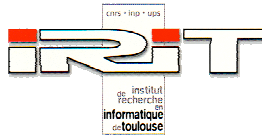


Algorithmique de la Planification

Cours 5 : Espaces d'états

Pierre REGNIER
IRIT - Université Paul Sabatier
<http://www.irit.fr/~Pierre.Regnier>



0. Plan de l'exposé

1. Introduction
 - 1.1. Planificateurs essentiels
 - 1.2. Définitions
2. Algorithmes
 - 2.1. Espace de recherche
 - 2.2. Largeur d'abord
 - 2.3. Profondeur d'abord
 - 2.4. Problèmes
 - 2.5. Algorithme général
 - 2.6. Bilan
3. Heuristiques
 - 3.1. Définitions
 - 3.2. Heuristiques classiques
 - 3.3. Relaxation du problème
 - 3.4. Heuristiques Sum
 - 3.5. Heuristiques Max
 - 3.6. Heuristiques FF
 - 3.7. Tests comparatifs

1.1. Planificateurs essentiels

HSP [Bonnet, Geffner, 1998]
 VVPLAN [Régner, Vidal, 1999]
 HSP-R [Bonnet, Geffner, 1999]
 ALTALT [Nguyen, Kambhampati, 2000]
 FF [Hoffman, 2001]
 YAHSP [Vidal, 2004]
 DOWNWARD [Helmert, Richter, 2004]

Problème de
planification

Plan solution

Recherche dans les espaces d'états

1.2. Définitions

- Un **état** E du monde de la planification est représenté par un ensemble fini de formules atomiques sans symbole de variable. Une formule atomique de base est aussi appelée un **fluent**.
- Un **opérateur** o est un modèle d'action. Il est représenté par son nom et un triplet <pr, ad, de> où pr, ad et de sont des ensembles finis de formules atomiques qui représentent ses préconditions, ajouts et retraits. Prec(o), Add(o), Del(o) dénotent respectivement les ensembles pr, ad, de de l'opérateur o. Une **action**, dénotée par a, est une instance de base d'un opérateur o (toutes les variables de o sont instanciées).

1.2. Définitions

- **Application** d'une action (chaînage avant) :
 - une action A est applicable sur un état E ssi $\text{Prec}(A) \subseteq E$,
 - le nouvel état est l'ensemble de fluents :

$$E \uparrow A = (E - \text{Del}(A)) \cup \text{Add}(A)$$
- **Régression** d'un état par une action (chaînage arrière) :
 - la régression d'un état (partiel) B à travers une action A est possible ssi :
 - A est pertinente : $\text{Add}(A) \cap B \neq \emptyset$ et
 - A est consistante avec B : $\text{Del}(A) \cap B = \emptyset$,
 - le nouvel état (partiel) est l'ensemble de fluents :

$$B \downarrow A = (B - \text{Add}(A)) \cup \text{Prec}(A)$$

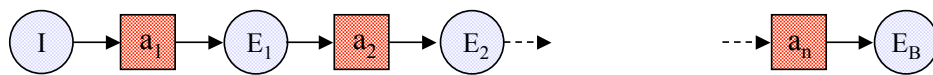
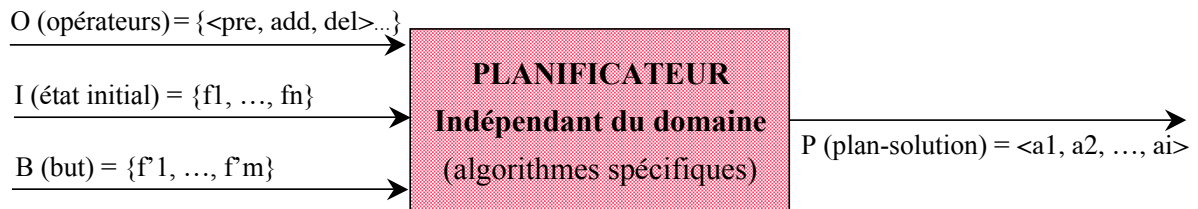
1.2. Définitions

- Un **plan séquentiel** P est une séquence finie (éventuellement vide) d'actions notée $\langle a_1, a_2, \dots, a_n \rangle$.
- L'**application** \mathcal{A} d'un plan d'actions séquentiel P sur un état E est définie par :

$$E \mathcal{A} P = \begin{array}{ll} \text{Si } P = \langle \rangle \text{ ou } E = \perp \text{ alors } E \\ \text{Sinon} & \text{Si } \text{Prec}(\text{tête}(P)) \subseteq E \\ & \text{Alors } (E \uparrow \text{tête}(P)) \mathcal{A} \text{ reste}(P) \\ & \text{Sinon } \perp. \end{array}$$
- Un **problème de planification** est un triplet $\langle O, I, B \rangle$ où :
 - O dénote un ensemble fini d'opérateurs utilisables dans le domaine de la planification considéré (A dénote l'ensemble des actions produites par instantiation des opérateurs de O),
 - I est l'état initial du problème, il est représenté par un ensemble fini de fluents,
 - B est le but du problème, il est représenté par un ensemble fini de fluents.

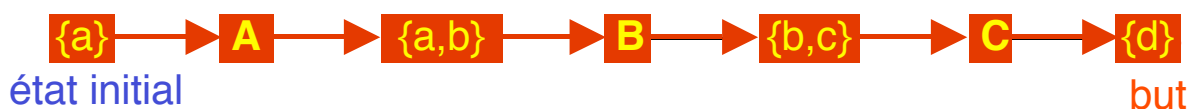
1.2. Définitions

- Un **plan-solution P** au problème de planification $\langle O, I, B \rangle$ est une séquence d'actions $\langle a_1, a_2, \dots, a_n \rangle$ telle que $B \subseteq (I \mathcal{A} P)$: l'application successive de ces actions à I, donne un état résultant qui contienne B (appelé état-but).



1.2. Définitions

$A : a \rightarrow +b$
 $B : a \rightarrow +c -a$
 $C : b \ c \rightarrow +d$
 $D : b \rightarrow +c +a -b$
 $E : c \rightarrow +d -c$



Plan solution : $\langle A, B, C \rangle$

2.1. Espace de recherche

- **Nœuds** : états du monde
 - état initial (en recherche avant, but en arrière : état partiel) ;
 - but : états buts (en recherche avant, initial en arrière) ;
- **Arcs** : actions (opérateurs instanciés) ;
- **Algorithmique** : classique, recherche avant, arrière, mixte...
 - **Aveugles** : profondeur, largeur, DFID...
 - **Informés** : Gloutons, A, A*, WA*, Aε, A*ε, A**, B, BF*, IDA*, HPA...
nécessité de **fonctions heuristiques** pour guider la recherche.

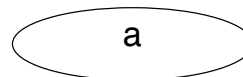
2.2. Largeur d'abord

$A : a \rightarrow +b$

$B : a \rightarrow +c -a$

$C : b \ c \rightarrow +d$

$D : c \rightarrow +b -c$



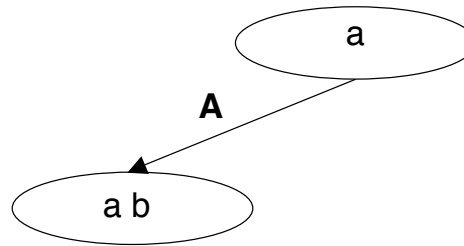
2.2. Largeur d'abord

$A : a \rightarrow +b$

$B : a \rightarrow +c -a$

$C : b c \rightarrow +d$

$D : c \rightarrow +b -c$



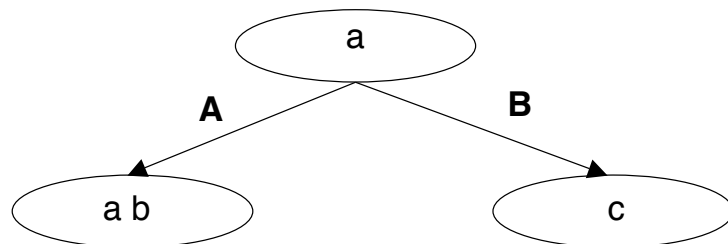
2.2. Largeur d'abord

$A : a \rightarrow +b$

$B : a \rightarrow +c -a$

$C : b c \rightarrow +d$

$D : c \rightarrow +b -c$



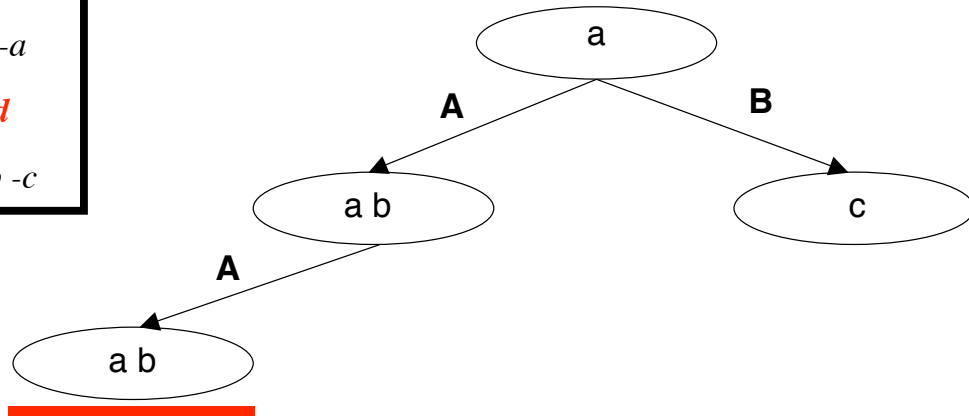
2.2. Largeur d'abord

$A : a \rightarrow +b$

$B : a \rightarrow +c -a$

$C : b c \rightarrow +d$

$D : c \rightarrow +b -c$



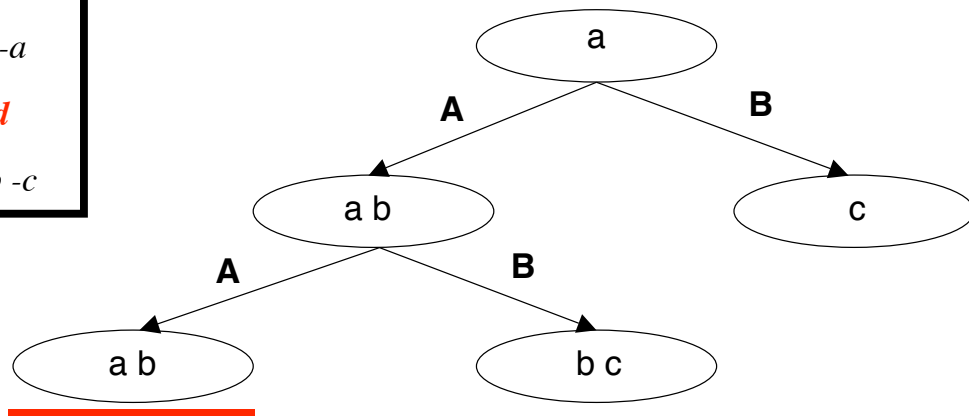
2.2. Largeur d'abord

$A : a \rightarrow +b$

$B : a \rightarrow +c -a$

$C : b c \rightarrow +d$

$D : c \rightarrow +b -c$



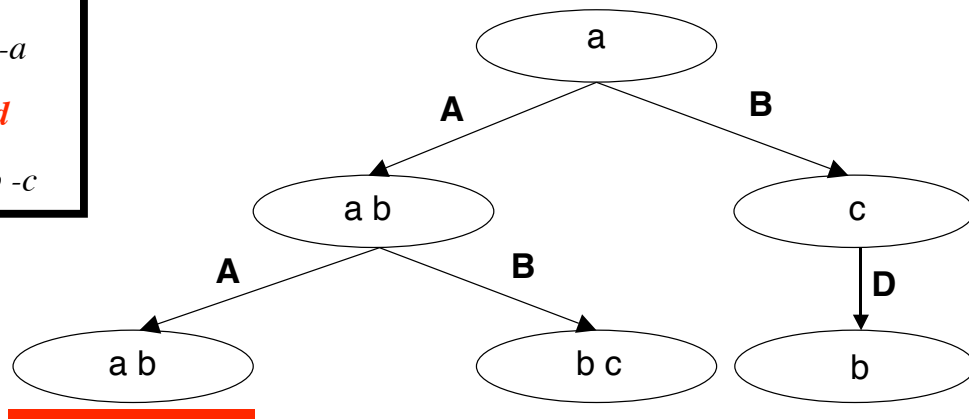
2.2. Largeur d'abord

$A : a \rightarrow +b$

$B : a \rightarrow +c -a$

$C : b c \rightarrow +d$

$D : c \rightarrow +b -c$



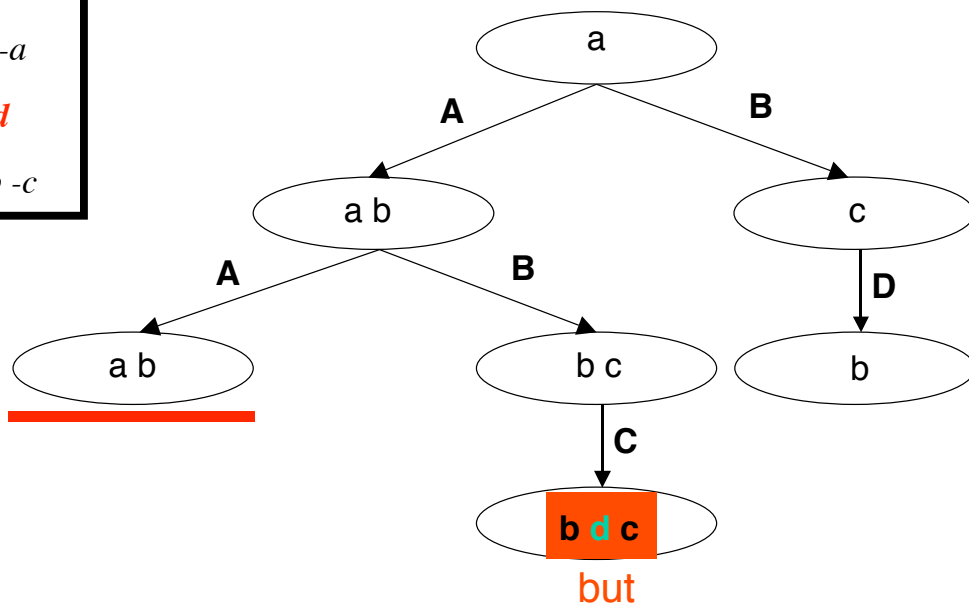
2.2. Largeur d'abord

$A : a \rightarrow +b$

$B : a \rightarrow +c -a$

$C : b c \rightarrow +d$

$D : c \rightarrow +b -c$



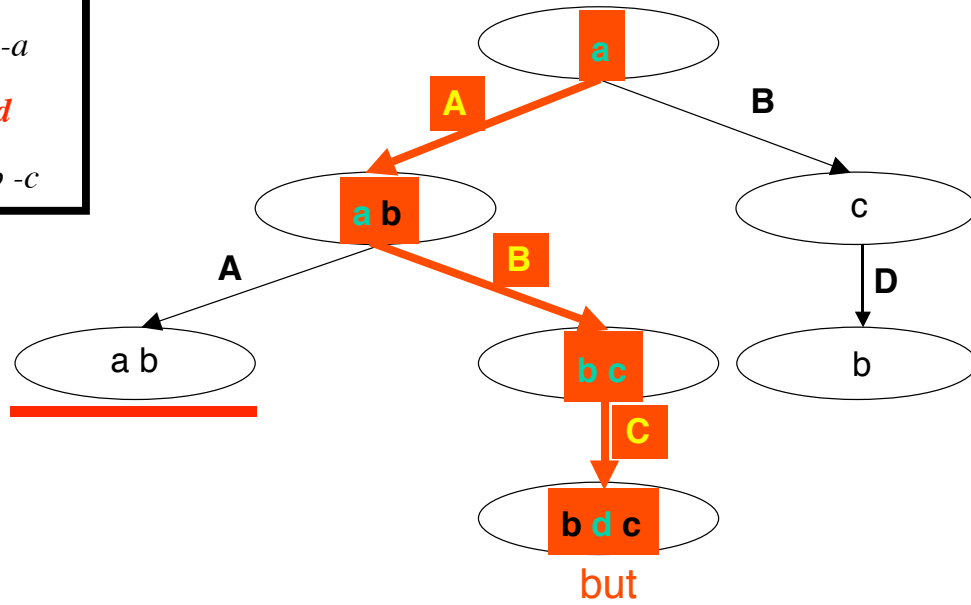
2.2. Largeur d'abord

$A : a \rightarrow +b$

$B : a \rightarrow +c -a$

$C : b c \rightarrow +d$

$D : c \rightarrow +b -c$



Plan solution : $\langle A, B, C \rangle$

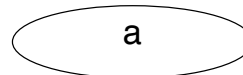
2.3. Profondeur d'abord

$A : a \rightarrow +b$

$B : a \rightarrow +c -a$

$C : b c \rightarrow +d$

$D : c \rightarrow +b -c$



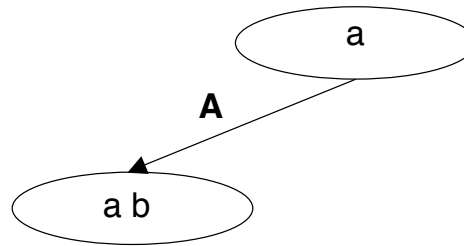
2.3. Profondeur d'abord

$A : a \rightarrow +b$

$B : a \rightarrow +c -a$

$C : b c \rightarrow +d$

$D : c \rightarrow +b -c$



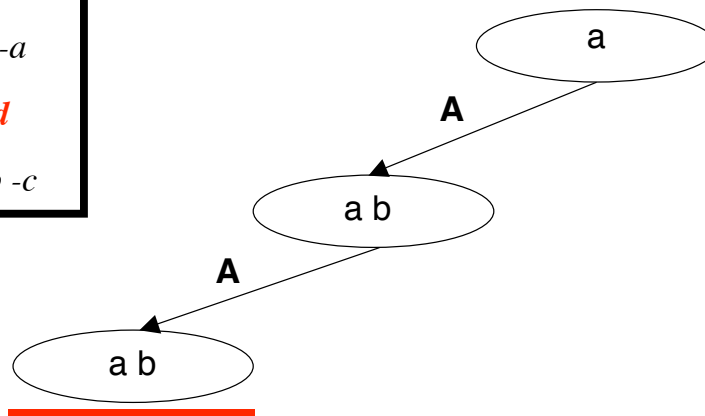
2.3. Profondeur d'abord

$A : a \rightarrow +b$

$B : a \rightarrow +c -a$

$C : b c \rightarrow +d$

$D : c \rightarrow +b -c$



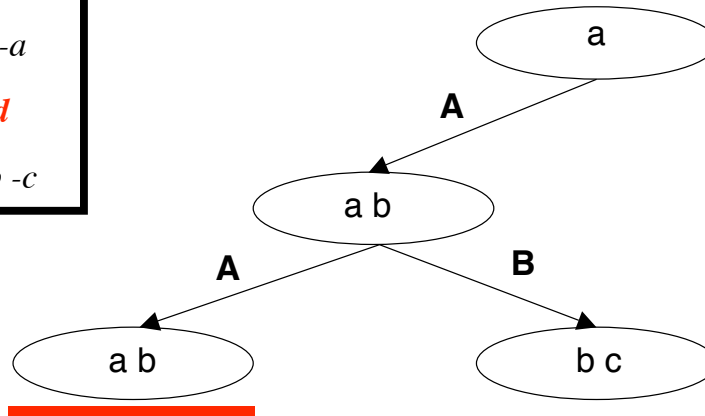
2.3. Profondeur d'abord

$A : a \rightarrow +b$

$B : a \rightarrow +c -a$

$C : b c \rightarrow +d$

$D : c \rightarrow +b -c$



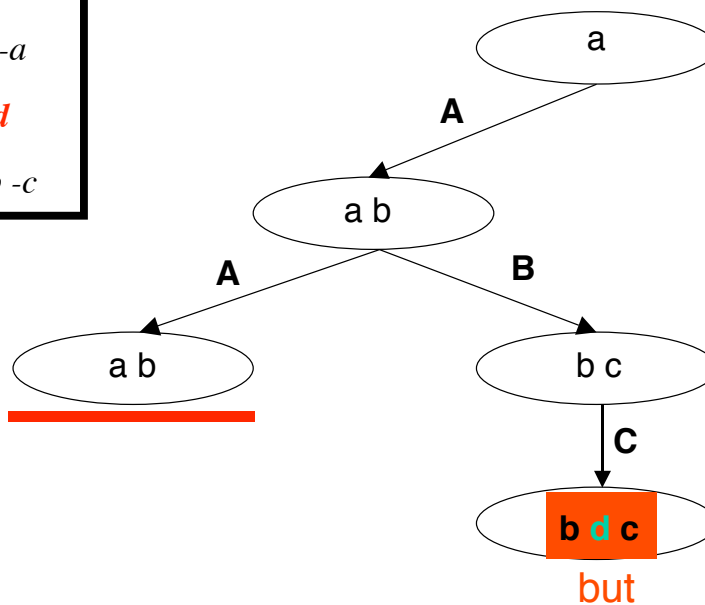
2.3. Profondeur d'abord

$A : a \rightarrow +b$

$B : a \rightarrow +c -a$

$C : b c \rightarrow +d$

$D : c \rightarrow +b -c$



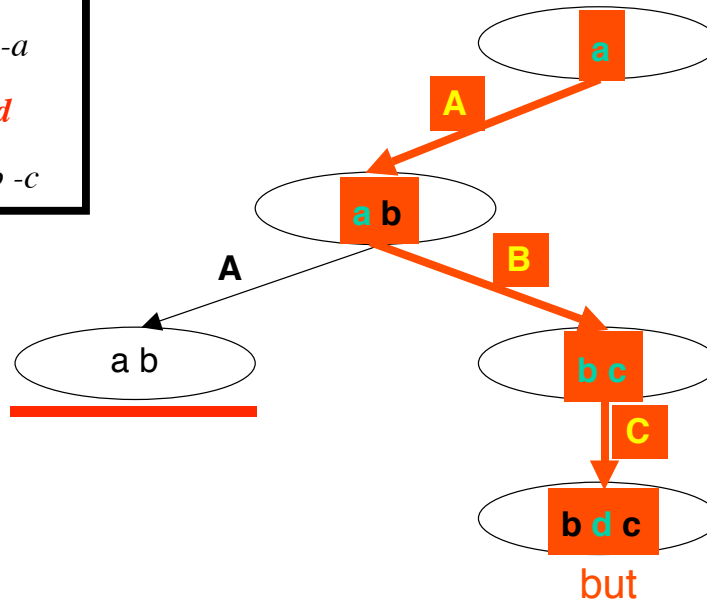
2.3. Profondeur d'abord

$A : a \rightarrow +b$

$B : a \rightarrow +c -a$

$C : b c \rightarrow +d$

$D : c \rightarrow +b -c$



Plan solution : $\langle A, B, C \rangle$

2.4 Problèmes

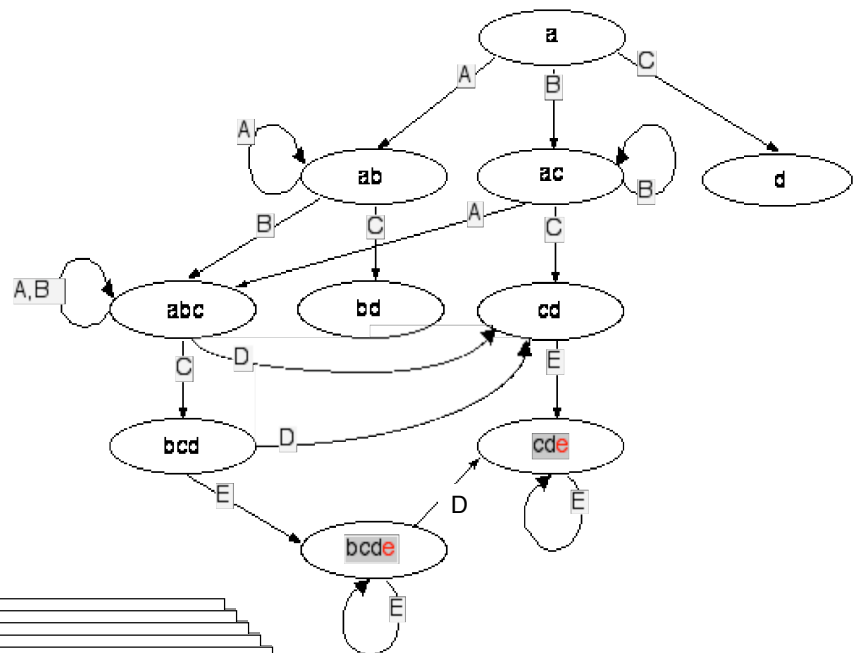
$A : a \rightarrow +b$

$B : a \rightarrow +c$

$C : a \rightarrow +d -a$

$D : b c \rightarrow +d -a -b$

$E : c d \rightarrow +e$



Plan solution 7 : $\langle B, C, E \rangle$

2.5. Algorithme général

Appel initial : RAEE(Actions,ÉtatInitial,But,<>) ;

Fonction RAEE(Actions,ÉtatCourant,But,P) ;

Terminaison :

Si But \subseteq ÉtatCourant **Alors** retourner(P) ;

Choix d'une action :

{ telle que Préconditions(a) \subseteq ÉtatCourant, a rapproche du But }

a \leftarrow **ChoisirAction**(ÉtatCourant,Actions) ;

Si ChoixImpossible **Alors** retourner(échec) ;

Mise à jour :

ÉtatCourant \leftarrow (ÉtatCourant - Retraits(a)) \cup Ajouts(a) ;

P \leftarrow P + <a> ;

Appel récursif :

RAEE(Actions,ÉtatCourant,But,P) ;

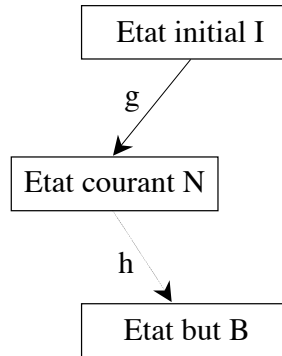
Fin.

2.6 Bilan

- **Performances (à nuancer) :**
 - Actuellement parmi les algorithmes les plus rapides ;
 - Construction de plans de milliers d'actions en quelques dizaines de minutes pour certains domaines ;
- **Atouts (à nuancer) :**
 - Les performances sur des domaines particuliers (logistics...) ;
- **Difficultés :**
 - Qualité des plans-solutions (nombre d'actions, flexibilité...) ;
 - Éviter les bouclages : mémorisation des états déjà explorés ;
 - Diriger la recherche : **fonctions heuristiques** indépendantes du domaine ;
- **Planificateurs :**
 - UNPOP, HSP, HSPr, HSP2, AltAlt, FF, YAHSP, Downward...

3.1. Définitions

- **Heuristique** : fonction d'évaluation d'état permettant de choisir parmi plusieurs d'entre eux celui qui promet d'être le plus efficace pour atteindre un but donné ;
- **Algorithmes A** : fonction heuristique de la forme $f(N) = g(N) + h(N)$;



$g(N)$: coût du chemin parcouru de l'état initial I jusqu'à N (fonction dynamique)

$h(N)$: estimation du coût du chemin restant à parcourir de l'état N jusqu'au but (fonction statique)

- **Algorithmes A*** : h minorante

h minorante (admissible)

h parfaite

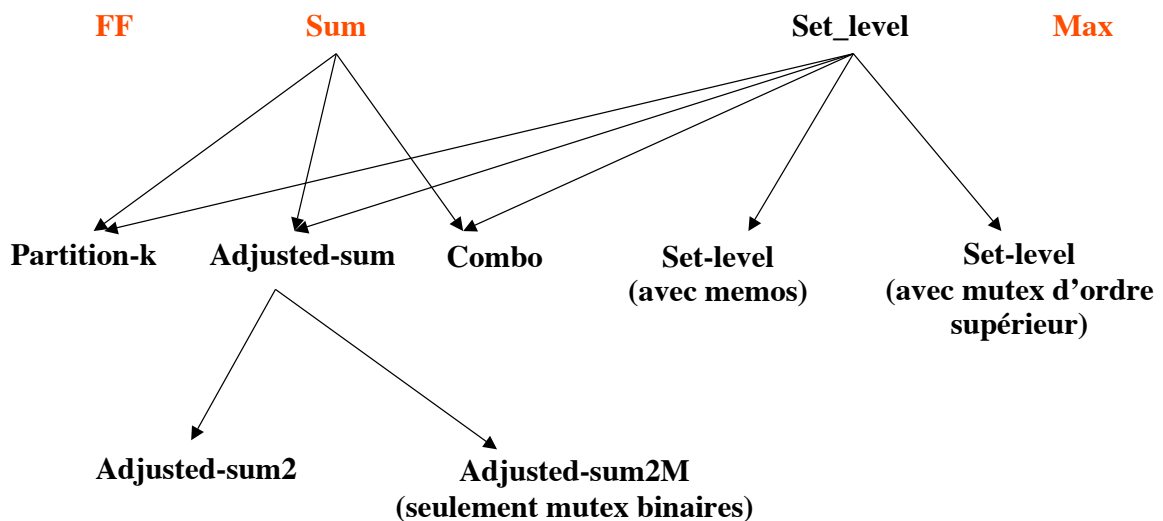
h majorante



3.2. Heuristiques classiques

Heuristiques non admissibles

Heuristiques admissibles



3.3. Relaxation du problème

$$\begin{aligned} A : a &\rightarrow +b -a \\ B : a &\rightarrow +c \\ C : a &\rightarrow +d -a \\ D : b c &\rightarrow +d -b \\ E : c d &\rightarrow +e \end{aligned}$$

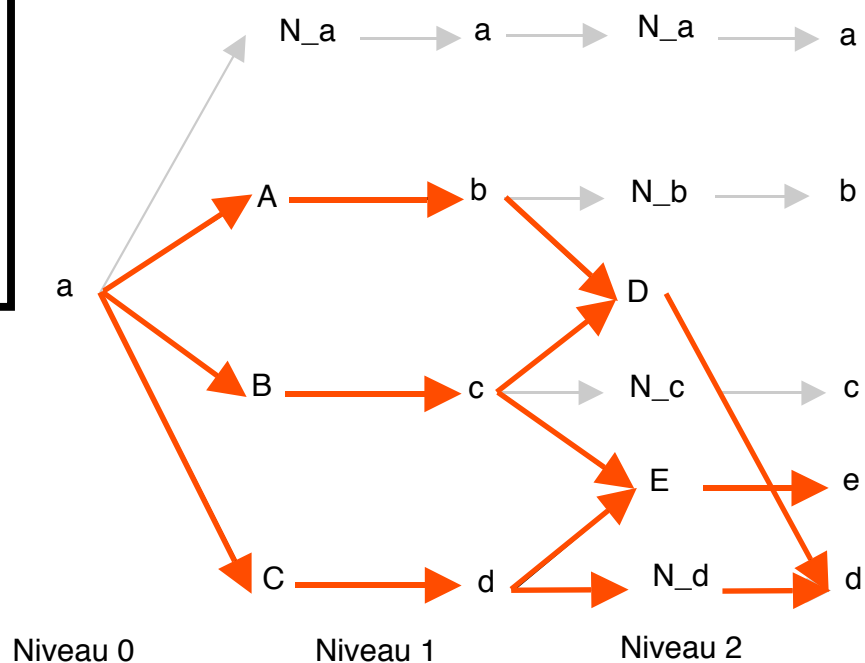
Domaine originel

$$\begin{aligned} A : a &\rightarrow +b \\ B : a &\rightarrow +c \\ C : a &\rightarrow +d \\ D : b c &\rightarrow +d \\ E : c d &\rightarrow +e \end{aligned}$$

Domaine relaxé
par suppression
des retraits

3.3. Relaxation du problème

$$\begin{aligned} A : a &\rightarrow +b \\ B : a &\rightarrow +c \\ C : a &\rightarrow +d \\ D : b c &\rightarrow +d \\ E : c d &\rightarrow +e \end{aligned}$$



3.3. Relaxation du problème

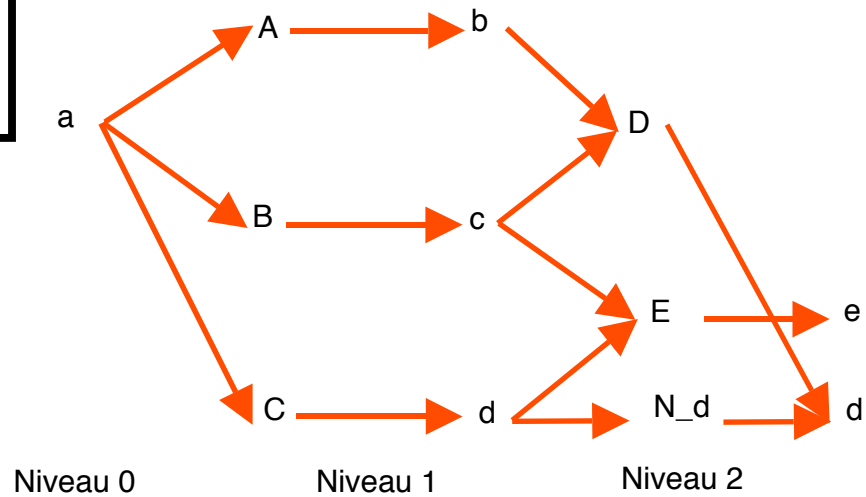
$A : a \rightarrow +b$

$B : a \rightarrow +c$

$C : a \rightarrow +d$

$D : b c \rightarrow +d$

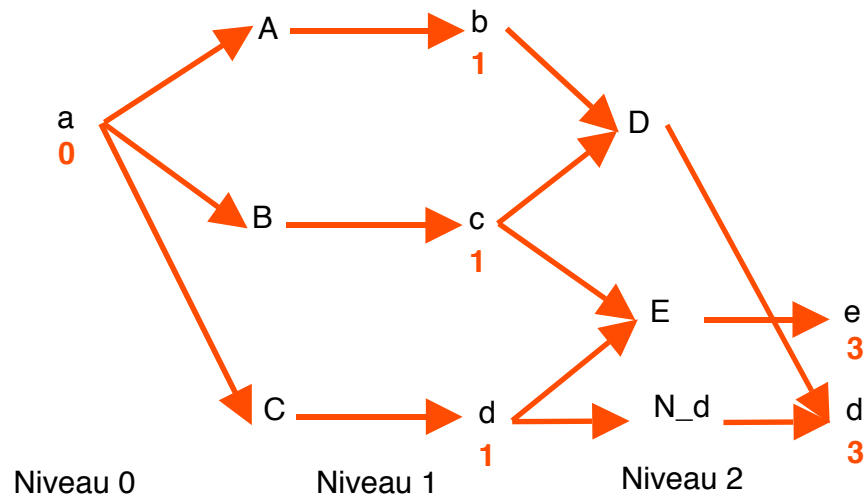
$E : c d \rightarrow +e$



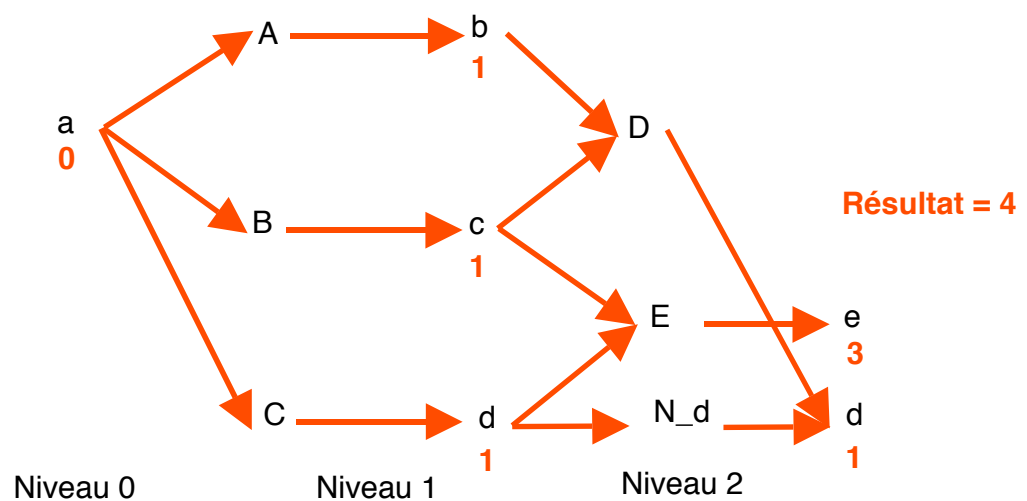
3.4. Heuristique Sum

	Heuristique (graphe relaxé)	Estimation / valeur réelle
HSP HSP2	Sum : $H_s^+(F) = \sum_{f \in F} h_s(f)$	Sur-estimation
	Max : $H_s^{max}(F) = \max_{f \in F} [h_s(f)]$	Sous-estimation
FF	FF : $H(F) = \sum_{i=0, \dots, m} O_i $ O_i : opérateurs au niveau i	Sur-estimation

3.4. Heuristique Sum



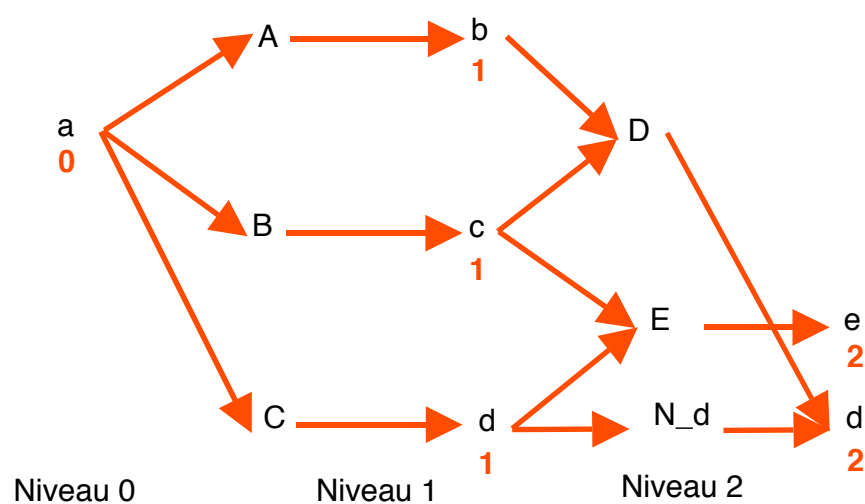
3.4. Heuristique Sum



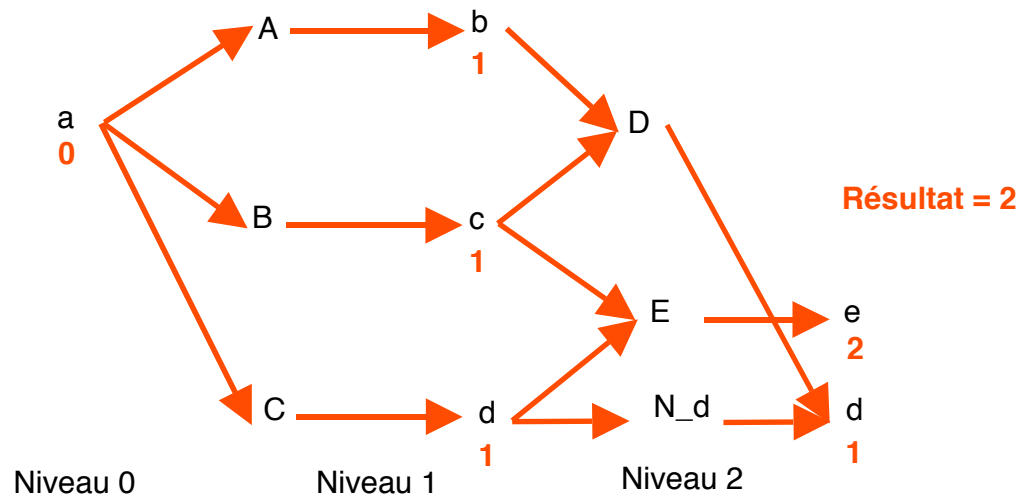
3.5. Heuristique Max

	Heuristique (graphe relaxé)	Estimation / valeur réelle
HSP HSP2	Sum : $H_s^+(F) = \sum_{f \in F} h_s(f)$	Sur-estimation
	Max : $H_s^{max}(F) = \max_{f \in F} [h_s(f)]$	Sous-estimation
FF	FF : $H(F) = \sum_{i=0, \dots, m} O_i $ O_i : opérateurs au niveau i	Sur-estimation

3.5. Heuristique Max



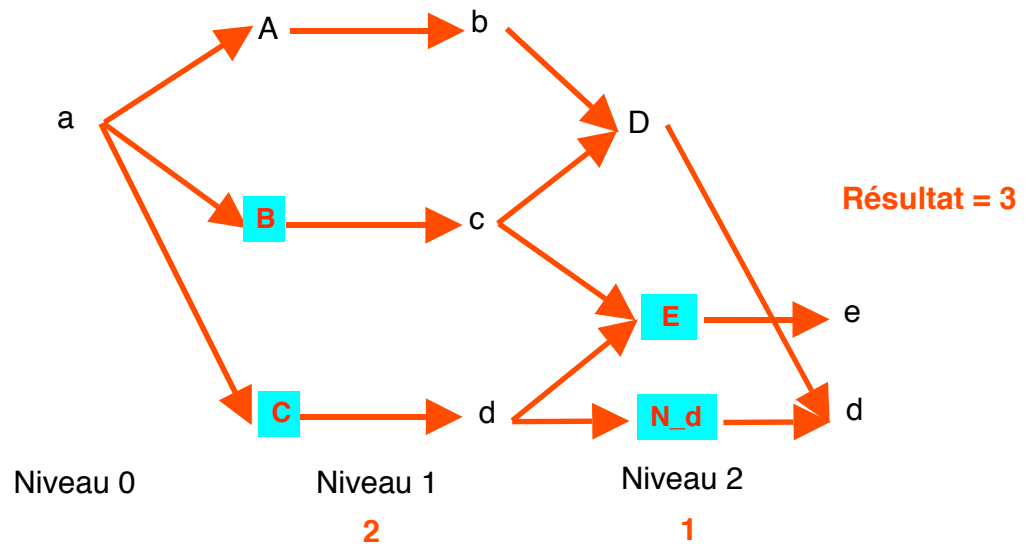
3.5. Heuristique Max



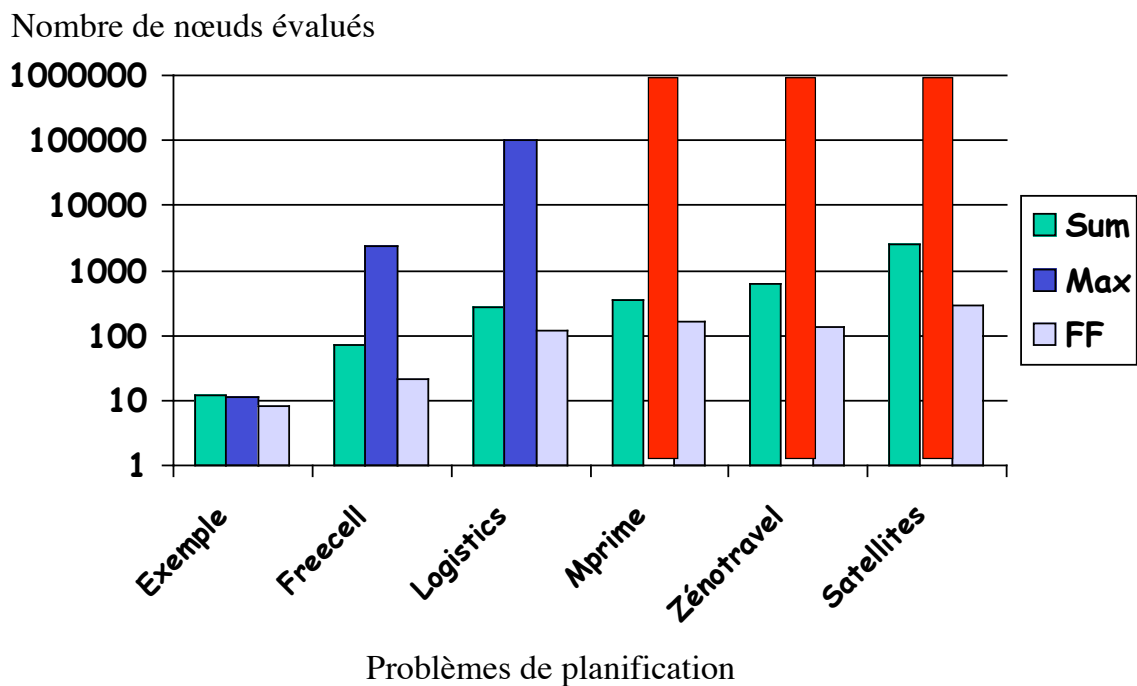
3.6. Heuristique FF

	Heuristique (graphe relaxé)	Estimation / valeur réelle
HSP HSP2	Sum : $H_s^+(F) = \sum_{f \in F} h_s(f)$	Sur-estimation
	Max : $H_s^{max}(F) = \max_{f \in F} [h_s(f)]$	Sous-estimation
FF	FF : $H(F) = \sum_{i=0, \dots, m} O_i $ O_i : opérateurs au niveau i	Sur-estimation

3.6. Heuristique FF



3.7. Tests comparatifs



3.7. Tests comparatifs

	Exemple	Freecell	Logistic	Mprime	Zénotravel	Satellites
Sum	1	1	1.1	1	1.2	1
Max	1	100	1700	*	*	*
FF	1	1	1	6	1	1

$$\text{Pertinence de l'heuristique : } P = \frac{\text{Nombre de nœuds développés}}{\text{Longueur du plan d'actions}}$$