Advanced Constraint Programming : Solveur pour *Social Golfeur Problem*

Coralie Marchau   Maxime Rekar

8 janvier 2025

Introduction

- Modélisation de Social Golfer Problem
- Solveur
- Comparaison
- Améliorations

Social Golfer Problem

- Problème d'optimisation combinatorial
- Problème posé Mai 1998
- Utilisé pour benchmark de cassage de symétries en Programmation par Contraintes

{4,6,9,15}	{3,5,10,16}	{2,8,11,13}	{1,7,12,14}
{4,7,10,13},	{3,8,9,14}	{2,5,12,15}	{1,6,11,16}
{4,8,12,16}	{3,7,11,15}	{2,6,10,14}	{1,5,9,13}
{13,14,15,16}	{9,10,11,12}	{5,6,7,8}	{1,2,3,4}

Variables :

- w *semaines*
- g *groupes par w*
- s *golfers dans g*
- q = g*s *total de joueurs par w*

Challenges :

- Problème hautement symétrique $w! * s! * g! * q!$ solutions symétriques possibles
- Défi de représentation des variables

Social Golfer Problem : Première version "classique"

$$S = \begin{array}{cc} & \begin{array}{ccccc} & & g & & \\ s & s & s & s & q \end{array} \\ w & \begin{array}{|c|c|c|c|}\hline \{\} & \{\} & \{\} & \{\} \\ \hline \{\} & \{\} & \{\} & \{\} \\ \hline \{\} & \{\} & \{\} & \{\} \\ \hline \{\} & \{\} & \{\} & \{\} \\ \hline \end{array} \end{array}$$

Contraintes :

- Chaque semaine, tous les joueurs sont dans un groupe
  $\bigcup_{i=1}^{w} \bigcup_{j=1}^{g} S_{i,j} = n$
- Tous les joueurs se croiseront une fois maximum
  $\bigcap_{i=1}^{s-1} \bigcap_{j=i}^{s} |S| \leq 1$
- Tous les groupes sont remplis
  $|S_{i,j}| = s, \forall i \in w, \forall j \in g$

Social Golfer Problem : Seconde version "Avancé"

S =

g

	s1	s2	s3	s4
w1	{1,2,3,4}	{5,6,7,8}	{9,10,11,12}	{13,14,15,16}
w2	{1,5}	{2}	{3}	{4}
w3	{1,9}	{2}	{3}	{4}
w4	{1,13}	{2}	{3}	{4}

q

Contraintes supplémentaires :

- La première ligne est initialisé $S[1, i] = [i * p + 1 : i * p + 4] \forall i \in [0 : s - 1]$
- Le joueur 1 est affecté au premier groupe de toute semaine $|S_{i,j}| = s, \forall i \in w, \forall j \in g$
- Partant de 2^{eme} semaine, $(i - 1 * p + 1) \in S[i, 1] \forall i \in [2 : w]$
- Partant de 2^{eme} semaine/groupe, $i \in S[i, j], \forall i \in [2 : w], \forall j \in [2 : g]$

- Génération de modèles Minizinc (lu aussi par notre parseur).
- Observations de cas triviaux ou impossibles.
  Réponse ? Étude rapide des cas où le modèle est prouvé impossible.
- Un cas observé : Si $w = g$ et $s \leq w$, alors modèle impossible

{}	{}	{}
{}	{}	{}
{}	{}	{}

Minizinc : Résultats (Gecode 6.3.0)

M{p,g,w,v}	{2,2,2,C}	{2,2,2,A}	{2,3,2,C}	{2,3,2,A}	{2,3,3,C}	{2,3,3,A}
Feasible ?	Y	Y	Y	Y	Y	Y
Time(s)	0.199	0.164	0.164	0.167	0.191	0.181
M{p,g,w,v}	{2,4,2,C}	{2,4,2,A}	{2,4,3,C}	{2,4,3,A}	{2,4,4,C}	{2,4,4,A}
Feasible ?	Y	Y	Y	Y	Y	Y
Time(s)	0.166	0.165	0.179	0.184	0.178	0.183
M{p,g,w,v}	{3,2,2,C}	{3,2,2,A}	{3,3,2,C}	{3,3,2,A}	{3,3,3,C}	{3,3,3,A}
Feasible ?	N	N	Y	Y	Y	Y
Time(s)	0.172	0.126	0.151	0.140	0.245	0.151
M{p,g,w,v}	{3,4,2,C}	{3,4,2,A}	{3,4,3,C}	{3,4,3,A}	{3,4,4,C}	{3,4,4,A}
Feasible ?	Y	Y	Y	Y	Y	Y
Time(s)	0.138	0.165	0.318	0.165	0.198	0.151
M{p,g,w,v}	{4,2,2,C}	{4,2,2,A}	{4,3,2,C}	{4,3,2,A}	{4,3,3,C}	{4,3,3,A}
Feasible ?	N	N	N	N	N	N
Time(s)	0.148	0.123	6.214	0.134	9.623	0.119
M{p,g,w,v}	{4,4,2,C}	{4,4,2,A}	{4,4,3,C}	{4,4,3,A}	{4,4,4,C}	{4,4,4,A}
Feasible ?	Y	Y	Y	Y	Y	Y
Time(s)	0.165	0.180	0.156	0.165	0.188	0.164

Solveur maison

- Développé sous Julia
- Utilisation des domaines finis avec cardinalités
- Domain(lb[liste rangée],up[liste rangée], cardMin, cardMax) = symétrie s ! retiré, et gain lors des opérations de filtrage ou entre domaines.
- Doublons interdits, si un élement est dans lb, il n'est pas dans up, et inversement
- Filtrage fonctionnel pour Union et Intersection

Solveur maison : Faiblesse

Solveur SGP avant Solveur CP

- Parseur uniquement pour les modèles écrits par le writer, soit que SGP
- Filtrage uniquement pour contraintes "union" et "intersection"

Vérification erronée

- Problème liées aux contraintes : la vérification n'est pas correcte, présence de vrai faux et de faux vrai
- Conséquence, la pile est rarement vide.

Solveur maison : Comparaison

M{p,g,w,v}	{2,2,2,C}	{2,2,2,A}	{2,3,2,C}	{2,3,2,A}	{2,4,2,C}	{3,2,2,A}
Time	0.051	0.00025	0.0012	0.002	0.066	0.0005
MZ_Reussi ?	Y	Y	Y	Y	Y	N
MZ_Time	0.199	0.164	0.164	0.167	0.166	0.126
M{p,g,w,v}	{3,2,2,C}	{3,3,2,A}	{3,3,3,C}	{4,2,2,A}	{4,2,2,C}	
Time	0.001	0.033	0.0203	0.008	0.066	
MZ_Reussi ?	N	Y	Y	N	N	
MZ_Time	0.172	0.140	0.245	0.123	0.148	

Solveur maison : Comparaison - Suite

- A prendre avec recul
- Sur les 28 modèles résolus par Minizinc (sur 36 modèles), 11 résolus par le solveur
- Quelques problèmes pas encore corrigés : Certains des modèles résolus par le solveur sont déclarées impossibles par MiniZinc...

Améliorations ?

- Pour le solveur :
  - Corriger la vérification avant de tenter des SGP de plus grande taille
  - Créer une file de domaines modifiés pour cibler plus précisément
  - Créer de nouvelles fonctions de filtrage pour d'autres types de contraintes
- Améliorer le parseur pour pouvoir charger n'importe quelle modèles de CPP
- Étudier plus la nature des problèmes de SGP pour améliorer le writer et limiter les modèles impossibles

Fin

Merci d'avoir écouté
Des questions ?

Introduction

- Social Golfer Problem model
- Solver
- Benchmark with Minizinc
- Ameliorations

Social Golfer Problem

- Combinatorial optimisation problem
- Problem from Mai 1998
- Often used as benchmark of symmetries breakers in Constraint Programmation.

{4,6,9,15}	{3,5,10,16}	{2,8,11,13}	{1,7,12,14}
{4,7,10,13},	{3,8,9,14}	{2,5,12,15}	{1,6,11,16}
{4,8,12,16}	{3,7,11,15}	{2,6,10,14}	{1,5,9,13}
{13,14,15,16}	{9,10,11,12}	{5,6,7,8}	{1,2,3,4}

Variables :

- w *Weeks*
- g *Groups per w*
- s *Golfers in g*
- q = g*s *total amounts of golfers per w*

Challenges :

- Highly symmetrical problem
  $w! * s! * g! * q!$ solutions symétriques possibles
- How to consider variables in CP

Social Golfer Problem : First classical version

S =

g

	s	s	s	s	q
w	{}	{}	{}	{}	
	{}	{}	{}	{}	
	{}	{}	{}	{}	
	{}	{}	{}	{}	

Constraints :

- Each week, all players are in one group
  $$\bigcup_{i=1}^{w} \bigcup_{j=1}^{g} S_{i,j} = n$$
- All players would met exactly once
  $\bigcap_{i=1}^{s-1} \bigcap_{j=i}^{s} |S| \leq 1$
- All groups are full
  $|S_{i,j}| = s, \forall i \in w, \forall j \in g$

Social Golfer Problem : Symmetries breakers

S =

	s1	s2	s3	s4
w1	{1,2,3,4}	{5,6,7,8}	{9,10,11,12}	{13,14,15,16}
w2	{1,5}	{2}	{3}	{4}
w3	{1,9}	{2}	{3}	{4}
w4	{1,13}	{2}	{3}	{4}

g q

Additional constraints :

- First lines is initialized $S[1,i] = [i * p + 1 : i * p + 4] \forall i \in [0 : s - 1]$
- Player 1 is affected in the first group on all week $|S_{i,j}| = s, \forall i \in w, \forall j \in g$
- Starting second week, $(i - 1 * p + 1) \in S[i,1] \forall i \in [2 : w]$
- Starting second week per group, $i \in S[i,j], \forall i \in [2 : w], \forall j \in [2 : g]$

Social Golfer Problem : Writer

- Minizinc models generator (also readed by our solver).
- Trivial or impossibles cases observed.
  Answer ? Quick study to learn when a model is impossible with proof.
- Eg : If $w = g$ and $s \leq w$, then the model is impossible.

{}	{}	{}
{}	{}	{}
{}	{}	{}

Minizinc : Results (Gecode 6.3.0)

M{p,g,w,v}	{2,2,2,C}	{2,2,2,A}	{2,3,2,C}	{2,3,2,A}	{2,3,3,C}	{2,3,3,A}
Feasible ?	Y	Y	Y	Y	Y	Y
Time(s)	0.199	0.164	0.164	0.167	0.191	0.181
M{p,g,w,v}	{2,4,2,C}	{2,4,2,A}	{2,4,3,C}	{2,4,3,A}	{2,4,4,C}	{2,4,4,A}
Feasible ?	Y	Y	Y	Y	Y	Y
Time(s)	0.166	0.165	0.179	0.184	0.178	0.183
M{p,g,w,v}	{3,2,2,C}	{3,2,2,A}	{3,3,2,C}	{3,3,2,A}	{3,3,3,C}	{3,3,3,A}
Feasible ?	N	N	Y	Y	Y	Y
Time(s)	0.172	0.126	0.151	0.140	0.245	0.151
M{p,g,w,v}	{3,4,2,C}	{3,4,2,A}	{3,4,3,C}	{3,4,3,A}	{3,4,4,C}	{3,4,4,A}
Feasible ?	Y	Y	Y	Y	Y	Y
Time(s)	0.138	0.165	0.318	0.165	0.198	0.151
M{p,g,w,v}	{4,2,2,C}	{4,2,2,A}	{4,3,2,C}	{4,3,2,A}	{4,3,3,C}	{4,3,3,A}
Feasible ?	N	N	N	N	N	N
Time(s)	0.148	0.123	6.214	0.134	9.623	0.119
M{p,g,w,v}	{4,4,2,C}	{4,4,2,A}	{4,4,3,C}	{4,4,3,A}	{4,4,4,C}	{4,4,4,A}
Feasible ?	Y	Y	Y	Y	Y	Y
Time(s)	0.165	0.180	0.156	0.165	0.188	0.164

Our solver

- Made in Julia
- Use finites domains with cardinalities
- Domain(lb[sorted list],up[sorted list], cardMin, cardMax) = symmetry s ! solved, gain on filtering operations or between domains.
- Element can only be in one bound. If element in lb, then the element isn't in up, and opposite.
- Focus on filtering in Union and Intersection

Our solver : Weakness

SGP Solver before a CP Solveur

- Parser only work for models written by the writer, so we can only use it for the SGP
- Filtering only for Union and Intersection constraints

Errors in verification

- Constraints linked problems : wrong verification, false positives and true negative in solutions
- Therefore, the pile isn't often empty.

Our solver : Benchmark

M{p,g,w,v}	{2,2,2,C}	{2,2,2,A}	{2,3,2,C}	{2,3,2,A}	{2,4,2,C}	{3,2,2,A}
Time	0.051	0.00025	0.0012	0.002	0.066	0.0005
MZ_Reussi ?	Y	Y	Y	Y	Y	N
MZ_Time	0.199	0.164	0.164	0.167	0.166	0.126
M{p,g,w,v}	{3,2,2,C}	{3,3,2,A}	{3,3,3,C}	{4,2,2,A}	{4,2,2,C}	
Time	0.001	0.033	0.0203	0.008	0.066	
MZ_Reussi ?	N	Y	Y	N	N	
MZ_Time	0.172	0.140	0.245	0.123	0.148	

Our solver : Benchmark

- Take with a pitch of salt
- On 28 solved models by Minizinc (on 36 models), 11 solved by our solver
- Unsolved problems : Some models solved by our solver are declared impossible by MiniZinc...

Fixing it and improving it ?

- For the solver :
  - Fix the verification before trying SGP of bigger size
  - Making a file of modified domains to make the pile more efficient to find filtering operations
  - Make new filtering functions for others types of constraints
- Improve the parser so he could load any models of CPP
- Deepen the study on SGP problem to improve the writer and limits impossibles models

End

Thx for listening
Any questions ?