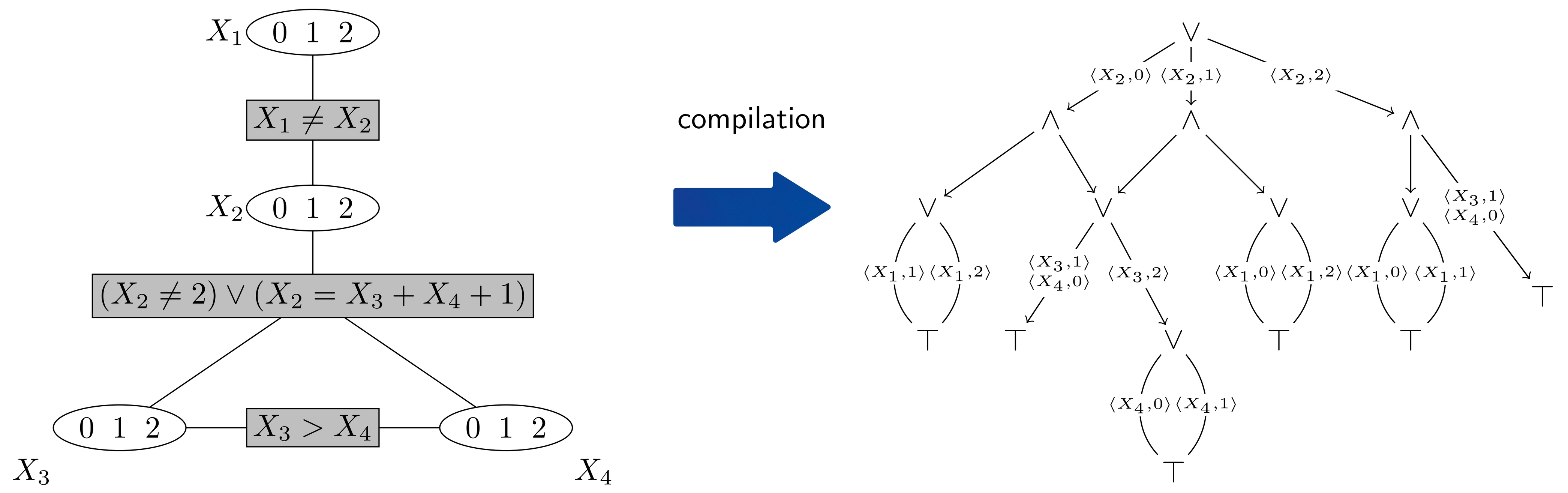
# Compiling Constraint Networks into Multivalued Decomposable Decision Graphs

*Frédéric Koriche* *Jean-Marie Lagniez* *Pierre Marquis* *Samuel Thomas*
CRIL, Université d'Artois & CNRS, Lens France
{koriche,lagniez,marquis,thomas}@cril.fr

## Motivations & Contributions

- Constraint programming is a *useful approach* for representing and solving combinatorial problems
- However *no performance guarantees* are offered for many tasks (consistency, solution counting, solution enumeration, optimization, etc.) in on-line applications
- Knowledge compilation provides guarantees for such tasks, by encoding the constraint network into an appropriate representation
- We defined *a language* MDDG *for compiling constraint networks* ($\mathcal{N}$), such that all aforementioned tasks can be achieved in polynomial time from MDDG representations
- We designed and evaluated a *compiler* (cn2mddg) targeting this language

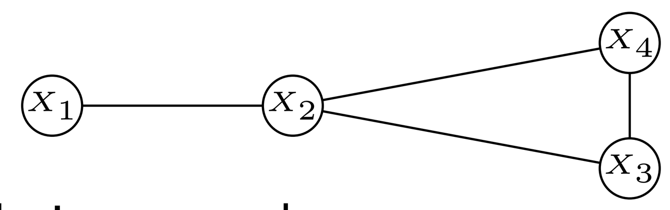
## Multivalued Decomposable Decision Graphs (MDDG)




- Decision-DNNF corresponds to the proper subset of MDDG where each variable is Boolean
- The key tractable queries and transformations offered by Decision-DNNFare also offered by MDDG

## A Top-Down Compiler

- Following the trace of a solver
- Taking into account the structure of the problem by considering its primal graph



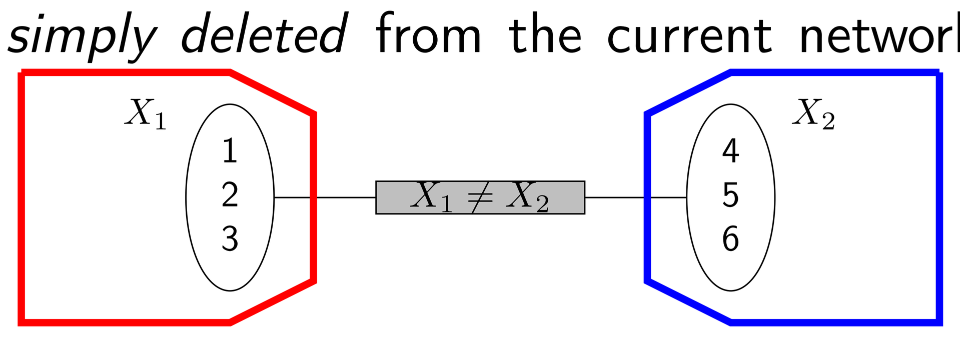
- Techniques used
  - Component analysis
  - Specific caching technique
  - Universal constraints
  - Specific variable selection heuristic

## Caching Technique

- Caching is a key technique of any compiler computing DAG-based representations
- Two networks are detected as "equivalent" when they are identical
- For an efficient caching, the size of the entries must be small
  - one stores the current domains of the current unassigned variables
  - $\forall C_i \in \mathcal{N}$, if $C_i$ is AC, $|C_i| > 2$ and $\exists X_j \in C_i$ s.t. $X_i$ has been reduced, then a projection of $C_i$ is saved

## Universal Constraints

- Universal constraints are constraints that are *necessarily satisfied* whatever the values given to the variables in their scopes
- Once detected, a universal constraint is *simply deleted* from the current network



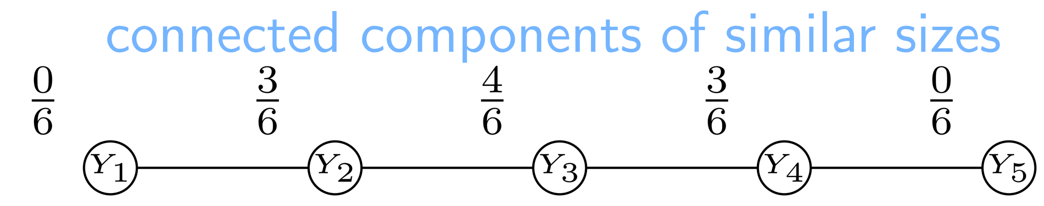
- The objective is to simplify the forthcoming treatments and *to promote decomposition*

## Variable Selection Heuristic

- The heuristics used for solution finding are not dedicated to knowledge compilation
- We considered a heuristic $bc$ based on the concept of betweenness centrality
- $bc$ relies on the network structure

$$bc(X_i) = \Sigma_{X_j \neq X_i \neq X_k} \frac{\sigma_{X_i}(X_j, X_k)}{\sigma(X_j, X_k)}$$

- Assigning the most central variables is a way to promote the generation of disjoint connected components of similar sizes



## Experimental Results

- Benchmarks: 173 CNs from 15 data sets (configuration, scheduling, frequency allocation, ...)
- Each input CN has been compiled into
  - a MDDG representation using our compiler cn2mddg
  - and
    - a CNF using the sparse encoding with a mixed clause encoding of the constraints
    - a Decision-DNNF using the compiler Dsharp
- A time limit of 3600s and a total amount of 8GiB of memory have been considered for each instance

| | CN | | | | | | | CNF - sparse mixed encoding | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | $\#\mathcal{X}$ | $\#C$ | maxA | maxD | tw | time | size | #pv | #pcl | time | size |
| rect-packingrpp09 | 2196 | 2353 | 10 | 36 | 19 | 1673.33 | 514754 | 37044 | 593518 | 375.66 | 16118647 |
| ghoulomb3-4-5 | 2033 | 2051 | 11 | 26 | 31 | 15.17 | 5162 | MO | MO | MO | MO |
| talent-concert | 325 | 352 | 46 | 316 | 52 | 1277.21 | 404437 | MO | MO | MO | MO |
| CostasArray10 | 110 | 338 | 4 | 19 | 23 | 10.39 | 13440 | 149564 | 841930 | TO | - |
| photophoto2 | 89 | 133 | 21 | 11 | 21 | 499.93 | 9564220 | 685555 | 14326576 | TO | - |
| rlfap-scen4 | 680 | 3967 | 2 | 44 | 30 | 3.47 | 52226 | 915553 | 4875002 | - | MO |
| renault-mod-32 | 111 | 154 | 10 | 42 | 11 | 20.39 | 160238 | 222582 | 1755876 | TO | - |
| renault-mod-11 | 111 | 149 | 10 | 42 | 10 | 16.22 | 41919 | 223718 | 1762294 | 3538.01 | 2399273 |
| driverlogw-08c | 408 | 9321 | 2 | 11 | 92 | 15.63 | 2931 | 9528 | 62825 | 6.42 | 139306 |

➟ cn2mddg succeeded in compiling 131 instances over 173 (32 TO and 10 MO)
➟ Dsharp succeeded in compiling 83 instances over 173 (24 TO and 39 MO)

## Conclusion and Perspectives

- Contribution: A top-down algorithm cn2mddg for compiling finite-domain CNs into MDDGs has been designed and evaluated
- Take-home message:
  - While a translation to CNF enabling to take advantage of an upstream SAT solver can be a competitive approach for the CSP problem, it turns out to be a bad idea when the objective is to *compile* a constraint network
  - Variable selection heuristics for CSP vs. variable selection heuristics for compilation
- Future work: Implementing additional queries and considering new heuristics for promoting decomposition