From puzzles to pandemics: Investigating the need of the Global Cardinality Constraint in LCG

Author: David Thomas Rockenzahn Gallegos (drockenzahngal@student.tudelft.nl)
Supervisor: M.L. Flippo
Responsible professor: Dr. E. Demirović,

TU Delft

Evaluating the usefulness of Global Cardinality constraint propagators in Lazy Clause Generation

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Figure ?: A Sudoku Puzzle, which can be solved using only GCCs
Tim Stellmach, CC0, via Wikimedia Commons

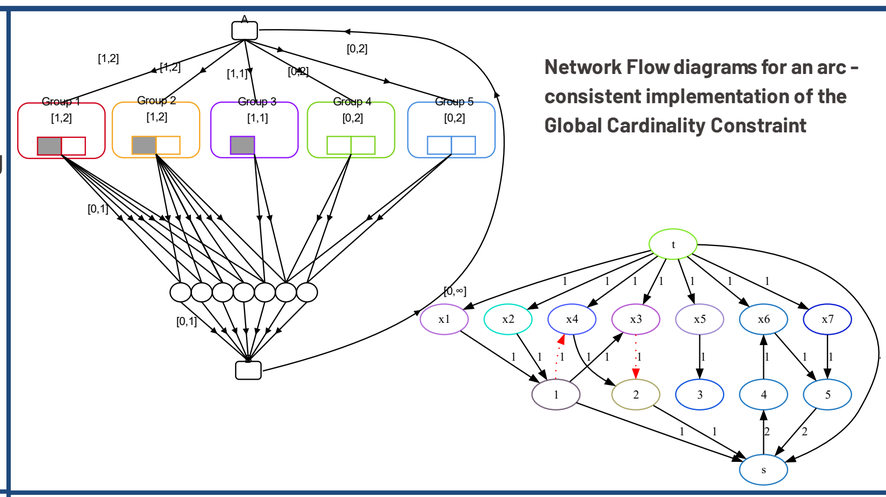
Sudoku can be solved using only the GCC

1 Introduction FDP Vs LCG

Finite Domain Propagation (**FDP**): Solve problems by **modeling restrictions** as **constraints.** Alternate between **Search** and **Propagation.**

Trade-off: runtime vs **pruning strength** [1]

Lazy Clause Generation (**LCG**)**:** Newer, state-of-the-art. Combines **Propagation** + **SAT Solver**



Network Flow diagrams for an arc - consistent implementation of the Global Cardinality Constraint

GCC Vs Decomposition

2 Research question

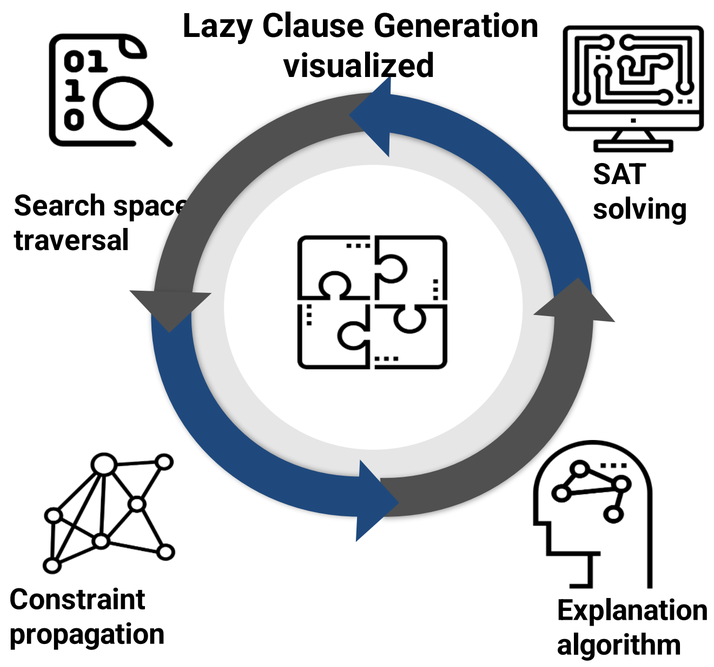
Global Cardinality Constraint (**GCC**): how many times values can be assigned to variables.

- Arises in real-world **scheduling problems** [2][3]
- GCC can be **decomposed** into smaller, simpler constraints.

LCG questions need for constraints like the **GCC**

- Decompositions can yield state-of-the-art solutions for some constraints. [1][4]
- **But**: Little research about implementing **GCC** propagators for **LCG**

Question: Can a GCC propagator in LCG compete against decomposition?



$$E_{\text{naive}}(X) = \bigwedge_{\forall x \in X} \left([x \geq \text{lb}(x)] \wedge [x \leq \text{ub}(x)] \bigwedge_{h \in H(x)} [x \neq h] \right)$$

$$E_{\text{filt1}}(X, v) = \left(\bigwedge_{x \in X} [x = v] \quad \textbf{if} \quad x = v \right)$$

$$E_{\text{filt2}}(X, v) = \left(\bigwedge_{x \in X} [x \neq v] \quad \textbf{if} \quad v \notin D(x) \right)$$

$$E_{\text{Regin}}(x, v, X, V, G_0, G) = \bigwedge_{(v', x') \in G_0} [x' \neq v'] \textbf{ if } (v', x') \notin G \wedge \text{J}(x', v', R(G), x, v)$$

The explanation algorithms: General explanation, Basic Filter explanation and Regin explanation

3 Methodology

Two propagators and**, two explanations** are **benchmarked** using the Minizinc Challenge [5] and Sudoku against decomposition.

Propagator algorithms:

- Regin GAC [6]
- Basic Filter

Explanation algorithms:

- Naive
- Purpose-built

Runtime Vs Pruning strength

4 Results

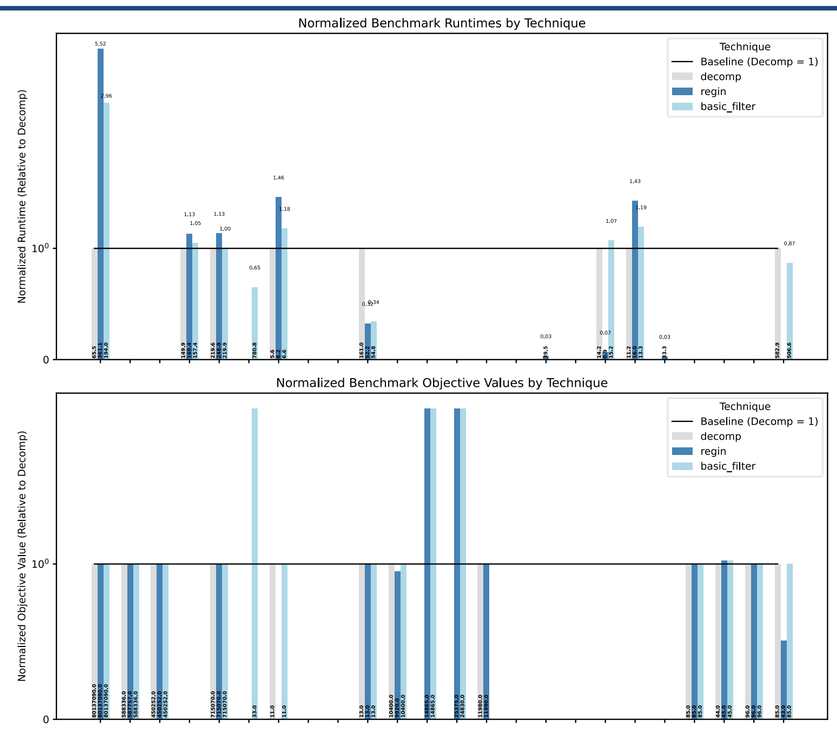
- Geometric mean increase ratio (compared to decomposition) of Regin GAC and Basic filter

Sudoku:

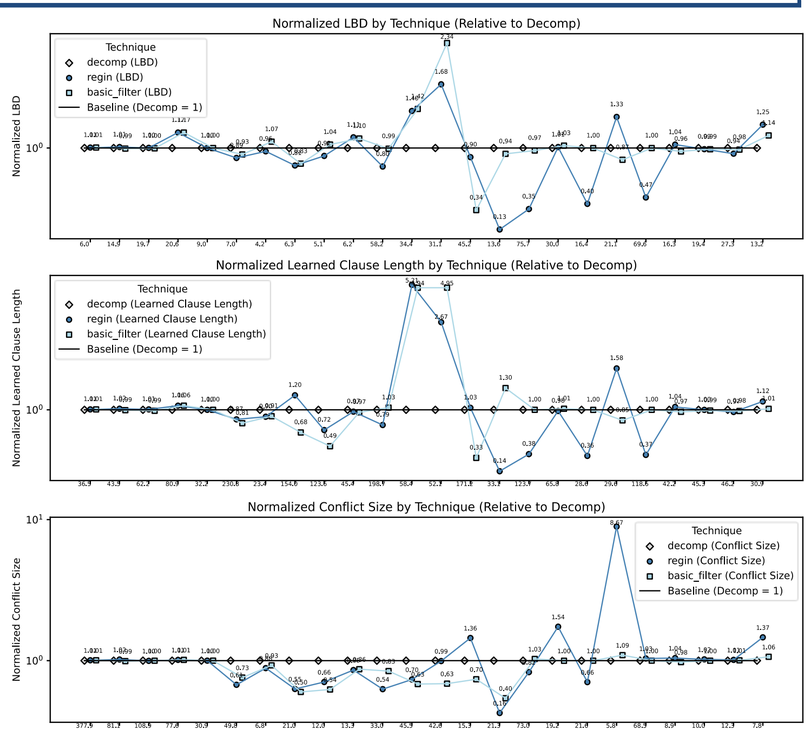
(Regin, Basic Filter)	Speedup	LBD	LCL	ACS
General expl	(3.42, 1.52)	(0.90, 1.05)	(0.91, 0.95)	(0.92, 0.95)
Purpose built expl	(3.19, 2.28)	(0.85, 1.00)	(0.91, 1.03)	(0.93, 0.84)

Minizinc challenge:

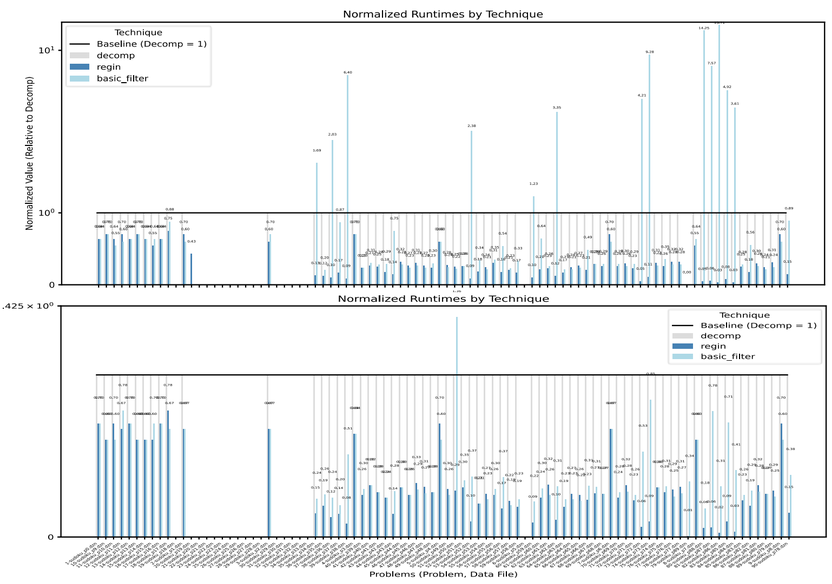
(Regin, Basic Filter)	Speedup	LBD	LCL	ACS
General expl	(1.34, 0.91)	(0.90, 1.05)	(0.91, 0.95)	(0.92, 0.95)
Purpose built expl	(1.36, 1.00)	(0.85, 1.0)	(0.91, 1.03)	(0.93, 0.84)



Minizinc benchmark, relative runtime (decomp = 1) of Regin GAC and Basic Filter using new explanation algorithms



Minizinc challenge relative (decomp = 1) LBD, LCL and ACS of Regin GAC and Basic filter, using new explanation algorithms



Sudoku benchmark relative runtime (decomposition = 1). Top: Naïve explanation. Bottom: Specialized explanation

5 Conclusion

- Regin GAC and Basic Filter superior against decomposition on Sudoku
- Regin GAC wins in runtime against decomposition on Minizinc challenge. Basic Filter roughly equal.
- New explanation greatly helps Basic Filter. Does not help Regin GAC as much because of high runtime cost and the clause length being larger than it should be
- New explanations have a 5% LBD reduction compared to the naïve one in both Sudoku and Minizinc challenge. How much does GCC contribute to LBD?
- Regin GAC achieved a lower (better) LBD compared to decomposition (85%) using new explanation.

References

- [1] Stuckey, P. J. (2010). Lazy clause generation: Combining the power of SAT and CP (and MIP?) solving. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6140 LNCS*. https://doi.org/10.1007/978-3-642-13520-0_3
- [2] Geibinger, T., Kletzander, L., Krainz, M., Mischek, F., Musliu, N., & Winter, F. (2021). Physician Scheduling During a Pandemic. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12735 LNCS*. https://doi.org/10.1007/978-3-030-78230-6_29
- [3] Musliu, N., Schutt, A., & Stuckey, P. J. (2018). Solver independent rotating workforce scheduling. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10848 LNCS*. https://doi.org/10.1007/978-3-319-93031-2_31
- [4] Schutt, A., Feydy, T., Stuckey, P. J., & Wallace, M. G. (2009). Why cumulative decomposition is not as bad as it sounds. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5732 LNCS*. https://doi.org/10.1007/978-3-642-04244-7_58
- [5] Stuckey, P. J., Feydy, T., Schutt, A., Tack, G., & Fischer, J. (2014). The MiniZinc challenge 2008-2013. In *AI Magazine* (Vol. 35, Issue 2). https://doi.org/10.1609/aimag.v35i2.2539
- [6] Regin, J. C. (1996). Generalized arc consistency for global cardinality constraint. *Proceedings of the National Conference on Artificial Intelligence*, *1*.

Image References:

- Algorithm by Andi wiyanto from https://thenounproject.com/browse/icons/term/algorithm/ (CC BY 3.0)
- Logic by Becris from https://thenounproject.com/browse/icons/term/logic/ (CC BY 3.0)
- Binary code by Kokota from https://thenounproject.com/browse/icons/term/binary--code/ (CC BY 3.0)
- Puzzle by Smalllike from https://thenounproject.com/browse/icons/term/puzzle/ (CC BY 3.0)
- Solution by Esha chaudary from https://thenounproject.com/browse/icons/term/solution/ (CC BY 3.0)
- Versus by Amethyst Studio from https://thenounproject.com/browse/icons/term/versus/ (CC BY 3.0)