$(i,\ h)$**:**
- **répéter**
  - **si** $h = \emptyset$**:**
    - $p \sim I_s$
    - $k \sim I_\pi$
  - **sinon:**
    - $p \sim \mathcal{B}_s(h)$
    - $k \sim \mathcal{B}_\pi(h)$
  - $Simultate(i, p, k, h, 0)$
- **jusqu'à** *Timeout()*
- **retourner** $\underset{a}{argmax}\ V(ha)$

definir *Rollout*$(i,\ p,\ k,\ h,\ depth)$**:**
- **si** $\gamma^{depth} < \epsilon$**:**
  - **retourner** 0
- $a \sim \pi_{rollout}(h, .)$
- $i', p', k', o, u \sim \mathcal{G}(i, p, k, a)$
- **retourner** $u +$ $\gamma.Rollout(i', p', k', haio, depth+1)$

definir *Simulate*$(i,\ p,\ k,\ h,\ depth)$**:**
- **si** $\gamma^{depth} < \epsilon$**:**
  - **retourner** 0
- **si** $h \notin \mathcal{T}$**:**
  - **pour** a **dans** A'**:**
    - $\mathcal{T}(ha) \leftarrow (N_{init}(ha), V_{init}(ha), \emptyset, \emptyset)$
  - **retourner** $Rollout(i, p, k, h, depth)$
- $a = \underset{b \in A'}{argmax}\ V(hb) + c\sqrt{\frac{\log N(h)}{N(hb)}}$
- $i', p', k', o, u \sim \mathcal{G}(i, p, k, a)$
- $U \leftarrow u +$ $\gamma.Simulate(i', p', k', haio, depth+1)$
- $\mathcal{B}_s(h) \leftarrow \mathcal{B}_s(h) \cup \{p\}$
- $\mathcal{B}_\pi(h) \leftarrow \mathcal{B}_\pi(h) \cup \{k\}$
- $N(h) \leftarrow N(h) + 1$
- $N(ha) \leftarrow N(ha) + 1$
- $V(ha) \leftarrow V(ha) + \frac{U - V(ha)}{N(ha}$
- **retourner** U

Sommaire

Dispositif

Instances

- Environnement : $(9/10, 5/100)$
- 2 comportements pour patrouilleur :
  1. Patrouille en zigzags (boustrophédon)
  2. Protéger la cible
  3. Transitions : probabilité de détection > 0.3

$\Rightarrow$ Taille espace d'état : $|S| = 272n^5$

$\Rightarrow$ Nombre de transitions : $54\,|S|$

Sélection

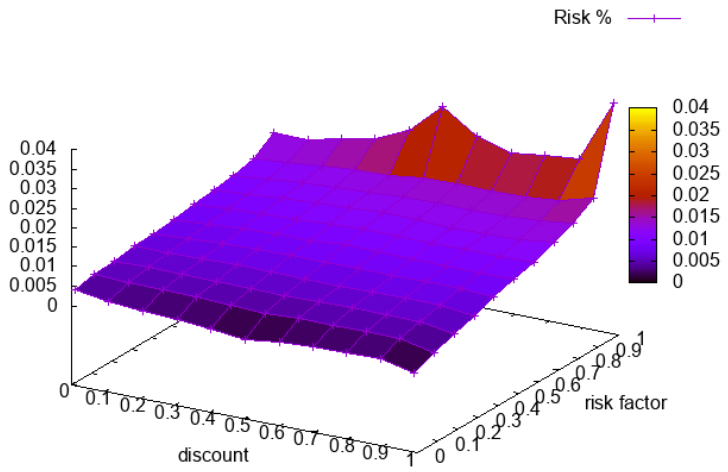
- *GridSearch* sur γ et β
- $<$ *risque* et $>$ *cible*

Évaluation

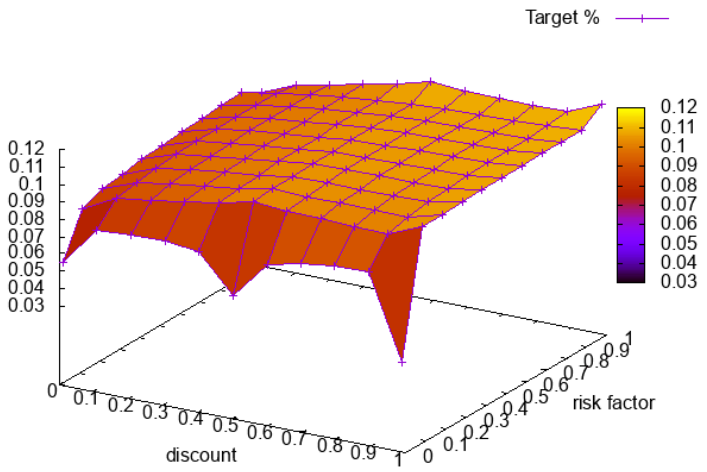
- On simule pendant 30 min :
  1. %interceptions
  2. %risque
  3. %cible
  4. %utilité

Sélection

Grille 8×8 :



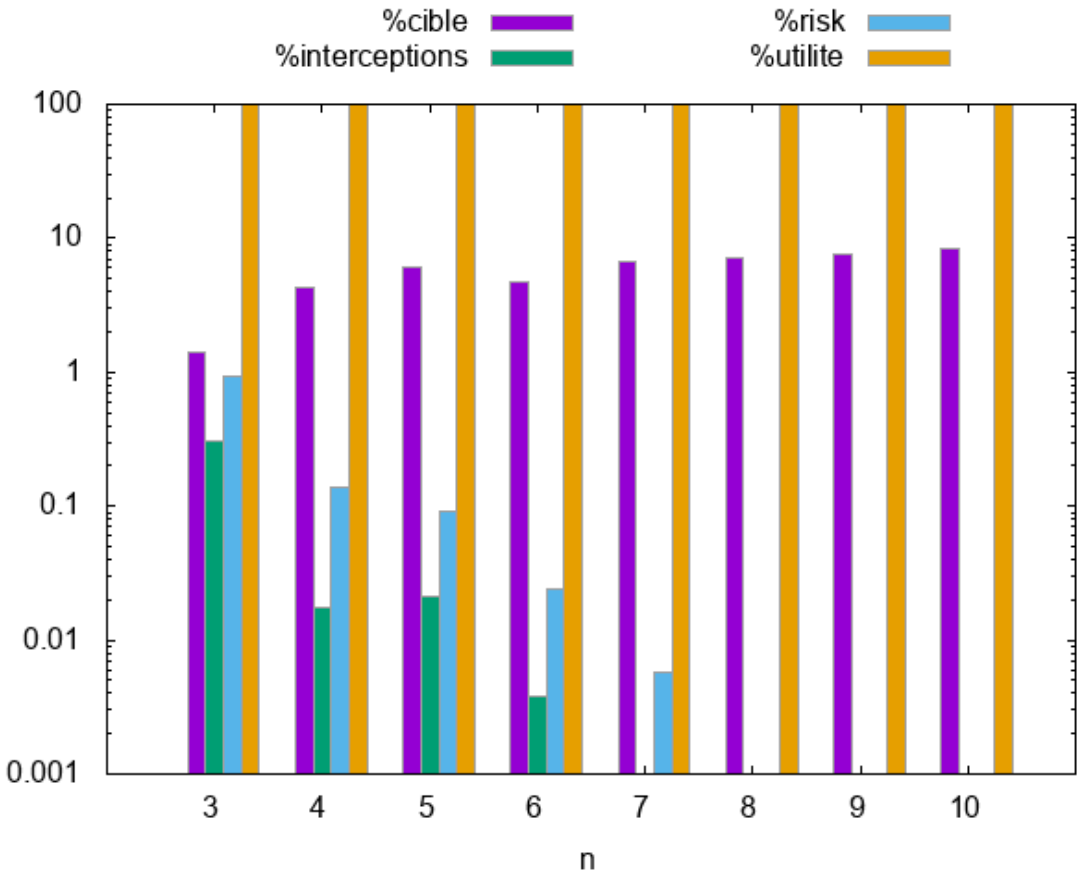
Risque(%) en fonction des paramètres libres



Cible (%) en fonction des paramètres libres

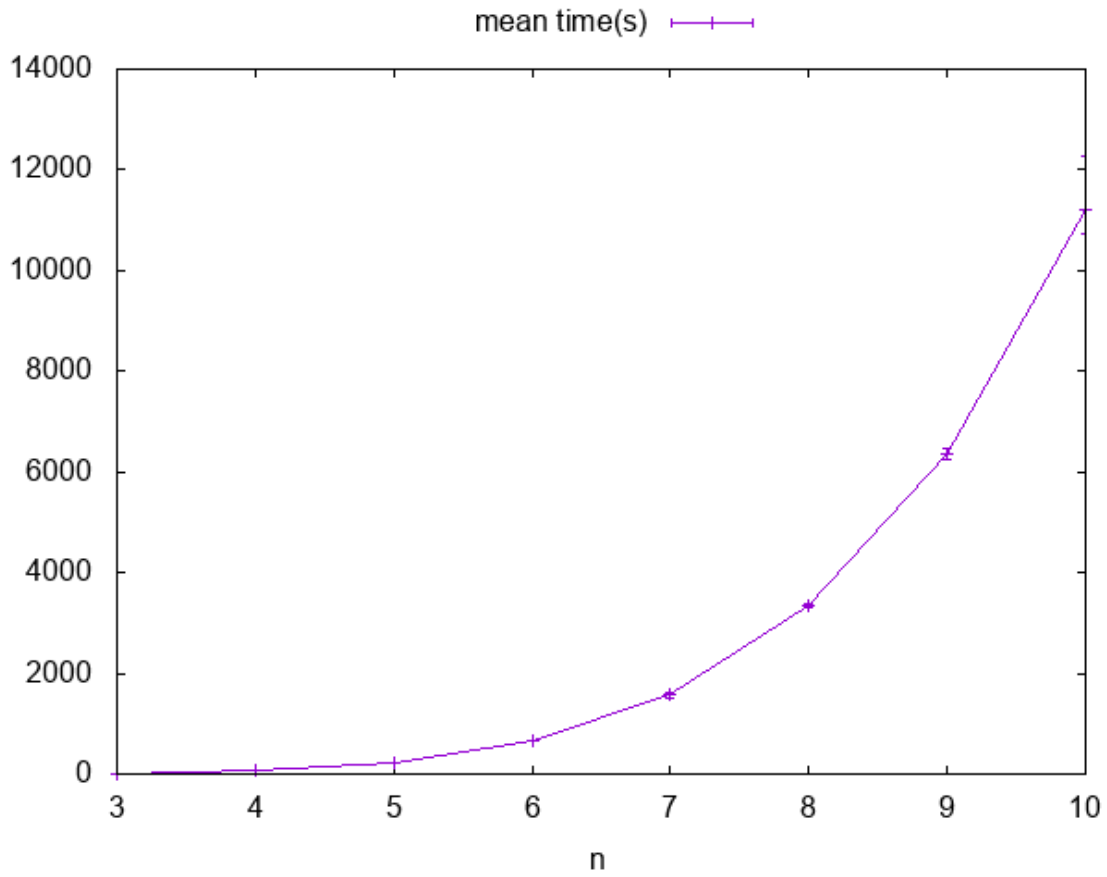
$\Rightarrow$ Compromis ($\gamma = 0.99, \beta = 1e-4$).

Métriques



Métriques, modèle observabilité complête

Temps de planification



Temps de planification, modèle observabilité complête

Sommaire

Conclusion

- Première modélisation « simple » du problème
- Résultats encourageants pour l'observabilité totale
- Bémol : fonction de valeur non-convexe
- Propose un algorithme pour l'observabilité partielle
- Implémentation + expériences à compléter
- Décomposition structure $\Rightarrow$ passage à l'échelle

Questions ?