

# Utilisation de croyances heuristiques pour la planification multi-agent dans le cadre des Dec-POMDP

Soutenance de thèse

Gabriel Corona

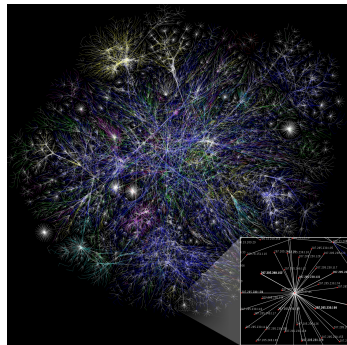
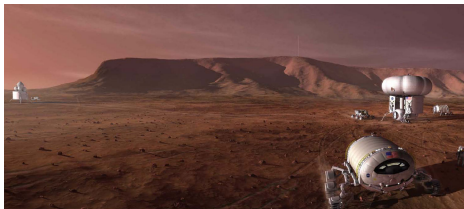
Loria

Équipe MAIA

11 avril 2011



# Contrôle décentralisé



- 1 État de l'art
  - Cadre formel
  - Programmation dynamique
  - Mémoire bornée
  
- 2 *Lookahead* approché
  - Principe
  - Résolution
  - Résultats
  
- 3 PSMBDP
  - Principe
  - Formulation
  - Résolution
  - Résultats
  
- 4 Conclusion

- 1 État de l'art
  - Cadre formel
  - Programmation dynamique
  - Mémoire bornée
  
- 2 *Lookahead* approché
  - Principe
  - Résolution
  - Résultats
  
- 3 PSMBDP
  - Principe
  - Formulation
  - Résolution
  - Résultats
  
- 4 Conclusion

## 1 État de l'art

### ■ Cadre formel

- Programmation dynamique
- Mémoire bornée

## 2 *Lookahead* approché

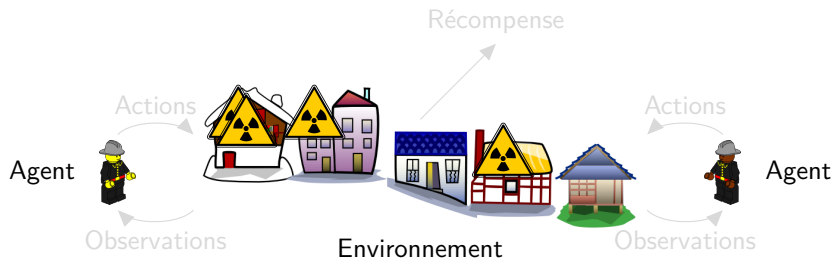
- Principe
- Résolution
- Résultats

## 3 PSMBDP

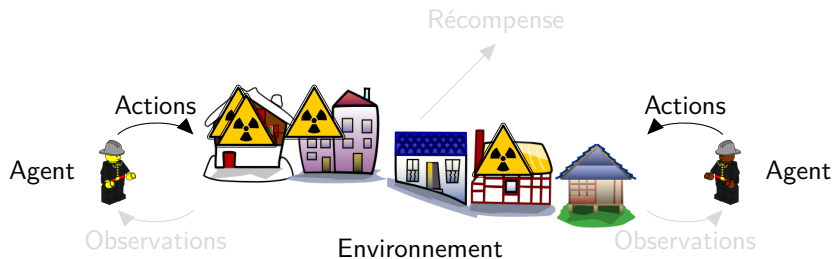
- Principe
- Formulation
- Résolution
- Résultats

## 4 Conclusion

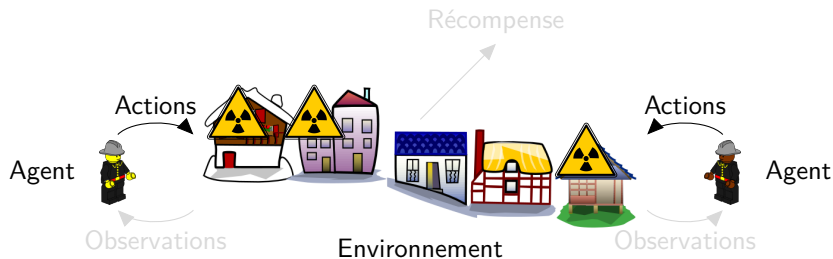
# Contrôle décentralisé



# Contrôle décentralisé

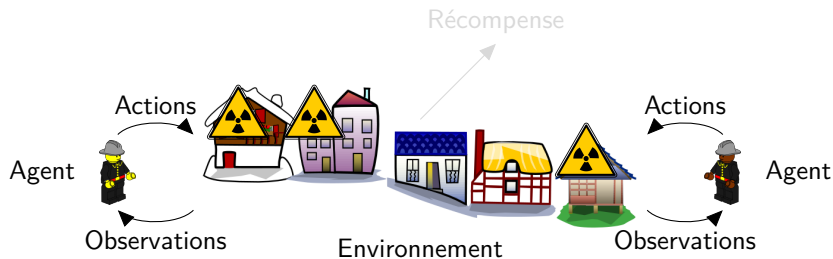


# Contrôle décentralisé

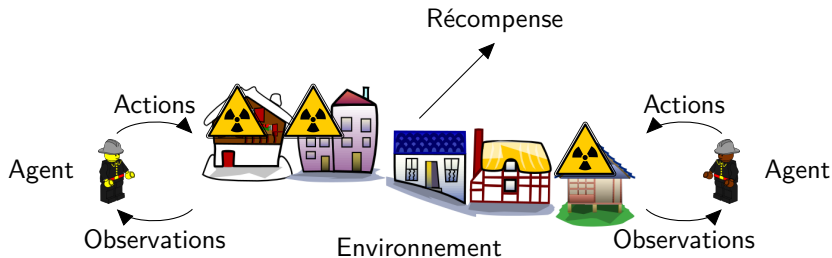




# Contrôle décentralisé



# Contrôle décentralisé



# Modèle Dec-POMDP

Decentralized Partially Observable Markov Decision Process [bernstein2000]

$$\mathcal{M} = \langle \mathcal{I}, \mathcal{S}, (\mathcal{A}_i)_{i \in \mathcal{I}}, \mathcal{T}, \mathcal{R}, (\Omega_i)_{i \in \mathcal{I}}, \mathcal{O} \rangle$$

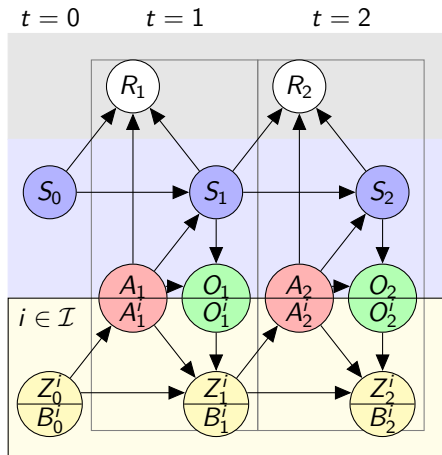
- $\mathcal{S}$ , ensemble des états
- $\mathcal{I}$ , ensemble des agents
- $\forall i \in \mathcal{I}, \mathcal{A}_i$ , ensemble des actions
- $\forall i \in \mathcal{I}, \Omega_i$ , ensemble des observations
- $\mathcal{T}(s'|s, a)$ , loi de transition
- $\mathcal{R}(s, a)$  récompenses immédiates
- $\mathcal{O}(o|s, a, s')$  loi d'observation

Problème : planification, horizon  $T$  fini

Objectif : maximiser les récompenses

$$\max E \left[ \sum_{t=1}^T R_t \right]$$

NEXP-difficile [bernstein2000]



# Croyance

Connaissance (bayésienne) de l'agent sur l'état du système



## Mono-agent

$$B_t(s) = Pr(S_t = s | B_0, A_1, O_1, \dots, A_t, O_t)$$

## Multi-agent

$$B_t^i(s, z_{-i}) = Pr(S_t = s, Z_t^{-i} = z_{-i} | B_0^i, A_1^i, O_1^i, \dots, A_t^i, O_t^i)$$

Raisonnement récursif : « je pense qu'il pense que je pense qu'il pense que je pense que je pense qu'il pense que ... »

# Croyance

Connaissance (bayésienne) de l'agent sur l'état du système



## Mono-agent

$$B_t(s) = Pr(S_t = s | B_0, A_1, O_1, \dots, A_t, O_t)$$

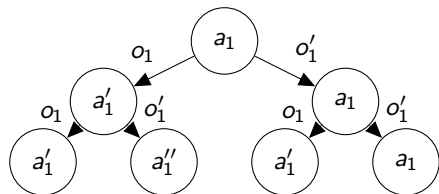
## Multi-agent

$$B_t^i(s, \mathbf{z}_{-i}) = Pr(S_t = s, \mathbf{Z}_t^{-i} = \mathbf{z}_{-i} | B_0^i, A_1^i, O_1^i, \dots, A_t^i, O_t^i)$$

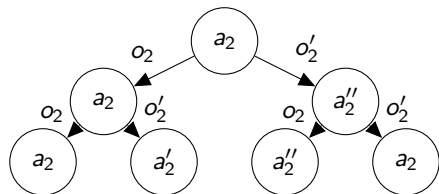
Raisonnement récursif : « je pense qu'il pense que je pense qu'il pense que je pense que je pense qu'il pense que ... »

# Arbres de politique

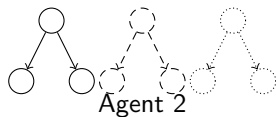
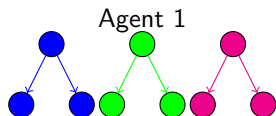
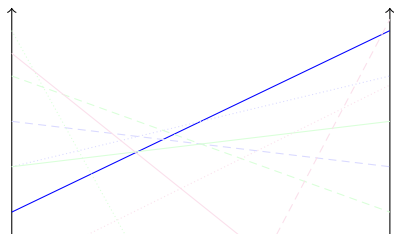
Agent 1



Agent 2



## Évaluation des politiques jointes



$$\Pr(s_1) = 1$$

$$\Pr(s_2) = 0$$

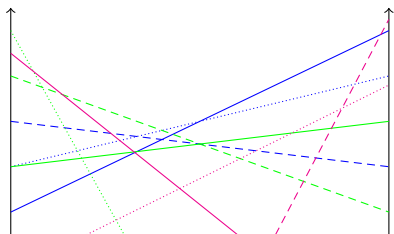
$$\Pr(s_1) = 0$$

$$\Pr(s_2) = 1$$

$$V_z(b) = E\left[\sum_{t'=t}^T R_{t'} | z, b\right] = \sum_s V_z(s) b(s)$$

$$V_{z_i}(b_i) = E\left[\sum_{t'=t}^T R_{t'} | z_i, b_i\right] = \sum_{s, z_{-i}} V_{(z_i, z_{-i})}(s) b(s, z_{-i})$$

## Évaluation des politiques jointes

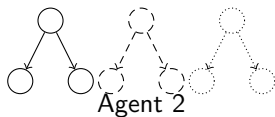
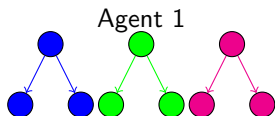


$$\Pr(s_1) = 1$$

$$\Pr(s_2) = 0$$

$$\Pr(s_1) = 0$$

$$\Pr(s_2) = 1$$



$$V_z(b) = E\left[\sum_{t'=t}^T R_{t'} | z, b\right] = \sum_s V_z(s) b(s)$$

$$V_{z_i}(b_i) = E\left[\sum_{t'=t}^T R_{t'} | z_i, b_i\right] = \sum_{s, z_{-i}} V_{(z_i, z_{-i})}(s) b(s, z_{-i})$$



## 1 État de l'art

- Cadre formel
- **Programmation dynamique**
- Mémoire bornée

## 2 *Lookahead* approché

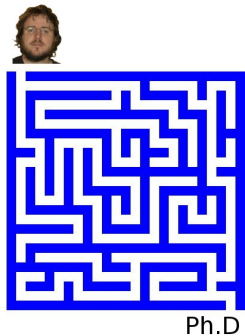
- Principe
- Résolution
- Résultats

## 3 PSMBDP

- Principe
- Formulation
- Résolution
- Résultats

## 4 Conclusion

# Programmation dynamique



[hansen2004]



Chaînage arrière



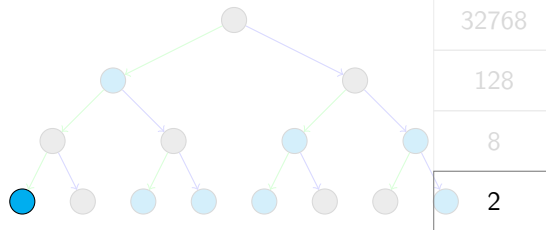
Chaînage avant



# Programmation dynamique

$$\mathcal{A}_i = \{a_i, a'_i\}, \Omega_i = \{o_i, o'_i\}$$

Énumération des politiques  
Évaluation des politiques jointes  
[hansen2004]

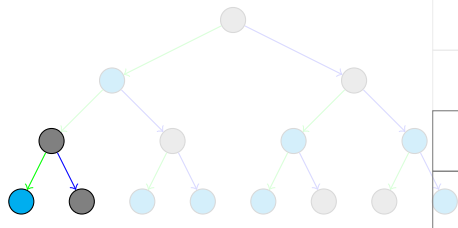


Mise à jour  
exhaustive

# Programmation dynamique

$$\mathcal{A}_i = \{a_i, a'_i\}, \Omega_i = \{o_i, o'_i\}$$

Énumération des politiques  
Évaluation des politiques jointes  
[hansen2004]



$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

$$2 \times 18^{38}$$

$$9 \times 10^9$$

$$2 \times 10^9$$

$$32768$$

$$128$$

$$8$$

$$2$$

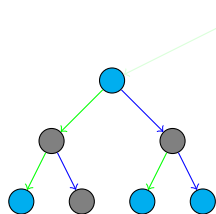
Mise à jour  
exhaustive



# Programmation dynamique

$$\mathcal{A}_i = \{a_i, a'_i\}, \Omega_i = \{o_i, o'_i\}$$

Énumération des politiques  
Évaluation des politiques jointes  
[hansen2004]



$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

$$2 \times 18^{38}$$

$$9 \times 10^9$$

$$2 \times 10^9$$

$$32768$$

$$128$$

$$8$$

$$2$$

Mise à jour  
exhaustive

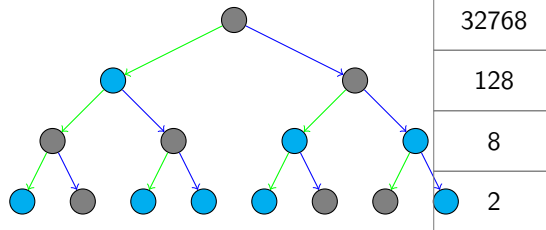


# Programmation dynamique

$$\mathcal{A}_i = \{a_i, a'_i\}, \Omega_i = \{o_i, o'_i\}$$

Énumération des politiques  
Évaluation des politiques jointes  
[hansen2004]

$$|\mathcal{A}_i|^{|\Omega_i|^T}$$



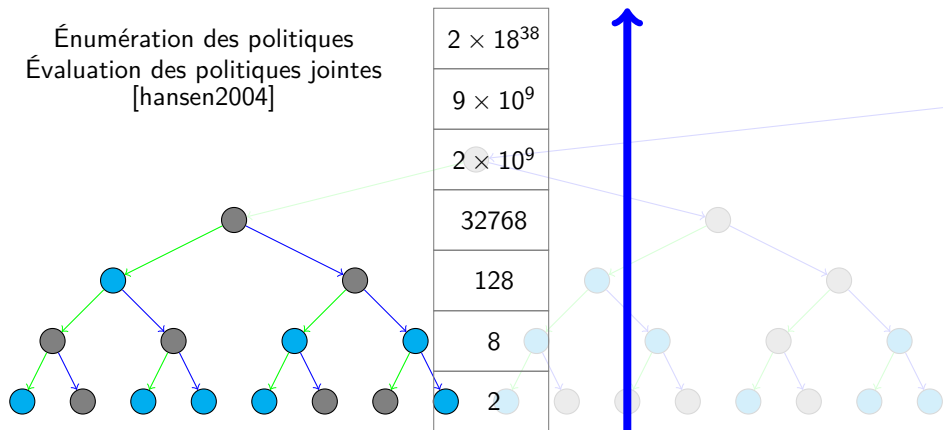
Mise à jour  
exhaustive

# Programmation dynamique

$$\mathcal{A}_i = \{a_i, a'_i\}, \Omega_i = \{o_i, o'_i\}$$

Énumération des politiques  
Évaluation des politiques jointes  
[hansen2004]

$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

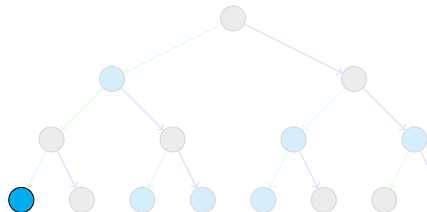


Mise à jour  
exhaustive

## Élagage

$$\mathcal{A}_i = \{a_i, a'_i\}, \Omega_i = \{o_i, o'_i\}$$

Supprimer les politiques inutiles  
 $\forall b_i, \exists z'_i \neq z_i, V_{z'_i}(b_i) \geq V_{z_i}(b_i)$   
 [hansen2004]



$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

|                    |
|--------------------|
| $2 \times 18^{38}$ |
| $9 \times 10^9$    |
| $2 \times 10^9$    |
| 32768              |
| 128                |
| 8                  |
| 2                  |

Mise à jour  
exhaustive

$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

|  |
|--|
| $2 \times 10^{19}$                       |
| <del><math>4 \times 10^{19}</math></del> |
| $4 \times 10^9$                          |
| <del><math>8 \times 10^9</math></del>    |
| 65536                                    |
| 131072                                   |
| 256                                      |
| 1024                                     |
| 16                                       |
| 32                                       |
| 4  |
| 8  |
| 2  |

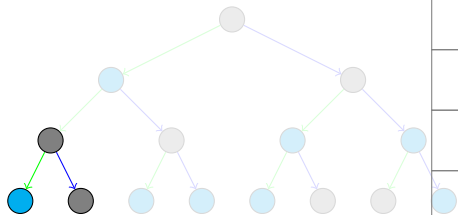
Élagage  
(exemple)



## Élagage

$$\mathcal{A}_i = \{a_i, a'_i\}, \Omega_i = \{o_i, o'_i\}$$

Supprimer les politiques inutiles  
 $\forall b_i, \exists z'_i \neq z_i, V_{z'_i}(b_i) \geq V_{z_i}(b_i)$   
 [hansen2004]



$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

|                    |
|--------------------|
| $2 \times 18^{38}$ |
| $9 \times 10^9$    |
| $2 \times 10^9$    |
| 32768              |
| 128                |
| 8                  |
| 2                  |

Mise à jour  
exhaustive

$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

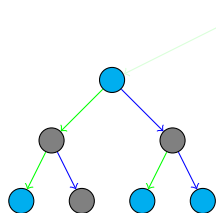
|                    |
|--------------------|
| $2 \times 10^{19}$ |
| $4 \times 10^{19}$ |
| $4 \times 10^9$    |
| $8 \times 10^9$    |
| 65536              |
| 131072             |
| 256                |
| 1024               |
| 16                 |
| 32                 |
| 4                  |
| 8                  |
| 2                  |

Élagage  
(exemple)

# Élagage

$$\mathcal{A}_i = \{a_i, a'_i\}, \Omega_i = \{o_i, o'_i\}$$

Supprimer les politiques inutilites  
 $\forall b_i, \exists z'_i \neq z_i, V_{z'_i}(b_i) \geq V_{z_i}(b_i)$   
 [hansen2004]



$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

$$2 \times 18^{38}$$

$$9 \times 10^9$$

$$2 \times 10^9$$

$$32768$$

$$128$$

$$8$$

$$2$$

Mise à jour  
exhaustive

$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

$$2 \times 10^{19}$$

$$4 \times 10^{19}$$

$$4 \times 10^9$$

$$8 \times 10^9$$

$$65536$$

$$131072$$

$$256$$

$$1024$$

$$16$$

$$32$$

$$4$$

$$8$$

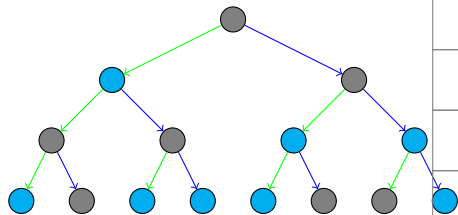
$$2$$

Élagage  
(exemple)

## Élagage

$$\mathcal{A}_i = \{a_i, a'_i\}, \Omega_i = \{o_i, o'_i\}$$

Supprimer les politiques inutiles  
 $\forall b_i, \exists z'_i \neq z_i, V_{z'_i}(b_i) \geq V_{z_i}(b_i)$   
 [hansen2004]



$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

$$2 \times 18^{38}$$

$$9 \times 10^9$$

$$2 \times 10^9$$

$$32768$$

$$128$$

$$8$$

$$2$$

Mise à jour  
exhaustive

$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

$$2 \times 10^{19}$$

$$4 \times 10^{19}$$

$$4 \times 10^9$$

$$8 \times 10^9$$

$$65536$$

$$131072$$

$$256$$

$$1024$$

$$16$$

$$32$$

$$4$$

$$8$$

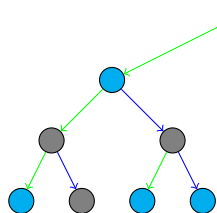
$$2$$

Élagage  
(exemple)

## Élagage

$$\mathcal{A}_i = \{a_i, a'_i\}, \Omega_i = \{o_i, o'_i\}$$

Supprimer les politiques inutiles  
 $\forall b_i, \exists z'_i \neq z_i, V_{z'_i}(b_i) \geq V_{z_i}(b_i)$   
 [hansen2004]



$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

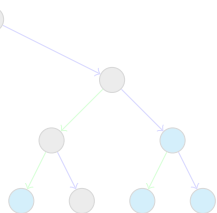
|                    |
|--------------------|
| $2 \times 18^{38}$ |
| $9 \times 10^9$    |
| $2 \times 10^9$    |
| 32768              |
| 128                |
| 8                  |
| 2                  |

Mise à jour  
exhaustive

$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

|  |
|--|
| $2 \times 10^{19}$                       |
| <del><math>4 \times 10^{19}</math></del> |
| $4 \times 10^9$                          |
| <del><math>8 \times 10^9</math></del>    |
| 65536                                    |
| 131072                                   |
| 256                                      |
| 1024                                     |
| 16                                       |
| 32                                       |
| 4  |
| 8  |
| 2  |

Élagage  
(exemple)



# Accessibilité

Élagage de  $z_i$  :

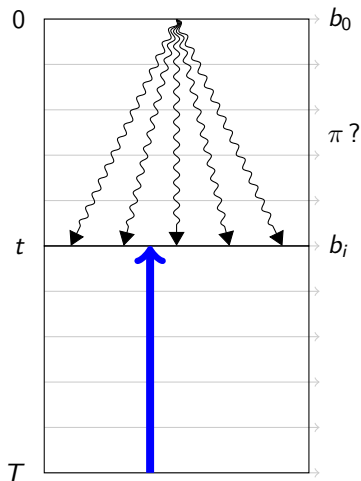
$$\forall b_i, \exists z'_i \neq z_i, V_{z'_i}(b_i) \geq V_{z_i}(b_i)$$

Prendre en compte l'accessibilité [szer2006c]  
[seuken2007a] [dibangoye2008]

Dépend de la politique  $\pi$  de 0 à  $t$  (inconnue)

PBDP [szer2006c] (*Point Based Dynamic Programming*) :

- échantillonner  $\pi$
- échantillonner  $b_i$



## 1 État de l'art

- Cadre formel
- Programmation dynamique
- Mémoire bornée

## 2 *Lookahead* approché

- Principe
- Résolution
- Résultats

## 3 PSMBDP

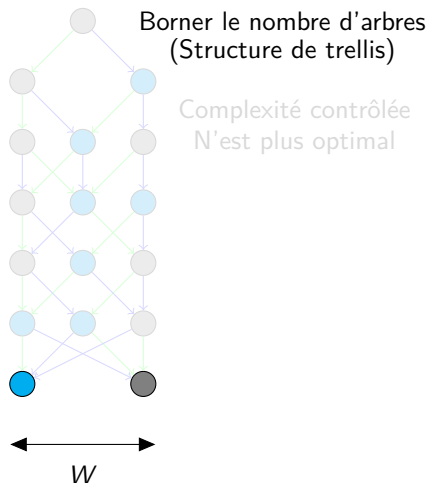
- Principe
- Formulation
- Résolution
- Résultats

## 4 Conclusion

## Mémoire bornée

MBDP (*Memory Bounded Dynamic Programming*) [seuken2007a]

$$\mathcal{A}_i = \{a_i, a'_i\}, \Omega_i = \{o_i, o'_i\}$$



$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

|                    |
|--------------------|
| $2 \times 18^{38}$ |
| $9 \times 18^9$    |
| $2 \times 10^9$    |
| 32768              |
| 128                |
| 8                  |
| 2                  |

Mise à jour  
exhaustive

$$|\mathcal{A}_i|^{|\Omega_i|^t}$$

|  |
|--|
| $2 \times 10^{19}$<br><del><math>4 \times 10^{19}</math></del> |
| $4 \times 10^9$<br><del><math>8 \times 10^9</math></del>       |
| 65536<br>131072  |
| 256<br>1024  |
| 16<br>32   |
| 4<br><del>8</del>  |
| 2  |

Élagage  
(exemple)

$$|\mathcal{A}_i| W^{|\Omega_i|}$$

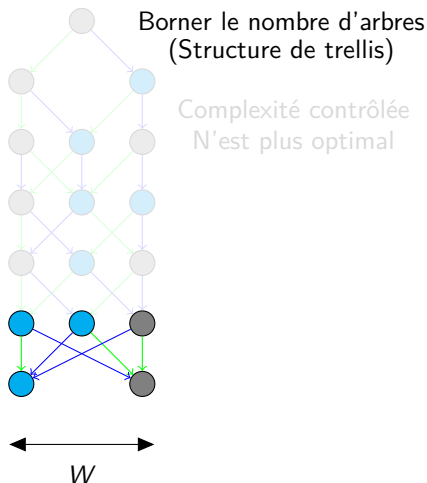
|                    |
|--------------------|
| 3<br><del>18</del> |
| 3<br><del>18</del> |
| 3<br><del>18</del> |
| 3<br><del>18</del> |
| 3<br><del>18</del> |
| 3<br><del>8</del>  |
| 2                  |

MBDP  
( $W = 3$ )

# Mémoire bornée

MBDP (*Memory Bounded Dynamic Programming*) [seuken2007a]

$$\mathcal{A}_i = \{a_i, a'_i\}, \Omega_i = \{o_i, o'_i\}$$



$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

|                    |
|--------------------|
| $2 \times 18^{38}$ |
| $9 \times 18^9$    |
| $2 \times 10^9$    |
| 32768              |
| 128                |
| 8                  |
| 2                  |

Mise à jour  
exhaustive

$$|\mathcal{A}_i|^{|\Omega_i|^t}$$

|  |
|--|
| $2 \times 10^{19}$<br><del><math>4 \times 10^{19}</math></del> |
| $4 \times 10^9$<br><del><math>8 \times 10^9</math></del>       |
| 65536<br>131072  |
| 256<br>1024  |
| 16<br>32   |
| 4<br><del>8</del>  |
| 2  |

Élagage  
(exemple)

$$|\mathcal{A}_i| W^{|\Omega_i|}$$

|                    |
|--------------------|
| 3<br><del>18</del> |
| 3<br><del>18</del> |
| 3<br><del>18</del> |
| 3<br><del>18</del> |
| 3<br><del>18</del> |
| 3<br><del>8</del>  |
| 2                  |

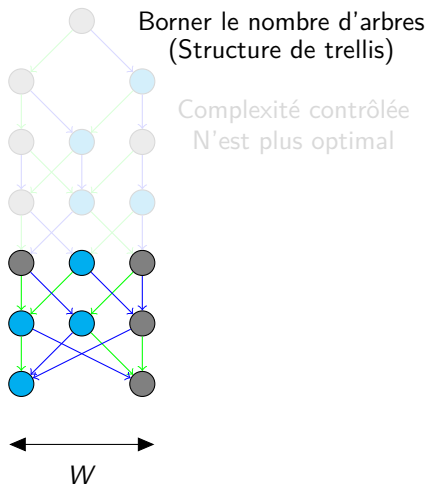
MBDP  
( $W = 3$ )



## Mémoire bornée

MBDP (*Memory Bounded Dynamic Programming*) [seuken2007a]

$$\mathcal{A}_i = \{a_i, a'_i\}, \Omega_i = \{o_i, o'_i\}$$



$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

|                    |
|--------------------|
| $2 \times 18^{38}$ |
| $9 \times 18^9$    |
| $2 \times 10^9$    |
| 32768              |
| 128                |
| 8                  |
| 2                  |

Mise à jour  
exhaustive

$$|\mathcal{A}_i|^{|\Omega_i|^t}$$

|  |
|--|
| $2 \times 10^{19}$<br><del><math>4 \times 10^{19}</math></del> |
| $4 \times 10^9$<br><del><math>8 \times 10^9</math></del>       |
| 65536<br>131072  |
| 256<br>1024  |
| 16<br>32   |
| 4<br>8   |
| 2  |

Élagage  
(exemple)

$$|\mathcal{A}_i| W^{|\Omega_i|}$$

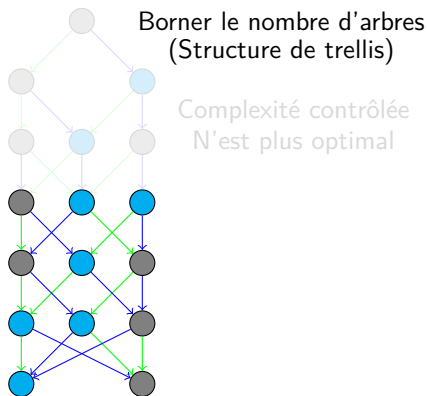
|                    |
|--------------------|
| 3<br><del>18</del> |
| 3<br><del>18</del> |
| 3<br><del>18</del> |
| 3<br><del>18</del> |
| 3<br><del>18</del> |
| 3<br><del>8</del>  |
| 2                  |

MBDP  
( $W = 3$ )

## Mémoire bornée

MBDP (*Memory Bounded Dynamic Programming*) [seuken2007a]

$$\mathcal{A}_i = \{a_i, a'_i\}, \Omega_i = \{o_i, o'_i\}$$

Borner le nombre d'arbres  
(Structure de trellis)Complexité contrôlée  
N'est plus optimal

$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

$$2 \times 18^{38}$$

$$9 \times 18^9$$

$$2 \times 10^9$$

$$32768$$

$$128$$

$$8$$

$$2$$

Mise à jour  
exhaustive

$$|\mathcal{A}_i|^{|\Omega_i|^t}$$

$$\begin{matrix} 2 \times 10^{19} \\ 4 \times 10^{19} \end{matrix}$$

$$\begin{matrix} 4 \times 10^9 \\ 8 \times 10^9 \end{matrix}$$

$$\begin{matrix} 65536 \\ 131072 \end{matrix}$$

$$\begin{matrix} 256 \\ 1024 \end{matrix}$$

$$\begin{matrix} 16 \\ 32 \end{matrix}$$

$$\begin{matrix} 4 \\ 8 \end{matrix}$$

$$2$$

Élagage  
(exemple)

$$|\mathcal{A}_i| W^{|\Omega_i|}$$

$$\begin{matrix} 3 \\ 18 \end{matrix}$$

$$\begin{matrix} 3 \\ 18 \end{matrix}$$

$$\begin{matrix} 3 \\ 18 \end{matrix}$$

$$\begin{matrix} 3 \\ 18 \end{matrix}$$

$$\begin{matrix} 3 \\ 18 \end{matrix}$$

$$\begin{matrix} 3 \\ 8 \end{matrix}$$

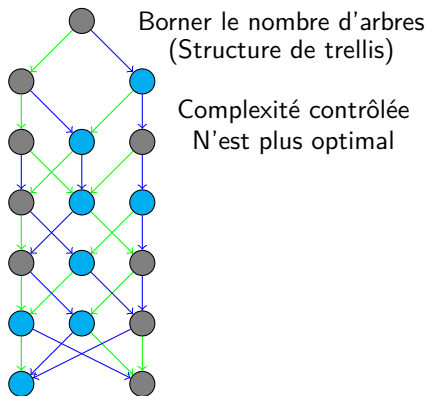
$$2$$

MBDP  
( $W = 3$ )

## Mémoire bornée

MBDP (*Memory Bounded Dynamic Programming*) [seuken2007a]

$$\mathcal{A}_i = \{a_i, a'_i\}, \Omega_i = \{o_i, o'_i\}$$



$$|\mathcal{A}_i|^{|\Omega_i|^T}$$

|                    |
|--------------------|
| $2 \times 18^{38}$ |
| $9 \times 18^9$    |
| $2 \times 10^9$    |
| 32768              |
| 128                |
| 8                  |
| 2                  |

Mise à jour  
exhaustive

$$|\mathcal{A}_i|^{|\Omega_i|^t}$$

|  |
|--|
| $2 \times 10^{19}$<br><del><math>4 \times 10^{19}</math></del> |
| $4 \times 10^9$<br><del><math>8 \times 10^9</math></del>       |
| 65536<br>131072  |
| 256<br>1024  |
| 16<br>32   |
| 4<br>8   |
| 2  |

Élagage  
(exemple)

$$|\mathcal{A}_i| W^{|\Omega_i|}$$

|         |
|---------|
| 3<br>18 |
| 3<br>18 |
| 3<br>18 |
| 3<br>18 |
| 3<br>18 |
| 3<br>8  |
| 2       |

MBDP  
(W = 3)

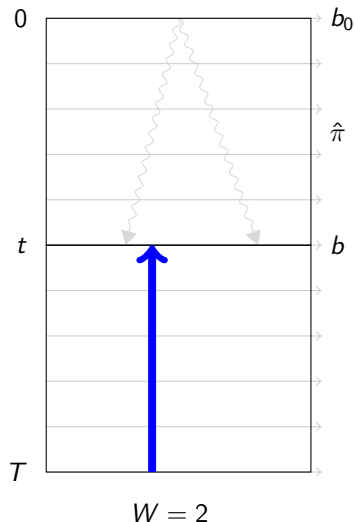
# Heuristique

Utilise la croyance centralisée :

$$B_t(s) = \Pr(S_t = s | B_0 = b_0, A_1, O_1, \dots, A_t, O_t)$$

Politique(s) heuristique(s)  $\hat{\pi}$

Sélectionne  $W$  points



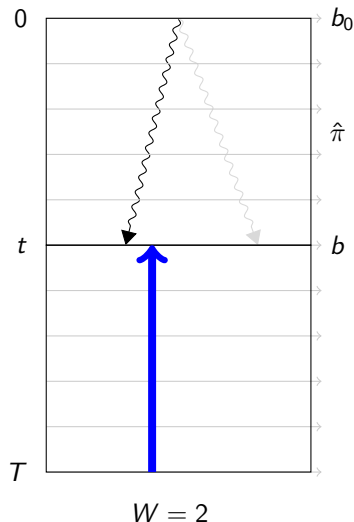
# Heuristique

Utilise la croyance centralisée :

$$B_t(s) = \Pr(S_t = s | B_0 = b_0, A_1, O_1, \dots, A_t, O_t)$$

Politique(s) heuristique(s)  $\hat{\pi}$

Sélectionne  $W$  points



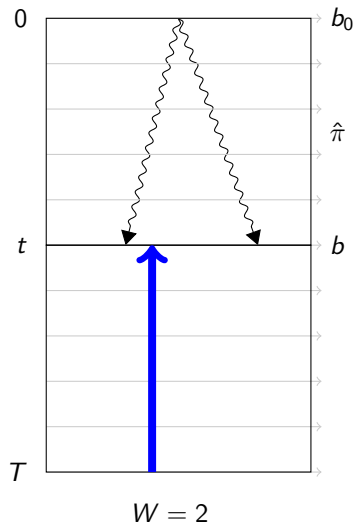
# Heuristique

Utilise la croyance centralisée :

$$B_t(s) = \Pr(S_t = s | B_0 = b_0, A_1, O_1, \dots, A_t, O_t)$$

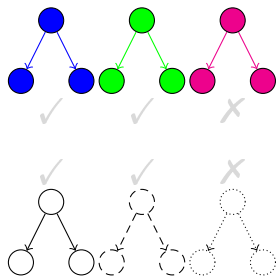
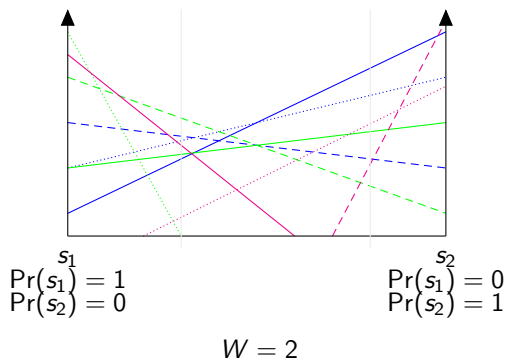
Politique(s) heuristique(s)  $\hat{\pi}$

Sélectionne  $W$  points



# Sélection des arbres de politique

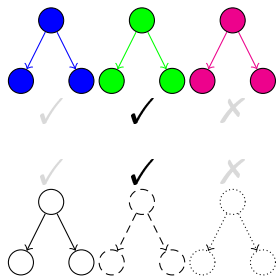
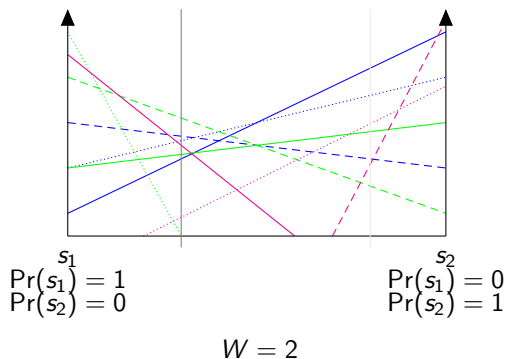
Optimisation indépendamment en  $W$  points (problème de *lookahead*)



Décomposé en  $W$  sous-problèmes indépendants

# Sélection des arbres de politique

Optimisation indépendamment en  $W$  points (problème de *lookahead*)

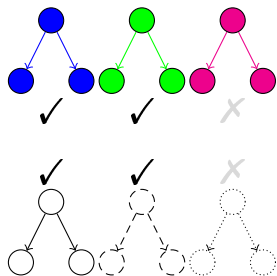
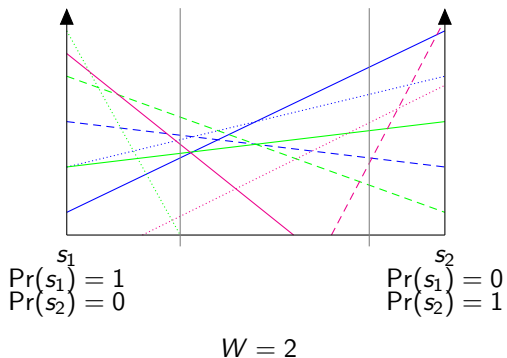


Décomposé en  $W$  sous-problèmes indépendants



# Sélection des arbres de politique

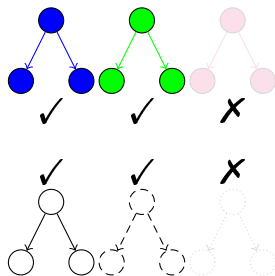
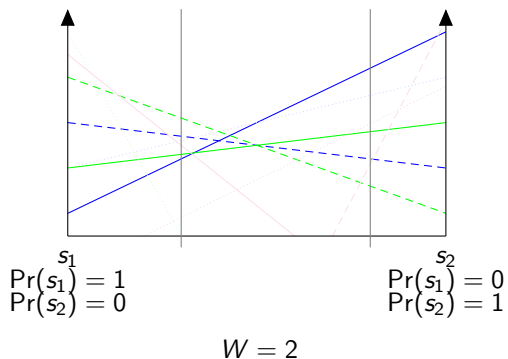
Optimisation indépendamment en  $W$  points (problème de *lookahead*)



Décomposé en  $W$  sous-problèmes indépendants

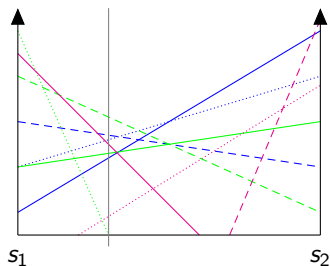
# Sélection des arbres de politique

Optimisation indépendamment en  $W$  points (problème de *lookahead*)



Décomposé en  $W$  sous-problèmes indépendants

# Opération de *Lookahead*



## MBDP [seuken2007a]

*Memory Bounded Dynamic Programming*  
Recherche exhaustive

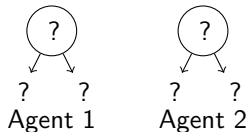
$$\prod_{i \in \mathcal{I}} |\mathcal{A}_i| W^{|\Omega_i|}$$

Problème d'optimisation combinatoire

## PBIP [dibangoye2008]

*Point Based Incremental Pruning*  
Recherche *branch-and-bound* :

- plus rapide
- même complexité de manière générale



# Contributions

Résolution approchée des problèmes de *lookahead*

« plus rapide (au risque d'être moins bon) »

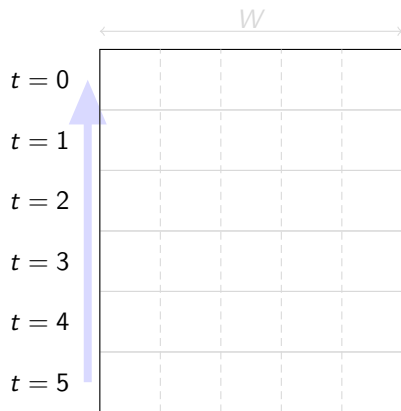
Connaissance heuristique sur le problème pour guider la planification

« meilleure qualité (au risque d'être plus lent) »

- 1 État de l'art
  - Cadre formel
  - Programmation dynamique
  - Mémoire bornée
  
- 2 *Lookahead* approché
  - Principe
  - Résolution
  - Résultats
  
- 3 PSMBDP
  - Principe
  - Formulation
  - Résolution
  - Résultats
  
- 4 Conclusion

- 1 État de l'art
  - Cadre formel
  - Programmation dynamique
  - Mémoire bornée
  
- 2 *Lookahead* approché
  - Principe
  - Résolution
  - Résultats
  
- 3 PSMBDP
  - Principe
  - Formulation
  - Résolution
  - Résultats
  
- 4 Conclusion

# Résumé de l'état de l'art



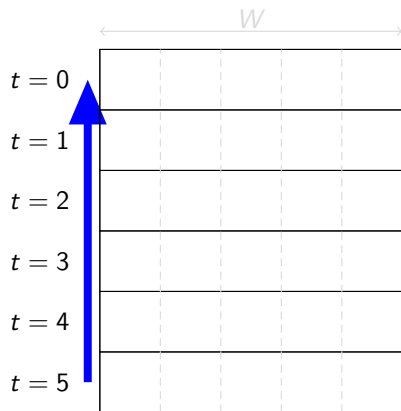
- 1 décomposition DP rétrograde
- 2 politiques bornées ( $W$ )
- 3  $W$  *lookaheads* indépendants

Limites :

- espace de recherche,  
 $\prod_{i \in \mathcal{I}} |\mathcal{A}_i| W^{|\Omega_i|}$
- $W$  très petit

Résolution exacte d'un sous-problème obtenu par de nombreuses approximations

# Résumé de l'état de l'art



- 1 décomposition DP rétrograde
- 2 politiques bornées ( $W$ )
- 3  $W$  *lookaheads* indépendants

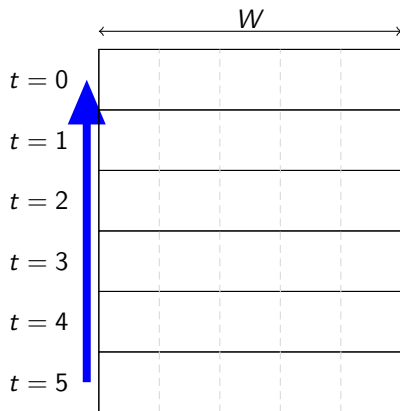
Limites :

- espace de recherche,  
 $\prod_{i \in \mathcal{I}} |\mathcal{A}_i| W^{|\Omega_i|}$
- $W$  très petit

Résolution exacte d'un sous-problème obtenu par de nombreuses approximations



# Résumé de l'état de l'art



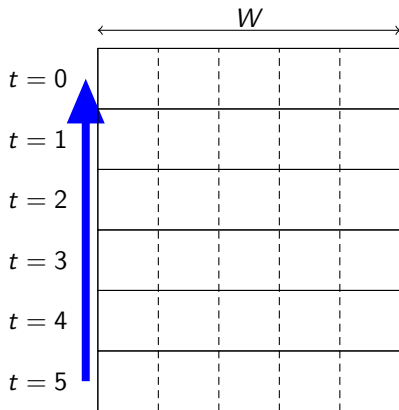
- 1 décomposition DP rétrograde
- 2 politiques bornées ( $W$ )
- 3  $W$  *lookaheads* indépendants

Limites :

- espace de recherche,  
 $\prod_{i \in \mathcal{I}} |\mathcal{A}_i| W^{|\Omega_i|}$
- $W$  très petit

Résolution exacte d'un sous-problème obtenu par de nombreuses approximations

# Résumé de l'état de l'art



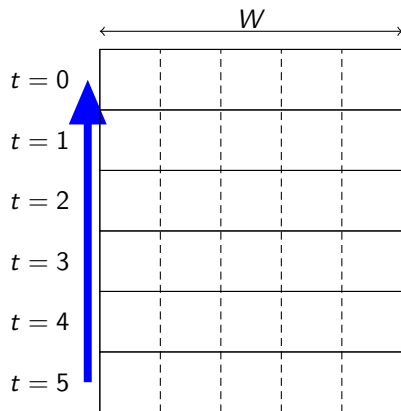
- 1 décomposition DP rétrograde
- 2 politiques bornées ( $W$ )
- 3  $W$  *lookaheads* indépendants

Limites :

- espace de recherche,  
 $\prod_{i \in \mathcal{I}} |\mathcal{A}_i| W^{|\Omega_i|}$
- $W$  très petit

Résolution exacte d'un sous-problème obtenu par de nombreuses approximations

# Résumé de l'état de l'art



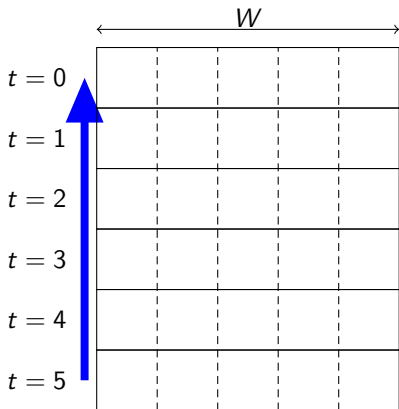
- 1 décomposition DP rétrograde
- 2 politiques bornées ( $W$ )
- 3  $W$  *lookaheads* indépendants

Limites :

- espace de recherche,  
 $\prod_{i \in \mathcal{I}} |\mathcal{A}_i| W^{|\Omega_i|}$
- $W$  très petit

Résolution exacte d'un sous-problème obtenu par de nombreuses approximations

# Lookahead approché



Résolution approchée,  
méta-heuristiques :

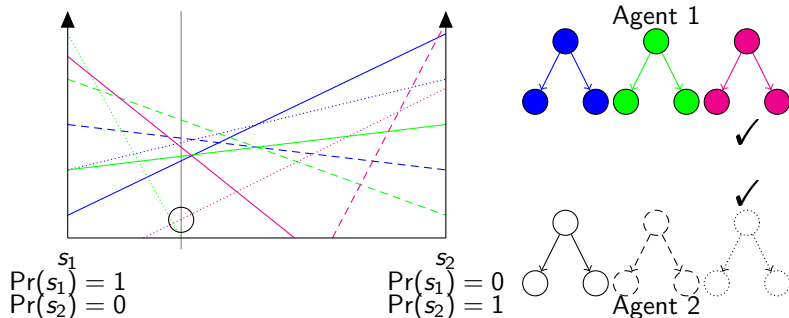
- ↘ complexité
- ↘ temps de calcul (à  $W$  constant)
- qualité attendue similaire (à  $W$  constant)
- ↗ arbres de politiques, qualité

Compromis  $W$ /résolution exacte

- 1 État de l'art
  - Cadre formel
  - Programmation dynamique
  - Mémoire bornée
  
- 2 *Lookahead* approché
  - Principe
  - **Résolution**
  - Résultats
  
- 3 PSMBDP
  - Principe
  - Formulation
  - Résolution
  - Résultats
  
- 4 Conclusion

# Optimisation locale

Recherche d'un maximum local : équilibre de Nash

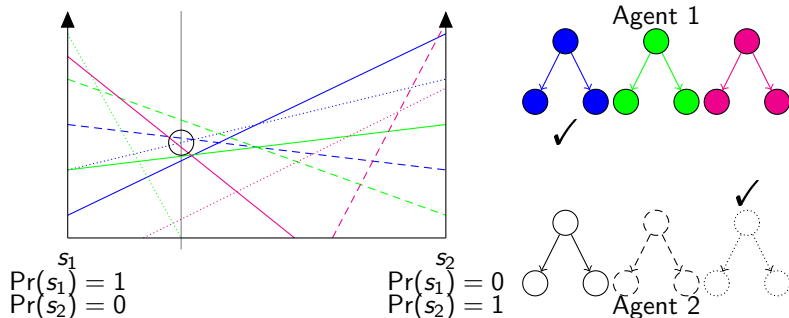


MBDP/NE : *Memory Bounded Dynamic Programming with Nash Equilibrium*

Pas une idée isolée : [kumar2010b], DecRSPI [wu2010], PBPG [wu2010b]

# Optimisation locale

Recherche d'un maximum local : équilibre de Nash

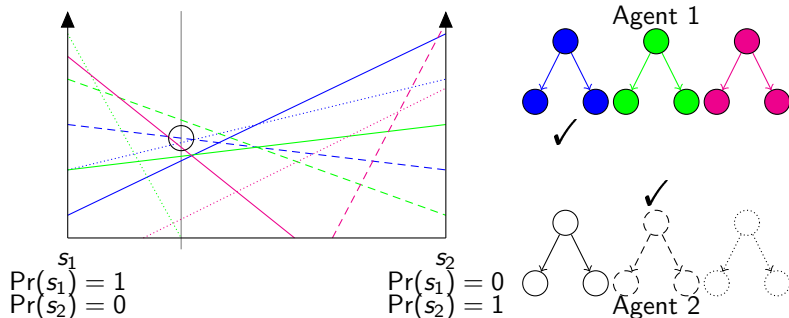


MBDP/NE : *Memory Bounded Dynamic Programming with Nash Equilibrium*

Pas une idée isolée : [kumar2010b], DecRSPI [wu2010], PBPG [wu2010b]

# Optimisation locale

Recherche d'un maximum local : équilibre de Nash



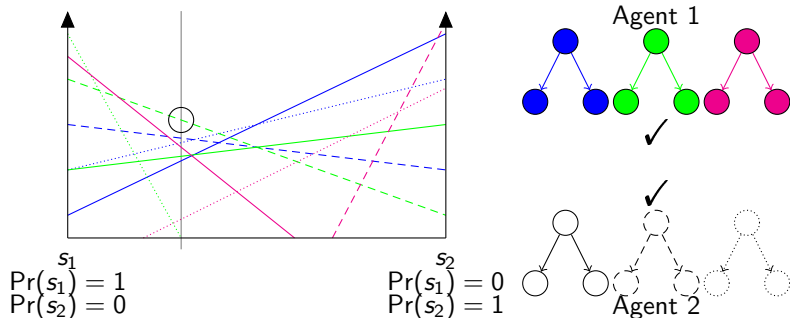
MBDP/NE : *Memory Bounded Dynamic Programming with Nash Equilibrium*

Pas une idée isolée : [kumar2010b], DecRSPI [wu2010], PBPG [wu2010b]



# Optimisation locale

Recherche d'un maximum local : équilibre de Nash



MBDP/NE : *Memory Bounded Dynamic Programming with Nash Equilibrium*

Pas une idée isolée : [kumar2010b], DecRSPI [wu2010], PBPG [wu2010b]

- 1 État de l'art
  - Cadre formel
  - Programmation dynamique
  - Mémoire bornée
  
- 2 *Lookahead* approché
  - Principe
  - Résolution
  - Résultats
  
- 3 PSMBDP
  - Principe
  - Formulation
  - Résolution
  - Résultats
  
- 4 Conclusion

# Résultats

| Algorithme | $W$ | $M$ | AEV   | $\sigma$ | Temps (s) |
|------------|-----|-----|-------|----------|-----------|
| PBIP/BeFS  | 7   | -   | 423.4 | 14.0     | 101       |
| MBDP/NE    | 7   | 30  | 423.0 | 9.6      | 83        |
| PBIP/BeFS  | 20  | -   | ?     | ?        | >24h      |
| MBDP/NE    | 20  | 30  | 445.2 | 7.2      | 822       |

TABLE : *Cooperative Box Pushing*,  $T = 20$

$$|\mathcal{I}| = 2, |\mathcal{A}_i| = 4, |\Omega_i| = 5$$

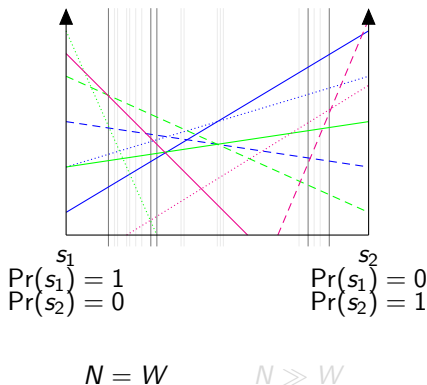
- 1 État de l'art
  - Cadre formel
  - Programmation dynamique
  - Mémoire bornée
  
- 2 *Lookahead* approché
  - Principe
  - Résolution
  - Résultats
  
- 3 PSMBDP
  - Principe
  - Formulation
  - Résolution
  - Résultats
  
- 4 Conclusion

- 1 État de l'art
  - Cadre formel
  - Programmation dynamique
  - Mémoire bornée
  
- 2 *Lookahead* approché
  - Principe
  - Résolution
  - Résultats
  
- 3 PSMBDP
  - **Principe**
  - Formulation
  - Résolution
  - Résultats
  
- 4 Conclusion

## Idée

|         | $W$            |                |                |                |                |
|---------|----------------|----------------|----------------|----------------|----------------|
| $t = 0$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ |
| $t = 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ |
| $t = 2$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ |
| $t = 3$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ |
| $t = 4$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ |
| $t = 5$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ |

Choix indépendant des arbres  
 Choix dépendant des arbres  
 $\Rightarrow$  Meilleure utilisation des arbres

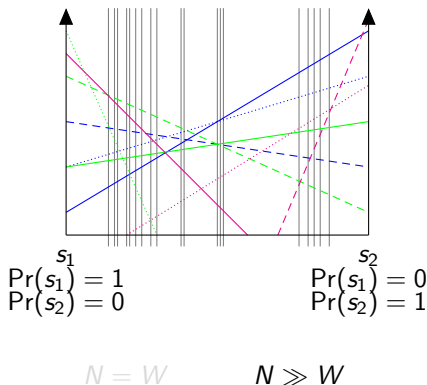


Faible nombre de points  
 Grand nombre de points  
 $\Rightarrow$  Plus d'information heuristique

## Idée

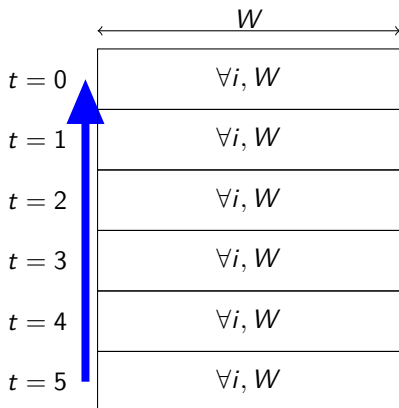
|         | $W$            |                |                |                |                |
|---------|----------------|----------------|----------------|----------------|----------------|
| $t = 0$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ |
| $t = 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ |
| $t = 2$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ |
| $t = 3$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ |
| $t = 4$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ |
| $t = 5$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ | $\forall i, 1$ |

Choix indépendant des arbres  
 Choix dépendant des arbres  
 $\Rightarrow$  Meilleure utilisation des arbres

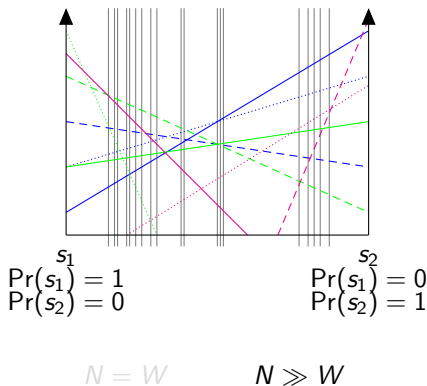


Faible nombre de points  
 Grand nombre de points  
 $\Rightarrow$  Plus d'information heuristique

## Idée



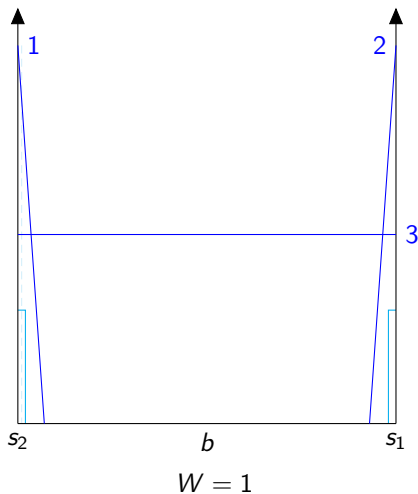
Choix indépendant des arbres  
 Choix dépendant des arbres  
 $\Rightarrow$  Meilleure utilisation des arbres



Faible nombre de points  
 Grand nombre de points  
 $\Rightarrow$  Plus d'information heuristique

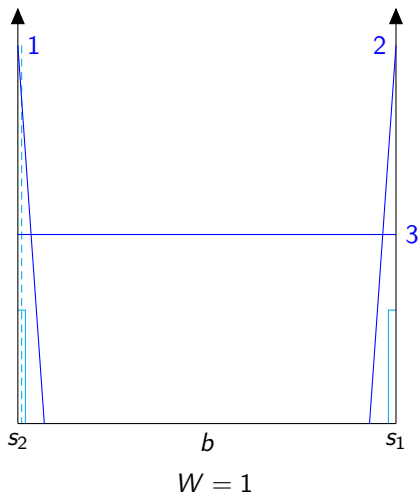


# Optimisation ponctuelle ou globale



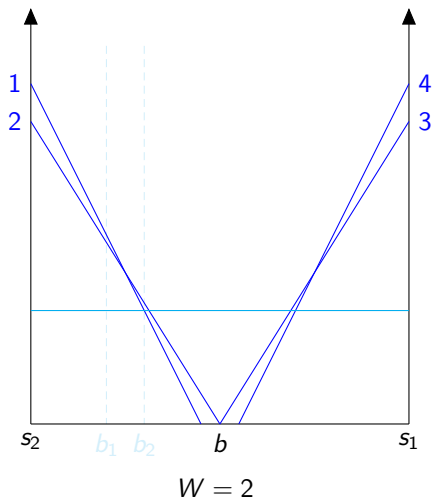
Optimiser ponctuellement en  $b_k$   
 problématique quand le nombre  
 d'arbres retenu est faible

# Optimisation ponctuelle ou globale



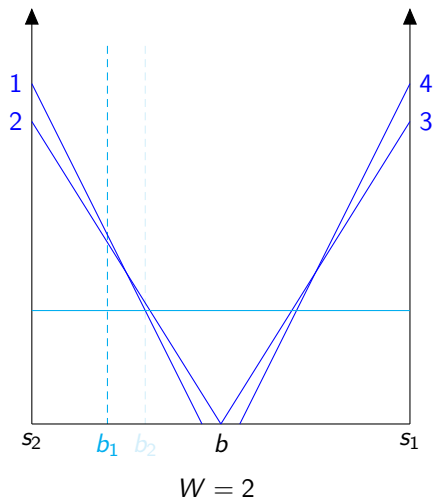
Optimiser ponctuellement en  $b_k$   
 problématique quand le nombre  
 d'arbres retenu est faible

# Choix indépendant ou dépendant des arbres



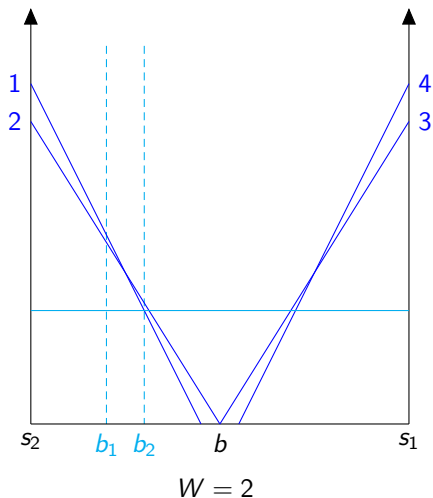
Le choix indépendant ignore la complémentarité entre les arbres.

# Choix indépendant ou dépendant des arbres



Le choix indépendant ignore la complémentarité entre les arbres.

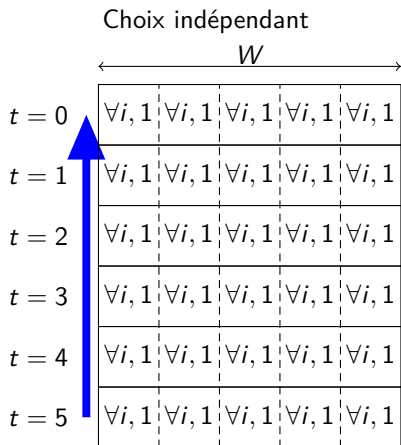
# Choix indépendant ou dépendant des arbres



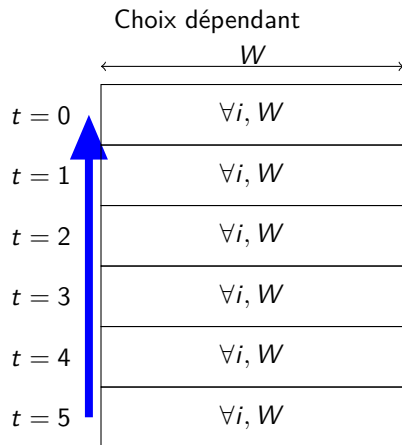
Le choix indépendant ignore la complémentarité entre les arbres.

- 1 État de l'art
  - Cadre formel
  - Programmation dynamique
  - Mémoire bornée
  
- 2 *Lookahead* approché
  - Principe
  - Résolution
  - Résultats
  
- 3 PSMBDP
  - Principe
  - **Formulation**
  - Résolution
  - Résultats
  
- 4 Conclusion

# Choix indépendant ou dépendant des arbres



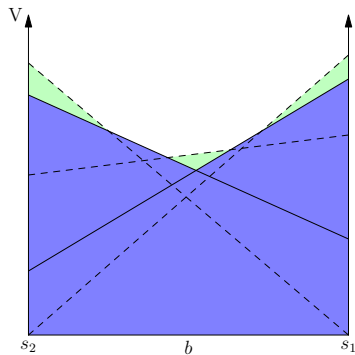
Choisir 1 arbre par agent ( $W$  fois)



Choisir (au plus)  $W$  arbres par agent

Quel critère ?

# Critère d'optimisation global

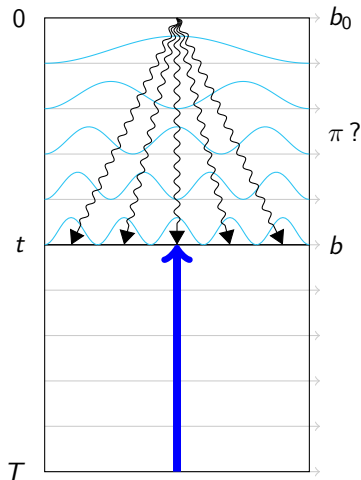


Critère moyen

$$\max_V \int V(b) db$$



# Distribution heuristique



Croyance centralisée  $B_t$  :

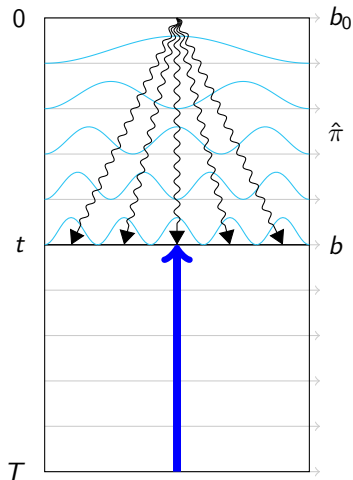
$$B_t(s) = Pr(S_t = s | B_0 = b_0, A_1, O_1, \dots, A_t, O_t)$$

$\pi$  ?

Distribution de probabilité *a priori* heuristique :

$$\mu(b) = p(B_t = b | \pi, B_0 = b_0)$$

# Distribution heuristique



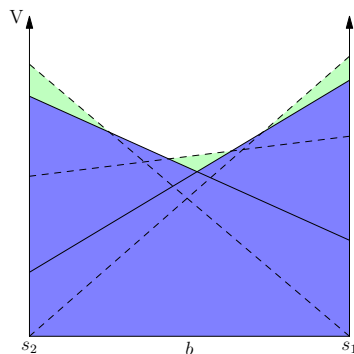
Croyance centralisée  $B_t$  :

$$B_t(s) = Pr(S_t = s | B_0 = b_0, A_1, O_1, \dots, A_t, O_t)$$

Distribution de probabilité *a priori* heuristique :

$$\mu(b) = p(B_t = b | \hat{\pi}, B_0 = b_0)$$

# Critère d'optimisation global



Critère moyen **espéré** échantillonné

$$\max_V \int \mu(b) V(b) db = \max_V E_{b \sim \mu} [V(b)]$$

$$\approx \frac{1}{N} \max_V \sum_{k=1}^N V(b_k)$$

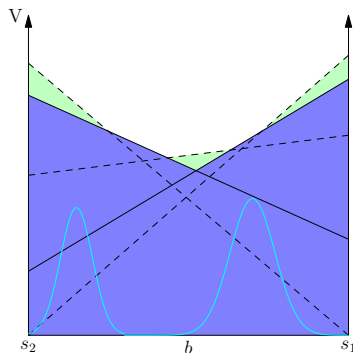
$\mu$  : distribution de probabilité sur l'espace des croyances centralisées (accessibilité)

$b_k$  : échantillons de  $\mu$  (Monte-Carlo)

On peut prendre  $N \gg W$

PSMBDP (*Policy Search Memory Bounded Dynamic Programming*)  
[corona2010a, corona2010b]

# Critère d'optimisation global



Critère ~~moyen~~ **espéré** échantillonné

$$\max_V \int \mu(b) V(b) db = \max_V E_{b \sim \mu} [V(b)]$$

$$\approx \frac{1}{N} \max_V \sum_{k=1}^N V(b_k)$$

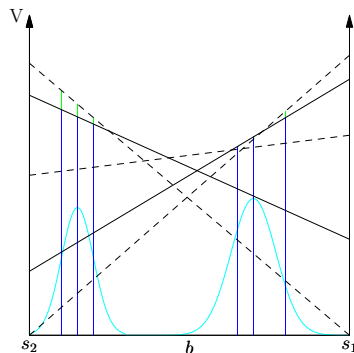
$\mu$  : distribution de probabilité sur l'espace des croyances centralisées (accessibilité)

$b_k$  : échantillons de  $\mu$  (Monte-Carlo)

On peut prendre  $N \gg W$

PSMBDP (*Policy Search Memory Bounded Dynamic Programming*)  
[corona2010a, corona2010b]

# Critère d'optimisation global



Critère ~~moyen~~ **espéré** échantillonné

$$\max_V \int \mu(b) V(b) db = \max_V E_{b \sim \mu} [V(b)]$$

$$\approx \frac{1}{N} \max_V \sum_{k=1}^N V(b_k)$$

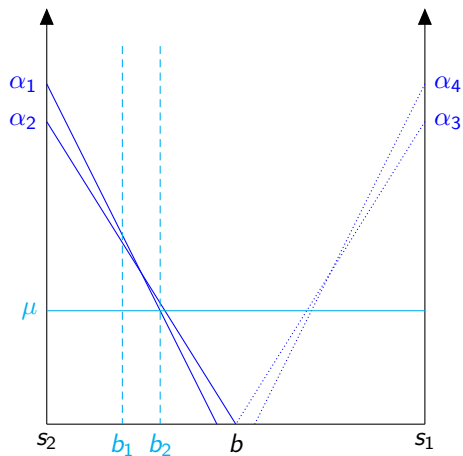
$\mu$  : distribution de probabilité sur l'espace des croyances centralisées (accessibilité)

$b_k$  : échantillons de  $\mu$  (Monte-Carlo)

On peut prendre  $N \gg W$

PSMBDP (*Policy Search Memory Bounded Dynamic Programming*)  
[corona2010a, corona2010b]

## Cas particulier



$$\max_V \sum_{k=1}^N V(b_k)$$

Si  $N = W$ ,

- meilleur arbre joint en chaque point
- équivalent aux méthodes à base de *lookahead*

Généralise MBDP, PBIP avec  $N \geq W$

- 1 État de l'art
  - Cadre formel
  - Programmation dynamique
  - Mémoire bornée
  
- 2 *Lookahead* approché
  - Principe
  - Résolution
  - Résultats
  
- 3 PSMBDP
  - Principe
  - Formulation
  - **Résolution**
  - Résultats
  
- 4 Conclusion

# Problème d'optimisation combinatoire

Pour chaque agent  $i \in \mathcal{I}$ , choisir au plus  $W$  arbres parmi  $|\mathcal{A}_i|W^{|\Omega_i|}$  :

$$\begin{aligned}
 \text{maximiser } & \sum_{k=1}^N \max_{q \in \mathcal{Z}^t} V_q(b_k) & \mathcal{Z}^t &= \prod_{i \in \mathcal{I}} \mathcal{Z}_i^t \\
 \text{sujet à } & \mathcal{Z}_i^t \subseteq \mathcal{A}_i \times (\mathcal{Z}_i^{t+1})^{\Omega_i} & \forall i \in \mathcal{I} \\
 & |\mathcal{Z}_i^t| \leq W & \forall i \in \mathcal{I}
 \end{aligned}$$

Espace de recherche énorme :

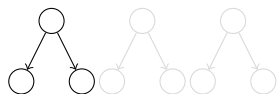
$$\prod_i \binom{|\mathcal{A}_i|W^{|\Omega_i|}}{W}$$

Recherche approchée, utilisation de méta-heuristiques

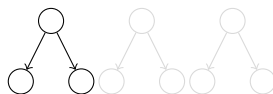


# Résolution approchée incrémentale gloutonne

- meilleure solution pour  $W = 1$  (*lookahead* pour  $b = \frac{1}{N} \sum b_k$ )
- ajouter progressivement des arbres ( $|\mathcal{I}|(W - 1)$  fois)



Agent 1



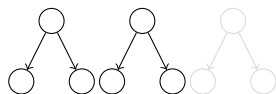
Agent 2

$$\prod_i \binom{|\mathcal{A}_i| W^{|\Omega_i|}}{W} \text{ devient } |\mathcal{A}_i| W^{|\Omega_i|} \text{ répété } |\mathcal{I}|(W - 1) \text{ fois}$$

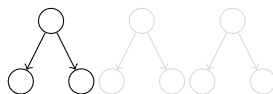
résolus par recherche *branch-and-bound* (ou approché, méta-heuristiques)

# Résolution approchée incrémentale gloutonne

- meilleure solution pour  $W = 1$  (*lookahead* pour  $b = \frac{1}{N} \sum b_k$ )
- ajouter progressivement des arbres ( $|\mathcal{I}|(W - 1)$  fois)



Agent 1



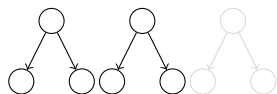
Agent 2

$$\prod_i \binom{|\mathcal{A}_i| W^{|\Omega_i|}}{W} \text{ devient } |\mathcal{A}_i| W^{|\Omega_i|} \text{ répété } |\mathcal{I}|(W - 1) \text{ fois}$$

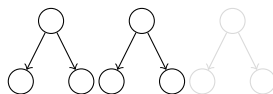
résolus par recherche *branch-and-bound* (ou approché, méta-heuristiques)

# Résolution approchée incrémentale gloutonne

- meilleure solution pour  $W = 1$  (*lookahead* pour  $b = \frac{1}{N} \sum b_k$ )
- ajouter progressivement des arbres ( $|\mathcal{I}|(W - 1)$  fois)



Agent 1



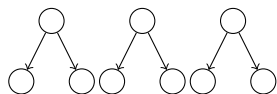
Agent 2

$$\prod_i \binom{|\mathcal{A}_i| W^{|\Omega_i|}}{W} \text{ devient } |\mathcal{A}_i| W^{|\Omega_i|} \text{ répété } |\mathcal{I}|(W - 1) \text{ fois}$$

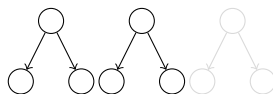
résolus par recherche *branch-and-bound* (ou approché, méta-heuristiques)

# Résolution approchée incrémentale gloutonne

- meilleure solution pour  $W = 1$  (*lookahead* pour  $b = \frac{1}{N} \sum b_k$ )
- ajouter progressivement des arbres ( $|\mathcal{I}|(W - 1)$  fois)



Agent 1



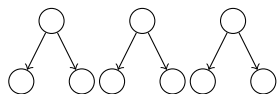
Agent 2

$$\prod_i \binom{|\mathcal{A}_i| W^{|\Omega_i|}}{W} \text{ devient } |\mathcal{A}_i| W^{|\Omega_i|} \text{ répété } |\mathcal{I}|(W - 1) \text{ fois}$$

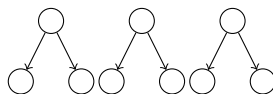
résolus par recherche *branch-and-bound* (ou approché, méta-heuristiques)

# Résolution approchée incrémentale gloutonne

- meilleure solution pour  $W = 1$  (*lookahead* pour  $b = \frac{1}{N} \sum b_k$ )
- ajouter progressivement des arbres ( $|\mathcal{I}|(W - 1)$  fois)



Agent 1



Agent 2

$$\prod_i \binom{|\mathcal{A}_i| W^{|\Omega_i|}}{W} \text{ devient } |\mathcal{A}_i| W^{|\Omega_i|} \text{ répété } |\mathcal{I}|(W - 1) \text{ fois}$$

résolus par recherche *branch-and-bound* (ou approché, méta-heuristiques)

- 1 État de l'art
  - Cadre formel
  - Programmation dynamique
  - Mémoire bornée
  
- 2 *Lookahead* approché
  - Principe
  - Résolution
  - Résultats
  
- 3 PSMBDP
  - Principe
  - Formulation
  - Résolution
  - Résultats
  
- 4 Conclusion

# Résultats

| Algorithm   | VEM     | $\sigma$ | Temps (s) |
|-------------|---------|----------|-----------|
| Optimal MDP | -35.81  | -        | -         |
| PBIP        | -160.25 | 20.84    | 171       |
| PBIP ‡      | -111.85 | 0.00     | 234       |
| PSMBDP      | -89.08  | 0.00     | 274       |
| PBIP        | -216.64 | 35.92    | 612       |
| PBIP ‡      | -111.85 | 0.00     | 156       |
| PSMBDP      | -78.43  | 0.00     | 735       |

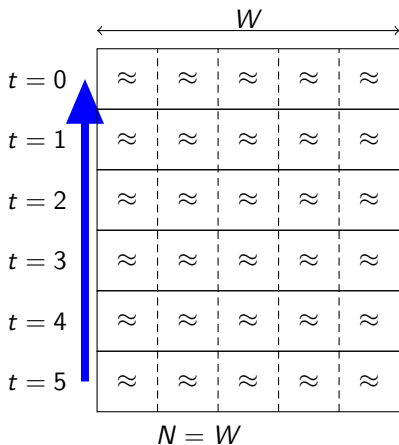
‡ : points tirés comme pour PSMBDP

TABLE : Problème des pompiers et problème des pompiers modifié,  $W = 7$ ,  $T = 20$ ,  $N = 100$ , 25 exécutions

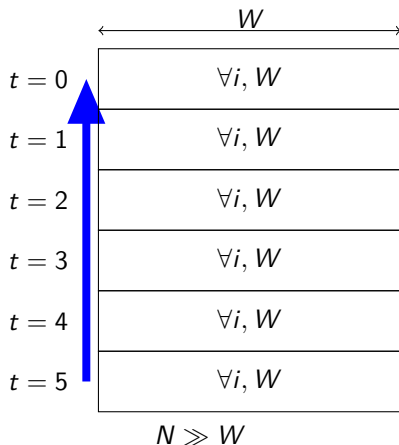
- 1 État de l'art
  - Cadre formel
  - Programmation dynamique
  - Mémoire bornée
  
- 2 *Lookahead* approché
  - Principe
  - Résolution
  - Résultats
  
- 3 PSMBDP
  - Principe
  - Formulation
  - Résolution
  - Résultats
  
- 4 Conclusion



## Résumé



- réduire la complexité
- augmenter  $W$  (et  $N$ )



- bonne utilisation des heuristiques
- complémentarité

# La fin



# Bibliographie I

-  Daniel S. Bernstein, Shlomo Zilberstein, and Neil Immerman.  
The Complexity of Decentralized Control of Markov Decision Processes.  
*In Mathematics of Operations Research*, page 2002, 2000.
-  Gabriel Corona and François Charpillet.  
Distribution sur les croyances pour la planification de Dec-POMDP.  
*Revue d'Intelligence Artificielle*, 24(4) :525–544, 2010.
-  Gabriel Corona and François Charpillet.  
Distribution over Beliefs for Dec-POMDP planning.  
*In Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, 2010.
-  Jilles S. Dibangoye, Abdel-Ilhah Mouaddib, and Brahim Chaib-draa.  
Recherche incrémentale à base de points pour la résolution des DEC-POMDPs.  
*In Actes des quinzièmes JFSMA*, Brest, France, October 2008.

# Bibliographie II



E. Hansen, D. Bernstein, and Shlomo Zilberstein.

Dynamic programming for partially observable stochastic games.

In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 709–715. AAAI Press, 2004.



Akshat Kumar and Shlomo Zilberstein.

Point-based backup for decentralized pomdps : Complexity and new algorithms.

In *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems*, pages 1315–1322, Toronto, Canada, 2010.



Sven Seuken and Shlomo Zilberstein.

Memory-Bounded Dynamic Programming for DEC-POMDPs.

In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligences (IJCAI-07)*. IJCAI, 2007.

# Bibliographie III



Daniel Szer.

*Contribution à la résolution des processus de décision markoviens décentralisés.*

PhD thesis, Université Henri-Poincaré, Nancy, France.



Feng Wu, Shlomo Zilberstein, and Xiaoping Chen.

Rollout Sampling Policy Iteration for Decentralized POMDPs.

In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, 2010.



Feng Wu, Shlomo Zilberstein, and Xiaoping Chen.

Point-Based Policy Generation for Decentralized POMDPs.

In *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems*, pages 1307–1314, Toronto, Canada, 2010.

# Droits d'auteurs

internet, ©The Opte Project, CC-BY-2.5

mars, domaine publique (NASA)

pompiers/maisons/radiations/mars/robots, domaine publique (OpenClipArt)

cycab, (honteusement récupéré sur le site du Lasmae)