Planification et coordination multiagents sous incertitude

Aurélie Beynier

CoCoMa, Master 2 ANDROIDE
2016-2017

18 octobre 2016

Organisation

- Aurélie Beynier : aurelie.beynier@lip6.fr
- 4 séances : 18/10, 25/10, 22/11, 29/11
- Planification et coordination multiagents
  - Incertitude, Observabilité, Distribution
  - Modèles Markoviens mono et multiagents
  - Complexité, algorithmes exacts et approchés
  - Résolution en ligne et apprentissage
  - Cadre non-coopératif et jeux bayésiens
  - Optimisation de contraintes distribuée
- Soutenances de projet : 1/12 (jeudi après-midi)
- Suivies de 5 séances par N. Maudet sur allocation de ressources, négociation, consensus, argumentation

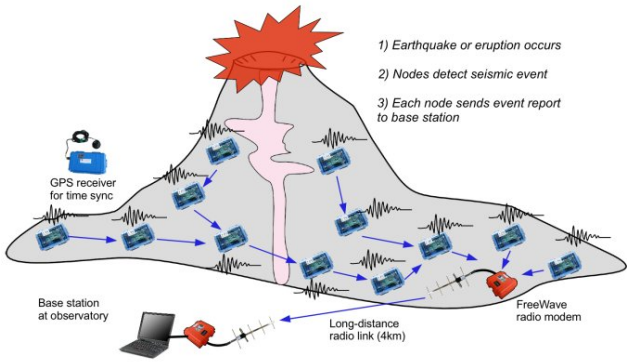
Plan

1. SMA, planification, incertitude
2. Planification mono-agent
   - Décision séquentielle sous incertitude
   - Processus Décisionnels de Markov
   - Processus Décisionnels de Markov Partiellement Observables
3. Planification multiagent sous incertitude
   - Problématiques multiagents
   - Notion d'observabilité
   - Exemples de problèmes
   - Formalismes
   - Exemples Applicatifs
   - Résolution de DEC-POMDPs

Contexte [Woo02, Wei99]

- Un groupe de n agents **délibératifs** interagissant dans un même environnement.
- Ces agents doivent anticiper leurs décisions afin de déterminer la suite d'actions la plus favorable permettant d'atteindre leurs buts.
- Nécessité de **planifier** et de **coordonner** les actions des agents

Contexte


1) Earthquake or eruption occurs
2) Nodes detect seismic event
3) Each node sends event report to base station
GPS receiver for time sync
Base station at observatory
Long-distance radio link (4km)
FreeWave radio modem

Pourquoi la coordination ?

Jennings [Jen96]

La coordination émane des caractéristiques des situations d'interaction :

- Éviter l'anarchie et le chaos
- Augmenter l'efficacité
- Répondre à des contraintes globales
- Distribuer l'information, l'expertise ou les ressources
- Prendre en compte les dépendances entre les actions des agents

Pourquoi la coordination ?

Decker et Lesser [DL95]

Coordination basée sur le point de vue d'un agent :

- L'agent décide de ses actions et ses choix affecte ses performances,
- l'ordre dans lequel les activités sont menées affecte les performances,
- l'instant auquel sont exécutées les actions affecte les performances.

Retour sur la planification classique

Hypothèses

- Observation complète de l'environnement :
  - Champ d'observation ? (portée des capteurs)
  - Observation parfaite ? (bruit des capteurs)
- Communication entre les agents :
  - Communication possible avec tous les agents ?
  - Communication sans bruit ?
  - Communication instantanée et sans bruit ?
- Actions déterministes
  - Modélisation parfaite de l'environnement ?
  - Environnement dynamique ?
  - Comportement parfait des effecteurs ?

Retour sur la planification classique

Dans les systèmes réels

- Calcul et exécution décentralisés des plans,
- Les actions ont des durées différentes,
- Le moment où les actions sont exécutées est important,
- Les plans peuvent être affinés par décomposition,
- L'environnement est partiellement observable,
- Le résultat de l'exécution d'une action est incertain.

Qu'est-ce que l'incertitude ?

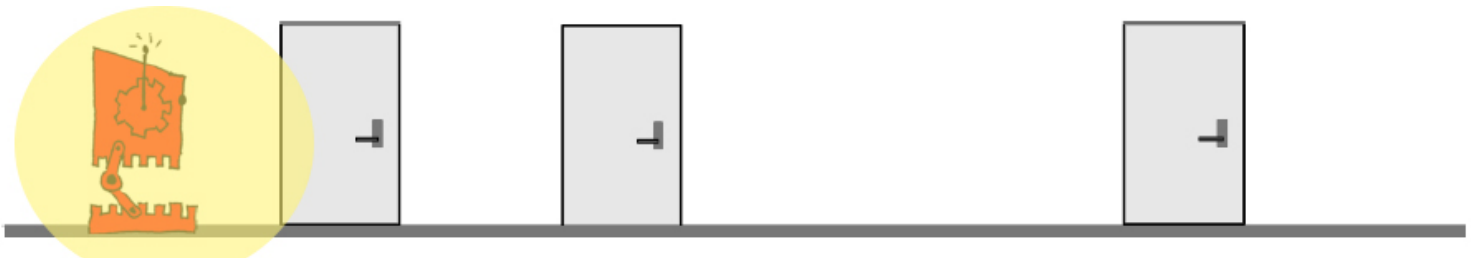
- Caractérise le fait qu'un agent ne puisse pas déterminer l'issue d'une action de façon déterministe : l'action ne conduit pas toujours au même résultat
- Concrètement : à partir d'un même état s, si un agent exécute plusieurs fois une même action a, il pourra arriver dans différents états s'.
- Par la suite, nous supposerons que nous disposons d'une représentation probabiliste de l'incertitude, c'est-à-dire qu'on représente les issues des actions avec une distribution de probabilités : $T : S \rightarrow \Pr(S)$, où $T(s)(s')$ est la probabilité de passer de l'état s à s' telle que $\sum_{s' \in S} T(s, s') = 1, \forall s \in S$

Origine de l'incertitude

Incertitude

Observation Partielle + Dynamicité de l'environnement → Incertitude

- Observation partielle de l'environnement et des autres agents : portée limitée des capteurs, non-observation des états internes des autres agents,...



Origine de l'incertitude - suite

- Observation bruitée
- Modélisation imparfaite ou incorrecte de la dynamique de l'environnement : impossibilité de modéliser tous les événements exogènes, toutes les issues possibles d'une action (toutes les caractéristiques de l'environnement et leur influence sur la dynamique du système,...)
- Effecteurs imparfaits

Gestion de l'incertitude [RN03]

- **Planification sans prise en compte des perceptions** : déterminer un plan qui puisse être exécuté quelles que soient les circonstances et qui ne nécessite donc pas de tenir compte des perceptions
- **Planification conditionnelle** : calculer un plan contingent ayant différentes branches correspondant aux différentes séquences d'exécution possibles. Les perceptions permettent, à l'exécution, de savoir dans quelle branche on se situe.
- **Replanification à l'exécution** : calculer un plan en utilisant des techniques classiques et détecter à l'exécution si on a dévié de la situation prévue. Si c'est le cas, on replanifie en tenant compte de la situation effective.
- **Planification continue** : l'agent planifie continuellement au cours de l'exécution en fonction des avancées et des résultats de ses actions.

Gestion de l'incertitude

- Hypothèse : on dispose d'une représentation probabiliste des issues des actions (ou on peut en apprendre une) $\rightarrow$ fonction de transition entre les états connue
- Objectifs : décider quelle est la meilleure action à faire à chaque pas de temps étant données les évolutions futures possibles $\rightarrow$ **Décision séquentielle**
- Idée : trouver un plan qui permet de maximiser les performances de l'agent (ou des agents) étant données les informations dont il dispose sur l'environnement $\rightarrow$ **Décision rationnelle**

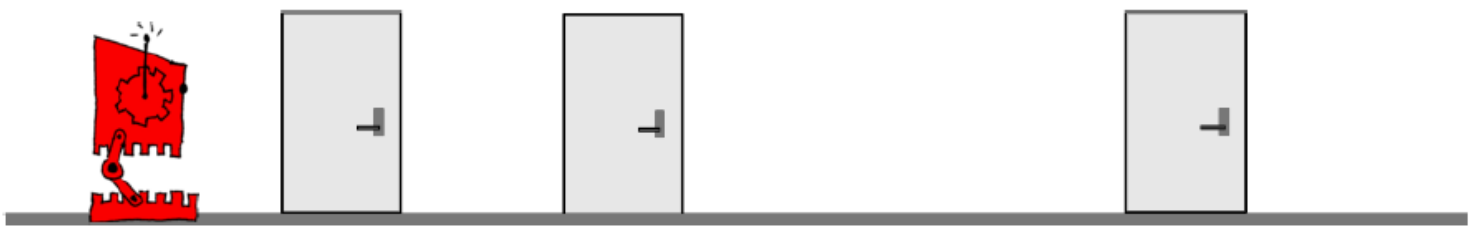
Décision séquentielle sous incertitude : cas mono-agent

- Commençons par considérer le cas (plus simple) d'**un seul agent**
- On considère des problèmes de **décision séquentielle** : à chaque pas de temps l'agent doit décider comment agir
- **Sous incertitude** : le résultat de l'action est incertain
- Différents niveaux d'observabilité sont envisageables :
  - Observabilité complète : l'agent connaît, à chaque pas de temps, l'état du système
  - Observabilité partielle : l'agent n'observe, à chaque pas de temps, qu'une partie de l'état du système. Il doit donc faire des hypothèses sur l'état sous-jacent à partir de ses observations

Décision séquentielle sous incertitude : cas mono-agent

Un exemple : Robot se déplaçant dans un couloir

- Distance parcourue par le robot incertaine : *lorsqu'il entreprend un déplacement long : la probabilité qu'il avance de 3 mètres est de* 20%, *la probabilité qu'il avance de 4 mètres est de* 60%, *la probabilité qu'il avance de 5 mètres est de* 20%.
- Le robot doit s'arrêter devant la troisième porte $\rightarrow$ objectif, récompense attribuée lorsque le robot atteint un tel état



extrait de Amato et al.

Rationalité et théorie de l'utilité

- Un agent possède des préférences sur les différentes issues des plans.
- Dans la théorie de l'utilité, les préférences sont représentées par un degré d'utilité.
- L'incertitude est représentée par des distributions de probabilités
- L'utilité espérée d'une action représente la moyenne pondérée des utilités sur les différentes issues de l'action :

$$UE(A|E) = \sum_i P(Result_i(A)|Do(A), E)U(Result_i(A))$$

En théorie de la décision, un agent est rationnel, si et seulement si, il choisit à chaque instant l'action qui maximise son utilité espérée.

Processus Décisionnels Markoviens

- Les Processus Décisionnels de Markov (MDPs) et les Processus Décisionnels de Markov Partiellement Observables (POMDPs) [Put05] permettent de formaliser des problèmes de décision séquentielle mono-agent sous incertitude.
- MDP pour le cas totalement observable
- POMDP pour le cas partiellement observable

Processus Décisionnels Markoviens

- Un MDP est défini par un tuple $\langle S, A, T, R \rangle$
  - S : l'ensemble des états s du système
  - A : l'ensemble des actions a de l'agent
  - T : la fonction de transition modélisant l'incertitude, $T(s'|s, a)$ est la probabilité de passer d'un état s à un état s' en effectuant l'action a
  - R : la fonction de récompense modélisant les objectifs de l'agent, $R(s, a)$ est la récompense obtenue par l'agent lorsque l'action a est exécutée partir de l'état s.

Processus Décisionnels Markoviens

Formalisons le problème de décision du robot dans le corridor sous forme d'un MDP

- Un MDP est défini par un tuple $\langle S, A, T, R \rangle$
  - S : ?
  - A : ?
  - T : ?
  - R : ?

Processus Décisionnels Markoviens

Formalisons le problème de décision du robot dans le corridor sous forme d'un MDP

- Un MDP est défini par un tuple $\langle S, A, T, R \rangle$
  - S : position du robot dans un espace à une dimension
  - A : longMove ou shortMove ou stop
  - T : $T(s, stop, s) = 1$, $T(1, move, 4) = 0.2$, $T(1, move, 5) = 0.6$, $T(1, move, 6) = 0.2$, $T(1, move, 1) = 0.2$, $T(1, move, 2) = 0.6$, $T(1, move, 3) = 0.2$,...
  - R : récompense positive si on est devant la porte, sinon récompense nulle

Processus Décisionnels Markoviens

- Une solution à un problème formalisé sous forme de MDP est une politique :

$$\pi : S \rightarrow A$$

- Par exemple : $1 \rightarrow \textit{longMove}$, $2 \rightarrow \textit{longMove}$, $\cdots$, $19 \rightarrow \textit{shortMove}$, $20 \rightarrow \textit{stay}$
- Une politique δ dans un état s peut être évaluée à un instant t en calculant la récompense espérée à partir de s :

$$V^{\delta_t}(s) = R(s, \delta_t(s)) + \gamma \sum_{s' \in S} T(s, \delta_t(s), s') \times V^{\delta_{t+1}}(s') \quad (1)$$

où$\gamma \in [0, 1[$ est un facteur d'actualisation

Processus Décisionnels Markoviens

- La politique optimale est telle que :

$$\pi^*(s) = argmax_a \left\{ R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') \times V^{\pi^*}(s') \right\} \quad (2)$$

- Des algorithmes efficaces de résolution existent pour calculer la politique optimale (itération de la valeur, itération des politiques, ...)

Processus Décisionnels Markoviens Partiellement Observables

- Un POMDP est défini par un tuple $\langle S, A, T, O, \Omega, R \rangle$
  - S, A, T, R identiques à un POMDP
  - O : l'ensemble des observations o de l'agent
  - Ω : la fonction d'observation, $\Omega(O|s', a)$ est la probabilité d'obtenir l'observation o lorsque l'agent exécute l'action a et et arrive dans l'état s'

Processus Décisionnels Markoviens Partiellement Observables

Formalisons le problème de décision du robot dans le corridor sous forme d'un POMDP

Le robot ne perçoit que la présence d'une porte ou d'un mur du corridor, pas sa position.

- S : ?
- A : ?
- T : ?
- R : ?
- O : ?
- Ω : ?

Processus Décisionnels Markoviens Partiellement Observables

Formalisons le problème de décision du robot dans le corridor sous forme d'un POMDP

- S : position du robot dans un espace à une dimension
- A : move ou stop
- T : $T(s, stop, s) = 1 \ \forall s,\ T(1, move, 4) = 0.2,$ $T(1, move, 5) = 0.6,\ T(1, move, 6) = 0.2,\ ...$
- R : récompense positive si on est devant la porte, sinon récompense nulle
- O : door ou corridor
- Ω : par exemple 10% de bruit sur les observations $P(door, 2, stay) = 0.1,\ P(corridor, 2, stay) = 0.9,\ ...$

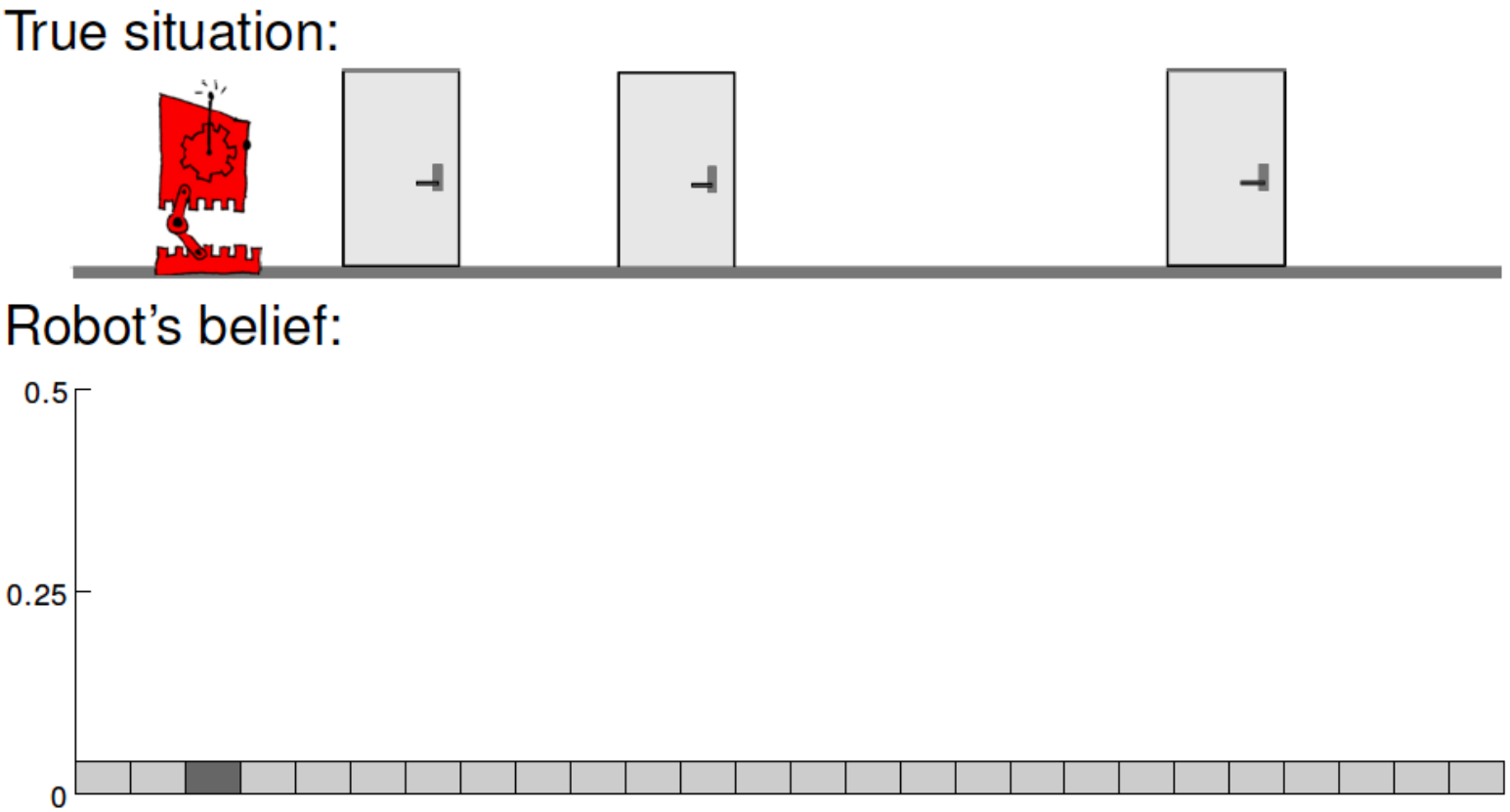
Processus Décisionnels Markoviens Partiellement Observables

- A partir de ses observations, l'agent essaye de déterminer dans quel état il se trouve : pour cela il maintient des *beliefs* sur son état : $b(s)$ est la probabilité que l'agent soit dans l'état s
- Les probabilité sur les beliefs sont mises à jour en appliquant la règle de Bayes :

$$b'(s') \propto p(o|s') \sum_s p(s'|s,a) b(s)$$

- Complexité algorithmique plus importante : on préfère les méthodes approchées (algo le plus efficace actuellement POMCP)

Processus Décisionnels Markoviens Partiellement Observables



- Observations: *door* or *corridor*, 10% noise.
- Action: moves 3 (20%), 4 (60%), or 5 (20%) states.

extrait de Amato et al.

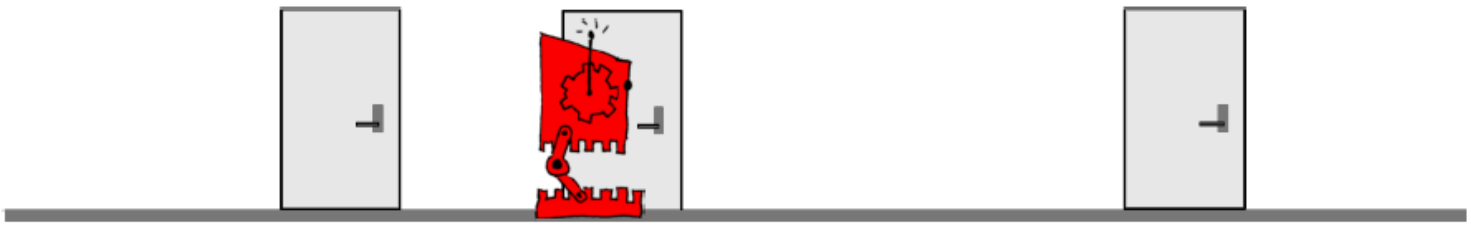
Processus Décisionnels Markoviens Partiellement Observables

- Observations: **door** or *corridor*, 10% noise.
- Action: moves 3 (20%), 4 (60%), or 5 (20%) states.

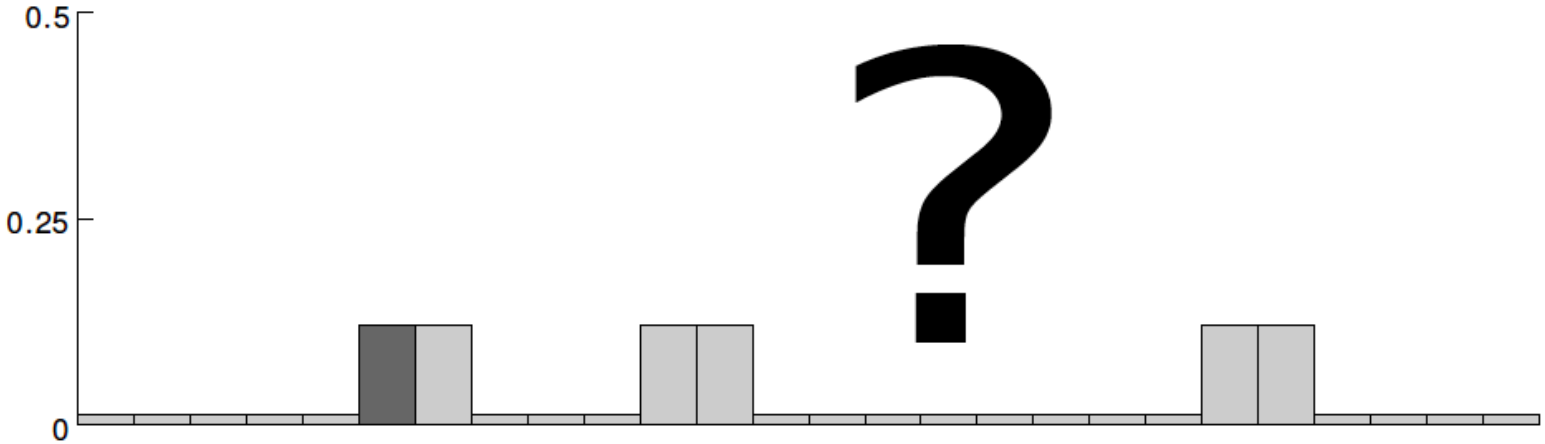
extrait de Amato et al.

Processus Décisionnels Markoviens Partiellement Observables

True situation:



Robot's belief:

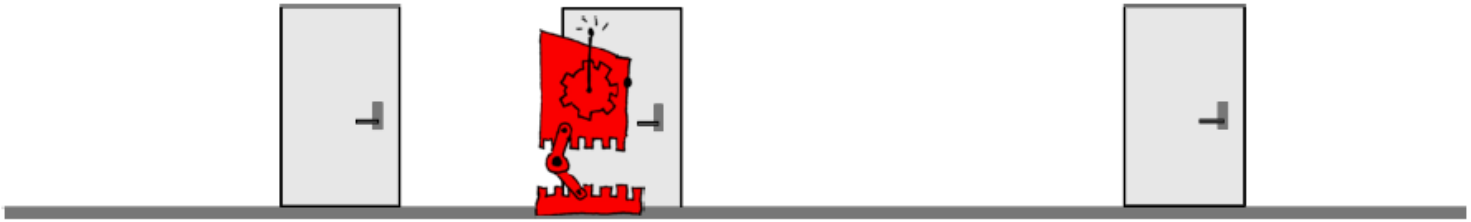


- Observations: **door** or *corridor*, 10% noise.
- Action: moves 3 (20%), 4 (60%), or 5 (20%) states.

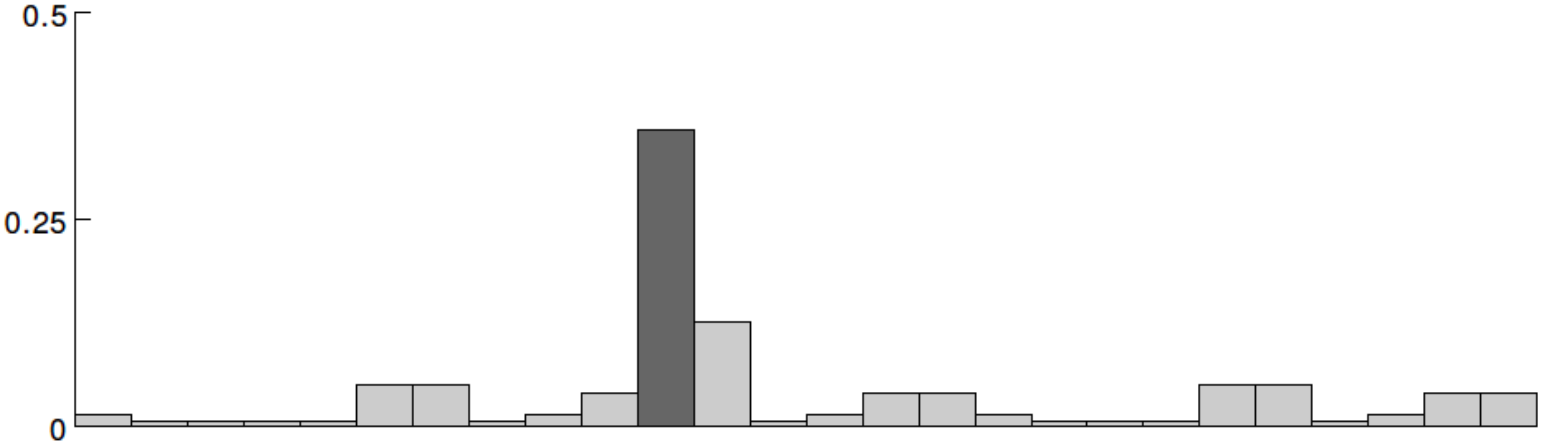
extrait de Amato et al.

Processus Décisionnels Markoviens Partiellement Observables

True situation:



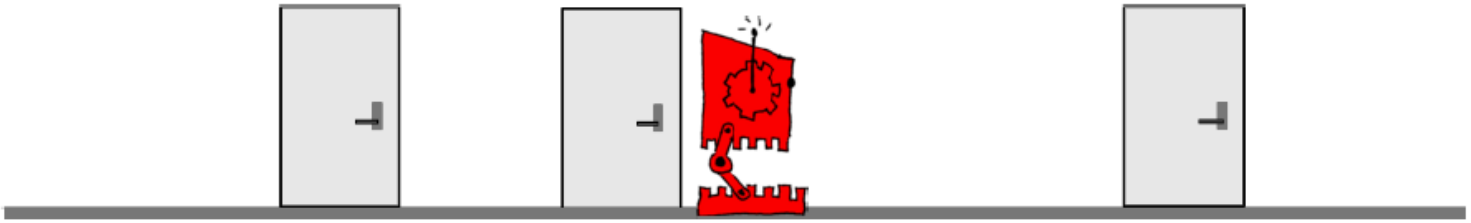
Robot's belief:



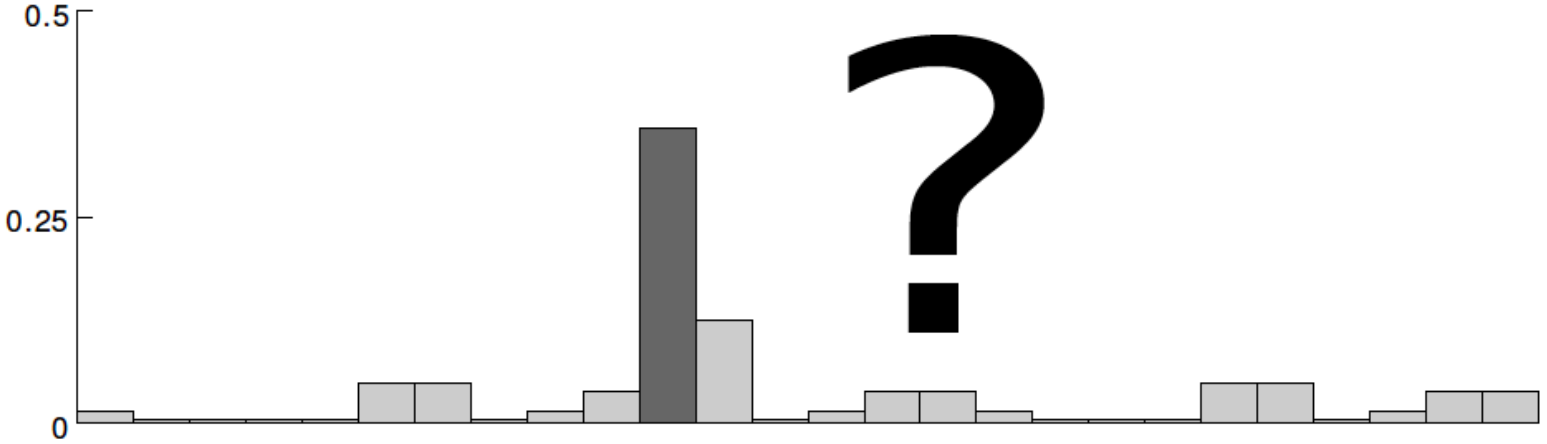
- Observations: **door** or *corridor*, 10% noise.
- Action: moves 3 (20%), 4 (60%), or 5 (20%) states.

Processus Décisionnels Markoviens Partiellement Observables

True situation:

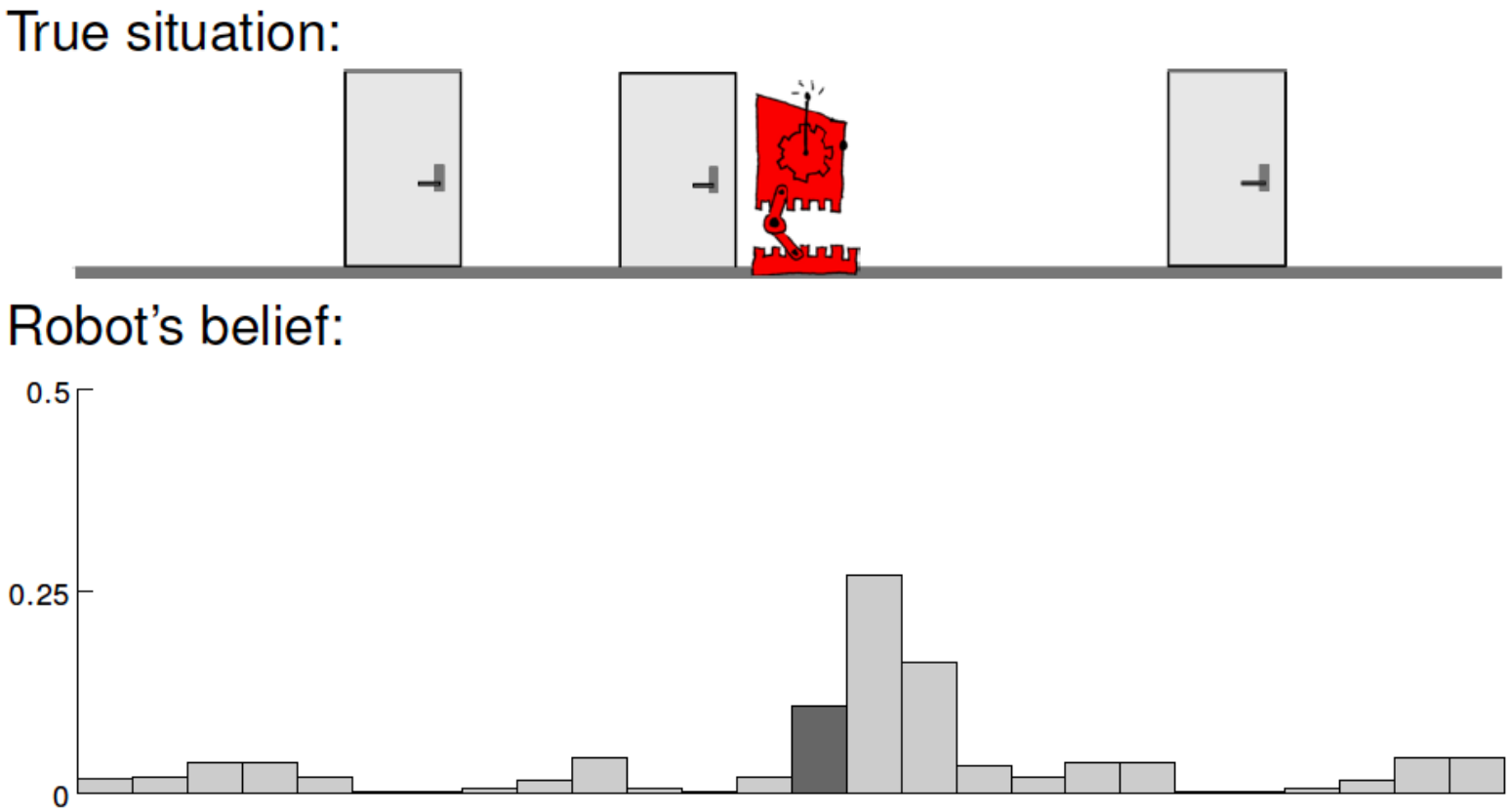


Robot's belief:



- Observations: *door* or **corridor**, 10% noise.
- Action: moves 3 (20%), 4 (60%), or 5 (20%) states.

Processus Décisionnels Markoviens Partiellement Observables



- Observations: *door* or **corridor**, 10% noise.
- Action: moves 3 (20%), 4 (60%), or 5 (20%) states.

Processus Décisionnels Markoviens

- Les MDPs et POMDPs ont été utilisés avec succès dans de nombreux domaines : routage dans les réseaux, étude de phénomènes biologiques et écologiques, gestion de dialogue, robotique exploratoire, problème de navigation,...
- Les MDPs et POMDPs sont mal adaptés à la formalisation de problèmes de décision multiagent :
  - 1 action par agent
  - 1 état pour chaque agent
  - les agents ont une observabilité partielle de l'état du système
  - le contrôle est décentralisé

Planification multiagent sous incertitude

- Revenons au cas multiagent
- **Plusieurs agents délibératifs** évoluant dans **un même environnement**
- Chaque agent est **autonome** et décide de ses actions : prise de décision distribuée (pas de décideur central, pas de communication ou bien restreinte)
- Décision séquentielle
- Incertitude

Planification multiagent sous incertitude

Dans les systèmes multiagents :

- l'issue d'une action d'un agent dépend souvent des actions des autres agents
- l'utilité d'un agent dépend souvent des états des autres agents
- nécessité de tenir compte des interactions entre agents et de coordonner les plans des agents
- les agents peuvent rarement se limiter à maximiser indépendamment leurs utilités individuelles

De plus :

- les agents ont souvent une observabilité partielle du système (environnement et autres agents)
- les agents ne connaissent donc pas les plans des autres agents
- les communications entre agents sont souvent limitées (bande passante limitée, coût en temps de communication, impossibilité physique de communiquer,...)

Problématique

Problématique

On souhaite former un groupe d'agents qui agit de manière rationnelle et coordonnée dans un environnement incertain et dynamique.
On s'intéressera ici aux agents coopératifs, c'est-à-dire ayant une fonction d'utilité commune qu'ils cherchent à maximiser.
Cadre non-coopératif abordé par la suite.

Problématique

Exemples d'applications

- Routage dans les réseaux
- Constellations de satellites
- Colonies de robots : navigation, exploration, missions de secours
- Gestion d'accès multiples (à un canal de communication par exemple)
- Réseaux de capteurs

Processus Décisionnels Markoviens Multiagent

Différents modèles proposés selon :

- le degré d'observabilité de l'état du système et des états des agents
- les indépendances entre les agents (transitions, observations, récompenses)
- la présence ou non de communication entre les agents

Différentes approches de résolution :

- Optimale ou approchée
- Centralisée ou décentralisée
- Exploitant ou non la structure du problème (contraintes de ressources, temporelles, structure des interactions)

Différents types d'observabilité

État du système

États des agents + état de l'environnement

Observabilité de l'état du système

- Individuellement Observable : chaque agent observe l'état du système
- Collectivement Totalement Observable : l'union des observations des agents permet de déduire l'état du système
- Collectivement Partiellement Observable : l'union des observations des agents permet de déduire une partie de l'état du système
- Non Observable : aucune observation de l'état du système

Différents types d'observabilité

État d'un agent

Défini selon le degré de délibération de l'agent.
Perceptions courantes de l'agent ou représentation interne de ses connaissances.

Observabilité de l'état d'un agent

- Totalement Observable
- Partiellement Observable
- Non Observable

Exemple de problème (1)

"Multiagent Tiger Domain" [NTY+03]

Exemple de problème (1)

"Multiagent Tiger Domain"

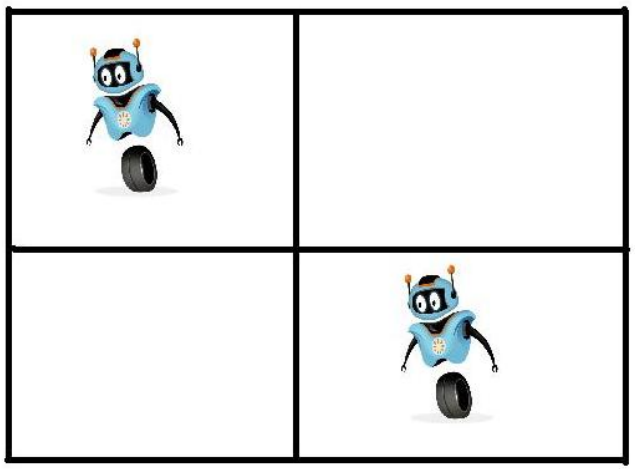
- Derrière l'une des portes un tigre, derrière l'autre un trésor
- 2 agents pouvant ouvrir individuellement ou conjointement une des portes
- Les agents peuvent indépendamment écouter s'ils entendent le tigre derrière l'une des portes. Les perceptions sont bruitées.
- S'ils ouvrent conjointement la porte derrière laquelle est le tigre, ils sont moins blessés que s'ils l'ouvrent individuellement (pénalité).
- Les agents reçoivent la récompense maximum lorsqu'ils ouvrent conjointement la porte du trésor

$\Rightarrow$ Les agents doivent se coordonner

Exemple de problème (2)

"Rencontre sur une grille" [BHZ05]

- Grille 2x2 et 2 robots
- Les robots ne perçoivent que les murs (contours de l'environnement)
- Pas de perception de la position de l'autre robot
- Les déplacements des robots sont incertains
- Objectif : les robots doivent rester le plus longtemps possible sur la même case



Exemple de problème (3)

"Multi-access broadcast channel" [BHZ05]

- Des nœuds émettent des messages en "broadcast" sur un canal commun.
- Les nœuds du canal ne peuvent pas émettre en même temps (collision).
- L'envoi d'un message sans collision rapporte une récompense.
- Le buffer d'envoi de chaque nœud a une capacité de 1 message. Lorsqu'un message est envoyé, l'ajout d'un nouveau message dans le buffer d'envoi est non-déterministe.
- Objectif : les nœuds cherchent à maximiser l'utilisation du canal.

Exemple de problème (4)

Exploration multi-robot [BM06, EMGST04]

- Un ensemble de m sites à explorer par un ensemble de n robots mobiles $n < m$
- Les robots ont des actions à réaliser sur chaque site. Il existe des dépendances entre les actions réalisées par des robots différents (contraintes de précédence, de simultanéité,...)
- Les robots ont des ressources limitées
- Chaque action peut échouer. L'exécution réussie d'une action (action menée à son terme sans erreur et en respectant les contraintes) rapporte une récompense.
- Objectif : maximiser les récompenses obtenues par le système

Processus Décisionnels de Markov Décentralisés

Type de contrôle	Centralisé		Décentralisé	
État du système collectivement totalement observable	Oui	Non	Oui	Non
Formalisme	MDP MMDP	POMDP	DEC-MDP	DEC-POMDP
			MTDP	

Table: Relations entre les DEC-POMDPs, DEC-MDPs, POMDPs et MDPs

MMDP (Multiagent Markov Decision Processes)

Définition

Un MMDP [Bou96] est défini par un tuple $< \alpha, \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} >$ tel que :

- α est le nombre d'agents du système.
- $\mathcal{S}$ correspond à l'ensemble des états s du système.
- $A =< A_1, \cdots, A_n >$ définit l'ensemble des actions jointes a des agents, A_i est l'ensemble des actions locales de l'agent i.
- $\mathcal{T}$ est un fonction de transition, elle donne la probabilité $T(s, a, s')$ que le système passe dans un état s' quand les agents exécutent l'action jointe a à partir de l'état s.
- $\mathcal{R}$ définit la fonction de récompense. $R(s, a, s')$ est la récompense obtenue par le système lorsqu'il passe d'un état s à un état s' en exécutant l'action a.

Processus Décisionnels de Markov Multi-agents

Résolution

- Un MMDP peut être vu comme un MDP ayant un grand espace d'états et d'actions.
- Résoudre un MMDP consiste à calculer une politique jointe $\pi = \langle \pi_1, \cdots, \pi_n \rangle$ où π_i correspond à la politique locale de l'agent i. Elle définit une fonction $\pi_i : S \rightarrow A_i$.
- Résolution à l'aide d'algorithmes classiques comme l'itération de la valeur.

Discussion

Les MMDP supposent que l'état du système soit individuellement observable ou bien que la communication soit gratuite $\Rightarrow$ peu réaliste dans les SMA.

Processus Décisionnels de Markov Partiellement Observables Décentralisés [Oli12]

Définition

Un DEC-POMDP est défini par un tuple $< \mathcal{S}, \mathcal{A}, \mathcal{T}, \Omega, \mathcal{O}, \mathcal{R} >$ tel que :

- $\mathcal{S}$ définit l'état global du système.
- $\mathcal{A} = \langle A_1, \cdots, A_n \rangle$ est l'ensemble des actions jointes et A_i définit l'ensemble des actions a_i de l'agent i.
- $\mathcal{T} = S \times A \times S \rightarrow [0, 1]$ définit la fonction de transition. $T(s, a, s')$ correspond à la probabilité que le système passe d'un état s à un état s' lorsque l'action jointe a est exécutée.
- $\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_n$ est l'ensemble des observations des agents et Ω_i est l'ensemble des observations de l'agent i.
- $\mathcal{O} = S \times A \times S \times \Omega \rightarrow [0, 1]$ définit la fonction d'observation. $\mathcal{O}(s, a, s', o = \langle o_1, \cdots, o_n \rangle)$ correspond à la probabilité que chaque agent i observe o_i lorsque les agents exécutent l'action jointe a à partir de l'état s et que le système arrive dans l'état s'.
- $\mathcal{R}$ définit la fonction de récompense. $R(s, a, s')$ est la récompense obtenue par le système lorsque les agents exécutent l'action a à partir de l'état s et arrivent dans l'état s'.

Processus Décisionnels de Markov Décentralisés

Définition

Un DEC-MDP est un DEC-POMDP où l'état du système est collectivement totalement observable :

$$\mathcal{O}(s, a, s', o = \langle o_1, \cdots, o_n \rangle) > 0 \text{ alors } P(s' | \langle o_1, \cdots, o_n \rangle) = 1$$

Attention : cela ne signifie pas que chaque agent observe totalement son état local. De plus, l'état global du système n'est pas pour autant individuellement observable.

Processus Décisionnels de Markov Décentralisés

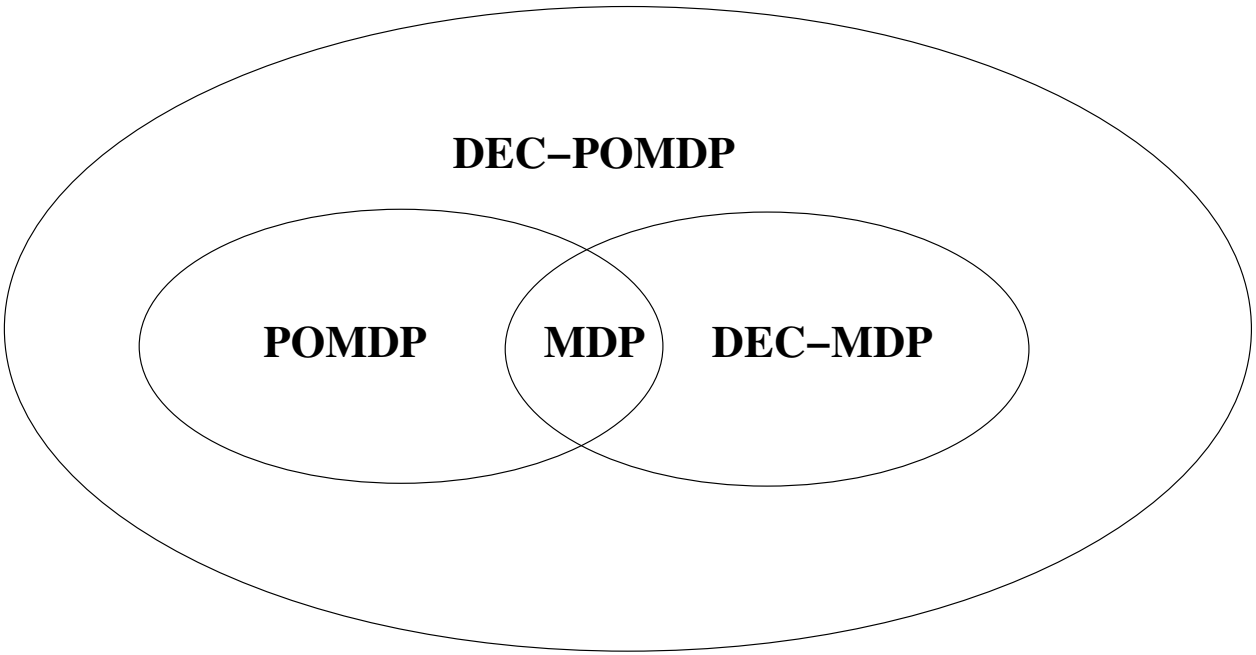


Figure: Relations entre les différents types de Processus Décisionnels de Markov

Processus Décisionnels de Markov Décentralisés

Type de contrôle	Centralisé		Décentralisé	
État du système collectivement totalement observable	Oui	Non	Oui	Non
Formalisme	MDP MMDP	POMDP	DEC-MDP	DEC-POMDP
			MTDP	

Table: Relations entre les DEC-POMDPs, DEC-MDPs, POMDPs et MDPs

Multiagent Tiger Domain

- États : S = {SL, SR} indique derrière quelle porte est le tigre
- Actions : $A_1 = A_2 = \{\text{OpenLeft, OpenRight, Listen}\}$
- Observations : $\Omega_1 = \Omega_2 = \{HL, HR\}$, indépendantes et bruitées
- La fonction de transition $\mathcal{T}$ spécifie, chaque fois qu'une porte est ouverte, où sont replacés le tigre et le trésor
- La fonction d'observation retourne soit *HL* soit *HR* avec différentes probabilités suivant l'action jointe et l'état du monde
- La fonction de récompense traduit les pénalités et les récompenses obtenues pour ouvrir individuellement ou conjointement une des portes. Par exemple : Si l'état est *SL* et l'action réalisée est $\langle OpenRight, OpenRight \rangle$ alors les agents obtiennent la récompense 20. Si l'état est SR, ils obtiennent la récompense -50.

Rencontre sur une grille

- États : positions des 2 robots (4 positions possibles pour chaque robot $\rightarrow$ 16 états)
- Actions : $A_1 = A_2 = \{\text{up, down, right, left, stay}\}$. Un déplacement vers un mur conduit à rester sur place.
- Observations : $\Omega_1 = \Omega_2 = \{WL, WR, WL\&WR, \emptyset\}$
- La fonction de transition $\mathcal{T}$ détermine si le robot se déplace dans la direction souhaitée ou non.
- La fonction d'observation retourne une des observations avec différentes probabilités suivant l'action jointe et l'état du monde.
- Fonction de récompense : les robots reçoivent une récompense de 1 quand ils sont sur la même case, 0 sinon.

Multi-Access Broadcast Channel

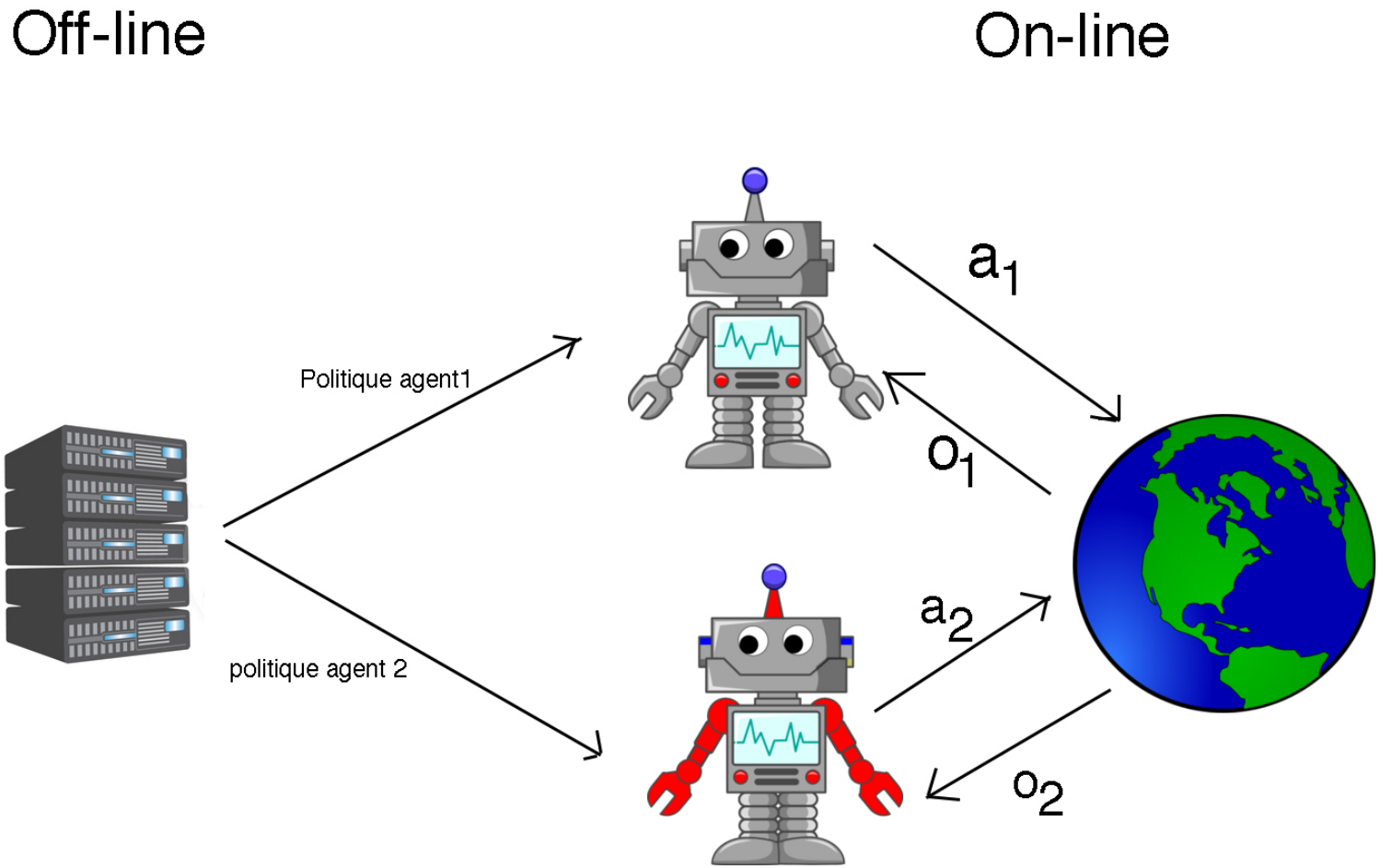
- États : 4 états possibles suivant que le buffer de chaque agent est vide ou non
- Actions : $A_1 = A_2 = \{\text{Send, Nothing}\}$
- Observations : chaque nœud observe si son buffer est vide ou non et si l'étape précédente à conduit à une collision, un envoi réussit ou rien $\rightarrow$ 5 observations.
- La fonction de transition $\mathcal{T}$ spécifie, comment est re-rempli un buffer quand il est vide (probabilité qu'un message apparaisse).
- Fonction d'observation déterministe
- Fonction de récompense : récompense de 1 quand un message est envoyé avec succès, 0 sinon

Politique distribuée

Résolution

- Résoudre un DEC-POMDP consiste à calculer une politique jointe $\pi = \langle \pi_1, \cdots, \pi_n \rangle$ où π_i correspond à la politique locale de l'agent i.
- Cette politique peut être calculée *off-line* de manière centralisée.
- La politique est toujours exécutée de manière distribuée : chaque agent doit donc être en mesure de déterminer quelle action exécuter à partir de ses observations.

Politique distribuée



Politique distribuée

Historique observations-actions

Un historique d'observations-actions pour un agent correspond à la séquence d'actions et d'observations réalisées par l'agent :

$$\bar{\theta}_i^t = (o_i^1, a_i^1, o_i^2, a_i^2, \cdots, o_i^t, a_i^t)$$

Historique observations

Un historique d'observations pour un agent correspond à la séquence d'observations réalisées par l'agent :

$$\bar{\theta}_i^t = (o_i^1, o_i^2, \cdots, o_i^t)$$

Politique distribuée

Politique : cas général

Dans le cas général, une politique fait correspondre une action à chaque historique observations-actions.

Politique déterministe

Une politique déterministe fait correspondre une action à chaque historique d'observations.

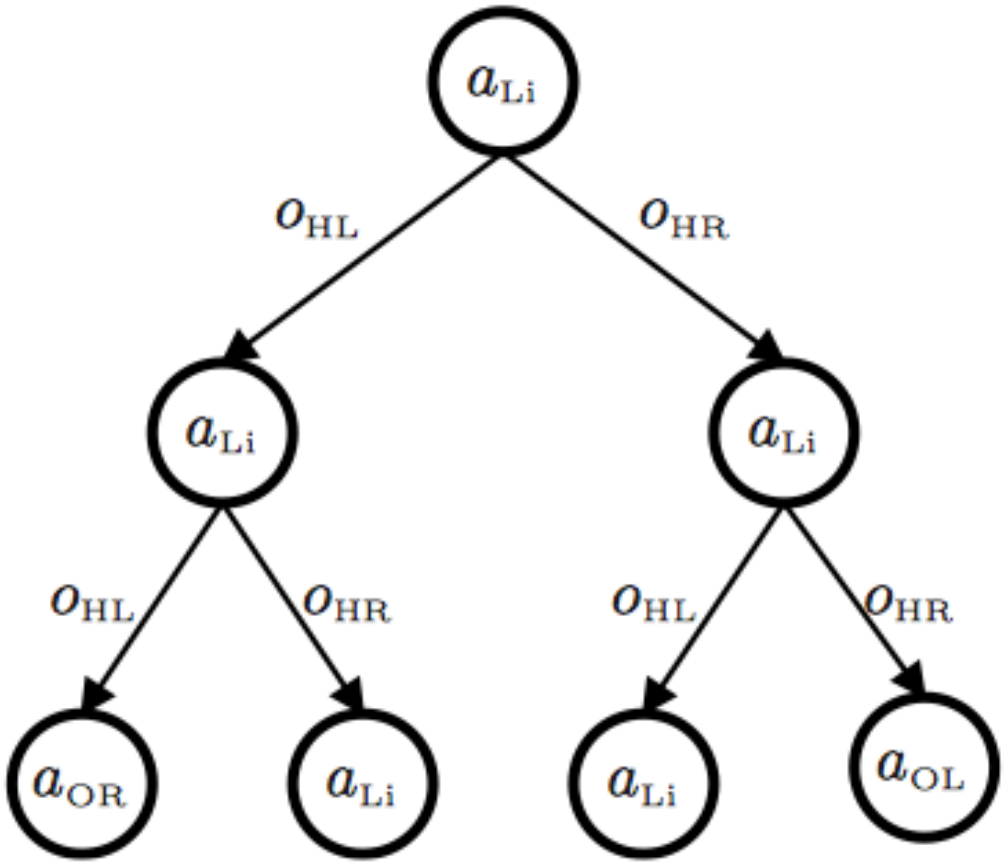
Politique optimale d'un DEC-POMDP

Il existe une politique optimale déterministe pour tout DEC-POMDP.
La politique optimale d'un DEC-POMDP à n agents est de la forme $\pi = \langle \pi_1, \cdots, \pi_n \rangle$ où :

$$\pi_i = \bar{O}_i \rightarrow A_i$$

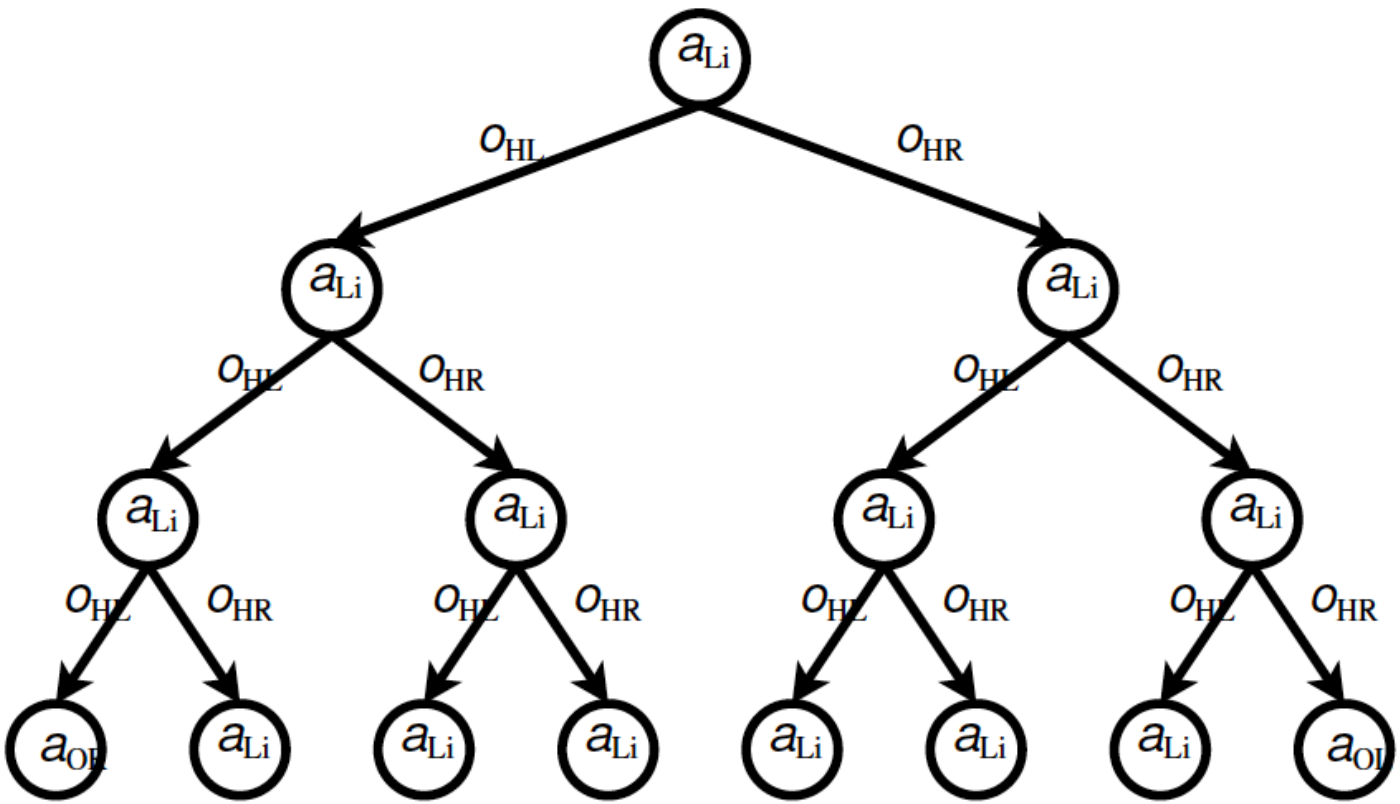
Politique distribuée

Exemple de politique individuelle pour DEC-Tiger (politique de 1 agent) avec horizon 3 :



Politique distribuée

Exemple de politique individuelle pour DEC-Tiger (politique de 1 agent) avec horizon 4 :

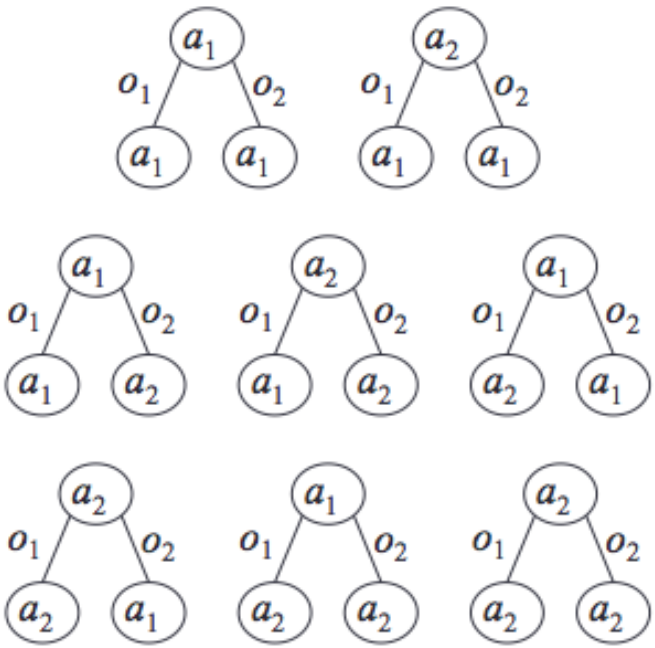


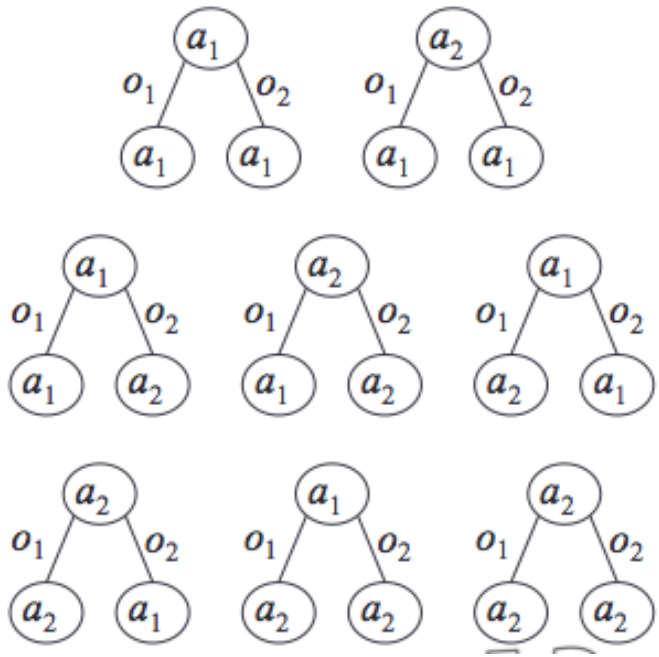
Calcul de politiques optimales

- Mais calculer une politique optimale pour un DEC-POMDP n'est pas simple !
- Algorithme naif :
  - Pour chaque politique jointe π, calculer $V(\pi)$
  - Sélectionner le max de $V(\pi)$
- Le nombre de politiques jointes est doublement exponentiel !
- Résoudre de manière optimale un DEC-POMDP est NEXP-complet [BZI02]

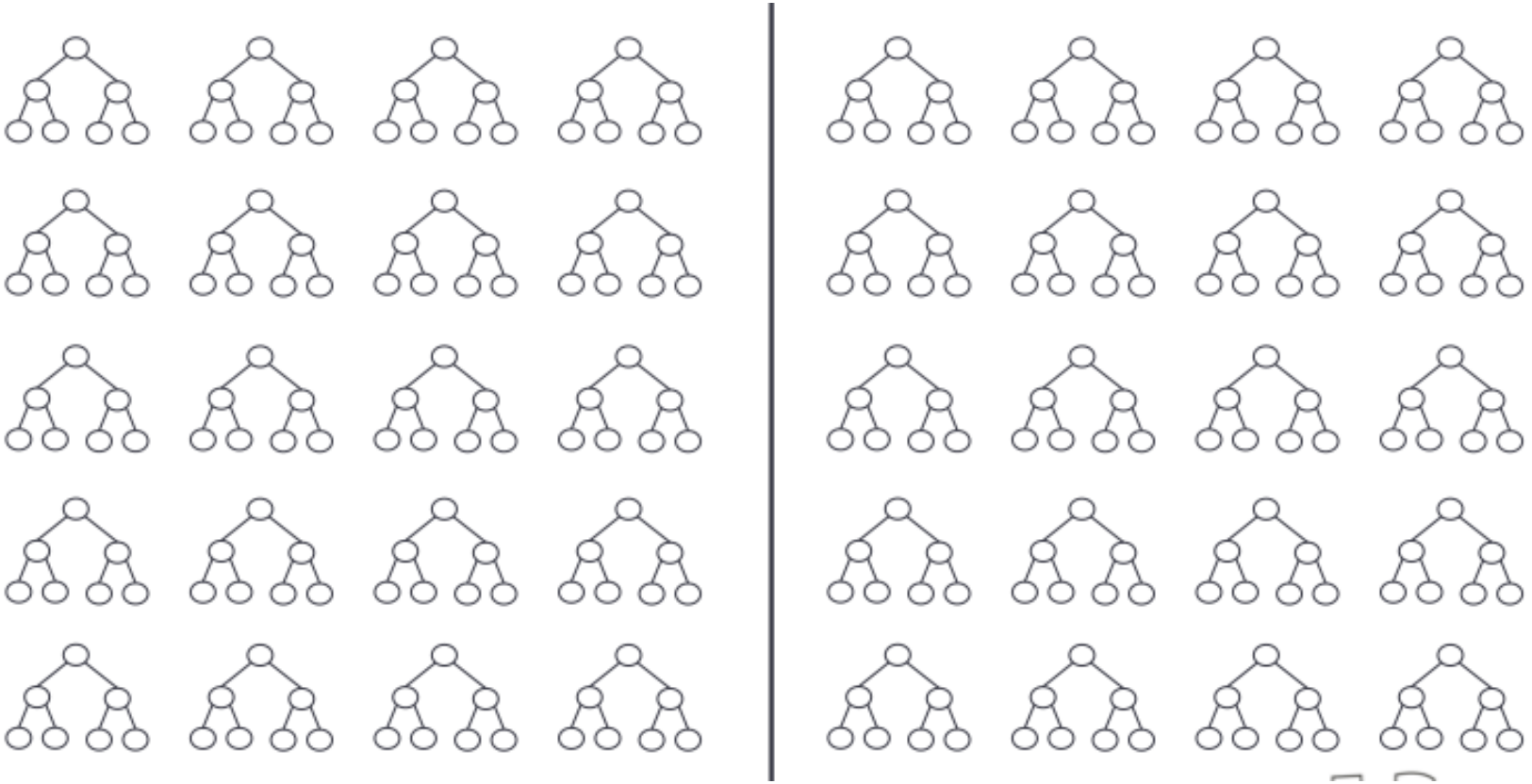
Calcul de politiques optimales

Calcul de politiques optimales





Calcul de politiques optimales



Classes de complexité

Cas particuliers :

- État du système individuellement partiellement observable : le problème peut se réduire à un MDP dont la résolution est P-Complète.
- État global du système n'est pas observable : le problème se réduità un MDP non observable (NOMDP) connu pour être NP-complet.

Observabilité de l'état du système	Individuellement Observable	Collectivement Observable	Collectivement Partiellement Observable	Non Observable
Complexité	P-complet	NEXP-Complet	NEXP-Complet	NP-Complet

Table: Observabilité et Complexité en temps

Résolution des DEC-POMDPs

- Approches optimales pour la résolution de DEC-POMDPs :
  - Élimination itérative de stratégies dominées
  - Heuristiques guidant la résolution (type A*)
  - Identifier des propriétés des problèmes pouvant être exploitées afin de diminuer la complexité de résolution : indépendances des observations, des transitions, existence d'états buts, localité des interactions,...
  - Ramener le problème à un MDP déterministe et à états continus
- Approches alternatives :
  - Chercher une solution approchée de la solution optimale.

Références I


D. Bernstein, E.A Hansen, and S. Zilberstein, *Bounded policy iteration for decentralized pomdps*, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (Edinburgh, Scotland), 2005.


A. Beynier and A.I Mouaddib, *An iterative algorithm for solving constrained decentralized markov decision processes*, The Twenty-First National Conference on Artificial Intelligence (AAAI-06), 2006.


C. Boutilier, *Planning, learning and coordination in multiagent decision processes*, Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge (TARK'96), 1996, pp. 195–201.


D. Bernstein, S. Zilberstein, and N. Immerman, *The complexity of decentralized control of mdps*, Mathematics of Operations Research, 2002, pp. 27(4):819–840.

Références II


K. Decker and V. Lesser, *Designing a Family of Coordination Algorithms*, Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95) (1995), 73–80.


R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun, *Approximate solutions for partially observable stochastic games with common payoffs*, Proceedings of the Third Joint Conference on Autnomous Agents and Multi Agent Systems, 2004.


N. R. Jennings, *Coordination techniques for distributed artificial intelligence*, 1996.


R. Nair, M. Tambe, M. Yokoo, S. Marsella, and D.V Pynadath, *Taming decentralized pomdps: Towards efficient policy computation for multiagent settings*, Proceedings of the International Joint Conference on Artificial Intelligence, 2003, pp. 705–711.

Références III


Frans A. Oliehoek, *Decentralized POMDPs*, Adaptation, Learning, and Optimization, vol. 12, pp. 471–503, Springer Berlin Heidelberg, Berlin, Germany, 2012.


M. L Puterman, *Markov decision processes : discrete stochastic dynamic programming*, Wiley-Interscience, New York, 2005.


S. Russell and P. Norvig, *Artificial intelligence : A modern approach*, Prentice Hall Series, 2003.


G Weiss, *Multiagent systems a modern approach to distributed artificial intelligence*, MIT Press, 1999.


M. Wooldridge, *An introduction to multiagent systems*, John Wiley and Sons, 2002.