

Notes sur The Big Deal

Lien de l'article : <https://synergies.univ-tlse3.fr/service/home/~/?auth=co&loc=fr&id=2312&part=2>

Lien de mes annotations : [https://github.com/DelmoteAdrien/stage_interfaces_tangibles/blob/main/recherche/anais/Problemes de contraintes/Biblio reseaux de contraintes/files/260/Naveh%20-%202010%20-%20The%20Big%20Deal%20Applying%20Constraint%20Satisfaction%20Tec.pdf](https://github.com/DelmoteAdrien/stage_interfaces_tangibles/blob/main/recherche/anais/Problemes%20de%20contraintes/Biblio%20reseaux%20de%20contraintes/files/260/Naveh%20-%202010%20-%20The%20Big%20Deal%20Applying%20Constraint%20Satisfaction%20Tec.pdf)

Dans cet article, l'auteur donne des exemples de domaines dans lesquels le CSP peut être utile et propose des manières avec lesquelles le CSP peut aider dans ces domaines

(J'ai mis des annotations mais elles ne sont pas visibles directement dans le Github, il faut télécharger le PDF. J'ai surligné des phrases qui me semblent importantes, heureusement qu'on peut voir ça.)

Domaines évoquées dans lesquels le CSP peut être utile :

- vérification du matériel
- gestion de la main d'œuvre
- configuration de trucks
- ingénierie des systèmes
- placement des machines virtuelles sur des hôtes physiques dans les datacentres
- planification des tâches pour les processeurs massivement parallèles (ou superordinateurs)
- tarification des services d'engagement
- gestion de la variabilité des lignes de produits

Domaines abordés plus en détail dans l'article :

- vérification du matériel
- gestion de la main d'œuvre
- configuration de trucks
- ingénierie des systèmes

2- Hardware Verification (vérification du matériel)

but : vérifier que le matériel fonctionne correctement durant la phase de conception

La première sous-partie (Model Checking) aurait pu m'aider, mais il y a juste un paragraphe pour dire que ce n'est pas la peine de trop en parler car l'audience est déjà familière avec le sujet

2.2- Stimuli Generation (génération du stimuli)

les techniques formelles de vérification ont des avantages
mais elles ont des difficultés avec des conceptions modernes
la vérification par stimulation compte pour 90 % de la vérification des efforts et ressources

le challenge majeur de ces méthodes est de créer des inputs ou « stimuli » qui sont :

- 1- valides selon la spécification du matériel et l'environnement de simulation
- 2- intéressants dans le sens où ils sont plus susceptibles de stimuler des zones de conception sujettes aux bugs
- 3- variés

exigence n°1 (valides) : modélisation la spécification du matériel et l'environnement de simulation en tant qu'ensembles de contraintes obligatoires sur les variables simulés

exigence n°2 (intéressants) :

- 1- modéliser les connaissances génériques des experts en tant qu'ensembles de contraintes « douces » et non obligatoires
- 2- l'ingénieur de vérification peut ajouter des contraintes obligatoires et non obligatoires à toute exécution particulière, dirigeant les stimuli vers les scénarios requis

exigence n°3 (variés) :

le solveur dispose généralement d'un mécanisme de diversification intégré
à chaque fois que le solveur est appelé avec le même ensemble de de contraintes, il donne un différent résultat (stimuli)

3- Workforce Management (gestion de la main d'œuvre)

but : assigner une équipe de professionnels à un projet

critères pour chaque professionnel :

- compétent (mais pas trop) pour le travail
- disponible dans les alentours ou capable de travailler en télétravail
- pas devoir s'occuper d'autres projets avant la fin de celui-ci
- avoir une affinité personnelle pour le travailler

critères pour l'équipe :

- distribution correcte de compétences et de niveau d'expérience
- pouvoir se conformer aux exigences de budget
- employés capables de travailler les uns avec les autres

critère en plus : à tout moment, on doit assigner les employés à autant de projets que possible

peut se traduire en CSP

certaines contraintes ont une fondation mathématique claire
d'autres sont plus soft (critères humains)

exemple du premier type de contraintes :

le même employé ne peut pas être assigné à 2 projets si ces projets se chevauchent dans le temps
→ contrainte globale : some-different

second type de contraintes → ensemble de préférences

ensemble de règles et de préférences défini par l'utilisateur dans une liste
ensuite traduit en ensemble de contraintes dures et douces
résolu par un solveur de contraintes

4- Truck Configuration (configuration de trucks)

les trucks larges (et chers) sont très configurables selon les besoins du clients

il n'y a pas 2 trucks identiques, sauf si le client a demandé plusieurs mêmes trucks

le client a des besoins spécifiques en tête et ne sera pas susceptible de faire des compromis dessus
le truck doit satisfaire les requêtes du client

tout en se conformant aux contraintes légaux, d'ingénierie, de fabrication et de marketing

- milliers de variables configurables
- dizaines de milliers de contraintes dessus

- le problème de configuration est NP-complet
- une configuration valide peut exister mais ne pas être trouvée par le configurateur en main
- => exemple classique dans laquelle une technologie plus forte peut faire la différence

5- Ingénierie des systèmes

conception de systèmes complexes implique plusieurs disciplines différents :

- ingénierie des exigences
- ingénierie électronique
- test
- analyse paramétrique
- ...

dans chaque discipline : un modèle de produits est géré

aujourd'hui, tous ces modèles sont gérés séparément

- limite les possibilités de :
- maintenir une traçabilité entre les différentes équipes et parties de projet
- synchroniser les données
- faire une analyse d'impact lorsqu'une partie du modèle change
- obtenir une conception optimale

problème : règles variées de validité dans lesquelles le modèle doit se conformer

- tenter de lier les différents modèles peut vite donner des inconsistances avec le respect de ces règles

la technologie de satisfaction des contraintes peut :

- s'assurer que les liens soient créés en concordance avec la méthodologie de conception
- détecter les différences entre les modèles
- déduire l'existence ou l'absence de liens
- => aide à la création d'un modèle unifié