Stratégies de décision dans les arbres de recherche pour jeux basées sur des informations incomplètes
Application au bridge : Apprentissage statistique des enchères et jeu de la carte optimal

Sébastien Mignot

Encadrant : Michèle Sebag
Laboratoire de Recherche en Informatique
Université Paris Sud

Plan

- Motivations, position du bridge
- Les enchères :
  - rappels sur l'apprentissage
  - description des données
  - approche & résultats
- Le jeu de la carte :
  - état de l'art
  - solveur en information complète
  - bandit manchot à rationalité limité
- Limitations, défis, perspectives

Motivations

- Jeux :
  - test d'intelligence pour les humains
  - cas d'école/démonstrateur pour les machines
- Problèmes résolus :
  - jeux à faible combinatoire & information complète
  - battre l'humain aux échecs (1997)
  - explorer l'espace des dames (2008)
- Défis en cours :
  - go
  - poker
  - bridge

Précisions sur le bridge ?

Objectifs du stage : Computer Bridge

- Jeu d'enchères :
  - Spécifications molles : conventions
  - Interprétations incertaines

- Jeu de la carte :
  - Combinatoire élevée 10^38
  - Information incomplète
  - Pas de fonction d'évaluation
  - Contrainte de décision en "temps réel"

Le jeu des enchères

Initialement

- Logiciels existants :
  ~15
  Niveau initié
  Règles de décision manuelles
  wbridge5 : 7000 règles, 4000 heures

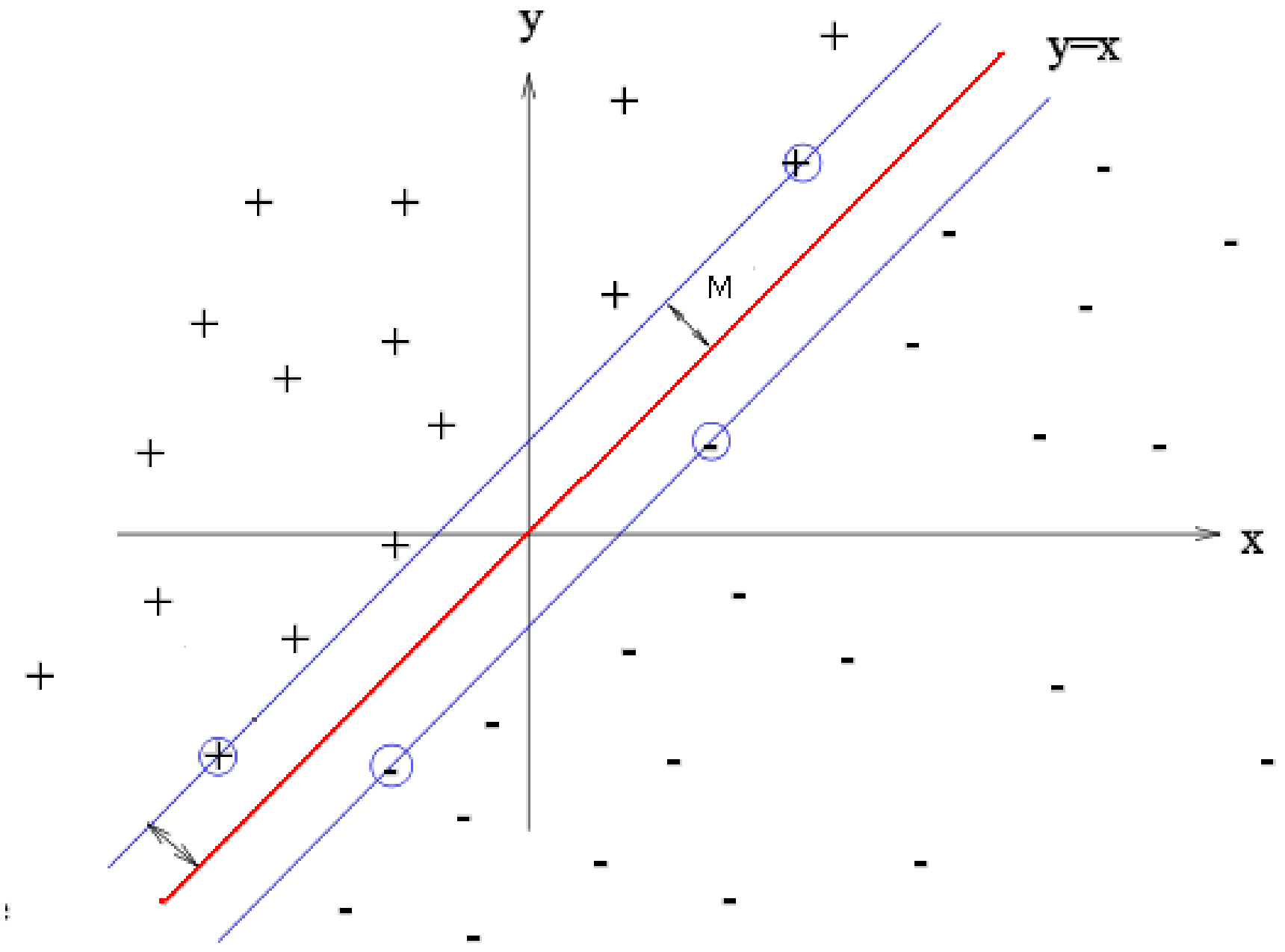
- Approche envisagée : Apprentissage statistique
  Gain espéré : complexité
  maintenance

Apprentissage statistique supervisé

- Bases d'exemples : $(x_i, y_i) \in \mathbb{R}^n \times Y$
- Y espace discret ou continu des labels
- Apprentissage : trouver la fonction $h : \mathbb{R}^n \to Y$ qui minimise l'espérance d'erreur
- SVM : cette fonction s'écrit sous la forme

$$h(x) = \sum a_i y_i < x_i, x >$$

(Vapnik, 1995 ; approche retenue)



Complexité d'apprentissage $O(sN^2)$

Complexité de classification $O(s)$

Données d'enchères disponibles

- Sources :
  - Tournois internationaux (30k)
  - Tournois nationaux (1k)
  - Server GoTo Bridge (60k)

- Caractéristiques :
  - Peu de bruit
  - hétérogénéité des conventions

Procédure

- Couper les données en trois
  - Entraînement
  - Validation
  - Test

- while (h(validation) mauvaise)
  changer la représentation ;
  h(test) ;

Jeu d'enchères par apprentissage statistique

- Y multivalué (38 classes)
- x = main, $annonce_1$,...,$annonce_n$
- y = $annonce_{n+1}$
- KPPV vs SVM
- Autres choix testés (one class svm, suite de questions : passe? contre? couleur? palier?)

Représentation proposée

- $Main \in \mathbb{N}^9$

  - Attributs primaires : nb de points d'honneur, nb de piques, ..., nb de trèfles
  - Attributs secondaires : main régulière, majeur 5ème, points de distribution, couleur 6ème && 6 à 10 points d'honneur

- $Chaque\ Annonce \in \mathbb{N}^7$

  normal/contre/surcontre, nb de plis annoncés, annonce faite à pique, ..., à trèfle, à sans-atout

- $Exemple \in \mathbb{N}^{9+7*nb\ d'annonces\ déjà\ faites}$

Validation

	Pass	1 Trèfle	1 Carreau	1 Cœur	1 Pique	1 SA	Autres
Nombre d'exemples	25377	8324	9426	5566	4133	6688	3270
Réponses Correctes	24051	7483	8549	5113	3834	5092	1839
%	94,77	89,9	90,7	91,86	92,77	76,14	56,24

Validation

	Pass	1 Trèfle	1 Carreau	1 Cœur	1 Pique	1 SA	Autres
Nombre d'exemples	23299	3	306	1325	2460	4179	31212
Réponses Correctes	21821	0	190	1194	2270	3396	24878
%	93,66	0	62,09	90,11	92,28	81,26	79,71

Le jeu de la carte

Etat de l'art

- Heuristiques pour les premières cartes
- Règles pour l'estimation des jeux cachés
- Solveur de Double Dummy (4 jeux connus) :
  Solveur(S) → int [ ] (nb plis, score)
- Jeu :
  - premières cartes : heuristique d'expertise
  - génération de situations de jeu S_i
  - moyenne des Solveur(S_i)

Limites

- Complétude & maintenance des heuristiques
- Pas de compromis exploration / exploitation
- Pas de prise en compte des impasses

Approche proposée : Bandit Bridgeur

- Se base sur les MAB :
  - Bras B_1, ..., B_n
  - recompenses v.a.
  - nb parties $n_{i,t}$
  - espérances empiriques $e_{i,t}$
  - cUCB (Auer et al. 2002) :

$$indice = Argmax\left(e_{j,t} + \sqrt{\frac{c * \log(t)}{n_{j,t}}},\ j = 1 \ldots \mathrm{K}\right)$$

Notre solveur de Double Dummy

- Implémente les algorithmes classiques (alpha-bêta pruning, lookup table)
- Approximations

- Solveur(S) → Objet o
  o.prediction(C) → int

  Conserve la table d'un appel à l'autre de cette fonction

  Pas équivalent à un bot qui résout la situation de jeu issue de S où l'on a joué C

Modélisation

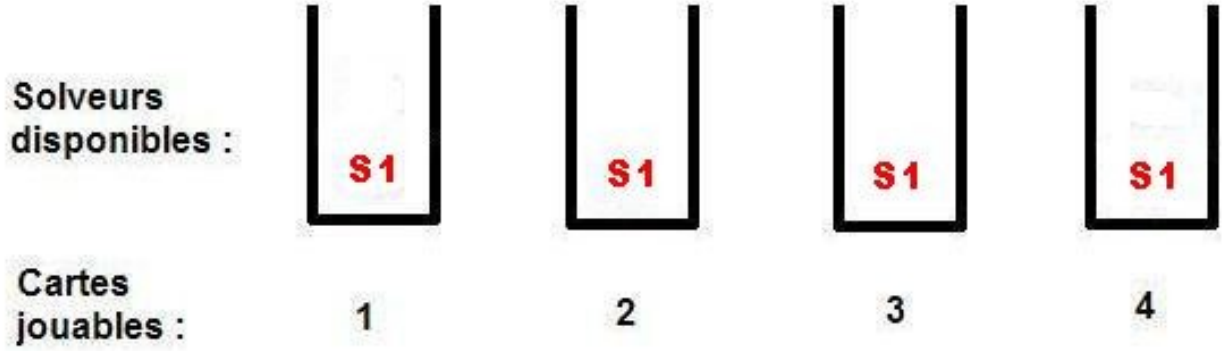
- Situation de jeu S incomplète
- n cartes $C_1,...,C_n$ possibles
- R variable aléatoire liée à la répartition des cartes dans les deux mains inconnues
- E_i l'espérance de gain du fait de jouer la carte C_i par rapport à la variable aléatoire R
- Objectif : trouver $i_0 = Argmax(E_i, i = 1...n)$

Idée générale

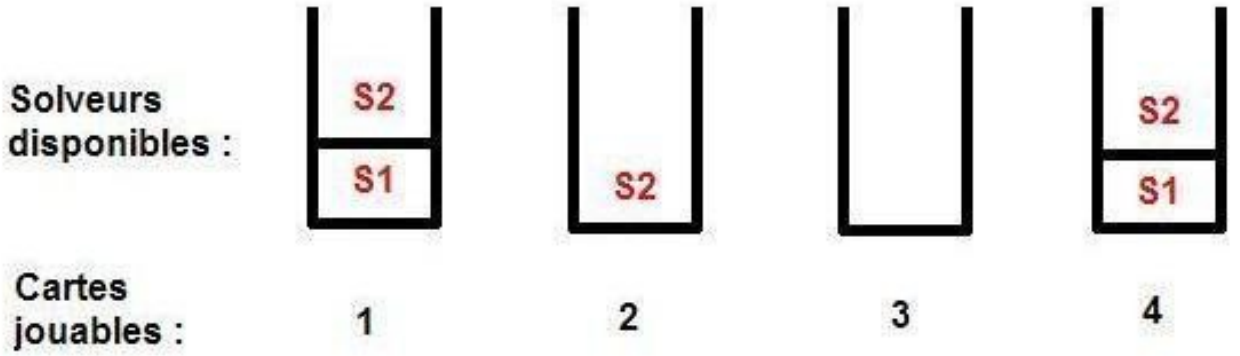
- Pour évaluer E_i :
  - Tirage aléatoire d'une valeur r pour R
  - Lancement du solveur sur la situation de jeu S_r avec la carte C_i
  - Moyennes des résultats retournés par carte
- Exploration-Exploitation :
  - Prendre quelques infos sur toutes les cartes
  - Multiplier les tirages & tests sur les meilleurs cartes pour pouvoir trouver La meilleure

Exemple

- Tirage aléatoire r de R, S_1 = Solveur(S_r)



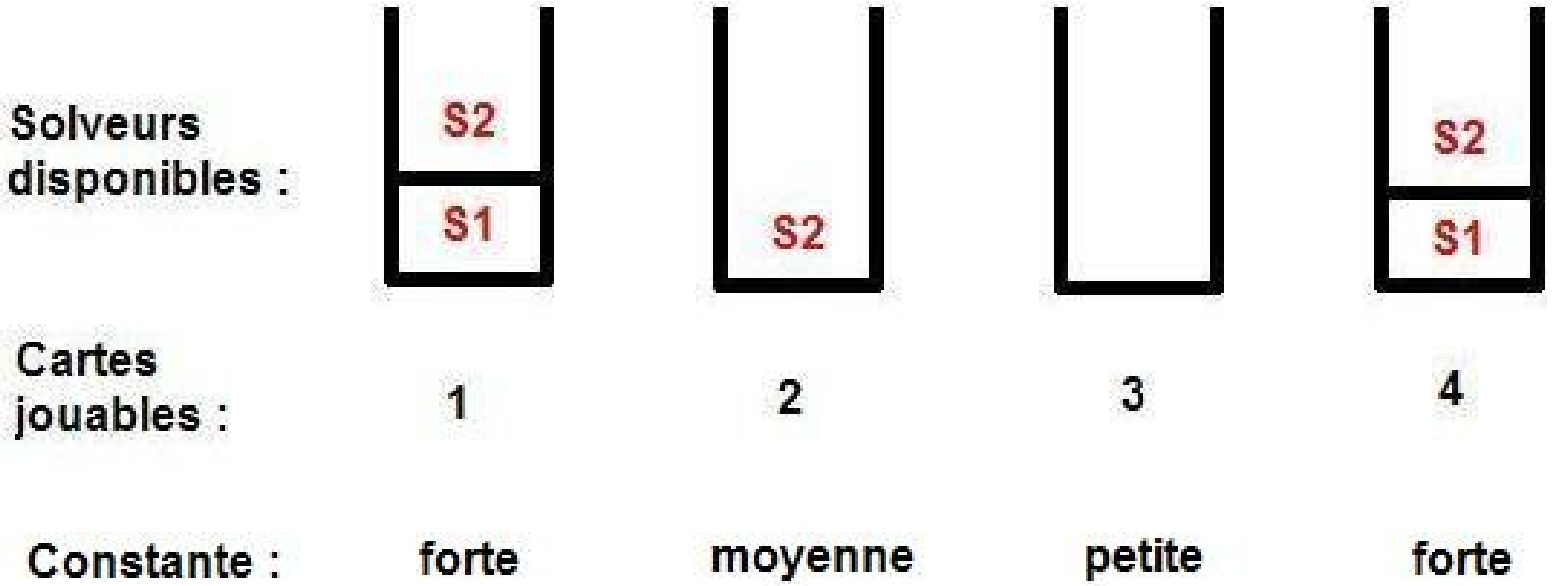
- C



Algorithme (resource-aware MAB)

- Tirages de r dès que besoin, files de solveurs
- Chaque carte vue comme un bras du bandit, avec reward = points retournés par le solveur
- Choix de la prochaine carte testée grâce à cUCB mais avec c consciente du coup en temps :

$$indice = Argmax\left(e_{j,t} + \sqrt{\frac{k_{rang(j)} * \log(t)}{n_{j,t}}},\ j = 1 \ldots \mathrm{K}\right)$$


Solveurs disponibles :
S2
S1
S2
S2
S1
Cartes jouables :
1
2
3
4
Constante :
forte
moyenne
petite
forte

Validation

- Manuelle :
  The Play of the Hand at Bridge (L.H.Watson)
  Niveau intermédiaire / 2ème série

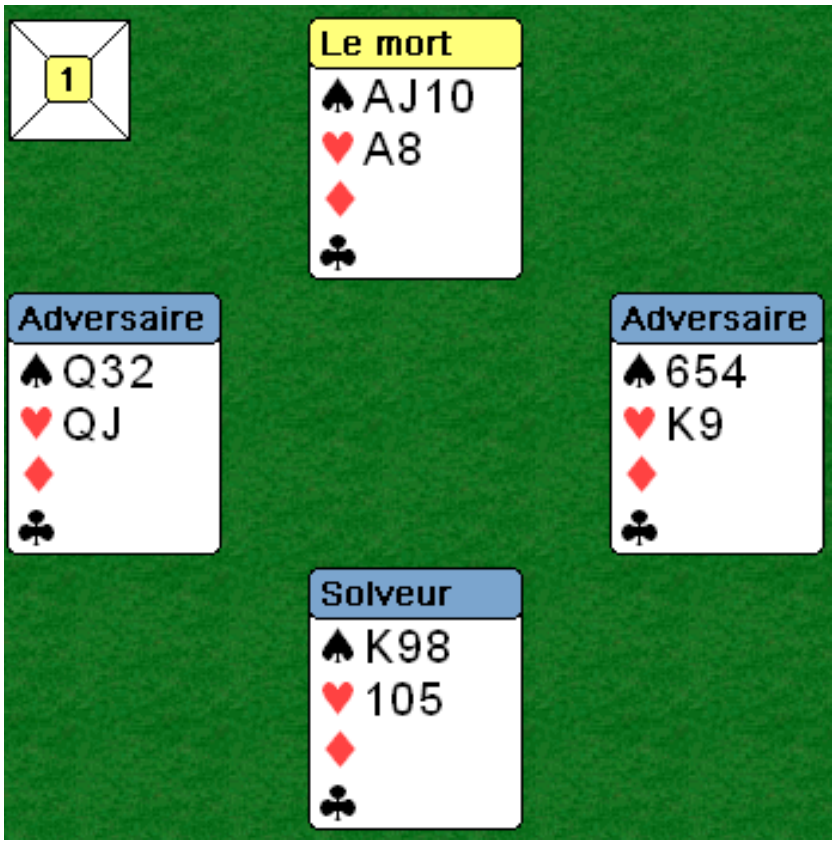
- Automatique en cours :
  Interfacer B^2 avec protocol Blue Chip Bridge

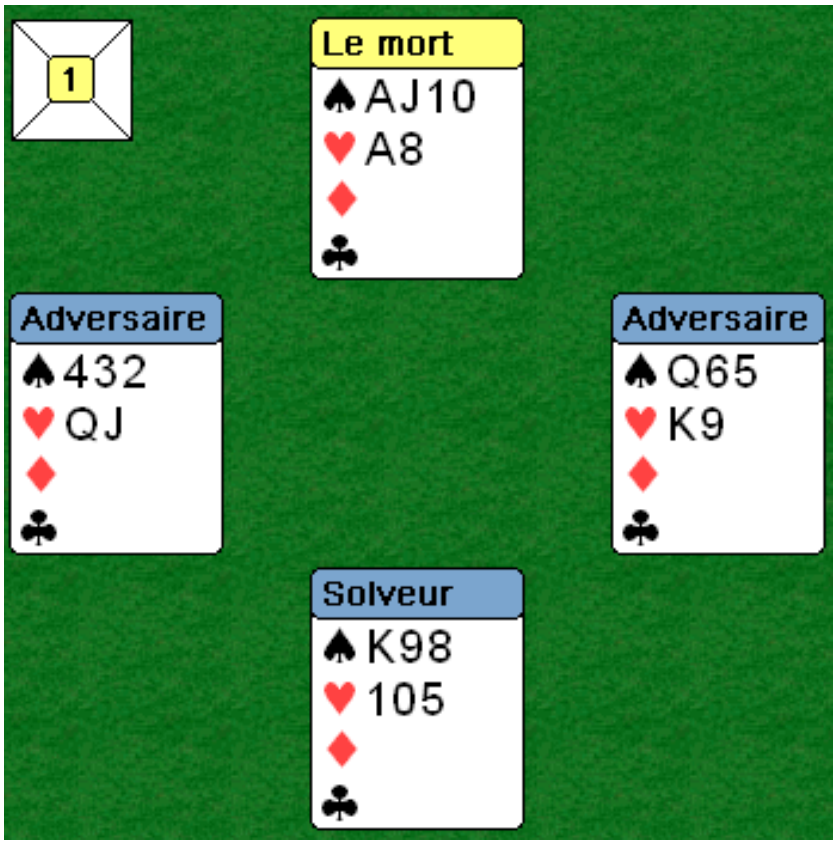
Conclusion

- Faisabilité des enchères par AA
- Premiers résultats en jeu de la carte
- Approche a permi :
  - Développement rapide
  - Code maintenable

Verrous restants

- Coups en temps sur les premières cartes :
  - Parallélisation
  - Heuristiques de remplacement :
    du MAB au many-armed bandit
- Inefficacité des heuristiques fondés sur la variance
- Absence de stratégie de prise d'information !!
  - Problème pour tous les bots de bridge
  - Solveur de double dummy "semi-idiot" ou idiot non concluant


1
Le mort
♠ AJ10
♥ A8
♦
♣
Adversaire
♠ Q32
♥ QJ
♦
♣
Adversaire
♠ 654
♥ K9
♦
♣
Solveur
♠ K98
♥ 105
♦
♣


1
Le mort
♠ AJ10
♥ A8
♦
♣
Adversaire
♠ 432
♥ QJ
♦
♣
Adversaire
♠ Q65
♥ K9
♦
♣
Solveur
♠ K98
♥ 105
♦
♣

Essais fait pour le solveur semi-idiot

- Choix aléatoire du fils
- Choix aléatoire du fils à l'ouverture
- Moyenne sur les fils
- Moyenne sur les 2 meilleurs fils
- Moyenne sur les 2 meilleurs fils à l'ouverture ou pour un seul joueur de l'équipe

Perspectives

Allocation optimale de la ressource

- Théoriser l'utilisation d'une forte équivalence au début de l'exploration
- Changer la modélisation du "reward" pour coller plus à l'intérêt du gain d'information et à son coup en temps

Gestion des alertes

- Expliciter les conventions d'enchères du bot
- Problème très général :
  E espace de recherche et Y espace des labels

  Hypothèse apprise : $h : E \to Y$

  Cherche : $(f_i)\ suite\ finie\ de\ fonction\ E_i \to Y$

  Tq $E_i \in \wp(E)\ et\ \forall (i, j), i \neq j, E_i \cap E_j = \emptyset$

$$\forall\, x \in E : proba(x \notin U\ E_i) \leqslant \epsilon_1$$

$$\forall\, i, \forall\, x \in E_i : proba(h(x) \neq f_i(x)) \leqslant \epsilon_2$$

Gestion des alertes

- Et telles que les fonctions f_i "vérifiables" par un expert humain
- Notion dure à définir
- Exemple possible : dans le cas $E = \mathbb{R}^P$
  On dit que f_i vérifiable si et seulement si
  E_i hypercube et $f_i(E_i) = \{y\}$
- Applications : certifications et tests de programmes, fouille de donnée (compréhension du phénomène & recherche de variables cachées par augmentation progressives du nombre de fonction f_i)


Questions ?