

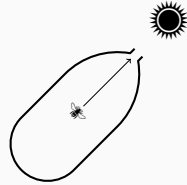
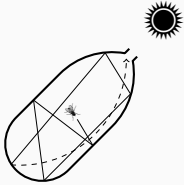
Identification de la meilleure heuristique pour les solveurs de problèmes combinatoires

Hugues Watez

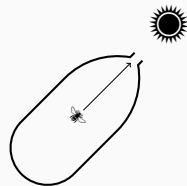
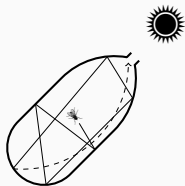
24 janvier 2024

Séminaire équipe COALA au LIS

L'analogie de la *mouche* et de l'*abeille*

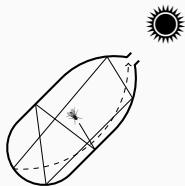


L'analogie de la *mouche* et de l'*abeille*

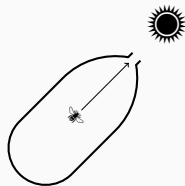


(a) 100 secondes pour résoudre cette instance

L'analogie de la *mouche* et de l'*abeille*

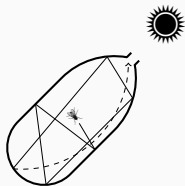


(a) 100 secondes pour résoudre cette instance

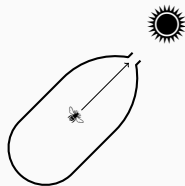


(b) 1 seconde pour résoudre l'instance

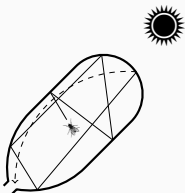
L'analogie de la *mouche* et de l'*abeille*



(a) 100 secondes pour résoudre cette instance

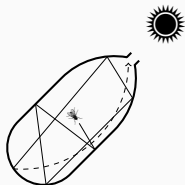


(b) 1 seconde pour résoudre l'instance

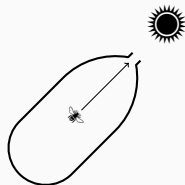


(c) 100 secondes pour résoudre l'instance

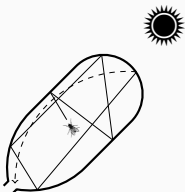
L'analogie de la *mouche* et de l'*abeille*



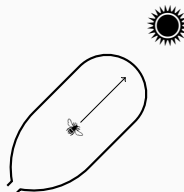
(a) 100 secondes pour résoudre cette instance



(b) 1 seconde pour résoudre l'instance



(c) 100 secondes pour résoudre l'instance



(d) Échec

Les heuristiques de la *mouche* et l'*abeille*

Résultats :

- La mouche résout 2 instances avec une moyenne de 100 secondes
- L'abeille résout 1 instance avec une moyenne de 1 seconde

Si un insecte tierce avait accès à ces deux heuristiques :

- Choix aléatoire ?
- Apprentissage ?

Les heuristiques de la *mouche* et l'*abeille*

Résultats :

- La mouche résout 2 instances avec une moyenne de 100 secondes
- L'abeille résout 1 instance avec une moyenne de 1 seconde

Si un insecte tierce avait accès à ces deux heuristiques :

- Choix aléatoire ?
- Apprentissage ?

Les heuristiques de la *mouche* et l'*abeille*

Résultats :

- La mouche résout 2 instances avec une moyenne de 100 secondes
- L'abeille résout 1 instance avec une moyenne de 1 seconde

Si un insecte tierce avait accès à ces deux heuristiques :

- Choix aléatoire ?
- Apprentissage ?

Les heuristiques de la *mouche* et l'*abeille*

Résultats :

- La mouche résout 2 instances avec une moyenne de 100 secondes
- L'abeille résout 1 instance avec une moyenne de 1 seconde

Si un insecte tierce avait accès à ces deux heuristiques :

- Choix aléatoire ?
- Apprentissage ?

Identification de la meilleure heuristique pour la satisfaction de contraintes

Programmation par contraintes



Framework basé sur le problème des bandits multi-bras



Tournoi adaptatif d'heuristiques



Focus sur l'heuristique *Conflict History Search*

Diversification de l'heuristique



Perturbation de l'heuristique



Export du framework

Vers l'optimisation de contraintes

Vers l'hybridation de solveurs PB/MILP

Identification de la meilleure heuristique pour la satisfaction de contraintes

Definition (Variable)

Une *variable* x est une entité à laquelle on associe une valeur. Cette valeur appartient à son *domaine*, noté $\text{dom}(x)$.

Remarque

Le degré d'une variable x correspond au nombre de contraintes l'impliquant, noté $\text{deg}(x)$.

Definition (Contrainte)

Une *contrainte* c est définie par un ensemble de variables, appelé *portée* (ou *scope*) de c et notée $\text{scp}(c)$, et par une relation mathématique qui décrit l'ensemble des tuples autorisés par c pour les variables de sa portée.

Remarque

L'arité d'une contrainte c correspond à la taille de sa portée.

Definition (CSP)

Un *Problème de Satisfaction de Contraintes* (ou *Réseau de Contraintes*) \mathcal{P} est défini par :

- un ensemble fini de **variables**, noté \mathcal{X} ;
- un ensemble fini de **contraintes**, noté \mathcal{C} , tel que $\forall c \in \mathcal{C}, \text{scp}(c) \subseteq \mathcal{X}$.

Definition (Solution)

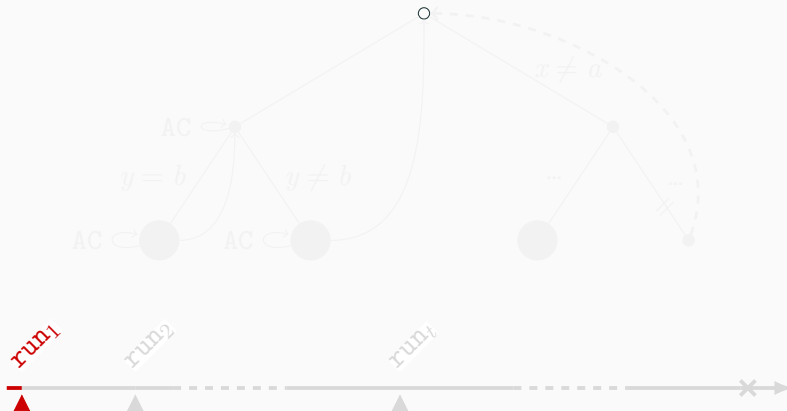
Une *solution* d'une instance CSP \mathcal{P} correspond à l'assignation d'une valeur à chaque variable de \mathcal{P} telle que toutes les contraintes de \mathcal{C} sont satisfaites.

Principe de résolution d'un solveur de contraintes

Approche : arbre binaire de recherche en profondeur avec retour en arrière

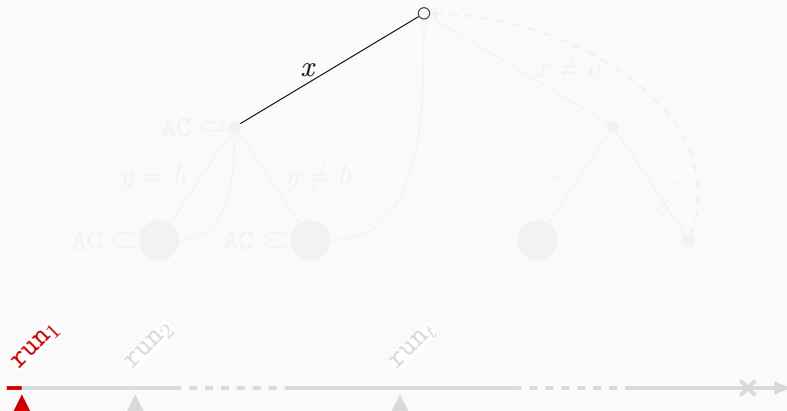
Principe de résolution d'un solveur de contraintes

1^{er} run : racine de l'arbre



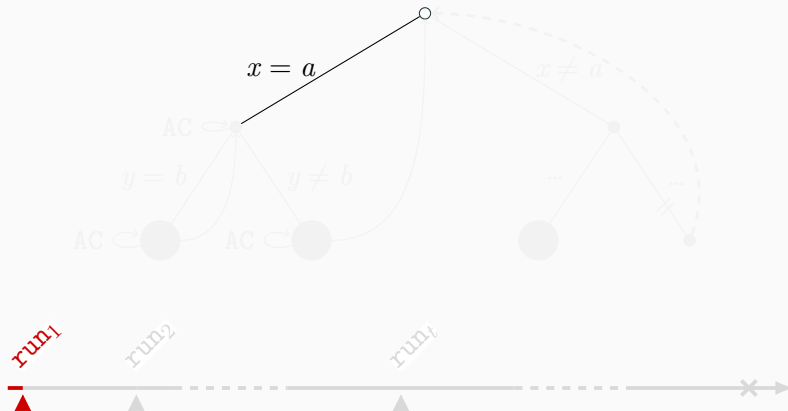
Principe de résolution d'un solveur de contraintes

Décision : l'heuristique de choix de variable sélectionne x



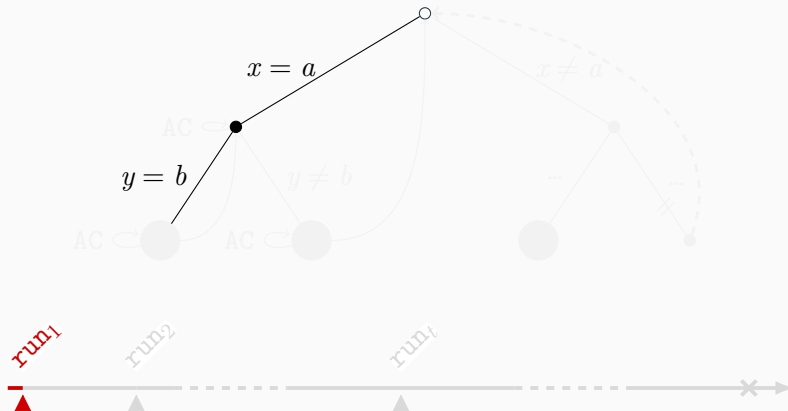
Principe de résolution d'un solveur de contraintes

Décision : l'heuristique de choix de valeurs sélectionne a



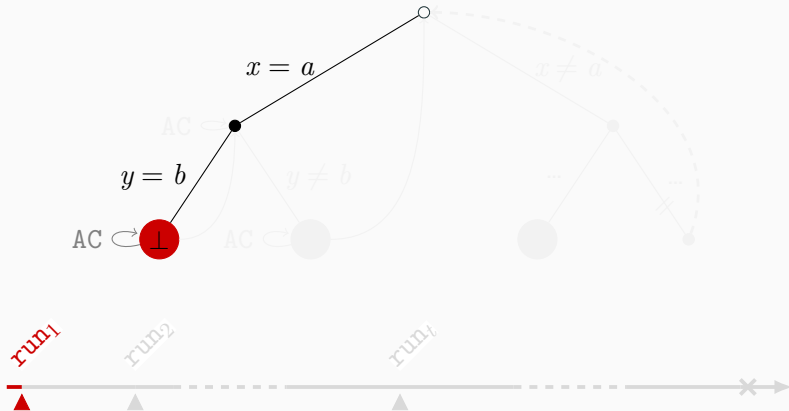
Principe de résolution d'un solveur de contraintes

Décision : sélection suivante (variable, valeur)



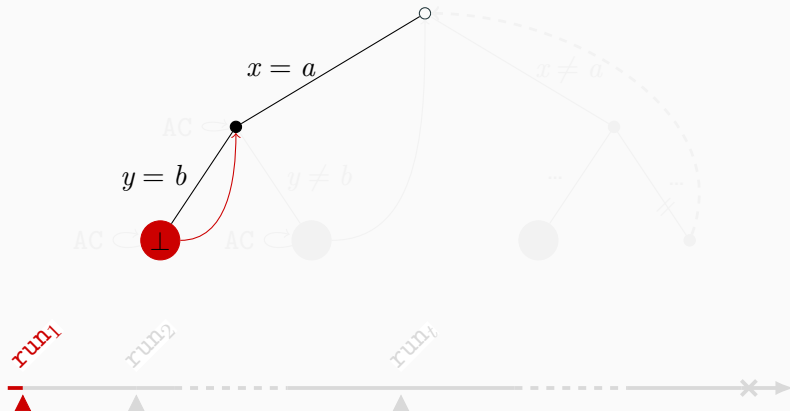
Principe de résolution d'un solveur de contraintes

Propagation : renforcement de l'arc-cohérence et conflit



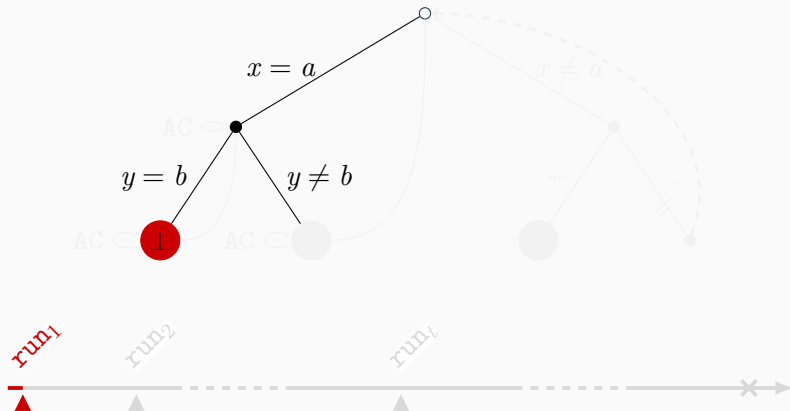
Principe de résolution d'un solveur de contraintes

Retour en arrière : nœud parent



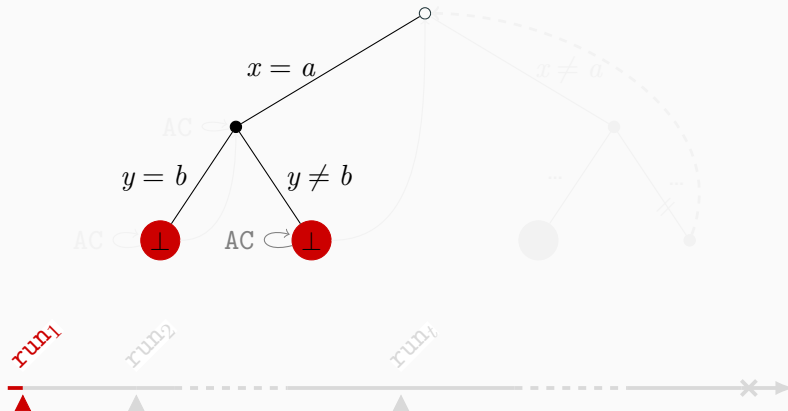
Principe de résolution d'un solveur de contraintes

Réfutation : on considère $y \neq b$



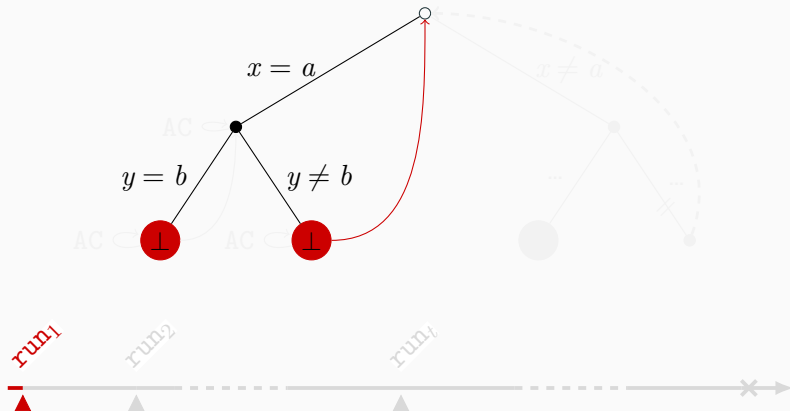
Principe de résolution d'un solveur de contraintes

Propagation : renforcement de la propriété d'AC et conflit



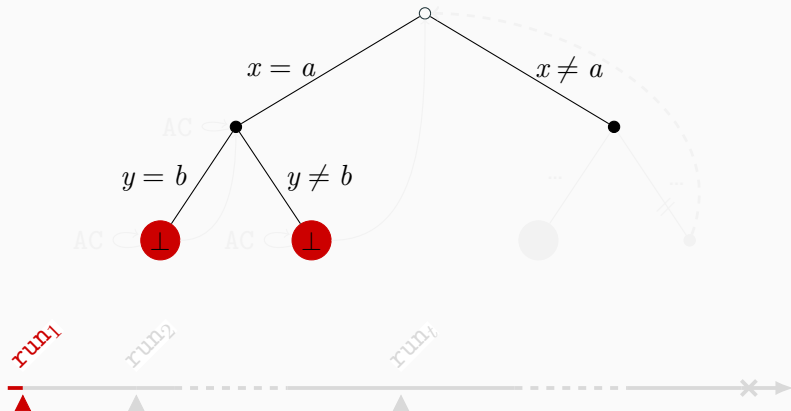
Principe de résolution d'un solveur de contraintes

Retour en arrière : nœud parent (nœud racine)



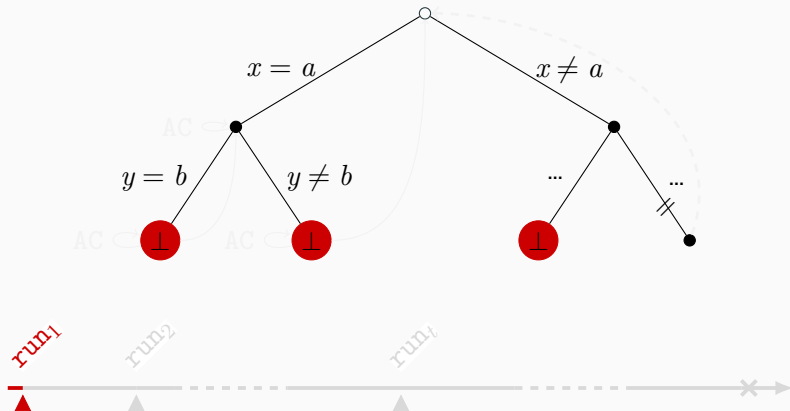
Principe de résolution d'un solveur de contraintes

Réfutation : $x \neq a$



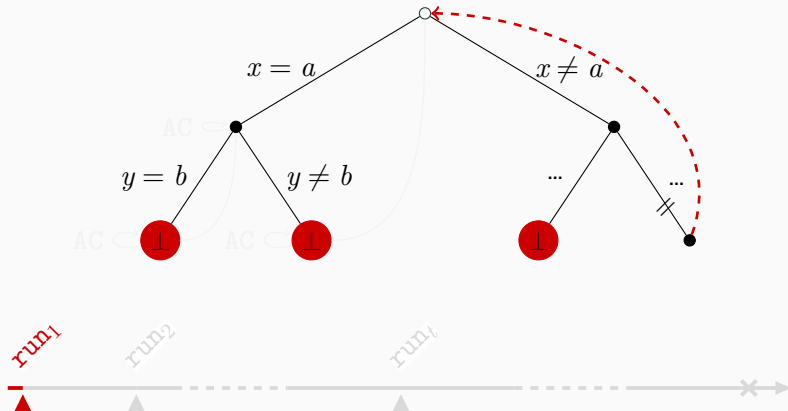
Principe de résolution d'un solveur de contraintes

Redémarrage : cutoff atteint et extraction de nogoods



Principe de résolution d'un solveur de contraintes

Redémarrage : retour au nœud racine



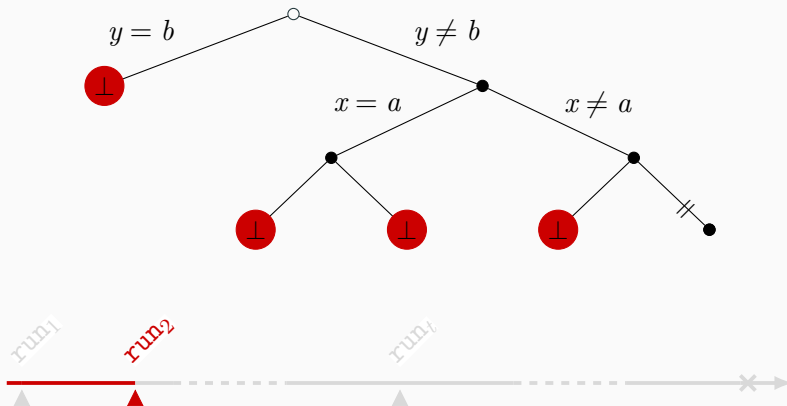
Principe de résolution d'un solveur de contraintes

2^{ème} run : nœud racine



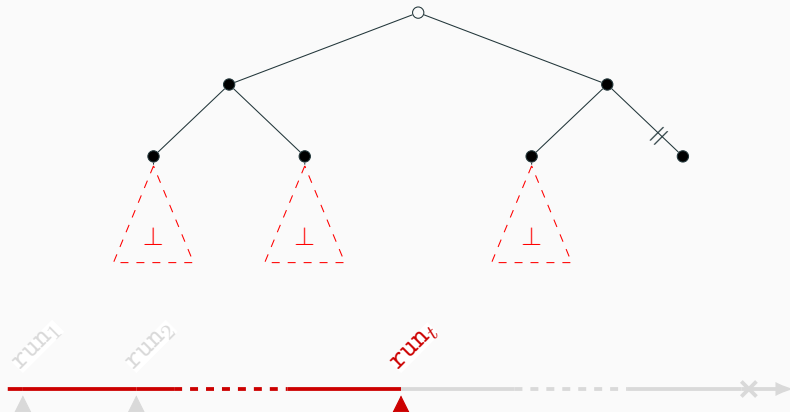
Principe de résolution d'un solveur de contraintes

2^{ème} run : cutoff atteint et redémarrage



Principe de résolution d'un solveur de contraintes

$t^{\text{ème}}$ **run** : cutoff atteint et redémarrage



Principe de résolution d'un solveur de contraintes

Fin de la résolution : satisfaisabilité | insatisfaisabilité | timeout



Interface pour les heuristiques de choix de variable

Soient les **méthodes** suivantes pour décrire une heuristique de choix de variable :

- `INIT()` : initialisation des structures de l'heuristique
- `AVANTRUN()` : appelé à chaque début de *run*
- `APRÈSCONFLIT(c)` : appelé après chaque conflit provoqué par *c*
- `APRÈSASSIGNATION(\mathcal{P}_{av} , \mathcal{P}_{ap} , x)` : après chaque assignation d'une variable *x*
- `SCORE(x)` : retourne le score de *x*

En cas de meilleurs scores identiques, le choix peut être délégué à une autre heuristique ($h_1 > h_2 > \dots$) : le principe de **tie-breaking**.

Interface pour les heuristiques de choix de variable

Soient les **méthodes** suivantes pour décrire une heuristique de choix de variable :

- `INIT()` : initialisation des structures de l'heuristique
- `AVANTRUN()` : appelé à chaque début de *run*
- `APRÈSCONFLIT(c)` : appelé après chaque conflit provoqué par *c*
- `APRÈSASSIGNATION(\mathcal{P}_{av} , \mathcal{P}_{ap} , x)` : après chaque assignation d'une variable *x*
- `SCORE(x)` : retourne le score de *x*

En cas de meilleurs scores identiques, le choix peut être délégué à une autre heuristique ($h_1 > h_2 > \dots$) : le principe de **tie-breaking**.

Interface pour les heuristiques de choix de variable

Soient les **méthodes** suivantes pour décrire une heuristique de choix de variable :

- `INIT()` : initialisation des structures de l'heuristique
- `AVANTRUN()` : appelé à chaque début de *run*
- `APRÈSCONFLIT(c)` : appelé après chaque conflit provoqué par *c*
- `APRÈSASSIGNATION(\mathcal{P}_{av} , \mathcal{P}_{ap} , x)` : après chaque assignation d'une variable *x*
- `SCORE(x)` : retourne le score de *x*

En cas de meilleurs scores identiques, le choix peut être délégué à une autre heuristique ($h_1 > h_2 > \dots$) : le principe de **tie-breaking**.

Interface pour les heuristiques de choix de variable

Soient les **méthodes** suivantes pour décrire une heuristique de choix de variable :

- `INIT()` : initialisation des structures de l'heuristique
- `AVANTRUN()` : appelé à chaque début de *run*
- `APRÈSCONFLIT(c)` : appelé après chaque conflit provoqué par *c*
- `APRÈSASSIGNATION($\mathcal{P}_{av}, \mathcal{P}_{ap}, x$)` : après chaque assignation d'une variable *x*
- `SCORE(x)` : retourne le score de *x*

En cas de meilleurs scores identiques, le choix peut être délégué à une autre heuristique ($h_1 > h_2 > \dots$) : le principe de **tie-breaking**.

Interface pour les heuristiques de choix de variable

Soient les **méthodes** suivantes pour décrire une heuristique de choix de variable :

- $\text{INIT}()$: initialisation des structures de l'heuristique
- $\text{AVANTRUN}()$: appelé à chaque début de *run*
- $\text{APRÈSCONFLIT}(c)$: appelé après chaque conflit provoqué par c
- $\text{APRÈSASSIGNATION}(\mathcal{P}_{av}, \mathcal{P}_{ap}, x)$: après chaque assignation d'une variable x
- $\text{SCORE}(x)$: retourne le score de x

En cas de meilleurs scores identiques, le choix peut être délégué à une autre heuristique ($h_1 > h_2 > \dots$) : le principe de **tie-breaking**.

Interface pour les heuristiques de choix de variable

Soient les **méthodes** suivantes pour décrire une heuristique de choix de variable :

- $\text{INIT}()$: initialisation des structures de l'heuristique
- $\text{AVANTRUN}()$: appelé à chaque début de *run*
- $\text{APRÈSCONFLIT}(c)$: appelé après chaque conflit provoqué par c
- $\text{APRÈSASSIGNATION}(\mathcal{P}_{av}, \mathcal{P}_{ap}, x)$: après chaque assignation d'une variable x
- $\text{SCORE}(x)$: retourne le score de x

En cas de meilleurs scores identiques, le choix peut être délégué à une autre heuristique ($h_1 > h_2 > \dots$) : le principe de **tie-breaking**.

Interface pour les heuristiques de choix de variable

Soient les **méthodes** suivantes pour décrire une heuristique de choix de variable :

- $\text{INIT}()$: initialisation des structures de l'heuristique
- $\text{AVANTRUN}()$: appelé à chaque début de *run*
- $\text{APRÈSCONFLIT}(c)$: appelé après chaque conflit provoqué par c
- $\text{APRÈSASSIGNATION}(\mathcal{P}_{av}, \mathcal{P}_{ap}, x)$: après chaque assignation d'une variable x
- $\text{SCORE}(x)$: retourne le score de x

En cas de meilleurs scores identiques, le choix peut être délégué à une autre heuristique ($h_1 > h_2 > \dots$) : le principe de **tie-breaking**.

Heuristique : dom/ddeg($\mathcal{P} = (\mathcal{X}, \mathcal{C}) : \text{CSP}$)

1 **Méthode** SCORE(x) :

2 └ retourner $\frac{|\text{dom}(x)|}{\text{ddeg}(x)}$

Heuristique adaptative : $wdeg^{ca.cd}$ [7]

Heuristique : $wdeg^{ca.cd}(\mathcal{P} = (\mathcal{X}, \mathcal{C}) : \text{CSP})$

1 Méthode $\text{INIT}()$:

```
2   pour chaque  $c \in \mathcal{C}$  faire
3       pour chaque  $x \in \text{scp}(c)$  faire
4            $w_c^x \leftarrow 0$ 
```

5 Méthode $\text{APRÈS CONFLIT}(c)$:

```
6   pour chaque  $x \in \text{fut}(c)$  faire
7        $r \leftarrow \frac{1}{|\text{scp}(c)| \times |\text{dom}(x)|}$ 
8        $w_c^x \leftarrow w_c^x + r$ 
```

9 Méthode $\text{SCORE}(x)$:

```
10  retourner  $\sum_{c \in \mathcal{C} : x \in \text{scp}(c) \wedge |\text{fut}(c)| > 1} w_c^x$ 
```

Heuristique adaptative : $wdeg^{ca.cd}$ [7]

Heuristique : $wdeg^{ca.cd}(\mathcal{P} = (\mathcal{X}, \mathcal{C}) : \text{CSP})$

```
1 Méthode INIT( ) :  
2   pour chaque  $c \in \mathcal{C}$  faire  
3     pour chaque  $x \in \text{scp}(c)$  faire  
4        $w_c^x \leftarrow 0$   
  
5 Méthode APRÈS CONFLIT( $c$ ) :  
6   pour chaque  $x \in \text{fut}(c)$  faire  
7      $r \leftarrow \frac{1}{|\text{scp}(c)| \times |\text{dom}(x)|}$   
8      $w_c^x \leftarrow w_c^x + r$   
  
9 Méthode SCORE( $x$ ) :  
10  retourner  $\sum_{c \in \mathcal{C} : x \in \text{scp}(c) \wedge |\text{fut}(c)| > 1} w_c^x$ 
```

Heuristique adaptative : $wdeg^{ca.cd}$ [7]

Heuristique : $wdeg^{ca.cd}(\mathcal{P} = (\mathcal{X}, \mathcal{C}) : \text{CSP})$

1 Méthode $\text{INIT}()$:

```
2   pour chaque  $c \in \mathcal{C}$  faire
3       pour chaque  $x \in \text{scp}(c)$  faire
4            $w_c^x \leftarrow 0$ 
```

5 Méthode $\text{APRÈSCONFLIT}(c)$:

```
6   pour chaque  $x \in \text{fut}(c)$  faire
7        $r \leftarrow \frac{1}{|\text{scp}(c)| \times |\text{dom}(x)|}$ 
8        $w_c^x \leftarrow w_c^x + r$ 
```

9 Méthode $\text{SCORE}(x)$:

```
10  retourner  $\sum_{c \in \mathcal{C} : x \in \text{scp}(c) \wedge |\text{fut}(c)| > 1} w_c^x$ 
```

Heuristique adaptative : $wdeg^{ca.cd}$ [7]

Heuristique : $wdeg^{ca.cd}(\mathcal{P} = (\mathcal{X}, \mathcal{C}) : \text{CSP})$

1 Méthode **INIT()** :

```
2   pour chaque  $c \in \mathcal{C}$  faire
3       pour chaque  $x \in \text{scp}(c)$  faire
4            $w_c^x \leftarrow 0$ 
```

5 Méthode **APRÈS CONFLIT**(c) :

```
6   pour chaque  $x \in \text{fut}(c)$  faire
7        $r \leftarrow \frac{1}{|\text{scp}(c)| \times |\text{dom}(x)|}$ 
8        $w_c^x \leftarrow w_c^x + r$ 
```

9 Méthode **SCORE**(x) :

```
10  retourner  $\sum_{c \in \mathcal{C} : x \in \text{scp}(c) \wedge |\text{fut}(c)| > 1} w_c^x$ 
```

Soit l'ensemble $\mathcal{H}^{\text{base}} = \{\text{dom/ddeg}, \text{abs}, \text{ibs}, \text{dom/wdeg}^{\text{chs}}, \text{wdeg}^{\text{ca.cd}}\}$:

Soit l'ensemble $\mathcal{H}^{\text{base}} = \{\text{dom}/\text{ddeg}, \text{abs}, \text{ibs}, \text{dom}/\text{wdeg}^{\text{chs}}, \text{wdeg}^{\text{ca.cd}}\}$:

dom/ddeg : taille de **domaine** et **degré** des variables

Soit l'ensemble $\mathcal{H}^{\text{base}} = \{\text{dom/ddeg}, \text{abs}, \text{ibs}, \text{dom/wdeg}^{\text{chs}}, \text{wdeg}^{\text{ca.cd}}\}$:

dom/ddeg : taille de **domaine** et **degré** des variables

abs : **activité** des variables

Soit l'ensemble $\mathcal{H}^{\text{base}} = \{\text{dom/ddeg}, \text{abs}, \text{ibs}, \text{dom/wdeg}^{\text{chs}}, \text{wdeg}^{\text{ca.cd}}\}$:

dom/ddeg : taille de **domaine** et **degré** des variables

abs : **activité** des variables

ibs : **impact** des variables

Soit l'ensemble $\mathcal{H}^{\text{base}} = \{\text{dom/ddeg}, \text{abs}, \text{ibs}, \text{dom/wdeg}^{\text{chs}}, \text{wdeg}^{\text{ca.cd}}\}$:

dom/ddeg : taille de **domaine** et **degré** des variables

abs : **activité** des variables

ibs : **impact** des variables

dom/wdeg^{chs} : **historique** de conflits des contraintes

Soit l'ensemble $\mathcal{H}^{\text{base}} = \{\text{dom}/\text{ddeg}, \text{abs}, \text{ibs}, \text{dom}/\text{wdeg}^{\text{chs}}, \text{wdeg}^{\text{ca.cd}}\}$:

dom/ddeg : taille de **domaine** et **degré** des variables

abs : **activité** des variables

ibs : **impact** des variables

dom/wdeg^{chs} : **historique** de conflits des contraintes

wdeg^{ca.cd} : **arité** et **domaine** des variables du conflit

instances CSP (\mathcal{I}_{CSP}) : 810 *instances* XCSP'17/18/19 (83 *familles*)

instances COP (\mathcal{I}_{COP}) : 697 instances XCSP'18 (51 familles)

instances CSP (\mathcal{I}_{CSP}) : 810 *instances* XCSP'17/18/19 (83 *familles*)

instances COP (\mathcal{I}_{COP}) : 697 instances XCSP'18 (51 *familles*)

nœud de calcul : CPU 3,3 *GHz* Intel XEON E5-2643 et 32 *GB* RAM

Contexte expérimental (fixe)

instances CSP (\mathcal{I}_{CSP}) : 810 *instances* XCSP'17/18/19 (83 *familles*)
instances COP (\mathcal{I}_{COP}) : 697 instances XCSP'18 (51 *familles*)
 nœud de calcul : CPU 3,3 *GHz* Intel XEON E5-2643 et 32 *GB RAM*
 timeout : 2400 *secondes*

Contexte expérimental (fixe)

instances CSP (\mathcal{I}_{CSP}) : 810 *instances* XCSP'17/18/19 (83 *familles*)
instances COP (\mathcal{I}_{COP}) : 697 instances XCSP'18 (51 *familles*)
 nœud de calcul : CPU 3,3 *GHz* Intel XEON E5-2643 et 32 *GB RAM*
 timeout : 2400 *secondes*
 solveur : ACE

Contexte expérimental (fixe)

instances CSP (\mathcal{I}_{CSP}) : 810 *instances* XCSP'17/18/19 (83 *familles*)
instances COP (\mathcal{I}_{COP}) : 697 instances XCSP'18 (51 *familles*)
 nœud de calcul : CPU 3,3 *GHz* Intel XEON E5-2643 et 32 *GB RAM*
 timeout : 2400 *secondes*
 solveur : ACE
suite de redémarrage suivie : $u \times \text{luby}_t$ où $u = 100$
 premiers termes : 100, 100, 200, 100, 100, 200, 400, ...

Contexte expérimental (fixe)

instances CSP (\mathcal{I}_{CSP}) : 810 *instances* XCSP'17/18/19 (83 *familles*)
instances COP (\mathcal{I}_{COP}) : 697 instances XCSP'18 (51 *familles*)
 nœud de calcul : CPU 3,3 *GHz* Intel XEON E5-2643 et 32 *GB RAM*
 timeout : 2400 *secondes*
 solveur : ACE
suite de redémarrage suivie : $u \times \text{luby}_t$ où $u = 100$
 premiers termes : 100, 100, 200, 100, 100, 200, 400, ...
 unité de *cutoff* : *mauvaises décisions*

Contexte expérimental (fixe)

instances CSP (\mathcal{I}_{CSP}) : 810 *instances* XCSP'17/18/19 (83 *familles*)

instances COP (\mathcal{I}_{COP}) : 697 instances XCSP'18 (51 *familles*)

nœud de calcul : CPU 3,3 *GHz* Intel XEON E5-2643 et 32 *GB RAM*

timeout : 2400 *secondes*

solveur : ACE

suite de redémarrage suivie : $u \times \text{luby}_t$ où $u = 100$

premiers termes : 100, 100, 200, 100, 100, 200, 400, ...

unité de *cutoff* : *mauvaises décisions*

heuristique de choix de valeur : min-dom

Contexte expérimental (fixe)

instances CSP (\mathcal{I}_{CSP}) :	810 <i>instances</i> XCSP'17/18/19 (83 <i>familles</i>)
instances COP (\mathcal{I}_{COP}) :	697 instances XCSP'18 (51 familles)
nœud de calcul :	CPU 3,3 GHz Intel XEON E5-2643 et 32 GB RAM
timeout :	2400 <i>secondes</i>
solveur :	ACE
suite de redémarrage suivie :	$u \times \text{luby}_t$ où $u = 100$ <i>premiers termes</i> : 100, 100, 200, 100, 100, 200, 400, ...
unité de <i>cutoff</i> :	<i>mauvaises décisions</i>
heuristique de choix de valeur :	min-dom
propriété de propagation :	<i>arc-cohérence</i>

Contexte expérimental (fixe)

instances CSP (\mathcal{I}_{CSP}) :	810 <i>instances</i> XCSP'17/18/19 (83 <i>familles</i>)
instances COP (\mathcal{I}_{COP}) :	697 instances XCSP'18 (51 <i>familles</i>)
nœud de calcul :	CPU 3,3 <i>GHz</i> Intel XEON E5-2643 et 32 <i>GB RAM</i>
timeout :	2400 <i>secondes</i>
solveur :	ACE
suite de redémarrage suivie :	$u \times \text{luby}_t$ où $u = 100$ <i>premiers termes</i> : 100, 100, 200, 100, 100, 200, 400, ...
unité de <i>cutoff</i> :	<i>mauvaises décisions</i>
heuristique de choix de valeur :	min-dom
propriété de propagation :	<i>arc-cohérence</i>
apprentissage :	<i>nogoods</i> de fin de run

$\Psi_{\text{base}} =$

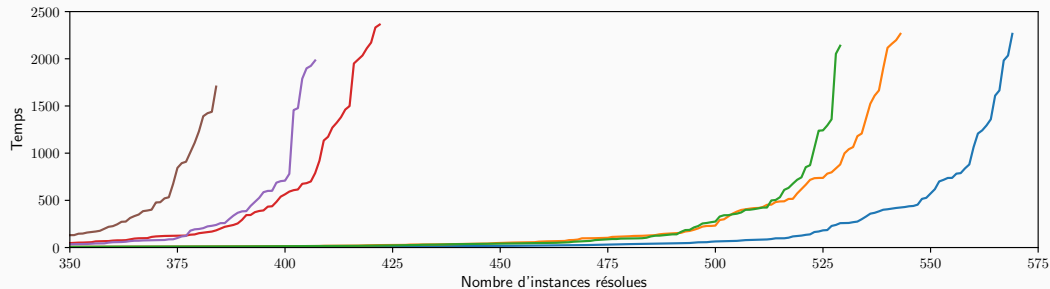
$$\bullet \left\{ \text{ACE}(h) \mid h \in \mathcal{H}^{\text{base}} \right\}$$

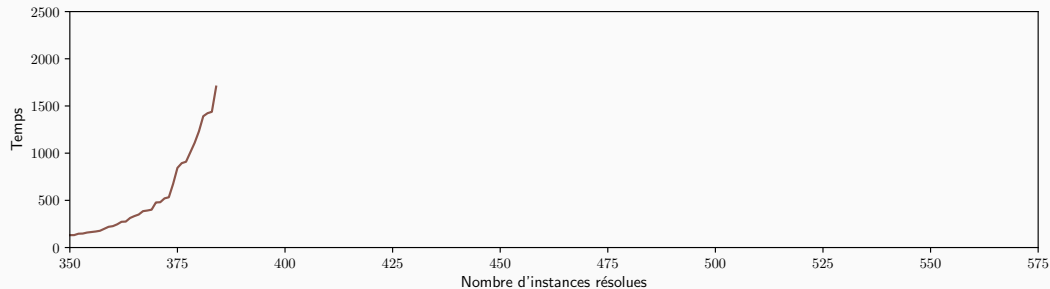
Heuristiques de base

$\Psi_{\text{base}} =$

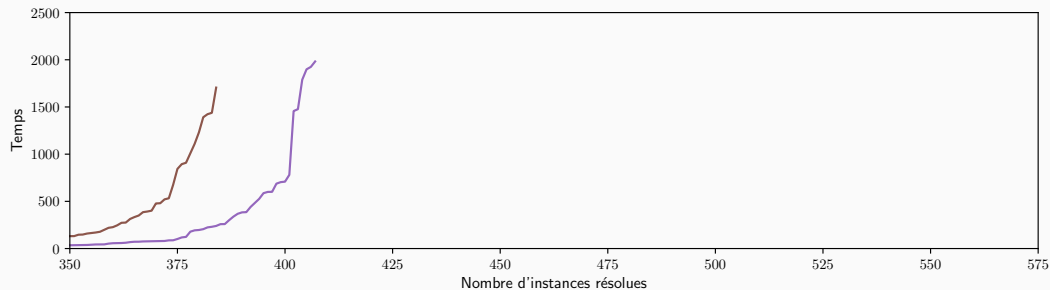
- $\left\{ \text{ACE}(h) \mid h \in \mathcal{H}^{\text{base}} \right\}$

Heuristiques de base

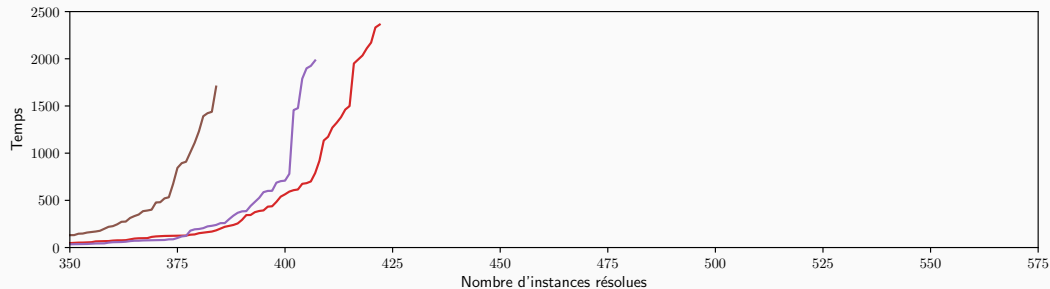




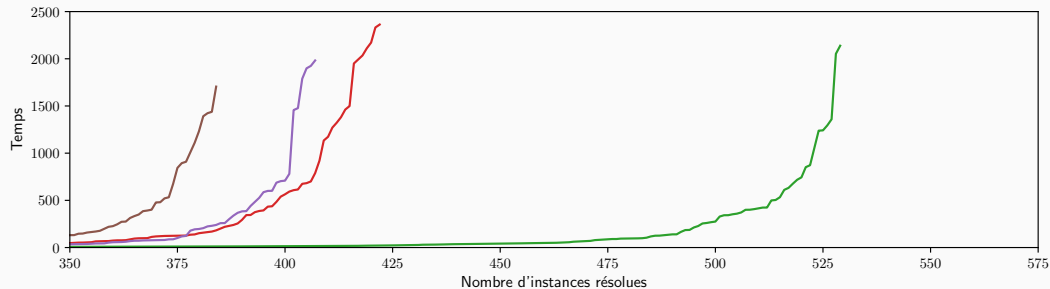
	#RÉS.	TPS (s)	#RÉS. COMM.	TPS COMM. (s)	#TOT.
ibs	384	467,669	314	14,795	569



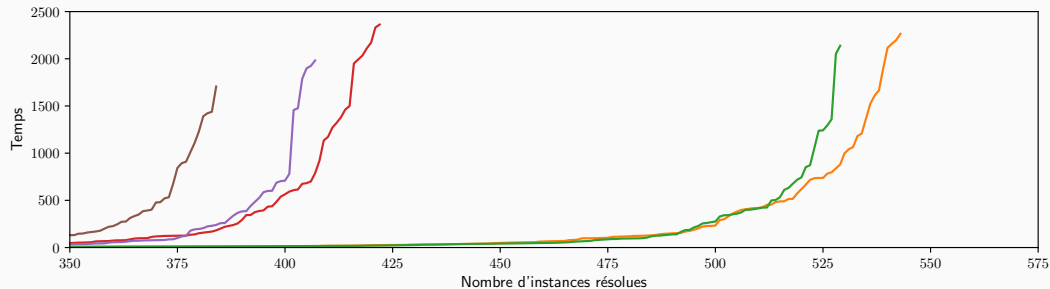
	#RÉS.	TPS (s)	#RÉS. COMM.	TPS COMM. (s)	#TOT.
dom/ddeg	407	413,055	314	5,472	569
ibs	384	467,669	314	14,795	569



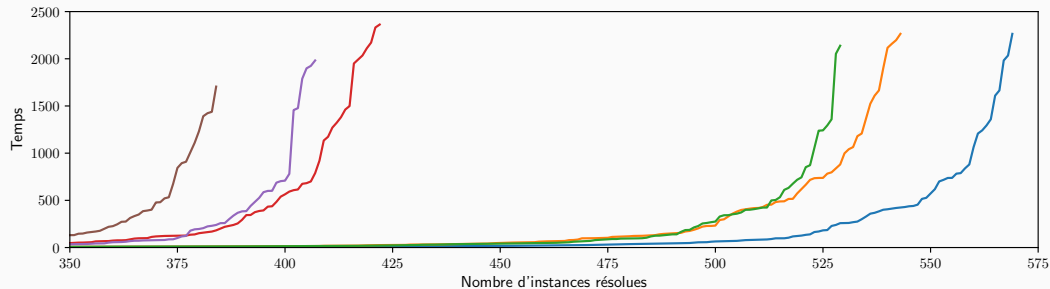
	#RÉS.	TPS (s)	#RÉS. COMM.	TPS COMM. (s)	#TOT.
abs	422	394,084	314	18,831	569
dom/ddeg	407	413,055	314	5,472	569
ibs	384	467,669	314	14,795	569



	#RÉS.	TPS (s)	#RÉS. COMM.	TPS COMM. (s)	#TOT.
dom/wdeg ^{chs}	529	125,819	314	4,029	569
abs	422	394,084	314	18,831	569
dom/ddeg	407	413,055	314	5,472	569
ibs	384	467,669	314	14,795	569



	#RÉS.	TPS (s)	#RÉS. COMM.	TPS COMM. (s)	#TOT.
$wdeg^{ca.cd}$	543	109,146	314	5,908	569
$dom/wdeg^{chs}$	529	125,819	314	4,029	569
abs	422	394,084	314	18,831	569
$dom/ddeg$	407	413,055	314	5,472	569
ibs	384	467,669	314	14,795	569



	#RÉS.	TPS (s)	#RÉS. COMM.	TPS COMM. (s)	#TOT.
$\text{VBS}(\mathcal{H}^{\text{base}})$	569	37,955	314	2,830	569
$\text{wdeg}^{\text{ca.cd}}$	543	109,146	314	5,908	569
$\text{dom}/\text{wdeg}^{\text{cha}}$	529	125,819	314	4,029	569
abs	422	394,084	314	18,831	569
dom/ddeg	407	413,055	314	5,472	569
ibs	384	467,669	314	14,795	569

Sélection de l'heuristique de choix de variable :

- par l'utilisateur à l'exécution du solveur
- par l'application d'une politique d'exploration-exploitation sur un ensemble d'heuristiques.

Sélection de l'heuristique de choix de variable :

- ~~par l'utilisateur à l'exécution du solveur~~
- par l'application d'une politique d'exploration-exploitation sur un ensemble d'heuristiques.

Problème de bandit multi-bras : pourquoi ce nom ?

Des bandits manchots aux différentes probabilités de gagner :



2%



1,5%



1%



1,5%



10%

Le problème du bandit multi-bras se caractérise par :

- la recherche d'un équilibre entre **exploration** et **exploitation**

Problème de bandit multi-bras : pourquoi ce nom ?

Des bandits manchots aux différentes probabilités de gagner :



2%



1,5%



1%



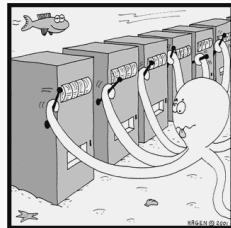
1,5%



10%

Le problème du bandit multi-bras se caractérise par :

- la recherche d'un équilibre entre **exploration** et **exploitation**



Problème de bandit multi-bras : pourquoi ce nom ?

Des bandits manchots aux différentes probabilités de gagner :



Le problème du bandit multi-bras se caractérise par :

- la recherche d'un équilibre entre **exploration** et **exploitation**

Problème de bandit multi-bras : pourquoi ce nom ?

Des bandits manchots aux différentes probabilités de gagner :

\mathcal{H}_1	\mathcal{H}_2	\mathcal{H}_3	\mathcal{H}_4	\mathcal{H}_5
2%	1,5%	1%	1,5%	10%

Le problème du bandit multi-bras se caractérise par :

- la recherche d'un équilibre entre **exploration** et **exploitation**

Description globale d'un bandit

Le problème des bandits est décrit comme un jeu où un **joueur** affronte l'**environnement**. À chaque tour t :

- le joueur **choisit une action** i_t parmi un ensemble d'actions A (machines, traitements, heuristiques, etc.)
- l'environnement **donne une récompense** $\mathcal{R}_t(i_t)$ au joueur pour l'action sélectionnée

Le but du joueur est de minimiser son **regret** au bout de T tours :

$$Regret_T = \max_{i \in A} \sum_{t=1}^T \mathcal{R}_t(i) - \sum_{t=1}^T \mathcal{R}_t(i_t)$$

Description globale d'un bandit

Le problème des bandits est décrit comme un jeu où un **joueur** affronte l'**environnement**. À chaque tour t :

- le joueur **choisit une action** i_t parmi un ensemble d'actions A (machines, traitements, heuristiques, etc.)
- l'environnement **donne une récompense** $\mathcal{R}_t(i_t)$ au joueur pour l'action sélectionnée

Le but du joueur est de minimiser son **regret** au bout de T tours :

$$Regret_T = \max_{i \in A} \sum_{t=1}^T \mathcal{R}_t(i) - \sum_{t=1}^T \mathcal{R}_t(i_t)$$

Description globale d'un bandit

Le problème des bandits est décrit comme un jeu où un **joueur** affronte l'**environnement**. À chaque tour t :

- le joueur **choisit une action** i_t parmi un ensemble d'actions A (machines, traitements, heuristiques, etc.)
- l'environnement **donne une récompense** $\mathcal{R}_t(i_t)$ au joueur pour l'action sélectionnée

Le but du joueur est de minimiser son **regret** au bout de T tours :

$$Regret_T = \max_{i \in A} \sum_{t=1}^T \mathcal{R}_t(i) - \sum_{t=1}^T \mathcal{R}_t(i_t)$$

Description globale d'un bandit

Le problème des bandits est décrit comme un jeu où un **joueur** affronte l'**environnement**. À chaque tour t :

- le joueur **choisit une action** i_t parmi un ensemble d'actions A (machines, traitements, heuristiques, etc.)
- l'environnement **donne une récompense** $\mathcal{R}_t(i_t)$ au joueur pour l'action sélectionnée

Le but du joueur est de minimiser son **regret** au bout de T tours :

$$Regret_T = \max_{i \in A} \sum_{t=1}^T \mathcal{R}_t(i) - \sum_{t=1}^T \mathcal{R}_t(i_t)$$

Description globale d'un bandit

Le problème des bandits est décrit comme un jeu où un **joueur** affronte l'**environnement**. À chaque tour t :

- le joueur **choisit une action** i_t parmi un ensemble d'actions A (machines, traitements, heuristiques, etc.)
- l'environnement **donne une récompense** $\mathcal{R}_t(i_t)$ au joueur pour l'action sélectionnée

Le but du joueur est de minimiser son **regret** au bout de T tours :

$$Regret_T = \max_{i \in A} \sum_{t=1}^T \mathcal{R}_t(i) - \sum_{t=1}^T \mathcal{R}_t(i_t)$$

Description globale d'un bandit

Le problème des bandits est décrit comme un jeu où un **joueur** affronte l'**environnement**. À chaque tour t :

- le joueur **choisit une action** i_t parmi un ensemble d'actions A (machines, traitements, heuristiques, etc.)
- l'environnement **donne une récompense** $\mathcal{R}_t(i_t)$ au joueur pour l'action sélectionnée

Le but du joueur est de minimiser son **regret** au bout de T tours :

$$Regret_T = \max_{i \in A} \sum_{t=1}^T \mathcal{R}_t(i) - \sum_{t=1}^T \mathcal{R}_t(i_t)$$

Soient les **méthodes** suivantes pour décrire un bandit :

- $\text{INIT}()$: initialisation des structures du bandit
- $\text{SELECT}(t)$: retourne l'action i_t sélectionnée à l'essai t
- $\text{MAJ}(\mathcal{R}_t)$: mise-à-jour des récompenses pour la sélection i_t

Soient les méthodes suivantes pour décrire un bandit :

- $\text{INIT}()$: initialisation des structures du bandit
- $\text{SELECT}(t)$: retourne l'action i_t sélectionnée à l'essai t
- $\text{MAJ}(\mathcal{R}_t)$: mise-à-jour des récompenses pour la sélection i_t

Soient les méthodes suivantes pour décrire un bandit :

- $\text{INIT}()$: initialisation des structures du bandit
- $\text{SELECT}(t)$: retourne l'action i_t sélectionnée à l'essai t
- $\text{MAJ}(\mathcal{R}_t)$: mise-à-jour des récompenses pour la sélection i_t

Soient les méthodes suivantes pour décrire un bandit :

- $\text{INIT}()$: initialisation des structures du bandit
- $\text{SELECT}(t)$: retourne l'action i_t sélectionnée à l'essai t
- $\text{MAJ}(\mathcal{R}_t)$: mise-à-jour des récompenses pour la sélection i_t

Le bandit ε -Greedy

Bandit : ε -Greedy(K : actions)

1 Méthode **INIT()** :

2 **pour chaque** action $i \in [K]$ **faire**

3 $n_1(i) \leftarrow 0$

4 $\hat{\mathcal{R}}_1(i) \leftarrow 0$

5 Méthode **SELECT(t)** :

6 Choix du paramètre $\varepsilon_t \in [0, 1]$

7 **retourner** $i_t \in [K] \leftarrow \begin{cases} \text{aléatoirement,} & \text{avec la probabilité } \varepsilon_t \\ \text{maximisant } \hat{\mathcal{R}}_t(i_t), & \text{sinon} \end{cases}$

8 Méthode **MAJ(\mathcal{R}_t)** :

9 **pour chaque** action $i \in [K]$ **faire**

10 $n_{t+1}(i) \leftarrow n_t(i) + \mathbb{1}_{i=i_t}$

11 $\hat{\mathcal{R}}_{t+1}(i) \leftarrow \frac{n_t(i) \times \hat{\mathcal{R}}_t(i) + \mathcal{R}_t(i)}{n_{t+1}(i)}$

Le bandit ε -Greedy

Bandit : ε -Greedy(K : actions)

1 Méthode **INIT**() :

2 **pour chaque** action $i \in [K]$ **faire**
3 $n_1(i) \leftarrow 0$
4 $\hat{\mathcal{R}}_1(i) \leftarrow 0$

5 Méthode **SELECT**(t) :

6 Choix du paramètre $\varepsilon_t \in [0, 1]$
7 **retourner** $i_t \in [K] \leftarrow \begin{cases} \text{aléatoirement,} & \text{avec la probabilité } \varepsilon_t \\ \text{maximisant } \hat{\mathcal{R}}_t(i_t), & \text{sinon} \end{cases}$

8 Méthode **MAJ**(\mathcal{R}_t) :

9 **pour chaque** action $i \in [K]$ **faire**
10 $n_{t+1}(i) \leftarrow n_t(i) + \mathbb{1}_{i=i_t}$
11 $\hat{\mathcal{R}}_{t+1}(i) \leftarrow \frac{n_t(i) \times \hat{\mathcal{R}}_t(i) + \mathcal{R}_t(i)}{n_{t+1}(i)}$

Le bandit ε -Greedy

Bandit : ε -Greedy(K : actions)

1 Méthode **INIT**() :

2 **pour chaque** action $i \in [K]$ **faire**
3 $n_1(i) \leftarrow 0$
4 $\hat{\mathcal{R}}_1(i) \leftarrow 0$

5 Méthode **SELECT**(t) :

6 Choix du paramètre $\varepsilon_t \in [0, 1]$
7 **retourner** $i_t \in [K] \leftarrow \begin{cases} \text{aléatoirement,} & \text{avec la probabilité } \varepsilon_t \\ \text{maximisant } \hat{\mathcal{R}}_t(i_t), & \text{sinon} \end{cases}$

8 Méthode **MAJ**(\mathcal{R}_t) :

9 **pour chaque** action $i \in [K]$ **faire**
10 $n_{t+1}(i) \leftarrow n_t(i) + \mathbb{1}_{i=i_t}$
11 $\hat{\mathcal{R}}_{t+1}(i) \leftarrow \frac{n_t(i) \times \hat{\mathcal{R}}_t(i) + \mathcal{R}_t(i)}{n_{t+1}(i)}$

Le bandit ε -Greedy

Bandit : ε -Greedy(K : actions)

1 Méthode **INIT()** :

2 **pour chaque** action $i \in [K]$ **faire**
3 $n_1(i) \leftarrow 0$
4 $\hat{\mathcal{R}}_1(i) \leftarrow 0$

5 Méthode **SELECT(t)** :

6 Choix du paramètre $\varepsilon_t \in [0, 1]$
7 **retourner** $i_t \in [K] \leftarrow \begin{cases} \text{aléatoirement,} & \text{avec la probabilité } \varepsilon_t \\ \text{maximisant } \hat{\mathcal{R}}_t(i_t), & \text{sinon} \end{cases}$

8 Méthode **MAJ(\mathcal{R}_t)** :

9 **pour chaque** action $i \in [K]$ **faire**
10 $n_{t+1}(i) \leftarrow n_t(i) + \mathbb{1}_{i=i_t}$
11 $\hat{\mathcal{R}}_{t+1}(i) \leftarrow \frac{n_t(i) \times \hat{\mathcal{R}}_t(i) + \mathcal{R}_t(i)}{n_{t+1}(i)}$

Bandit : UCB1(K : actions)

```
1 Méthode INIT() :  
2   pour chaque action  $i \in [K]$  faire  
3      $n_1(i) \leftarrow 0$   
4      $\hat{\mathcal{R}}_1(i) \leftarrow 0$   
  
5 Méthode SELECT( $t$ ) :  
6   retourner  $i_t \in [K]$  maximisant  $\hat{\mathcal{R}}_t(i_t) + \sqrt{\frac{8 \ln(t)}{n_t(i_t)}}$   
  
7 Méthode MAJ( $\mathcal{R}_t$ ) :  
8   pour chaque action  $i \in [K]$  faire  
9      $n_{t+1}(i) \leftarrow n_t(i) + \mathbb{1}_{i=i_t}$   
10     $\hat{\mathcal{R}}_{t+1}(i) \leftarrow \frac{n_t(i) \times \hat{\mathcal{R}}_t(i) + \mathcal{R}_t(i)}{n_{t+1}(i)}$ 
```

Bandit : UCB1(K : actions)

1 Méthode **INIT**() :

2 **pour chaque** action $i \in [K]$ **faire**
3 $n_1(i) \leftarrow 0$
4 $\hat{\mathcal{R}}_1(i) \leftarrow 0$

5 Méthode **SELECT**(t) :

6 **retourner** $i_t \in [K]$ maximisant $\hat{\mathcal{R}}_t(i_t) + \sqrt{\frac{8 \ln(t)}{n_t(i_t)}}$

7 Méthode **MAJ**(\mathcal{R}_t) :

8 **pour chaque** action $i \in [K]$ **faire**
9 $n_{t+1}(i) \leftarrow n_t(i) + \mathbb{1}_{i=i_t}$
10 $\hat{\mathcal{R}}_{t+1}(i) \leftarrow \frac{n_t(i) \times \hat{\mathcal{R}}_t(i) + \mathcal{R}_t(i)}{n_{t+1}(i)}$

Bandit : UCB1(K : actions)

1 Méthode **INIT**() :

2 **pour chaque** *action* $i \in [K]$ **faire**
3 $n_1(i) \leftarrow 0$
4 $\hat{\mathcal{R}}_1(i) \leftarrow 0$

5 Méthode **SELECT**(t) :

6 **retourner** $i_t \in [K]$ maximisant $\hat{\mathcal{R}}_t(i_t) + \sqrt{\frac{8 \ln(t)}{n_t(i_t)}}$

7 Méthode **MAJ**(\mathcal{R}_t) :

8 **pour chaque** *action* $i \in [K]$ **faire**
9 $n_{t+1}(i) \leftarrow n_t(i) + \mathbb{1}_{i=i_t}$
10 $\hat{\mathcal{R}}_{t+1}(i) \leftarrow \frac{n_t(i) \times \hat{\mathcal{R}}_t(i) + \mathcal{R}_t(i)}{n_{t+1}(i)}$

Soit l'ensemble $\mathcal{B} = \{\varepsilon\text{-Greedy}, \text{UCB1}, \text{MOSS}, \text{TS}, \text{EXP3}, \text{UNI}\}$:

$\varepsilon\text{-Greedy}$: exploration fixée par ε

UCB1 : simple borne supérieur de confiance

MOSS : prend en compte le nombre de bras

TS : maintient une distribution beta

EXP3 : pondération exponentielle pour l'exploration et l'exploitation

UNI : choix aléatoire et uniforme

Soit l'ensemble $\mathcal{B} = \{\varepsilon\text{-Greedy}, \text{UCB1}, \text{MOSS}, \text{TS}, \text{EXP3}, \text{UNI}\}$:

$\varepsilon\text{-Greedy}$: exploration fixée par ε

UCB1 : simple borne supérieur de confiance

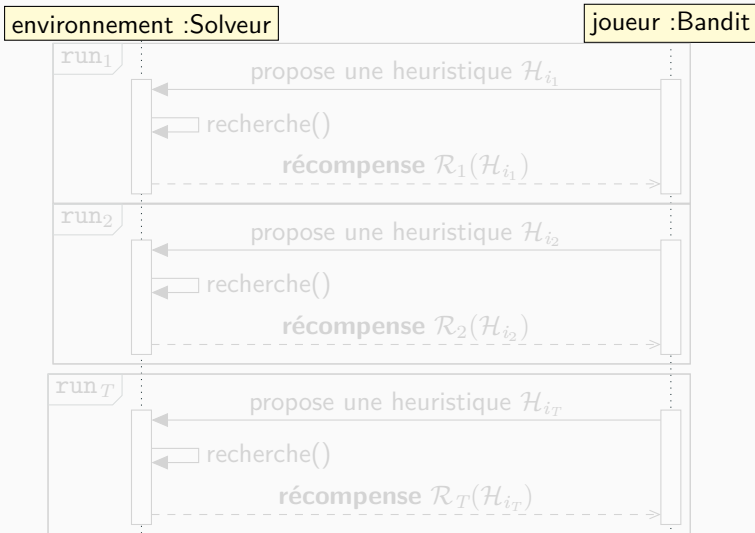
MOSS : prend en compte le nombre de bras

TS : maintient une distribution beta

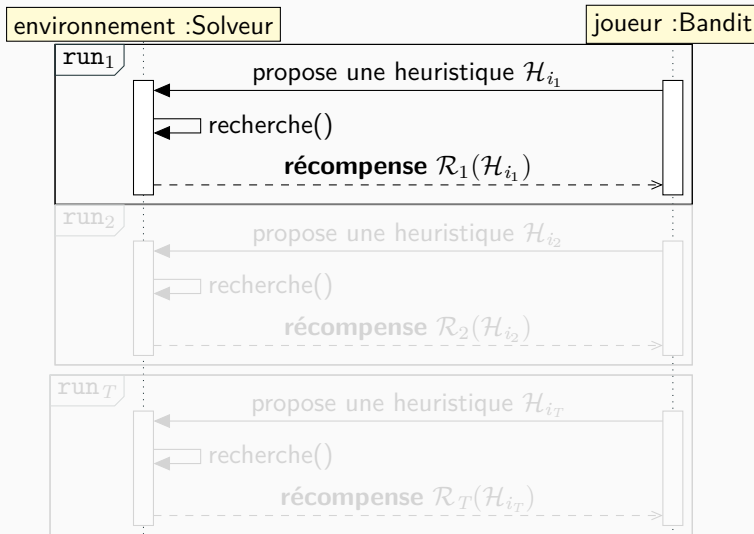
EXP3 : pondération exponentielle pour l'exploration et l'exploitation

UNI : choix aléatoire et uniforme

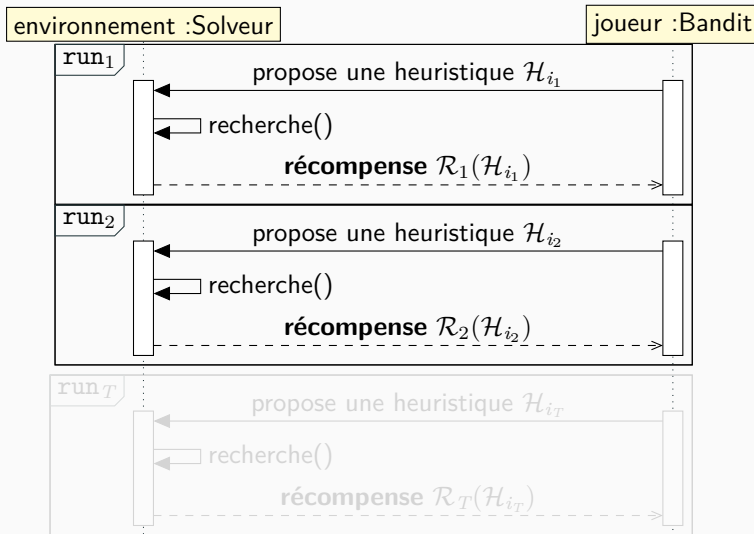
Proposition de lien entre bandit et heuristiques d'un solveur CSP



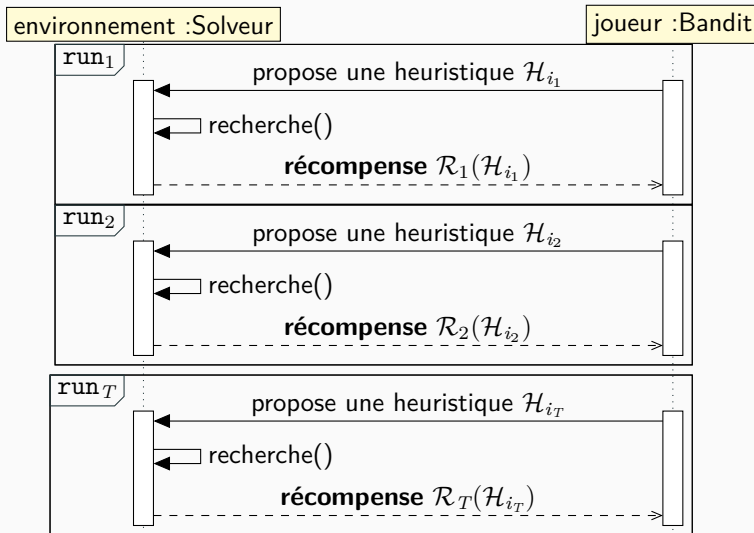
Proposition de lien entre bandit et heuristiques d'un solveur CSP



Proposition de lien entre bandit et heuristiques d'un solveur CSP



Proposition de lien entre bandit et heuristiques d'un solveur CSP



RES [6] : description algorithmique du framework basé sur les redémarrages

Heuristique : $\text{RES}(\mathcal{H} : \text{heuristiques}, \mathcal{B} : \text{politique de sélection}, \mathcal{R} : \text{fonction de récompense})$

```
1 Méthode INIT() :
2    $t \leftarrow 0$ 
3    $K \leftarrow |\mathcal{H}|$ 
4    $\text{INIT}_{\mathcal{B}}(K)$ 
5   pour chaque  $i \in [K]$  faire
6      $\text{INIT}_{\mathcal{H}_i}()$ 

7 Méthode AVANTRUN() :
8    $t \leftarrow t + 1$ 
9    $i_t \leftarrow \text{SELECT}_{\mathcal{B}}(t)$ 
10   $\text{AVANTRUN}_{\mathcal{H}_{i_t}}()$ 

11 Méthode APRÈSRUN() :
12   $\text{MAJ}_{\mathcal{B}}(\mathcal{R}_t)$ 

13 Méthode APRÈSCONFLIT(c) :
14   $\text{APRÈSCONFLIT}_{\mathcal{H}_{i_t}}()$ 

15 Méthode APRÈSASSIGNATION( $\mathcal{P}_{av}, \mathcal{P}_{ap}$ ) :
16   $\text{APRÈSASSIGNATION}_{\mathcal{H}_{i_t}}(\mathcal{P}_{av}, \mathcal{P}_{ap})$ 

17 Méthode SCORE(x) :
18  retourner  $\text{SCORE}_{\mathcal{H}_{i_t}}(x)$ 
```

RES [6] : description algorithmique du framework basé sur les redémarrages

Heuristique : $\text{RES}(\mathcal{H} : \text{heuristiques}, \mathcal{B} : \text{politique de sélection}, \mathcal{R} : \text{fonction de récompense})$

1 Méthode **INIT()** :
2 $t \leftarrow 0$
3 $K \leftarrow |\mathcal{H}|$
4 $\text{INIT}_{\mathcal{B}}(K)$
5 **pour chaque** $i \in [K]$ **faire**
6 $\text{INIT}_{\mathcal{H}_i}()$

7 Méthode **AVANTRUN()** :
8 $t \leftarrow t + 1$
9 $i_t \leftarrow \text{SELECT}_{\mathcal{B}}(t)$
10 $\text{AVANTRUN}_{\mathcal{H}_{i_t}}()$

11 Méthode **APRÈSRUN()** :
12 $\text{MAJ}_{\mathcal{B}}(\mathcal{R}_t)$

13 Méthode **APRÈSCONFLIT(c)** :
14 $\text{APRÈSCONFLIT}_{\mathcal{H}_{i_t}}()$

15 Méthode **APRÈSASSIGNATION**($\mathcal{P}_{av}, \mathcal{P}_{ap}$) :
16 $\text{APRÈSASSIGNATION}_{\mathcal{H}_{i_t}}(\mathcal{P}_{av}, \mathcal{P}_{ap})$

17 Méthode **SCORE(x)** :
18 **retourner** $\text{SCORE}_{\mathcal{H}_{i_t}}(x)$

RES [6] : description algorithmique du framework basé sur les redémarrages

Heuristique : RES(\mathcal{H} : heuristiques, \mathcal{B} : politique de sélection, \mathcal{R} : fonction de récompense)

1 Méthode INIT() :

```
2    $t \leftarrow 0$   
3    $K \leftarrow |\mathcal{H}|$   
4   INIT $_{\mathcal{B}}(K)$   
5   pour chaque  $i \in [K]$  faire  
6     INIT $_{\mathcal{H}_i}()$ 
```

7 Méthode AVANTRUN() :

```
8    $t \leftarrow t + 1$   
9    $i_t \leftarrow \text{SELECT}_{\mathcal{B}}(t)$   
10  AVANTRUN $_{\mathcal{H}_{i_t}}()$ 
```

11 Méthode APRÈSRUN() :

```
12  MAJ $_{\mathcal{B}}(\mathcal{R}_t)$ 
```

13 Méthode APRÈSCONFLIT(c) :

```
14  APRÈSCONFLIT $_{\mathcal{H}_{i_t}}()$ 
```

15 Méthode APRÈSASSIGNATION($\mathcal{P}_{av}, \mathcal{P}_{ap}$) :

```
16  APRÈSASSIGNATION $_{\mathcal{H}_{i_t}}(\mathcal{P}_{av}, \mathcal{P}_{ap})$ 
```

17 Méthode SCORE(x) :

```
18  retourner SCORE $_{\mathcal{H}_{i_t}}(x)$ 
```

RES [6] : description algorithmique du framework basé sur les redémarrages

Heuristique : $\text{RES}(\mathcal{H} : \text{heuristiques}, \mathcal{B} : \text{politique de sélection}, \mathcal{R} : \text{fonction de récompense})$

```
1 Méthode INIT() :  
2    $t \leftarrow 0$   
3    $K \leftarrow |\mathcal{H}|$   
4    $\text{INIT}_{\mathcal{B}}(K)$   
5   pour chaque  $i \in [K]$  faire  
6      $\text{INIT}_{\mathcal{H}_i}()$ 
```

```
7 Méthode AVANTRUN() :  
8    $t \leftarrow t + 1$   
9    $i_t \leftarrow \text{SELECT}_{\mathcal{B}}(t)$   
10   $\text{AVANTRUN}_{\mathcal{H}_{i_t}}()$ 
```

```
11 Méthode APRÈSRUN() :  
12    $\text{MAJ}_{\mathcal{B}}(\mathcal{R}_t)$ 
```

```
13 Méthode APRÈSCONFLIT( $c$ ) :  
14    $\text{APRÈSCONFLIT}_{\mathcal{H}_{i_t}}()$ 
```

```
15 Méthode APRÈSASSIGNATION( $\mathcal{P}_{av}, \mathcal{P}_{ap}$ ) :  
16    $\text{APRÈSASSIGNATION}_{\mathcal{H}_{i_t}}(\mathcal{P}_{av}, \mathcal{P}_{ap})$ 
```

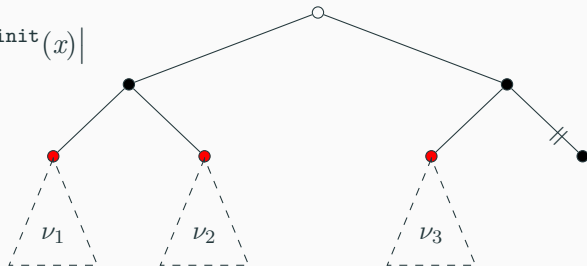
```
17 Méthode SCORE( $x$ ) :  
18   retourner  $\text{SCORE}_{\mathcal{H}_{i_t}}(x)$ 
```

Trois fonctions de récompense : « Normalized Pruned Tree Sizes » [6]

npts est basée sur la taille des arbres élagués durant un run t :

$$\text{pts}(\mathcal{T}_t) = \sum_{\Delta \in \Sigma(\mathcal{T}_t)} \prod_{x \in \mathcal{X} - \text{vars}(\Delta)} |\text{dom}^{\text{init}}(x)|$$

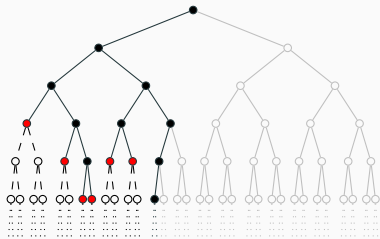
$$\mathcal{R}_t^{\text{npts}} = \frac{\log(\text{pts}(\mathcal{T}_t))}{\log(\prod_{x \in \mathcal{X}} |\text{dom}^{\text{init}}(x)|)}$$



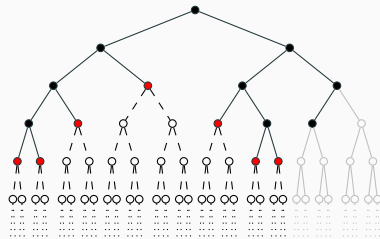
Trois fonctions de récompense : « Explored Sub-Tree » [5]

esb est basée sur la proportion de nœuds visités par rapport au produit Cartésien des domaines des variables sélectionnées pendant le run :

$$\mathcal{R}_t^{\text{esb}}(i_t) = \frac{\log(\text{size}(\mathcal{T}_t))}{\log\left(\prod_{x \in \text{vars}(\mathcal{T}_t)} |\text{dom}(x)|\right)}$$



(a) $\text{esb}(\mathcal{T}_{\text{left}}) = \frac{\log(16)}{\log(2^6)} \approx 0.67$



(b) $\text{esb}(\mathcal{T}_{\text{right}}) = \frac{\log(16)}{\log(2^5)} = 0.8$

Trois fonctions de récompense : « Average of Unassigned Variable Ratios » [2]

auvr donne la moyenne de la profondeur des conflits au cours d'une exécution :

$$\mathcal{R}_t^{\text{auvr}}(i_t) = \frac{1}{|\text{cft}(\mathcal{T}_t)|} \sum_{\nu \in \text{cft}(\mathcal{T}_t)} \frac{|\text{fut}(\nu)|}{|\mathcal{X}|}$$

Étude similaire : framework NOD [8]

Heuristique : NOD(\mathcal{H} : heuristiques, \mathcal{B} : politique de sélection)

```
1 Méthode INIT() :
2    $d \leftarrow 0$ 
3    $K \leftarrow |\mathcal{H}|$ 
4   INIT $_{\mathcal{B}}$ ( $K$ )
5   pour chaque  $i \in [K]$  faire
6     INIT $_{\mathcal{H}_i}$ ()

7 Méthode AVANTRUN() :
8   pour chaque  $i \in [K]$  faire
9     AVANTRUN $_{\mathcal{H}_i}$ ()

10 Méthode AVANTASSIGNATION() :
11    $d \leftarrow d + 1$ 
12    $i_d \leftarrow \text{SELECT}_{\mathcal{B}}(d)$ 

13 Méthode APRÈSASSIGNATION( $\mathcal{P}_{av}$ ,  $\mathcal{P}_{ap}$ ) :
14   pour chaque  $i \in [K]$  faire
15     APRÈSASSIGNATION $_{\mathcal{H}_i}(\mathcal{P}_{av}, \mathcal{P}_{ap})$ 

16 Méthode APRÈSCONFLIT( $c$ ) :
17   pour chaque  $i \in [K]$  faire
18     APRÈSCONFLIT $_{\mathcal{H}_i}()$ 

19 Méthode BACKTRACK( $d'$ ) :
20   MAJ $_{\mathcal{B}}(\mathcal{R}_{d'}^{\text{size}})$  //  $\mathcal{R}_d^{\text{size}} = 1 - \frac{\text{size}(\mathcal{T}_{\nu_d})}{\max_{d' \in [d]} \text{size}(\mathcal{T}_{\nu_{d'}})}$ 

21 Méthode SCORE( $x$ ) :
22   retourner SCORE $_{\mathcal{H}_{i_d}}(x)$ 
```

Étude similaire : framework NOD [8]

Heuristique : NOD(\mathcal{H} : heuristiques, \mathcal{B} : politique de sélection)

```
1 Méthode INIT() :
2    $d \leftarrow 0$ 
3    $K \leftarrow |\mathcal{H}|$ 
4   INIT $\mathcal{B}$ ( $K$ )
5   pour chaque  $i \in [K]$  faire
6     INIT $\mathcal{H}_i$ ()

7 Méthode AVANTRUN() :
8   pour chaque  $i \in [K]$  faire
9     AVANTRUN $\mathcal{H}_i$ ()

10 Méthode AVANTASSIGNATION() :
11    $d \leftarrow d + 1$ 
12    $i_d \leftarrow \text{SELECT}_{\mathcal{B}}(d)$ 

13 Méthode APRÈSASSIGNATION( $\mathcal{P}_{av}, \mathcal{P}_{ap}$ ) :
14   pour chaque  $i \in [K]$  faire
15     APRÈSASSIGNATION $\mathcal{H}_i(\mathcal{P}_{av}, \mathcal{P}_{ap})$ 

16 Méthode APRÈSCONFLIT( $c$ ) :
17   pour chaque  $i \in [K]$  faire
18     APRÈSCONFLIT $\mathcal{H}_i()$ 

19 Méthode BACKTRACK( $d'$ ) :
20    $\text{MAJ}_{\mathcal{B}}(\mathcal{R}_{d'}^{\text{size}})$  //  $\mathcal{R}_d^{\text{size}} = 1 - \frac{\text{size}(\mathcal{T}_{\nu_d})}{\max_{d' \in [d]} \text{size}(\mathcal{T}_{\nu_{d'}})}$ 

21 Méthode SCORE( $x$ ) :
22   retourner SCORE $\mathcal{H}_{i_d}(x)$ 
```

$\Psi_{\text{NOD,RES}} =$

- Ψ_{base}

Heuristiques de base

- $\left\{ \text{ACE}(\text{NOD}(\mathcal{H}, \mathcal{B})) \mid \mathcal{B} \in \{\varepsilon\text{-Greedy}, \text{UCB1}, \text{MOSS}, \text{TS}, \text{UNI}\} \right\}$

NOD

- $\left\{ \text{ACE}(\text{RES}(\mathcal{H}, \mathcal{B}, \mathcal{R})) \mid \begin{array}{l} \mathcal{B} \in \{\varepsilon\text{-Greedy}, \text{UCB1}, \text{MOSS}, \text{TS}, \text{EXP3}, \text{UNI}\} \\ \mathcal{R} \in \{\text{auvr}, \text{esb}, \text{npts}\} \end{array} \right\}$

RES

$\Psi_{\text{NOD,RES}} =$

- Ψ_{base}

Heuristiques de base

- $\left\{ \text{ACE}(\text{NOD}(\mathcal{H}, \mathcal{B})) \mid \mathcal{B} \in \{\varepsilon\text{-Greedy}, \text{UCB1}, \text{MOSS}, \text{TS}, \text{UNI}\} \right\}$ NOD

- $\left\{ \text{ACE}(\text{RES}(\mathcal{H}, \mathcal{B}, \mathcal{R})) \mid \begin{array}{l} \mathcal{B} \in \{\varepsilon\text{-Greedy}, \text{UCB1}, \text{MOSS}, \text{TS}, \text{EXP3}, \text{UNI}\} \\ \mathcal{R} \in \{\text{auvr}, \text{esb}, \text{npts}\} \end{array} \right\}$ RES

$$\Psi_{\text{NOD,RES}} =$$

- Ψ_{base}

Heuristiques de base

- $\left\{ \text{ACE}(\text{NOD}(\mathcal{H}, \mathcal{B})) \mid \mathcal{B} \in \{\varepsilon\text{-Greedy}, \text{UCB1}, \text{MOSS}, \text{TS}, \text{UNI}\} \right\}$

NOD

- $\left\{ \text{ACE}(\text{RES}(\mathcal{H}, \mathcal{B}, \mathcal{R})) \mid \begin{array}{l} \mathcal{B} \in \{\varepsilon\text{-Greedy}, \text{UCB1}, \text{MOSS}, \text{TS}, \text{EXP3}, \text{UNI}\} \\ \mathcal{R} \in \{\text{auvr}, \text{esb}, \text{npts}\} \end{array} \right\}$

RES

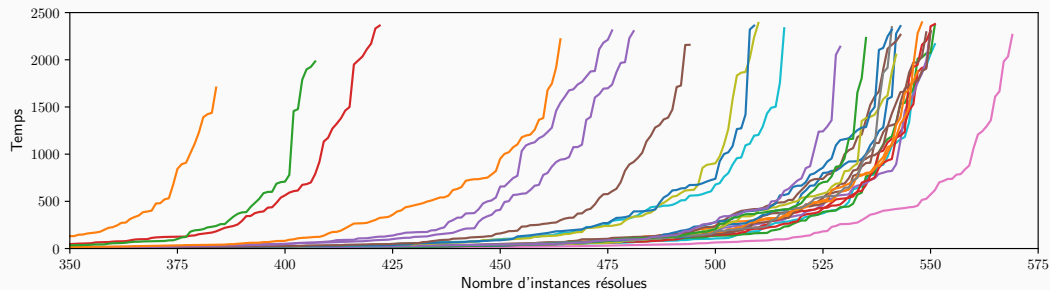
$$\Psi_{\text{NOD,RES}} =$$

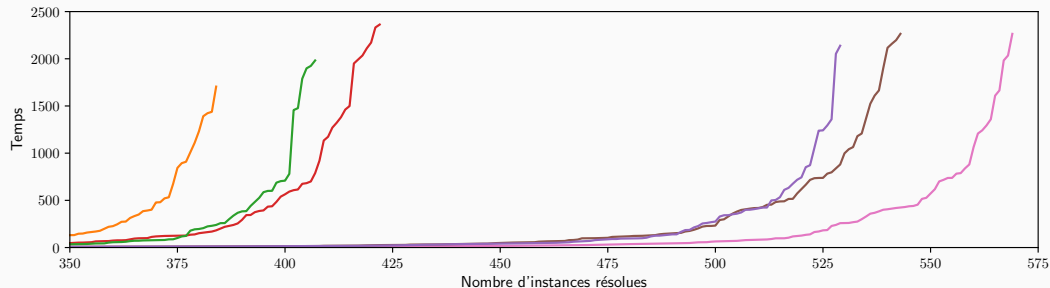
$$\blacksquare \Psi_{\text{base}}$$

Heuristiques de base

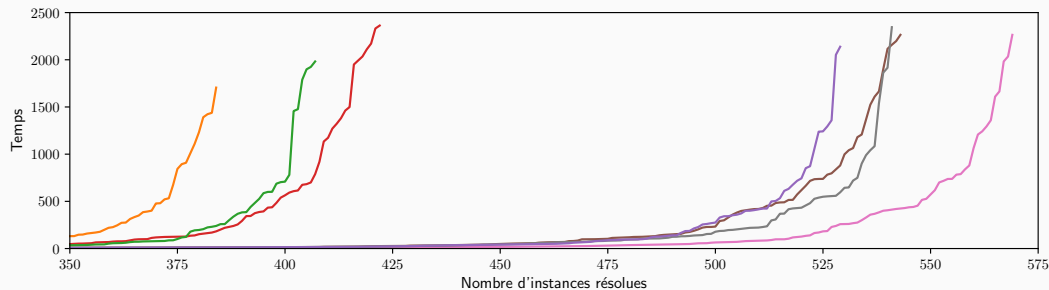
$$\blacksquare \left\{ \text{ACE}(\text{NOD}(\mathcal{H}, \mathcal{B})) \mid \mathcal{B} \in \{\varepsilon\text{-Greedy}, \text{UCB1}, \text{MOSS}, \text{TS}, \text{UNI}\} \right\} \quad \text{NOD}$$

$$\blacksquare \left\{ \text{ACE}(\text{RES}(\mathcal{H}, \mathcal{B}, \mathcal{R})) \mid \begin{array}{l} \mathcal{B} \in \{\varepsilon\text{-Greedy}, \text{UCB1}, \text{MOSS}, \text{TS}, \text{EXP3}, \text{UNI}\} \\ \mathcal{R} \in \{\text{auvr}, \text{esb}, \text{npts}\} \end{array} \right\} \quad \text{RES}$$

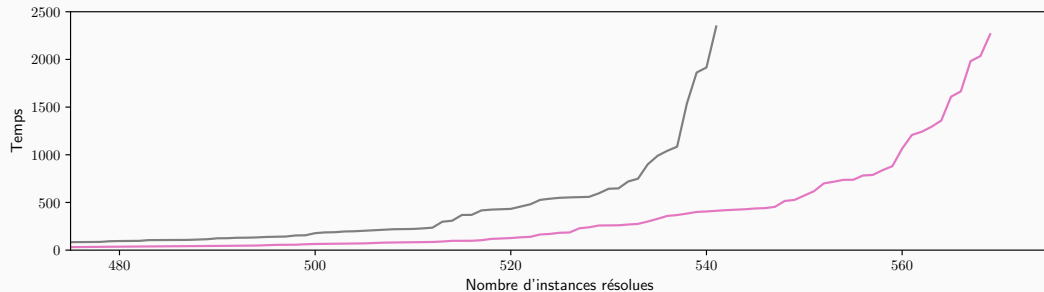




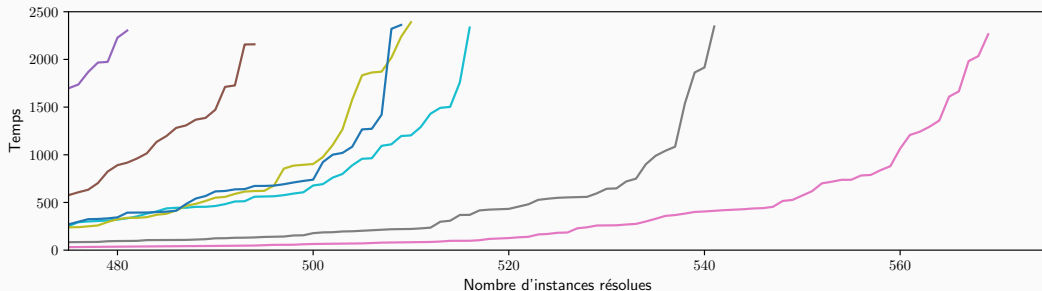
	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
VBS($\mathcal{H}^{\text{base}}$)	569	61,955	579
wdeg^{ca.cd}	543	133,146	579
dom/wdeg^{chs}	529	149,819	579
abs	422	418,084	579
dom/ddeg	407	437,055	579
ibs	384	491,669	579



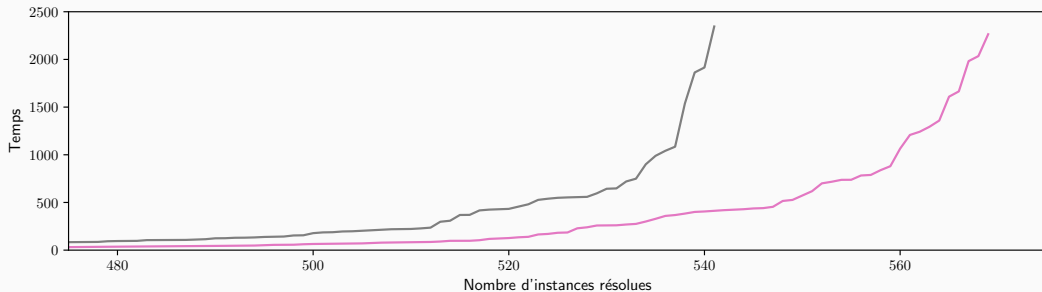
	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
VBS($\mathcal{H}^{\text{base}}$)	569	61,955	579
wdeg ^{ca.cd}	543	133,146	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UNI}}$)	541	123,503	579
dom/wdeg ^{chs}	529	149,819	579
abs	422	418,084	579
dom/ddeg	407	437,055	579
ibs	384	491,669	579



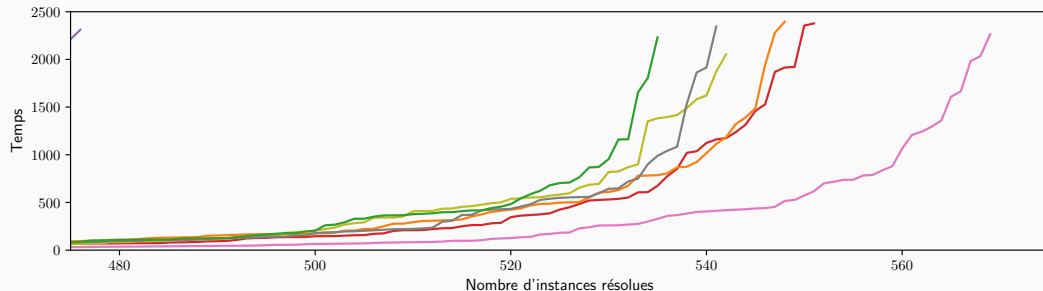
	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
$\text{VBS}(\mathcal{H}^{\text{base}})$	569	61,955	579
$\text{RES}(\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UNI}})$	541	123,503	579



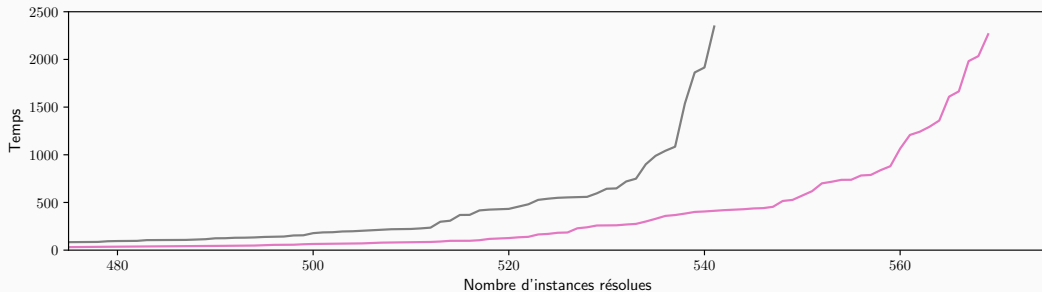
	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
VBS($\mathcal{H}^{\text{base}}$)	569	61,955	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UNI}}$)	541	123,503	579
NOD($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{e-Greedy}}$)	516	190,617	579
NOD($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{MOSS}}$)	510	203,713	579
NOD($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UCB1}}$)	509	202,562	579
NOD($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{TS}}$)	494	240,985	579
NOD($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UNI}}$)	481	281,166	579



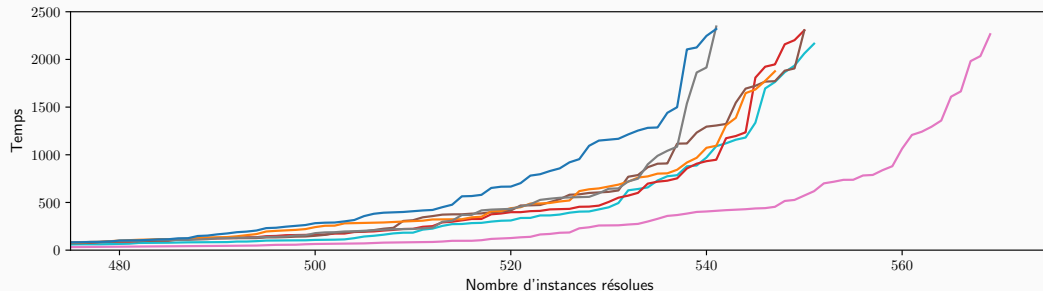
	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
$\text{VBS}(\mathcal{H}^{\text{base}})$	569	61,955	579
$\text{RES}(\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UNI}})$	541	123,503	579



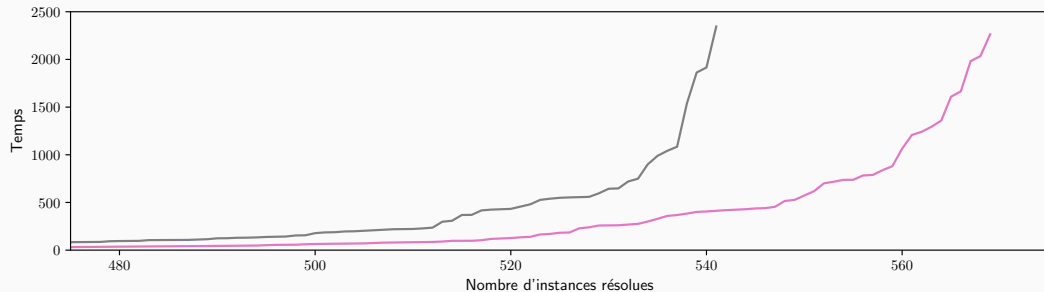
	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
VBS($\mathcal{H}^{\text{base}}$)	569	61,955	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UCB1}}, \mathcal{R}^{\text{auvr}}$)	551	108,580	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{MOSS}}, \mathcal{R}^{\text{auvr}}$)	548	114,581	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{TS}}, \mathcal{R}^{\text{auvr}}$)	542	126,516	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UNI}}$)	541	123,503	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{e-Greedy}}, \mathcal{R}^{\text{auvr}}$)	535	135,861	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{EXP3}}, \mathcal{R}^{\text{auvr}}$)	476	297,679	579



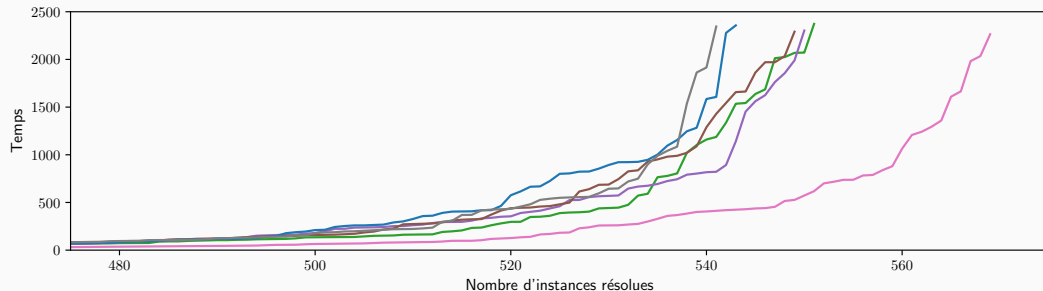
	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
$\text{VBS}(\mathcal{H}^{\text{base}})$	569	61,955	579
$\text{RES}(\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UNI}})$	541	123,503	579



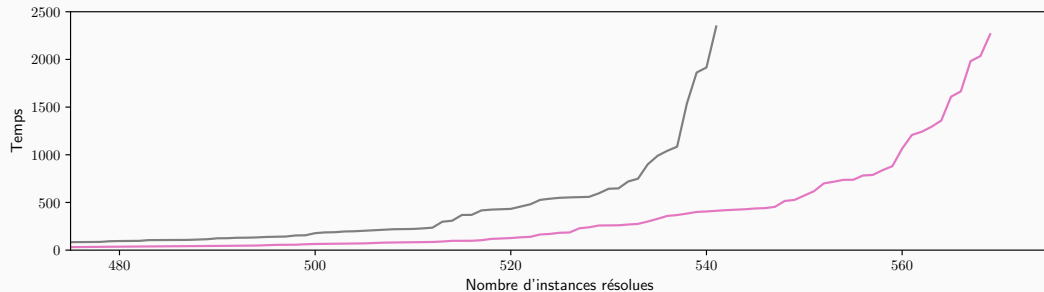
	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
VBS($\mathcal{H}^{\text{base}}$)	569	61,955	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{EXP3}}, \mathcal{R}^{\text{esb}}$)	551	106,237	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{TS}}, \mathcal{R}^{\text{esb}}$)	550	110,488	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UCB1}}, \mathcal{R}^{\text{esb}}$)	550	114,934	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{MOSS}}, \mathcal{R}^{\text{esb}}$)	547	115,487	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UNI}}$)	541	123,503	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{Greedy}}, \mathcal{R}^{\text{esb}}$)	541	135,424	579



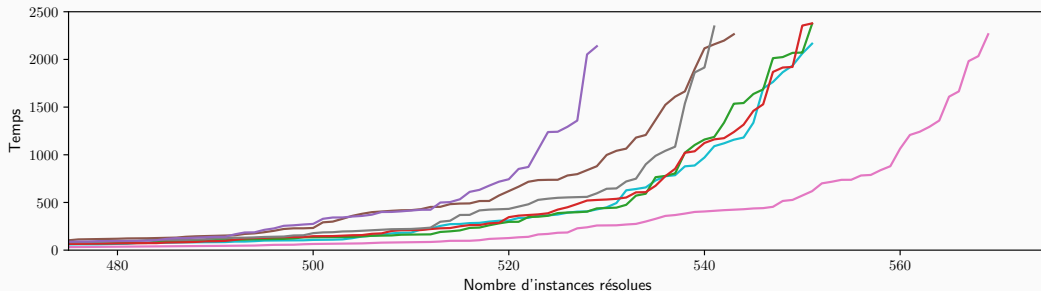
	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
$\text{VBS}(\mathcal{H}^{\text{base}})$	569	61,955	579
$\text{RES}(\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UNI}})$	541	123,503	579



	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
VBS($\mathcal{H}^{\text{base}}$)	569	61,955	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UCB1}}, \mathcal{R}^{\text{npts}}$)	551	108,391	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{TS}}, \mathcal{R}^{\text{npts}}$)	550	109,398	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{MOSS}}, \mathcal{R}^{\text{npts}}$)	549	115,772	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{Greedy}}, \mathcal{R}^{\text{npts}}$)	543	125,181	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UNI}}$)	541	123,503	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{EXP3}}, \mathcal{R}^{\text{npts}}$)	464	318,980	579



	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
$\text{VBS}(\mathcal{H}^{\text{base}})$	569	61,955	579
$\text{RES}(\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UNI}})$	541	123,503	579



	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
VBS($\mathcal{H}^{\text{base}}$)	569	61,955	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{EXP3}}, \mathcal{R}^{\text{esb}}$)	551	106,237	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UCB1}}, \mathcal{R}^{\text{npts}}$)	551	108,391	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UCB1}}, \mathcal{R}^{\text{auvr}}$)	551	108,580	579
wdeg ^{ca.cd}	543	133,146	579
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UNI}}$)	541	123,503	579
dom/wdeg ^{chs}	529	149,819	579

DÉFI 1 : la **stratégie uniforme** est **plus efficace** que l'état de l'art.

DÉFI 2 : des couples **bandit/fonction de récompense** dépassent la **stratégie uniforme** (y compris dans cette étude sur les heuristiques de solveur SAT [1]).

En chemin vers le DÉFI 3 (pistes d'amélioration) :

- taille et nombre de runs **irréguliers** : comparaison non-équitable des bras et convergence compromise
- distribution des récompenses :
 - **non-stationnaires** (l'environnement change avec le temps) → phénomène réduit avec une phase d'entraînement [2]
 - variance de la distribution et sensibilité du bandit

Critiques du modèle actuel

DÉFI 1 : la **stratégie uniforme** est **plus efficace** que l'état de l'art.

DÉFI 2 : des couples **bandit/fonction de récompense** dépassent la **stratégie uniforme** (y compris dans cette étude sur les heuristiques de solveur SAT [1]).

En chemin vers le DÉFI 3 (pistes d'amélioration) :

- taille et nombre de runs **irréguliers** : comparaison non-équitable des bras et convergence compromise
- distribution des récompenses :
 - **non-stationnaires** (l'environnement change avec le temps) → phénomène réduit avec une phase d'entraînement [2]
 - variance de la distribution et sensibilité du bandit

DÉFI 1 : la **stratégie uniforme** est **plus efficace** que l'état de l'art.

DÉFI 2 : des couples **bandit/fonction de récompense** dépassent la **stratégie uniforme** (y compris dans cette étude sur les heuristiques de solveur SAT [1]).

En chemin vers le DÉFI 3 (pistes d'amélioration) :

- taille et nombre de runs **irréguliers** : comparaison non-équitable des bras et convergence compromise
- distribution des récompenses :
 - **non-stationnaires** (l'environnement change avec le temps) → phénomène réduit avec une phase d'entraînement [2]
 - variance de la distribution et sensibilité du bandit

DÉFI 1 : la **stratégie uniforme** est **plus efficace** que l'état de l'art.

DÉFI 2 : des couples **bandit/fonction de récompense** dépassent la **stratégie uniforme** (y compris dans cette étude sur les heuristiques de solveur SAT [1]).

En chemin vers le DÉFI 3 (pistes d'amélioration) :

- taille et nombre de runs **irréguliers** : comparaison non-équitable des bras et convergence compromise
- distribution des récompenses :
 - **non-stationnaires** (l'environnement change avec le temps) → phénomène réduit avec une phase d'entraînement [2]
 - variance de la distribution et sensibilité du bandit

Critiques du modèle actuel

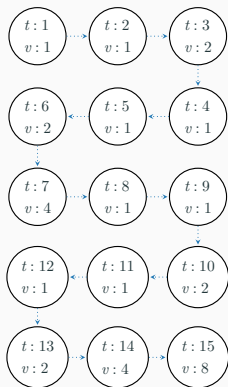
DÉFI 1 : la **stratégie uniforme** est **plus efficace** que l'état de l'art.

DÉFI 2 : des couples **bandit/fonction de récompense** dépassent la **stratégie uniforme** (y compris dans cette étude sur les heuristiques de solveur SAT [1]).

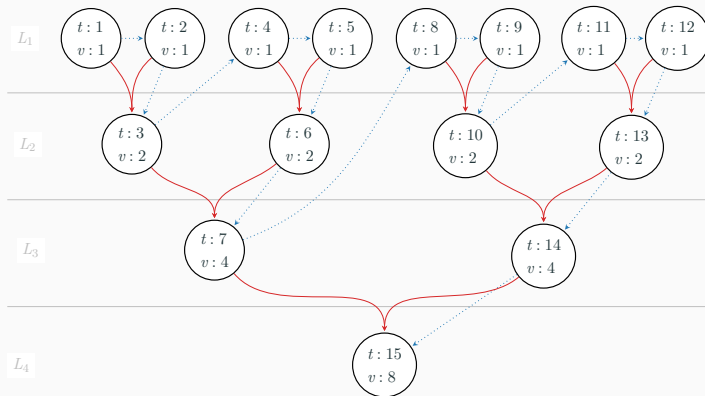
En chemin vers le DÉFI 3 (pistes d'amélioration) :

- taille et nombre de runs **irréguliers** : comparaison non-équitable des bras et convergence compromise
- distribution des récompenses :
 - **non-stationnaires** (l'environnement change avec le temps) → phénomène réduit avec une phase d'entraînement [2]
 - variance de la distribution et sensibilité du bandit

De Luby séquentiel à Luby arborescent

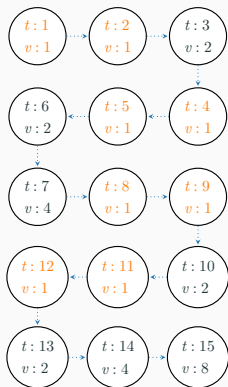


(a) Vue séquentielle

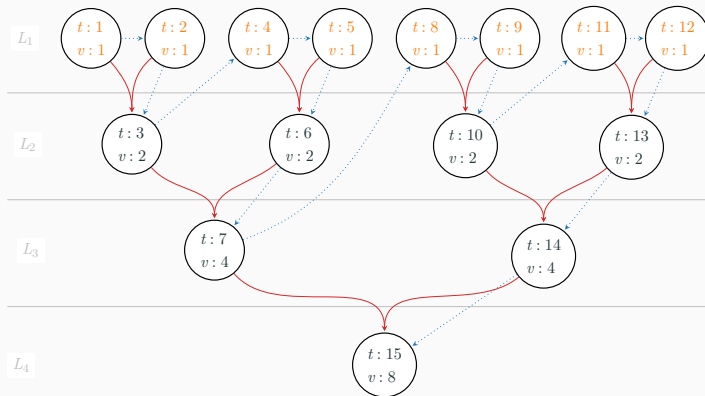


(b) Vue arborescente

De Luby séquentiel à Luby arborescent

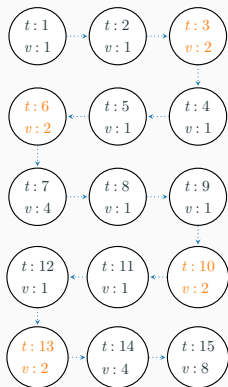


(a) Vue séquentielle

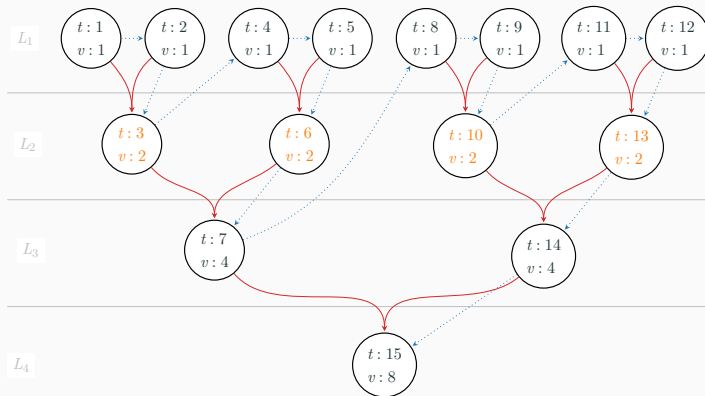


(b) Vue arborescente

De Luby séquentiel à Luby arborescent

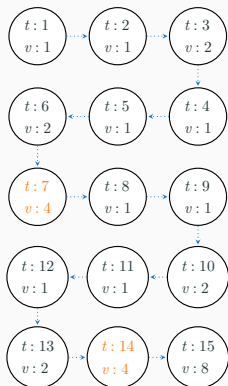


(a) Vue séquentielle

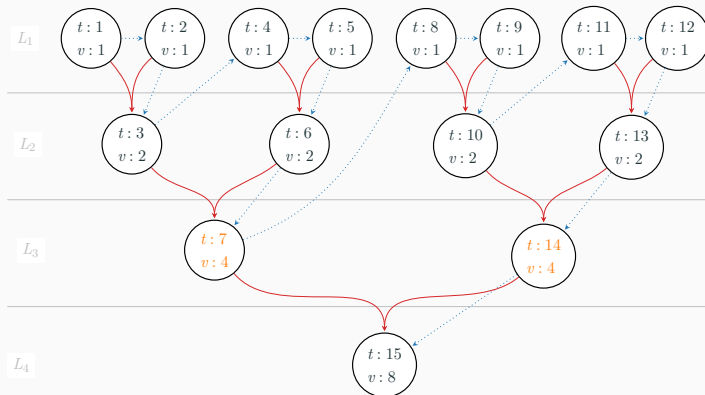


(b) Vue arborescente

De Luby séquentiel à Luby arborescent

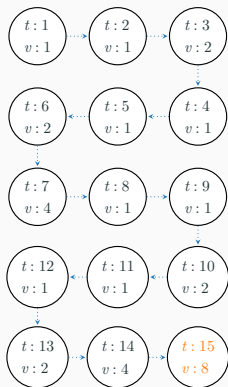


(a) Vue séquentielle

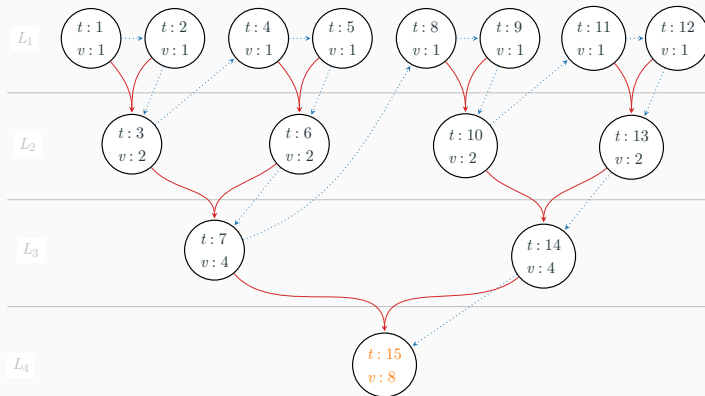


(b) Vue arborescente

De Luby séquentiel à Luby arborescent



(a) Vue séquentielle



(b) Vue arborescente

Un bandit adapté aux solveurs CSP : « Adaptive Single Tournament » [4]

Bandit : $\text{ST}(K : \text{actions}, d : \text{profondeur de l'arbre}, M : \text{échantillonnage}, \text{duel} : \text{opérateur gagnant})$

1 Méthode $\text{INIT}()$:

```
2   pour chaque  $m \in 1..2^{d-2}$  faire
3      $(C_1^{2m-1}, C_1^{2m}) \leftarrow$  choix aléatoire de  $(i, j) \in [K] \times [K] | i \neq j$ 
```

4 Méthode $\text{SELECT}(t)$:

```
5    $l, c \leftarrow \text{index\_ST}(\lceil t/M \rceil)$ 
6    $i_t \leftarrow C_l^c$ 
7   retourner  $i_t$ 
```

8 Méthode $\text{MAJ}(\mathcal{R}_t)$:

```
9    $l, c \leftarrow \text{index\_ST}(\lceil t/M \rceil)$ 
10  ajout de  $\mathcal{R}_t(i_t)$  dans  $R_l^c$ 
11  si  $|R_l^c| = M$  &  $c$  est pair alors
12     $C_{l+1}^{(c-1)/2} \leftarrow \text{duel}(C_l^c, C_l^{c'})$  //  $\max_{r \in R_l^c}(r) > \max_{r' \in R_l^{c'}}(r') ? R_l^c : R_l^{c'}$ 
```

Un bandit adapté aux solveurs CSP : « Adaptive Single Tournament » [4]

Bandit : $\text{ST}(K : \text{actions}, d : \text{profondeur de l'arbre}, M : \text{échantillonnage}, \text{duel} : \text{opérateur gagnant})$

1 **Méthode** $\text{INIT}()$:

```
2   pour chaque  $m \in 1..2^{d-2}$  faire
3      $(C_1^{2m-1}, C_1^{2m}) \leftarrow$  choix aléatoire de  $(i, j) \in [K] \times [K] | i \neq j$ 
```

4 **Méthode** $\text{SELECT}(t)$:

```
5    $l, c \leftarrow \text{index\_ST}(\lceil t/M \rceil)$ 
6    $i_t \leftarrow C_l^c$ 
7   retourner  $i_t$ 
```

8 **Méthode** $\text{MAJ}(\mathcal{R}_t)$:

```
9    $l, c \leftarrow \text{index\_ST}(\lceil t/M \rceil)$ 
10  ajout de  $\mathcal{R}_t(i_t)$  dans  $R_l^c$ 
11  si  $|R_l^c| = M$  &  $c$  est pair alors
12     $C_{l+1}^{(c-1)/2} \leftarrow \text{duel}(C_l^c, C_l^{c'})$  //  $\max_{r \in R_l^c}(r) > \max_{r' \in R_l^{c'}}(r') ? R_l^c : R_l^{c'}$ 
```

Un bandit adapté aux solveurs CSP : « Adaptive Single Tournament » [4]

Bandit : $\text{ST}(K : \text{actions}, d : \text{profondeur de l'arbre}, M : \text{échantillonnage}, \text{duel} : \text{opérateur gagnant})$

```
1 Méthode INIT() :  
2   | pour chaque  $m \in 1..2^{d-2}$  faire  
3   |   |  $(C_1^{2m-1}, C_1^{2m}) \leftarrow$  choix aléatoire de  $(i, j) \in [K] \times [K] | i \neq j$   
  
4 Méthode SELECT( $t$ ) :  
5   |  $l, c \leftarrow \text{index\_ST}(\lceil t/M \rceil)$   
6   |  $i_t \leftarrow C_l^c$   
7   | retourner  $i_t$   
  
8 Méthode MAJ( $\mathcal{R}_t$ ) :  
9   |  $l, c \leftarrow \text{index\_ST}(\lceil t/M \rceil)$   
10  | ajout de  $\mathcal{R}_t(i_t)$  dans  $R_l^c$   
11  | si  $|R_l^c| = M$  &  $c$  est pair alors  
12  |   |  $C_{l+1}^{(c-1)/2} \leftarrow \text{duel}(C_l^c, C_l^{c'})$  //  $\max_{r \in R_l^c}(r) > \max_{r' \in R_l^{c'}}(r') ? R_l^c : R_l^{c'}$ 
```

Un bandit adapté aux solveurs CSP : « Adaptive Single Tournament » [4]

Bandit : $\text{ST}(K : \text{actions}, d : \text{profondeur de l'arbre}, M : \text{échantillonnage}, \text{duel} : \text{opérateur gagnant})$

1 Méthode $\text{INIT}()$:

```
2   pour chaque  $m \in 1..2^{d-2}$  faire
3      $(C_1^{2m-1}, C_1^{2m}) \leftarrow \text{choix aléatoire de } (i, j) \in [K] \times [K] | i \neq j$ 
```

4 Méthode $\text{SELECT}(t)$:

```
5    $l, c \leftarrow \text{index\_ST}(\lceil t/M \rceil)$ 
6    $i_t \leftarrow C_l^c$ 
7   retourner  $i_t$ 
```

8 Méthode $\text{MAJ}(\mathcal{R}_t)$:

```
9    $l, c \leftarrow \text{index\_ST}(\lceil t/M \rceil)$ 
10  ajout de  $\mathcal{R}_t(i_t)$  dans  $R_l^c$ 
11  si  $|R_l^c| = M$  &  $c$  est pair alors
12     $C_{l+1}^{(c-1)/2} \leftarrow \text{duel}(C_l^c, C_l^{c'})$  //  $\max_{r \in R_l^c}(r) > \max_{r' \in R_l^{c'}}(r') ? R_l^c : R_l^{c'}$ 
```

Un bandit adapté aux solveurs CSP : « Adaptive Single Tournament » [4]

Bandit : ST(K : actions, d : profondeur de l'arbre, M : échantillonnage, duel : opérateur gagnant)

1 Méthode **INIT**() :

```
2   pour chaque  $m \in 1..2^{d-2}$  faire
3      $(C_1^{2m-1}, C_1^{2m}) \leftarrow$  choix aléatoire de  $(i, j) \in [K] \times [K] | i \neq j$ 
```

4 Méthode **SELECT**(t) :

```
5    $l, c \leftarrow \text{index\_ST}(\lceil t/M \rceil)$ 
6    $i_t \leftarrow C_l^c$ 
7   retourner  $i_t$ 
```

8 Méthode **MAJ**(\mathcal{R}_t) :

```
9    $l, c \leftarrow \text{index\_ST}(\lceil t/M \rceil)$ 
10  ajout de  $\mathcal{R}_t(i_t)$  dans  $R_l^c$ 
11  si  $|R_l^c| = M$  &  $c$  est pair alors
12     $C_{l+1}^{(c-1)/2} \leftarrow \text{duel}(C_l^c, C_l^{c'})$  //  $\max_{r \in R_l^c}(r) > \max_{r' \in R_l^{c'}}(r') ? R_l^c : R_l^{c'}$ 
```

$\Psi_{\text{ST}} =$

- Ψ_{base}

Heuristiques de base

- Ψ_{RES}

RES

- $\left\{ \text{ACE}(\text{RES}(\mathcal{H}, \mathcal{B}_M^{\text{ST}}, \mathcal{R})) \mid \begin{array}{l} M \in \{1, 2, 4, 8, 16, 32\} \\ \mathcal{R} \in \{\text{auvr}, \text{esb}, \text{npts}\} \end{array} \right\}$

ST

$\Psi_{\text{ST}} =$

- Ψ_{base}

Heuristiques de base

- Ψ_{RES}

RES

- $\left\{ \text{ACE}(\text{RES}(\mathcal{H}, \mathcal{B}_M^{\text{ST}}, \mathcal{R})) \mid \begin{array}{l} M \in \{1, 2, 4, 8, 16, 32\} \\ \mathcal{R} \in \{\text{auvr}, \text{esb}, \text{npts}\} \end{array} \right\}$

ST

$\Psi_{ST} =$

- Ψ_{base}

Heuristiques de base

- Ψ_{RES}

RES

- $\left\{ \text{ACE}(\text{RES}(\mathcal{H}, \mathcal{B}_M^{ST}, \mathcal{R})) \mid \begin{array}{l} M \in \{1, 2, 4, 8, 16, 32\} \\ \mathcal{R} \in \{\text{auvr}, \text{esb}, \text{npts}\} \end{array} \right\}$

ST

$\Psi_{ST} =$

- Ψ_{base}

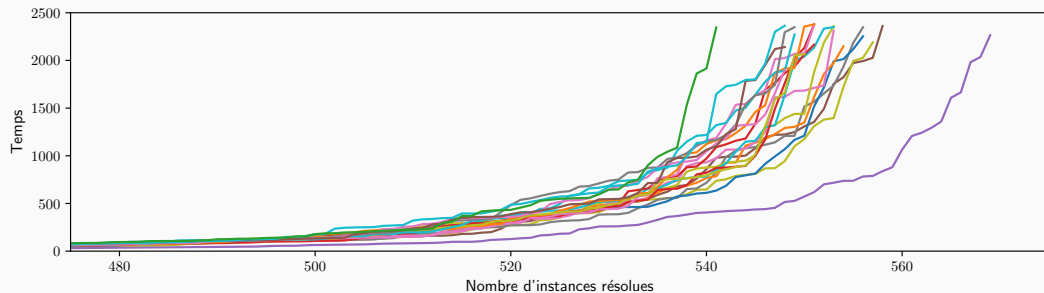
Heuristiques de base

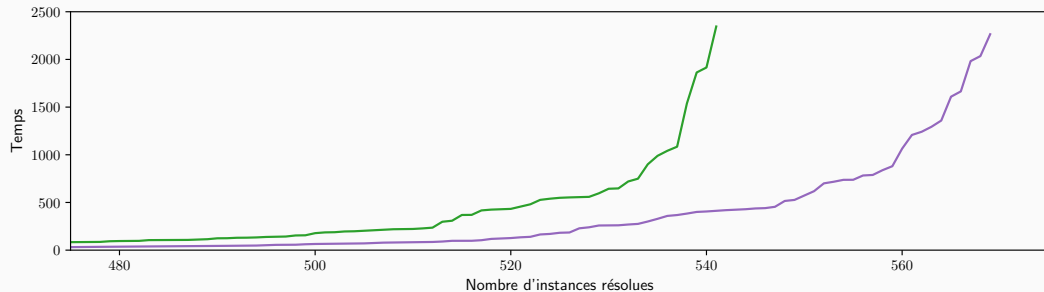
- Ψ_{RES}

RES

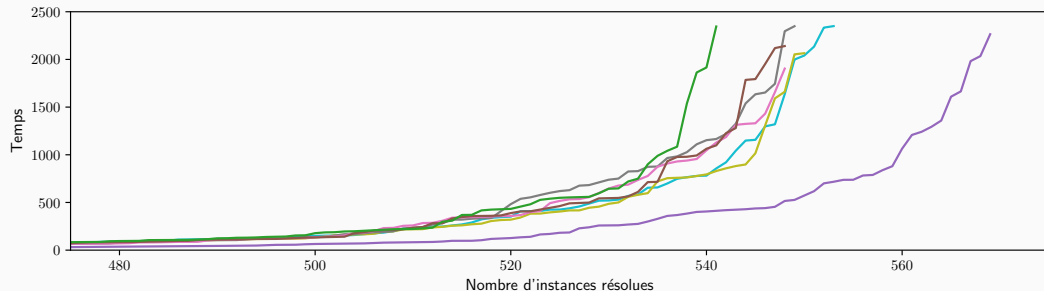
- $\left\{ \text{ACE}(\text{RES}(\mathcal{H}, \mathcal{B}_M^{ST}, \mathcal{R})) \mid \begin{array}{l} M \in \{1, 2, 4, 8, 16, 32\} \\ \mathcal{R} \in \{\text{auvr}, \text{esb}, \text{npts}\} \end{array} \right\}$

ST

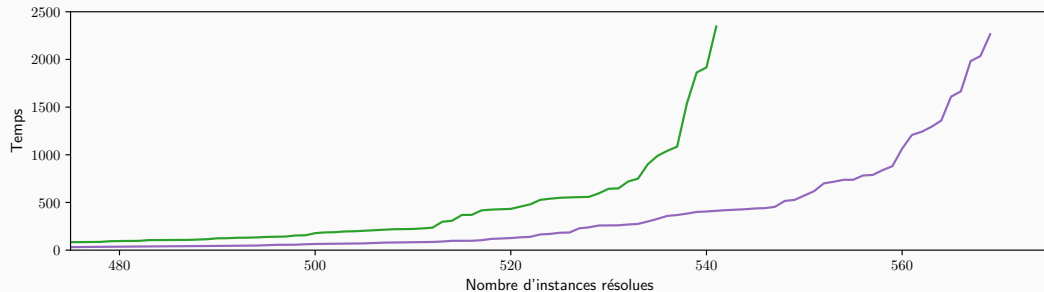




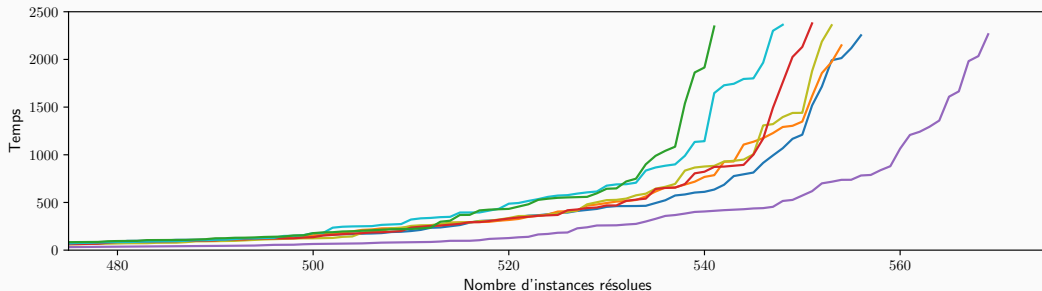
	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
VBS($\mathcal{H}^{\text{base}}$)	569	52,355	575
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UNI}}$)	541	113,903	575



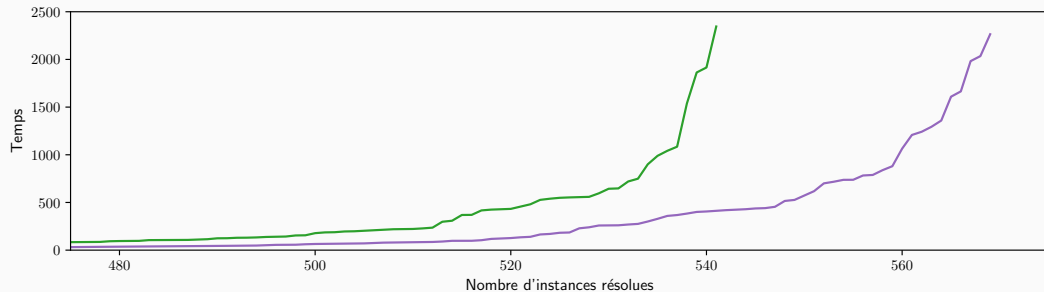
	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
VBS(\mathcal{H}^{base})	569	52,355	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,2}^{ST}, \mathcal{R}^{auvr}$)	553	95,691	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,4}^{ST}, \mathcal{R}^{auvr}$)	550	95,156	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,32}^{ST}, \mathcal{R}^{auvr}$)	549	104,730	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,16}^{ST}, \mathcal{R}^{auvr}$)	548	101,431	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,8}^{ST}, \mathcal{R}^{auvr}$)	548	102,801	575
RES($\mathcal{H}^{base}, \mathcal{B}^{UNI}$)	541	113,903	575



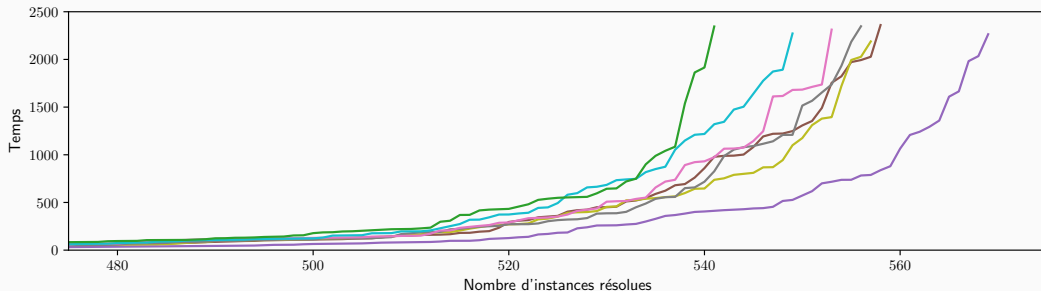
	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
$VBS(\mathcal{H}^{base})$	569	52,355	575
$RES(\mathcal{H}^{base}, \mathcal{B}^{UNI})$	541	113,903	575



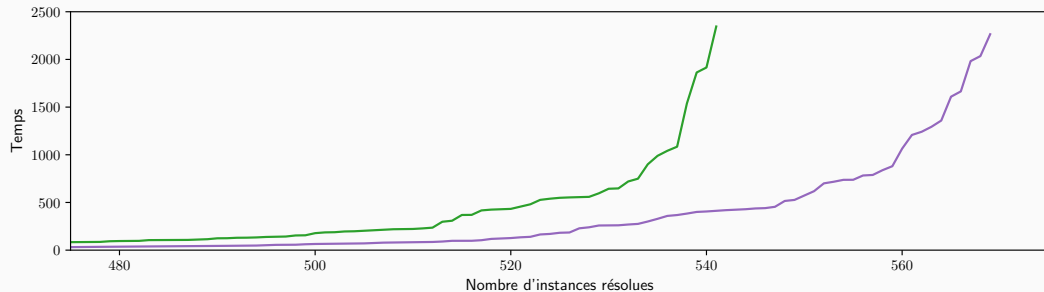
	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
VBS(\mathcal{H}^{base})	569	52,355	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,4}^{ST}, \mathcal{R}^{esb}$)	556	86,485	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,2}^{ST}, \mathcal{R}^{esb}$)	554	91,031	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,16}^{ST}, \mathcal{R}^{esb}$)	553	92,761	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,8}^{ST}, \mathcal{R}^{esb}$)	551	94,457	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,32}^{ST}, \mathcal{R}^{esb}$)	548	107,999	575
RES($\mathcal{H}^{base}, \mathcal{B}^{UNI}$)	541	113,903	575



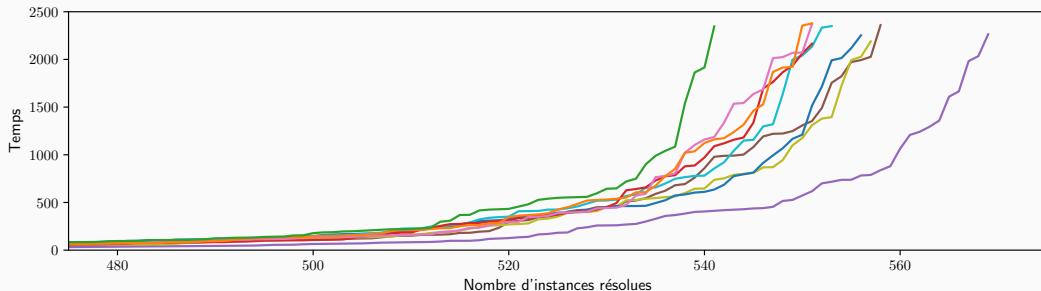
	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
VBS($\mathcal{H}^{\text{base}}$)	569	52,355	575
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UNI}}$)	541	113,903	575



	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
VBS(\mathcal{H}^{base})	569	52,355	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,8}^{ST}, \mathcal{R}^{npts}$)	558	85,655	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,4}^{ST}, \mathcal{R}^{npts}$)	557	82,517	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,2}^{ST}, \mathcal{R}^{npts}$)	556	86,275	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,16}^{ST}, \mathcal{R}^{npts}$)	553	92,632	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,32}^{ST}, \mathcal{R}^{npts}$)	549	103,125	575
RES($\mathcal{H}^{base}, \mathcal{B}^{UNI}$)	541	113,903	575

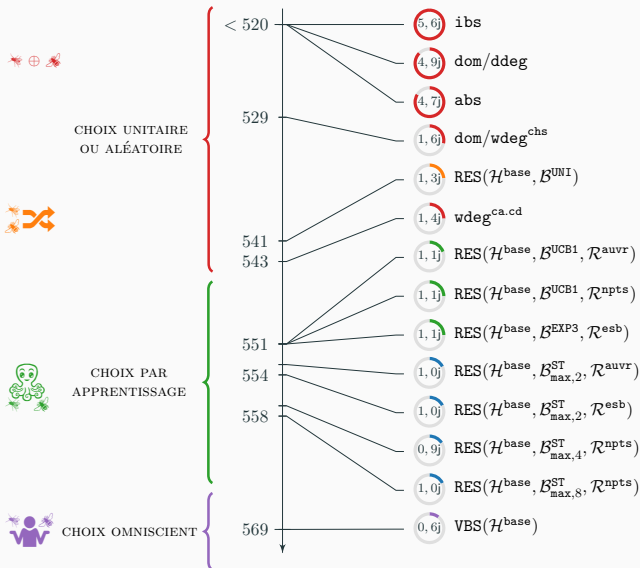


	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
VBS($\mathcal{H}^{\text{base}}$)	569	52,355	575
RES($\mathcal{H}^{\text{base}}, \mathcal{B}^{\text{UNI}}$)	541	113,903	575



	NOMBRE DE RÉOLUTIONS (#)	TEMPS DE RÉOLUTION (s)	NOMBRE TOTAL D'INSTANCES (#)
VBS(\mathcal{H}^{base})	569	52,355	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,8}^{ST}, \mathcal{R}^{npts}$)	558	85,655	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,4}^{ST}, \mathcal{R}^{npts}$)	557	82,517	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,4}^{ST}, \mathcal{R}^{esb}$)	556	86,485	575
RES($\mathcal{H}^{base}, \mathcal{B}_{max,2}^{ST}, \mathcal{R}^{auvr}$)	553	95,691	575
RES($\mathcal{H}^{base}, \mathcal{B}^{EXP3}, \mathcal{R}^{esb}$)	551	96,637	575
RES($\mathcal{H}^{base}, \mathcal{B}^{UCB1}, \mathcal{R}^{npts}$)	551	98,791	575
RES($\mathcal{H}^{base}, \mathcal{B}^{UCB1}, \mathcal{R}^{auvr}$)	551	98,980	575
RES($\mathcal{H}^{base}, \mathcal{B}^{UNI}$)	541	113,903	575

Bilan sur l'apprentissage de la meilleure heuristique



Conclusion et perspectives

ST propose de **nouvelles performances** grâce, en partie, à une meilleure équité en terme de :

- duel de bras sur des runs de même taille
- choix uniforme des duels à chaque petits runs (aucune heuristique n'est écartée)
- des duels (booléens) insensibles à la variance de la distribution
- plus le run est grand, plus il est associé à une heuristique efficace à ce moment-là

Quelques pistes d'améliorations :

- à chaque duel, les connaissances du run perdant ne sont pas utilisées → possibilité de fusion de connaissances :
 - échange des scores inter-heuristiques ?
 - à partir de la base de nogoods ?

Conclusion et perspectives

ST propose de **nouvelles performances** grâce, en partie, à une meilleure équité en terme de :

- duel de bras sur des runs de même taille
- choix uniforme des duels à chaque petits runs (aucune heuristique n'est écartée)
- des duels (booléens) insensibles à la variance de la distribution
- plus le run est grand, plus il est associé à une heuristique efficace à ce moment-là

Quelques pistes d'améliorations :

- à chaque duel, les connaissances du run perdant ne sont pas utilisées → possibilité de fusion de connaissances :
 - échange des scores inter-heuristiques ?
 - à partir de la base de nogoods ?

Conclusion et perspectives

ST propose de **nouvelles performances** grâce, en partie, à une meilleure équité en terme de :

- duel de bras sur des runs de même taille
- choix uniforme des duels à chaque petits runs (aucune heuristique n'est écartée)
- des duels (booléens) insensibles à la variance de la distribution
- plus le run est grand, plus il est associé à une heuristique efficace à ce moment-là

Quelques pistes d'améliorations :

- à chaque duel, les connaissances du run perdant ne sont pas utilisées → possibilité de fusion de connaissances :
 - échange des scores inter-heuristiques ?
 - à partir de la base de nogoods ?

Conclusion et perspectives

ST propose de **nouvelles performances** grâce, en partie, à une meilleure équité en terme de :

- duel de bras sur des runs de même taille
- choix uniforme des duels à chaque petits runs (aucune heuristique n'est écartée)
- des duels (booléens) insensibles à la variance de la distribution
- plus le run est grand, plus il est associé à une heuristique efficace à ce moment-là

Quelques pistes d'améliorations :

- à chaque duel, les connaissances du run perdant ne sont pas utilisées → possibilité de fusion de connaissances :
 - échange des scores inter-heuristiques ?
 - à partir de la base de nogoods ?

Conclusion et perspectives

ST propose de **nouvelles performances** grâce, en partie, à une meilleure équité en terme de :

- duel de bras sur des runs de même taille
- choix uniforme des duels à chaque petits runs (aucune heuristique n'est écartée)
- des duels (booléens) insensibles à la variance de la distribution
- plus le run est grand, plus il est associé à une heuristique efficace à ce moment-là

Quelques pistes d'améliorations :

- à chaque duel, les connaissances du run perdant ne sont pas utilisées → possibilité de fusion de connaissances :
 - échange des scores inter-heuristiques ?
 - à partir de la base de nogoods ?

Conclusion et perspectives

ST propose de **nouvelles performances** grâce, en partie, à une meilleure équité en terme de :

- duel de bras sur des runs de même taille
- choix uniforme des duels à chaque petits runs (aucune heuristique n'est écartée)
- des duels (booléens) insensibles à la variance de la distribution
- plus le run est grand, plus il est associé à une heuristique efficace à ce moment-là

Quelques pistes d'améliorations :

- à chaque duel, les connaissances du run perdant ne sont pas utilisées → possibilité de fusion de connaissances :
 - échange des scores inter-heuristiques ?
 - à partir de la base de nogoods ?

Focus sur l'heuristique *Conflict History Search*

Conflict History Search [3]

Heuristique : $\text{wdeg}^{\text{chs}}(\mathcal{P} = (\mathcal{X}, \mathcal{C}) : \text{CN}, \alpha_0)$

1 **Méthode** INIT() :

```
2   time  $\leftarrow$  0
3   pour chaque  $c \in \mathcal{C}$  faire
4        $w_c \leftarrow 0$ 
5        $t_c \leftarrow 0$ 
```

6 **Méthode** AVANTRUN() :

```
7   pour chaque  $c \in \mathcal{C}$  faire
8        $w_c \leftarrow w_c \times 0.995^{\text{time} - t_c}$ 
9    $\alpha \leftarrow \alpha_0$ 
```

10 **Méthode** APRÈSCONFLIT(c) :

```
11    $r \leftarrow \frac{1}{\text{time} - t_c + 1}$ 
12    $\alpha \leftarrow \max(6/100, \alpha - 10^{-6})$  //  $\alpha - 10^{-4}$  dans [2]
13    $w_c \leftarrow (1 - \alpha) \times w_c + \alpha \times r$ 
14   time  $\leftarrow$  time + 1
15    $t_c \leftarrow$  time
```

16 **Méthode** SCORE(x) :

```
17   retourner  $\sum_{c \in \mathcal{C} : x \in \text{scp}(c) \wedge |\text{fut}(c)| > 1} w_c$ 
```

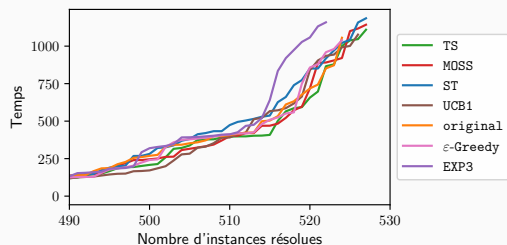
Deux propositions de diversification de $\text{dom}/\text{wdeg}^{\text{chs}}$ avec le framework RES :

- variation de α_0 ($[1]$) : $1/10, 3/10, 5/10, 7/10, 9/10$
- variation du tie-breaker : `lex`, `deg`, `rand1`, `rand2`, `rand3`

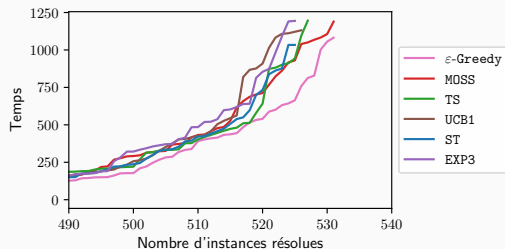
Deux propositions de diversification de $\text{dom}/\text{wdeg}^{\text{chs}}$ avec le framework RES :

- variation de α_0 ($[1]$) : $1/10, 3/10, 5/10, 7/10, 9/10$
- variation du tie-breaker : $\text{lex}, \text{deg}, \text{rand}_1, \text{rand}_2, \text{rand}_3$

Expérimentation de la diversification de $\text{dom}/\text{wdeg}^{\text{chs}}$



(a) Diversification de α_0



(b) Diversification du tie-breaker

Figure 5 – Amélioration de l'heuristique $\text{dom}/\text{wdeg}^{\text{chs}}$ par méthodes de diversification (fonction de récompense : npts)

La perturbation d'heuristique PER :

- l'heuristique de l'utilisateur \mathcal{H}_u + l'heuristique aléatoire \mathcal{H}_r
- étend le framework RES :
 - un run \rightarrow une heuristique sélectionnée
 - les choix de \mathcal{H}_r viennent perturber les scores de \mathcal{H}_u

Cela permet de donner un poids à des variables qui n'auraient pas été explorées en temps normal.

La perturbation d'heuristique PER :

- l'heuristique de l'utilisateur \mathcal{H}_u + l'heuristique aléatoire \mathcal{H}_r
- étend le framework RES :
 - un run \rightarrow une heuristique sélectionnée
 - les choix de \mathcal{H}_r viennent perturber les scores de \mathcal{H}_u

Cela permet de donner un poids à des variables qui n'auraient pas été explorées en temps normal.

La perturbation d'heuristique PER :

- l'heuristique de l'utilisateur \mathcal{H}_u + l'heuristique aléatoire \mathcal{H}_r
- étend le framework RES :
 - un run \rightarrow une heuristique sélectionnée
 - les choix de \mathcal{H}_r viennent perturber les scores de \mathcal{H}_u

Cela permet de donner un poids à des variables qui n'auraient pas été explorées en temps normal.

La perturbation d'heuristique PER :

- l'heuristique de l'utilisateur \mathcal{H}_u + l'heuristique aléatoire \mathcal{H}_r
- étend le framework RES :
 - un run \rightarrow une heuristique sélectionnée
 - les choix de \mathcal{H}_r viennent perturber les scores de \mathcal{H}_u

Cela permet de donner un poids à des variables qui n'auraient pas été explorées en temps normal.

La perturbation d'heuristique PER :

- l'heuristique de l'utilisateur \mathcal{H}_u + l'heuristique aléatoire \mathcal{H}_r
- étend le framework RES :
 - un run \rightarrow une heuristique sélectionnée
 - les choix de \mathcal{H}_r viennent **perturber les scores** de \mathcal{H}_u

Cela permet de donner un poids à des variables qui n'auraient pas été explorées en temps normal.

La perturbation d'heuristique PER :

- l'heuristique de l'utilisateur \mathcal{H}_u + l'heuristique aléatoire \mathcal{H}_r
- étend le framework RES :
 - un run \rightarrow une heuristique sélectionnée
 - les choix de \mathcal{H}_r viennent **perturber les scores** de \mathcal{H}_u

Cela permet de donner un poids à des variables qui n'auraient pas été explorées en temps normal.

Expérimentation de la perturbation de $\text{dom/wdeg}^{\text{chs}}$

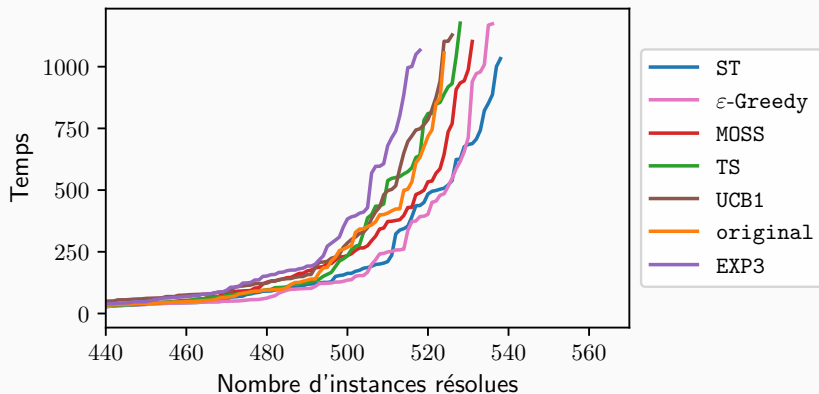
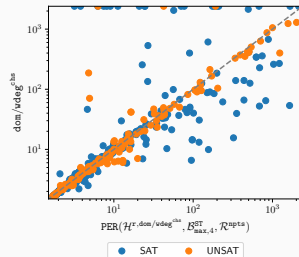
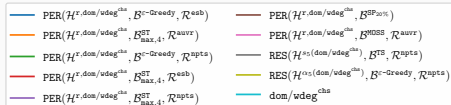
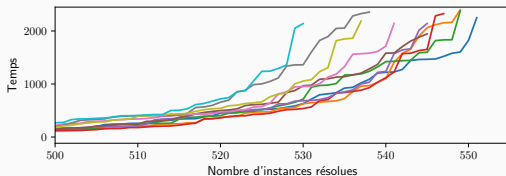


Figure 6 – Amélioration de l'heuristique $\text{dom/wdeg}^{\text{chs}}$ par perturbation (fonction de récompense : npts)

Synthèse des améliorations de $\text{dom/wdeg}^{\text{chs}}$

	#RÉS.	TPS (s)	#RÉS. COMM.	TPS COMM. (s)	#TOT.	%PERT.
$\text{PER}(\mathcal{H}^{r,\text{dom/wdeg}^{\text{chs}}}, \mathcal{B}^{\varepsilon\text{-Greedy}}, \mathcal{R}^{\text{esb}})$	551	76,818	512	20,991	565	60.7
$\text{PER}(\mathcal{H}^{r,\text{dom/wdeg}^{\text{chs}}}, \mathcal{B}_{\text{max},4}^{\text{ST}}, \mathcal{R}^{\text{auvr}})$	549	79,783	512	23,987	565	39.7
$\text{PER}(\mathcal{H}^{r,\text{dom/wdeg}^{\text{chs}}}, \mathcal{B}^{\varepsilon\text{-Greedy}}, \mathcal{R}^{\text{npts}})$	549	82,437	512	22,785	565	46.0
$\text{PER}(\mathcal{H}^{r,\text{dom/wdeg}^{\text{chs}}}, \mathcal{B}_{\text{max},4}^{\text{ST}}, \mathcal{R}^{\text{esb}})$	547	78,404	512	19,822	565	42.5
$\text{PER}(\mathcal{H}^{r,\text{dom/wdeg}^{\text{chs}}}, \mathcal{B}_{\text{max},4}^{\text{ST}}, \mathcal{R}^{\text{npts}})$	545	84,510	512	22,614	565	35.7
$\text{PER}(\mathcal{H}^{r,\text{dom/wdeg}^{\text{chs}}}, \mathcal{B}^{\text{SP}_{20\%}})$	545	88,422	512	25,565	565	20.0
$\text{PER}(\mathcal{H}^{r,\text{dom/wdeg}^{\text{chs}}}, \mathcal{B}^{\text{MOSS}}, \mathcal{R}^{\text{auvr}})$	541	94,875	512	24,826	565	51.2
$\text{RES}(\mathcal{H}^{s_5}(\text{dom/wdeg}^{\text{chs}}), \mathcal{B}^{\text{TS}}, \mathcal{R}^{\text{npts}})$	538	107,659	512	20,777	565	—
$\text{RES}(\mathcal{H}^{\alpha_5}(\text{dom/wdeg}^{\text{chs}}), \mathcal{B}^{\varepsilon\text{-Greedy}}, \mathcal{R}^{\text{npts}})$	537	101,111	512	21,390	565	—
$\text{dom/wdeg}^{\text{chs}}$	530	113,823	512	18,575	565	—



Comme présenté dans [2], $\text{dom/wdeg}^{\text{chs}}$ peut être améliorée par diversification du paramètre α_0 , mais aussi par diversification du tie-breaker.

Nous avons remarqué que l'heuristique est particulièrement sensible au framework PER en doublant le nombre d'instances résolues par rapport aux précédentes techniques.

Comme présenté dans [2], $\text{dom/wdeg}^{\text{chs}}$ peut être améliorée par diversification du paramètre α_0 , mais aussi par diversification du tie-breaker.

Nous avons remarqué que l'heuristique est particulièrement sensible au framework PER en doublant le nombre d'instances résolues par rapport aux précédentes techniques.

Export du framework

Definition (COP)

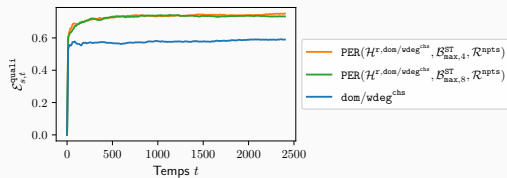
Un *Problème d'Optimisation sous Contraintes* \mathcal{P} est défini par :

- un ensemble fini de **variables**, noté \mathcal{X} ;
- un ensemble fini de **contraintes**, noté \mathcal{C} , tel que $\forall c \in \mathcal{C}, \text{scp}(c) \subseteq \mathcal{X}$;
- une **fonction objectif** $\mathcal{O} = \text{obj}(\mathcal{P})$ devant être maximisée ou minimisée.

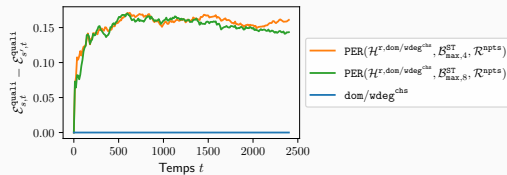
Definition (Solution optimale (COP))

Pour la minimisation (resp. maximisation), une *solution optimale* de \mathcal{P} est une solution pour laquelle la valeur objectif $\text{obj}(\mathcal{P})$ est inférieure ou égale (resp. supérieure ou égale) à la valeur de toute autre solution.

Optimisation de contraintes - Amélioration de $\text{dom/wdeg}^{\text{chs}}$



(a) Vue directe



(b) Vue par rapport à l'heuristique d'origine

Figure 8 – Amélioration de l'heuristique $\text{dom/wdeg}^{\text{chs}}$ par perturbation (fonction de récompense : npts)

Optimisation de contraintes - Identification de la meilleure heuristique

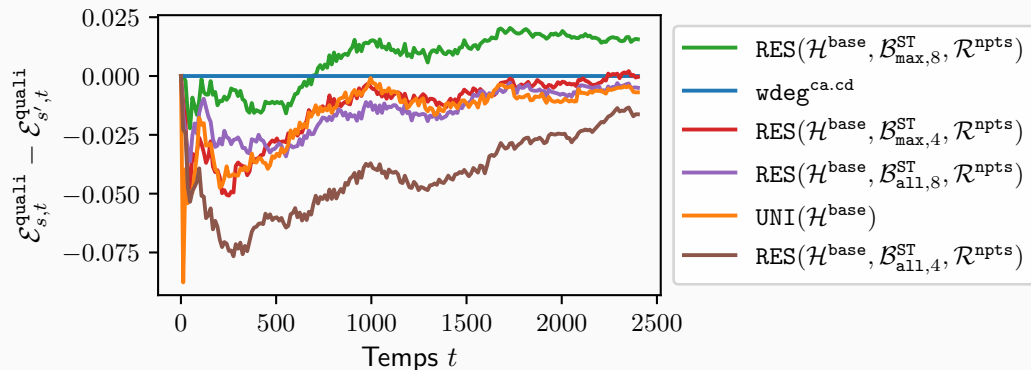


Figure 9 – Identification de la meilleure heuristique dans $\mathcal{H}^{\text{base}}$

L'identification de la meilleure heuristique ne semble que **peu efficace** :

- on retrouve, a minima, les performances de la meilleure heuristique de $\mathcal{H}^{\text{base}}$
- peut être lié au fait, qu'en optimisation, seules une ou deux heuristiques de $\mathcal{H}^{\text{base}}$ dominant

La **diversification** semble **plus efficace** !

Dans les deux cas, il pourrait être intéressant de :

- personnaliser la fonction de récompense en prenant en compte l'amélioration de la borne

L'identification de la meilleure heuristique ne semble que **peu efficace** :

- on retrouve, a minima, les performances de la meilleure heuristique de $\mathcal{H}^{\text{base}}$
- peut être lié au fait, qu'en optimisation, seules une ou deux heuristiques de $\mathcal{H}^{\text{base}}$ dominant

La **diversification** semble **plus efficace** !

Dans les deux cas, il pourrait être intéressant de :

- personnaliser la fonction de récompense en prenant en compte l'amélioration de la borne

L'identification de la meilleure heuristique ne semble que **peu efficace** :

- on retrouve, a minima, les performances de la meilleure heuristique de $\mathcal{H}^{\text{base}}$
- peut être lié au fait, qu'en optimisation, seules une ou deux heuristiques de $\mathcal{H}^{\text{base}}$ dominant

La **diversification** semble **plus efficace** !

Dans les deux cas, il pourrait être intéressant de :

- personnaliser la fonction de récompense en prenant en compte l'amélioration de la borne

Deux solveurs aux principes de résolutions différents :

- *Solveur Pseudo-Booléen* : principe de résolution proche des solveurs SAT
- *Solveur MILP (Mixed-Integer Linear Programming)* : principe de résolution basée sur la relaxation des contraintes d'intégrité des variables et sur l'usage répétitif du simplexe et de branchement sur les variables (branch&cut).

Mais ayant, comme intersection, la modélisation des problèmes :

- Minimisation de $\sum_i c_i \cdot x_i$
- tel que $\sum_j a_{ij} \cdot x_j \leq b_i \quad \forall i$

Deux solveurs aux principes de résolutions différents :

- *Solveur Pseudo-Booléen* : principe de résolution proche des solveurs SAT
- *Solveur MILP (Mixed-Integer Linear Programming)* : principe de résolution basée sur la relaxation des contraintes d'intégrité des variables et sur l'usage répétitif du simplexe et de branchement sur les variables (branch&cut).

Mais ayant, comme intersection, la modélisation des problèmes :

- Minimisation de $\sum_i c_i \cdot x_i$
- tel que $\sum_j a_{ij} \cdot x_j \leq b_i \quad \forall i$

Conditions :

- sans échange d'informations
- exécution des solveurs SAT4j-PB et CPLEX
- instances SAT de la dernière compétition PB (2016)

Observations :

- VBS dont le timeout est divisé par 2 (\leftrightarrow choix uniforme) permet de récupérer une large quantité d'instances (supérieur aux deux solveurs)
- SAT4j domine mais CPLEX résout presque instantanément des instances UNSAT (où SAT4j produit un timeout)

L'échange d'informations telles que les bornes ou l'apprentissage de l'espace de recherche (plans coupants) sont des techniques de l'état de l'art pour améliorer la performance de ces hybridations.

Une perspective en prolongement des précédents sujets : usage d'AST pour attribuer du temps de calcul aux solveurs.

L'échange d'informations telles que les bornes ou l'apprentissage de l'espace de recherche (plans coupants) sont des techniques de l'état de l'art pour améliorer la performance de ces hybridations.

Une perspective en prolongement des précédents sujets : usage d'AST pour attribuer du temps de calcul aux solveurs.

Identification de la meilleure heuristique pour les solveurs de problèmes combinatoires

Hugues Watez

24 janvier 2024

Séminaire équipe COALA au LIS

PER : description algorithmique du framework RES perturbé

Heuristique : PER(\mathcal{H} : heuristiques, \mathcal{B} : politique de sélection, \mathcal{R} : fonction de récompense)

```
1 Méthode INIT() :
2    $t \leftarrow 0$ 
3    $K \leftarrow |\mathcal{H}|$ 
4   INIT $_{\mathcal{B}}(K)$ 
5   pour chaque  $i \in [K]$  faire
6     INIT $_{\mathcal{H}_i}()$ 

7 Méthode AVANTRUN() :
8    $t \leftarrow t + 1$ 
9    $i_t \leftarrow \text{SELECT}_{\mathcal{B}}(t)$ 
10  pour chaque  $i \in [K]$  faire
11    AVANTRUN $_{\mathcal{H}_i}()$ 

12 Méthode APRÈSRUN() :
13   MAJ $_{\mathcal{B}}(\mathcal{R}_t)$ 

14 Méthode APRÈSCONFLIT( $c$ ) :
15   pour chaque  $i \in [K]$  faire
16     APRÈSCONFLIT $_{\mathcal{H}_i}()$ 

17 Méthode APRÈSASSIGNATION( $\mathcal{P}_{av}$ ,  $\mathcal{P}_{ap}$ ) :
18   pour chaque  $i \in [K]$  faire
19     APRÈSASSIGNATION $_{\mathcal{H}_i}(\mathcal{P}_{av}, \mathcal{P}_{ap})$ 

20 Méthode SCORE( $x$ ) :
21   retourner SCORE $_{\mathcal{H}_{i_t}}(x)$ 
```

PER : description algorithmique du framework RES perturbé

Heuristique : PER(\mathcal{H} : heuristiques, \mathcal{B} : politique de sélection, \mathcal{R} : fonction de récompense)

```
1 Méthode INIT() :
2    $t \leftarrow 0$ 
3    $K \leftarrow |\mathcal{H}|$ 
4   INIT $\mathcal{B}$ ( $K$ )
5   pour chaque  $i \in [K]$  faire
6     INIT $\mathcal{H}_i$ ()

7 Méthode AVANTRUN() :
8    $t \leftarrow t + 1$ 
9    $i_t \leftarrow \text{SELECT}_{\mathcal{B}}(t)$ 
10  pour chaque  $i \in [K]$  faire
11    AVANTRUN $\mathcal{H}_i$ ()

12 Méthode APRÈSRUN() :
13   MAJ $\mathcal{B}$ ( $\mathcal{R}_t$ )

14 Méthode APRÈSCONFLIT( $c$ ) :
15   pour chaque  $i \in [K]$  faire
16     APRÈSCONFLIT $\mathcal{H}_i$ ()

17 Méthode APRÈSASSIGNATION( $\mathcal{P}_{av}$ ,  $\mathcal{P}_{ap}$ ) :
18   pour chaque  $i \in [K]$  faire
19     APRÈSASSIGNATION $\mathcal{H}_i$ ( $\mathcal{P}_{av}$ ,  $\mathcal{P}_{ap}$ )

20 Méthode SCORE( $x$ ) :
21   retourner SCORE $\mathcal{H}_{i_t}$ ( $x$ )
```

PER : description algorithmique du framework RES perturbé

Heuristique : PER(\mathcal{H} : heuristiques, \mathcal{B} : politique de sélection, \mathcal{R} : fonction de récompense)

```
1 Méthode INIT() :  
2    $t \leftarrow 0$   
3    $K \leftarrow |\mathcal{H}|$   
4   INIT $_{\mathcal{B}}(K)$   
5   pour chaque  $i \in [K]$  faire  
6     INIT $_{\mathcal{H}_i}()$ 
```

```
7 Méthode AVANTRUN() :  
8    $t \leftarrow t + 1$   
9    $i_t \leftarrow \text{SELECT}_{\mathcal{B}}(t)$   
10  pour chaque  $i \in [K]$  faire  
11    AVANTRUN $_{\mathcal{H}_i}()$ 
```

```
12 Méthode APRÈSRUN() :  
13   MAJ $_{\mathcal{B}}(\mathcal{R}_t)$ 
```

```
14 Méthode APRÈSCONFLIT( $c$ ) :  
15   pour chaque  $i \in [K]$  faire  
16     APRÈSCONFLIT $_{\mathcal{H}_i}()$ 
```

```
17 Méthode APRÈSASSIGNATION( $\mathcal{P}_{av}, \mathcal{P}_{ap}$ ) :  
18   pour chaque  $i \in [K]$  faire  
19     APRÈSASSIGNATION $_{\mathcal{H}_i}(\mathcal{P}_{av}, \mathcal{P}_{ap})$ 
```

```
20 Méthode SCORE( $x$ ) :  
21   retourner SCORE $_{\mathcal{H}_{i_t}}(x)$ 
```

Optimisation de contraintes - Évaluation des expérimentations

Nous proposons une méthode d'évaluation de la qualité des bornes obtenues pour tout solveur s , en tout temps t et pour toute instance i (on considère que l'objectif est à minimiser) :

- $b_{s,t}^i$ est la borne obtenue pour le triplet (i, s, t)
- $\min_t^i = \min_{s \in \Psi} b_{s,t}^i$ est la borne minimale du couple (i, t) (tout solveur confondu)
- $\max_t^i = \max_{s \in \Psi} b_{s,t}^i$ est la borne maximale du couple (i, t) (tout solveur confondu)
- $n_{s,t}^i = \begin{cases} 0, & \text{si } b_{s,t}^i = \infty \\ 1, & \text{si } \max_t^i = \min_t^i \\ \frac{\max_t^i - b_{s,t}^i}{\max_t^i - \min_t^i}, & \text{sinon} \end{cases}$ est la borne normalisée pour le triplet (i, s, t)

Ainsi, $\mathcal{E}_{s,t}^{\text{quali}} = \frac{1}{|\mathcal{I}|} \times \sum_{i \in \mathcal{I}} n_{s,t}^i$ est la mesure de qualité des bornes pour le solveur s au temps t .

Références

- [1] M. S. CHERIF, D. HABET et C. TERRIOUX. **“Combining VSIDS and CHB Using Restarts in SAT”**. In : *Proceedings of the 27th International Conference on Principles and Practice of Constraint Programming (CP'21)*. Oct. 2021.
- [2] M. S. CHERIF, D. HABET et C. TERRIOUX. **“On the Refinement of Conflict History Search Through Multi-Armed Bandit”**. In : *IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*. Baltimore, United States : IEEE, nov. 2020, p. 264-271. DOI : 10.1109/ICTAI50040.2020.00050.

- [3] D. HABET et C. TERRIOUX. **“Conflict History Based Heuristic for Constraint Satisfaction Problem Solving”**. In : *Journal of Heuristics* 27.6 (juin 2021), p. 951-990. DOI : 10.1007/s10732-021-09475-z.
- [4] Frederic KORICHE et al. **“Best Heuristic Identification for Constraint Satisfaction”**. In : *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. Sous la dir. de Lud De RAEDT. Main Track. International Joint Conferences on Artificial Intelligence Organization, juill. 2022, p. 1859-1865. DOI : 10.24963/ijcai.2022/258. URL : <https://doi.org/10.24963/ijcai.2022/258>.
- [5] H. WATTEZ, F. KORICHE et A. PAPARRIZOU. **“Perturbation des heuristiques de branchement dans la résolution de contraintes”**. In : *Actes des 16es Journées Francophones de Programmation par Contraintes (JFPC'21)*. Juin 2021.

- [6] H. WATTEZ et al. **“Learning Variable Ordering Heuristics with Multi-Armed Bandits and Restarts”**. In : *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI'20)*. Frontiers in Artificial Intelligence and Applications 325, IOS Press 2020, ISBN 978-1-64368-100-9. Santiago de Compostela (virtual), Spain, août 2020. DOI : 10.3233/FAIA200115.
- [7] H. WATTEZ et al. **“Refining Constraint Weighting”**. In : *Proceedings of the 31st International Conference on Tools with Artificial Intelligence (ICTAI'19)*. Nov. 2019, p. 71-77. DOI : 10.1109/ICTAI.2019.00019.
- [8] W. XIA et R. H. C. YAP. **“Learning Robust Search Strategies Using a Bandit-Based Approach”**. In : *Proceedings of AAAI'18*. 2018, p. 6657-6665.