

Modélisation et résolution pour la décision – TP

On s'intéresse à la modélisation et la résolution par SAT et CSP du problème de Schur. Le problème consiste à mettre n balles étiquetées de 1 à n dans k ($k \geq 3$) boîtes de sorte que pour tout triplet de balles x, y, z , tel que $x + y = z$, les trois balles ne sont pas toutes dans la même boîte.

1. Écrire un programme avec un langage de votre choix qui, étant données les valeurs de n et k , génère l'instance SAT correspondante au format DIMACS.
2. Pour des valeurs de k et n , générer les instances SAT correspondantes au format DIMACS.
3. Utiliser le solveur SAT *minisat 2.2* disponible depuis le lien <http://minisat.se/MiniSat.html> pour déterminer la satisfiabilité de ces instances.
4. Écrire un deuxième programme qui, étant données les valeurs de n et k , génère l'instance CSP correspondante au format XCSP3.
5. Utiliser un solveur CSP de la compétition XCSP3 pour déterminer s'il existe une solution à ces instances.
6. Écrire un troisième programme qui prend en entrée la solution générée par le solveur SAT (respectivement CSP) et fournit en sortie le contenu de chaque case sous la format suivant : chaque ligne représente le contenu d'une seule boîte (k lignes au total donc) et affiche les numéros des balles, séparés par des espaces, contenues dans la boîte.

Instances et paramètres expérimentaux

Le tableau suivant indique les instances à générer puis à tester selon les valeurs de n et k . Par ailleurs, la limite de temps accordée par instance est de 600 secondes.

k	n
3	20, 23, 24, 43, 60, 100
4	60, 66, 67, 100
5	140, 150, 160, 170, 171

Règles et rendu

1. Le TP peut être réalisé en binôme.
2. Les codes sources doivent être commentés.
3. Fournir un rapport au format pdf qui inclut :
 - Une description détaillée des modélisations réalisées en SAT et CSP.
 - Le résultat des tests obtenus par les solveurs SAT et CSP en indiquant notamment les temps de calcul ainsi que la configuration matériel utilisée (CPU et mémoire).
 - Des commentaires sur les résultats obtenus et une comparaison des modélisations SAT et CSP au vu de ces résultats.
4. Fournir une archive qui contient les répertoires suivants :
 - `src` contenant les codes sources
 - `sat.instances` contenant les instances SAT générées
 - `csp.instances` contenant les instances CSP générées
5. Le rendu doit se faire via AMETICE