ÉCHANTILLONNAGE DE SOLUTIONS PRATIQUEMENT UNIFORME EN PROGRAMMATION PAR CONTRAINTES

JFPC22

Gilles Pesant[1], Claude-Guy Quimper[2], Hélène Verhaeghe[1]

27 Juin 2022

[1] Polytechnique Montréal, Montréal, Canada, *gilles.pesant@polymtl.ca*, *helene.verhaeghe@polymtl.ca*
[2] Université Laval, Québec, Canada, *claude-guy.quimper@ift.ulaval.ca*

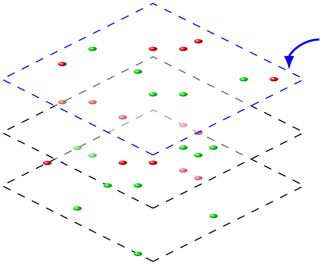
POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

MOTIVATION

MOTIVATIONS

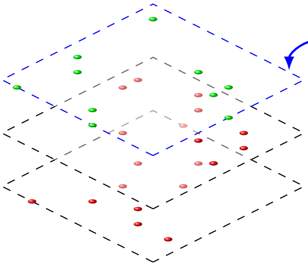
- soit un espace combinatoire contraint (défini par un modèle CP)
- échantillonner ses solutions aléatoirement et uniformément
- e.g. pour les tests de boîte noire matériel/logiciel
  pour l'équité algorithmique

Source: medium.com/hackernoon



$x_i = d_j$ $\quad i, j$ choisi aléatoirement

partitionnement uniforme ($\approx 1/|D_i|$)



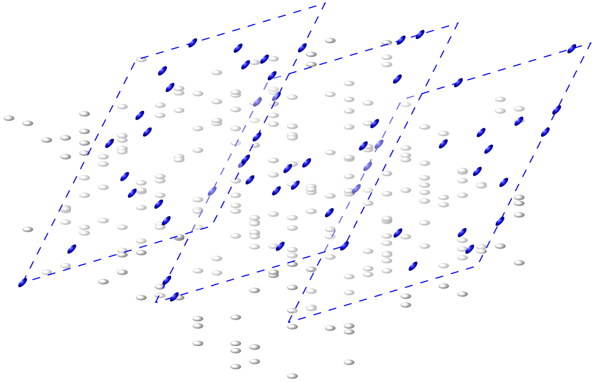
$x_i = d_j$ $\quad$ i, j choisi aléatoirement

pas d'indépendance par pair

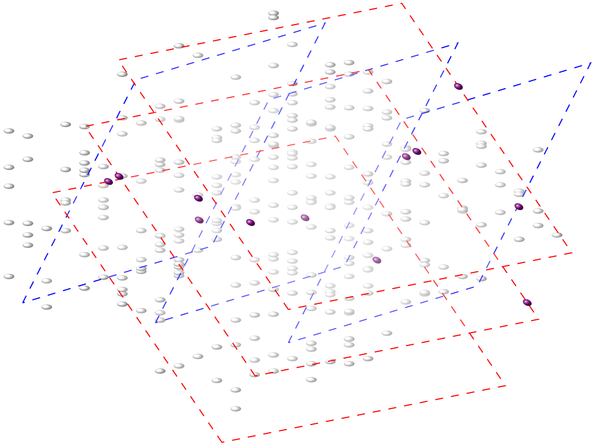
Sur des modèles SAT

- pour le dénombrement de modèles (Chakraborty, Meel, Vardi, CP 2013)
- pour l'échantillonnage (Meel et al., AAAI 2016)
- contraintes de parité aléatoire (XOR)

$\mathbf{a}_1\mathbf{x} \equiv 2 \pmod{5}$ $\Rightarrow$ $\approx$ 1 parmi 5 solutions



$$\mathbf{a_1 x} \equiv 2 \pmod 5 \quad \wedge \quad \mathbf{a_2 x} \equiv 4 \pmod 5 \quad \Rightarrow \quad \approx 1 \text{ parmi 25 solutions}$$



Contraintes d'égalités modulaires linéaires

$$A\boldsymbol{x} + \boldsymbol{b} \equiv \boldsymbol{c} \pmod{p}$$

système de m égalités sur n variables de domaine fini $\boldsymbol{x}$
matrice A et vecteur $\boldsymbol{b}$ de coefficiants aléatoires parmis $\{0, 1, \ldots, p-1\}$
modulo premier $p \geq$ taille du domaine

Chakraborty et al., Approximate probabilistic inference via word-level counting, AAAI 2016
Pour deux solutions $\boldsymbol{x_1}$ et $\boldsymbol{x_2}$,

partitionnement uniforme: $\Pr[A\boldsymbol{x_1} = \boldsymbol{b} \pmod{p}] = \frac{1}{p^m}$

indépendance par pair: $\Pr[A\boldsymbol{x_1}{=}\boldsymbol{b} \pmod{p} \mid A\boldsymbol{x_2}{=}\boldsymbol{b} \pmod{p}]{=}\frac{1}{p^m}$

Contraintes d'inégalités modulaires linéaires

$$A\boldsymbol{x} + \boldsymbol{b} \leq \boldsymbol{c} \pmod{p}$$

système de m inégalités sur n variables de domaine fini $\boldsymbol{x}$
matrice A et vecteur $\boldsymbol{b}$ de coefficiants aléatoires parmis $\{0, 1, \ldots, p-1\}$
modulo premier p $\geq$ taille du domaine

Pesant et al., On the usefulness of linear modular arithmetic in constraint programming, CPAIOR 2021

Pour deux solutions $\boldsymbol{x_1}$ et $\boldsymbol{x_2}$,

partitionnement uniforme: $\Pr[A\boldsymbol{x_1} + \boldsymbol{b} \leq \boldsymbol{c} \pmod{p}] = \frac{\prod_{i=1}^{m}(c_i+1)}{p^m}$

indépendance par pair: $\Pr[A\boldsymbol{x_1}+\boldsymbol{b}\leq\boldsymbol{c} \pmod{p} \mid A\boldsymbol{x_2}+\boldsymbol{b}\leq\boldsymbol{c} \pmod{p}]=\frac{\prod_{i=1}^{m}(c_i+1)}{p^m}$

APPROCHE D’ÉCHANTILLONNAGE UNIFORME

Idée globale

Soit un modèle CP φ,

1. déterminer le mélange d'égalités et inégalité requis suivant la fraction d'échantillonnage λ
2. ajout des contraintes linéaire modulaire à φ (réduction de l'espace de solution)
3. enumération des solutions de l'espace réduit

POLYTECHNIQUE MONTRÉAL

Etape 1: Trouver le bon mélange

$$\frac{\prod_{i=1}^{m'}(c_i+1)}{p^m} \approx \lambda, \quad m' \leq m,$$

On cherche, par valeur de m croissante, une factorisation d'un numérateur entier proche de $\lambda \times p^m$

Example

$\lambda = 0.02, p = 11 \quad \Rightarrow \quad m = 3, F = \langle 9, 3\rangle$

avec une erreur sous 1.5% ($\frac{9\times 3}{11^3} = 0.020285$)

POLYTECHNIQUE MONTRÉAL

Etape 2: Ajout des contraintes

...en pratique, sous forme de contraintes tables

Example

$$p = 11, m = 3, F = \langle 9, 3 \rangle$$

$$\Downarrow$$

$$\begin{aligned} \mathbf{a_1 x} &= b_1 \pmod{11} \\ \mathbf{a_2 x} + b_2 &\leq 8 \pmod{11} \\ \mathbf{a_3 x} + b_3 &\leq 2 \pmod{11} \end{aligned}$$

Etape 3: Génération des solutions

...en utilisant votre heuristique de branchement favoris

Bonus: les échantillons sont nécessairement distinct

EVALUATION

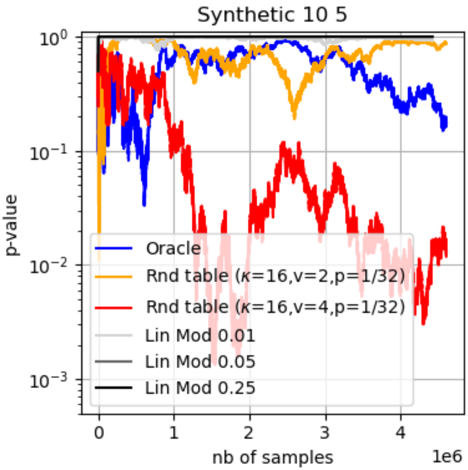
Instances de test:

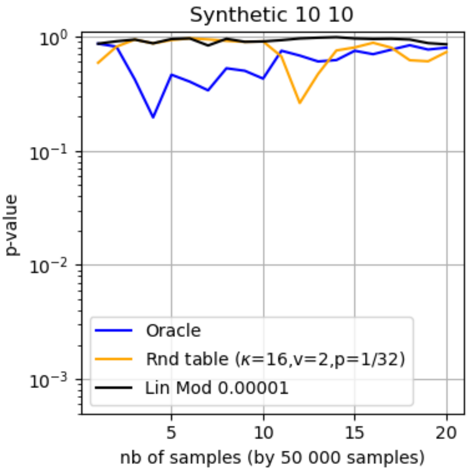
- N-Queens
- Myciel (coloration de graphe)
- Features Models (configuration de logiciel)
- synthetic (#solutions calculable)

Algorithmes comparés:

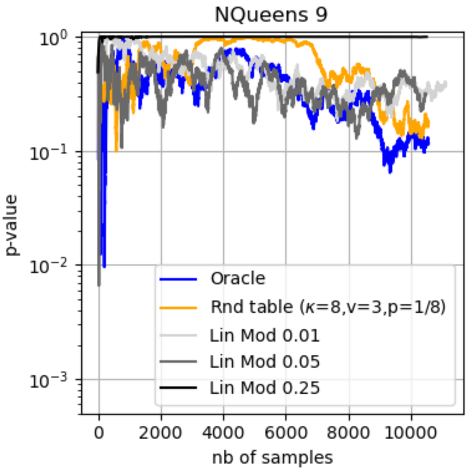
- un oracle
- Random Table (Vavrille, Truchet, Prud'homme, CP 2021)

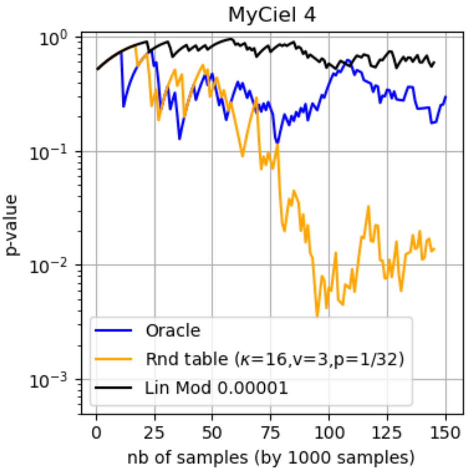
Evolution de l'uniformité par p-valeur (haut = proche de l'uniformité)



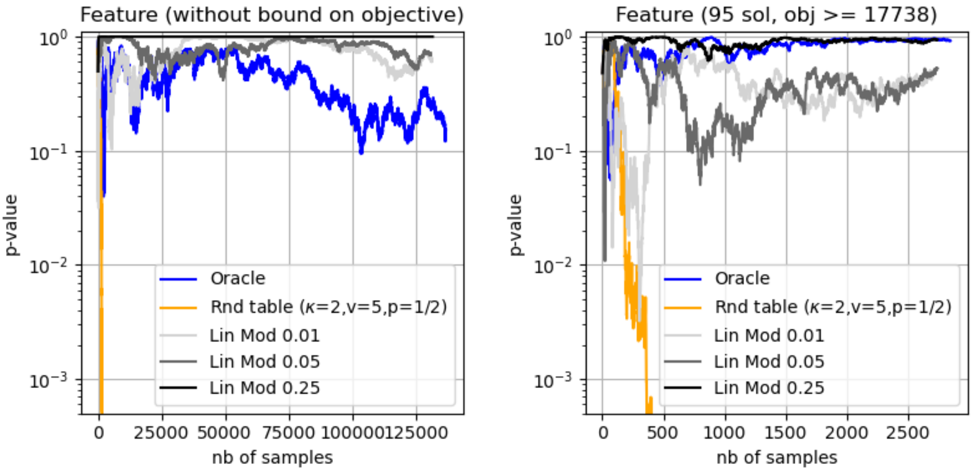


Evolution de l'uniformité par p-valeur (haut = proche de l'uniformité)





Evolution de l'uniformité par p-valeur (haut = proche de l'uniformité)



Ratio de temps entre les autres approches et la nôtre

Echantillonnage de 0.001% des solutions

Instance	#solns	$\frac{\text{\#solns}}{\lvert\text{space}\rvert}$	Runtime ratio			
			Oracle	Random table		
				$v = 2$	$v = 3$	$v = 4$
15-Queens	2.28e+6	5.2e-12	5.4	2e-3	3e-3	1e-2
Myciel 4	1.42e+8	3.0e-07	12.6	0.9	1.3	4.8
Synthetic A	1.53e+5	1.6e-02	105.1	4.7	5.3	5.3
Synthetic B	9.92e+8	1.0e-01	1183.9	2.7	13.3	137.9

CONCLUSION

Notre approche d'échantillonage est:

- simple: ajout de contraintes au modèle CP
- uniforme: répartition uniforme des solutions
- efficace: en comparaison à d'autre methode
- précise: réduction de l'espace

Perspectives :

- experimentation sur des instances de plus en plus grandes
- application aux tests de lociciels
- application au problème d'échange de reins

Merci pour votre attention!

Des questions?