IA - DPLL

Julian Hurst

1 Utilisation

./dpll -x problème n fichier
permet d"exporter un fichier de clauses pour le problème (pigeon ou dames) donné pour n.
./dpll fichier heuristique
permet de lancer dpll sur l'ensemble de clauses du fichier "fichier".
Ces options peuvent être combinées (ex : ./dpll -x pigeon 8 pigeon8 Dames/dames4 3)

2 Types de fichier

Fichier forme char

Ces fichiers contiennent un ensemble de clauses écrites en caractères. Par exemple :
a b a -b Pour (a v b) n (a v -b)

Fichier forme int

Ces fichiers contiennent un ensemble de clauses écrites en entiers. Par exemple :
1 3 1 4 Pour (a v b) n (a v -b)

3 Structures de données

lit

lit est un ensemble de littéraux et contient un tableau d'entiers censé representer les littéraux et une variable int size qui représente la taille de l'ensemble de littéraux.

clause

clause est un ensemble de clauses qui contient un tableau d'ensembles de littéraux et, comme lit, une variable int size qui représente le nombre de clauses.

4 Principales fonctions

clause * readficlit(char *file)

Cette fonction lit un fichier file (sous la forme char), stocke les clauses dans une structure clause * et la renvoie.

clause * readficnum(char *file)

Cette fonction lit un fichier file (sous la forme int), stocke les clauses dans une structure clause * et la renvoie.

int dpll(clause *cl, int h, clause *solutions)

dpll implémente l'algorithme dpll sur l'ensemble de clauses cl en utilisant l'heuristique h et stocke les résultats dans solutions. Cet algorithme consiste à choisir un mono-littéral dans cl, s'il n'en existe pas il choisit un littéral pur. Dans le cas où il n'y a ni un mono-littéral, ni un littéral pur, un littéral est choisit en utilisant l'heuristique h. Si l'ensemble de clauses est devient incohérant on revient au dernier littéral choisit par l'heuristique et on applique son opposé. Si aucun littéral n'a été choisit par heuristique, ou qu'on est arrivé à la fin des possibilités de solution, la fonction renvoie 0. Si au moins une solution est trouvée la fonction renvoie 1 si elle arrive à la fin des possibilités et trouve ainsi toutes les solutions.
Plusieurs variables permettent à cette fonction de faire du backtracking ou de garder des informations intéressantes :

- lit taken : contient tous les littéraux de la branche actuelle. Cette variable permet de revenir à une configuration de clauses précédente grâce à la fonction rollback (voir plus bas).
- lit takenrand : contient tous les littéraux pris par heuristique dans l'ordre de l'arbre. takenrand permet de savoir si on est à la fin des possibilités de solution pour cl grâce à la fonction back (voir plus bas).
- lit taken_all : contient tous les littéraux de l'arbre et permet de savoir combien de noeuds ont été parcourus.

int inconsistent(clause cl)

Cette fonction vérifie si l'ensemble de clauses cl est cohérant, c'est-à-dire si on a une clause vide (par ex : a et -a).

int back(lit takenrand)

Cette fonction va servir à savoir quand il faut revenir à un ensemble de clauses plus ancien dans l'algorithme dpll en regardant s'il existe un littéral pur dans takenrand.
Si oui la fonction renvoie l'opposé du littéral (nouveau littéral à tester) sinon elle renvoie 0.

int findlit(int x, lit taken)

Findlit cherche un littéral parmi un ensemble de littéraux et renvoie son indice. Si la valeur n'est pas trouvée, -1 est renvoyé.

void choicelit(int l, clause *cl)

Cette fonction applique le choix d'un littéral l sur l'ensemble de clauses cl (enlève toutes les clauses contenant l et -l de toutes les clauses).

clause * rollback(clause *cl, lit taken, int b)

Rollback applique choicelit sur tous les littéraux de taken jusqu'à l'indice b. En ce faisant on peut effectuer le backtrack de l'ensemble de clauses.

int monolit(clause cl)

Cette fonction renvoie le premier mono-littéral trouvé dans l'ensemble de clauses cl.

int veriflitpurs(clause cl)

Cette fonction renvoie le premier littéral pur trouvé dans l'ensemble de clauses cl.

void copylit(clause *cl, lit taken, int s)

Copylit copie tous les littéraux de taken dans la clause s de cl (cl->C[s])

void pigeongen(char *file, int n)

Cette fonction génère un fichier file pour le problème des n-pigeons.

void damesgen(char *file, int n)

Cette fonction génère un fichier file pour le problème des n-dames.

5 Heuristiques

int firstsatisfy(clause cl)

Cette fonction est la première heuristique. Elle consiste à choisir en premier, le littéral qui apparaît le plus dans la base de clauses.

int firstfail(clause cl)

Cette fonction est la première heuristique. Elle consiste à choisir en premier, le littéral dont l'opposé apparaît le plus dans la base de clauses.

int firstfailbis(clause cl)

Cette fonction est la première heuristique. Elle consiste à choisir en premier, le littéral dont l'opposé apparaît le plus dans les clauses les plus courtes (de tailles petites). Dans ce cas l'heuristique prend les clauses de taille inférieure à la taille de la plus grande clause moins un. Soit $max(taille(cl)) - 1$

6 Temps de calcul

Les calculs sont fait sur un ordinateur :
Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz
16GO de RAM
Les affichages sont redirigés afin de prendre en compte que le temps de calcul. Les commandes effectuées sont :
./dpll fichier heuristique > /dev/null

Pas d'heuristique (Choix du premier littéral positif)

fichier	temps
Pigeons/pigeon5	0.002s
Pigeons/pigeon6	0.008s
Pigeons/pigeon7	0.055s
Pigeons/pigeon8	0.653s
Pigeons/pigeon9	8.248s
Pigeons/pigeon10	2m02.99s=122.99s

firstsatisfy

fichier	temps
Pigeons/pigeon5	0.002s
Pigeons/pigeon6	0.007s
Pigeons/pigeon7	0.057s
Pigeons/pigeon8	0.639s
Pigeons/pigeon9	8.288s
Pigeons/pigeon10	2m03.37s=123.37s

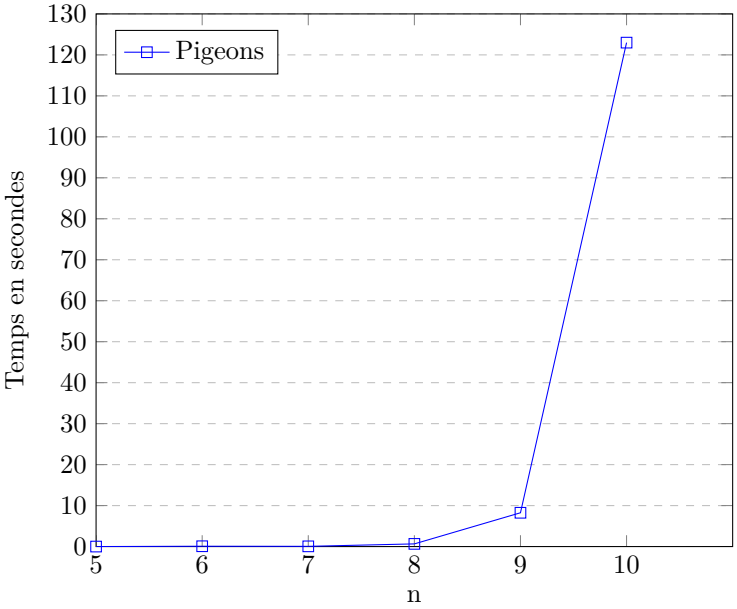
firstfail

fichier	temps
Pigeons/pigeon5	0.002s
Pigeons/pigeon6	0.007s
Pigeons/pigeon7	0.057s
Pigeons/pigeon8	0.619s
Pigeons/pigeon9	8.425s
Pigeons/pigeon10	2m00.90s=120.90s

firstfailbis

fichier	temps
Pigeons/pigeon5	0.002s
Pigeons/pigeon6	0.007s
Pigeons/pigeon7	0.055s
Pigeons/pigeon8	0.621s
Pigeons/pigeon9	8.329s
Pigeons/pigeon10	2m02.32s=122.32s

Temps de calcul pour DPLL sur Pigeons



Temps de calcul pour DPLL sur Dames