Synthèse Multi-objectifs dans les Processus Décisionnels de Markov

Florent Delgrange

Mab2 Sciences Informatiques

UMONS Université de Mons

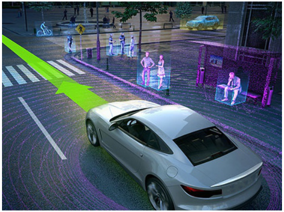
Faculté des Sciences

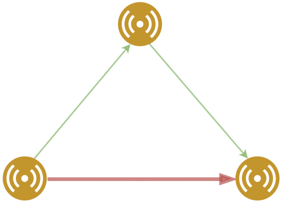
Année académique 2017-2018

Vérification et synthèse

- Systèmes réactifs dans des environnements stochastiques
  → *exemples* : protocoles de communication, réseaux de capteurs, voitures autonomes, etc.

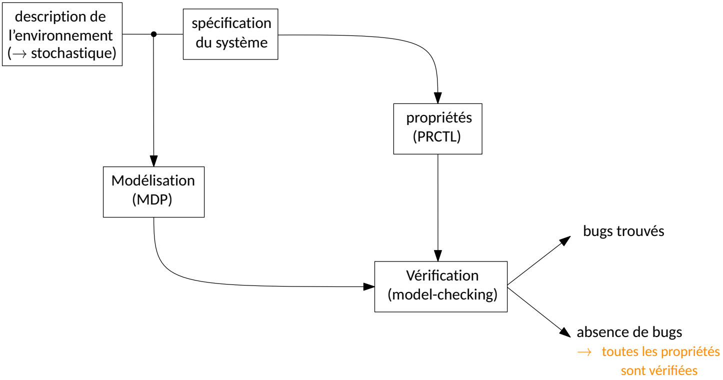
- **Vérification** de l'**exactitude** du comportement de tels systèmes
- **Synthèse** de **stratégies** satisfaisant des **objectifs** dans de tels systèmes
  → **Décisif !**
  - absence de bugs
  - sécurité





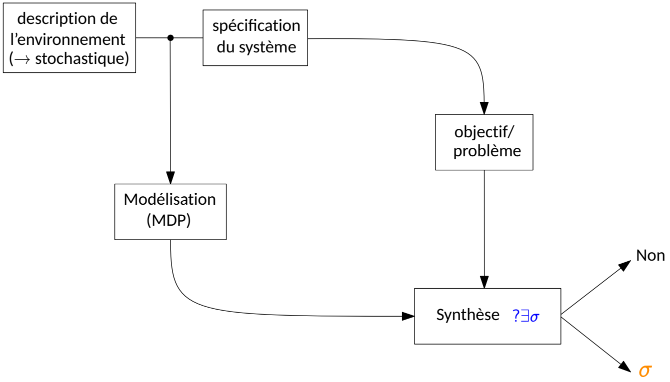
Vérification formelle

- Le *testing* démontre la présence de bugs, mais pas leur absence !
- Les algorithmes de *model-checking* permettent d'automatiquement **vérifier formellement** des *propriétés* dans un système



Synthèse de stratégie

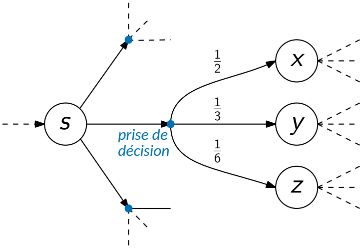
- Construire la stratégie satisfaisant un problème dans le système



But du mémoire

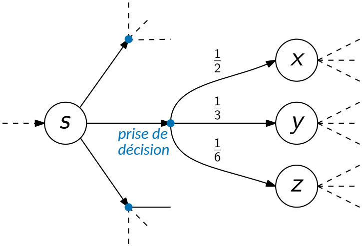
- Synthèse de **stratégies** satisfaisant des problèmes **mono-objectifs** de **plus court chemin stochastique**
- Synthèse de **stratégies** satisfaisant ***simultanément*** plusieurs objectifs en milieu **stochastique**
- Étudier ***PRCTL***, une logique permettant d'**exprimer des propriétés** pour les modèles probabilistes ainsi que ses algorithmes de **model-checking**
- Apporter une contribution à Storm
  - ⇝ Model-checker probabiliste en développement
  - ⇝ Implémente des algorithmes récents de model-checking multi-objectif
  - ⇝ Étude de l'outil (algorithmes, articles, etc.)

Processus décisionnel de Markov (MDP)



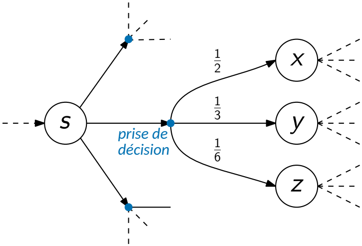
- *Système* ***stochastique***
- permet de modéliser à la fois des **situations probabilistes** et **non-déterministes** (i.e., nécessitant des prises de décision)
- peut être enrichi avec une **fonction de poids**, permettant de pondérer le **coût** de chaque décision

Processus décisionnel de Markov (MDP)



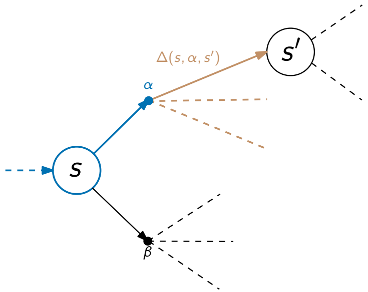
- S est un ensemble fini d'états et A est un ensemble fini d'actions,
  → $\forall s \in S,\ A(s) \subseteq A$ est l'ensemble des *actions activées* de s,

Processus décisionnel de Markov (MDP)



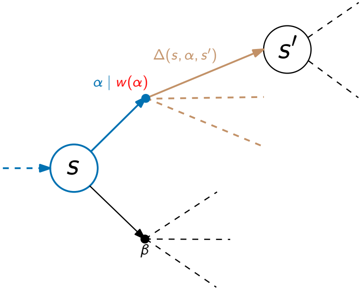
- S est un ensemble fini d'états et A est un ensemble fini d'actions,
  - $\rightarrow$ $\forall s \in S,\ A(s) \subseteq A$ est l'ensemble des *actions activées* de s,
- $\Delta : S \times A \times S \rightarrow [0, 1] \cap \mathbb{Q}$ est une fonction probabiliste de transition,

Processus décisionnel de Markov (MDP)



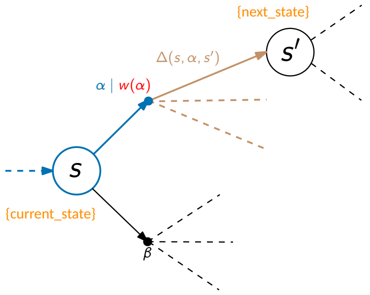
- S est un ensemble fini d'états et A est un ensemble fini d'actions,
  - $\longrightarrow$ $\forall s \in S,\ A(s) \subseteq A$ est l'ensemble des *actions activées* de s,
- $\Delta : S \times A \times S \to [0, 1] \cap \mathbb{Q}$ est une fonction probabiliste de transition,

Processus décisionnel de Markov (MDP)



- S est un ensemble fini d'états et A est un ensemble fini d'actions,
  - $\longrightarrow$ $\forall s \in S,\ A(s) \subseteq A$ est l'ensemble des *actions activées* de s,
- $\Delta : S \times A \times S \rightarrow [0, 1] \cap \mathbb{Q}$ est une fonction probabiliste de transition,
- $w : A \rightarrow \mathbb{N}_0$ est une fonction de pondération,

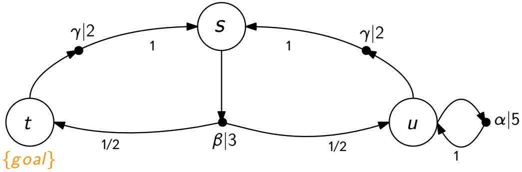
Processus décisionnel de Markov (MDP)



- S est un ensemble fini d'états et A est un ensemble fini d'actions,
  - $\longrightarrow$ $\forall s \in S,\ A(s) \subseteq A$ est l'ensemble des *actions activées* de s,
- $\Delta : S \times A \times S \rightarrow [0, 1] \cap \mathbb{Q}$ est une fonction probabiliste de transition,
- $w : A \rightarrow \mathbb{N}_0$ est une fonction de pondération,
- AP est un ensemble de propositions atomiques et $L : S \rightarrow AP$ est une fonction de *labelling* ou d'*étiquetage* d'états
  - $\longrightarrow$ utilisé pour la vérification du modèle par *model-checking*

Processus décisionnel de Markov (MDP)

Exemple

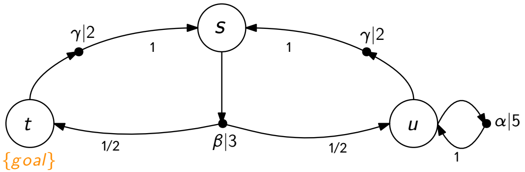


- $S = \{s, t, u\}$
- $A = \{\alpha, \beta, \gamma\}$
- $L(t) = \{goal\}$
- $A(s) = \{\beta\}, A(t) = \{\gamma\}, A(u) = \{\alpha, \beta\}$
- $w(\alpha) = 5, w(\beta) = 3, w(\gamma) = 2$
- $\Delta(s_0, \beta, s_1) = \Delta(s_0, \beta, s_2) = \frac{1}{2}$

Chemin : $\pi = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \dots$ où $\Delta(s_i, \alpha_{i+1}, s_{i+1}) > 0, \forall i \in \mathbb{N}$

Processus décisionnel de Markov (MDP)

Exemple



- $S = \{s, t, u\}$
- $A = \{\alpha, \beta, \gamma\}$
- $L(t) = \{goal\}$
- $A(s) = \{\beta\}, A(t) = \{\gamma\}, A(u) = \{\alpha, \beta\}$
- $w(\alpha) = 5, w(\beta) = 3, w(\gamma) = 2$
- $\Delta(s_0, \beta, s_1) = \Delta(s_0, \beta, s_2) = \frac{1}{2}$

Chemin : $\pi = s \xrightarrow{\beta} t \xrightarrow{\gamma} s \xrightarrow{\beta} (u \xrightarrow{\alpha})^{\omega} \in Paths(s)$

Stratégies

Une *stratégie* σ choisit à chaque étape une action activée $\alpha \in A(s)$ de l'état courant s

→ résout le non-déterminisme

→ une stratégie peut utiliser…

- de la ***mémoire*** (finie) : choisit les actions en fonction d'une quantité d'informations finie récoltée dans le passé
- de l'***aléatoire*** : choisit l'action selon une distribution de probabilité sur $A(s)$

→ les stratégies les plus simples sont les stratégies *pures* (i.e., sans aléatoire) et sans mémoire $\rightsquigarrow$ $\sigma : S \to A$

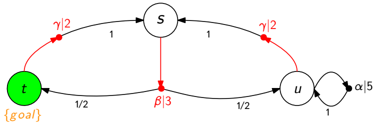
Chaîne de Markov induite par stratégie

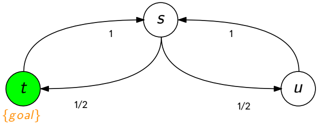
Une fois que la **stratégie contrôle les décisions du MDP**, ce dernier a un **comportement purement stochastique**

⟶ Une chaîne de Markov est induite par la stratégie

- On peut mesurer la probabilité des **évènements** $E \subseteq Paths(s)$ dans la chaîne de Markov induite par toute stratégie $\sigma \rightsquigarrow \mathbb{P}_s^{\sigma}(E)$
  - $\lozenge T = \{\pi = s_0 s_1 s_2 \cdots \in Paths(\mathcal{M}) \mid \exists n \in \mathbb{N},\ s_n \in T\}$

  $\rightsquigarrow$ **atteindre** T, où $T \subseteq S$ est un sous-ensemble d'**états cibles**
  - $?\exists \sigma,\ \mathbb{P}_s^{\sigma}(\lozenge\{t\}) = 1$





Problèmes de décisions

Existe-t-il une stratégie σ qui satisfait ...

Problèmes mono-objectifs

- **SR** : une haute probabilité d'accessibilité stochastique
- **SSP-E** : une bonne espérance du coût pour atteindre la cible
- **SSP-P** : une accessibilité à la cible avec un coût limité sous une haute probabilité
- **SP-G** : une garantie d'une borne en terme de coût pour atteindre la cible

Problèmes multi-objectifs

- **SSP-WE** : une bonne espérance du coût pour atteindre la cible sous une garantie de pire cas
- **MOSR** : plusieurs problèmes **SR** *simultanément*
- **SSP-PQ** : plusieurs problèmes **SSP-P** *simultanément*

Problèmes de décisions

Existe-t-il une stratégie σ qui satisfait ...

Problèmes mono-objectifs

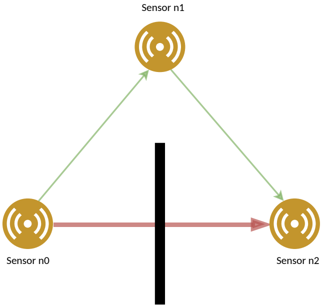
- **SR** : une haute probabilité d'accessibilité stochastique
- **SSP-E** : une bonne espérance du coût pour atteindre la cible
- **SSP-P** : une accessibilité à la cible avec un coût limité sous une haute probabilité
- **SP-G** : une garantie d'une borne en terme de coût pour atteindre la cible

Problèmes multi-objectifs

- **SSP-WE** : une bonne espérance du coût pour atteindre la cible sous une garantie de pire cas
- **MOSR** : plusieurs problèmes **SR** *simultanément*
- **SSP-PQ** : plusieurs problèmes **SSP-P** *simultanément*

Problème multi-objectif : exemple

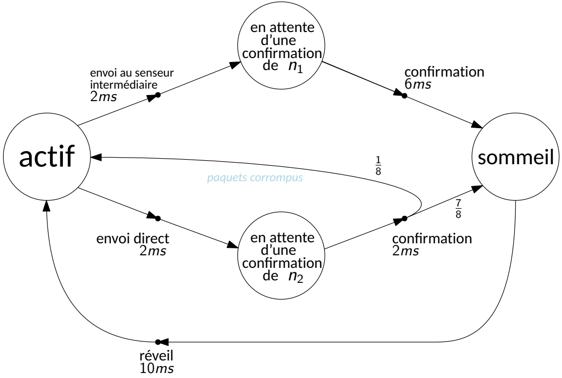
Communication entre noeuds dans un réseau de capteurs



- Un obstacle sépare n_0 et n_2
- Communication directe $n_0 \rightarrow n_2$
  - plus rapide que de passer par un noeud intermédiaire
  - risque de corruption des paquets envoyés (bruit)
- Communication indirecte : $n_0 \rightarrow n_1 \rightarrow n_2$
  - plus lent (n_1 doit attendre la confirmation de réception du paquet par n_2 et n_0 doit attendre la confirmation de n_1)
  - risque de corruption de paquet négligeable

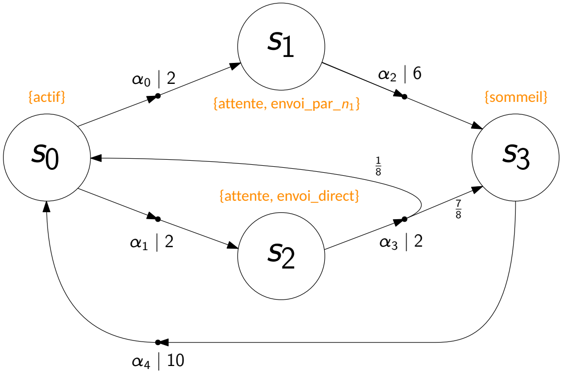
Problème multi-objectif : exemple

Communication entre noeuds dans un réseau de capteurs



Problème multi-objectif : exemple

Communication entre noeuds dans un réseau de capteurs



Plus court chemin stochastique

Somme tronquée

- Les problèmes de décision présentés concernent le problème de *plus court chemin stochastique*
- Calculer le coût des chemins $\rightsquigarrow$ ***somme tronquée*** :
  - soit $T \subseteq S$ un ensemble d'états cibles, et $\pi = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \cdots \in Paths(\mathcal{M})$

$$\mathsf{TS}^T(\pi) = \begin{cases} \sum_{i=1}^{n} w(\alpha_i) & \text{si } s_n \text{ est la première visite de } T, \\ +\infty & \text{si } T \text{ n'est jamais atteint dans } \pi \end{cases}$$

Plus court chemin stochastique

SSP-E : Bonne espérance de coût pour atteindre la cible

- $\mathbb{E}[X] = \sum_i x_i \cdot \mathbb{P}(X = x_i)$
  - ⇝ moyenne pondérée par les probabilités
- $\mathbb{E}_s^\sigma(\mathsf{TS}^T)$: coût moyen attendu pour atteindre T depuis s

$$?\exists \sigma,\ \mathbb{E}_s^\sigma(\mathsf{TS}^T) \leq \ell$$

Plus court chemin stochastique

SSP-E : Bonne espérance de coût pour atteindre la cible

- $\mathbb{E}[X] = \sum_i x_i \cdot \mathbb{P}(X = x_i)$
  - $\rightsquigarrow$ moyenne pondérée par les probabilités
- $\mathbb{E}_s^\sigma(\mathsf{TS}^T)$: coût moyen attendu pour atteindre T depuis s

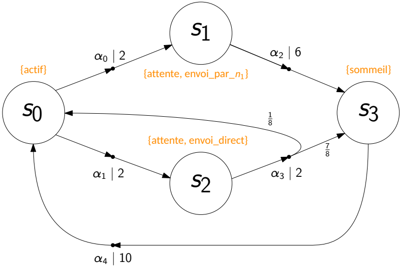
$$?\exists\sigma,\ \mathbb{E}_s^\sigma(\mathsf{TS}^T) \leq \ell$$

- Peut être résolu par **programmation linéaire** en temps **polynomial en la taille du modèle**
- Requiert une stratégie **pure** et **sans mémoire**

Plus court chemin stochastique

SSP-E : Bonne espérance de coût pour atteindre la cible

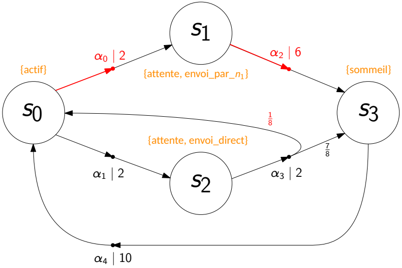
$$?\exists\sigma,\ \mathbb{E}_s^{\sigma}(\mathsf{TS}^{\,\text{sommeil}}) \leq 6$$



Plus court chemin stochastique

SSP-E : Bonne espérance de coût pour atteindre la cible

$$?\exists\sigma,\ \mathbb{E}_s^{\sigma}(\mathsf{TS}^{\,\text{sommeil}}) \leq 6$$

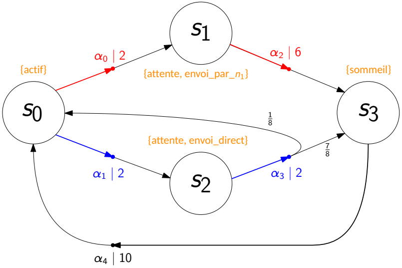


- $\mathbb{E}_{s_0}^{\sigma}(\mathsf{TS}^{\,\text{sommeil}}) = 8$

Plus court chemin stochastique

SSP-E : Bonne espérance de coût pour atteindre la cible

$$?\exists \sigma,\ \mathbb{E}_s^{\sigma}(\mathrm{TS}^{\text{sommeil}}) \leq 6$$

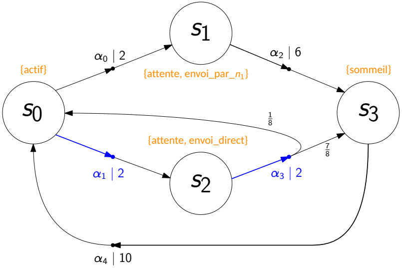


- $\mathbb{E}_{s_0}^{\sigma}(\mathrm{TS}^{\text{sommeil}}) = 8$
- $\mathbb{E}_{s_0}^{\sigma}(\mathrm{TS}^{\text{sommeil}}) = 4.57$

Plus court chemin stochastique

SSP-E : Bonne espérance de coût pour atteindre la cible

$$?\exists \sigma,\ \mathbb{E}^{\sigma}_{s}(\mathrm{TS}^{\,\text{sommeil}}) \leq 6$$



- $\cancel{\mathbb{E}^{\sigma}_{s_0}(\mathrm{TS}^{\,\text{sommeil}}) = 8} > 6$
- $\mathbb{E}^{\sigma}_{s_0}(\mathrm{TS}^{\,\text{sommeil}}) = 4.57 \leq 6$

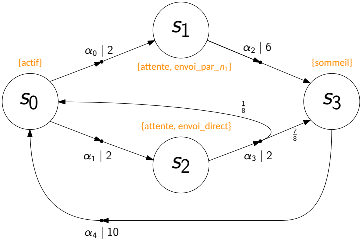
Bonne espérance sous un pire cas

SSP-WE : worst case expectation

- SSP-WE : **assurer** une **garantie** en terme de coût pour **atteindre la cible** tout en ayant une **bonne espérance** pour atteindre la cible

$$?\exists \sigma,\ \forall \pi \in Paths^{\sigma}(s),\ \mathrm{TS}^{T}(\pi) \leq \ell_1 \ \wedge\ \mathbb{E}_s^{\sigma}(\mathrm{TS}^{T}) \leq \ell_2$$

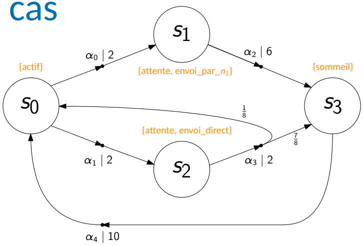
- Complexité en temps **pseudo-polynomiale** en ℓ_1
- Requiert une stratégie à **mémoire** finie



- Assurer un duty cycle de $12\ ms$
- Bonne espérance pour atteindre sommeil en respectant ce duty cycle ?

Bonne espérance sous un pire cas

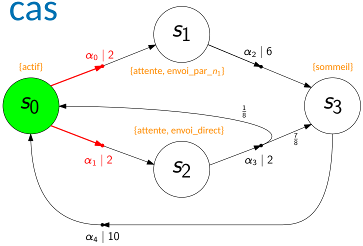
Algorithme



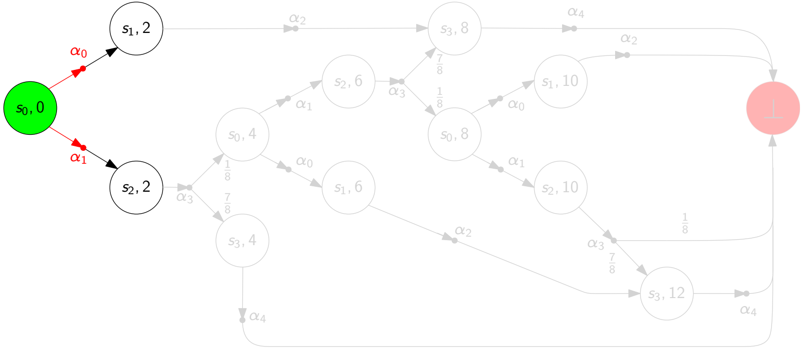
1. ***Déplier*** $\mathcal{M}$ jusque ℓ_1 depuis s
   - $\longrightarrow$ jusque 12 depuis s_0 (actif)

Bonne espérance sous un pire cas

Algorithme

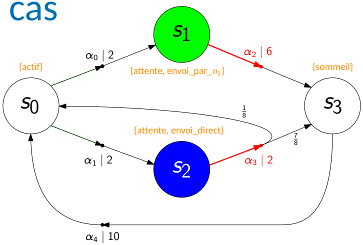


1. ***Déplier*** $\mathcal{M}$ jusque ℓ_1 depuis s
   - $\longrightarrow$ jusque 12 depuis s_0 (actif)

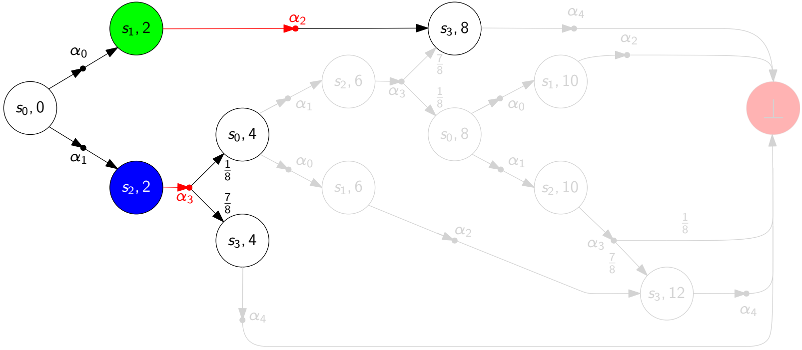


Bonne espérance sous un pire cas

Algorithme

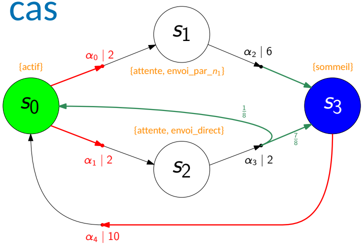


1. ***Déplier*** $\mathcal{M}$ jusque ℓ_1 depuis s
   $\longrightarrow$ jusque 12 depuis s_0 (actif)

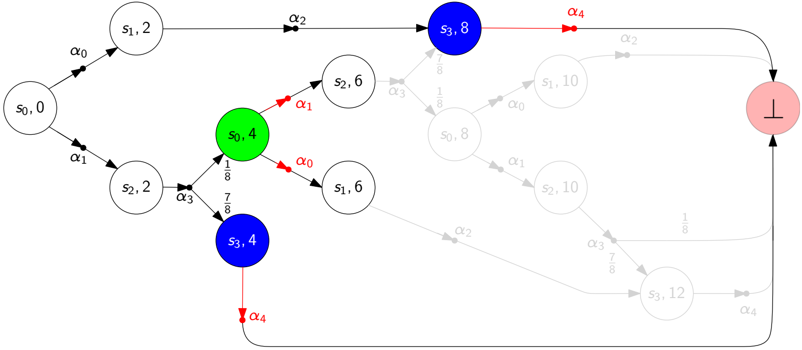


Bonne espérance sous un pire cas

Algorithme

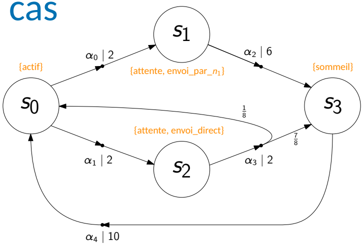


1. ***Déplier*** $\mathcal{M}$ jusque ℓ_1 depuis s
   - $\longrightarrow$ jusque 12 depuis s_0 (actif)

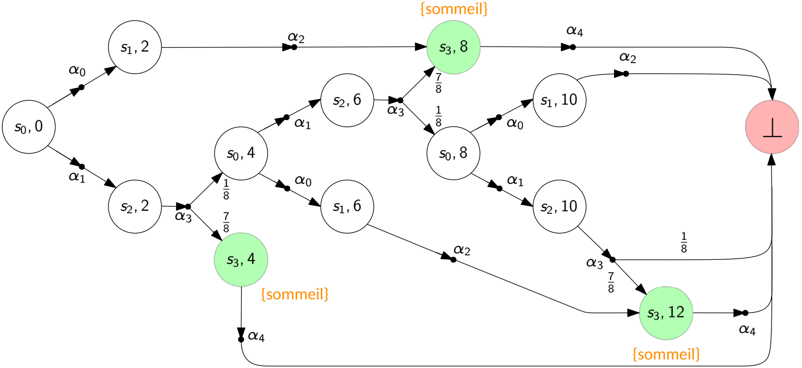


Bonne espérance sous un pire cas

Algorithme



1. ***Déplier*** $\mathcal{M}$ jusque ℓ_1 depuis s
   - $\longrightarrow$ jusque 12 depuis s_0 (actif)

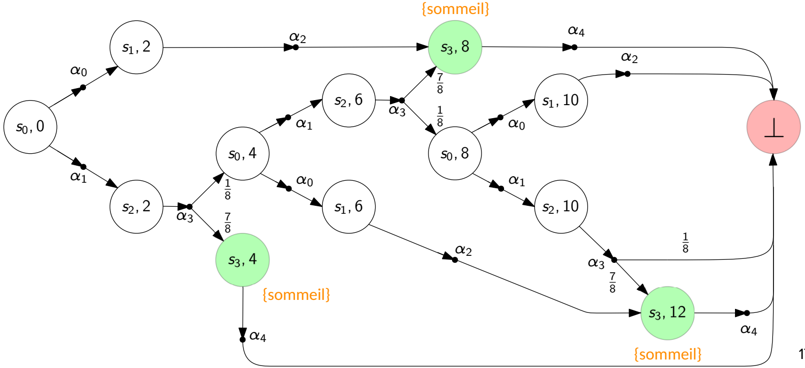


Bonne espérance sous un pire cas

Algorithme

1. ***Déplier*** $\mathcal{M}$ jusque ℓ_1 $\Longrightarrow$ temps pseudo-polynomial en ℓ_1

$$?\exists \sigma,\ \forall \pi \in \mathit{Paths}^{\sigma}(s),\ \mathsf{TS}^{\mathrm{sommeil}}(\pi) \leq 12 \wedge \mathbb{E}^{\sigma}_{s_0}(\mathsf{TS}^{\mathrm{sommeil}}) \leq 6$$

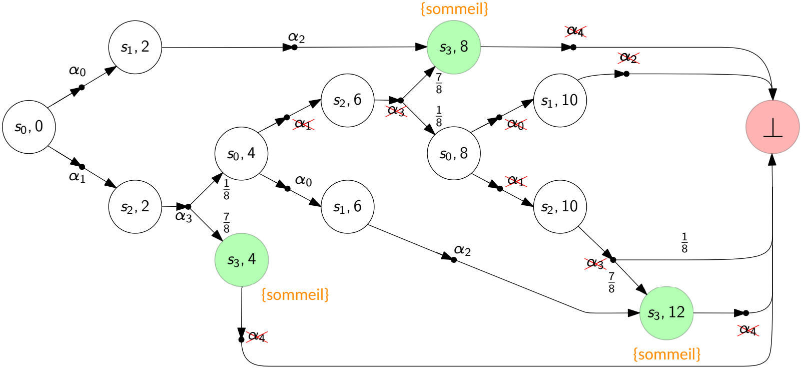


Bonne espérance sous un pire cas

Algorithme

2. Calculer l'ensembldе des actions ***safe*** $\mathbb{A}$ de $\mathcal{M}_{\ell_1}$

$$?\exists\sigma,\ \forall\pi \in Paths^{\sigma}(s),\ \mathsf{TS}^{\text{sommeil}}(\pi) \leq 12 \wedge \mathbb{E}^{\sigma}_{s_0}(\mathsf{TS}^{\text{sommeil}}) \leq 6$$

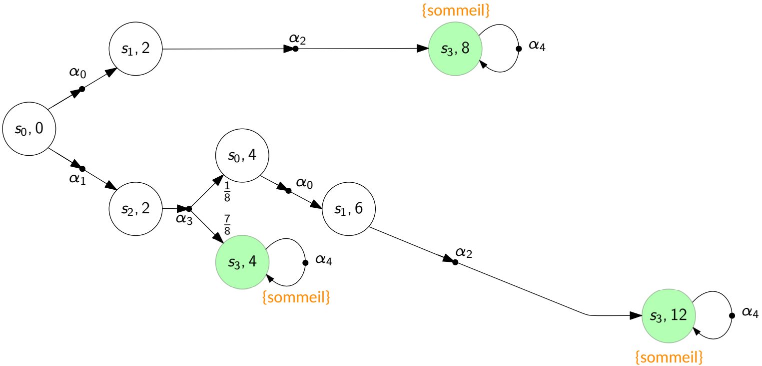


Bonne espérance sous un pire cas

Algorithme

3. ***Limiter*** le dépliage $\mathcal{M}_{\ell_1}$ aux actions ***safe*** de $\mathbb{A} \rightsquigarrow \mathcal{M}_{\ell_1}^{\mathbb{A}}$

$$\rightsquigarrow ?\exists \sigma^*,\ \mathbb{E}_{(s_0,0)}^{\sigma^*}(\mathsf{TS}^{\{(s_3,v)\,|\,v\leq 12\}}) \leq 6$$

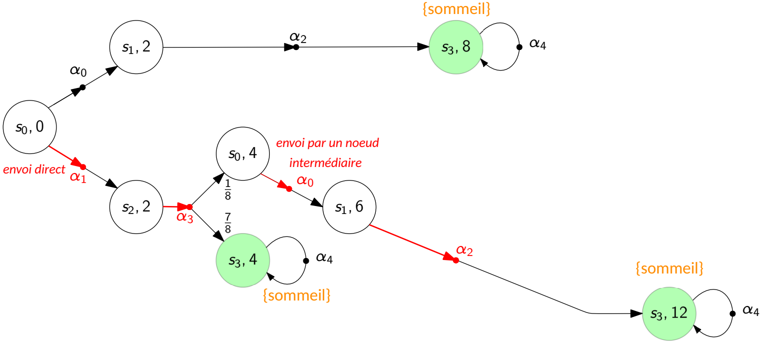


Bonne espérance sous un pire cas

Algorithme

4. Résoudre le problème de bonne espérance jusqu'aux cibles dans $\mathcal{M}^{\mathbb{A}}_{\ell_1}$

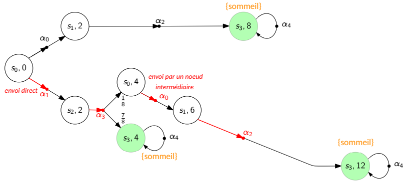
$$\rightsquigarrow \mathbb{E}^{\sigma^*}_{(s_0,0)}(\mathsf{TS}^{\{(s_3,v)\,|\,v\leq 12\}}) = \frac{7}{8}\cdot 4 + \frac{1}{8}\cdot 12 = 5$$



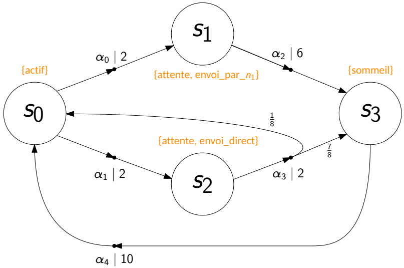
Bonne espérance sous un pire cas

Algorithme

$$\forall \pi \in Paths^{\sigma}(s),\ TS^{\text{sommeil}}(\pi) \leq 12 \land \mathbb{E}^{\sigma}{}_{s_0}(TS^{\text{sommeil}}) \leq 6$$



σ^* sans mémoire dans $\mathcal{M}^{\mathbb{A}}_{12}$



σ à mémoire finie dans $\mathcal{M}$

- **Stratégie optimale σ** : tester une fois un envoi direct et passer par le noeud n_1 si l'envoi direct est un échec.

Résultats

Problème	Temps	Stratégie	
		type	*mémoire*
SR (accessibilité)	$\mathrm{P}(\mathcal{M})$	pure	sans mémoire
SSP-E (bon coût moyen attendu)	$\mathrm{P}(\mathcal{M})$	**pure**	**sans mémoire**
SSP-P (requête percentile)	$\mathrm{P}(\mathcal{M}) \cdot \mathrm{P}_{ps}(\ell)$	pure	$\mathrm{P}_{ps}(\ell)$
SP-G (garantie de coût)	$\mathrm{P}(\mathcal{M})$	pure	sans mémoire
SSP-WE (garantie + bonne espérance)	$\mathrm{P}(\mathcal{M}) \cdot \mathrm{P}_{ps}(\ell)$	**pure**	$\mathrm{P}_{ps}(\ell)$
MOSR (accessibilité multiple + états cibles absorbants)	$\mathrm{P}(\mathcal{M})$	randomisée	sans mémoire
MOSR (accessibilité multiple)	$\mathrm{P}(\mathcal{M}) \cdot \mathrm{E}(\mathcal{Q})$	randomisée	$\mathrm{E}(\mathcal{Q})$
SSP-PQ (Multiple requêtes percentiles sur une seule dimension)	$\mathrm{P}(\mathcal{M}) \cdot \mathrm{P}_{ps}(\ell_{\max})$	randomisée	$\mathrm{P}_{ps}(\ell)$
SSP-PQ (Multiples requêtes percentiles sur multiple dimensions)	$\mathrm{P}(\mathcal{M}) \cdot \mathrm{E}(\mathcal{Q})$	randomisée	$\mathrm{E}(\mathcal{Q})$

Table – P : polynomial – P_{ps} : pseudo-polynomial – E : exponentiel

Perspectives

- **Problème d'explosion de l'espace d'état** : abstraction du système en utilisant les jeux stochastiques, exploration de l'espace d'état par machine learning
- **Stratégies compréhensibles** : stratégies "moins optimales" mais sans aléatoire
- **Pas de dépliage**
- **Autres fonctions de coût** : mean-payoff, discounted sum, etc.

→ Implémentation de nouveaux algorithmes dans Storm