

Notes sur « SATLIB - Benchmark Problems »

Lien : <https://www.cs.ubc.ca/~hoos/SATLIB/benchm.html>

Plusieurs problèmes de contrainte abordés dans ce lien, on élimine ceux qui sont aléatoires

"Flat" Graph Colouring

Lien description : <https://www.cs.ubc.ca/~hoos/SATLIB/Benchmarks/SAT/GCP/descr.html>

- [flat30-60](#): 30 vertices, 60 edges - 100 instances, all satisfiable
- [flat50-115](#): 50 vertices, 115 edges - 1000 instances, all satisfiable
- [flat75-180](#): 75 vertices, 180 edges - 100 instances, all satisfiable
- [flat100-239](#): 100 vertices, 239 edges - 100 instances, all satisfiable
- [flat125-301](#): 125 vertices, 301 edges - 100 instances, all satisfiable
- [flat150-360](#): 150 vertices, 360 edges - 100 instances, all satisfiable
- [flat175-417](#): 175 vertices, 417 edges - 100 instances, all satisfiable
- [flat200-479](#): 200 vertices, 479 edges - 100 instances, all satisfiable

graphe $G = (V, E)$ avec $V = \{v_1, v_2, \dots, v_n\}$ un ensemble de sommets et E un ensemble d'arc connectant ces sommets

on doit trouver une coloration telle que les sommets connectés ont toujours une couleur différente les uns des autres

but pour le pb d'optimisation : trouver une coloration avec un nombre minimal de couleurs

but pour le pb de décision : vérifier si une coloration du graphe avec cette règle existe pour nb couleurs

tous ces sets ont été satisfiables avec 3 couleurs

représentable ou non avec une interface tangible :

peut-être dynamique

pas forcément possible de tout représenter

peut-être sous-graphe par sous-graphe en voyant les combinaisons possibles pour chaque sous-graphe et en les comparant

peut-être utiliser transBOARD pour des problèmes de graphe

"Morphed" Graph Colouring

Lien description : <https://www.cs.ubc.ca/~hoos/SATLIB/Benchmarks/SAT/SW-GCP/descr.html>

- [sw100-8-lp0-c5](#): 100 vertices, 400 edges, $p=1$ - 100 instances, all satisfiable
- [sw100-8-lp1-c5](#): 100 vertices, 400 edges, $p=2^1$ - 100 instances, all satisfiable
- [sw100-8-lp2-c5](#): 100 vertices, 400 edges, $p=2^2$ - 100 instances, all satisfiable
- [sw100-8-lp3-c5](#): 100 vertices, 400 edges, $p=2^3$ - 100 instances, all satisfiable
- [sw100-8-lp4-c5](#): 100 vertices, 400 edges, $p=2^4$ - 100 instances, all satisfiable
- [sw100-8-lp5-c5](#): 100 vertices, 400 edges, $p=2^5$ - 100 instances, all satisfiable

- [sw100-8-lp6-c5](#): 100 vertices, 400 edges, $p=2^6$ - 100 instances, all satisfiable
- [sw100-8-lp7-c5](#): 100 vertices, 400 edges, $p=2^7$ - 100 instances, all satisfiable
- [sw100-8-lp8-c5](#): 100 vertices, 400 edges, $p=2^8$ - 100 instances, all satisfiable
- [sw100-8-p0-c5](#): 100 vertices, 400 edges, $p=0$ - 1 instance, satisfiable

graphe $G = (V, E)$ avec $V = \{v_1, v_2, \dots, v_n\}$ un ensemble de sommets et E un ensemble d'arc connectant ces sommets

on doit trouver une coloration telle que les sommets connectés ont toujours une couleur différente les uns des autres

but pour le pb d'optimisation : trouver une coloration avec un nombre minimal de couleurs

but pour le pb de décision : vérifier si une coloration du graphe avec cette règle existe pour nb couleurs

tous ces sets ont été satisfiables avec 5 couleurs

NB : vérifier si ces sets seraient satisfiables avec un nombre de couleurs plus bas

représentable ou non avec une interface tangible :

peut-être dynamique

pas forcément possible de tout représenter

peut-être sous-graphe par sous-graphe en voyant les combinaisons possibles pour chaque sous-graphe et en les comparant

peut-être utiliser transBOARD pour des problèmes de graphe

Planning

Blocksworld

Lien description :

<https://www.cs.ubc.ca/~hoos/SATLIB/Benchmarks/SAT/PLANNING/BlocksWorld/descr.html>

[blocksworld](#): 7 instances, all satisfiable

problème de planification

une table et N blocs

ces blocs peuvent être mis l'un sur l'autre, avec le bloc le plus bas toujours sur la table

opération possible : mettre le plus haut bloc d'une pile au-dessus d'une autre pile ou sur la table

soit une configuration initiale de blocs et une configuration voulue

le but est de trouver une suite d'opérations permettant d'arriver à la configuration voulue

→ un plan linéaire

les blocs ne peuvent être déplacés que s'il n'y a aucun autre bloc dessus

et ils ne peuvent être mis que sur des blocs sans autre bloc dessus ou sur la table

but optimisation : trouver la suite d'opérations la plus courte

but décision : vérifier si une suite de M opérations existe

subtilité par rapport à Hanoi : on peut mettre les piles sur toute la table et il y a pas de système de « le plus grand doit pas aller sur le petit »

représentable ou non avec une interface tangible :

probablement représentable

notamment avec une interface tangible impliquant des blocs, pour un problème Blocks World entier ou pour un sous-problème

Logistics

Lien description : <https://www.cs.ubc.ca/~hoos/SATLIB/Benchmarks/SAT/PLANNING/Logistics/descr.html>

logistics: 3 instances, all satisfiable

paquets doivent être déplacés entre différents endroits de différentes villes
dans villes : paquets dans des trucks
hors des villes : paquets dans des avions
trucks et avions ont une capacité limitée
3 opérations possibles : charger, décharger, déplacer
2 états possibles : dans, à
il y a une configuration initiale de paquets, trucks et avions
on veut arriver à une configuration voulue de paquets, trucks et avion
il peut y avoir plusieurs actions en même temps tant qu'il n'y a pas de conflit par rapport à leurs préconditions et effets
but pb d'optimisation : trouver le plan le plus court pour aller de la configuration initiale à la configuration voulue
but pb de décision : vérifier si un plan d'une certaine longueur existe

représentable ou non avec une interface tangible :
probablement représentable

All Intervall Series

Lien description : <https://www.cs.ubc.ca/~hoos/SATLIB/Benchmarks/SAT/AIS/descr.html>

ais: 4 instances, all satisfiable

pb arithmétique inspiré par pb musical

pb musical :
12 classes standards de hauteur musical représentés par nombres entiers de 0 à 11
trouver une série dans laquelle chaque classe apparaît exactement 1 fois
et dans laquelle les intervalles musicaux entre notes voisines couvrent l'ensemble des intervalles depuis la seconde mineure (1 demi-ton) jusqu'à la septième majeure (11 demi-tons)
c-à-d : pour chaque intervalle, il y a une paire d'intervalles voisines entre lesquelles cette intervalle-là apparaît

pb arithmétique :
 n : nombre entier
 Z_n : ensemble d'entiers naturels de 0 à $n-1$ (donc $Z_n = \{0, 1, \dots, n-1\}$)
trouver $s = (s_1, \dots, s_n)$ tel que :
- s est une permutation de Z_n
- le vecteur interne $v = (|s_2 - s_1|, \dots, |s_n - s_{n-1}|)$ est une permutation de $Z_n - \{0\} = \{1, \dots, n-1\}$

représentable ou non avec une interface tangible :
c'est peut-être possible, ce serait bien d'essayer d'y réfléchir

SAT-encoded Quasigroup (or Latin square) instances

Lien description : <https://www.cs.ubc.ca/~hoos/SATLIB/Benchmarks/SAT/QG/qg.descr.html>

[qg](#): 22 instances

ensemble S

un rectangle latin indexé par S : un tableau $|S| \times |S|$ tel que chaque colonne et chaque ligne du tableau est une permutation des éléments de S

$|S|$ ordre du rectangle latin

pour un rectangle latin d'ordre v, contraintes suivantes de base :

(1) $(x * u = y, x * w = y) \implies u = w$

(2) $(u * x = y, w * x = y) \implies u = w$

(3) $(x * y = u, x * y = w) \implies u = w$

(4) $(x * y = 0) \mid (x * y = 1) \mid \dots \mid (x * y = (v-1))$

contraintes supplémentaires peuvent être ajoutées

yes : IIIII IIIII \rightarrow 10

no : IIIII IIIII II \rightarrow 12

sur 22 instances, 12 n'ont pas pu être satisfaites, ça pourrait être bien de voir pourquoi

représentable ou non avec une interface tangible :

à voir, mais je ne pense pas que ce soit représentable par une interface tangible

SAT-encoded bounded model checking instances

Lien description : <https://www.cs.ubc.ca/~hoos/SATLIB/Benchmarks/SAT/BMC/description.html>

[bmc](#): 13 instances

vérifier si un modèle M (souvent un design hardware) satisfait une propriété temporelle P

dans tous les chemins avec longueur $\leq k$

problème BMC peut être réduit à un problème de satisfiabilité propositionnelle

si propriété est sous forme d'invariant, alors elle a une structure similaire à celle de plusieurs

problèmes de planification AI

structure générale de la formule de l'invariant BMC :

$$I_0 \wedge \bigwedge_{i=0}^{k-1} p(i, i+1) \wedge \bigvee_{i=0}^k \neg P_i$$

où :

- I_0 est l'état initial

- $p(i, i+1)$ est une formule représentation la transition entre les cycles i et i+1

- P_i est la propriété dans le cycle i

cette formule ne peut être satisfaite que s'il existe un état atteignable dans un cycle i ($\leq k$) qui contredit la propriété P_i

13 instances toutes résolues selon certaines méthodes

représentable ou non avec une interface tangible :

à voir, mais je ne pense pas que ce soit représentable par une interface tangible

DIMACS Benchmark Instances

Large SAT-encoded Graph Colouring problems

GCP: Large SAT-encoded Graph Colouring problems - 4 instances, all satisfiable [description \(html\)](#)

même chose que pour "Flat" Graph Colouring et "Morphed" Graph Colouring

Instances for problem in learning the parity function

PARITY: Instances for problem in learning the parity function - 20 instances, all satisfiable [description \(html\)](#)

identifier une fonction booléennes inconnues avec l'aide d'échantillons I/O de la fonction
une fonction correspondant à ce problème n'a pas besoin d'être une fonction satisfaisant toutes ces
samples

tant que la fonction en question satisfait une limite maximale d'erreurs commises sur les
échantillons

représentable ou non avec une interface tangible :

à voir, mais je ne pense pas que ce soit représentable par une interface tangible

Instances from a problem in inductive inference

II: Instances from a problem in inductive inference - 41 instances, all satisfiable [description \(html\)](#)

voir sous-partie précédente

SAT-encoding of Towers of Hanoi

HANOI: SAT-encoding of Towers of Hanoi - 2 instances, all satisfiable [description \(html\)](#)

3 tours et n disques de diamètre décroissant sur une des tours

il faut déplacer ces disques d'un tour vers l'autre selon les règles suivantes :

(1) un mouvement place un disque sur une autre tour

(2) seul un disque à la fois peut être déplacés

(3) un disque ne peut pas être déplacé au-dessus d'un plus petit disque

but : trouver une séquence de mouvements permettant de passer de la configuration initiale au but

représentable ou non avec une interface tangible :

probablement représentable avec une interface tangible dynamique

Circuit fault analysis: bridge fault

BF: Circuit fault analysis: bridge fault - 4 instances, all unsatisfiable [description \(html\)](#)

représentable ou non avec une interface tangible :

Circuit fault analysis: single-stuck-at fault

[SSA](#): Circuit fault analysis: single-stuck-at fault - 4 instances satisfiable, 4 instances unsatisfiable [description \(html\)](#)

représentable ou non avec une interface tangible :

Pigeon hole problem

[PHOLE](#): Pigeon hole problem - 5 instances, all unsatisfiable [description \(html\)](#)

vérifier s'il est possible de mettre $n+1$ pigeons dans n trous sans que 2 pigeons soient dans le même trou

bien entendu, problème insoluble

représentable ou non avec une interface tangible :

probablement représentable avec une interface tangible

possiblement avec l'aide de boîtes et de blocs

mais j'ai du mal à en voir l'intérêt

Encoded 2-colouring forced to be unsatisfiable

[PRET](#): Encoded 2-colouring forced to be unsatisfiable - 8 instances, all unsatisfiable [description \(html\)](#)

Note à moi-même : « forced to be unsatisfiable » comme si l'idée que ce soit irrésoluble fasse partie des buts