

# Rapport de Projet

HAI916I

Tom Simula  
Daniel Azevedo Gomes  
Sébastien Prud'homme Gateau

**Git :** [https://github.com/TomSimula/CP\\_Airplane](https://github.com/TomSimula/CP_Airplane)

## **Réponses aux questions :**

**1 et 2 :** utiliser l'option '-bi' ou '-br' et spécifier l'instance de test précédé de '-i'

**4 et 5 :** spécifier simplement l'instance de test précédé de '-i' et '-a' pour chercher toutes les solutions

### **Question 3 :**

Modélisation sous forme d'un réseau de contraintes (X,D,C)

$X = \{\text{position des } N \text{ diviseurs}\}$

$D = \{[0, \dots, M]\}$  où M est égal au nombre de blocs.

$C = \{$

Pas séparateurs au niveau des sorties de secours,

La classe à l'avant de l'avion doit disposer d'au moins deux blocs,

Les distances entre les séparateurs doivent être toutes différentes.

Il doit y avoir un diviseur à la position 0 et un second à la position M.

$\}$

### **Question 6 :**

#### **optimisations sur les contraintes :**

- **Model.increasing** : Permet de chercher une solution contenant les valeurs de position des diviseurs dans l'ordre croissant => Cela évite à la fois d'avoir des diviseurs dans le désordre et d'avoir des redondances de solutions (diviseurs 1 et 2 interchangés par exemple)

### Recherche de stratégies :

- Les stratégies suivantes ont été testées :
  - *minDomLBSearch*,
  - *minDomUBSearch*,
  - *intVarSearchSearch*
  - *inputOrderLBSearch*
  - *inputOrderUBSearch*,
  - *activityBasedSearch*,
  - *randomSearch*

Nous avons pu observer une réduction générale du temps de recherche pour presque toutes les stratégies établies.

voici les résultats de ces stratégies selon différents critères (tests réalisés sur l'instance 4):

Stratégie	temps d'exécution	nombre de backtracks	fails	nombre de noeuds	vitesse de parcours de noeuds (en noeuds par sec)
par défaut	20.648 sec	594 813	297 390	297 423	14404.6 n/s
activityBasedSearch	8.929 sec	189 185	94 551	94 634	10598.1 n/s
inputOrderLBSearch	10.425 sec	207 709	103 838	103 871	9964 n/s
minDomLBSearch	11.853 sec	275 821	137 894	137 927	11.636 n/s
intVarSearchSearch	12.301 sec	184 903	92 435	92 468	7516.9 n/s
inputOrderUBSearch	15.083 sec	424 739	212 353	212 386	14080.9 n/s
minDomUBSearch	15.511 sec	444 123	222 045	222 078	14317.7 n/s
randomSearch	294.935 sec	5 488 341	2 744 154	2 744 187	9 304.4 n/s

On peut remarquer que seule la stratégie aléatoire a un effet négatif sur tous les critères. Les autres ont l'effet inverse et sont toutes meilleures que la stratégie par défaut. Cependant ces résultats peuvent varier selon l'instance, il sera donc nécessaire de tester les stratégies pour chaque cas si on tient à en déduire la plus efficace.