

## Parallel Hybrid Best-First Search

---



**Author : Abdelkader Beldjilali**

**Supervisors : Dr. Simon De Givry, Dr. David Allouche**

**Tuesday 10th September 2019 - 13:30 - Room Caudron 019**

**ENAC**



# Plan

- 1 **Introduction**
- 2 **Formalisme Weighted Constraint Satisfaction Problem**
  - Vocabulaire
  - Structure de valuation pour les WCSP
  - Exemples
  - Toulbar2 : solveur de Réseaux de fonctions de coûts
  - Fonction objectif du WCSP
- 3 **HBFS**
  - Hybrid Best-First Search
- 4 **Parallélisation**
  - Master-Worker
  - Parallel HBFS
  - Quelques résultats
- 5 **Conclusion**
- 6 **Perspectives**



## Présentation du candidat

- Nom : BELDJILALI Abdelkader,
- En reconversion *sur le tard* à l'ENAC,
- Formation IENAC, Majeure SITA,
- PFE du 18 mars au 15 septembre 2019 en cours à l'INRA Toulouse.





# Présentation du Service d'accueil



- Département : Mathématiques et Informatique Appliquées (MIA)
- Unité : Mathématiques et Informatique Appliquées de Toulouse (MIAT) anciennement unité de Biométrie et Intelligence Artificielle,
- Site de Auzeville-Tolosane,
- Equipe SAB : Statistics and Algorithms for Biology



## Rôle de l'équipe

- Développe des méthodes en mathématiques, statistiques, probabilités,
- et en informatique : modélisation, optimisation combinatoire, réseaux de contraintes, algorithmique,
- Implémente ces méthodes dans toulbar2, un solveur C++ de CFN (réseau de fonctions de coûts)
- Collabore avec l'IRIT, l'ONERA, l'Université de Caen (GREYC), l'Université d'Aix-Marseille (LSIS), l'Institut de recherche en Intelligence Artificielle de Barcelone (CSIC), la Chinese University of Hong-Kong.





## Présentation du sujet

Objectif assigné : Paralléliser l'algorithme de recherche hybride en meilleur d'abord (HBFS)

- *hybride* car HBFS combine un BFS et un DFS borné par un nombre adaptatif de backtracks,
- DFS : Depth-First Search,
- BFS : Best-First Search,
- HBFS est l'algorithme par défaut dans toulbar2,
- On reste à un niveau abstrait : un problème est un hypergraphe,
- Hypergraphe : Les nœuds = variables, arêtes = contraintes
- Le résultat de la modélisation est un fichier au format wcsp donné en entrée à toulbar2.



# Plan

- 1 Introduction
- 2 **Formalisme Weighted Constraint Satisfaction Problem**
  - Vocabulaire
  - Structure de valuation pour les WCSP
  - Exemples
  - Toulbar2 : solveur de Réseaux de fonctions de coûts
  - Fonction objectif du WCSP
- 3 HBFS
  - Hybrid Best-First Search
- 4 Parallélisation
  - Master-Worker
  - Parallel HBFS
  - Quelques résultats
- 5 Conclusion
- 6 Perspectives



## Vocabulaire

- Un WCSP est un problème de satisfaction de contraintes pondérées (weighted)
- Une *fonction de coût* est appelée aussi *contrainte souple* ou *contrainte évaluée*.





## Structure de valuation

Une structure de valuation  $V$  comprend notamment :

- ① Un Ensemble de coûts possibles :  $E = \{0, \dots, k\}$  avec  $k$  élément *absorbant*,
- ② Un opérateur d'agrégation  $+_k$  tel que  
 $\forall a, b \in E, \quad a +_k b = \min(a + b, k),$
- ③  $V$  : cadre algébrique pour combiner les contraintes souples.



## Modélisation d'une contrainte dure

Soit un ensemble de variables  $X = \{x_1, x_2\}$  de domaines identiques  $D_1 = D_2 = \{a, b\}$ , avec  $a, b \in \mathbb{N}$ .

- $x_1 \neq x_2$  : contrainte dure binaire  
 $f_{\{x_1, x_2\}}(t[x_1], t[x_2]) = f_{12}(t[x_1], t[x_2])$

$f_{12}$	a	b
a	k	0
b	0	k



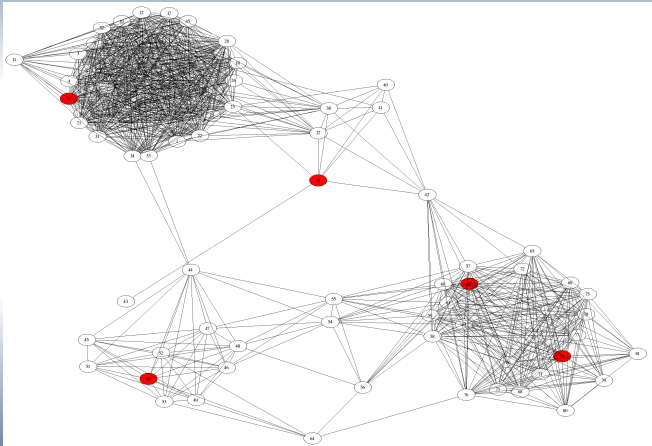
## Modélisation d'une contrainte souple

- contrainte souple binaire  $g_{12}(t[x_1], t[x_2])$  avec une structure de valuation telle que  $k = 5$

$g_{12}$	a	b
a	1	3
b	4	2



# Toulbar2 : solveur de Réseaux de fonctions de coûts





## Fonction objectif ou Valuation du WCSP

Cas arité maximale = 2

$$Val(t) = Val(t_1, \dots, t_n) = w_{\emptyset} + \sum_{i=1}^n w_i(t_i) + \sum_{w_{ij} \in C}^n w_{ij}(t_i, t_j)$$

Résoudre un WCSP c'est trouver une affectation de toutes les variables qui minimise sa valuation.



## Référence



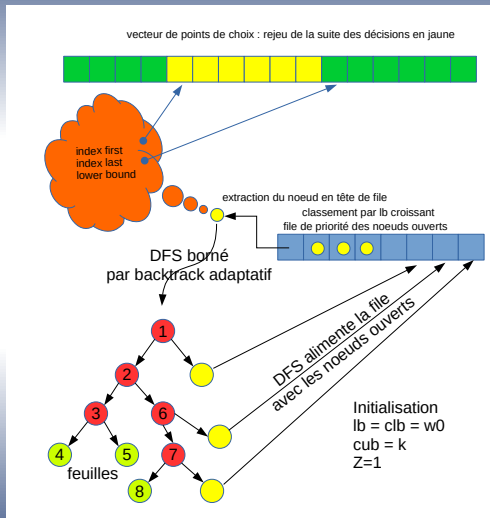
Ralphs, Ted and Shinano, Yuji and Berthold, Timo and Koch, Thorsten",  
title = Parallel Solvers for Mixed Integer Linear Optimization, booktitle =  
Handbook of Parallel Constraint Reasoning, chapter = Parallel Solvers for  
Mixed Integer Linear Optimization, pages = 283–336, year = 2018,  
publisher = "Springer International Publishing"



# Plan

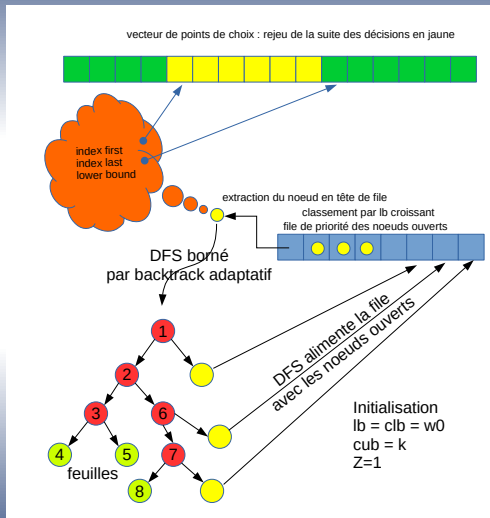
- 1 Introduction
- 2 Formalisme Weighted Constraint Satisfaction Problem  
Vocabulaire  
Structure de valuation pour les WCSP  
Exemples  
Toulbar2 : solveur de Réseaux de fonctions de coûts  
Fonction objectif du WCSP
- 3 **HBFS**  
Hybrid Best-First Search
- 4 Parallélisation  
Master-Worker  
Parallel HBFS  
Quelques résultats
- 5 Conclusion
- 6 Perspectives

# File de priorité *open*

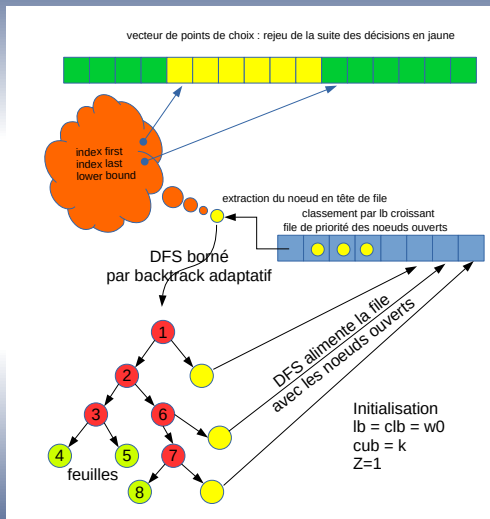




# open contenair des nœuds frontières



# Initialisation du HBFS



## Référence principale



Allouche, David and de Givry, Simon and Katsirelos, George and Schiex, Thomas and Zytnicki, Matthias, title="Anytime Hybrid Best-First Search with Tree Decomposition for Weighted CSP", booktitle="Principles and Practice of Constraint Programming", 2015.



# Plan

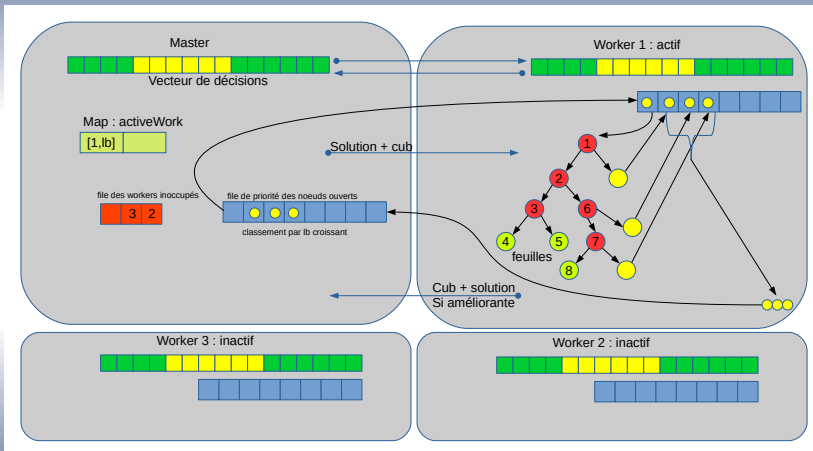
- 1 Introduction
- 2 Formalisme Weighted Constraint Satisfaction Problem
  - Vocabulaire
  - Structure de valuation pour les WCSP
  - Exemples
  - Toulbar2 : solveur de Réseaux de fonctions de coûts
  - Fonction objectif du WCSP
- 3 HBFS
  - Hybrid Best-First Search
- 4 **Parallélisation**
  - Master-Worker
  - Parallel HBFS
  - Quelques résultats
- 5 Conclusion
- 6 Perspectives



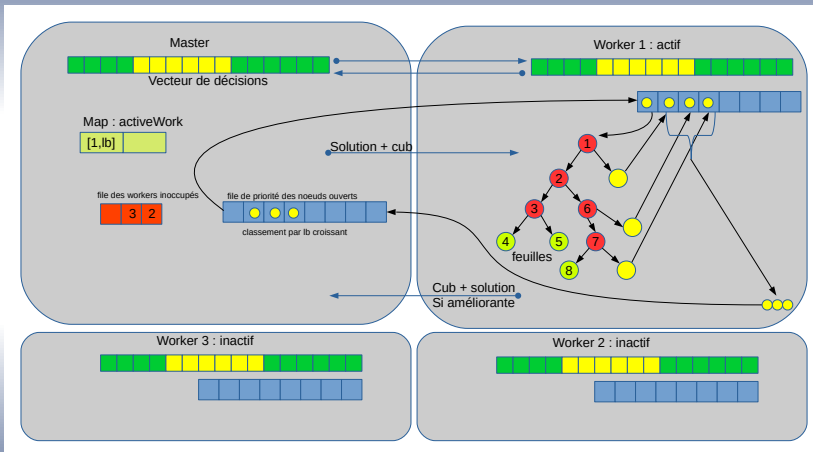
## Paradigme Master-Worker

- Un processus maître distribue les sous-problèmes aux processus "workers",
- Les workers s'exécutent sur un processeur et retournent le résultat de leur recherche suivant un certain critère,
- Ce sont les seuls échanges Master-Worker : le master ne dérange pas le worker qui travaille et le worker ne parle pas en travaillant.
- Pas d'échanges entre workers. Tout passe par le master.

# Le master envoie le nœud racine



# master : maj de idleQ et activeWork

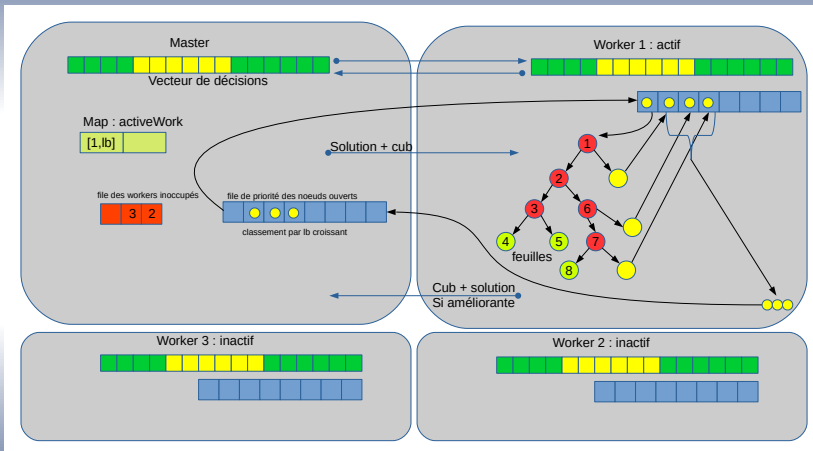


## Le worker prélève le nœud dans *open*

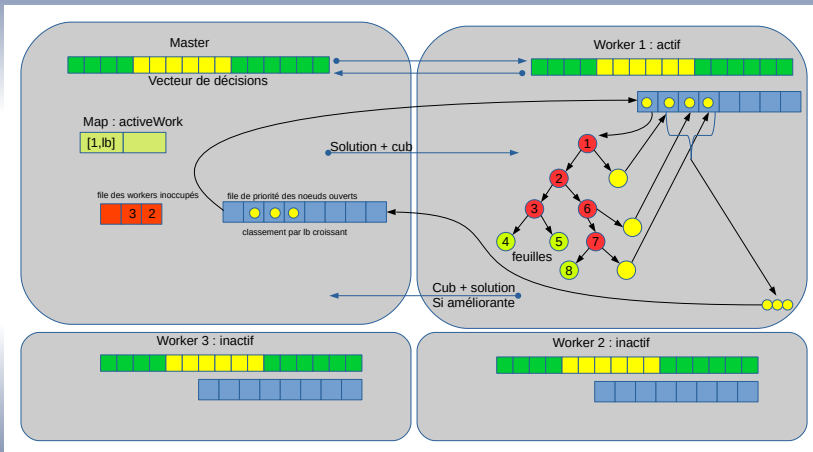




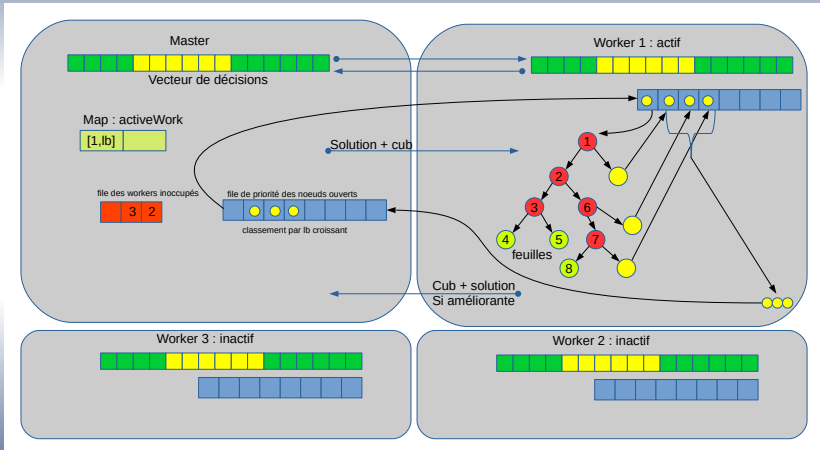
# Le worker traite le nœud par le DFBB



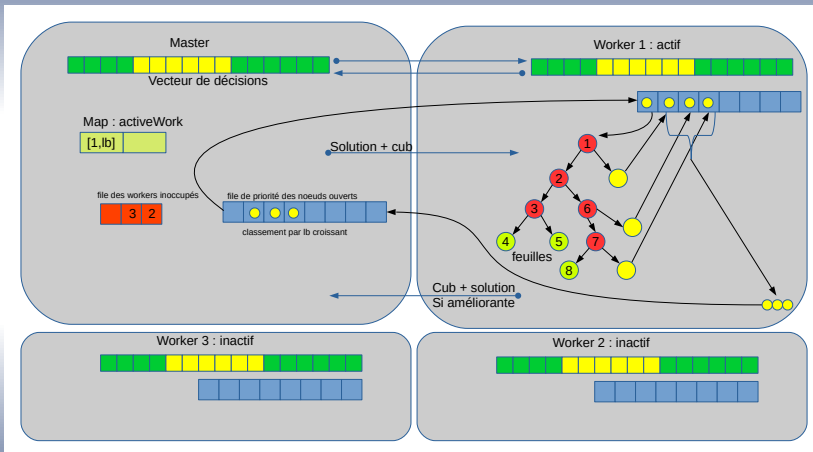
# Arbre binaire de décisions



## Feuille et affectation complète : maj du *cub*

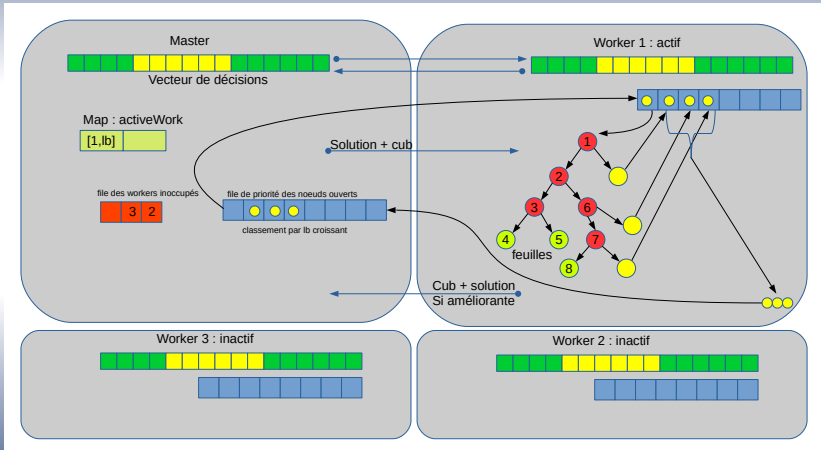


# condition d'élagage $lb \geq cub$



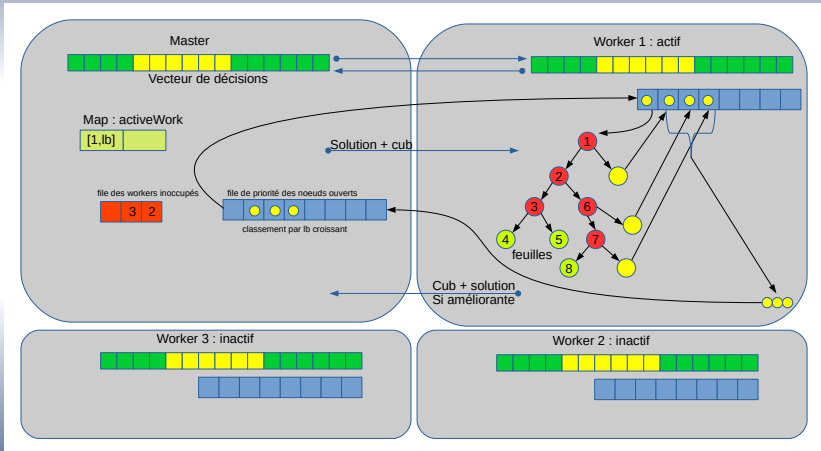


# Z adaptatif : compromis propagation/diversification et DFBB



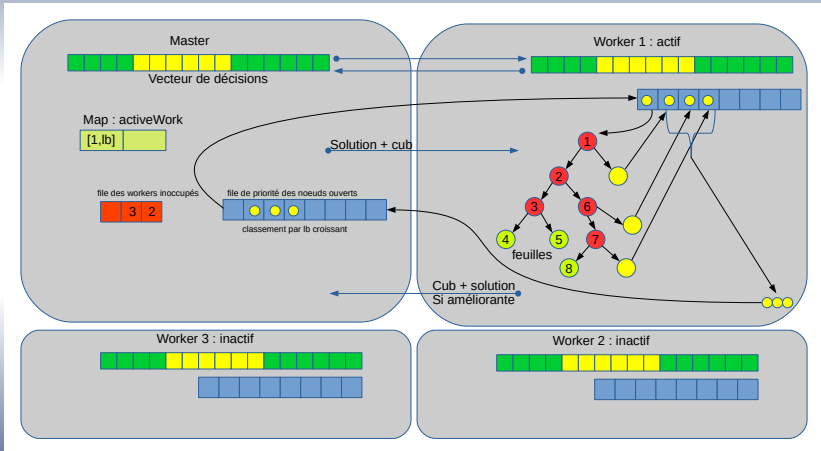


# Le DFBB du worker place les nœuds ouverts dans sa file *open*



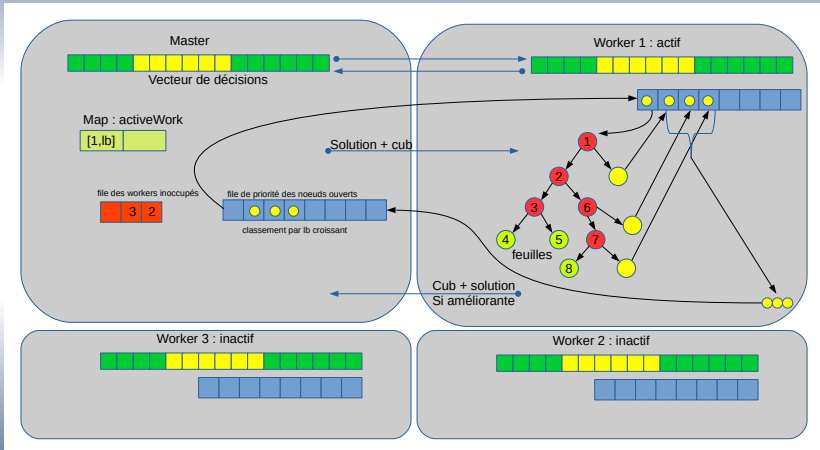


# Le worker envoie ses nœuds ouverts au master qui les range dans sa file *open*





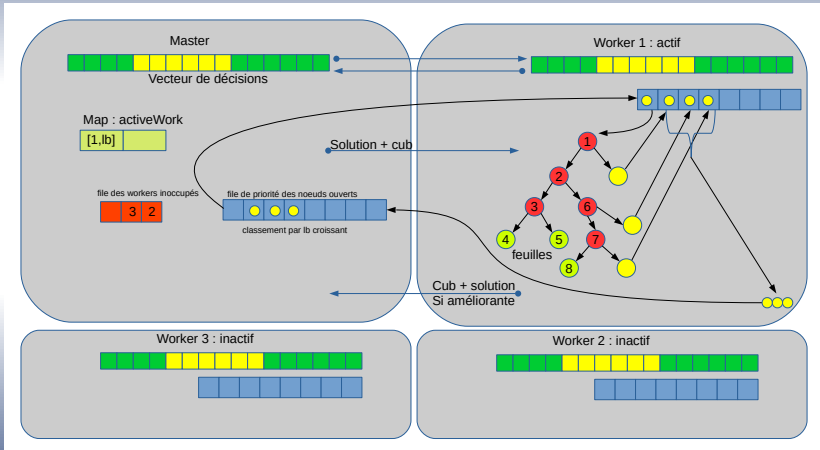
# Le worker envoie au master le *cub* trouvé et $\delta$



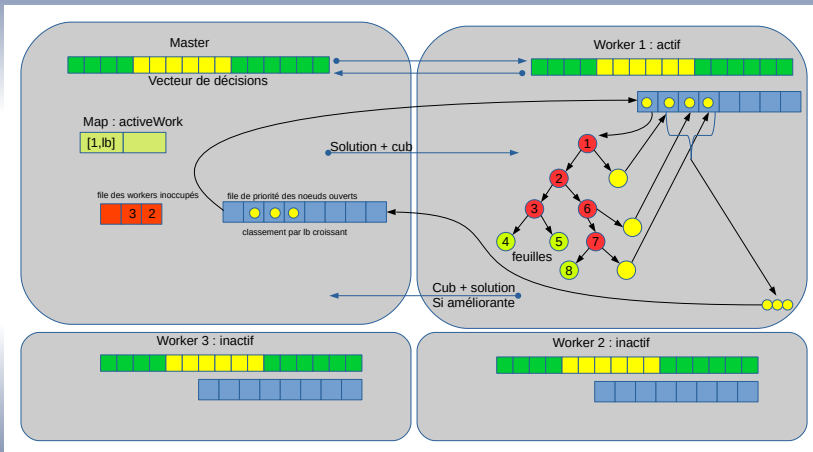




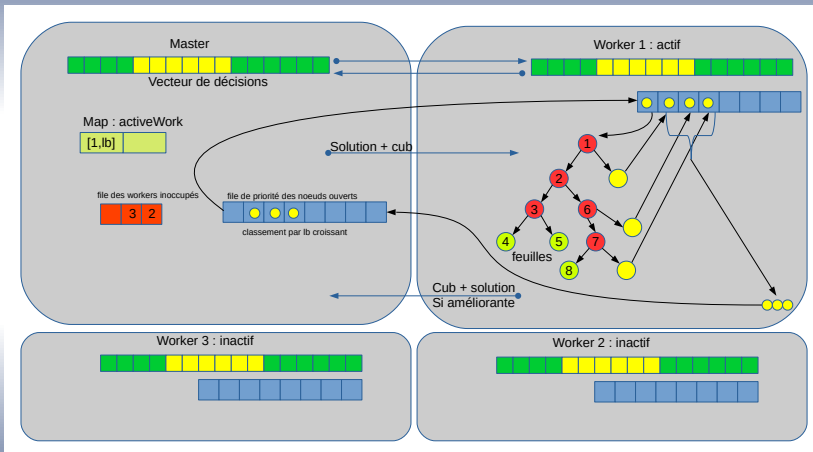
maj de cub et clb :=  
 $\max(\text{clb}, \min(\text{lb}(\text{openWorker}), \text{lb}(\text{open})))$



# maj de la file des workers libres : 1 3 2

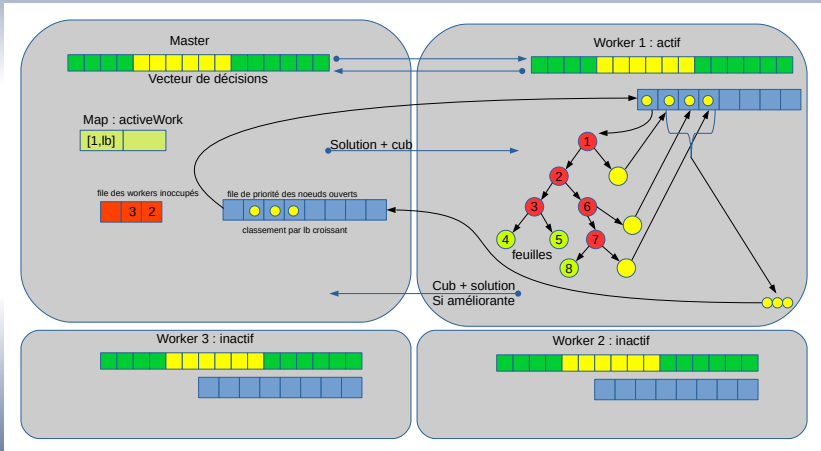


conditions d'arrêt :  $clb \geq cub$  ou ...





# plus de travail à distribuer(open vide) et en cours





## Quelques résultats sur serveur 24 cpu

wcsp file	Serial Time (s)	Parallel time (s)	speedup	Efficacité (%)	optimal ub
graph11	318,33	143,49	2,22	9,24	3080
capmp1	171,79	81,72	2,10	8,76	2460099
scen06	1024,89	48,34	21,20	88,34	3389
pedigree18	311,94	25,47	12,25	51,03	620119799
nug12	196,98	9,78	20,14	83,92	578
404.wcsp	33,31	2,05	16,25	67,70	114



## Référence principale



**Ralphs, Ted and Shinano, Yuji and Berthold, Timo and Koch, Thorsten, Handbook of Parallel Constraint Reasoning, Parallel Solvers for Mixed Integer Linear Optimization, 2018.**



# Plan

- 1 **Introduction**
- 2 **Formalisme Weighted Constraint Satisfaction Problem**
  - Vocabulaire
  - Structure de valuation pour les WCSP
  - Exemples
  - Toulbar2 : solveur de Réseaux de fonctions de coûts
  - Fonction objectif du WCSP
- 3 **HBFS**
  - Hybrid Best-First Search
- 4 **Parallélisation**
  - Master-Worker
  - Parallel HBFS
  - Quelques résultats
- 5 **Conclusion**
- 6 **Perspectives**



# Conclusion

On vient donc de survoler en 3 points :

- Le formalisme WCSP,
- Le HBFS séquentiel
- Le paradigme Master-Worker,
- Le Parallel HBFS,
- Quelques résultats obtenus.





# Perspectives

Par ordre croissant de priorité, on peut notamment envisager les 3 pistes suivantes :

- ❶ Les speedups décevants sont imputables pour une part au majorant de Amdahl  $\Rightarrow$  réduire la fraction séquentielle e.g. étude de la parallélisation des pré-traitements,
- ❷ Améliorer la *scalability* ou échelonnabilité : explorer d'autres paradigmes qui augmentent le nb de masters ou permettent les com. entre workers.
- ❸ Etude de la parallélisation du BTD-HBFS qui prend en compte la structure du problème.



## S'il reste du temps ...

Démonstration HBFS vs PHBS sur serveur 24 cœurs.