Planification et coordination multiagents sous incertitude

Aurélie Beynier

CoCoMa, Master 2 ANDROIDE

25 octobre 2016

Résolution des DEC-POMDPs

Complexité

Résoudre de manière optimale un DEC-POMDP est NEXP-complet [BZI02] (même pour 2 agents).

- 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.

Plan

1. Résolution Optimale
   - Propriétés des problèmes et complexité
   - Programmation dynamique
   - Élimination de stratégies dominées
   - Heuristiques guidant la résolution : MAA*
   - MDP à états continus
   - Résultats expérimentaux
2. Résolution Approchée
   - JESP
   - Limitation de la mémoire
   - Horizon Infini
3. MADP Toolbox
4. Conclusion

Propriétés recherchées

Objectif

Identifier des propriétés des problèmes traités qui puissent être exploitées lors de la résolution afin de diminuer la complexité du calcul de la solution optimale.

Propriétés intéressantes

- Existence d'indépendances entre les agents : observations indépendantes, transitions indépendantes
- Existence d'états buts
- Situations d'interaction restreintes

Transitions Indépendantes

Définition

Un DEC-POMDP est dit à transitions indépendantes [BZLG03] si la fonction de transition peut être factorisée en un produit de probabilités tel que :

$$P = \prod_{i=1}^{n} P_i \text{ où } P_i = Pr(s_i'|s_i, a_i).$$

Lorsqu'un DEC-POMDP est à transitions indépendantes, la probabilité qu'un agent i passe d'un état s_i à un état s_i' ne dépend que de l'action a_i qu'il a exécutée. Les actions des autres agents n'ont pas d'influence sur la transition de l'agent i.

Transitions Indépendantes

Pour deux agents :

$$\forall s_1, s_1' \in S_1, \forall s_2, s_2' \in S_2, \forall a_1 \in A_1, \forall a_2 \in A_2,$$

$$Pr(s_1'|(s_1, s_2), a_1, a_2, s_2') = Pr(s_1'|s_1, a_1) \wedge$$

$$Pr(s_2'|(s_1, s_2), a_2, a_1, s_1') = Pr(s_2'|s_2, a_2)$$

Observations Indépendantes

Définition

Un DEC-POMDP est dit à observations indépendantes [GZ04] si la probabilité d'observation $\mathcal{O}$ peut être décomposée en n probabilités d'observations $\mathcal{O}_i$ telles que :

$$\mathcal{O}_i(o_i|\langle s_1, \cdots, s_n\rangle, \langle a_1, \cdots, a_n\rangle, \langle s'_1, \cdots, s'_n\rangle, \langle o_1, \cdots, o_{i-1}, o_{i+1}, \cdots, o_n\rangle) = Pr(o_i|s_i, a_i, s'_i).$$

L'observation-indépendance caractérise les problèmes où les observations de chaque agent sont indépendantes des actions, états et observations des autres agents.

Observations Indépendantes

Pour deux agents :

$$\forall o_1 \in \Omega_1, \forall o_2 \in \Omega_2, \forall s = (s_1, s_2), s' = (s'_1, s'_2) \in \mathcal{S}, \forall a_1 \in A_1, \forall a_{\in} A_2,$$

$$Pr(o_1 | (s_1, s_2), a_1, a_2, (s'_1, s'_2), o_2) = Pr(o_1 | s_1, a_1, s'_1) \wedge$$

$$Pr(o_2 | (s_1, s_2), a_1, a_2, (s'_1, s'_2), o_1) = Pr(o_2 | s_2, a_2, s'_2)$$

$$\mathcal{O}(o_1, o_2 | (s_1, s_2), a_1, a_2, (s'_1, s'_2)) =$$

$$Pr(o_1 | (s_1, s_2), a_1, a_2, (s'_1, s'_2), o_2) \times Pr(o_2 | (s_1, s_2), a_1, a_2, (s'_1, s'_2), o_1)$$

Orientation par un but

Définition

Un DEC-POMDP est dit orienté par des buts [GZ04] si les conditions suivantes sont remplies :

1. Il existe un sous ensemble non vide $\mathcal{G}$ de l'ensemble des états $\mathcal{S}$ représentant les états but du système. Au moins un état $g \in \mathcal{G}$ est accessible par une politique jointe.
2. Le problème a un horizon fini T.
3. Toute action a_i d'un agent i a un cout $C(a_i) < 0$.
4. La fonction de récompense est telle que $R(s, \langle a_1, \cdots, a_n \rangle, s') = \sum_{i=1}^{n} C(a_i)$.
5. Si à l'horizon T, le système a atteint un état but $s \in \mathcal{G}$ alors, une récompense supplémentaire est attribuée au système pour avoir atteint un état but.

Un tel DEC-POMDP est communément noté GO-DEC-POMDP.

Impact sur la complexité

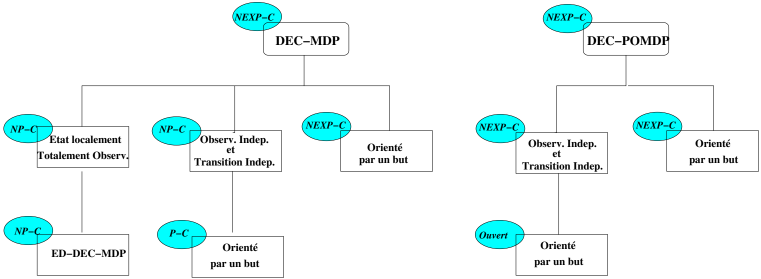


Figure: Complexité des problèmes en fonction des propriétés vérifiées

Situations d'interactions restreintes

ND-POMDP : Networked Distributed POMDP [NPMM05]

- Chaque agent n'interagit qu'avec un nombre restreint de voisins : par exemple les nœuds voisins dans un réseau
- Structure d'interactions statique
- Pas de dépendance entre les observations et les transitions. Uniquement des dépendances sur les récompenses.
- Le calcul de la politique d'un agent i ne doit tenir compte que des politiques des agents avec lequel i est en interaction.

Situations d'interactions restreintes

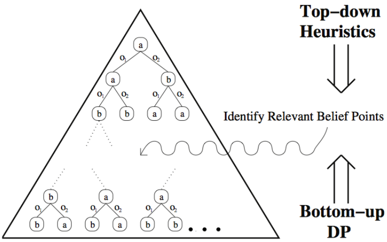
DyLim = Dynamic Local Interaction Model [CM11]

- Structure des interactions non définie a priori et pouvant être dynamique
- Séparation du problème en 2 composantes :
  - Le problème de décision individuel de chaque agent ne tenant pas compte des autres agents
  - Le problème de coordination traitant l'influence des décisions individuelles sur les agents en interaction
- Pour chaque état, on dispose de 2 valeurs, V^{ind} et V^{coord} et on sélectionne l'action offrant le meilleur compromis entre ces 2 valeurs.

Programmation dynamique au cadre multiagent

Extension du principe de la programmation dynamique utilisé dans le cadre des POMDPs [HBZ04] :

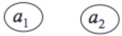
- Approche bottom-up
- Planification à horizon fini
- A partir d'un ensemble d'arbres de politiques de profondeur t, on construit les arbres de politiques de profondeur $t+1$



[SZ07]

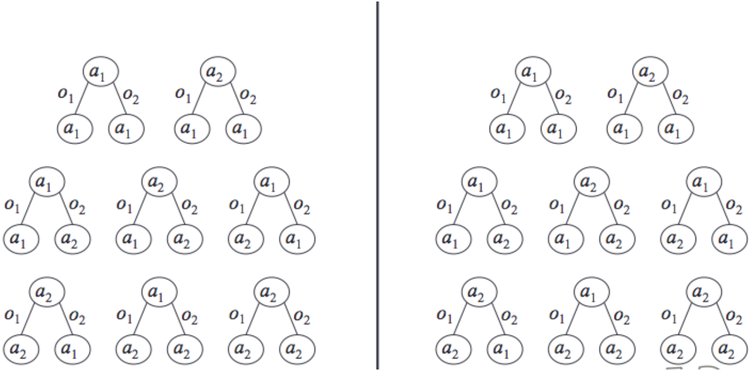
Nombre de politiques

h = 1



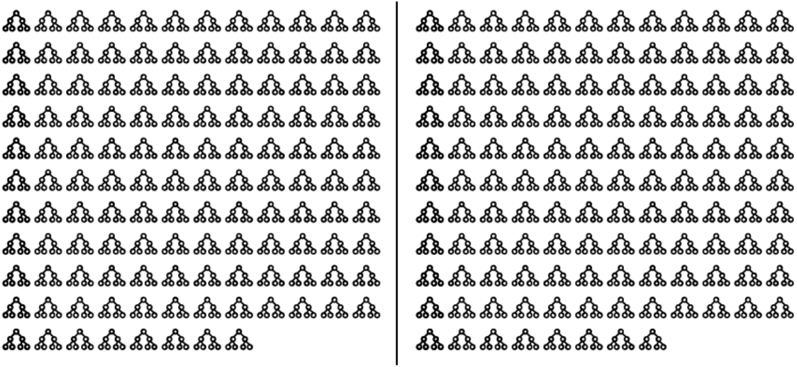
Nombre de politiques

h = 2



Nombre de politiques

$h = 3$



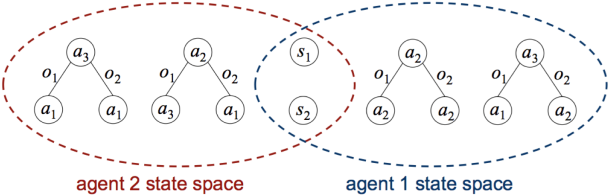
Nombre de politiques individuelles

horizon	nb. pol. indiv.
1	2
2	8
3	128
4	32768
5	$2.1475e^{+}09$
6	$9.2234e^{+}18$
7	$1.7014e^{+}38$
8	$5.7896e^{+}76$

$\rightarrow$ il faut trouver plus efficace

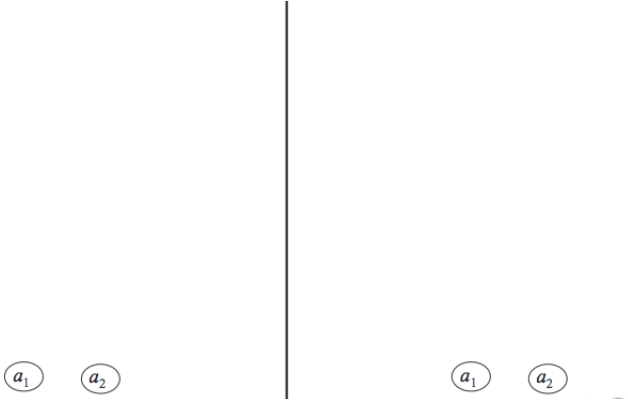
Élimination itérative de stratégies dominées

- A chaque étape, il est possible d'éliminer des stratégies (arbres de politiques) dominées [HBZ04]
- A l'étape suivante, on étend uniquement les arbres non éliminés
- **Difficulté :** la valeur d'une stratégie individuelle dépend des stratégies des autres agents. Ces stratégies doivent donc être prises en compte lors de la recherche des stratégies dominées
- $\rightarrow$ on définit des états de croyances sur les états du système et les politiques des autres agents



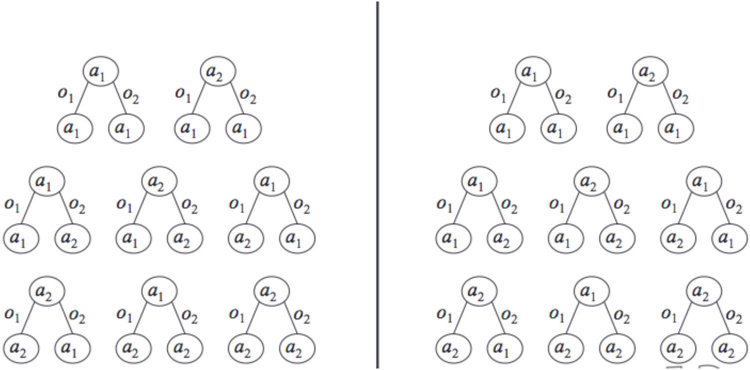
Élimination itérative de stratégies dominées

h = 1



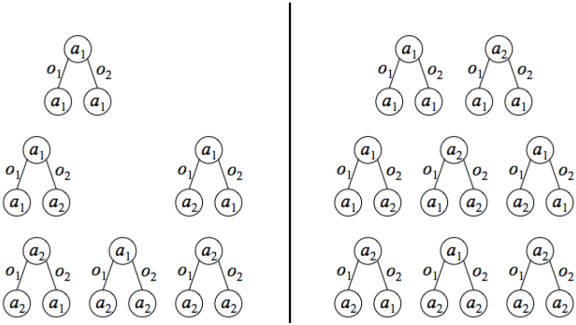
Élimination itérative de stratégies dominées

h = 2



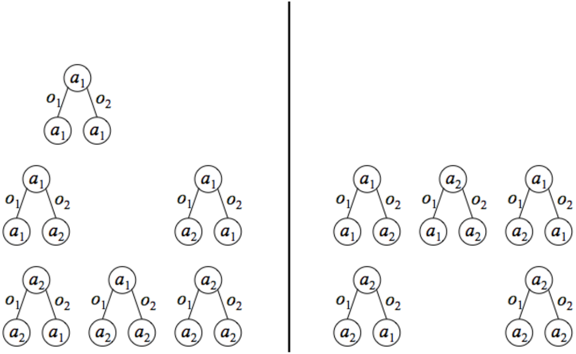
Élimination itérative de stratégies dominées

h = 2, élagage des stratégies de l'agent 1



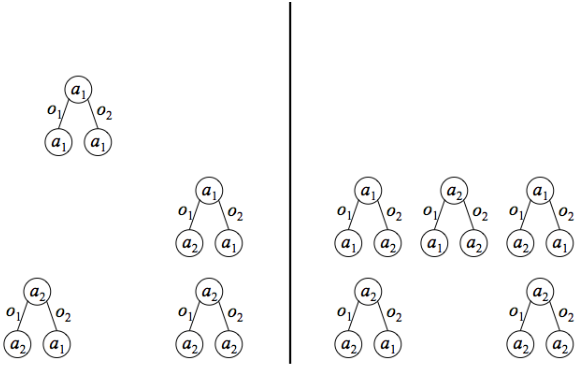
Élimination itérative de stratégies dominées

h = 2, élagage des stratégies de l'agent 2



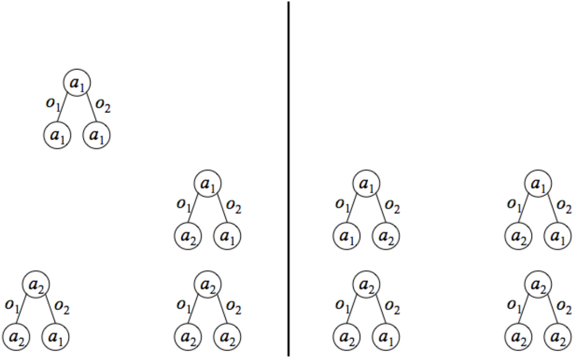
Élimination itérative de stratégies dominées

h = 2, élagage des stratégies de l'agent 1



Élimination itérative de stratégies dominées

h = 2, élagage des stratégies de l'agent 2

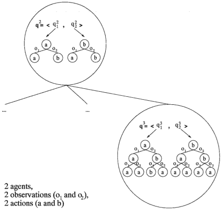


Élimination itérative de stratégies dominées

- Quand plus aucune stratégie (de profondeur t) ne peut plus être éliminée, on construit les arbres de profondeur $t + 1$ à partir des stratégies non éliminées.
- La programmation dynamique couplée à l'élimination itérative de stratégies dominées permet de calculer une politique jointe optimale.

Recherche Heuristique

- Approche top-down : on crée toutes les politiques jointes en partant de la première étape de décision
- Planification à horizon fini



- Développer tout l'arbre des politiques jointes possibles est trop coûteux
- On utilise des heuristiques afin de déterminer quelle est la branche la plus prometteuse

Recherche Heuristique

MAA* : Multiagent A* [SCZ05]

- $F(n) = G(n) + H(n)$
- $G(n)$: récompense perçue depuis la racine de l'arbre jusqu'à l'étape n
- $H(n)$: estimation de la récompense pouvant être perçue à partir de l'étape n
- Comme dans A*, on doit utiliser une heuristique admissible : la récompense doit être surestimée
- Heuristique admissible : valeur du POMDP sous-jacent
- MAA* calcule une solution optimale
- Cas moyen plus efficace que la recherche exhaustive, pire cas équivalent à la recherche exhaustive

Conversion en MDP à états continus

Rappel : la plupart des algorithmes planifient de manière centralisée des politiques qui sont exécutées de manière décentralisée

Principe de l'approche [DABC14] :

- Définir dans un état d'occupation les connaissances communes : distribution de probabilités sur les états du systèmes et les historiques possibles des agents
- un état d'occupation est une distribution $\nu(s, \theta^t) = Pr(s, \theta^t | \pi^{0:t-1}, b_0)$
- L'état d'occupation est un statistique suffisante pour optimiser la politique future

Conversion en MDP à états continus

- Le DEC-POMDP est transformé en un MDP à états continus (états d'occupation)
- Les actions représentent les politiques séparables possibles.
- Les méthodes de résolution des POMDPs (et MDP à états continus) peuvent être appliquées (donc planification centralisée MAIS avec exécution distribuée)
- Obtention d'une politique jointe optimale
- Horizon fini ou infini
- Approche la plus efficace actuellement

Recherche Heuristique

h	MILP	DP-LPC	DP-IPG	GMAA — Q_{BG} IC	ICE	heur
BroadcastChannel, ICE solvable to $h = 900$						
2	0.38	≤ 0.01	0.09	≤ 0.01	≤ 0.01	≤ 0.01
3	1.83	0.50	56.66	≤ 0.01	≤ 0.01	≤ 0.01
4	34.06	*	*	≤ 0.01	≤ 0.01	≤ 0.01
5	48.94			≤ 0.01	≤ 0.01	≤ 0.01
Dec-Tiger, ICE solvable to $h = 6$						
2	0.69	0.05	0.32	≤ 0.01	≤ 0.01	≤ 0.01
3	23.99	60.73	55.46	≤ 0.01	≤ 0.01	≤ 0.01
4	*	–	2286.38	0.27	≤ 0.01	0.03
5			–	21.03	0.02	0.09
FireFighting (2 agents, 3 houses, 3 firelevels), ICE solvable to $h \gg 1000$						
2	4.45	8.13	10.34	≤ 0.01	≤ 0.01	≤ 0.01
3	–	–	569.27	0.11	0.10	0.07
4			–	950.51	1.00	0.65
GridSmall, ICE solvable to $h = 6$						
2	6.64	11.58	0.18	0.01	≤ 0.01	≤ 0.01
3	*	–	4.09	0.10	≤ 0.01	0.42
4			77.44	1.77	≤ 0.01	67.39
Recycling Robots, ICE solvable to $h = 70$						
2	1.18	0.05	0.30	≤ 0.01	≤ 0.01	≤ 0.01
3	*	2.79	1.07	≤ 0.01	≤ 0.01	≤ 0.01
4		2136.16	42.02	≤ 0.01	≤ 0.01	0.02
5		–	1812.15	≤ 0.01	≤ 0.01	0.02
Hotel 1, ICE solvable to $h = 9$						
2	1.92	6.14	0.22	≤ 0.01	≤ 0.01	0.03
3	315.16	2913.42	0.54	≤ 0.01	≤ 0.01	1.51
4	–	–	0.73	≤ 0.01	≤ 0.01	3.74
5			1.11	≤ 0.01	≤ 0.01	4.54
9			8.43	0.02	≤ 0.01	20.26
10			17.40	#	#	
15			283.76			
Cooperative Box Pushing (Q_{POMDP}), ICE solvable to $h = 4$						
2	3.56	15.51	1.07	≤ 0.01	≤ 0.01	≤ 0.01
3	2534.08	–	6.43	0.91	0.02	0.15
4	–		1138.61	*	328.97	0.63

	problem primitives n	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{A}_i\rvert$	$\lvert\mathcal{O}_i\rvert$
Dec-Tiger	2	2	3	2
BroadcastChannel	2	4	2	2
GridSmall	2	16	5	2
Cooperative Box Pushing	2	100	4	5
Recycling Robots	2	4	3	2
Hotel 1	2	16	3	4
FireFighting	2	432	3	2

Conversion en MDP à états continus [DABC14]

The multi-agent tiger problem ($|S| = 2, |Z| = 4, |A| = 9, K = 3$)

T	MILP	LPC	IPG	ICE	FB-HSVI(ρ) 0	FB-HSVI(ρ) 1	FB-HSVI(ρ) 2	$v_\epsilon(\eta^0)$
2	–	0.17	0.32	0.01	0.05	0.03	**0.03**	−4.00
3	4.9	1.79	55.4	0.01	2.17	0.06	**0.40**	5.1908
4	72	534	2286	108	9164	2.66	**1.36**	4.8027
5				347		22.2	**9.65**	7.0264
6						171.3	**24.42**	10.381
7							**33.11**	**9.9935**
8							**41.21**	**12.217**
9							**58.51**	**15.572**
10							**65.57**	**15.184**
The recycling-robot problem ($|S| = 4, |Z| = 4, |A| = 9, K = 1$)								
2	–	–	0.30	36	0.03	0.02	**0.01**	7.000
3	–	–	1.07	36	0.05	0.47	**0.10**	10.660
4	–	–	42.0	72	0.85	0.65	**0.30**	13.380
5	–	–	1812	72	1.52	0.87	**0.34**	16.486
10					5.06	2.83	**0.52**	31.863
30					62.8	37.9	**1.13**	93.402
70						78.1	**2.13**	216.47
100						259	**2.93**	**308.78**
The mars-rovers problem ($|S| = 256, |Z| = 81, |A| = 36, K = 3$)								
2	–	–	83	1.0	0.21	0.09	**0.10**	5.80
3	–	–	389	1.0	2.84	0.21	**0.23**	9.38
4				103	104.2	1.73	**0.47**	10.18
5						6.38	**0.82**	**13.26**
6						8.16	**3.97**	**18.62**
7						11.13	**5.81**	**20.90**
8						35.49	**22.8**	**22.47**
9						57.47	**26.5**	**24.31**
10						316.2	**62.7**	**26.31**

- Time and value on benchmarks
- Blank space = algorithm over time (200s)
- Red for fastest and previously unsolvable horizons
- K is the largest history window used

Inria informatiques mathématiques — MIT Massachusetts Institute of Technology — UPMC Sorbonne Universités

Résolution approchée

Principe

- Sacrifier les exigences concernant l'optimalité de la solution et chercher une solution approchant la solution optimale.
- Donner des garanties concernant le type de solution (équilibre ? optimum local ?) ou la distance par rapport à l'optimal (difficile en pratique).

JESP : Joint Equilibrium Search for Policies [NTY+03]

Principe

- Amélioration itérative des stratégie des agents à horizon fini
- A chaque itération on améliore la politique d'un agent en gardant les politiques des autres agents fixes
- Calcul de la meilleure réponse aux stratégies (fixées) des autres agents
- On s'arrête quand plus aucune stratégie ne peut être améliorée

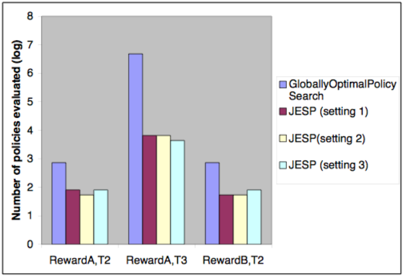
Solution

- Convergence garantie vers un optimum local
- Équilibre de Nash
- Pas de borne à l'optimal

JESP : Joint Equilibrium Search for Policies [NTY+03]

Améliorations

- Inclure les principes de la programmation dynamique en créant un POMDP augmenté : DP-JESP
- Utiliser les DCOP pour le cas des ND-POMDPs : LID-JESP [NPMM05]



Programmation dynamique à mémoire bornée [SZ07]

MBDP : Memory Bounded Dynamic Programming

Principe

- Borner le nombre de politiques à horizon fini gardées à chaque étape de l'algorithme de programmation dynamique : `maxTrees`
- Les politiques gardées sont sélectionnées à partir d'heuristiques (après élimination des stratégies dominées)
- Combinaison des approches top-down et bottom-up
- Complexité linéaire en l'horizon du problème

Solution

- Qualité dépend de `maxTrees`
- Pas de borne à l'optimal
- De nombreuses déclinaisons : IMBDP, MBDP-OC, ...

Programmation dynamique à mémoire bornée [SZ07]

	MABC Problem				Tiger Problem				
Horizon	Optimal	PBDP	Random	**MBDP**	Optimal	JESP	PBDP	Random	**MBDP**
2	2.00	2.00	1	**2.00**	-4.00	-4.00	-4.00	-92.44	**-4.00**
3	2.99	2.99	1.49	**2.99**	5.19	-6.00	5.19	-138.67	**5.19**
4	3.89	3.89	1.99	**3.89**	4.80	?	4.80	-184.89	**4.80**
5	-	4.70	2.47	**4.79**	-	?	-	-231.11	**5.38**
6	-	5.69	2.96	**5.69**	-	?	-	-277.33	**9.91**
7	-	6.59	3.45	**6.59**	-	?	-	-323.56	**9.67**
8	-	7.40	3.93	**7.49**	-	-	-	-369.78	**9.42**
9	-	-	4.41	**8.39**	-	-	-	-416.00	**12.57**
10	-	-	4.90	**9.29**	-	-	-	-462.22	**13.49**
100	-	-	48.39	**90.29**	-	-	-	-4,622.22	**93.24**
1,000	-	-	483.90	**900.29**	-	-	-	-46,222.22	**819.01**
10,000	-	-	4,832.37	**9,000.29**	-	-	-	-462,222.22	**7930.68**
100,000	-	-	48,323.09	**90,000.29**	-	-	-	-4,622,222.22	**78252.18**

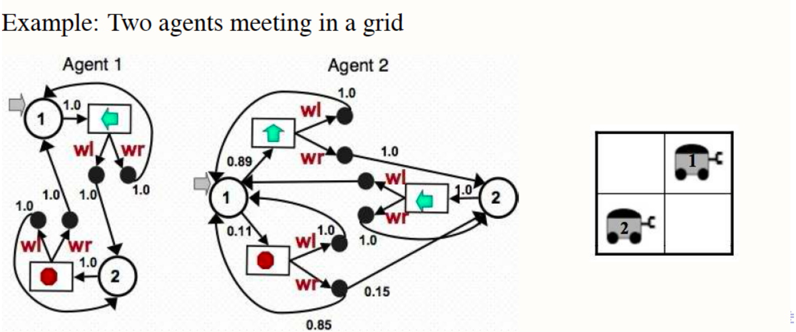
Horizon Infini

Problématique

Comment représenter une politique à horizon infini ?

Solution

Utilisation de contrôleurs stochastiques à états finis



Horizon Infini

- Amélioration de contrôleurs stochastiques à états finis par programmation linéaire [BHZ05]
- Formulation par programme non linéaire [ADZ07]
- Optimisation des paramètres des contrôleurs par Expectation Maximization [KZ12]

Size	DEC-BPI	NLP	EM	DEC-BPI	EM
1	4.687	9.1	9.05	< 1s	< 1s
2	4.068	9.1	9.05	< 1s	< 1s
3	8.637	9.1	9.05	2s	1.7s
4	7.857	9.1	9.05	5s	4.62s

Table 1: Broadcast channel: Policy value, execution time

MADP Toolbox

- Il n'est pas nécessaire d'implémenter les algorithmes de résolution des DEC-POMDPs lorsqu'on souhaite résoudre un tel problème.
- MADP Toolbox rassemble les principaux algorithmes de résolution des DEC-POMDPs :
  `https://github.com/MADPToolbox/MADP`
  `http://www.fransoliehoek.net/index.php?fuseaction=software.madp`
- Le problème de décision soit être décrit dans un fichier sous la forme d'un fichier *.dpomdp*
- Il est possible de résoudre le problème (obtention d'une politique) et de tester la qualité de cette politique (par simulation)

Conclusion

- De nombreux travaux au cours de la dernière décennie
- Approches optimales limitées à de petites tailles de problèmes
- Algorithmes approchés efficaces en pratique mais pas de borne à l'optimal
- La plupart des algorithmes planifient de manière centralisée
- Pistes actuelles : exploiter la localité des interactions
- Quelques travaux en apprentissage (réseaux de neurones) pour la résolution en ligne
- La plupart des travaux traitent le cadre coopératif

Références I

- C. Amato, Bernstein D.S., and S. Zilberstein, *Optimizing memory-bounded controllers for decentralized pomdps*, Proceedings of the Twenty Third Conference on Uncertainty in Artificial Intelligence, 2007.
- 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.
- D. Bernstein, S. Zilberstein, and N. Immerman, *The complexity of decentralized control of mdps*, Mathematics of Operations Research, 2002, pp. 27(4):819–840.
- R. Becker, S. Zilberstein, V. Lesser, and C. Goldman, *Transition-independent decentralized markov decision processes*, Proceedings of the Second International Joint Conference on Autonomous Agents and Multi Agent Systems (Melbourne, Australia), July 2003, pp. 41–48.

Références II

Arnaud Canu and Abdel-Illah Mouaddib, *Dynamic local interaction model : formalisation et algorithmes*, Journées Francophones sur les Systèmes Multi-Agents (JFSMA), 2011.

Jilles S. Dibangoye, Christopher Amato, Olivier Buffet, and Francçois Charpillet, *Exploiting separability in multiagent planning with continuous-state mdps*, Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems (Richland, SC), AAMAS '14, International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 1281–1288.

Claudia Goldman and Shlomo Zilberstein, *Decentralized control of cooperative systems: Categorization and complexity analysis*, Journal of Artificial Intelligence Research **22** (2004), 143–174.

E.A Hansen, D. Bernstein, and S. Zilberstein, *Dynamic programming for partially observable stochastic games*, Proceedings of the Nineteenth National Conference on Artificial Intelligence, 2004.

Références III

Akshat Kumar and Shlomo Zilberstein, *Anytime planning for decentralized pomdps using expectation maximization*, CoRR **abs/1203.3490** (2012).

R. Nair, V. Pradeep, T. Milind, and Y. Makoto, *Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs*, Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05), 2005.

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.

D. Szer, F. Charpillet, and S. Zilberstein, *MAA*: A heuristic search algorithm for solving decentralized POMDPs*, Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, 2005.

Références IV

S. Seuken and S. Zilberstein, *Memory-bounded dynamic programming for dec-pomdps*, Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI), 2007, pp. 2009–2015.