

BESSAÏ Sofiane
FAUCONNIER Axel

Intelligence Artificielle : Rapport de projet

Manuel d'utilisation

Makefile :

- make all pour compiler, cela générera l'exécutable demo
- make all OPTS='-DPRINT_SOL -DPRINT_NOEUD -DPRINT_CONTRAINTES'
 - L'option -DPRINT_SOL affiche les solutions trouvés pendant une recherche
 - L'option -DPRINT_NOEUD affiche le nombre de nœuds visités
 - L'option -DPRINT_CONTRAINTES affiche le nombre de contraintes traitées
 - Pour changer d'options : make clean puis recompiler

Fonctions :

Générateurs :

Génèrent le CSP du problème des pigeons ou des dames à l'ordre n, allocation dynamique des tuples, libérer la mémoire avec la fonction fourni.

- void generate_pigeons(int n, CSP * csp);
- void generate_dames(int n, CSP * csp);
- void free_CSP(CSP * csp);

Algorithmes de résolution :

Ces fonctions renvoient le nombre de solutions du CSP en argument, possibilité d'afficher les solutions, nombres de nœuds visités ou de contraintes traitées avec les options de compilation vues plus haut.

- int backtrack(CSP * csp);
- int forward_checking(CSP * csp);
- int forward_checking_md(CSP * csp);
- int backjumping(CSP * csp) ;
- int backjumping_mc(CSP * csp) ;

Output :

Écrit les différents tableaux d'un CSP dans un fichier.

- void write_csp(const CSP * csp, const char * nom_fichier);

Exemple de fonction main :

```
int main()
{
    CSP csp ;
    int r ;

    generate_dames(8, &csp) ;

    r = forward_checking_md(&csp) ;

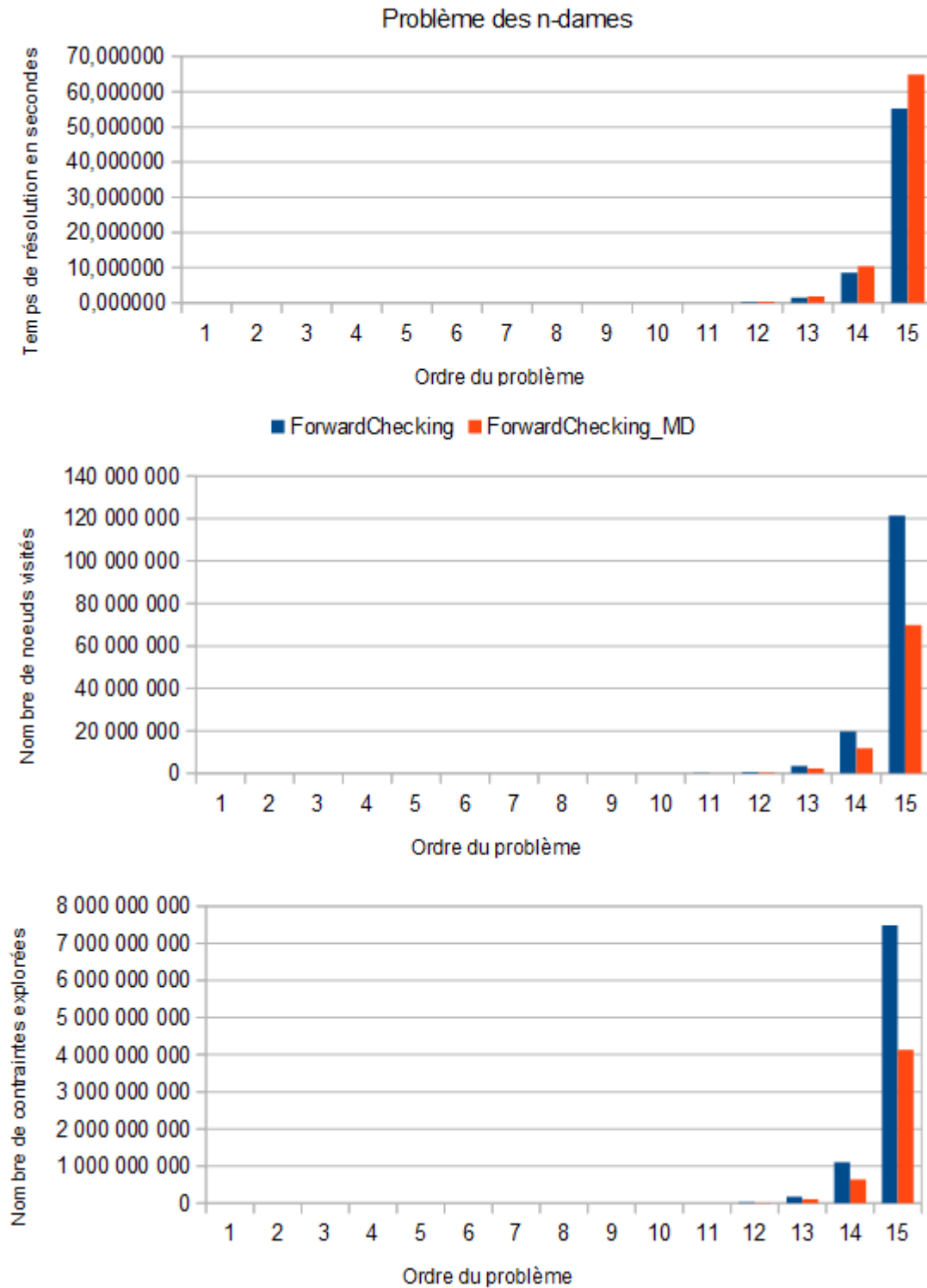
    printf("\nOn dénombre %d solutions à ce CSP\n", r);

    free_CSP(&csp) ;

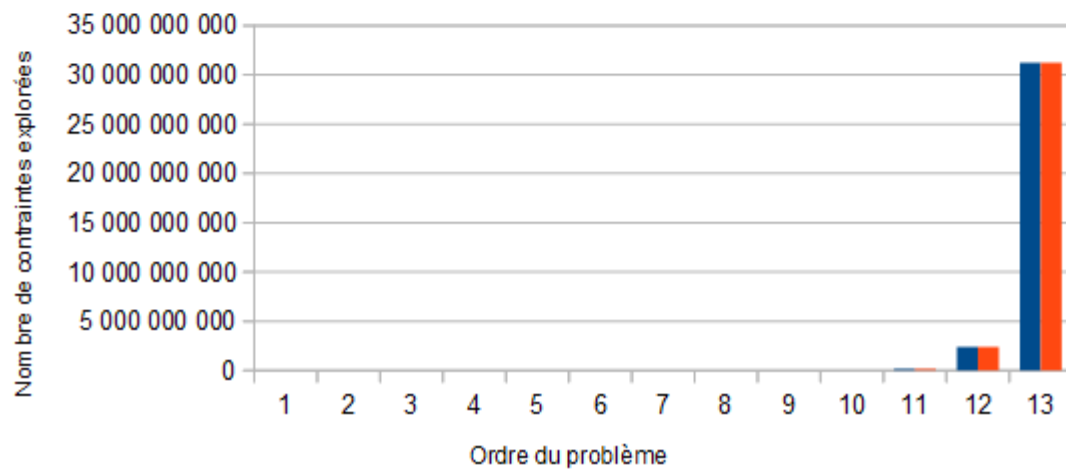
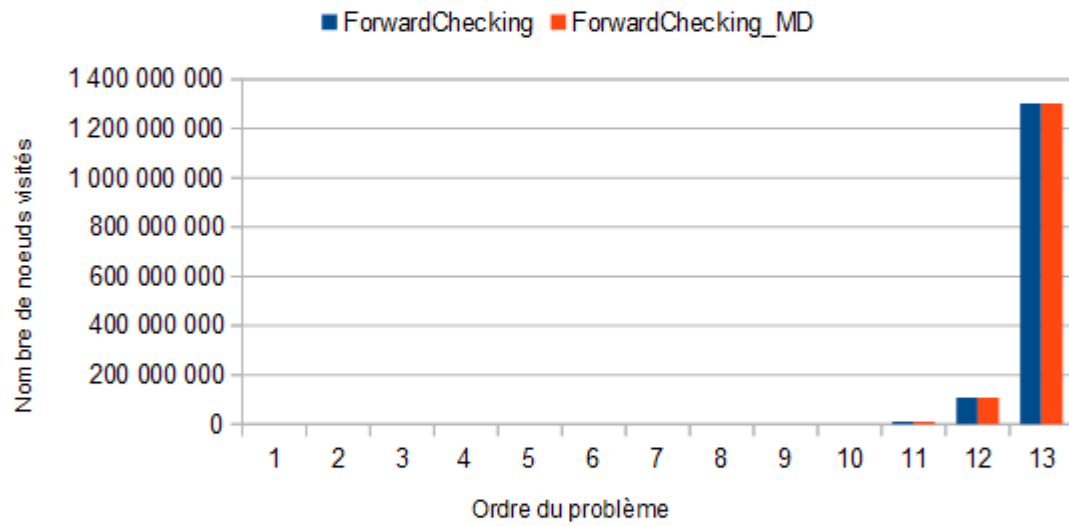
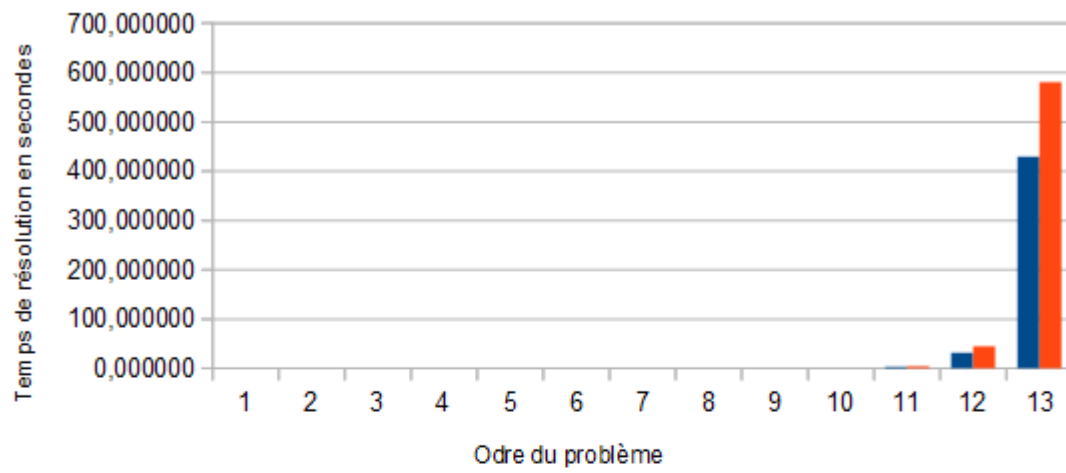
    return 0 ;
}
```

Comparaisons des algorithmes

Comparaison de l'efficacité de Forward-Checking avec et sans heuristique :



Problème des pigeons



Interprétation :

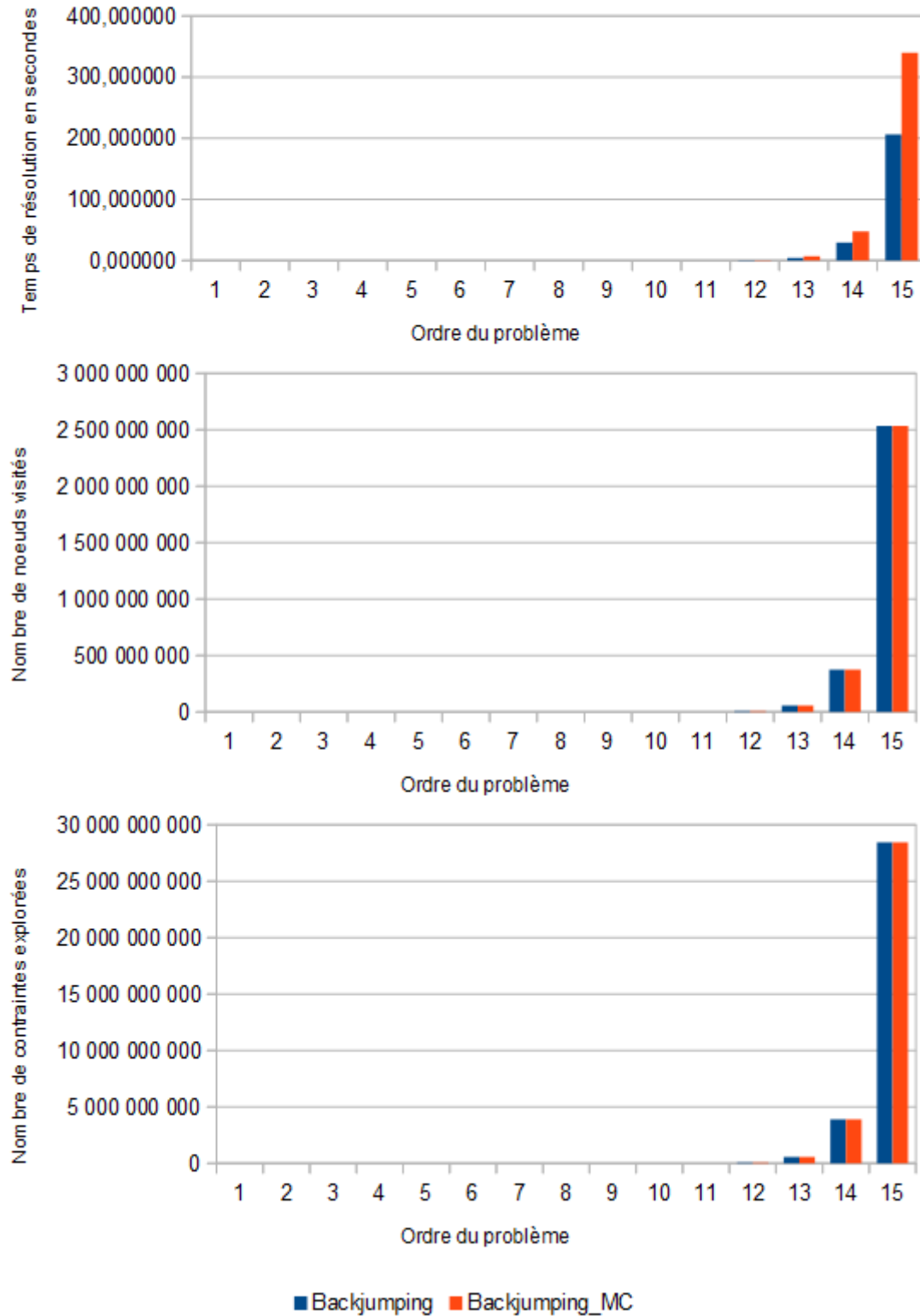
Nous pouvons constater que l'heuristique n'est d'aucune utilité à la résolution du problème des pigeons, en revanche l'heuristique MD divise pratiquement par 2 le nombre de nœuds développés et de contraintes explorées pour le problème des n-dames, malgré cela l'algorithme sans heuristique reste plus rapide.

Nous pensons que cela est principalement dû à une implémentation non optimale de l'algorithme.

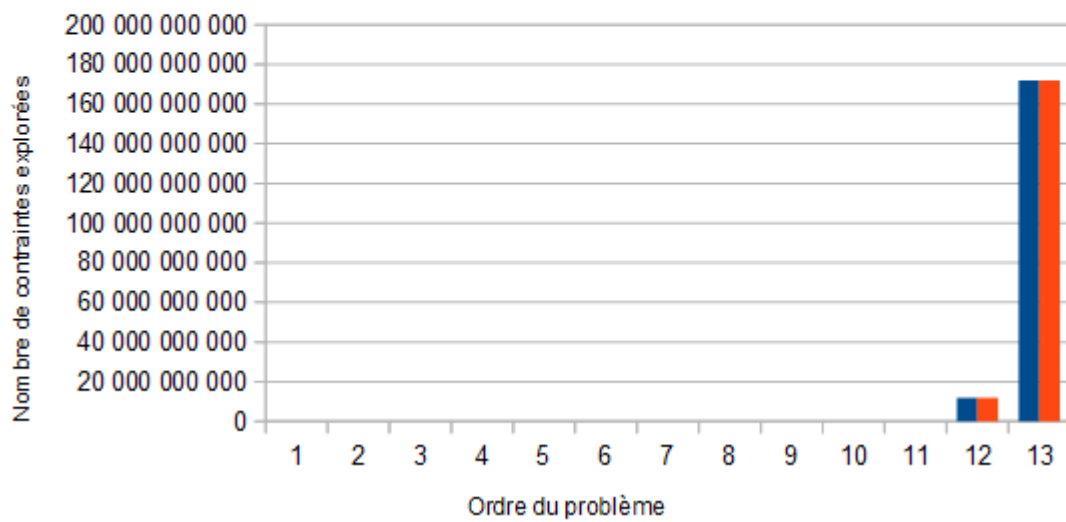
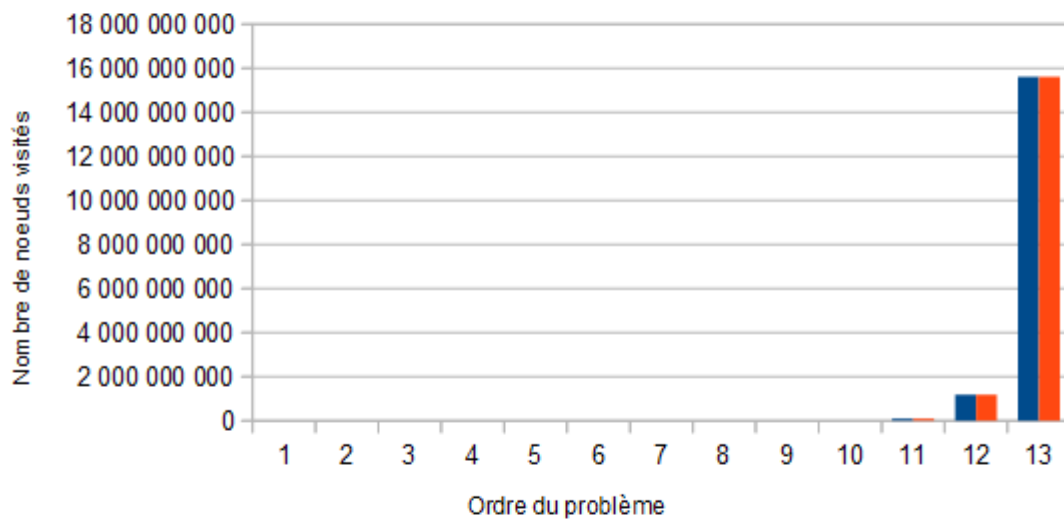
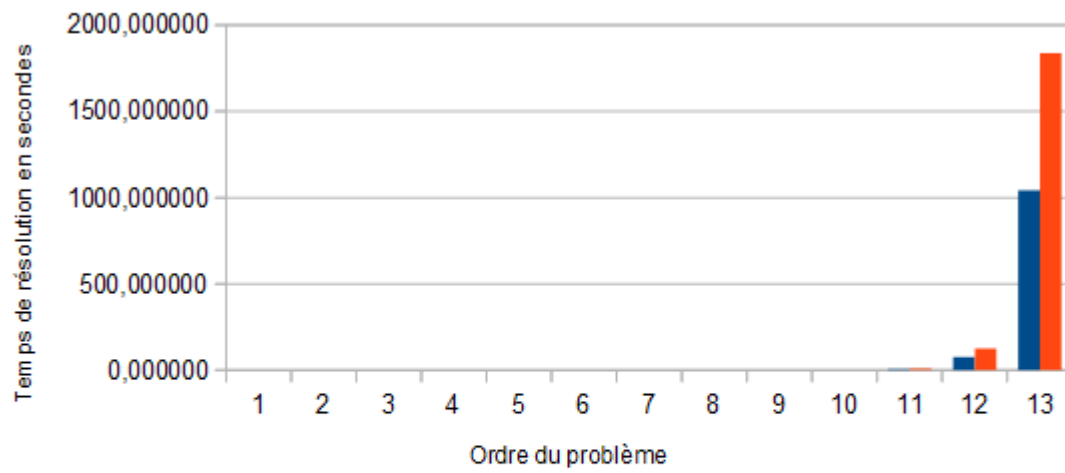
Il est aussi possible qu'à partir d'un ordre $n > 15$, l'algorithme avec heuristique passe devant en temps, en effet le gain de nœuds et de contraintes se fait de manière non linéaire.

Comparaison de l'efficacité entre backjump avec et sans heuristique

Problème des n-dames



Problème des pigeons

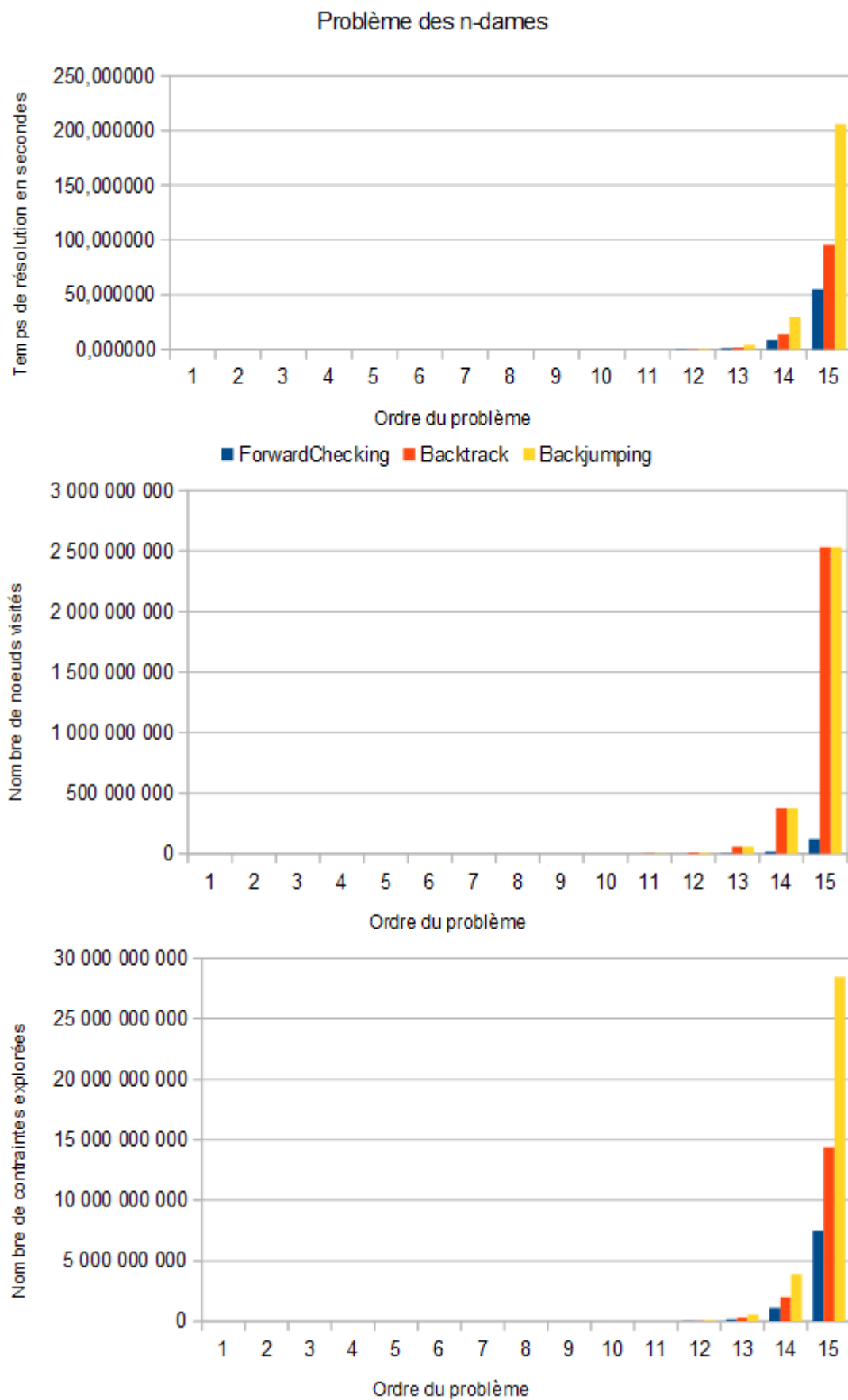


Interprétation :

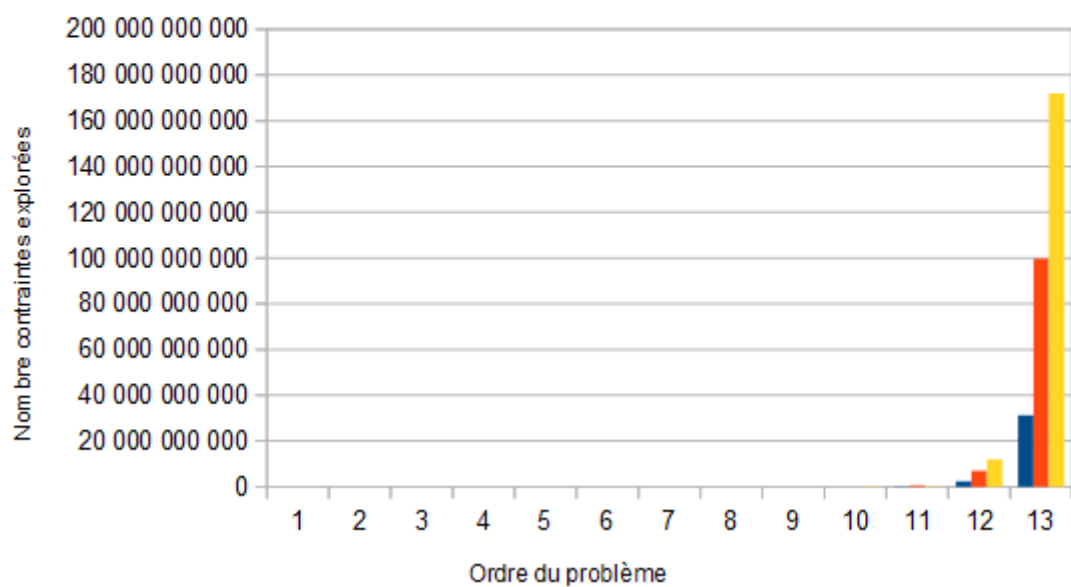
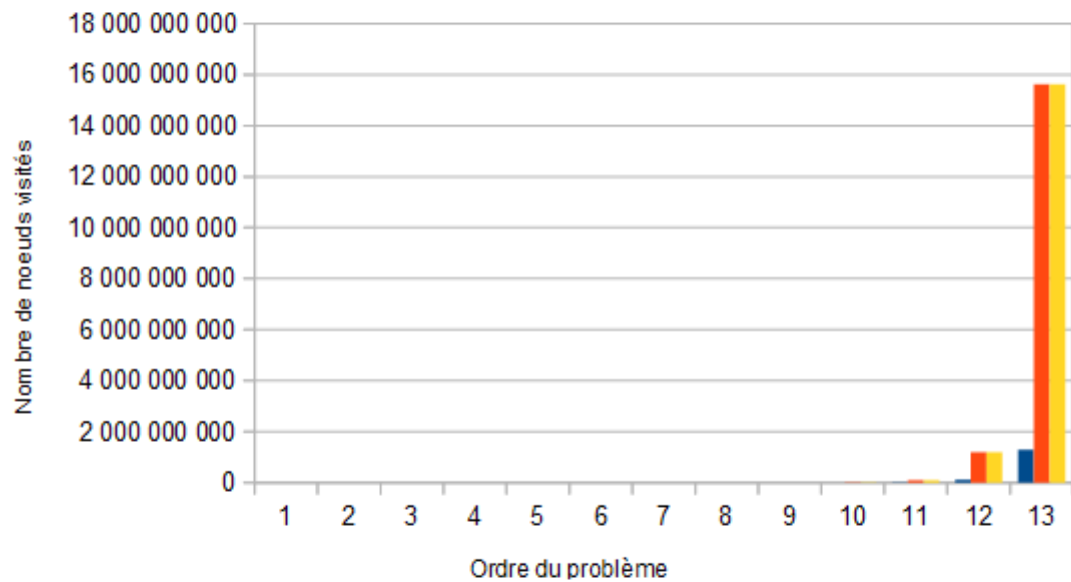
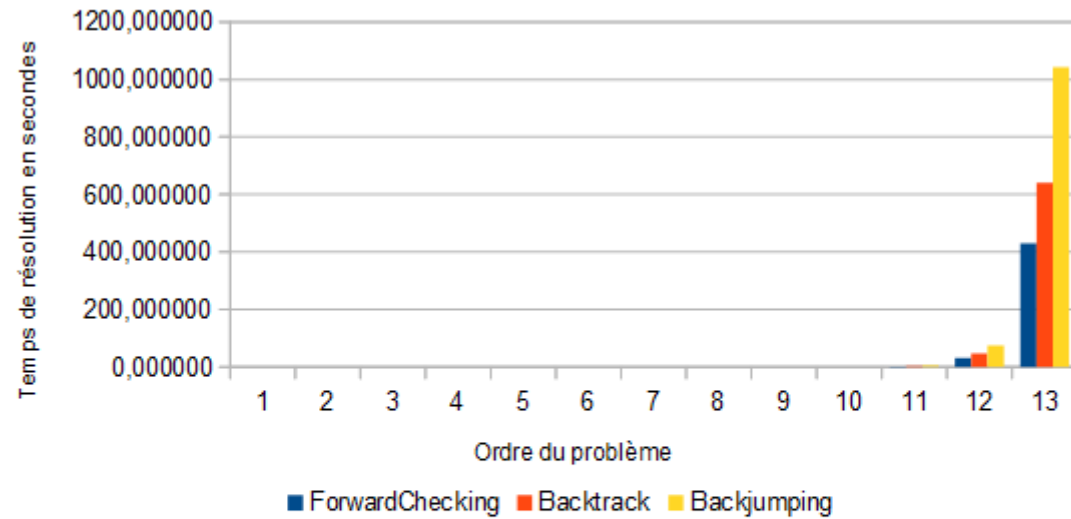
D'après nos résultats, l'heuristique MC n'apporte aucune amélioration en ce qui concerne la résolution de ces CSP puisque le nombre de nœud visités et de contraintes explorées est respectivement le même pour chaque problème.

L'heuristique MC n'a aucune influence sur ces CSP a contrainte de type AllDiff

Comparaison entre Forward-Checking et Backjumping :



Problème des pigeons



Interprétation :

Nous avons choisi d'inclure dans ces graphiques les résultats donnés par l'algorithme de Backtrack simple car la comparaison est intéressante.

Nous constatons donc bien évidemment la supériorité de Forward-Checking face à Backjumping, on remarque que Backjumping et Backtrack partagent le même nombre de nœuds développés, de plus Backjumping traite presque 2 fois plus de contraintes que Backtrack.

Ce résultat étonnant proviendrait d'une implémentation non optimale de l'algorithme de Backjumping.

Le graphique des nœuds montre que Backtrack et Backjumping développe un arbre identique, donc le Backjump intelligent de l'algorithme Backjumping effectue au final un Backtrack classique. Selon nous, ce phénomène viendrait des contraintes Alldiff qui lient l'ensemble des variables de chacun des problèmes.

Conclusion :

Après analyse des nos résultats, nous pouvons affirmer que l'algorithme Forward-Checking est plus performant que celui de Backjumping notamment par sa capacité à éviter le développement d'un grand nombre de nœuds.

Référence :

Tout nos résultats sont disponibles dans le fichier 'Resultat-IA.ods'.