

## REMARQUES

CHARLES PRUD'HOMME

- Les limites ne sont plus actives si on appelle *solver.nextSolution()*;
- Les limites pourraient être regroupées en 1 seul thread.
- Les boolean storables peuvent sans doute être déclarés via un bitset
- Il semble primordial de reconsidérer les *ContradictionExceptions*, il s'agit, sans doute possible, d'un axe d'amélioration! cf. *Sandbox*.
- Sur *TestSolveur#testDecompOpt()*, avec  $n=7$ , où il y a peu de propagation, et énormément de solutions (167772160), quasiment 10% du temps est passé à lever les exceptions (au nombre de 55924054). Pour ce cas précis, une implémentation sans exceptions, mais avec des enums, permet de passer de 157 sec. à 120 sec (23%).
- La "profondeur" d'exécution est également importante. Dans *Sandbox* on voit bien que lever une exception en prof. 7 coûte plus qu'en prof.1. A méditer...
- L'ajout du Delta dans les méthodes du Domain *#updateLowerBound()* et *#updateUpperBound()* a un coût non négligeable. Il faut alors s'assurer que ces deltas sont utilisées!
- Passer *Constraint.NOTIFY* à *true* à un impact sur les performances de Knapsack-Test alors qu'il a l'effet inverse sur QueenTest. Les 2 problèmes ne définissent pas de contraintes communes : KS repose sur SCALAR (GEQ et LEQ) et EQ alors que Q repose sur NEQ. On peut alors se poser la question : le type de notification doit-il être général ou bien particulier (reposant sur l'arité des contraintes par ex.)? Par contre, l'ordre de propagation des événements en serait bouleverser, certains seraient propagés à leur création, d'autres bien plus tard.
- Les performances de GALAK v CHOCO sur un problème d'optimisation sont assez mitigées. Alors que si on compare ces mêmes performances sur le même problème mais où l'on cherche toutes les solutions (avec une limite de temps par ex.), le résultat est sans appel. Etant donné la faible complexité du code en différentiel, c'est assez embêtant ...
- L'ajout du domaine bornée met en évidence l'importance de l'ordre de traitement des événements. Un domaine [1,4] à qui l'on a supprimé (dans cet ordre) les valeurs 2,3,4 devrait alors être instantié à 1, alors que dans GALAK il vaut [1,3]. Dans CHOCO, la mutualisation des événements permet de capter ce comportement.