

Diagnostic à la demande

sur un système à événements discrets

20 septembre 2011

Projet de fin d'études



Groupe DISCO



Diagnostic anytime

1 Présentation du sujet

2 Contributions

3 Expérimentations

Diagnostic anytime

1

Présentation du sujet

2

Contributions

3

Expérimentations

Diagnostic anytime

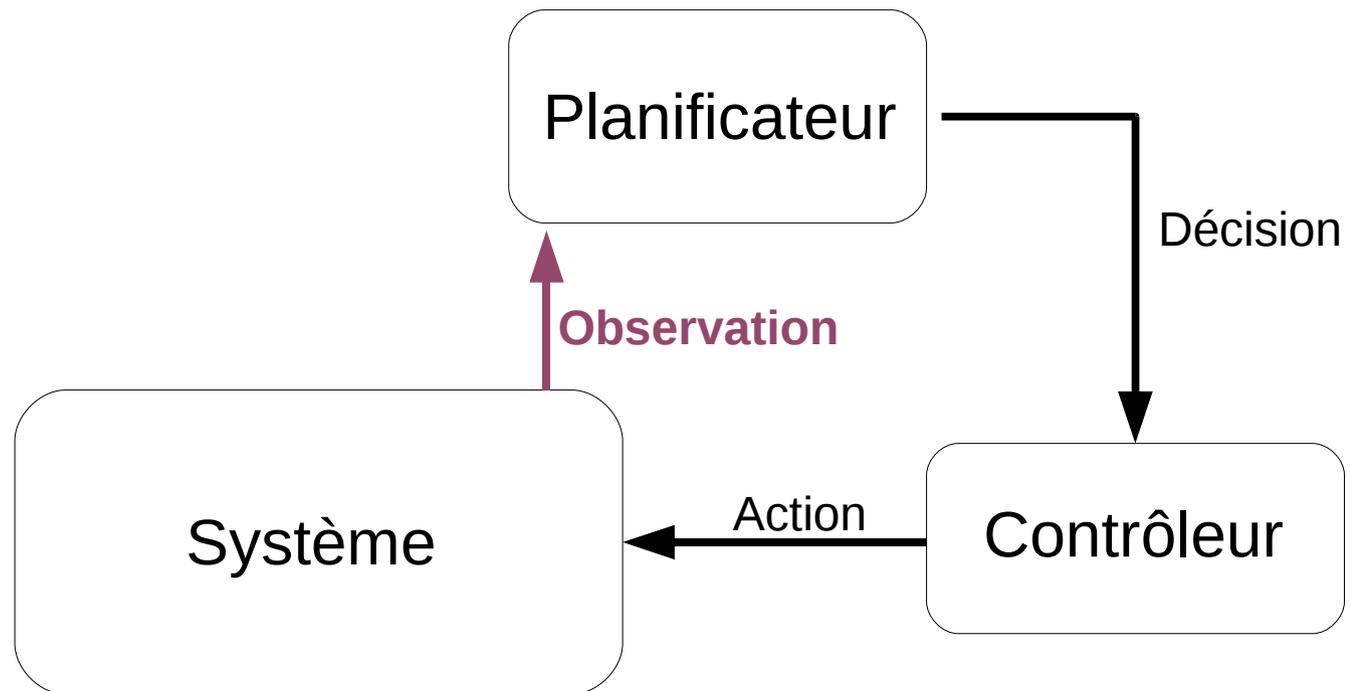
- 1** Présentation du sujet
- 2** Contributions
- 3** Expérimentations

Présentation du sujet

1

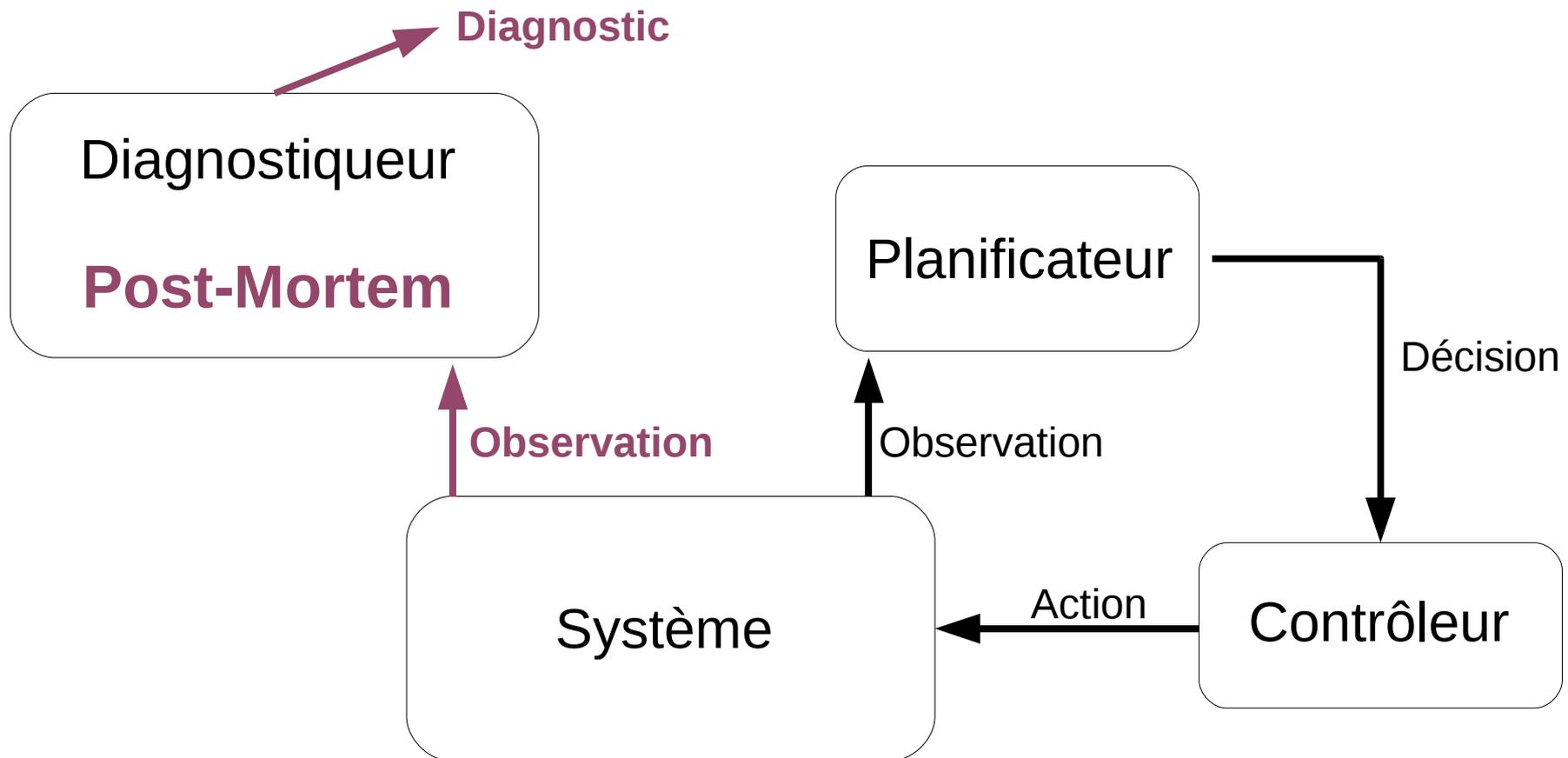
Situation du problème

- Système à événements discrets
 - contrôle à partir des événements observables



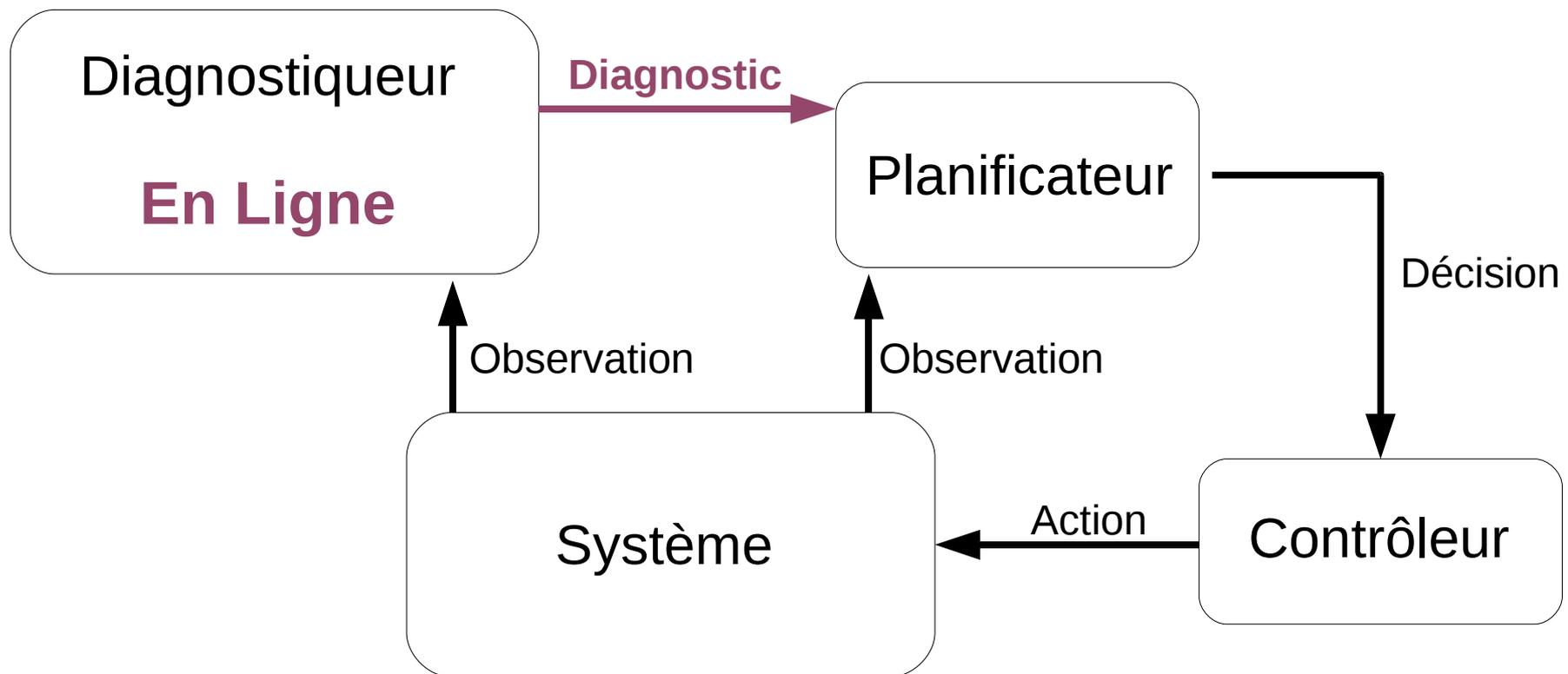
Situation du problème

- Système à événements discrets
 - contrôle à partir des événements observables
- Diagnostic
 - détection, isolation et identification des fautes



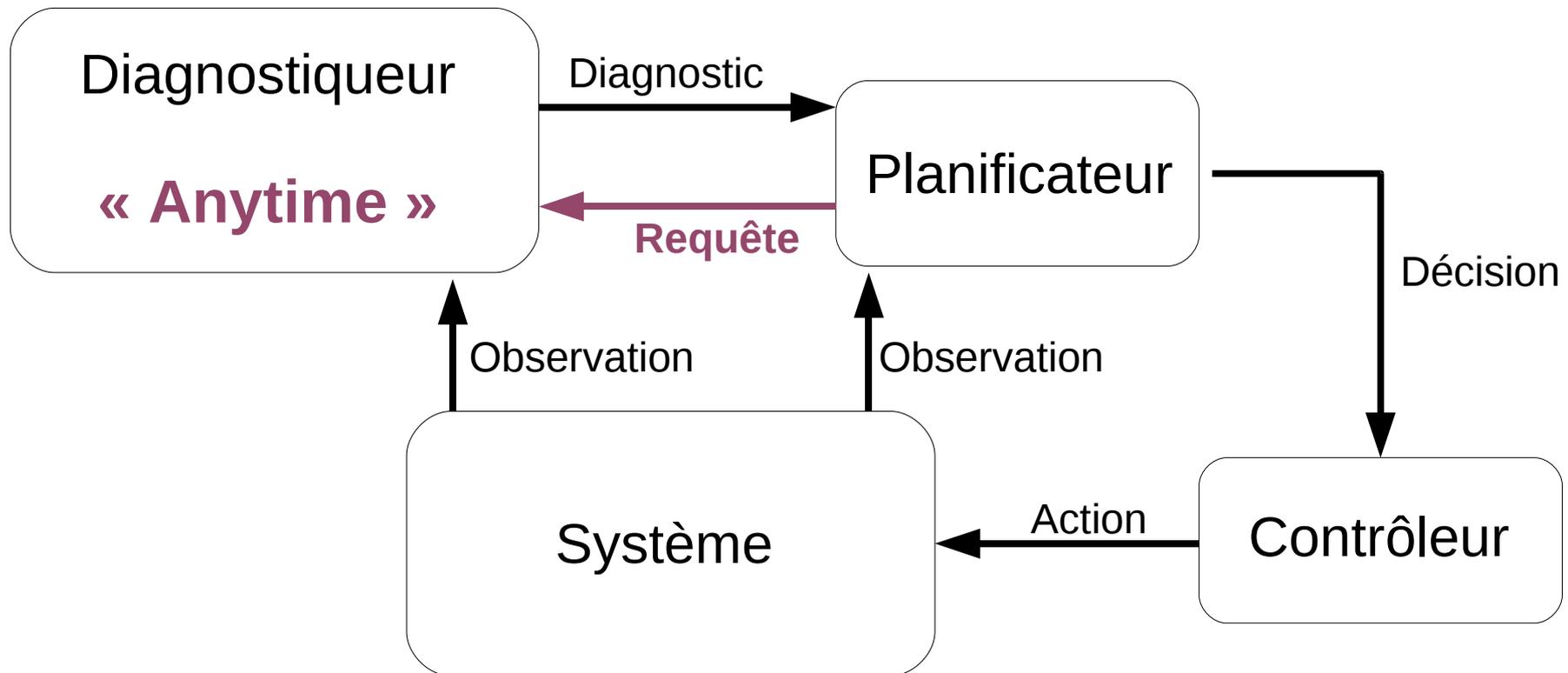
Situation du problème

- Système à événements discrets
 - contrôle à partir des événements observables
- Diagnostic
 - détection, isolation et identification des fautes



Situation du problème

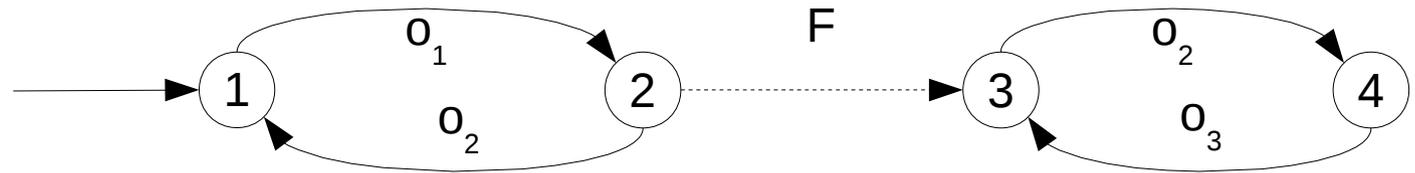
- Système à événements discrets
 - contrôle à partir des événements observables
- Diagnostic
 - détection, isolation et identification des fautes



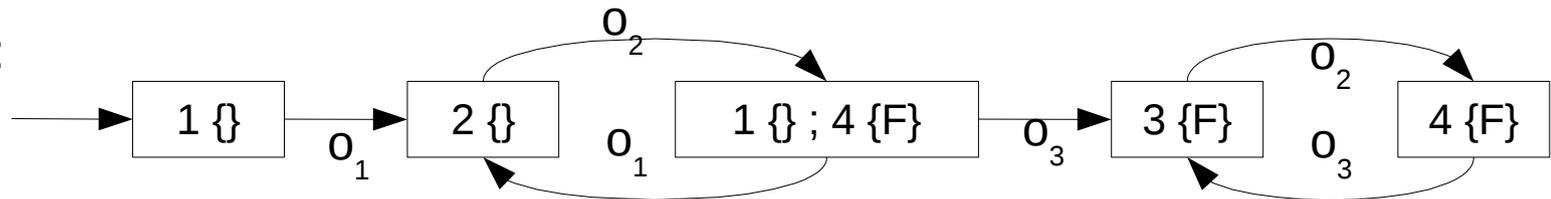
Diagnostic en ligne

Diagnosticateur de Sampath [Sampath, 96]

- Automate du modèle global :



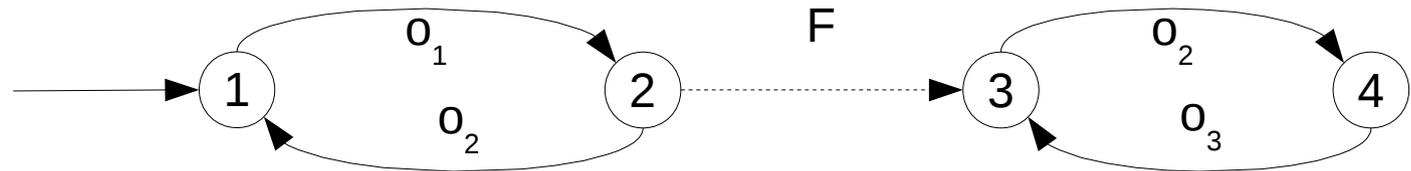
- Automate du diagnostiqueur :



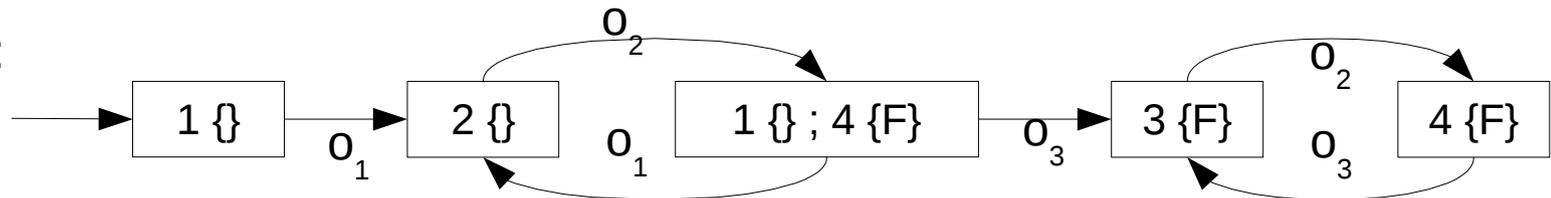
Diagnostic en ligne

Diagnosticheur de Sampath [Sampath, 96]

- Automate du modèle global :



- Automate du diagnostiqueur :



- Résultats pour la séquence d'observations (o1, o2, o3) :

Observation	Etat de croyance	Diagnostic
Etat initial	1 { }	Sain
o1	2 { }	Sain
o2	1 { } ; 4 { F }	Ambigu
o3	3 { F }	Fautif

Justification de l'Anytime

Diagnostic en ligne

- A partir du modèle complet du diagnostiqueur de Sampath
 - **Temps de réponse constant**
 - **Complexité spatiale** (non embarquable)
- A partir d'un modèle moins pré-compilé
 - **Complexité temporelle**

Justification de l'Anytime

Diagnostic en ligne

- A partir du modèle complet du diagnostiqueur de Sampath
 - **Temps de réponse constant**
 - **Complexité spatiale** (non embarquable)
- A partir d'un modèle moins pré-compilé
 - **Complexité temporelle**

Diagnostic Anytime

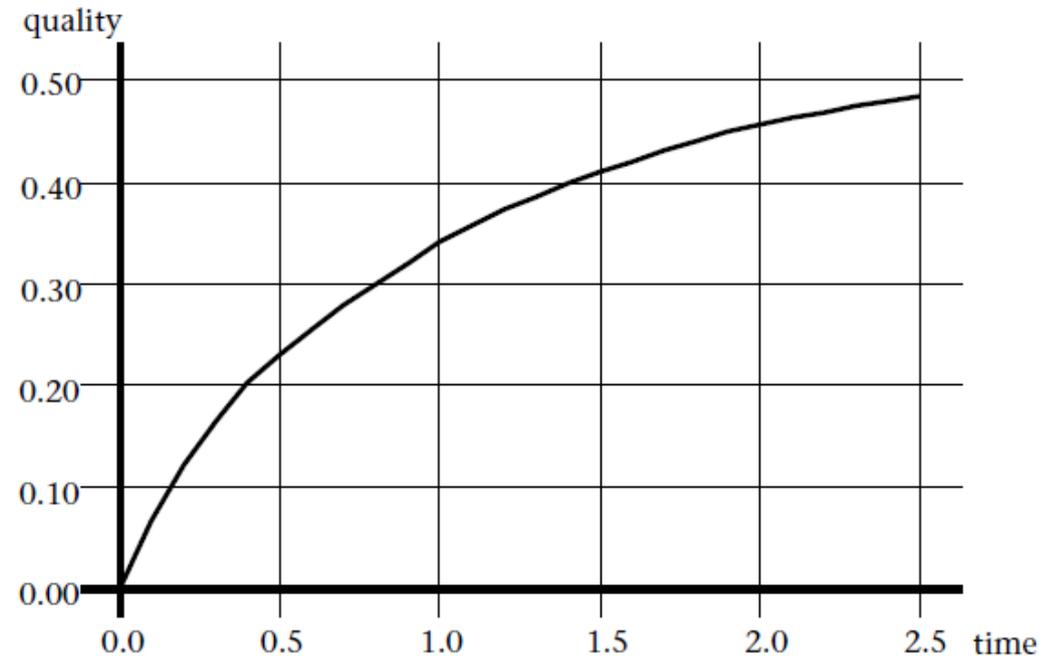
- Algorithme **interruptible** à tout instant
 - Temps de calcul trop long
 - Requête du planificateur
- Résultat tout de même « **correct** »

Algorithme Anytime

- **Algorithme classique**
 - Qualité binaire

Algorithme Anytime

- **Algorithme classique**
 - Qualité binaire
- **Algorithme anytime**
 - Qualité des résultats **augmente avec le temps de calcul** [Zilberstein, 96]
- **Nouvelles notions de qualité**
 - Certitude
 - Précision : planification
 - **Spécificité** : diagnostic



Contributions

2

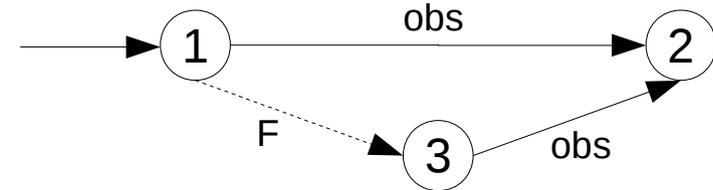


Diagnostic Anytime

- Les résultats intermédiaires doivent être **corrects**
 - Toujours **englober le résultat final**
 - Départ : **états cibles de l'observation** diagnostiquée
 - Associe chaque état à toutes les combinaisons de fautes possibles

Diagnostic Anytime

- Les résultats intermédiaires doivent être **corrects**
 - Toujours **englober le résultat final**
 - Départ : **états cibles de l'observation** diagnostiquée
 - Associe chaque état à toutes les combinaisons de fautes possibles
- Amélioration de la qualité du diagnostic
 - **Confirmation** d'une solution
 - **Raffinage** d'un état



Etat visité	Etats à visiter	Etat de croyance	
		Non-anytime	Anytime
1 { }	2 { } ; 3 {F}		2 { <i>non-raffiné</i> }
2 { }	3 {F}	2 { }	2 { <i>non-raffiné</i> } ; 2 { }
3 {F}	2 {F}	2 { }	2 { <i>non-raffiné</i> } ; 2 { }
2 {F}		2 { } ; 2 {F}	2 { } ; 2 {F}

Algorithmes

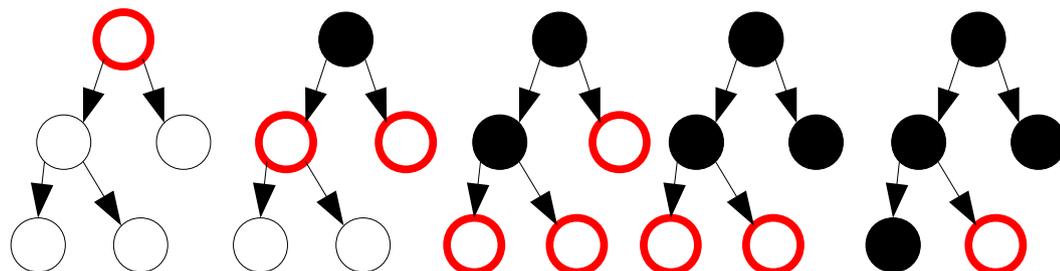
- Utilisation des bibliothèques C++ de **DIADES**
 - Départ : algorithme de base (non-anytime) : 300 lignes
 - Algorithme anytime final : 2500 lignes

Algorithmes

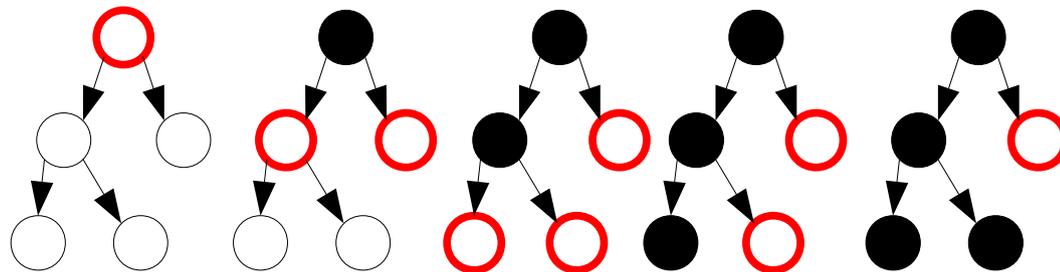
- Utilisation des bibliothèques C++ de **DIADES**
 - Départ : algorithme de base (non-anytime) : 300 lignes
 - Algorithme anytime final : 2500 lignes

- Recherche d'arbre [Russel]

→ Largeur d'abord (BFS) :



→ Profondeur d'abord (DFS) :



○ Etat à visiter ; ● Etat visité

Algorithmes (2)

Processus

Algorithmes (2)

Processus

- Recherche **en arrière**, en profondeur d'abord
 - En arrière : pour le **raffinage** des états de départ
 - En profondeur : pour raffiner les états un par un
- Recherche **en avant**, en largeur d'abord
 - En avant : **confirmation rapide** de solutions
 - En largeur : rapprocher les solutions pour la recherche en arrière

Algorithmes (2)

Processus

- Recherche **en arrière**, en profondeur d'abord
 - En arrière : pour le **raffinage** des états de départ
 - En profondeur : pour raffiner les états un par un
- Recherche **en avant**, en largeur d'abord
 - En avant : **confirmation rapide** de solutions
 - En largeur : rapprocher les solutions pour la recherche en arrière
- **Fonction de raffinage** des anciennes étapes
 - Recherche en avant : **abstractions transmises**
 - Recherche en arrière : **abstractions créées**

Avantages :

- Si il y a une longue pause entre deux observations
- Pour les informations utiles mais pas pressées

Algorithmes (2)

Processus

- Recherche **en arrière**, en profondeur d'abord
 - En arrière : pour le **raffinage** des états de départ
 - En profondeur : pour raffiner les états un par un
 - Recherche **en avant**, en largeur d'abord
 - En avant : **confirmation rapide** de solutions
 - En largeur : rapprocher les solutions pour la recherche en arrière
 - **Fonction de raffinage** des anciennes étapes
 - Recherche en avant : **abstractions transmises**
 - Recherche en arrière : **abstractions créées**
- Avantages :
- Si il y a une longue pause entre deux observations
 - Pour les informations utiles mais pas pressées

Sélection des processus => **6 versions** possibles

Expérimentations

3



Critères de qualité

Les critères

- Critère de raffinage
 - Nombre de **solutions couvertes** par le résultat intermédiaire
- Critère de confirmation
 - Nombre de **solutions confirmées**
- Critère de faute
 - **Statut des fautes** (sain, sûr, ambigu, ...)

Critères de qualité

Les critères

- Critère de raffinage
 - Nombre de **solutions couvertes** par le résultat intermédiaire
- Critère de confirmation
 - Nombre de **solutions confirmées**
- Critère de faute
 - **Statut des fautes** (sain, sûr, ambigu, ...)

Les mesures

- Comparaison à **temps fixé par l'algorithme de base** (non-anytime)
 - Interruption et mesure des critères
- Comparaison avec un **seuil de qualité**
 - Mesure du temps quand tous les critères ont dépassé le seuil

Expérimentations

Plan des expérimentations

- **Génération d'une série** de modèles
 - Uniquement un paramètre varie
- **Analyse des résultats** des deux programmes de mesure
 - Limitations, intervalle de fonctionnement correct
- **Fixe la valeur** du paramètre
 - Au milieu de l'intervalle de fonctionnement correct

Expérimentations

Plan des expérimentations

- **Génération d'une série** de modèles
 - Uniquement un paramètre varie
- **Analyse des résultats** des deux programmes de mesure
 - Limitations, intervalle de fonctionnement correct
- **Fixe la valeur** du paramètre
 - Au milieu de l'intervalle de fonctionnement correct

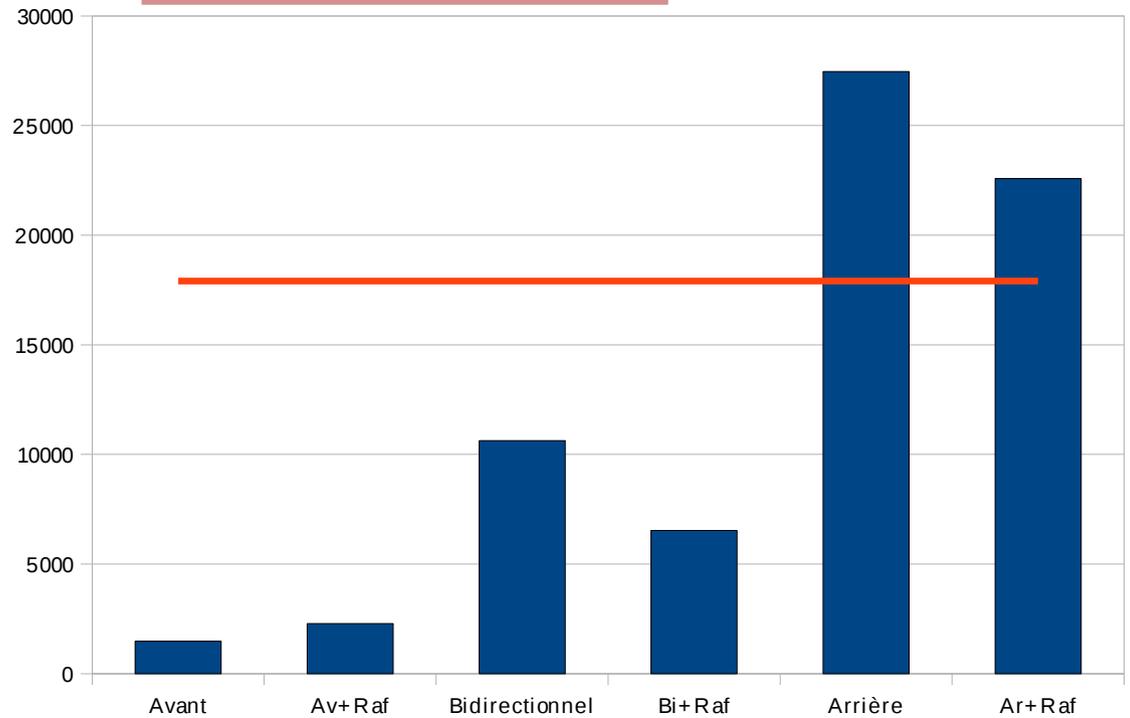
Paramètre	Variation	Intervalle	Choix
Nombre d'états	20 - 100	60 - 100	80
Nombre max de transitions sortantes par état	1 - 16	9 - 16	13
Nombre d'observations	5 - 35	5 - 25	10
Nombre de fautes	1 - 8	1 - 6	5

Choix de la version

Temps moyen par étape :

- Algorithme non-anytime (ligne)
- Algorithmes anytime pour atteindre 80% du résultat final (histogramme)

→ Recherche **en arrière trop lente**

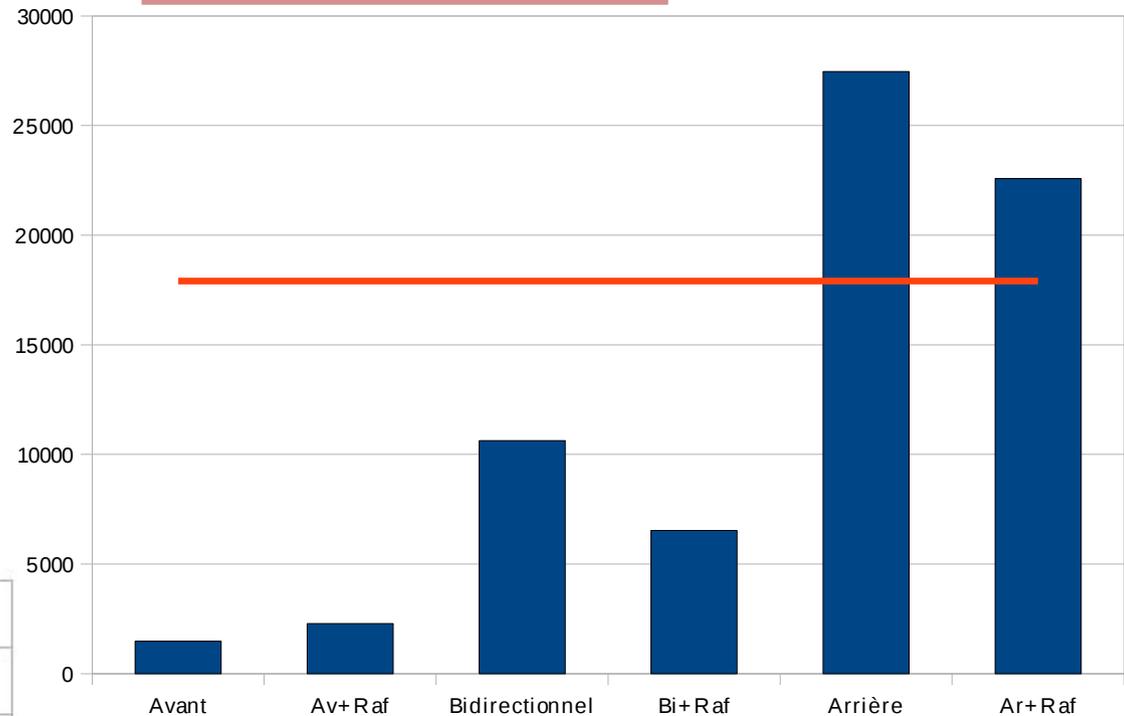
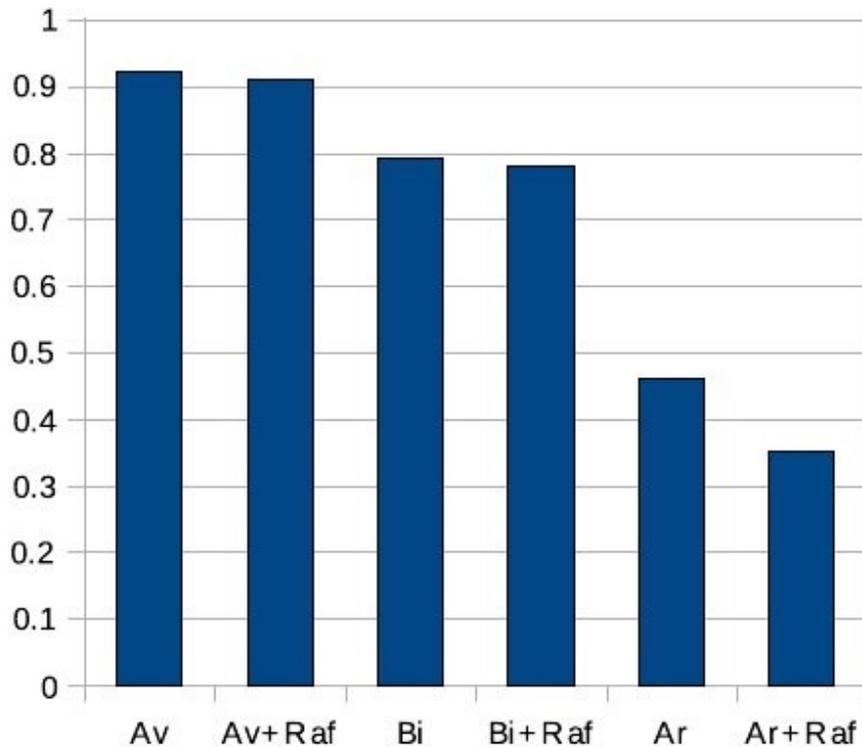


Choix de la version

Temps moyen par étape :

- Algorithme non-anytime (ligne)
- Algorithmes anytime pour atteindre 80% du résultat final (histogramme)

→ Recherche **en arrière trop lente**



Critère de confirmation

(instants d'interruptions fixés par l'algorithme non-anytime)

- Fonction de **raffinage ne ralentit pas**
- Recherche **en avant 10% moins rapide** que l'algorithme de base
- Recherche **bidirectionnelle 20% moins rapide**, mais permet de **cibler les états**

Conclusion

A decorative graphic consisting of several horizontal lines in red and white, extending across the width of the slide below the title.

Conclusion

- Algorithme de diagnostic anytime
 - **Interruptible** et retourne toujours un **résultat correct**
 - Non interrompu : donne le même résultat que l'algorithme de base
- Architecture de processus en parallèle
 - Recherche bidirectionnelle ; fonction de raffinement
 - **6 versions** possibles
- Mesure des performances
 - **3 critères** de qualité et **2 programmes** de mesure
 - Recherche des limitations de l'algorithme anytime
 - Recherche de la version la plus efficace

Conclusion

- Algorithme de diagnostic anytime
 - **Interruptible** et retourne toujours un **résultat correct**
 - Non interrompu : donne le même résultat que l'algorithme de base
- Architecture de processus en parallèle
 - Recherche bidirectionnelle ; fonction de raffinement
 - **6 versions** possibles
- Mesure des performances
 - **3 critères** de qualité et **2 programmes** de mesure
 - Recherche des limitations de l'algorithme anytime
 - Recherche de la version la plus efficace

Perspectives

- **Stratégies de recherche** plus complexes
 - Utilisation de critères pour changer le type de recherche
- Prendre en compte des **requêtes du planificateur**
 - Concentrer les recherches sur une faute particulière

Bibliographie

- *[Russel]* Stuart Russell et Peter Norvig, *Artificial Intelligence, A Modern Approach*, chap. 3-4, Prentice Hall.
- *[Sampath, 96]* Meera Sampath, Raja Sengupta, Stéphane Lafortune, Kasim Sinnamohideen, et Demosthenis C. Teneketzis, *Failure Diagnosis Using Discrete-Event Models*, IEEE transactions on control systems technology Vol. 4 No. 2 pages 105-124, mars 1996.
- *[Zilberstein, 96]* Shlomo Zilberstein, *Using Anytime Algorithms in Intelligent Systems*, AI Magazine Vol. 17 No. 3 pages 73-83, automne 1996.

Diagnostic à la demande

sur un système à événements discrets

20 septembre 2011

Projet de fin d'études



Groupe DISCO

