

Stratégies de Recherche de Structure par un Algorithme Génétique

A. Delaplace, T. Brouard, H. Cardot

Université François-Rabelais de Tours, Laboratoire d'informatique (EA 2101)
Département informatique de l'École Polytechnique de l'Université de Tours

Mai 2006 / JFRB - Valenciennes

Plan

- ① Contenu de l'exposé
- ② Problématique de l'Apprentissage de la Structure
 - Présentation
 - Méthodes existantes
- ③ Algorithme Développé
 - L'algorithme génétique
 - Politique de pénalisation
 - Méthode adaptative
- ④ Résultats des différentes méthodes
 - Scores obtenus
 - Structures obtenues
 - Influence des paramètres
- ⑤ Conclusions et Perspectives

Détermination de la structure d'un réseau bayésien

- Auprès d'un expert

Détermination de la structure d'un réseau bayésien

- Auprès d'un expert
- Automatique

Détermination de la structure d'un réseau bayésien

- Auprès d'un expert
- Automatique
 - (trois principales familles de méthodes)

Détermination de la structure d'un réseau bayésien

- Auprès d'un expert
- Automatique
 - (trois principales familles de méthodes)
 - Tests Statistiques

Détermination de la structure d'un réseau bayésien

- Auprès d'un expert
- Automatique
 - (trois principales familles de méthodes)
 - Tests Statistiques
 - Evaluation

Détermination de la structure d'un réseau bayésien

- Auprès d'un expert
- Automatique
 - (trois principales familles de méthodes)
 - Tests Statistiques
 - Evaluation
 - Espace des équivalents

Apprentissage par tests statistiques

- Construire un Graphe Dirigé Acyclique (DAG) par une suite de tests statistiques d'ordre croissant

Apprentissage par tests statistiques

- Construire un Graphe Dirigé Acyclique (DAG) par une suite de tests statistiques d'ordre croissant
- Orienter la structure obtenue

Apprentissage par tests statistiques

- Construire un Graphe Dirigé Acyclique (DAG) par une suite de tests statistiques d'ordre croissant
- Orienter la structure obtenue

Avantages :

- Méthode intuitive

Inconvénients :

Apprentissage par tests statistiques

- Construire un Graphe Dirigé Acyclique (DAG) par une suite de tests statistiques d'ordre croissant
- Orienter la structure obtenue

Avantages :

- Méthode intuitive
- Construction graduelle de la structure

Inconvénients :

Apprentissage par tests statistiques

- Construire un Graphe Dirigé Acyclique (DAG) par une suite de tests statistiques d'ordre croissant
- Orienter la structure obtenue

Avantages :

- Méthode intuitive
- Construction graduelle de la structure

Inconvénients :

- Multiplicité des tests

Apprentissage par tests statistiques

- Construire un Graphe Dirigé Acyclique (DAG) par une suite de tests statistiques d'ordre croissant
- Orienter la structure obtenue

Avantages :

- Méthode intuitive
- Construction graduelle de la structure

Inconvénients :

- Multiplicité des tests
- Problème de fiabilité des tests

Apprentissage par attribution d'un score

- Attribuer un score évaluant la concordance des (in)dépendances représentées par la structure avec les observations

Apprentissage par attribution d'un score

- Attribuer un score évaluant la concordance des (in)dépendances représentées par la structure avec les observations
- Rechercher la structure optimisant le score

Apprentissage par attribution d'un score

- Attribuer un score évaluant la concordance des (in)dépendances représentées par la structure avec les observations
- Rechercher la structure optimisant le score

Avantages :

- Possibilité de compromis entre plusieurs solutions

Inconvénients :

Apprentissage par attribution d'un score

- Attribuer un score évaluant la concordance des (in)dépendances représentées par la structure avec les observations
- Rechercher la structure optimisant le score

Avantages :

- Possibilité de compromis entre plusieurs solutions

Inconvénients :

- Taille de l'espace de recherche ($N^{2^{\mathcal{O}(N)}}$)

Recherche dans l'espace des équivalents

Deux graphes sont dits équivalents s'ils représentent la même probabilité jointe

- On parcourt directement l'espace des équivalents

Recherche dans l'espace des équivalents

Deux graphes sont dits équivalents s'ils représentent la même probabilité jointe

- On parcourt directement l'espace des équivalents

Avantages :

- Parcours simplifié : deux DAGs équivalents = 1 classe d'équivalence

Inconvénients :

Recherche dans l'espace des équivalents

Deux graphes sont dits équivalents s'ils représentent la même probabilité jointe

- On parcourt directement l'espace des équivalents

Avantages :

- Parcours simplifié : deux DAGs équivalents = 1 classe d'équivalence

Inconvénients :

- Espace faiblement restreint (Perlman & Gillispie 2002)

Recherche dans l'espace des équivalents

Deux graphes sont dits équivalents s'ils représentent la même probabilité jointe

- On parcourt directement l'espace des équivalents

Avantages :

- Parcours simplifié : deux DAGs équivalents = 1 classe d'équivalence

Inconvénients :

- Espace faiblement restreint (Perlman & Gillispie 2002)
- Méthodes encore lourdes

Solution Proposée

- Objectif : trouver une structure optimale, sans connaissance a priori (ordre, fan-in)

Solution Proposée

- Objectif : trouver une structure optimale, sans connaissance a priori (ordre, fan-in)
- Emploi d'une méthode stochastique, pour un problème combinatoirement difficile

Solution Proposée

- Objectif : trouver une structure optimale, sans connaissance a priori (ordre, fan-in)
- Emploi d'une méthode stochastique, pour un problème combinatoirement difficile
- Un algorithme génétique

Algorithme Génétique

Le schéma général d'un algorithme évolutionnaire :

```
INITIALISER une population P0
CALCULER FITNESS pour chaque individu
TantQue pas CRITERE D'ARRET
{
  SELECTIONNER PARENTS
  RECOMBINER les parents
  MUTER la descendance
  CALCULER FITNESS pour les nouveaux individus
  REMPLACER certains parents par certains descendants
}
```

Algorithme Génétique

- Représentation : un individu \Leftrightarrow un DAG (matrice d'adjacence)

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Algorithme Génétique

- Représentation : un individu \Leftrightarrow un DAG (matrice d'adjacence)

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- Mutation : Sur les ensembles de parents $Pa(X_i), i \in 1 \dots N$ (colonnes) : ajout, soustraction ou inversion d'un arc

Algorithme Génétique : fitness

Score BDeu (Heckerman. et al., 1995)

$$S_{BD}(\mathcal{G}|D) = P(\mathcal{G}|D) \propto P(\mathcal{G}) \prod_{i=1}^N \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

- α_{ijk} : paramètres d'une distribution de Dirichlet

Algorithme Génétique : fitness

Score BDeu (Heckerman. et al., 1995)

$$S_{BD}(\mathcal{G}|D) = P(\mathcal{G}|D) \propto P(\mathcal{G}) \prod_{i=1}^N \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

- α_{ijk} : paramètres d'une distribution de Dirichlet
- r_i et q_i : nombre de valeurs pouvant être prises par X_i et $Pa(X_i)$

Algorithme Génétique : fitness

Score BDeu (Heckerman. et al., 1995)

$$S_{BD}(\mathcal{G}|D) = P(\mathcal{G}|D) \propto P(\mathcal{G}) \prod_{i=1}^N \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

- α_{ijk} : paramètres d'une distribution de Dirichlet
- r_i et q_i : nombre de valeurs pouvant être prises par X_i et $Pa(X_i)$
- N_{ijk} nombre d'occurrences simultanées de $X_i = x_k$ et $Pa(X_i) = Pa(X_i)_j$

Algorithme Génétique : fitness

Score BDeu (Heckerman. et al., 1995)

$$S_{BD}(\mathcal{G}|D) = P(\mathcal{G}|D) \propto P(\mathcal{G}) \prod_{i=1}^N \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

- α_{ijk} : paramètres d'une distribution de Dirichlet
- r_i et q_i : nombre de valeurs pouvant être prises par X_i et $Pa(X_i)$
- N_{ijk} nombre d'occurrences simultanées de $X_i = x_k$ et $Pa(X_i) = Pa(X_i)_j$

Score décomposable, équivalent et consistant

Algorithme Génétique : Opérateur de réparation

- Si un descendant n'est pas un DAG

Algorithme Génétique : Opérateur de réparation

- Si un descendant n'est pas un DAG
- Calcul de l'information mutuelle entre les couples de variables X_A et X_B :

$$W(X_A, X_B) = \sum_{x_a, x_b} \frac{N_{ab}}{N} \log \frac{N_{ab}N}{N_a N_b}$$

Algorithme Génétique : Opérateur de réparation

- Si un descendant n'est pas un DAG
- Calcul de l'information mutuelle entre les couples de variables X_A et X_B :

$$W(X_A, X_B) = \sum_{x_a, x_b} \frac{N_{ab}}{N} \log \frac{N_{ab}N}{N_a N_b}$$

- L'arc portant la plus faible information mutuelle est supprimé

Principe général

Optimiser la recherche dans l'espace des graphes dirigés acycliques :

- Deux méthodes :

Principe général

Optimiser la recherche dans l'espace des graphes dirigés acycliques :

- Deux méthodes :
 - Une méthode de pénalisation des optima locaux

Principe général

Optimiser la recherche dans l'espace des graphes dirigés acycliques :

- Deux méthodes :
 - Une méthode de pénalisation des optima locaux
 - Approche auto-adaptative du taux de mutation

Pénalisation

- Au terme d'un nombre prédéterminé d'itérations sans amélioration, la classe d'équivalence du meilleur individu est stockée

Pénalisation

- Au terme d'un nombre prédéterminé d'itérations sans amélioration, la classe d'équivalence du meilleur individu est stockée
- Le fitness de tout individu, présent et à venir, appartenant cette classe est pénalisé (mis à une valeur très faible)

Pénalisation

- Au terme d'un nombre prédéterminé d'itérations sans amélioration, la classe d'équivalence du meilleur individu est stockée
- Le fitness de tout individu, présent et à venir, appartenant cette classe est pénalisé (mis à une valeur très faible)
- A terme, le meilleur optimum local rencontré est renvoyé

Pénalisation

- Au terme d'un nombre prédéterminé d'itérations sans amélioration, la classe d'équivalence du meilleur individu est stockée
- Le fitness de tout individu, présent et à venir, appartenant cette classe est pénalisé (mis à une valeur très faible)
- A terme, le meilleur optimum local rencontré est renvoyé
- Différent d'une méthode Tabou : individus pénalisés mais non interdits

Approche Dynamique

Principe :

- Paramètres statiques \rightarrow suboptimalité

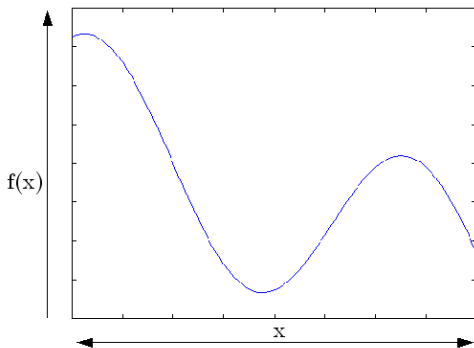
Approche Dynamique

Principe :

- Paramètres statiques \rightarrow suboptimalité
- Paramètres dynamiques \rightarrow adaptation au problème

Approche Dynamique

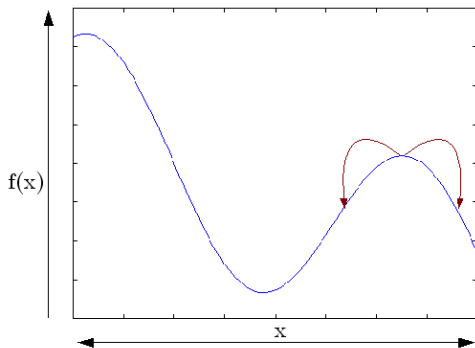
Problème :



Population située autour d'un optimum local

Approche Dynamique

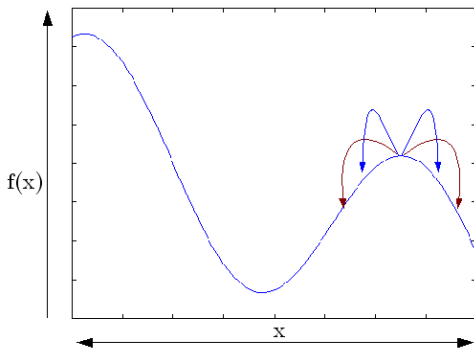
Problème :



Mutation statique : si le taux de mutation est insuffisant

Approche Dynamique

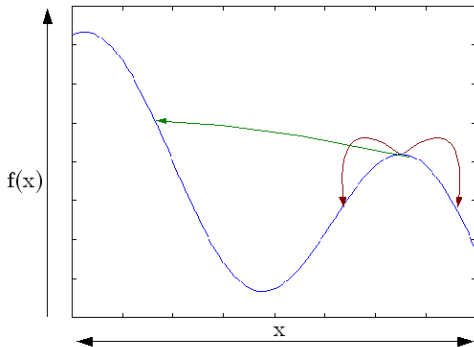
Problème :



Mutation dynamique : si le taux de mutation converge

Approche Dynamique

Objectif :



Permettre une augmentation contrôlée du taux de mutation

Approche Dynamique

Implémentation :

Stratégie de mutation adaptative

A chaque cycle de mutation, pour un individu I ayant un taux de mutation P_m :

Approche Dynamique

Implémentation :

Stratégie de mutation adaptative

A chaque cycle de mutation, pour un individu I ayant un taux de mutation P_m :

- Muter l'individu I suivant P_m : $(I, P_m) \rightarrow (I', P'_m)$

Approche Dynamique

Implémentation :

Stratégie de mutation adaptative

A chaque cycle de mutation, pour un individu I ayant un taux de mutation P_m :

- Muter l'individu I suivant $P_m : (I, P_m) \rightarrow (I', P'_m)$
- Si $f(I') > f(I) : P'_m \leftarrow \gamma \times P_m, P_m \leftarrow \omega \times P_m$

$$\gamma < 1, \omega > 1$$

Approche Dynamique

Implémentation :

Stratégie de mutation adaptative

A chaque cycle de mutation, pour un individu I ayant un taux de mutation P_m :

- Muter l'individu I suivant P_m : $(I, P_m) \rightarrow (I', P'_m)$
- Si $f(I') > f(I)$: $P'_m \leftarrow \gamma \times P_m$, $P_m \leftarrow \omega \times P_m$
- Si $f(I') \leq f(I)$: $P'_m \leftarrow \omega \times P_m$, $P_m \leftarrow \gamma \times P_m$

$$\gamma < 1, \omega > 1$$

Approche Dynamique

Implémentation :

Stratégie de mutation adaptative

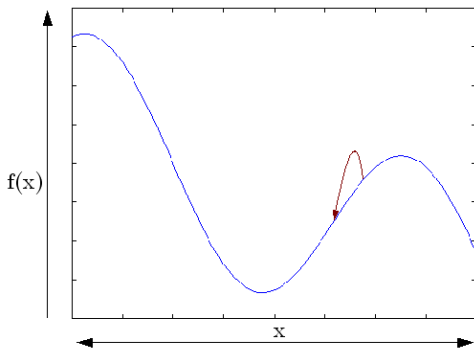
A chaque cycle de mutation, pour un individu I ayant un taux de mutation P_m :

- Muter l'individu I suivant P_m : $(I, P_m) \rightarrow (I', P'_m)$
- Si $f(I') > f(I)$: $P'_m \leftarrow \gamma \times P_m$, $P_m \leftarrow \omega \times P_m$
- Si $f(I') \leq f(I)$: $P'_m \leftarrow \omega \times P_m$, $P_m \leftarrow \gamma \times P_m$

$$\gamma < 1, \omega > 1 \quad \frac{1}{N} \leq P_m \leq \frac{4}{N}$$

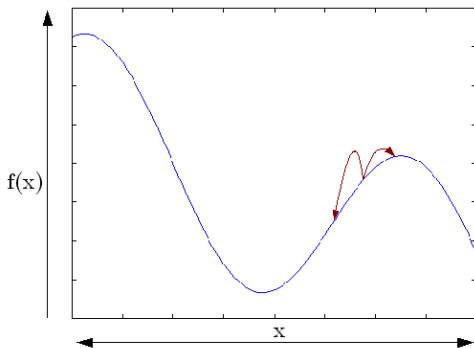
- Augmenter les chances des moins bons individus et affiner la recherche locale des meilleurs

Approche Dynamique



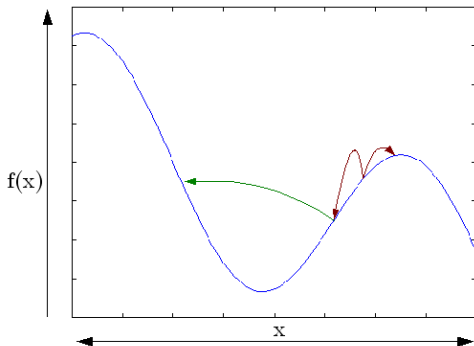
Mutation = dégradation

Approche Dynamique



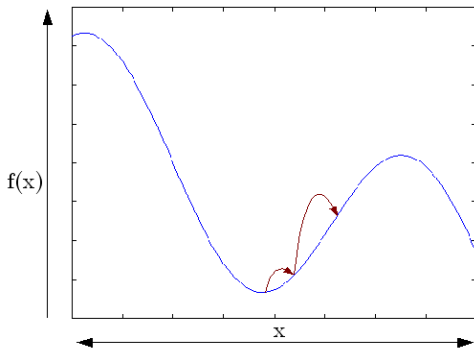
Mutation = dégradation \rightarrow diminuer le taux P_m du muté

Approche Dynamique



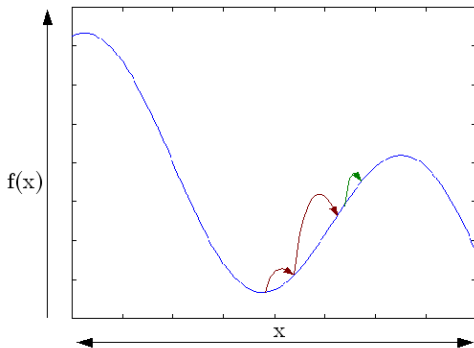
Mutation = dégradation \rightarrow diminuer le taux P_m du muté
 \rightarrow augmenter le taux P_m du mutant

Approche Dynamique



Mutation = amélioration \rightarrow augmenter le taux P_m du muté

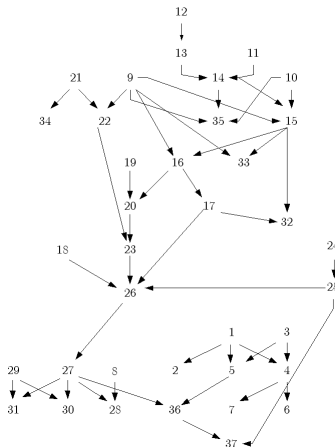
Approche Dynamique



Mutation = dégradation \rightarrow augmenter le taux P_m du muté
 \rightarrow diminuer le taux P_m du mutant

Tests

Structure recherchée : réseau ALARM



Tests

Conditions :

- Critère d'arrêt : 2500 itérations

Tests

Conditions :

- Critère d'arrêt : 2500 itérations
- Ordre des variables inconnus

Tests

Conditions :

- Critère d'arrêt : 2500 itérations
- Ordre des variables inconnus
- Taux de mutation statique : $P_m = \frac{1}{N}$

Tests

Conditions :

- Critère d'arrêt : 2500 itérations
- Ordre des variables inconnus
- Taux de mutation statique : $P_m = \frac{1}{N}$
- Population de 50 individus, initialisée à 0

Tests

Conditions :

- Critère d'arrêt : 2500 itérations
- Ordre des variables inconnus
- Taux de mutation statique : $P_m = \frac{1}{N}$
- Population de 50 individus, initialisée à 0
- Résultats moyennés sur 10 instances pour l'AG

Tests

Conditions :

- Critère d'arrêt : 2500 itérations
- Ordre des variables inconnus
- Taux de mutation statique : $P_m = \frac{1}{N}$
- Population de 50 individus, initialisée à 0
- Résultats moyennés sur 10 instances pour l'AG
- Résultats pour AG, AG-P et AG-D, respectivement

Réseau ALARM : scores

	ALARM		
	1000	5000	10000
Original	-11492	-56504	-111279
AG Simple	-11536 ± 65	-56738 ± 217	-111526 ± 244
AG-P	-11506 ± 35*	-56550 ± 93*	-111385 ± 135*
AG-D (1.1)	-11525 ± 29	-56607 ± 128	-111511 ± 154
AG-D (1.2)	-11511 ± 60	-56713 ± 175	-111496 ± 167
AG-D (1.3)	-11543 ± 52	-56693 ± 191	-111449 ± 181
Glouton	-11659	-56848	-111932

Réseau ALARM : structures

	ALARM		
	1000	5000	10000
AG Simple	24.1(12)	22.5(9)	21.3(13)
AG-P	24.3(11)	14.9(5)*	13.3(4)*
AG-D (1.1)	23.5(16)*	20(10)	19(12)
AG-D (1.2)	23.5(11)*	17.1(5)	18.3(7)
AG-D (1.3)	25.1(12)	20.8(12)	15.8(7)
Glouton	37	34	33

Réseau ALARM : scores

Meilleures performances

	ALARM		
	1000	5000	10000
Original	-11492	-56504	-111279
AG Simple	-11536	-56578	-111321
AG-P	-11458*	-56450*	-111219*
AG-D (1.1)	-11497	-56466	-111330
AG-D (1.2)	-11439	-56472	-111223
AG-D (1.3)	-11460	-56495	-111223

Influence de ω

- Base limitée/peu représentative : ω faible

Influence de ω

- Base limitée/peu représentative : ω faible
- Base représentative : plus grande diversité préférable

Conclusion

- Les deux procédés améliorent les performances de l'AG

Conclusion

- Les deux procédés améliorent les performances de l'AG
- Pénalisation : plus performante

Conclusion

- Les deux procédés améliorent les performances de l'AG
- Pénalisation : plus performante
- Méthode adaptative : plus "naturelle"

Conclusion

- Les deux procédés améliorent les performances de l'AG
- Pénalisation : plus performante
- Méthode adaptative : plus "naturelle"
- Déplacement du problème : encore des paramètres ?

Conclusion

- Les deux procédés améliorent les performances de l'AG
- Pénalisation : plus performante
- Méthode adaptative : plus "naturelle"
- Déplacement du problème : encore des paramètres ?

- Choix de la variabilité

Perspectives

Poursuivre l'étude des paramètres

- Nécessité d'une meilleure appréhension de l'espace

Perspectives

Poursuivre l'étude des paramètres

- Nécessité d'une meilleure appréhension de l'espace
- Influence de la recombinaison

Perspectives

Poursuivre l'étude des paramètres

- Nécessité d'une meilleure appréhension de l'espace
- Influence de la recombinaison
- Non pas vers un algorithme "boîte noire" mais vers une meilleure compréhension du problème

Merci de votre attention
{alain.delaplace, thierry.brouard, hubert.cardot}@univ-tours.fr