Graphical Models for Hybrid AI

Design with Intuition[1] and Logic[2]

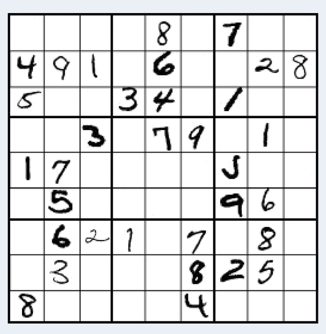
Graphical models (GMs), Guaranteed Optimization and Hybrid AI

T. Schiex, S. Barbe, S. de Givry, G. Katsirelos, D. Simoncini

The "Design with intuition and logic" chair is organized around the use of guaranteed optimization over graphical models as a way to connect ML/DL (using stochastic GMs such as Markov Random Fields or Bayesian Nets) with Automated reasoning (using Boolean GMs, as in Constraint Programming). This is achieved using Toulbar2, an exact award-winning "Cost Function Network" prover that deals with Boolean and numerical tensors. Because exact optimization of discrete GMs is decision NP-complete, improving algorithmic efficiency is always crucial (block 3, 5 and 6). Applications (CPD) also motivate the resolution of new queries (block 4). With the growth of ML/DL we also interact with ML, trying to inject data in the graphical model's factors (criteria or constraints - block 1) or using CP to exactly optimize GM learning (eg. Bayes Nets, block 2).

1 - Learning how to play Sudoku[6]

For learning "how to reason", learning how to play Sudoku from solved grids has become a standard benchmark. Recurrent Relational Net (RRN, NeurIPS18), a Graph neural network, solves 96.6% of hard minimal grids using a 180,000 grids training set, SATNet (ICML'19), with a differentiable convex optimization layer, solves 98.3% of *easy* grids using 10,000 samples.



We propose a non-end-to-end-differentiable architecture combining an ADMM-based approximate L1-regularized loglikelihood learning algorithm (AISTATS'17) with TOULBAR2. We reach 100% accuracy on hard problems from 10,024 samples. While non-differentiable, our architecture is able to learn how to play the Visual Sudoku, outperforming SATNet both in accuracy and sampling efficiency.

As a preamble to learning energy functions for protein design, we designed a neural network taking Sudoku grid coordinates as inputs and producing a full pairwise GM as output. The loss function combines pseudo-loglikelihood, Hinge loss (computed from TOULBAR2's solution) and L1 regularization of the output. This fully differentiable architecture reaches 100% accuracy on hard grids, using 1,000 training samples.

2 - Learning Optimal Bayesian Network Structures[5]

Bayesian Network Structure Learning (BNSL) from discrete observations corresponds to finding a compact Bayes Net model which best explains the data. BNSL is NP-hard.

We developed ELSA (Exact Learning of Bayesian Network Structure Using Acyclicity Reasoning). ELSA extends CPBayes by using a polynomial-time algorithm that discovers a class of cluster cuts that provably improve the linear relaxation, producing strong bounds. It also uses a more compact representation of candidate parent sets.

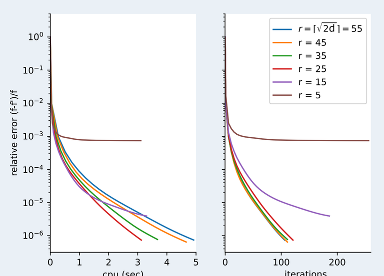
We compared ELSA to state-of-the-art BNSL constraint programming (CPBayes) and integer programming (GOBNILP) solvers on several datasets from UCI ML Repository with a maximum CPU time of 10 hours. The † symbol indicates time-outs.

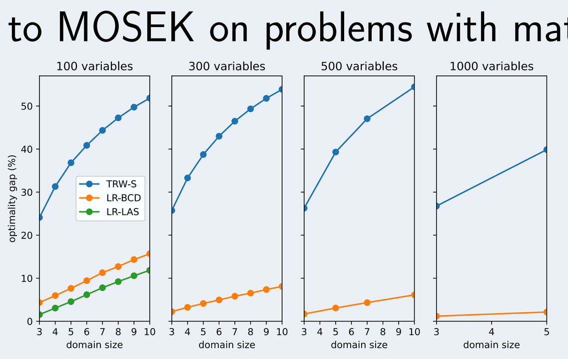
Data Set	n	Σd	CPBayes Time	GOBNILP Time	ELSA Time
bnetflix.ts	100	446406	**629.0**	†	1 065.1
plants.test	69	520148	†	†	**18 981.9**
jester.ts	100	531961	†	†	**10 166.0**
accidents.ts	111	568160	†	**1 274.0**	2 238.7
plants.valid	69	684141	†	†	**12 347.6**
jester.test	100	770950	†	†	**17 637.8**
bnetflix.test	100	1103968	**3 525.2**	†	8 197.7
bnetflix.valid	100	1325818	**1 456.6**	†	9 282.0
accidents.test	111	1425966	†	4 975.6	**3 661.7**

3 - Efficient SDP Bounds for Discrete Graphical Models[3]

Computing the extremum (Maximum A Posteriori/MAP) assignment of a Graphical Model is NP-Hard. Different methods derived from linear programming (LP) and semidefinite programming (SDP) have been defined to compute bounds. We developed a dedicated block coordinate descent (BCD) algorithm which can be applied to a multi-label GM-dedicated SDP relaxation with diagonal and linear constraints.

For scalability, our algorithm follows a Burer-Monteiro scheme with a low rank factorization. On GMs optimization, it is far more efficient than open source solvers such as CVXPY but also dedicated industrial solvers such as MOSEK, both in terms of memory and cpu-time. We reached a speed-up of 20 compared to MOSEK on problems with matrices of size 1500 x 1500.

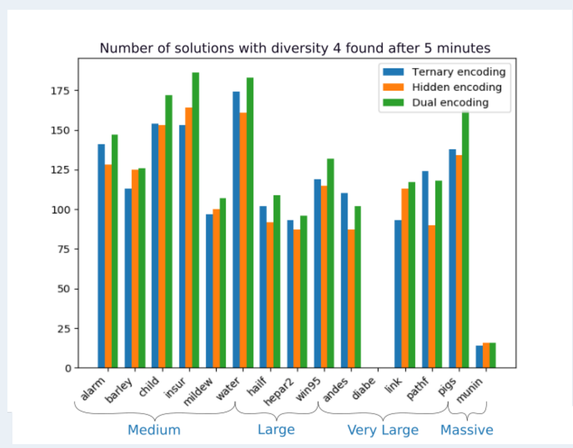


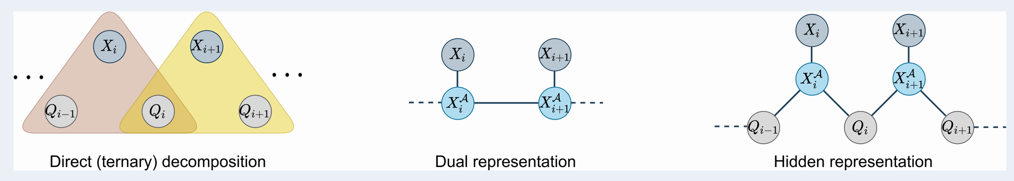


On the hardest dense instances our approach outperforms the (LP-related) message passing algorithm TRW-S providing tighter bounds on the optimal value. Despite being Burer-Monteiro based, our method applies directly to non-binary variables. The figure shows optimality gaps for domain sizes varying from 3 to 10.

4 - Guaranteed Diverse Solutions in Graphical Models[4,7]

To produce a library of diverse solutions of minimum cost, for GMs representing approximate or uncertain knowledge, we introduce an incremental solver and show that it can enumerate local minima. Among various existing approaches, our approach is the only one that guarantees diversity and minimum cost (given previously generated solutions) with no compromise or approximation. We encode the (weighted) Hamming distance constraint using optimized ternary and pairwise decompositions of the CP REGULAR global constraint. We show that the dual encoding gives the best CPU performances.





A comparison of our encodings on various Maximum Probability Explanation problems on Bayes Nets of various sizes from `http://www.bnlearn.com/bnrepository`.

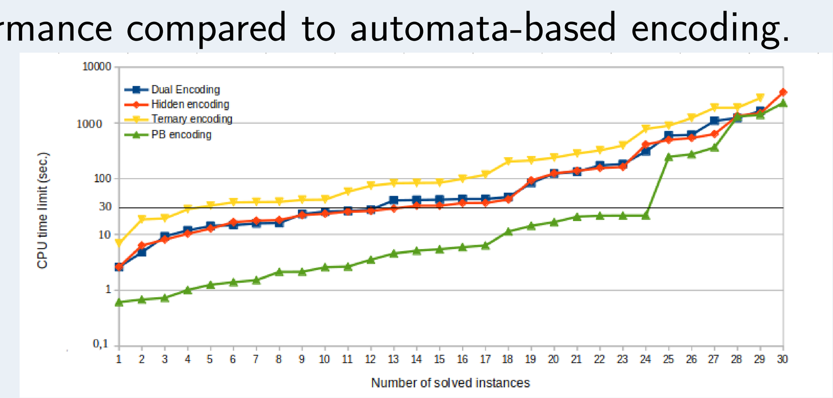
5 - Multiple-Choice Knapsack Constraint in Graphical Models[2]

There is no native way to express linear constraints in a graphical model. This is in large part due to the algorithms used to compute lower bounds, which require all constraints to be expressed in extension. We defined a specialized structure that encodes Multiple-Choice Knapsack Constraint (MCKC) without inducing an exponential representation. Using that structure it is possible derive a local lower bound for each MCKC and associated unary costs (costs associated to the values). We implemented this method in TOULBAR2, where we compute a lower bound at each node of a branch and bound algorithm. This new possibility of embedding linear constraints in graphical models provides greater modeling flexibility and allows the GM solver TOULBAR2 to solve problems that were previously out of its reach, like capacitated warehouse location problem or Knapsack Problem with Conflict Graph, where it is competitive with CPLEX. MCKC can also encode the Hamming distance constraint used to produce diverse solutions. On Computational Protein Design problem (CPD), MCKC shown competitive performance compared to automata-based encoding.

	TB2	CPLEX tuple	CPLEX direct
C1	708	694	**720**
C3	571	477	**609**
C10	**485**	316	459
R1	**720**	703	**720**
R3	**709**	566	676
R10	524	368	**528**

	TB2	CPLEX tuple	CPLEX direct
C1	**720**	701	**720**
C3	**712**	507	639
C10	**609**	337	559
R1	**720**	704	**720**
R3	**717**	578	682
R10	**634**	385	598

Number of solved instances (left) and number of times a solver found the best solution within the time limit (right) for six different classes of KPCG.



Cactus plot of CPU solving time (log scale) to compute 10 diverse solutions on CPD problems, using different encodings of Hamming distance constraint.

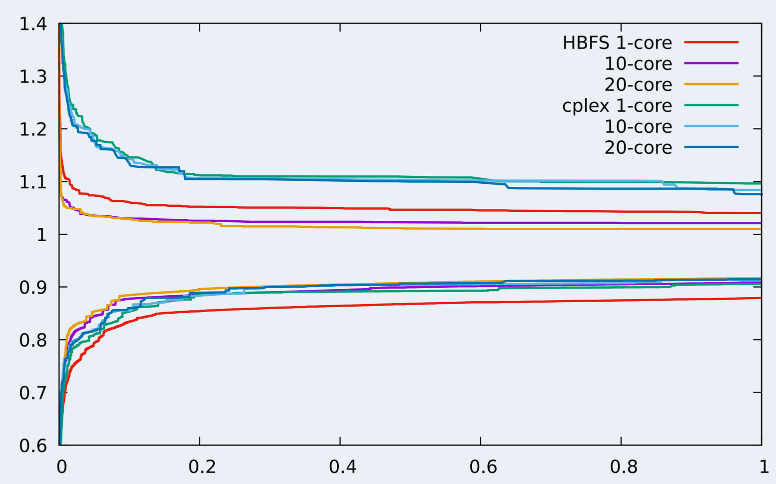
6 - Parallel Hybrid Best First Search[1]

We implement a parallel Branch-and-Bound method in TOULBAR2. Our Hybrid Best-First Search (HBFS) method is based on a Master-Worker protocol using MPI. It combines depth-first search, for the workers, and best-first search, for the master. We observed important speed-ups compared to the sequential version in some difficult benchmarks. Parallel HBFS compares also favorably with parallel CPLEX in terms of normalized lower and upper bounds on a larger set of instances.

instance	n	d	HBFS # of cores 10	20	180	1800
maxclique/brock200_1	200	2	8.3	18.3	**154.5**	70.3
maxclique/p_hat700-1	700	2	3.4	7.2	**20.2**	13.1
maxclique/sanr200_0.9	200	2	10.7	16.8	147.0	**689.2**
cpd/1BRS	38	178	5.1	10.2	13.5	**33.5**
cpd/1CDL	38	170	3.9	6.1	20.1	**24.7**
cpd/1GVP	52	170	6.4	14.3	**45.47**	32.0

Speed-ups: ratio of solving times between parallel HBFS (using n cores) and sequential HBFS obtained on a server (10&20 cores) and the CALMIP cluster (from 180 to 1,800 cores) on difficult DIMACS Max-Clique and Computational Protein Design instances.

Normalized lower and upper bounds as time passes on 134 instances of CPD, Max-Clique, Linkage, and Warehouse benchmarks. A unit of '1' on the X-axis corresponds to $3,600$ seconds x-axis and to the optimum or best known cost on the Y-axis.



References

1. Beldjilali, A. *et al. Parallel Hybrid Best-First Search for Cost Function Networks.* Submitted to the 28th International Conference on Principles and Practice of Constraint Programming, CP. 2022.
2. Montalbano, P. *et al. Multiple-choice knapsack constraint in graphical models.* in *the 19th International Conference on the Integration of Constraint Programming, AI and OR - CPAIOR* (2022).
3. Durante, V. *et al.* Efficient semidefinite bounds for multi-label discrete graphical models. *arXiv preprint arXiv:2111.12491, under submission.* (2021).
4. Ruffini, M. *et al.* Guaranteed diversity and optimality in cost function network based computational protein design methods. *Algorithms* **14,** 168 (2021).
5. Trösser, F. *et al. Improved Acyclicity Reasoning for Bayesian Network Structure Learning with Constraint Programming.* in *the 30th International Joint Conference on Artificial Intelligence, IJCAI* (2021), 4250–4257.
6. Brouard, C. *et al. Pushing Data into CP Models Using Graphical Model Learning and Solving.* in *the 26th International Conference on Principles and Practice of Constraint Programming, CP* (2020), 811–827.
7. Ruffini, M. *et al. Guaranteed Diversity & Quality for the Weighted CSP.* in *31st IEEE International Conference on Tools with Artificial Intelligence, ICTAI* (IEEE, 2019), 18–25.