Probabilistic Reasoning Evaluation

Adnan Darwiche, Rina Dechter
Arthur Choi, Vibhav Gogate and Lars Otten

Scope

- Probability of evidence, PE
- Node marginals, MAR
- Most probable explanation, MPE
- Exact and approximate solvers
- Bayesian and Markov networks
- More than a thousand benchmarks
- 26 solvers from 7 groups

Evaluation Environment

- Cluster lent to us by Prof. Eleazar Eskin (UCLA)
- 5 Linux machines (CentOS 4.5)
  - Intel Xeon, Dual Quad-Core, 2.33GHz, 8GB RAM
  - 2 solvers per machine (1 solver per CPU)
    - 10 solvers running concurrently
  - Each solver limited to:
    - 20 minutes CPU time, 3GB RAM

Agenda

- Benchmark description
- Solver description
- Evaluations:
  - MPE
  - Exact PE/MAR
  - Approximate PE/MAR
- Concluding remarks and discussion

Benchmark Description

Benchmarks

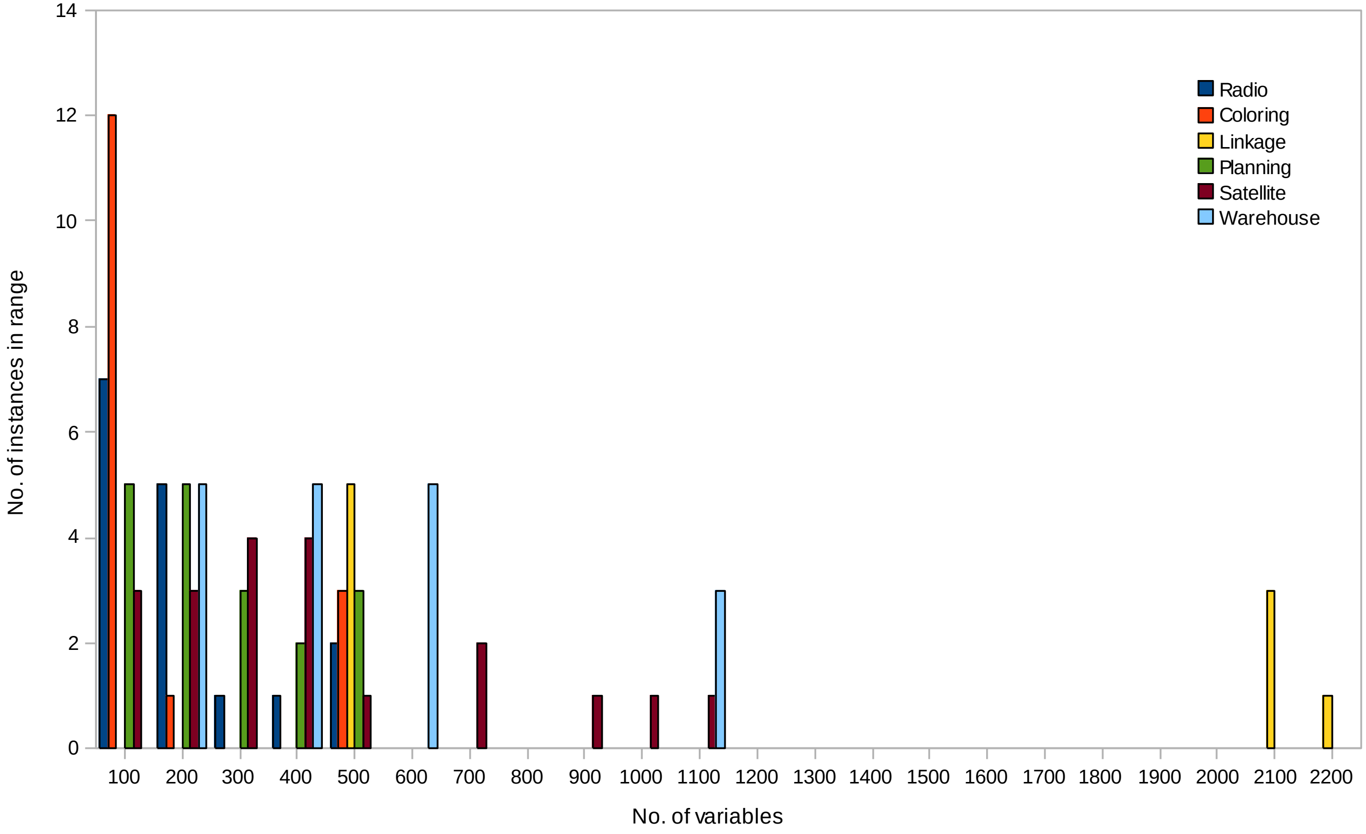
- Linkage 1 / WCSP
  - Submitter: Thomas Schiex (INRA France)
  - Domain: Linkage and converted weighted CSPs
  - Type: Linkage is Bayes, others Markov, all for MPE
    - 9 Linkage: max. domain size 45
    - 16 Radio Freq.: max. domain size 44
    - 16 Coloring: max. domain size 5
    - 18 Planning: max. domain size 27
    - 20 Satellite: max. domain size 4
    - 18 Warehouse: max. domain size 200
  - Treewidth: many have width ~30-60

Benchmarks

- Linkage 1 / WCSP
  - Submitter: Thomas Schiex (INRA France)
  - weighted CSP networks also submitted as benchmarks for 3rd Max-CSP competition
    - http://cpai.ucc.ie/
  - encourage cross-field comparisons?

Benchmarks

- Linkage 1 / WCSP (contd.)



Benchmarks

- bn2o
  - Submitter: Jirka Vomlel and Petr Savicky (Academy of Sciences of the Czech Republic)
  - Domain: two-layer noisy-or Bayesian networks
  - Type: Bayes for MAR/PRE
    - 18 instances
    - All variables binary
    - 45, 50, or 55 variables
  - Treewidth: ~24-27
    - some exact PE solvers run out of memory (given 3GB)

Benchmarks

- Diagnose
  - Submitter: John M. Agosta (Intel Corp.)
  - Domain: diagnostic Bayesian networks, hand-built
  - Type: Bayes for MAR
  - Randomly generated evidence
    - 2 Instances, each with 50 different sets of randomly generated evidence (leaf nodes only)
    - 203 and 359 variables, respectively
    - Max. domain size 7 and 6, respectively

Benchmarks

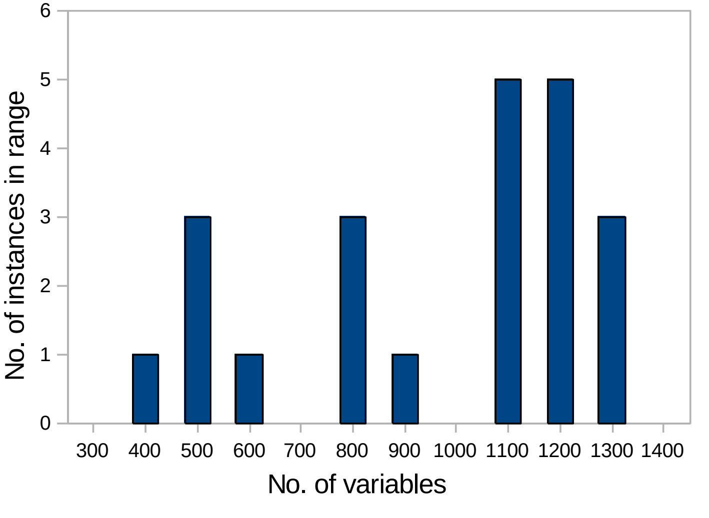
- Diagnose
  - Submitter: John M. Agosta (Intel Corp.)
  - Domain: diagnostic Bayesian networks, hand-built
    - nodes assume causal independence (e.g., noisy-max)
    - relatively large for networks constructed by hand
      - ~200-300 nodes, ~300-600 edges
    - treewidth ~11-18: still easy for exact solvers

Benchmarks

- Grids
  - Submitter: Tian Sang (University of Washington)
  - Domain: Grid networks, from 12x12 to 50x50 with varying level of determinism
    - roughly, 50%, 75%, or 90% of the parameters are 0/1
    - treewidth: ~12-50
  - Type: Bayes for PRE
    - 320 Instances
    - Between 144 and 2,500 binary variables
    - Evidence by assigning value 1 to leaf node

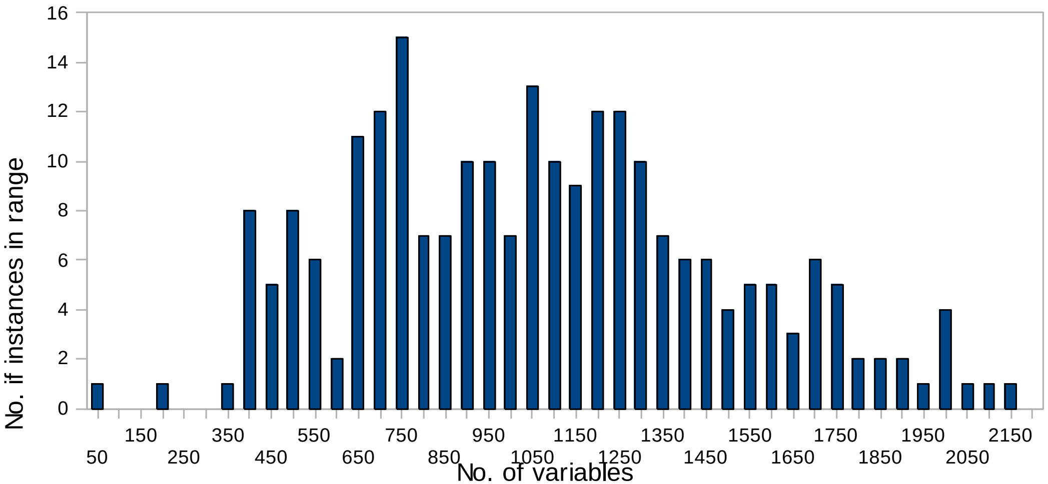
Benchmarks

- Linkage 2
  - Submitter: Dechter group (UC Irvine)
  - Domain: Genetic linkage
  - Type: Markov for MPE
    - 22 instances
    - Max. domain size between 3 and 7
    - Treewidth: ~20-35



Benchmarks

- Promedas
  - Submitter: Vicenc Gomez (University Nijmegen)
  - Domain: Medical diagnosis, real-world cases, converted from noisy-or
  - Type: Markov for MAR/PRE
    - 238 Instances
    - Binary variables



Benchmarks

- Promedas
  - Submitter: Vicenc Gomez (University Nijmegen)
  - Domain: Medical diagnosis, real-world cases
  - QMR-DT like networks; layered noisy-or model
    - Bayesian network model converted to Markov network after performing simplifications (pruning unobserved nodes, negative findings, compact representation of noisy-or, etc)
  - Treewidths range from 1 (tree) to ~60
    - most are too difficult for exact algorithms

Benchmarks

- UAI-06 MPE and PRE
  - Submitter: Used in UAI'06 evaluation
  - Domain: Various
  - Type: Bayes for MPE and PRE, respectively
    - 57 MPE instances
    - 78 PRE instances
    - For details, see last UAI evaluation

Benchmarks

- Relational
  - Submitter: UCLA
  - Domain: Relational Bayesian networks constructed from the Primula tool
  - Type: Bayes for MAR/PRE
    - 251 networks, with binary variables
      - 150 Blockmap: 700 to 59,404 variables
      - 80 Mastermind: 1,220 to 3,692 variables
      - 11 Friends & Smoker: 10 to 76,212 variables
      - 10 Students: 376 variables
    - Large networks with large treewidths, but with high levels of determinism

Benchmark Summary (exact)

- 9 sets:

		Bys	Mkv	bin
– weighted-CSP	(97)		•	
– bn2o (diagnosis)	(18)	•		•
– hand-built	(100)	•		
– grids	(320)	•		•
– linkage	(22)		•	
– Promedas	(238)		•	•
– UAI-06 (MPE)	(57)	•		
– UAI-06 (PE)	(78)	•		
– relational	(251)	•		•
– **TOTAL**	**(1181)**	**(824)**	**(357)**	**(507)**

Benchmark Summary (appr/mpe)

- 9 sets:

		Bys	Mkv	bin
– weighted-CSP	(97)		•	
– bn2o (diagnosis)	(18)	•		•
– hand-built	(0/100)	•		
– grids	(32/320)	•		•
– linkage	(22)		•	
– Promedas	(238)		•	•
– UAI-06 (MPE)	(57)	•		
– UAI-06 (PE)	(78)	•		
– relational	(35/251)	•		•
– **TOTAL**	**(577)**	**(220)**	**(357)**	**(323)**

Solver Description

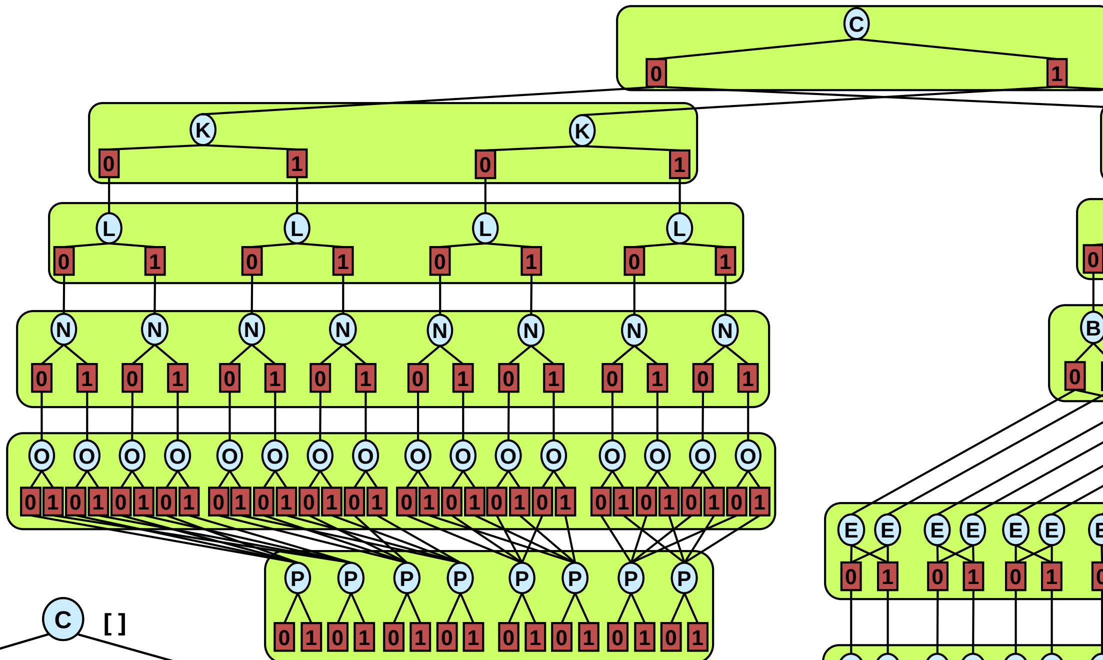
Solvers

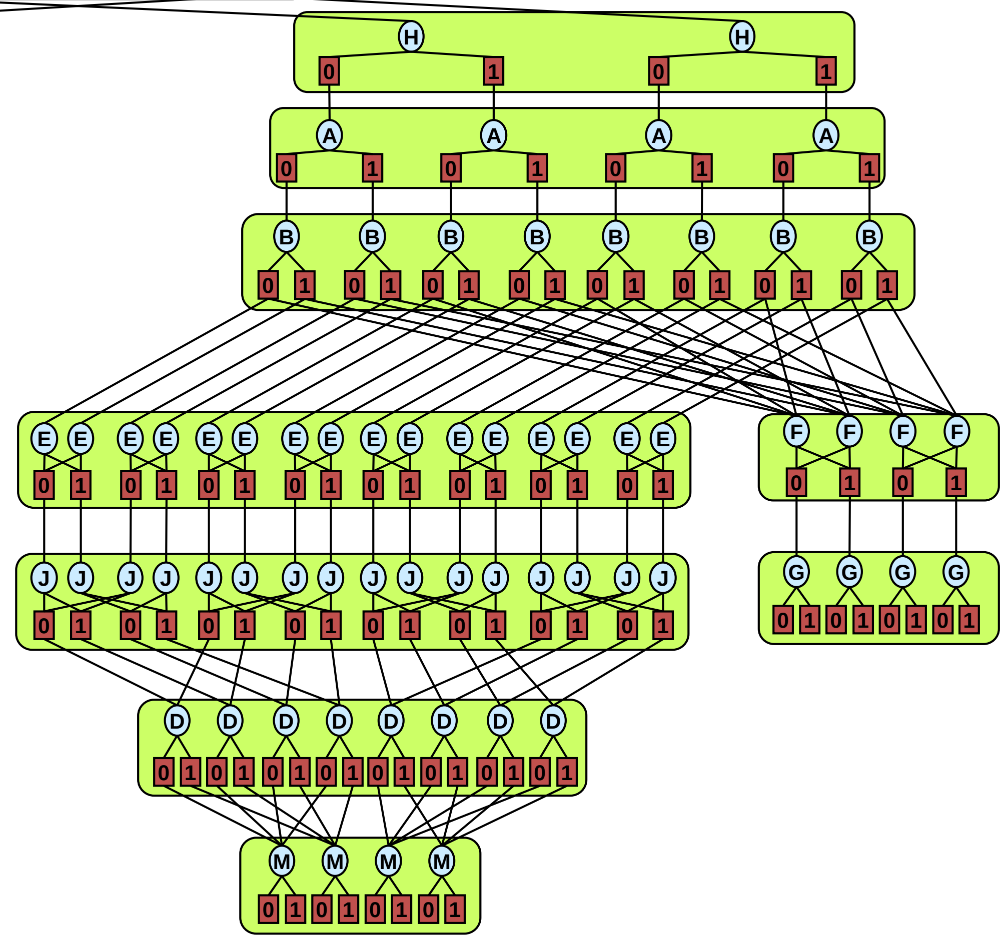
- 7 teams and 26 solvers:
  - INRA/ ONERA/ UPC/ LSIS (2)
  - UC-IRVINE (12)
  - UCLA (4)
  - UBC (1)
  - HUGIN (2)
  - Pittsburgh (3)
  - UPF/Radboud University, Nijmegen (2)

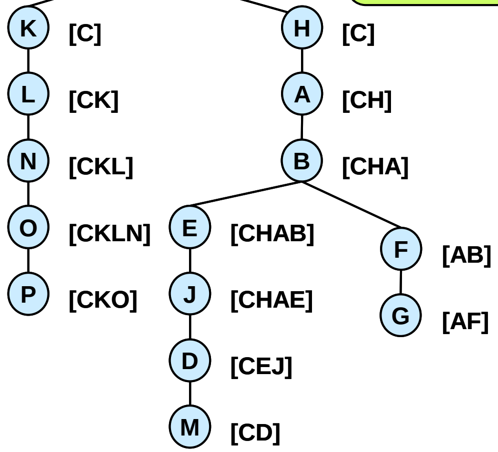
UC-Irvine

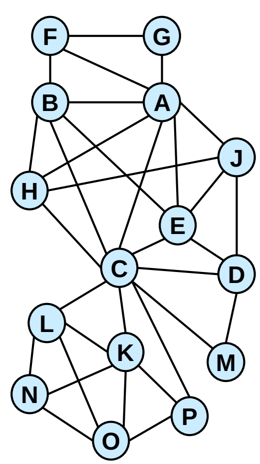
UCI Team: AOBB/AOBF

Radu Marinescu and Rina Dechter









http://graphmod.ics.uci.edu/group/Software

(C K H A B E J L N O D P M F G)

UCI Team: AOBB/AOBF

Radu Marinescu and Rina Dechter
http://graphmod.ics.uci.edu/group/Software

- Solver Type
  - Exact and anytime (AOBB), exact but not anytime (AOBF)
- Types of problems
  - Combinatorial optimization: mpe, weighted csps
- Types of networks
  - Bayesian and Markov
- Primary Method
  - Best-first and depth-first AND/OR search,
  - full context-based caching,
  - pre-compiled mini-bucket heuristic with an i-bound,
  - pesudo-tree guided by min-fill or hypergraph partitioning.
  - AOBB: Unit resolution for determinism, initial upper bound (gls+)
- **AOBB(12), AOBB(16), AOBB(20), AOBF(12), AOBF(16), AOBF(20)**

INRA/ ONERA/ UPC/ LSIS

Toulbar2 C++ solver

- Marti Sanchez[1], Sylvain Bouveret[2], Simon de Givry[1], Federico Heras[3], Philippe Jegou[4], Javier Larrosa[3], Samba Ndiaye[4], Emma Rollon[3], Thomas Schiex[1], Cyril Terrioux[4], Gerard Verfaillie[2], Matthias Zytnicki[1]
-

1. INRA, Toulouse, France
2. ONERA, Toulouse, France
3. UPC, Barcelona, Spain
4. LSIS, Marseilles, France

toulbar2

- Exact method, only for MPE task
- Depth-First Branch and Bound algorithm
  - Binary branching scheme instead of value enumeration
  - Dynamic variable and value ordering heuristics
  - Basic form of Conflict Back-Jumping (Lecoutre et al, ECAI 2006)
  - Variable elimination of small degree (2) during search (Larrosa et al, JAIR 2005)
- Pruning scheme
  - No initial upper bound
  - Lower bound produced by problem reformulation during search
    - Soft local consistency EDAC for binary (Heras et al, IJCAI 2005) and ternary (Sanchez et al, Constraints 2007) cost functions
    - Larger arity cost functions are delayed until they become ternary (their minimum cost is exploited in preprocessing only)

toulbar2 with tree decomposition (toulbar2/BTD)

- Depth-First Branch and Bound exploiting a Tree Decomposition (Terrioux et al, ECAI 2004) (Givry et al, AAAI 2006)
  - Min-fill tree decomposition heuristic (Marseilles' toolkit) in preprocessing
  - Root selection maximizing cluster size
  - Same search as toulbar2 inside clusters (DVO, CBJ, VE(2))
  - Full caching (no memory restriction)
- Russian Doll Search pruning scheme (Lemaitre et al, AAAI 1996)
  - Solves all cluster subtrees before solving the whole problem
  - Combines RDS, EDAC, and caching lower bounds

- Open-source available at http://mulcyber.toulouse.inra.fr/gf/project/toulbar2 (release 0.7)

UBC

GLS+: efficient local search for MPE

Frank Hutter
The University of British Columbia (UBC), Vancouver, Canada

Joint work with Holger Hoos (UBC) and
Thomas Stützle (Universite Libre de Bruxelles, Brussels, Belgium)

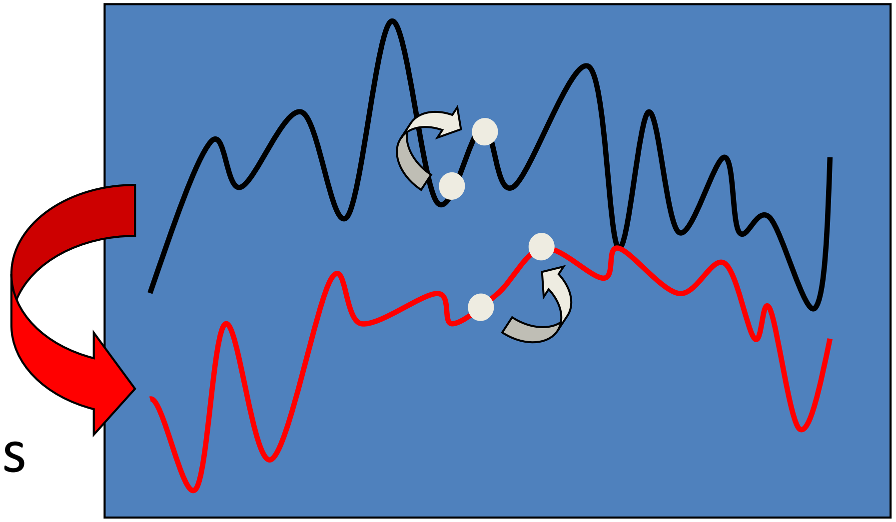
- Problem tackled: MPE
- Solver type: local search
- Characteristics:
  - Anytime algorithm
  - Often finds optimal solutions quickly, but can never prove optimality
  - Conceptually simple
  - Runtime is largely independent of tree width

Guided Local Search [Voudoris 1997]

Subclass of
Dynamic Local Search

Iteratively:

1) Local search ! local optimum
2) Modify evaluation function by
   penalizing some solution features



First applied to MPE by [Park, 2002]

- Solution features for MPE are partial assigments
- Penalties can be thought of as additional (temporary) factors
- Evaluation fct. = Objective fct. - sum of respective penalties

GLS$^+$ [Hutter, Hoos & Stützle, 2005]

- Differences to original GLS & avg. speedups
  - Modified evaluation function: » 10 times faster
  - Caching: » 10 times faster, more for larger instances
  - Parameter tuning: » 100 times faster (!)
  - Initialization: up to 10 times faster, doesn't always help

  ! With local search the devil is in the detail
- Optional pre-processing with partial Mini-Buckets
  - Large speedups for hard instances with low treewidth
  - Slowdowns for high treewidth ! disabled for UAI evaluation
- Code & datasets online (unchanged since 2005)
  http://www.cs.ubc.ca/labs/beta/Projects/SLS4MPE
- Possible uses
  - MPE solving under high treewidth & tight time constraints (where proven optimality is not important)
  - Initialize other algorithms (e.g. upper bound in B&B)

Hugin

Hugin Solver

Team: Frank Jensen

- Task: Exact MAR (node marginals)
- Built from standard Hugin software components (except the parser), implemented in C
- Handles only Bayesian networks (not Markov networks)
- Algorithm: Preprocessing + Junction-Tree
- Remove links to children from instantiated nodes
- Moralize
- Triangulate using the “Total Weight” method with max-number-of-separators = 4000
- Create the junction tree
- Propagate and compute all node marginals

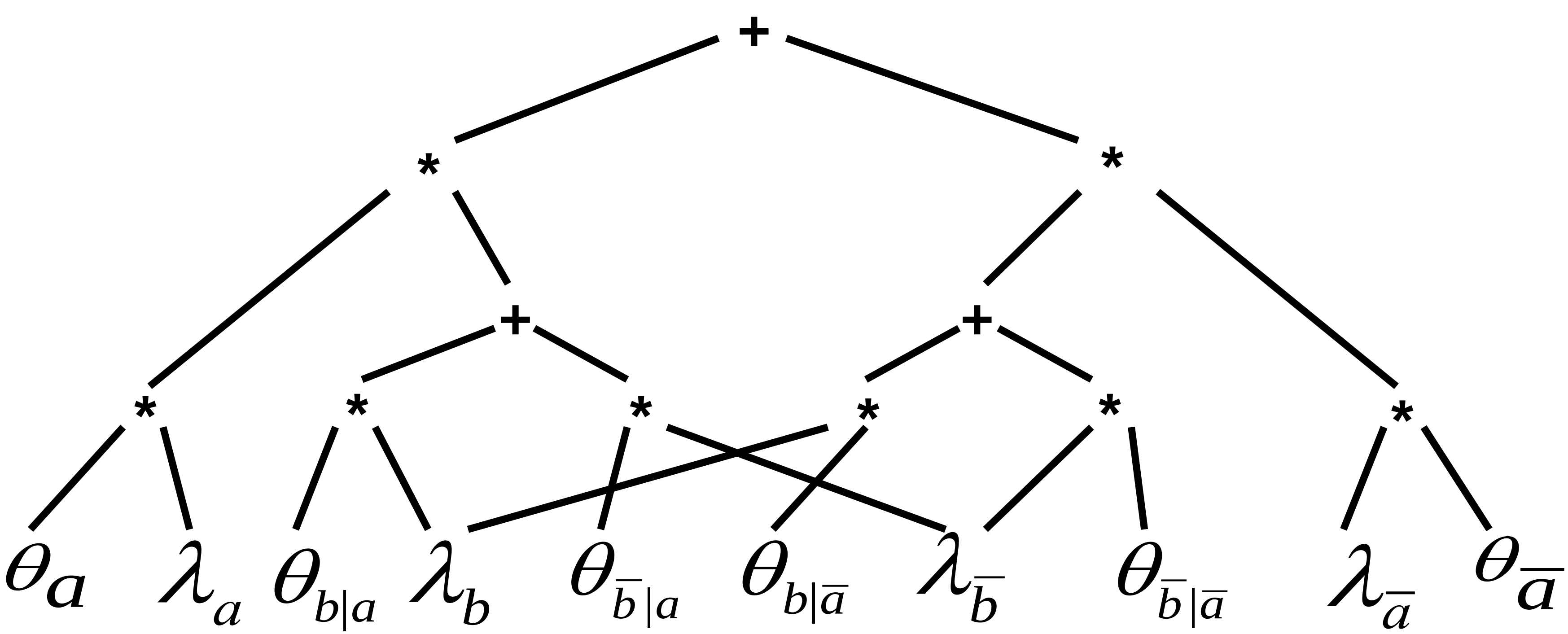
UCLA

ACE

ucla-ace-pe / ucla-ace-mar

Mark Chavira, Adnan Darwiche

- **Solver Type: Exact**
- **Problems: P(e) and marginals**
- **Networks: Bayesian and Markov**
- **Primary Method: Compilation into Arithmetic Circuits (ACs)**



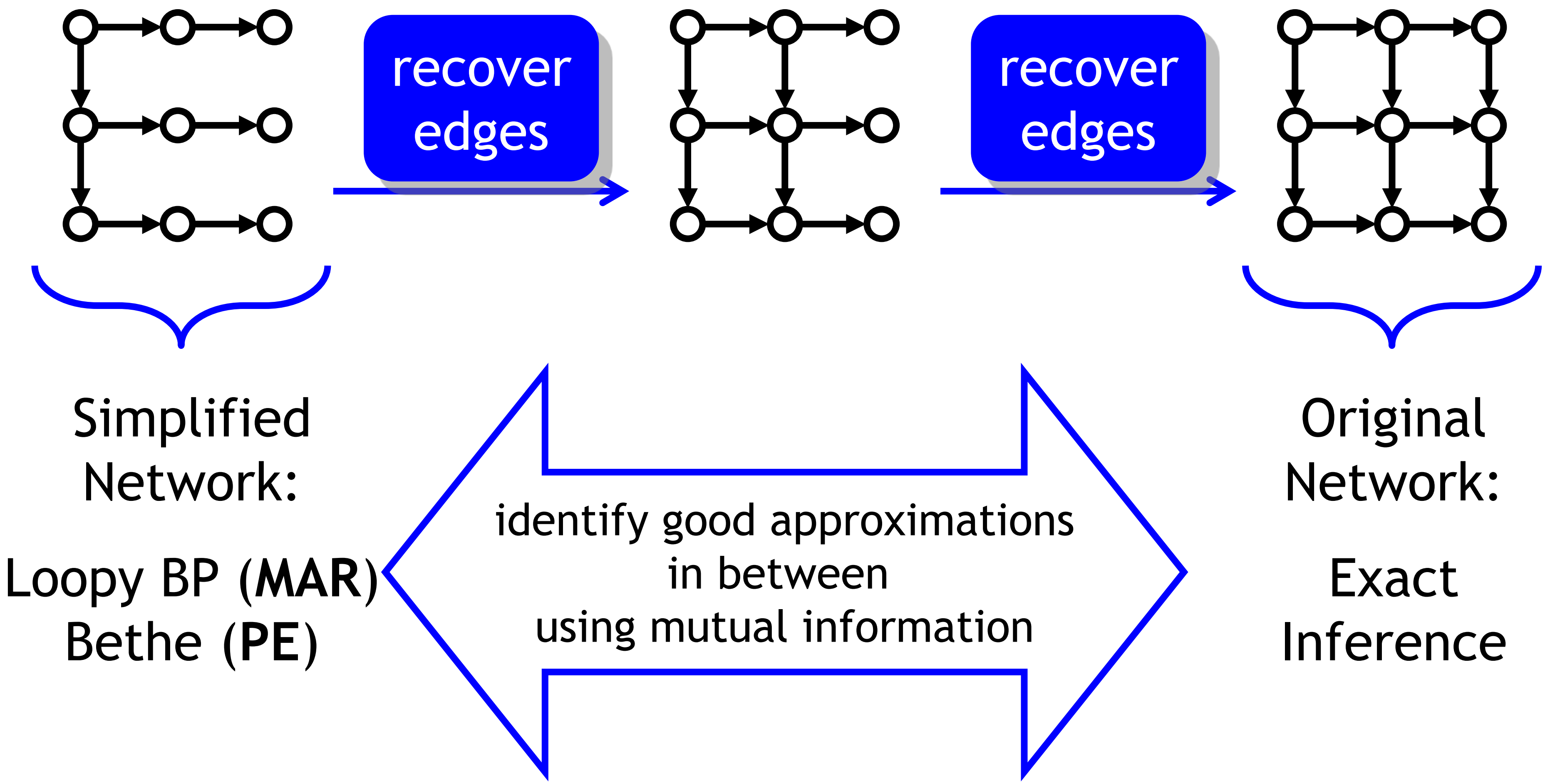
Algorithm

- Learn additional evidence
  - For each zero probability and each evidence, create a clause; run unit resolution
- Prune based on original and learned evidence
  - Remove evidence variables from tables
  - Remove variables appearing in a single table if not in query
- If minfill says problem is easy, apply variable elimination
  - If computing P(e), apply standard VE
  - If computing marginals, compile using VE
- If minfill says problem is difficult, apply knowledge compilation
  - Encode network into CNF
  - Encode determinism and equal parameters
  - Run c2d knowledge compiler

ED-BP

ucla-edbp-pe / ucla-edbp-mar

Arthur Choi, Adnan Darwiche



UPF/Radboud University, Nijmegen

Truncating the loop series expansion for BP

Vicenç Gómez   Hilbert J.Kappen

Department of Biophysics
Radboud University, Nijmegen, The Netherlands

Main ideas

- Iterative Belief Propagation (**IBP**) may provide an accurate approximation in loopy graphs.
- Explicit reconstruction of the exact inference can be done using **Loop Calculus** (Chertkov & Chernyak, '06).

Finite Loop expansion of the partition function Z, whose first term Z_{BP} corresponds to the **Bethe Free energy** (obtained using IBP):

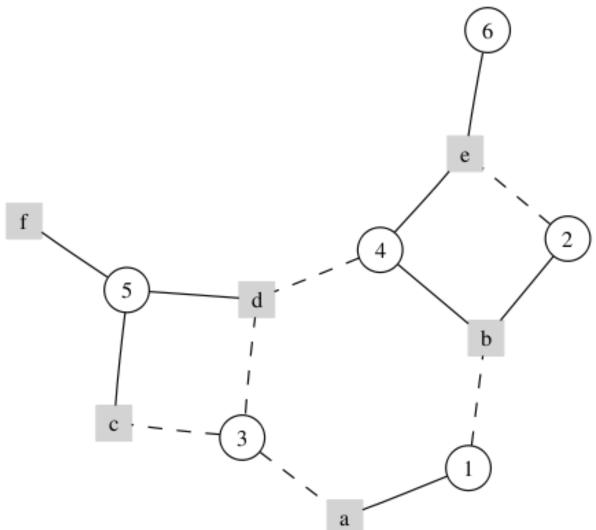
$$Z = Z_{BP}\left(1 + \sum_{\mathcal{C}} r(C)\right),$$

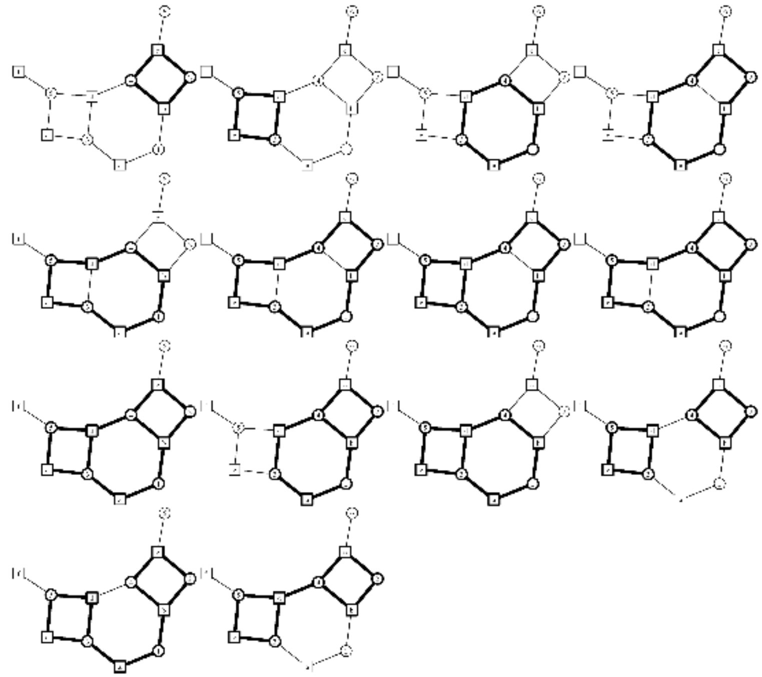
Each $r(C)$ corresponds to a **generalized loop** (*subgraph with all nodes with degree at least two*) and can be computed at the fixed point of IBP.

UAI '08 workshop

Truncated Loop Series expansion for Belief Propagation (TLSBP)

Example:





We propose the following solver inspired in (Gómez et al '07):

1. Run **IBP**.
2. Bounded **search** by length ℓ of *simple* loops (degree exactly 2).
3. **Merge** loops iteratively until no new loops of length $\leq \ell$ are found.
4. Compute the approximated partition function $Z_{\mathcal{C}'}$ considering a subset $\mathcal{C}' \subseteq \mathcal{C}$.

Characteristics of TLSBP solver

- Using bitsets to represent generalized loops, operations can be done efficiently. Ex: **merge** $\equiv$ bitwise OR.
- **MAR** task is approximated using **conditioning**: $P_i(x_i) = \frac{Z^{x_i}}{\sum_{x_i'} Z^{x_i'}}$
- Constrained to binary variables.
- Anytime algorithm for approximate inference.
- Combines propagation and search.
- Approximates **PE** and **MAR** tasks.

References:

- Gómez V., Mooij J. M., Kappen H. J. "Truncating the loop series expansion for belief propagation", *JMLR* 8(Sep):1987-2016, 2007
- Chertkov M., Chernyak V. Y., "Loop series for discrete statistical models on graphs", *J. Stat. Mech.* P06009, 2006

University of Pittsburgh

Pr(E) & marginals

Foundations of the algorithms (*SMILE*☺)

1. Clustering algorithm at the foundation of the program [Lauritzen & Spiegelhalter] (Pr(E) as the nor
2. Relevance reasoning, based on Geiger 1990], as structured in [summarized in [Druzdzel & Sue
3. For very large models: Relevan 1997] and Relevance-based Inc 1999].

Relevance steps:

1. In p(E), focusing inference on the evidence nodes
2. Removal of barren nodes
3. Evidence absorption
4. Removal of nuisance nodes
5. Reuse of valid posteriors

Full references are included in *GeNIe* on-line help, http://genie.sis.pitt.edu/.

Engineering (Tomek Sowinski).

- C++ implementation (*SMILE*☺)
- Extensively tested (over eight years of academic and industrial use)

Approximate marginals: EPIS-BN

- **Importance sampling combined with loopy belief propagation [Yuan & Druzdzel, 2003].**
- **SMILE☺ algorithm runs for a predefined number of samples; for the competition we run forever.**
- **No checking for convergence.**
- **No special treatment of determinism, no caching.**

UC-Irvine

UCI Team: VEC

Vibhav Gogate and Rina Dechter

http://graphmod.ics.uci.edu/group/Software

- Solver Type
  - Both Exact and Anytime
- Task: Probability of Evidence, Partition function
- Primary Method.
  - Variable Elimination + Conditioning (Pearl ’88)
  - SAT based singleton consistency

VEC Algorithm

- Algorithm (Network P)
  - Reduction Step (Input: P, Output: P')
    - Convert the zero probabilities in P to a SAT problem F
    - For each variable-value pair X=a
      - if (F and X=a) has no solutions (use minisat Een and Sörensson 06)
        - » Remove X=a from P
  - If the reduced network P' has a "reasonable" treewidth
    - Solve using Bucket elimination
  - Else
    - Remove K variables from P' so that its treewidth is reasonable.
    - z=0
    - For all value assignments $\mathbf{X_k=k}$ to the K variables
      - If (F and $\mathbf{X_k=k}$) has a solution
        - » z=z+Bucket-elimination(P' | $\mathbf{X_k=k}$)
  - Return Z

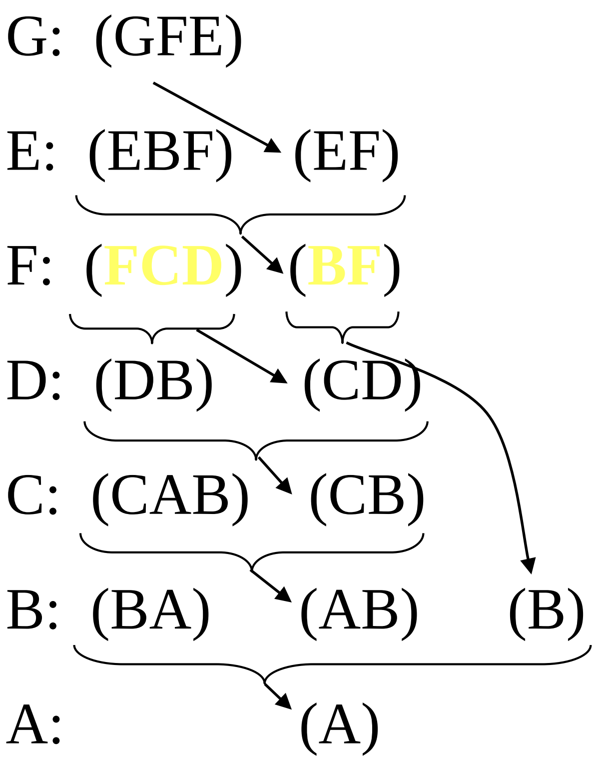
UCI Team: IJGP

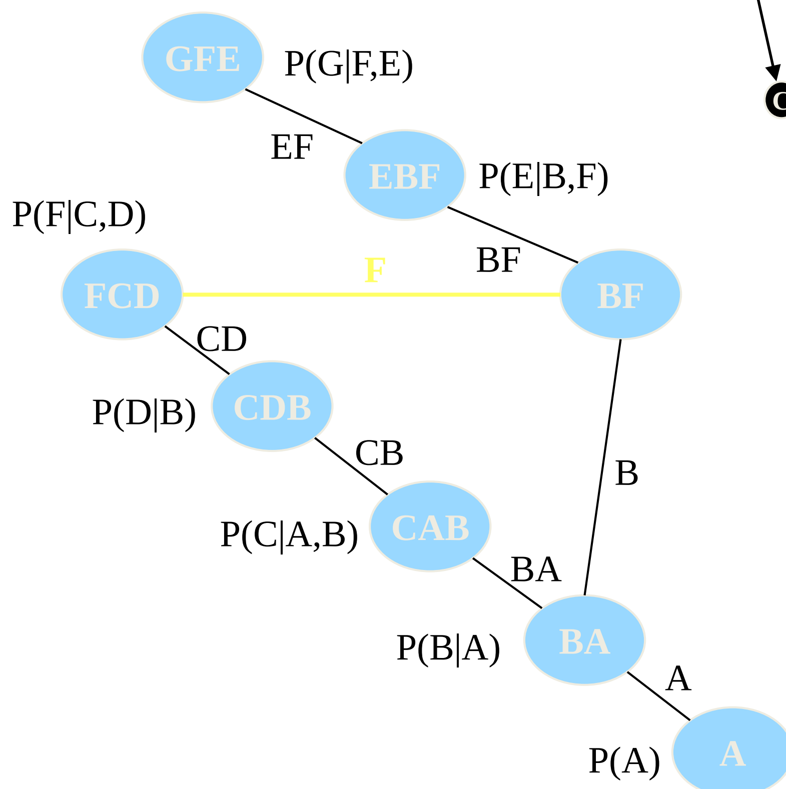
Vibhav Gogate and Rina Dechter

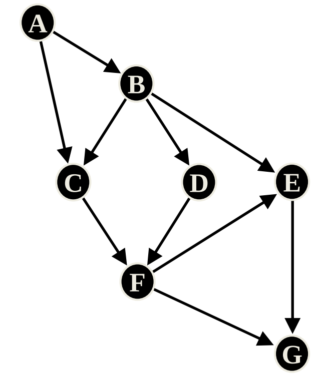
http://graphmod.ics.uci.edu/group/Software

- Solver Type
  - Approximate
- Task: Marginals
- Primary Method.
  - Iterative Join Graph propagation (Dechter-Kask-Mateescu, 2002)
    - A variant of Generalized Belief Propagation
  - Algorithm (Network P)
  - Reduction Step (Input: P, Output: P’)
    - Sat based reduction as in VEC
  - For i=1 to treewidth do
    - Run IJGP(i) until convergence
    - Report the marginals from the output of IJGP

Structuring IJGP







a) schematic mini-bucket(i), i=3

b) arc-labeled join-graph decomposition

Run Belief propagation on the arc-labeled join-graph until convergence

UCI Team: SampleSearch

Vibhav Gogate and Rina Dechter

- Solver Type
  - Approximate
- Task: Both Marginals and P(E)
- Primary Method.
  - SampleSearch
  - Importance Sampling whose proposal distribution is computed from the output of IJGP
- Algorithm:
  - Reduction Step (Input: P, Output: P')
    - Sat based reduction as in VEC
  - Run IJGP (i=3)
  - Run SampleSearch with proposal from output of IJGP(i=3)

SampleSearch (Gogate and Dechter , 2007)

- Importance sampling may suffer from the rejection problem
  - zero weight samples
- Introduce backtracking search in sampling
  - Search until a non-zero weight sample is found
- Samples from the backtrack-free distribution
  - Proposal distribution with all zero weight tuples removed

Evaluating Solvers

Evaluating MPE Solvers

MPE results

- 9 Solvers:
  - *inra** : Anytime, exact
  - *aobb** : Anytime, exact
  - *aobf** : not anytime, exact
  - *ubc*: anytime, approximate
- Benchmarks specifics:
  - 32 Grids and 35 Relational instances.
  - No Diagnose.
- Measures: cumulative times, number-solved, average error

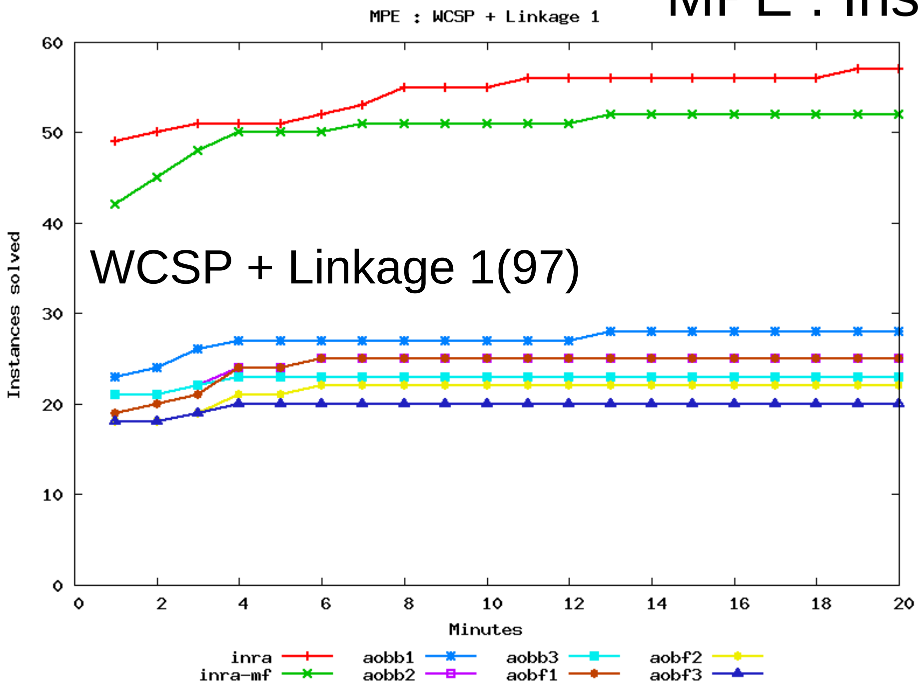
MPE results

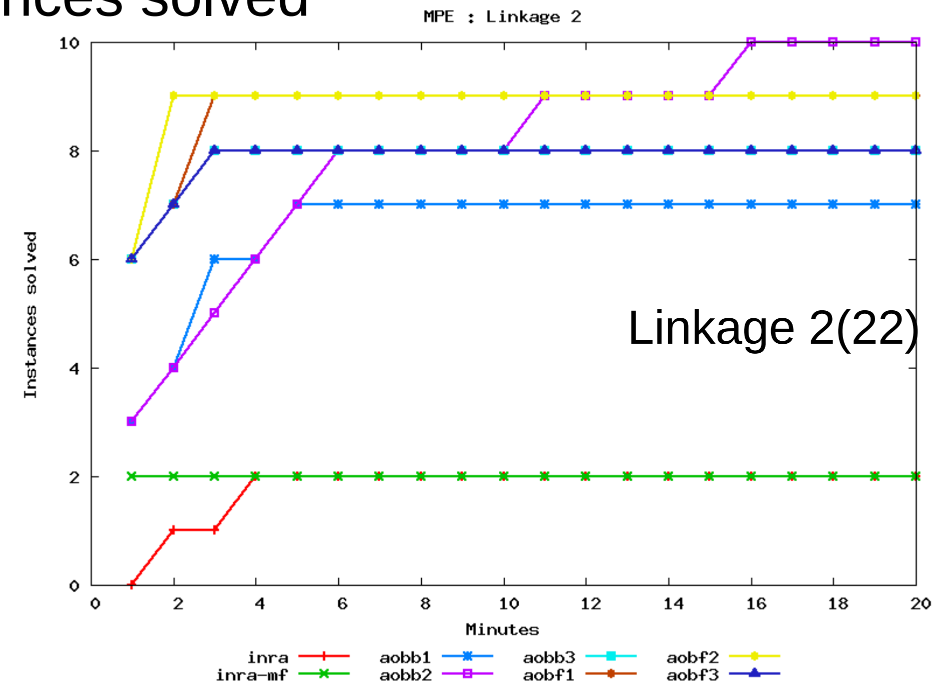
- 9 Solvers:
  - *inra** : Anytime, exact
    - Branch and Bound style algorithm (Schiex, Jegou, Larossa et al.)
    - Toulbar solver available online
  - *aobb** : Anytime, exact
    - Branch and Bound style algorithm (Marinescu and Dechter, 2006)
  - *aobf** : not anytime, exact
    - Best first search style algorithm (Marinescu and Dechter, 2006)
  - *ubc*: anytime, approximate
    - Local Search technique
- Benchmarks specifics:
  - 32 Grids and 35 Relational instances.
  - No Diagnose.
- Measures: cumulative times, number-solved, average error

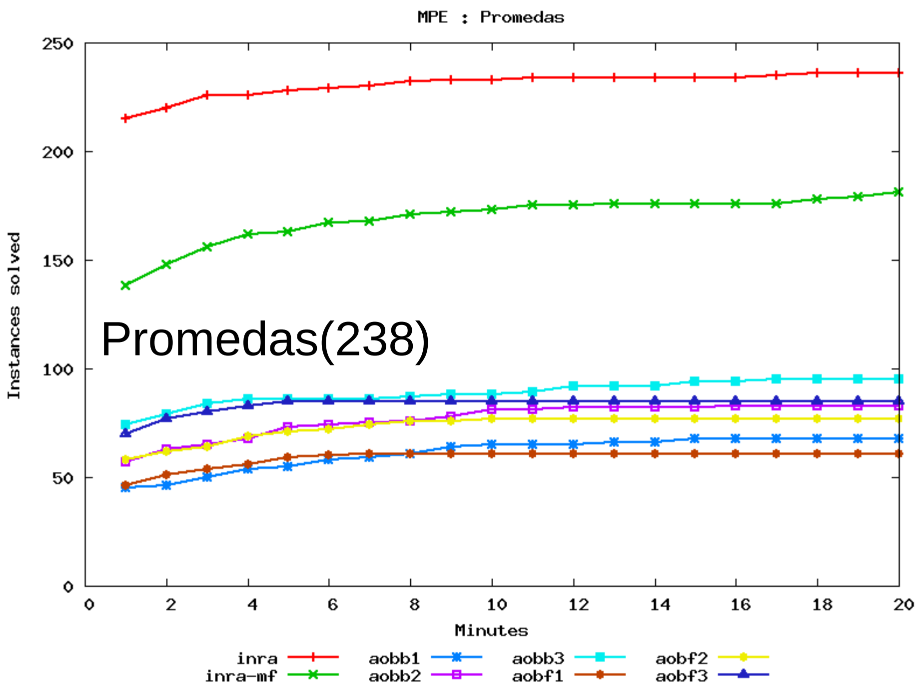
MPE : Instances solved

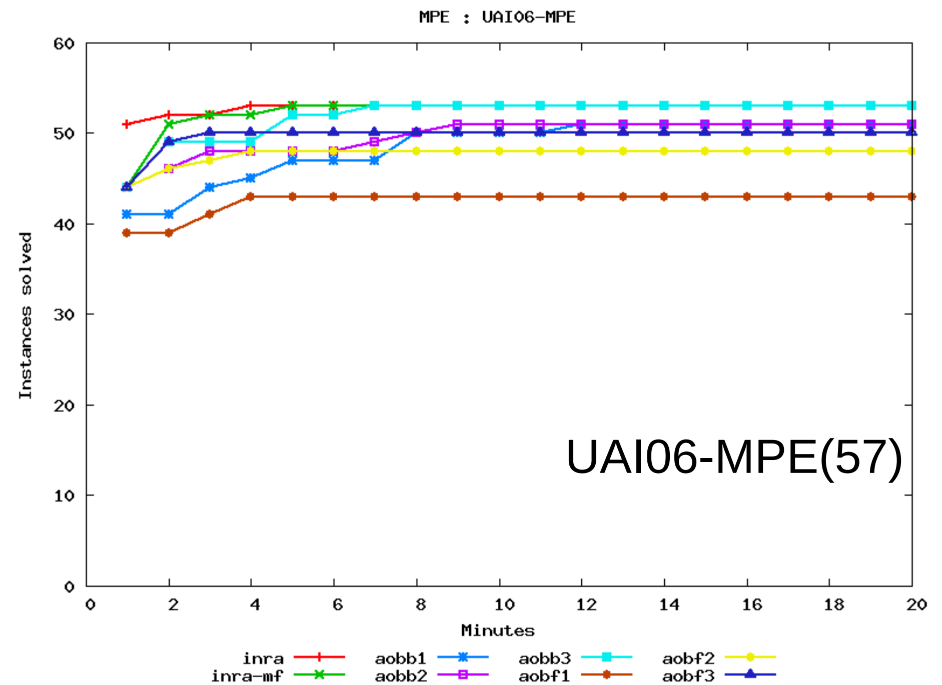
- Plot number of instances solved over time.
  - "Solved" = solver reports solution and terminates with exit status 0.
    - Note: some runs of INRA (~1%) seem to not produce exact results.
- Only for 8 exact solvers.
- Plotted per benchmark class.

MPE : Instances solved

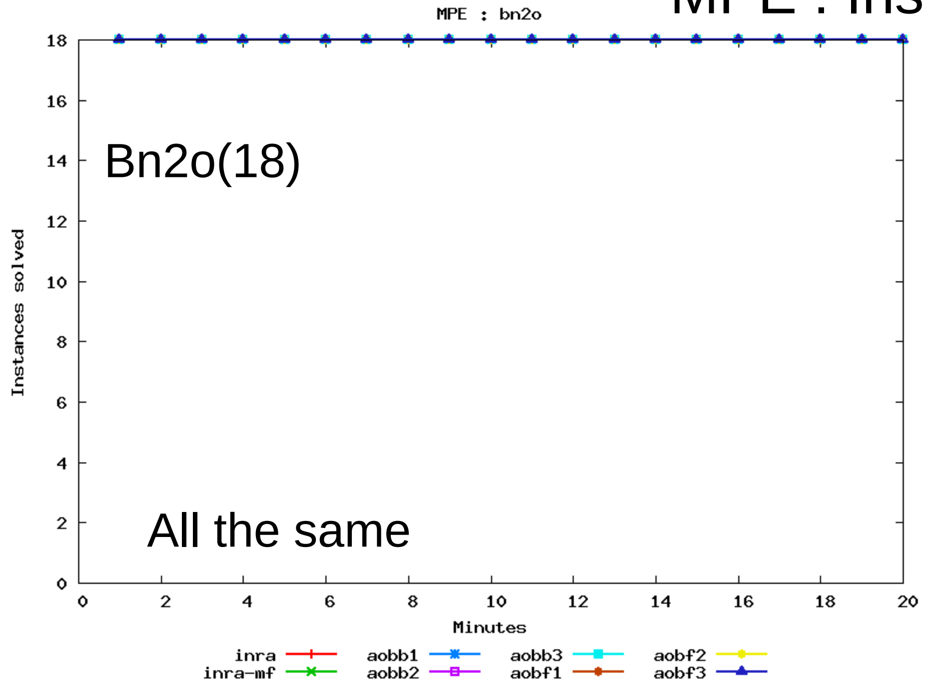


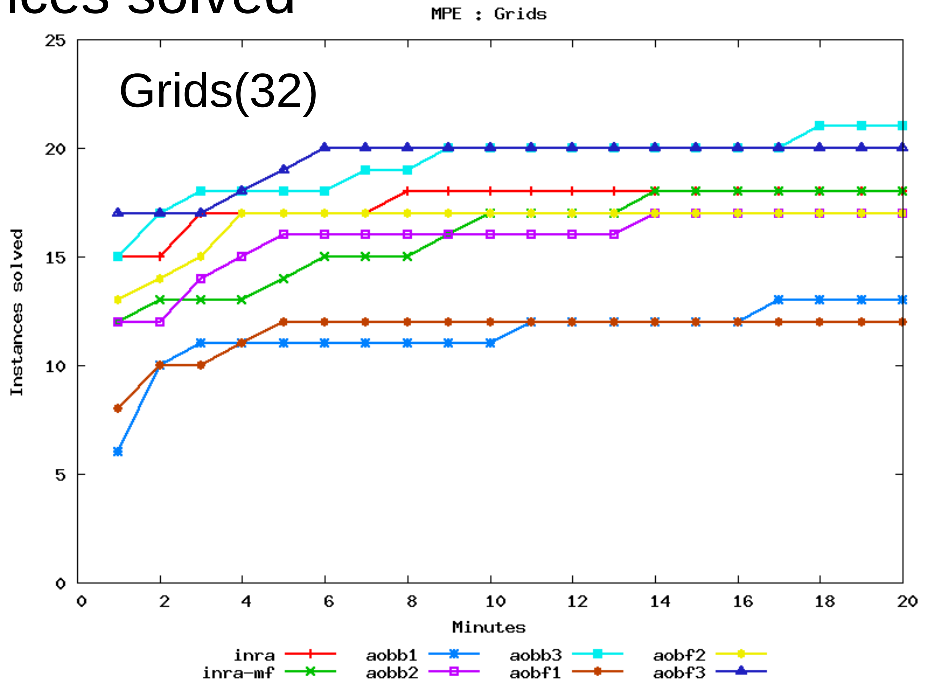


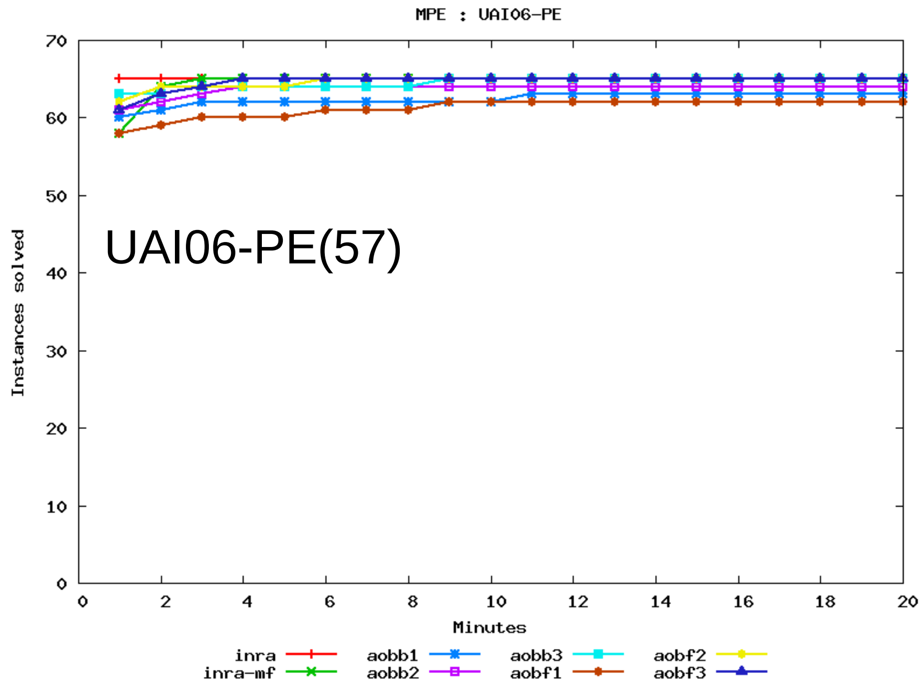


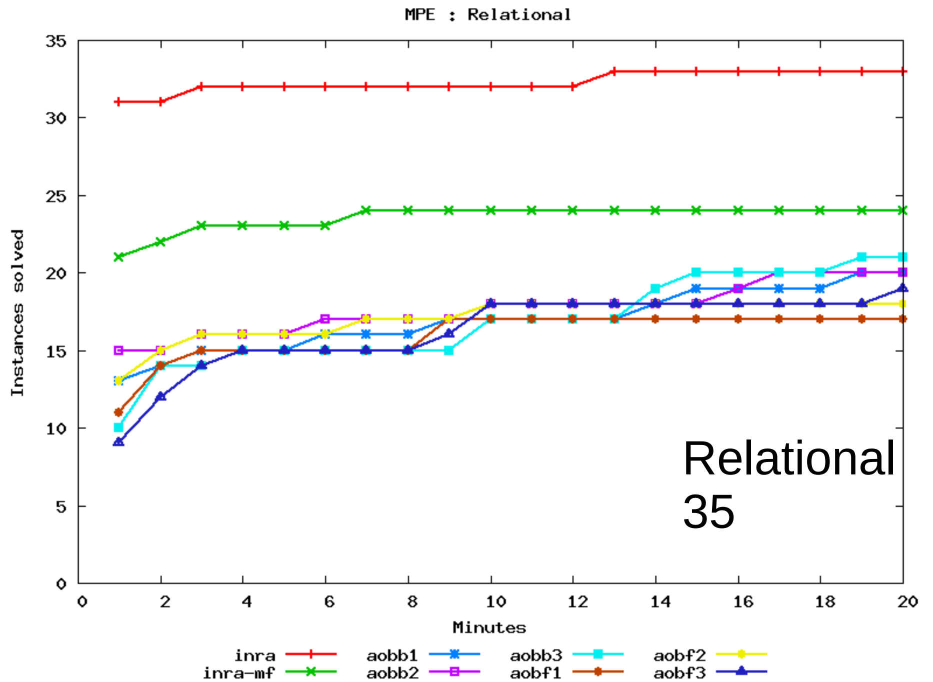


MPE : Instances solved



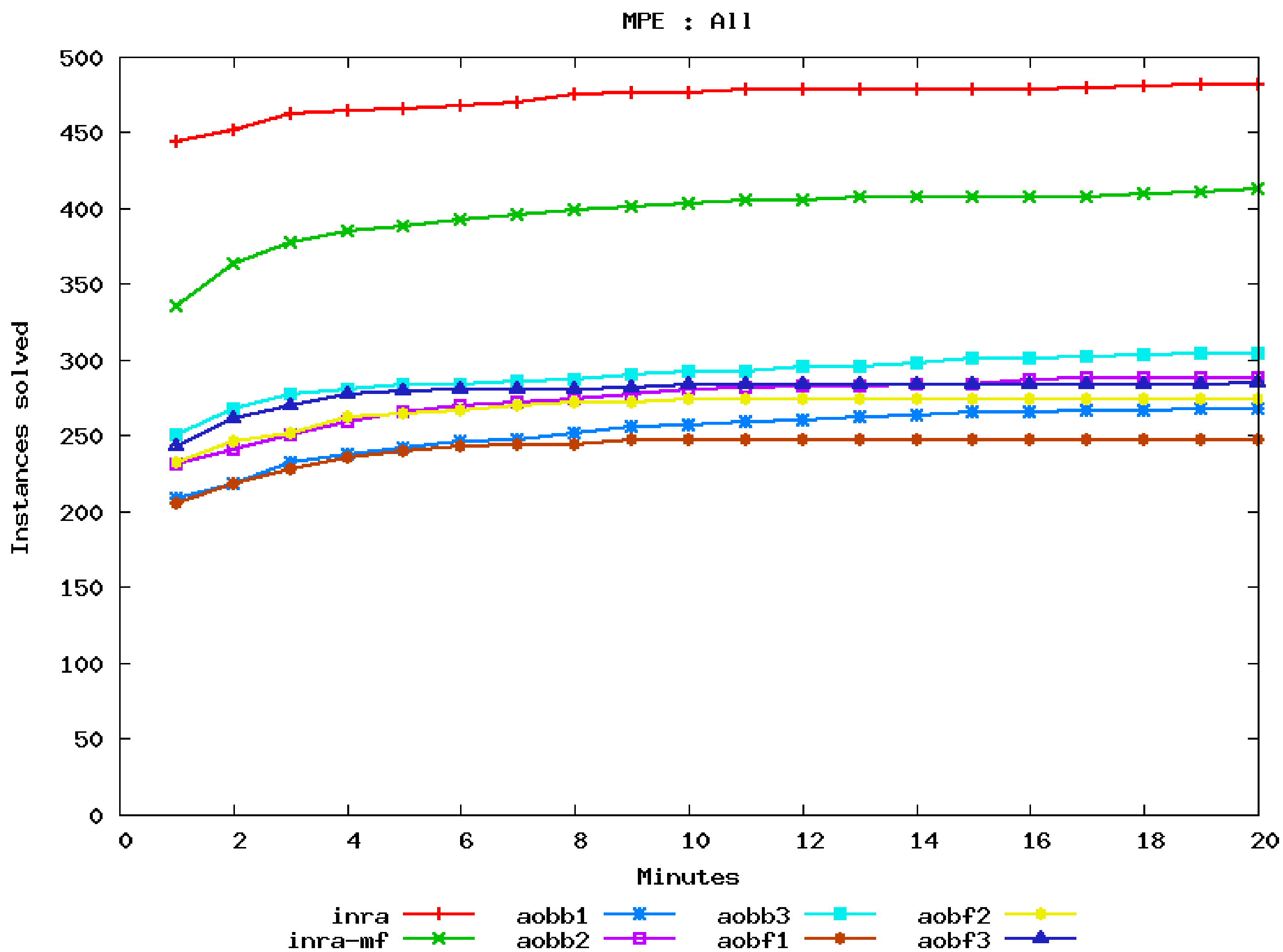






MPE : Instances solved overall

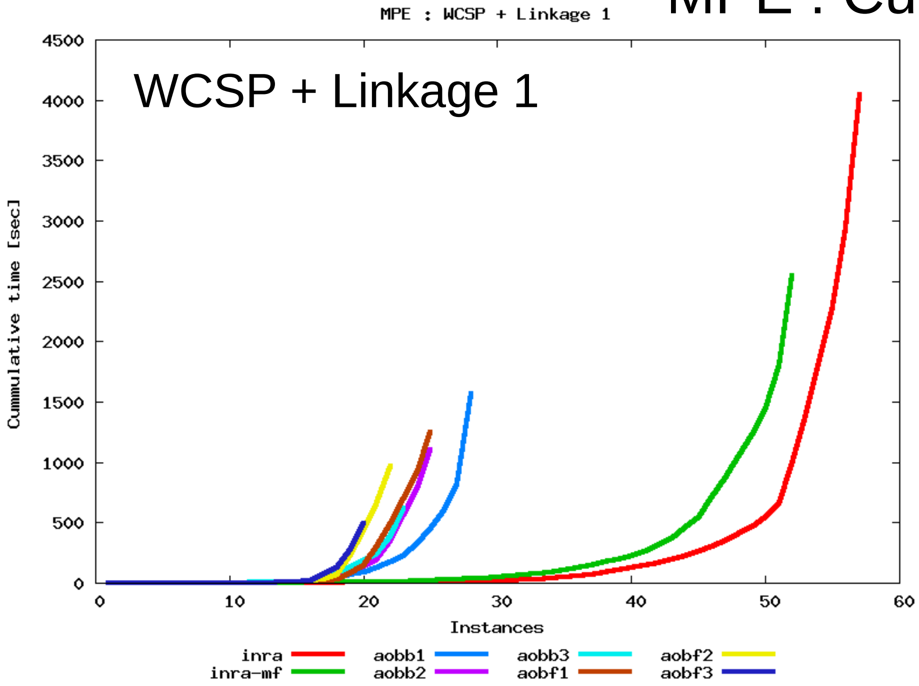
- Note: Not weighed by problem class size, biased to some classes/solvers.

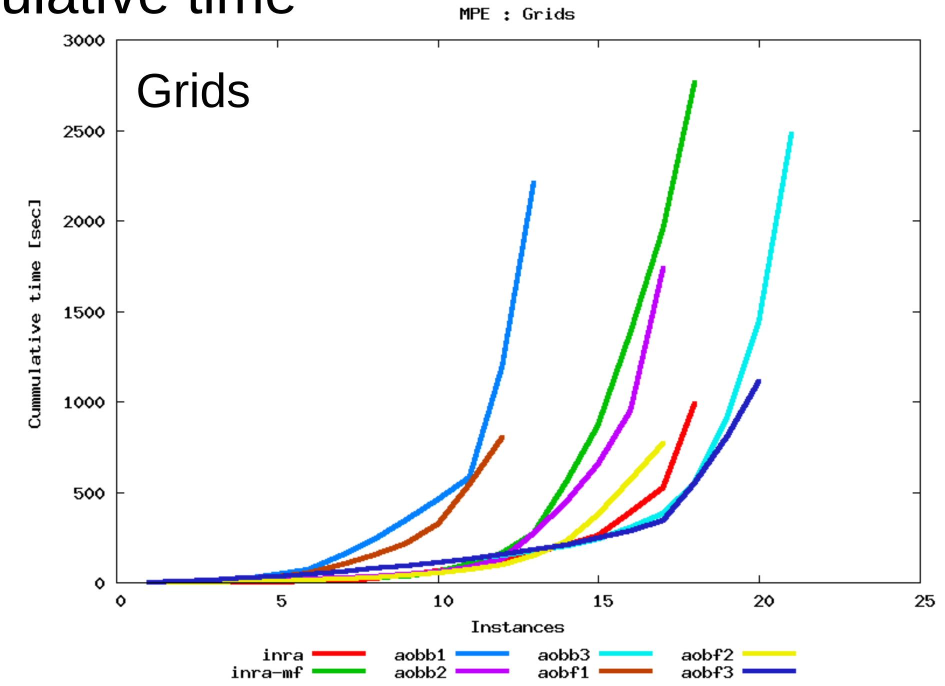


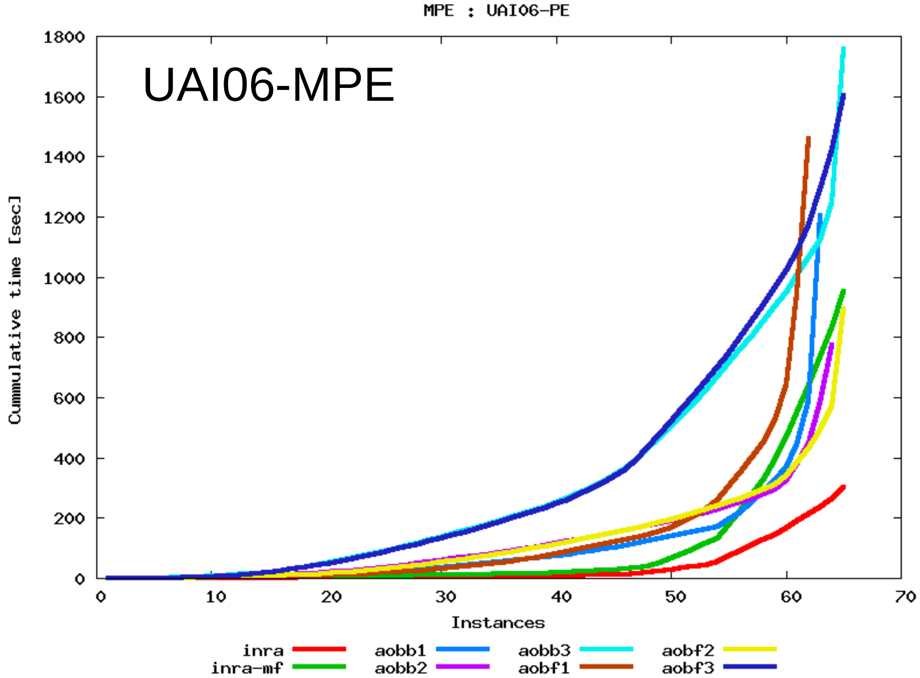
MPE : Cumulative time

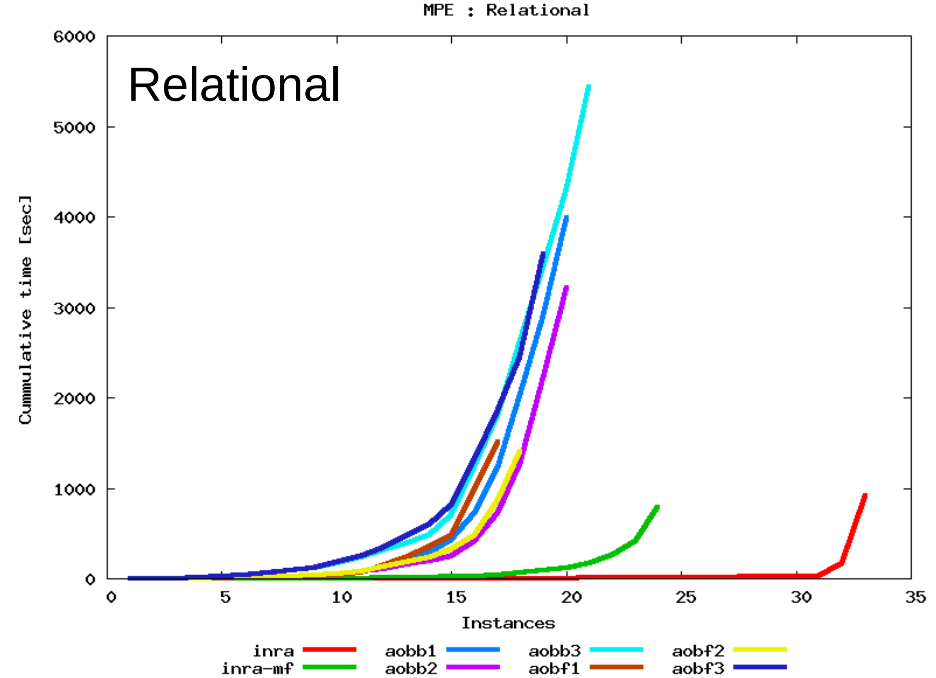
- For each solver:
  - Order solved instances by time.
  - Value at point *x* : Cumulative time to solve first *x* instances.
- Interpretation:
  - Further right = more instances solved.
  - Lower = less time needed to solve instances.
- Only for 8 exact solvers.
- Plotted per benchmark class and overall.

MPE : Cumulative time

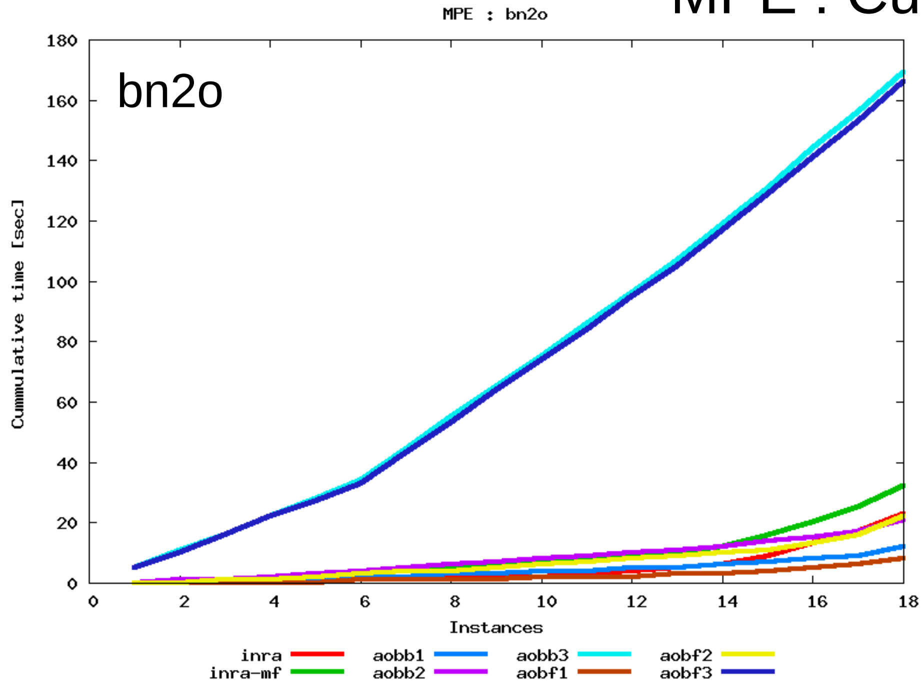


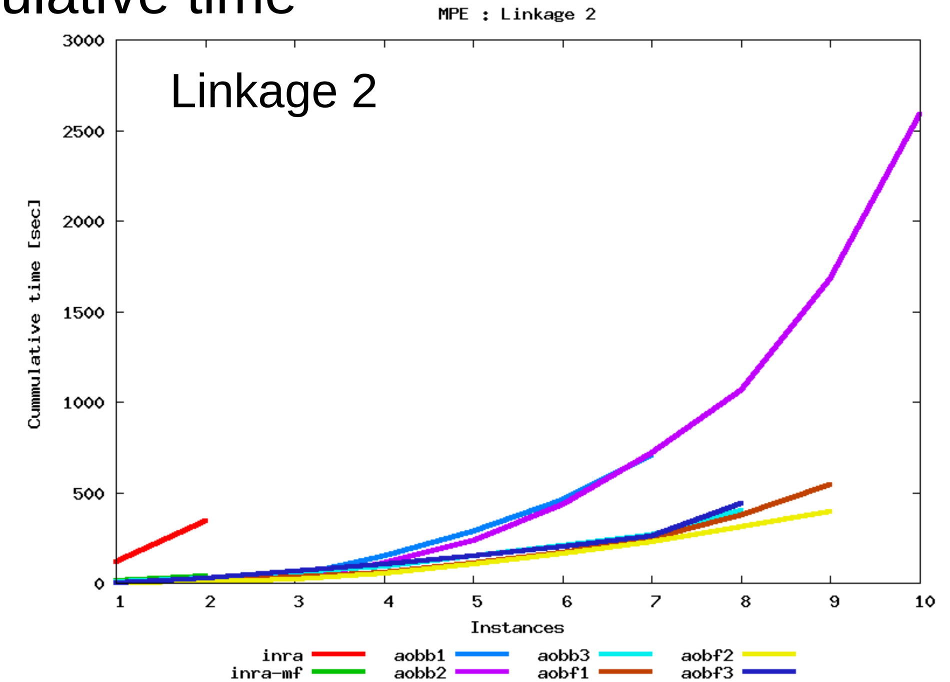


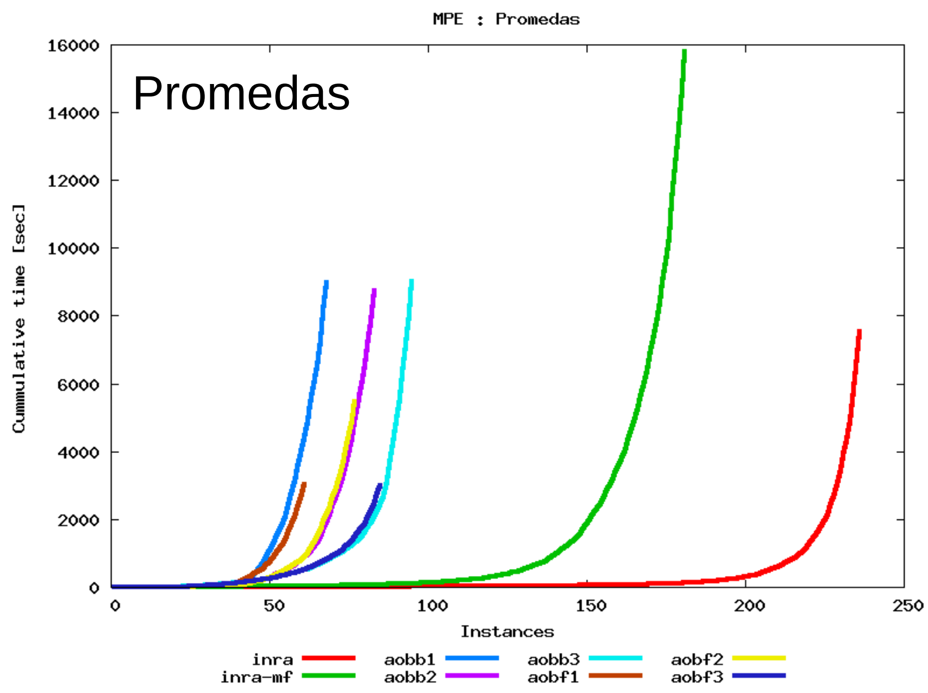


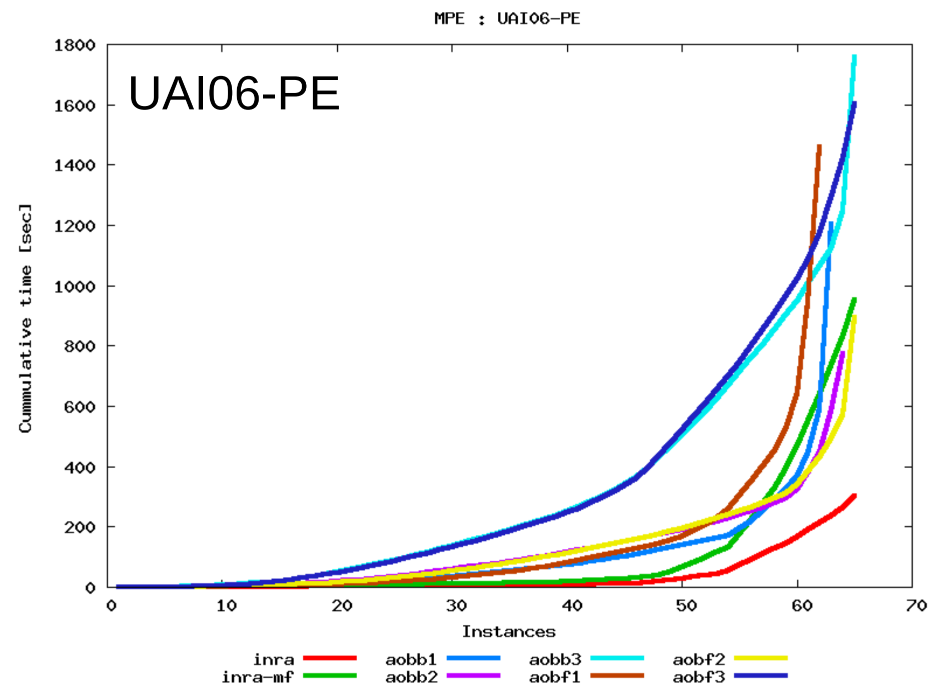


MPE : Cumulative time



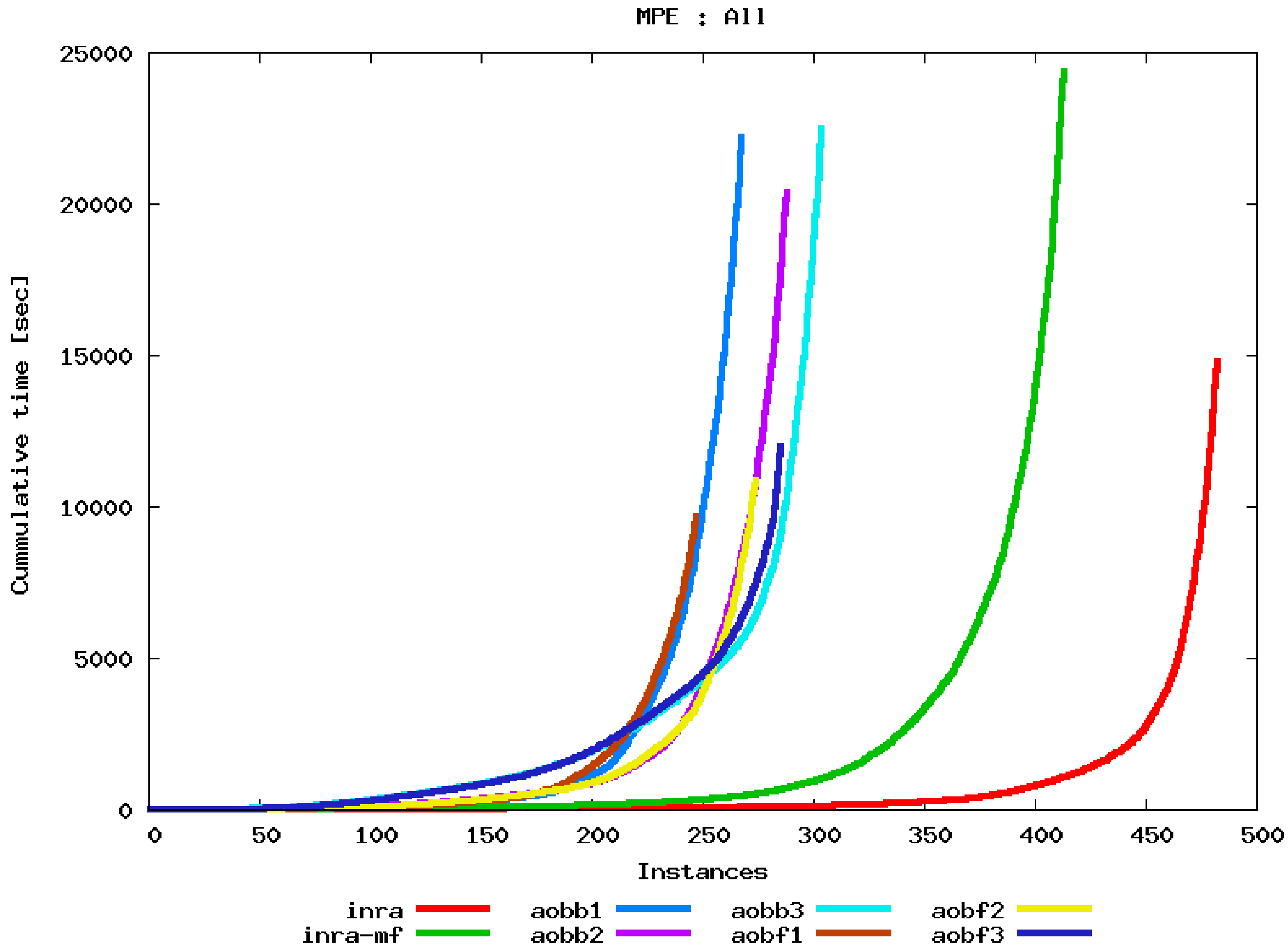






MPE : Cumulative time overall

- Note: Not weighed by problem class size, biased to some classes/solvers.



Summary

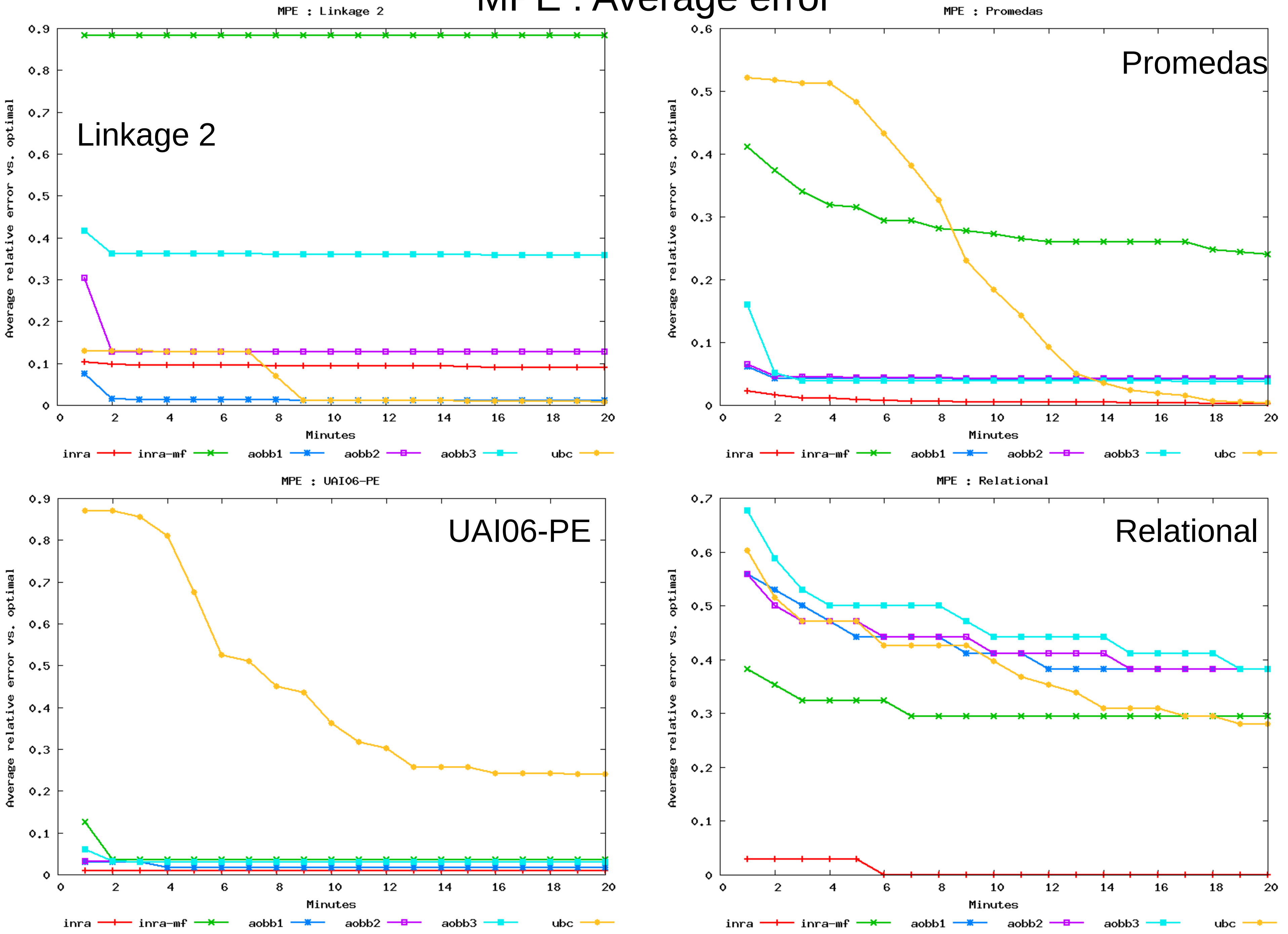
- **#solved**
  - **INRA best on 3**: promedas, wcsp, relational
  - **AOBB3 best on 2**: linkage, grids
  - On 3: performance comparable

- **Cumulative time**
  - **INRA** best on 5
  - **AOBB** best on 3

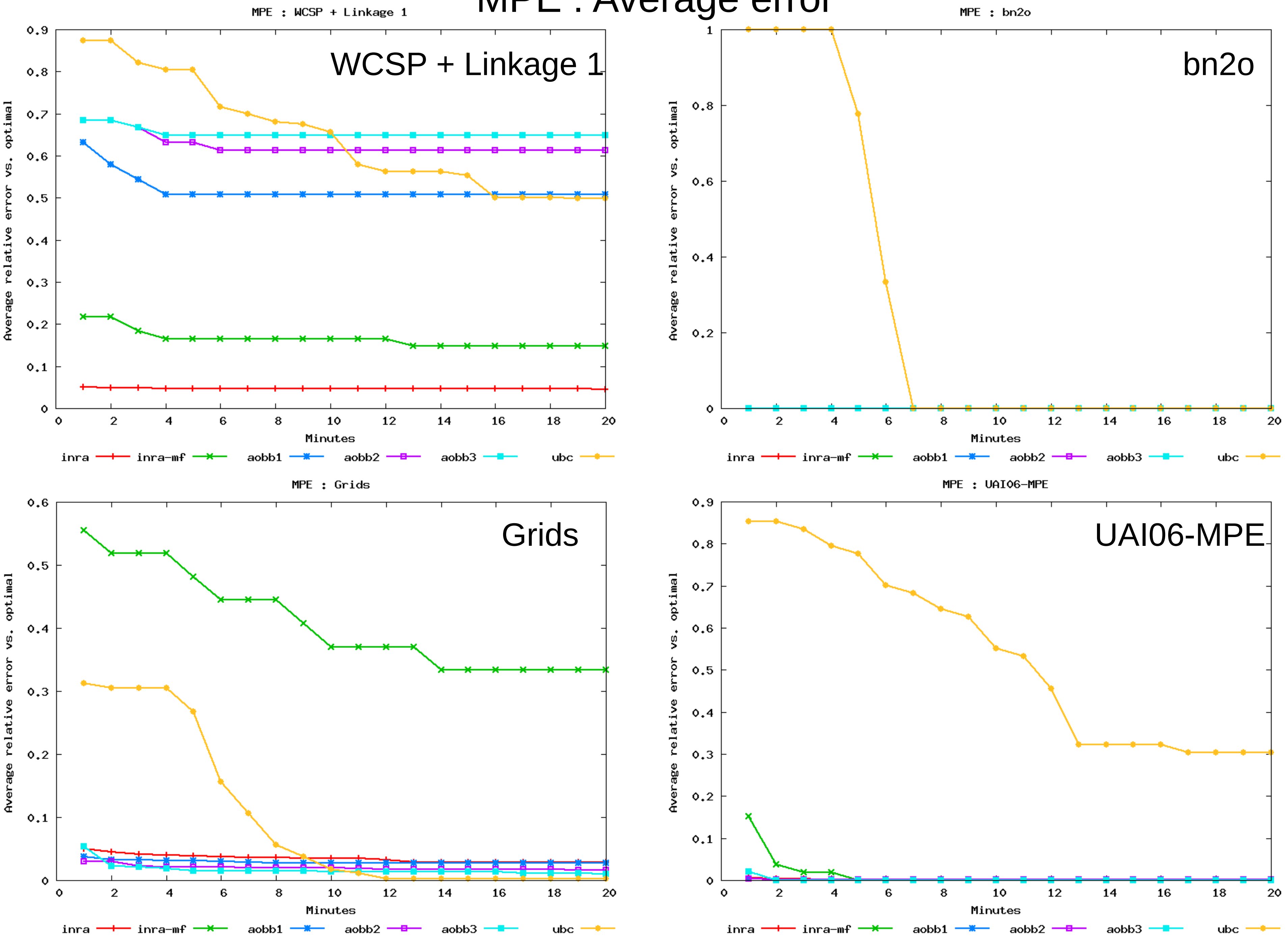
MPE : Average error

- Only for anytime solvers (not AOBF, with ubc).
- Collect subset of solved instances.
  - Get solution z from exact solvers.
- For each solver and instance, look at approximate solution at time t :
  - No solution present : *Error = 1*
  - Solution z' present: *Error =*
    $0.5 * (1 - 10^{- | (\log z - \log z') / \log z |})$
  - Average over instances.
  - Lower error score is better

MPE : Average error

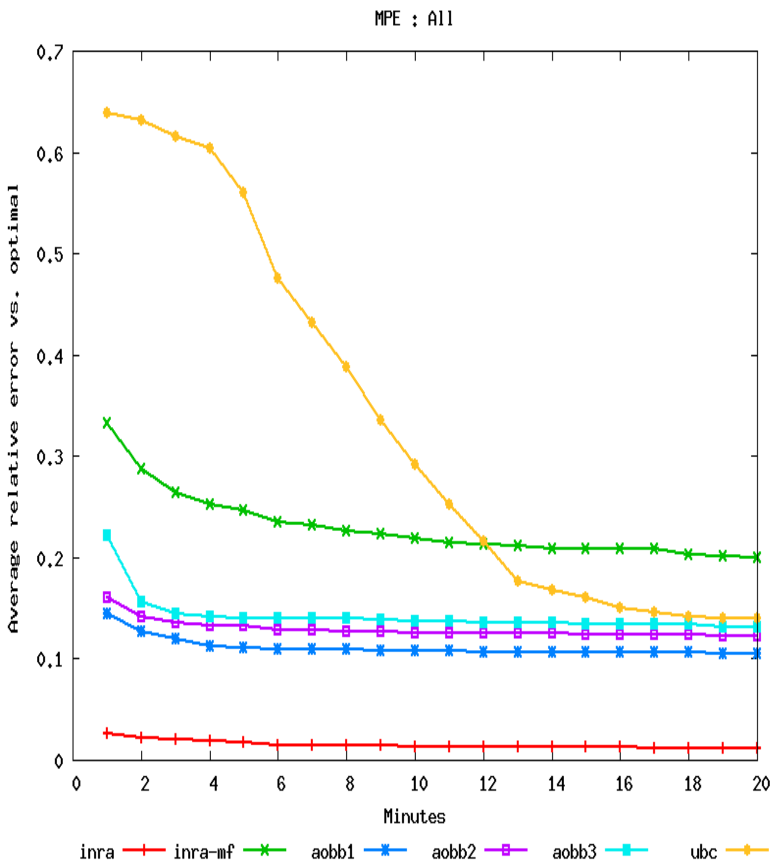


MPE : Average error



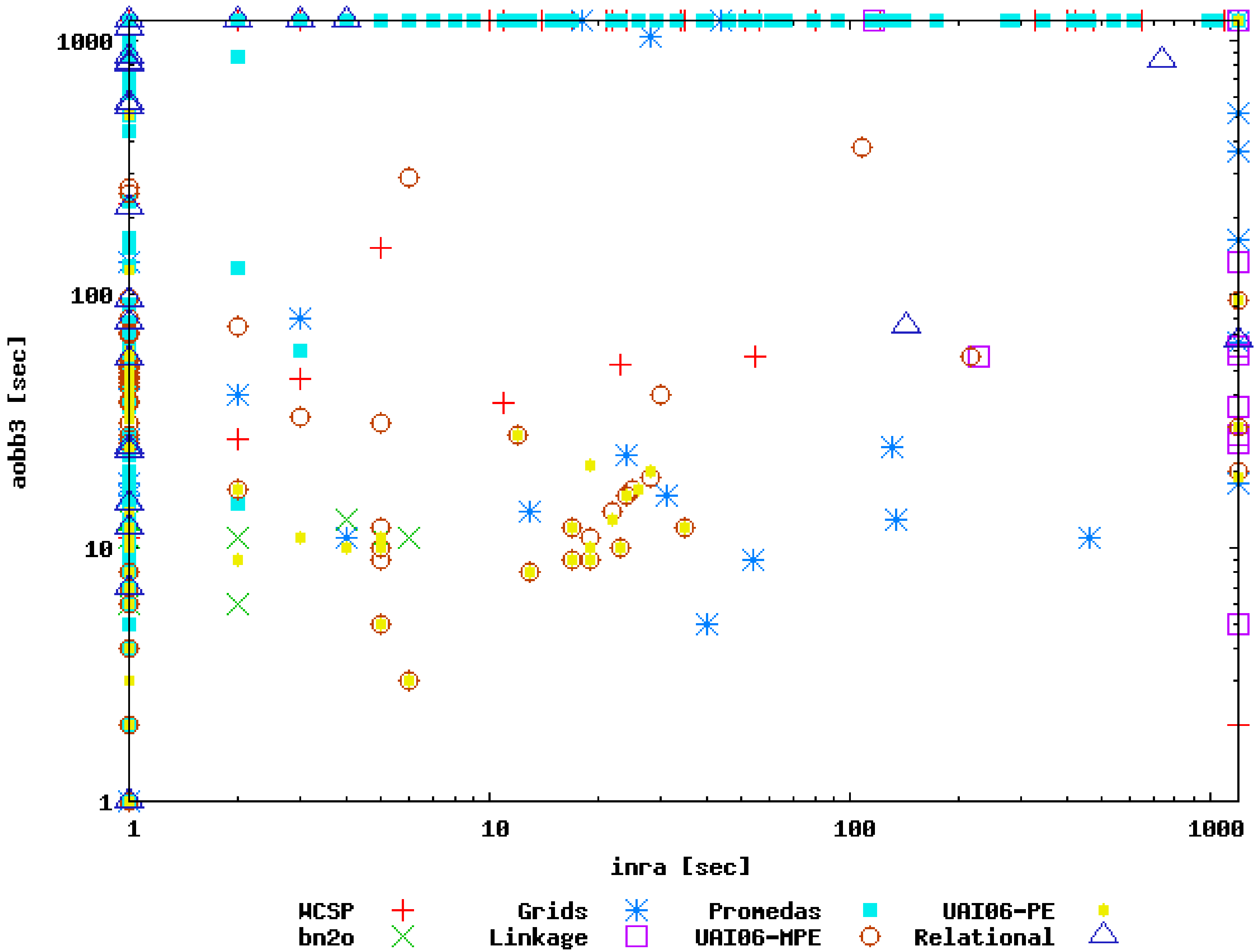
MPE : Average error overall

- Note: Not weighed by problem class size, biased to some classes/solvers.



- **INRA** better by far on **3**
- **AOBB1** better by far on **1**
- **AOBB3** better **1**
- **UBC** better on **1**
- **Anytime behavior:**
  - **INRA's change very little with time**
  - **AOBB changes somewhat with time**
  - **UBC is truly anytime**

Scatter diagrams, exact mpe depicting times for Inra vs AOBB3



Evaluating Exact PE/MAR Solvers

Benchmark Summary (exact)

- 9 sets:

		Bys	Mkv	bin
– weighted-CSP	(97)		•	
– bn2o (diagnosis)	(18)	•		•
– hand-built	(100)	•		
– grids	(320)	•		•
– linkage	(22)		•	
– Promedas	(238)		•	•
– UAI-06 (MPE)	(57)	•		
– UAI-06 (PE)	(78)	•		
– relational	(251)	•		•
– **TOTAL**	**(1181)**	**(824)**	**(357)**	**(507)**

Exact PE results

- 5 Solvers:
  - Bayes only:
    - *hugin-single, hugin-double, pitt-pe*
  - Bayes and Markov:
    - *irvine-vec, ucla-ace-pe*

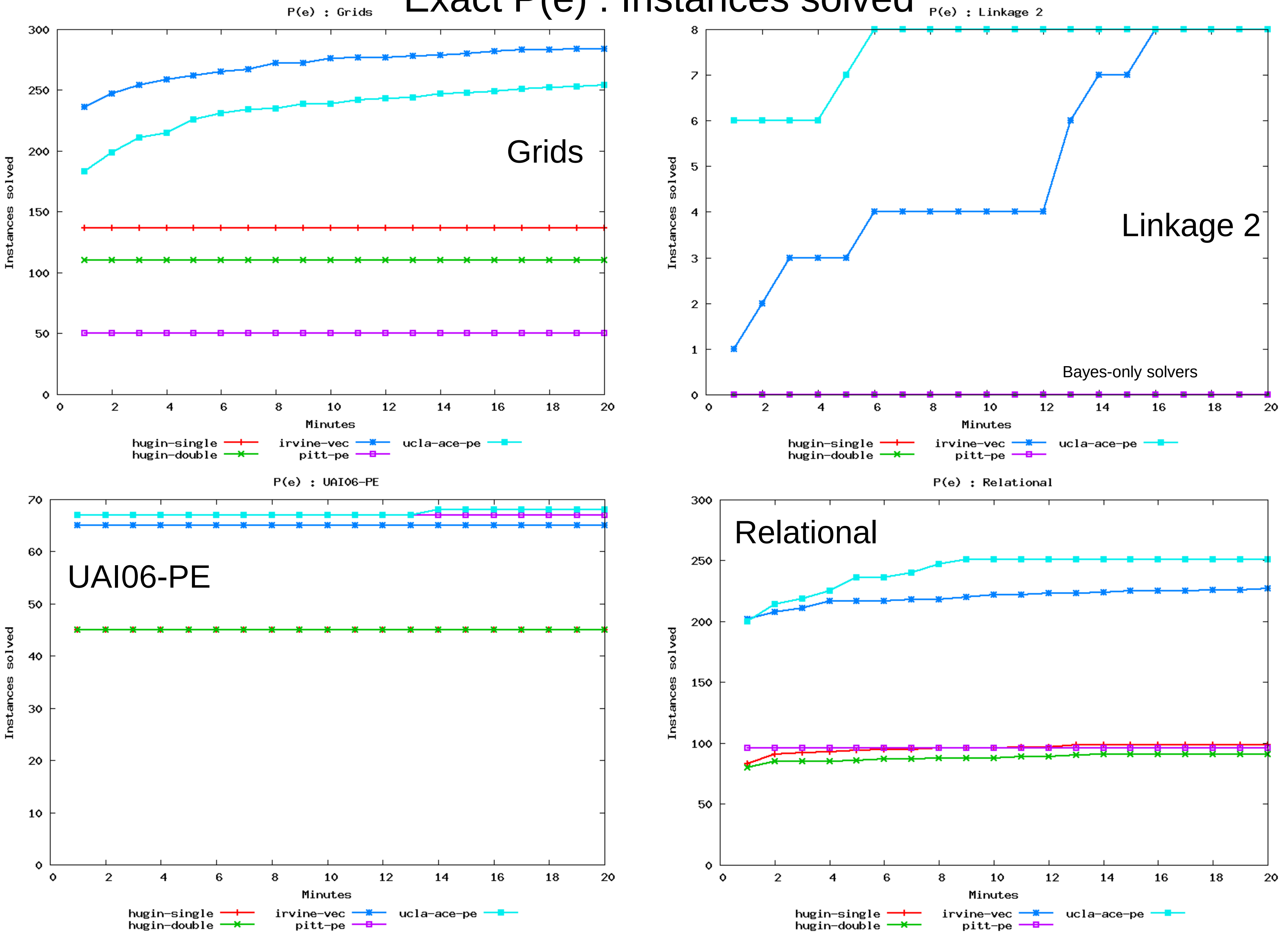
Exact PE results

- 5 Solvers:
- Bayes only:
  1. *hugin-single,*
  2. *hugin-double*
     - *Based on Hugin software (hugin.com)*
  3. *pitt-pe*
     - *Based on smile Genie library*
     - *University of Pittsburgh (Druzdzel et al.)*
     - *Relevance based reasoning*

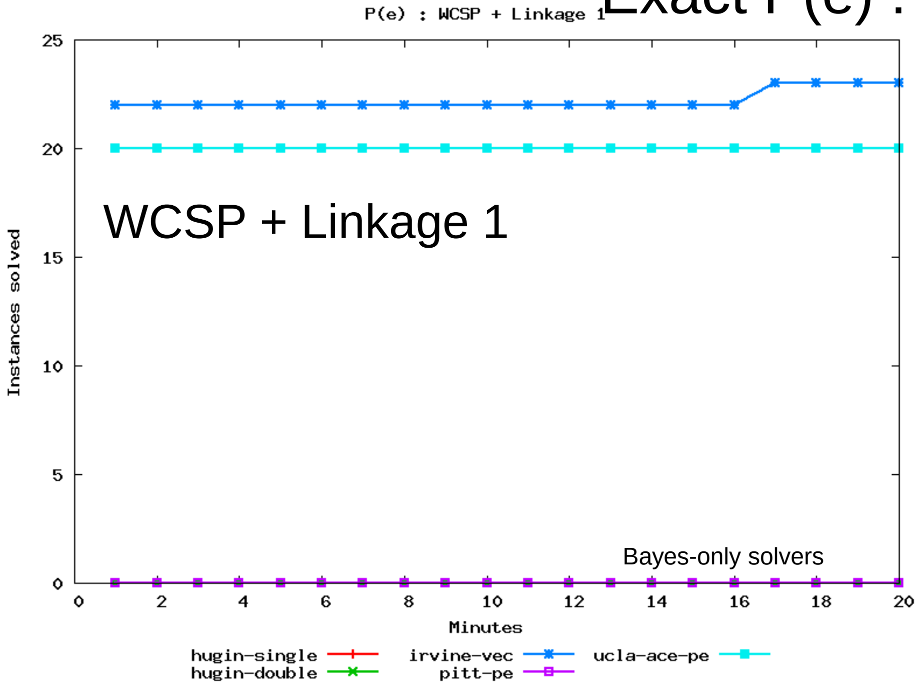
Exact PE results

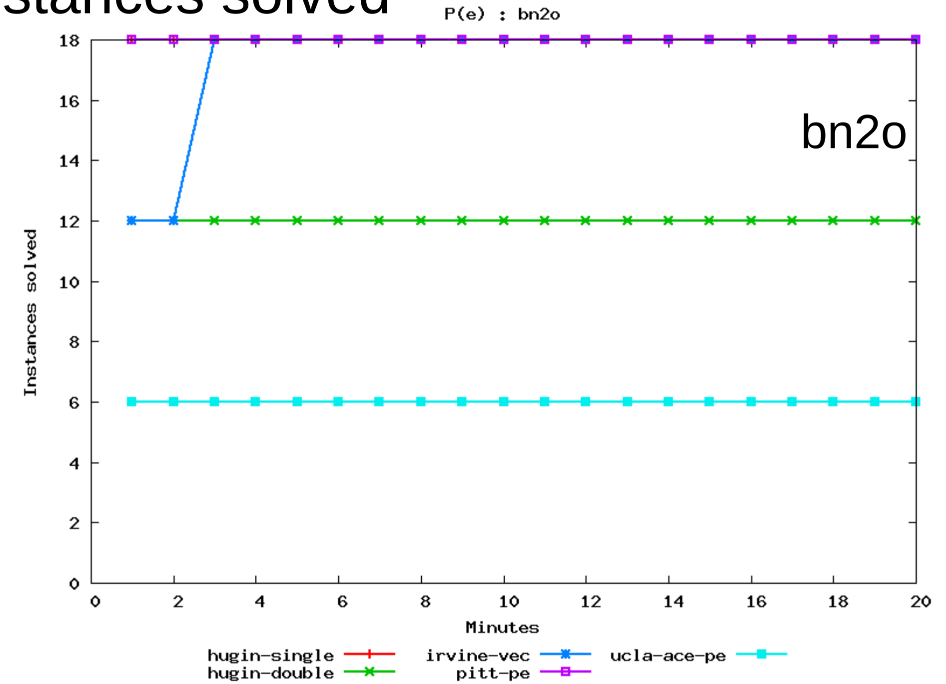
1. Irvine-VEC
   – Based on Bucket-elimination + conditioning approach
   – Minisat used internally to remove determinism from the network
2. UCLA-ACE
   – Based on C2D compiler
   – SAT based reasoning to remove and infer evidence

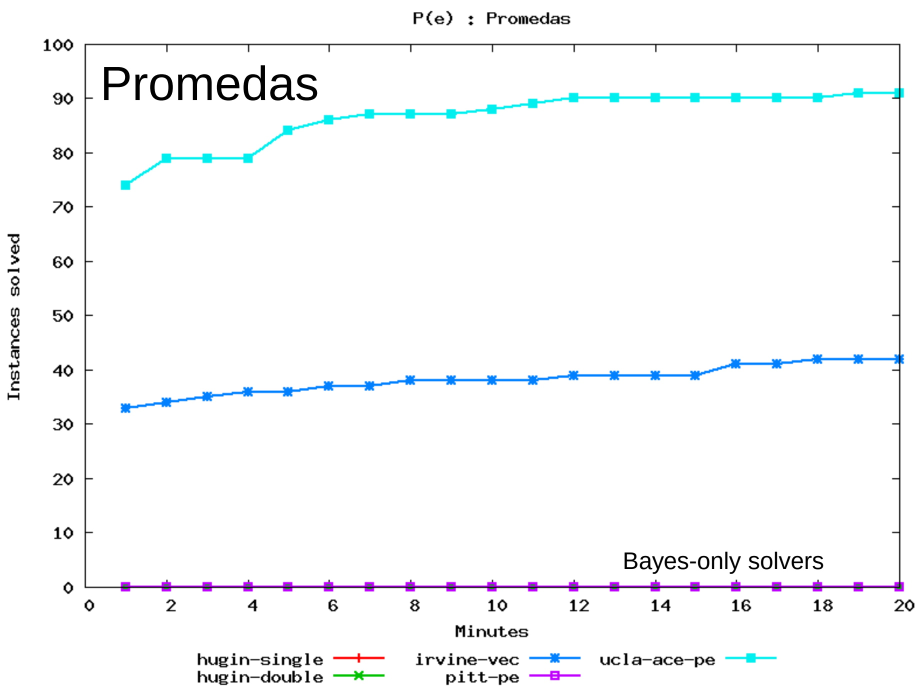
Exact P(e) : Instances solved

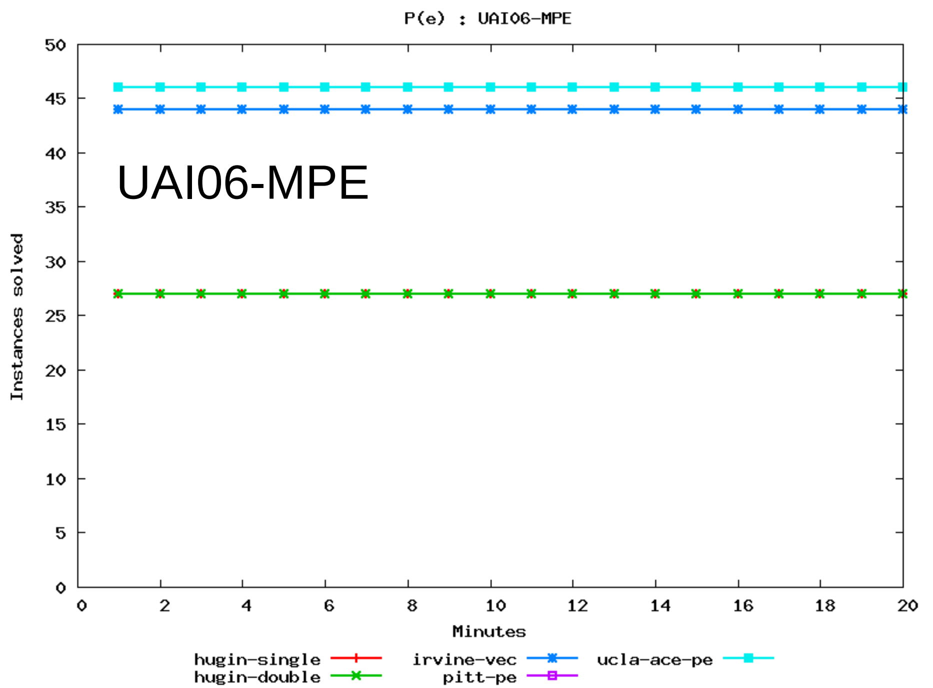


Exact P(e) : Instances solved







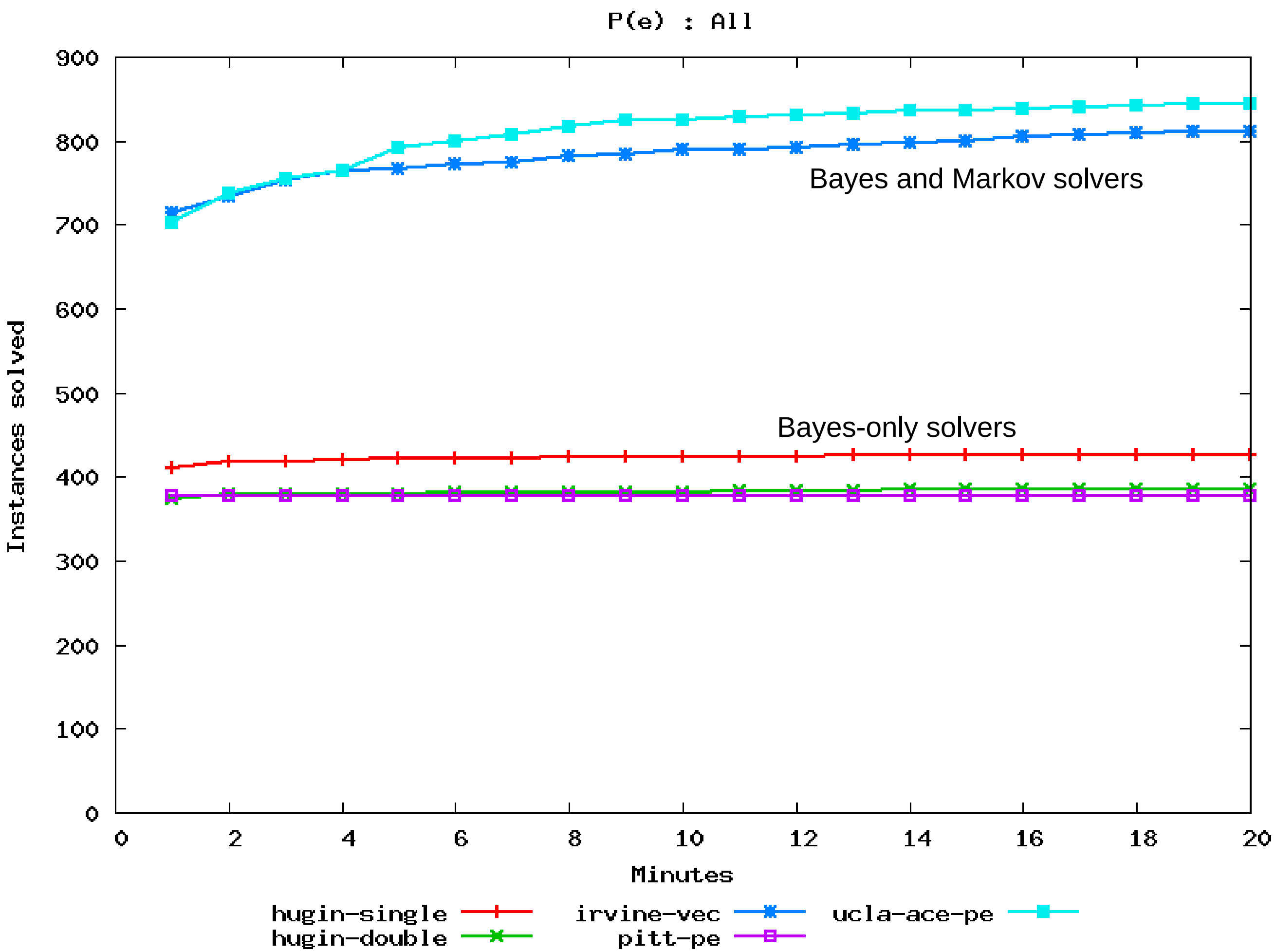


Summary

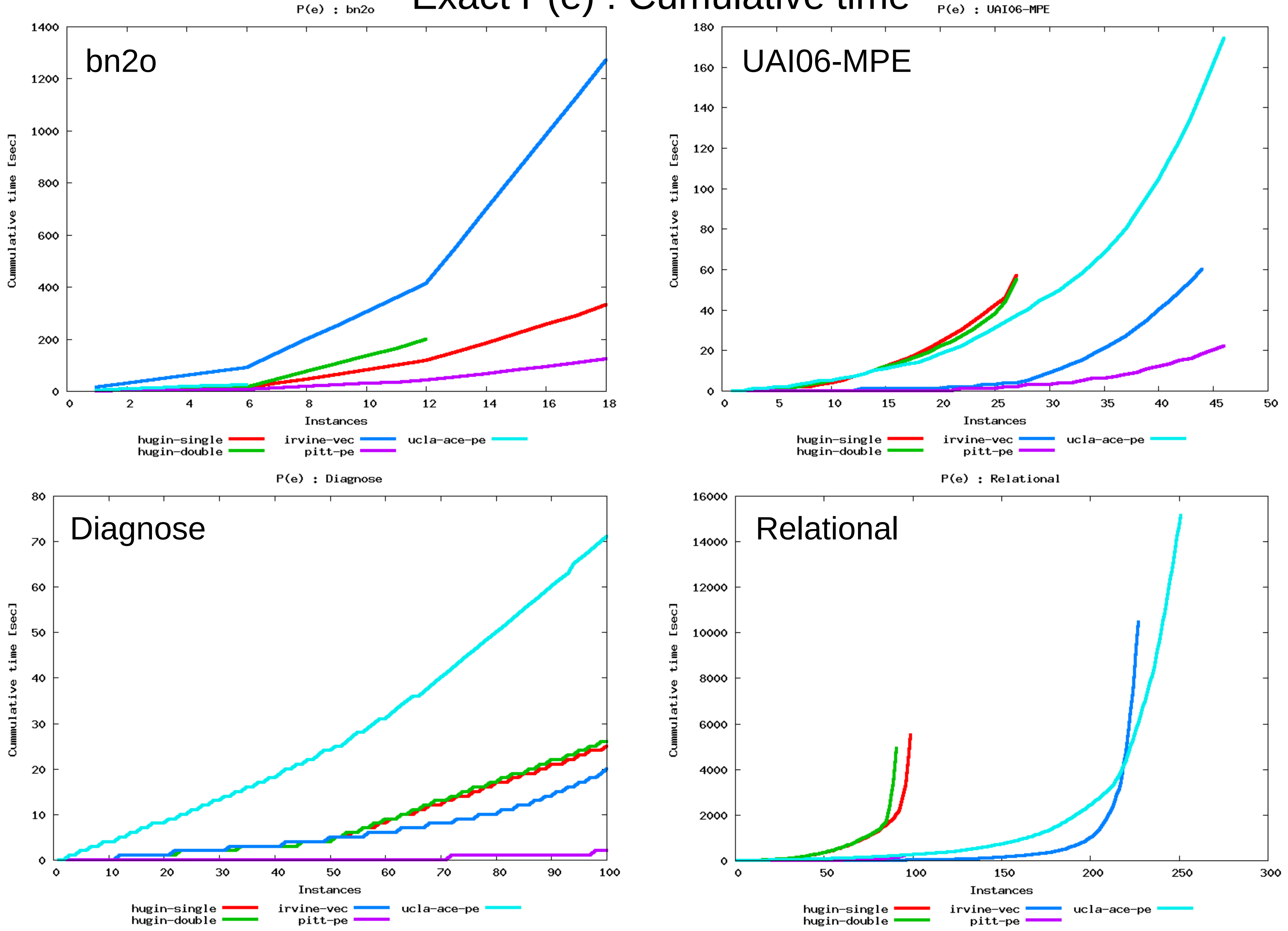
- *Bayes only:*
  - ucla-ace-pe solved more on 3 families
  - Irvine-vec solved more on 2 families
  - pitt-pe solved more on 1 family
- *Markov only:*
  - ucla-ace-pe solved more on 2 families
  - Irvine-vec solved more on 2 family
- *All networks:*
  - ucla-ace-pe solved more on 5 families
  - Irvine-vec solved more on 4 family

Exact PE : Instances solved overall

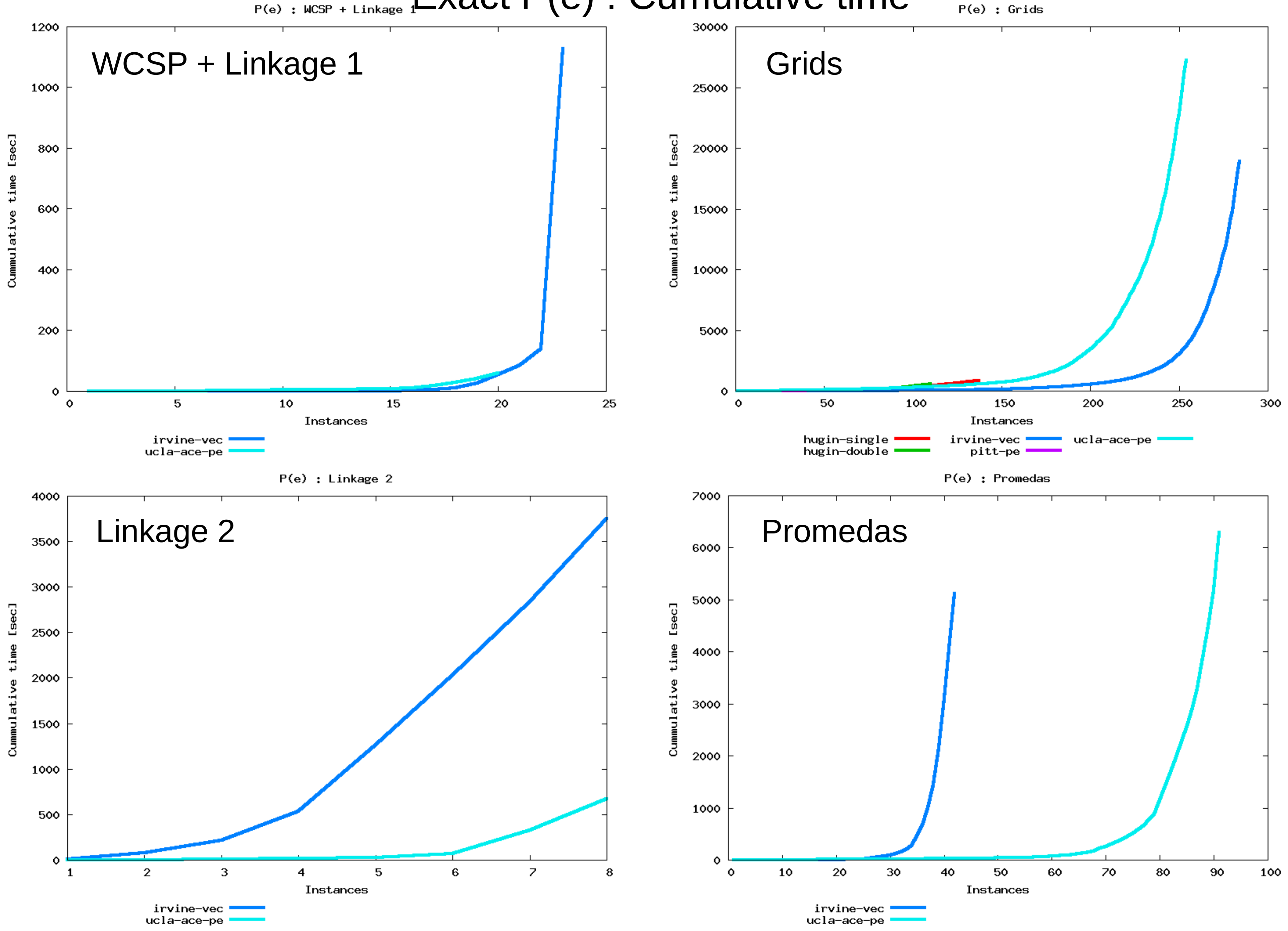
- Note: Not weighed by problem class size, biased to some classes/solvers.



Exact P(e) : Cumulative time



Exact P(e) : Cumulative time



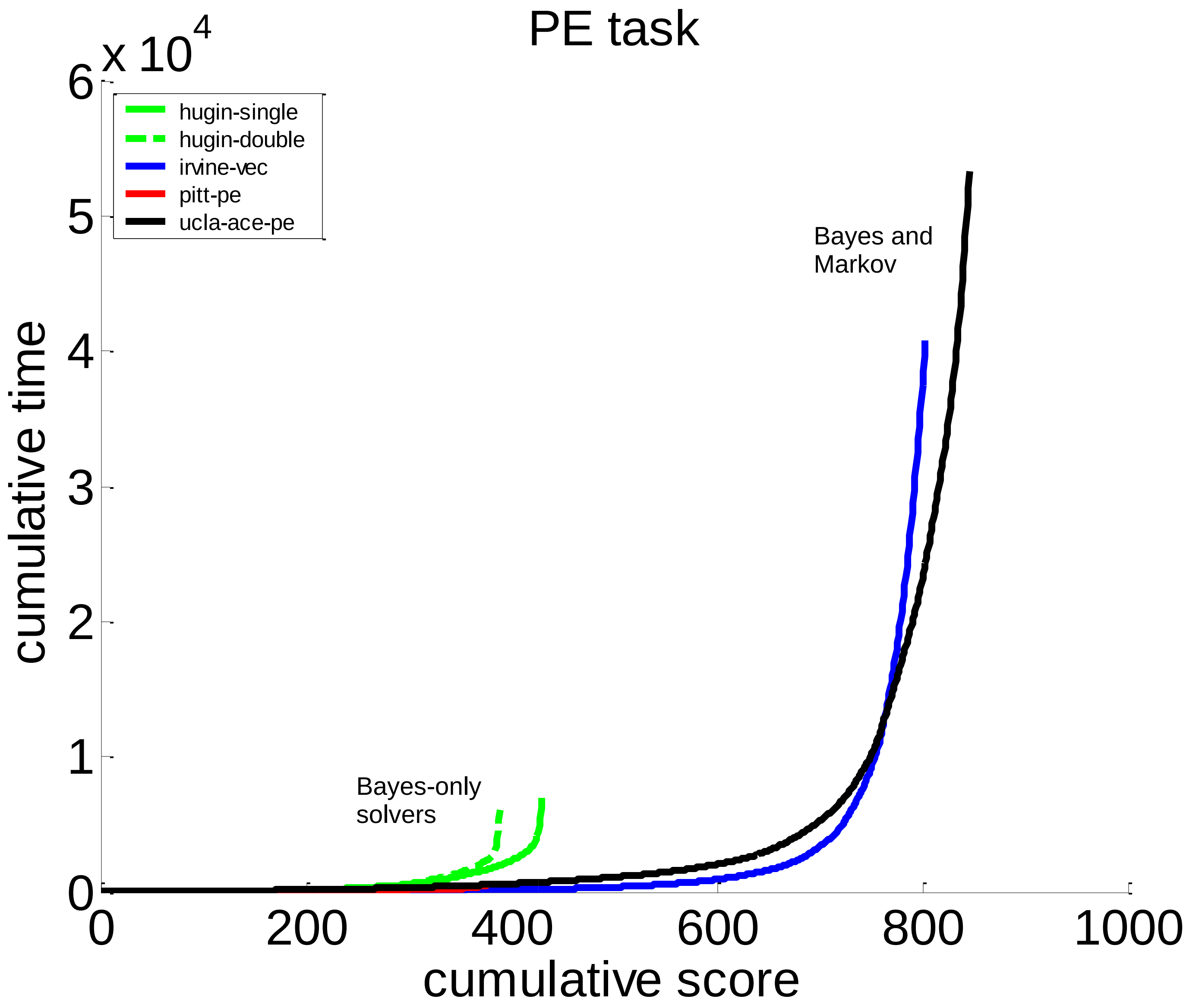
Summary

(some families show no dominance)

- *Bayes only:*
  - pitt-pe slightly better on 3 families
  - Irvine-vec dominates on 1 family
- *Markov only:*
  - ucla-ace-pe dominates on 2 families
  - Irvine-vec dominates on 1 family
- *All networks:*
  - ucla-ace-pe dominates on 2 families
  - Irvine-vec dominates on 2 families

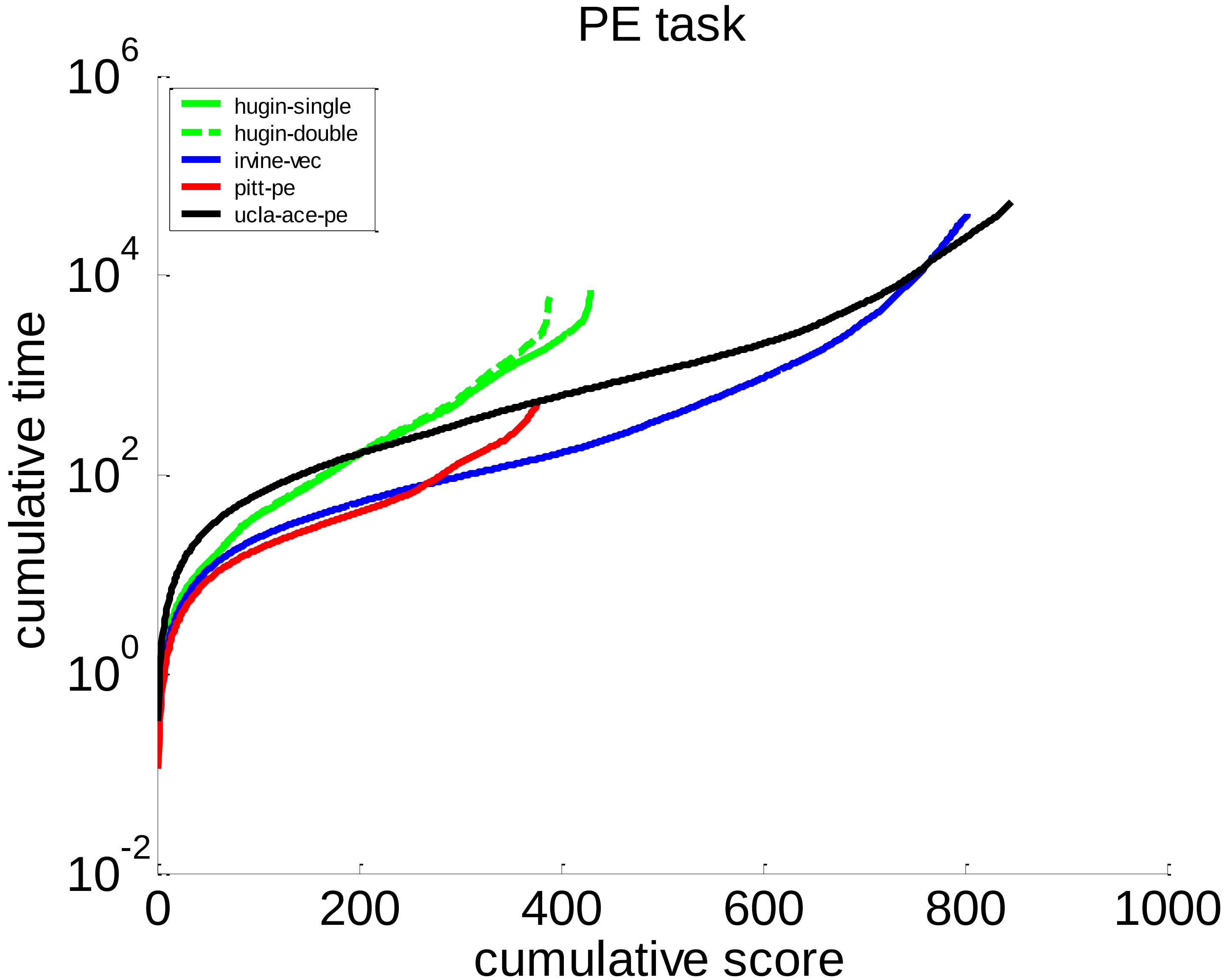
Cumulative time overall

Note: Not weighed by problem class size, biased to some classes/solvers



Cumulative time overall

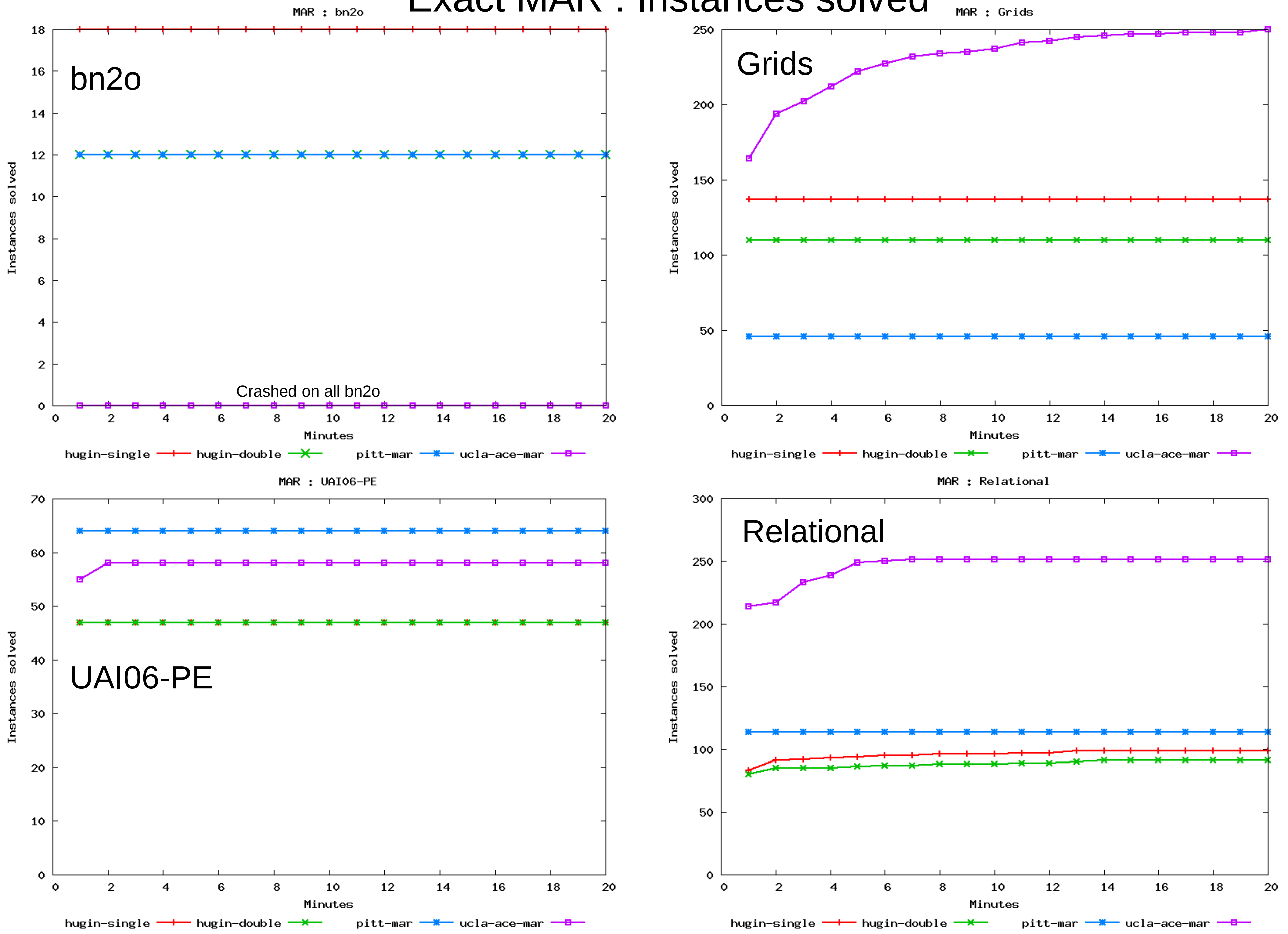
Note: Not weighed by problem class size, biased to some classes/solvers



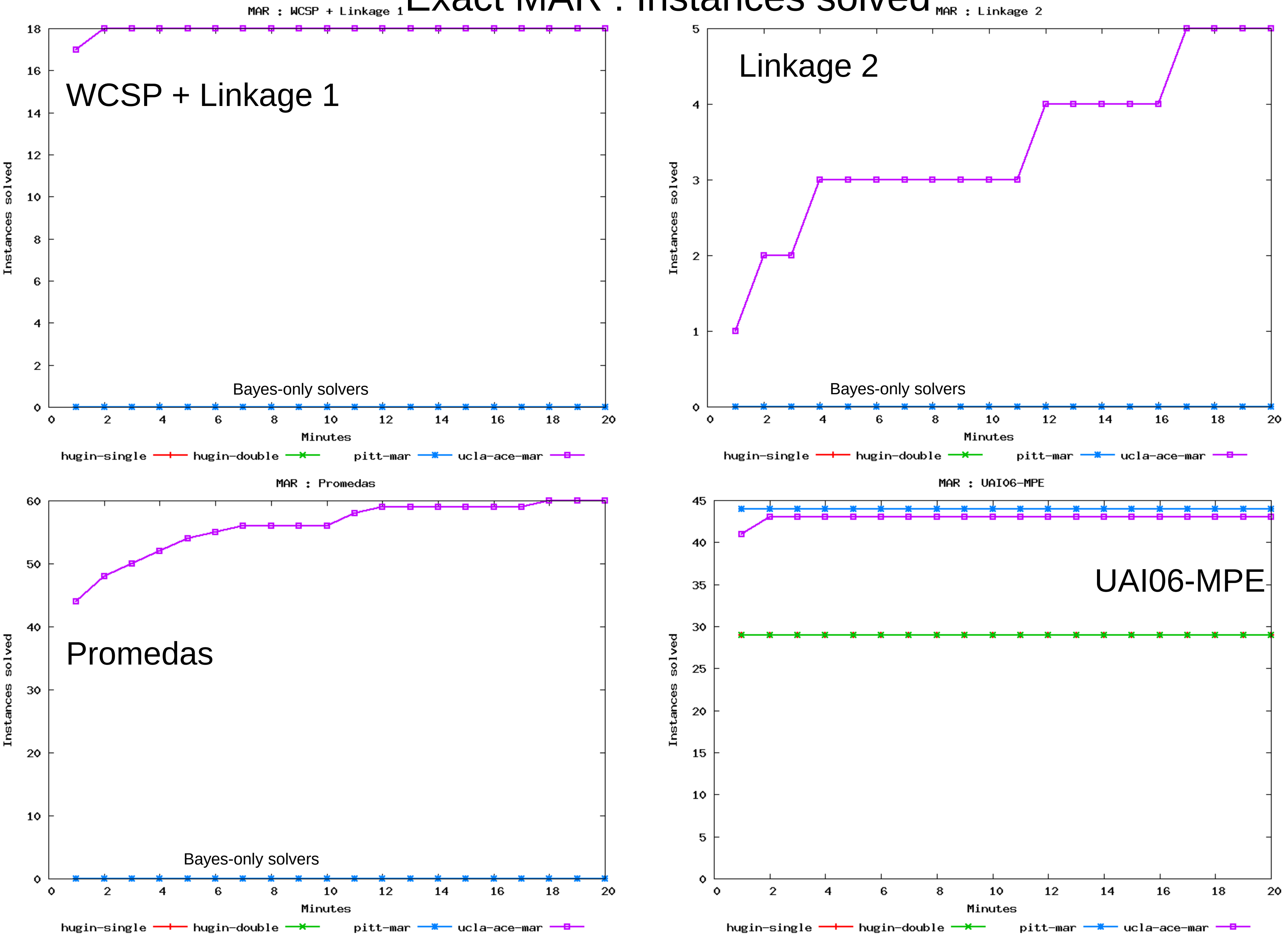
Exact MAR results

- 4 Solvers:
  - Bayes only:
    - *hugin-single, hugin-double, pitt-mar*
  - Bayes and Markov:
    - *ucla-ace-mar*

Exact MAR : Instances solved



Exact MAR : Instances solved

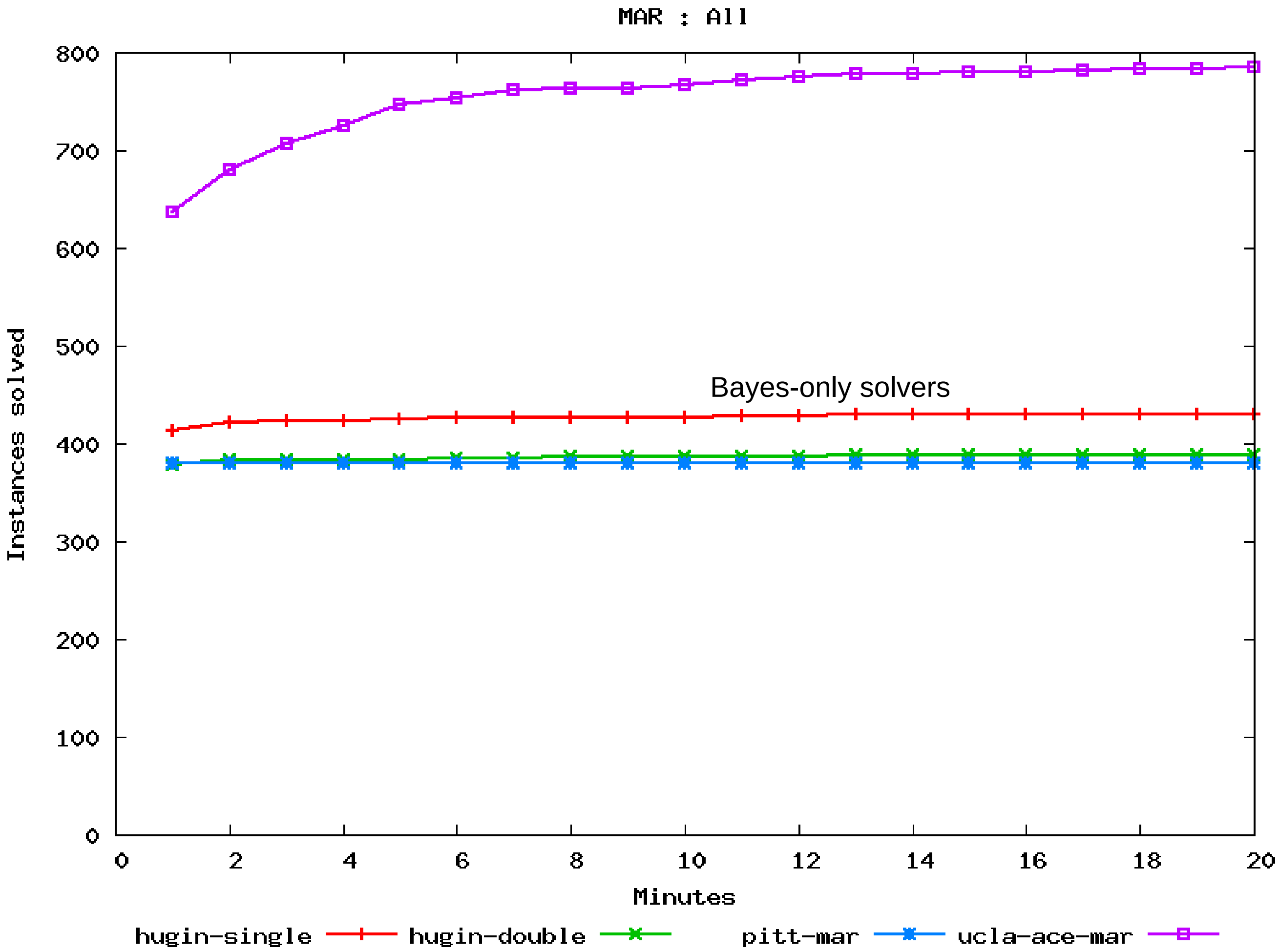


Summary

- *Bayes only:*
  - pitt-mar solved more on 3 families
  - ucla-ace-mar solved more on 2 families
  - hugin-single solved more on 1 family

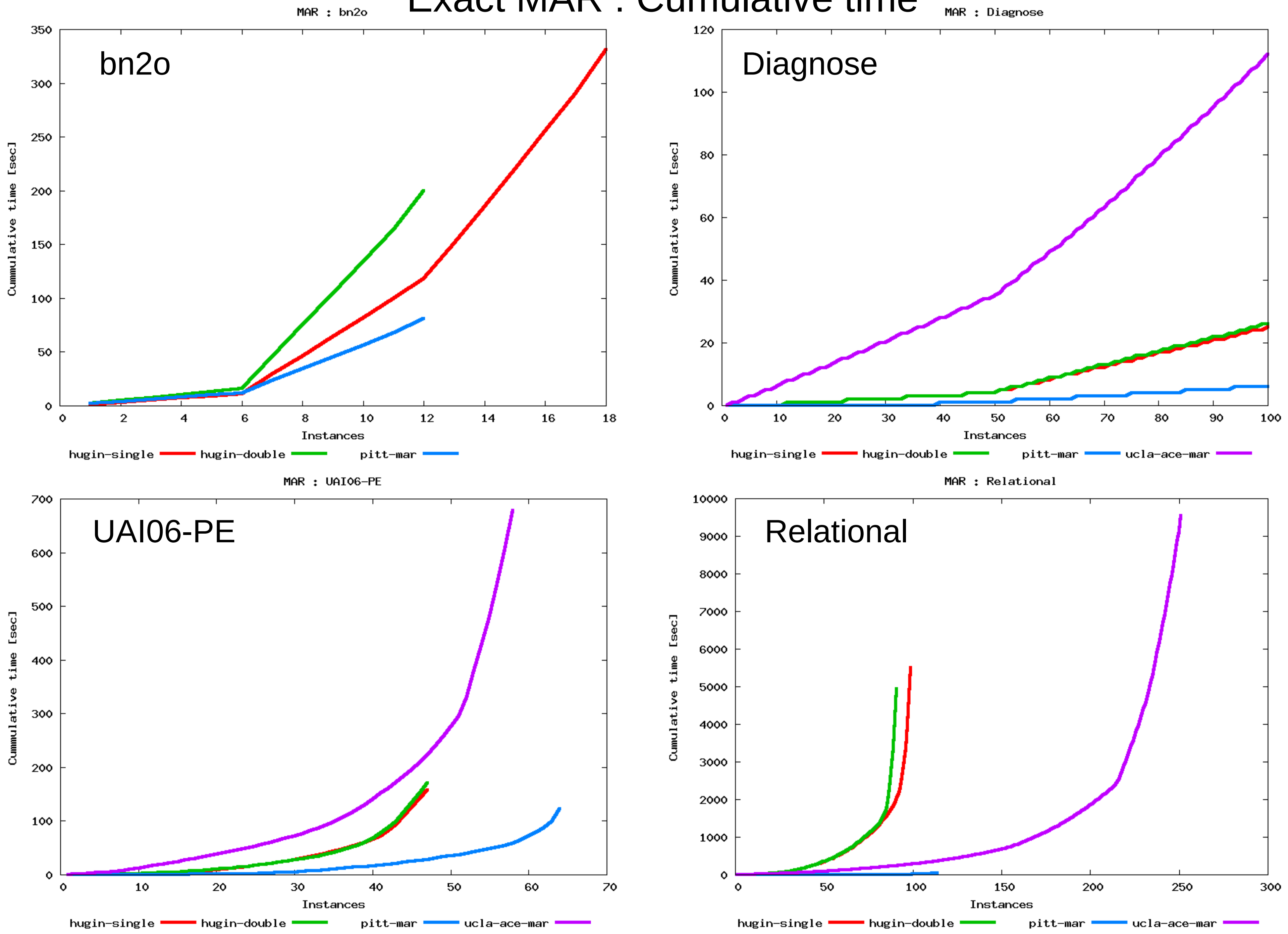
MAR : Instances solved overall

- Note: Not weighed by problem class size, biased to some classes/solvers.

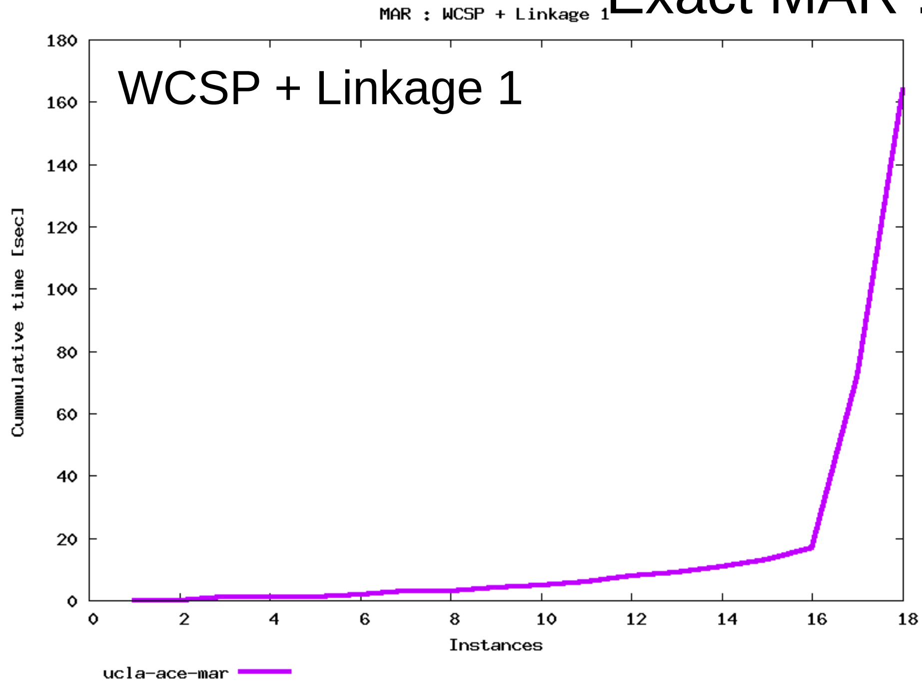


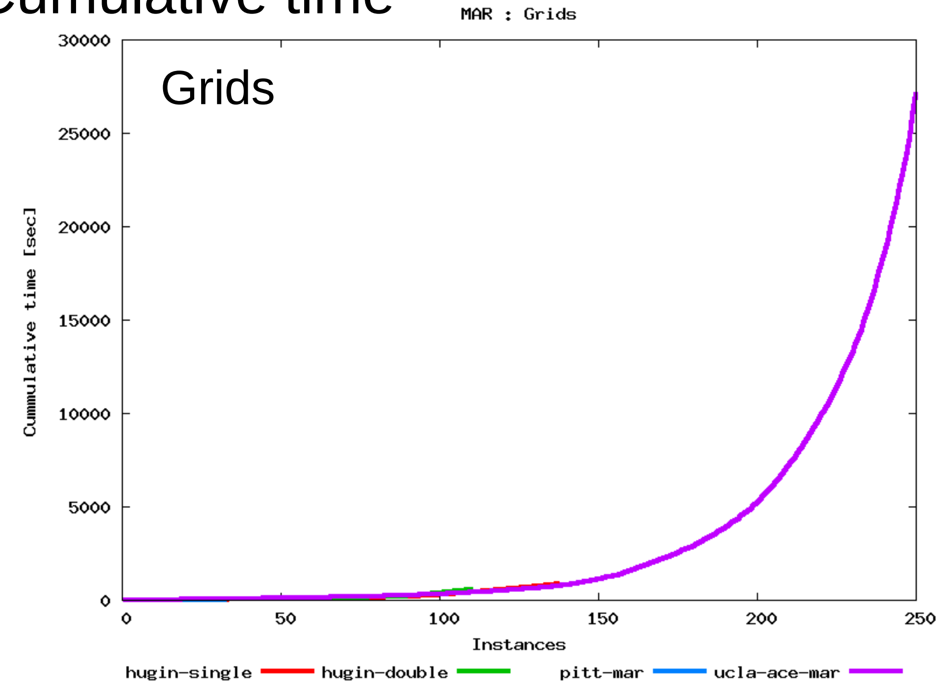
Grids..

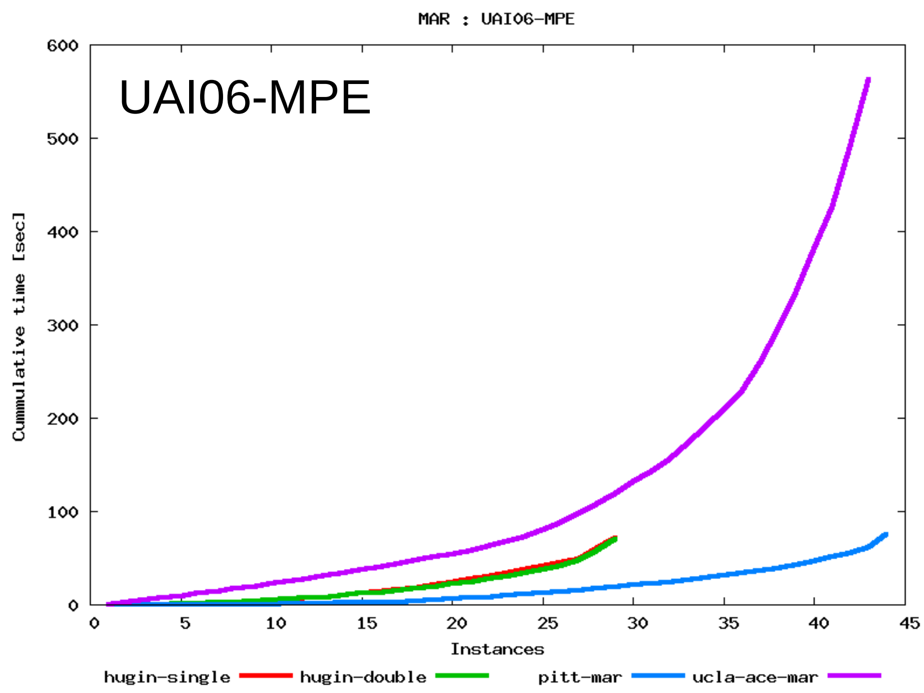
Exact MAR : Cumulative time

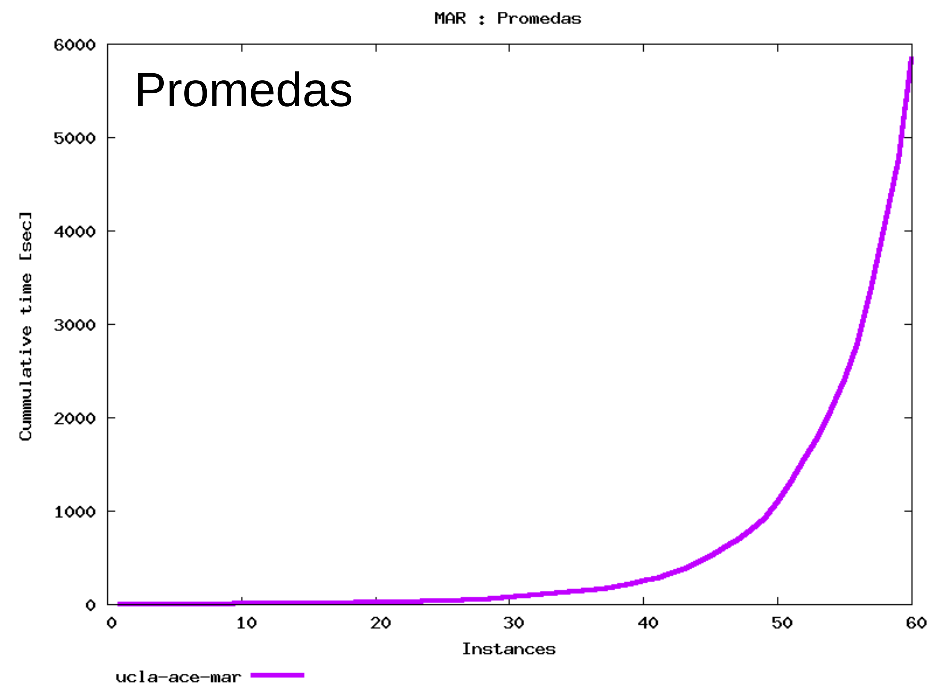


Exact MAR : Cumulative time







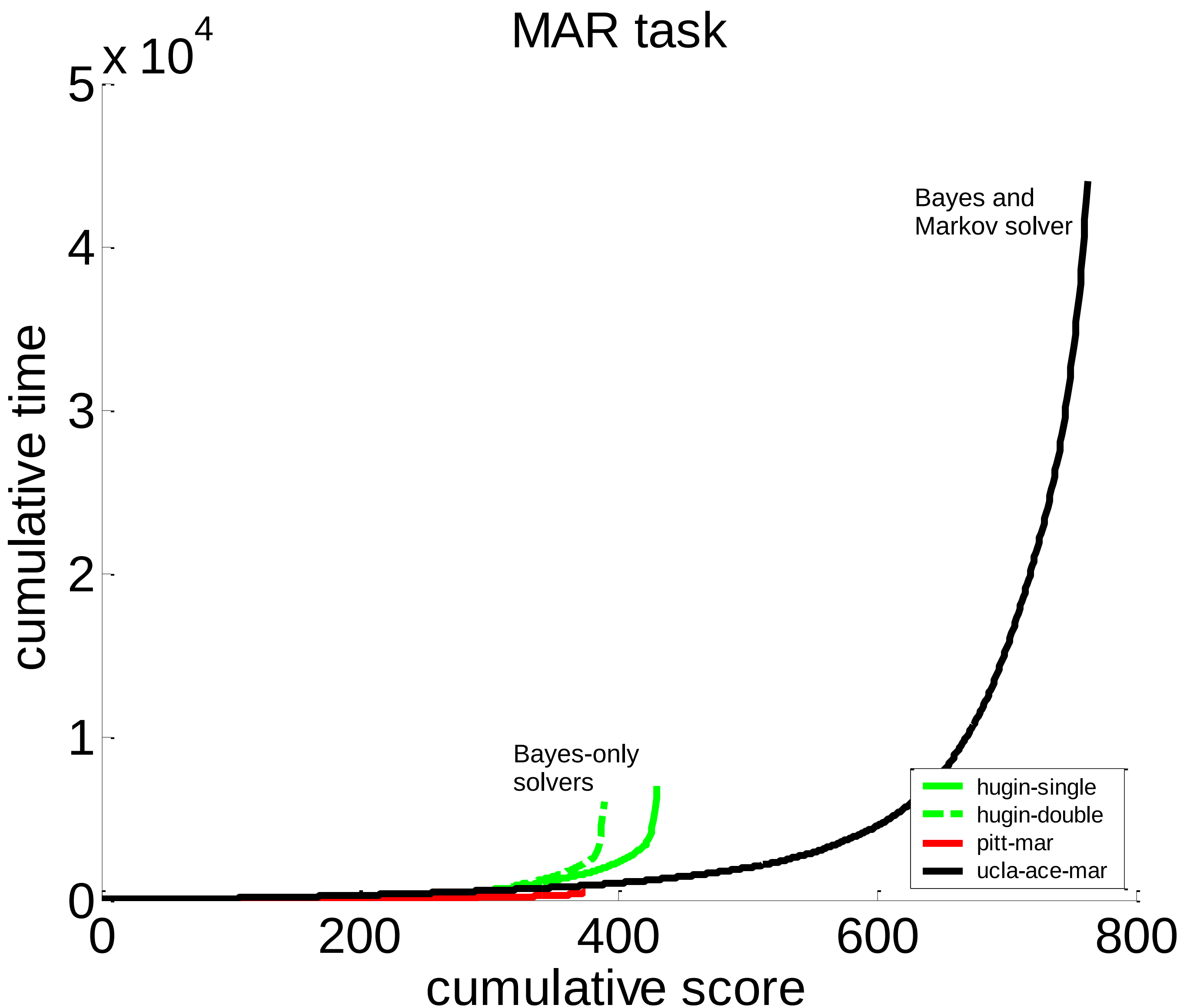


Summary

- *Bayes only:*
  - pitt-mar dominated on 3 families
  - ucla-ace-mar dominated on 1 family

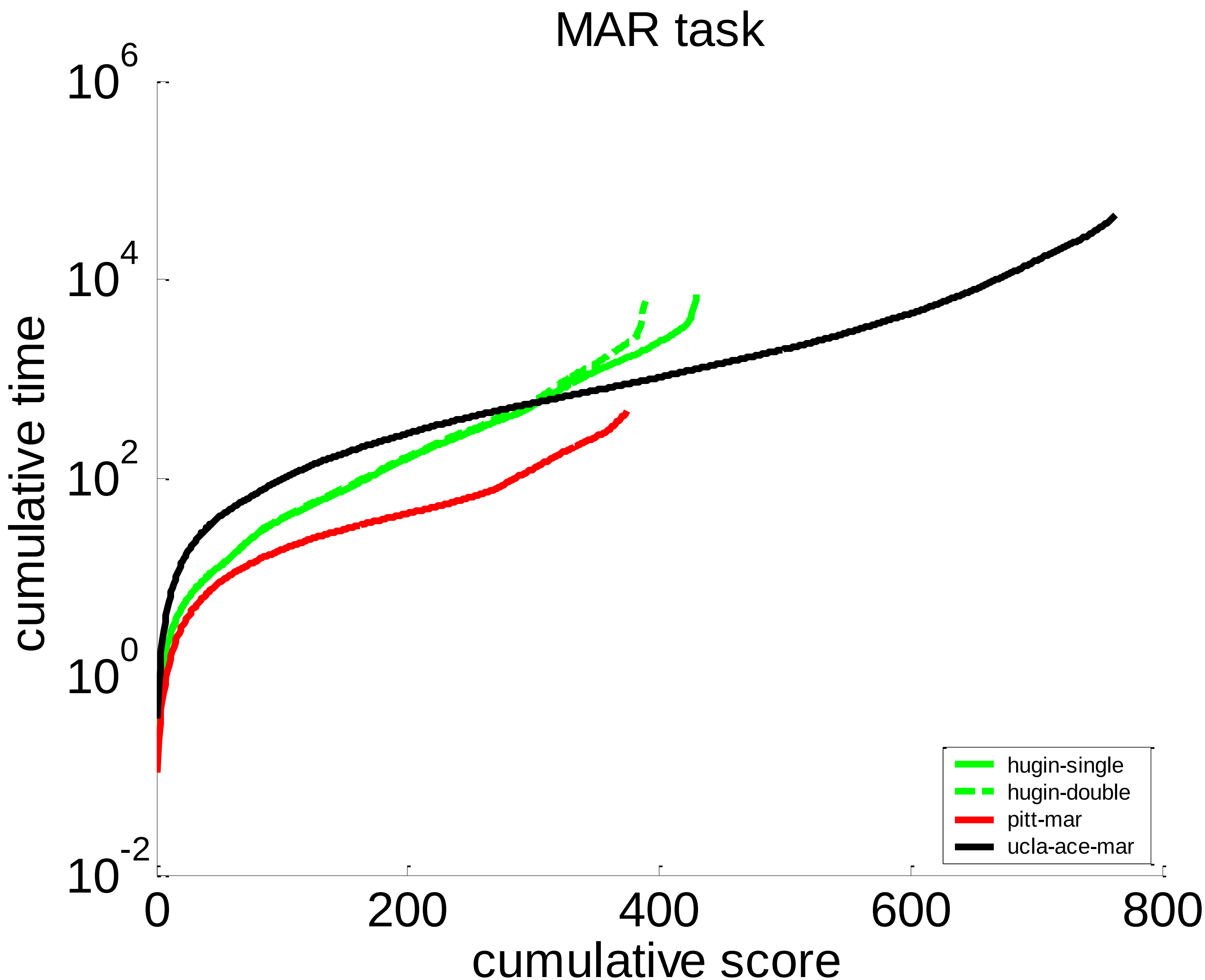
Exact MAR: Cumulative time overall

Note: Not weighed by problem class size, biased to some classes/solvers.



Exact MAR: Cumulative time overall

Note: Not weighed by problem class size, biased to some classes/solvers.



Evaluating Approximate PE Solvers

Benchmark Summary (appr/pe)

- 9 sets:

		Bys	Mkv	bin
– weighted-CSP	(97)		•	
– bn2o (diagnosis)	(18)	•		•
– hand-built	(0/100)	•		
– grids	(32/320)	•		•
– linkage	(22)		•	
– Promedas	(238)		•	•
– UAI-06 (MPE)	(57)	•		
– UAI-06 (PE)	(78)	•		
– relational	(35/251)	•		•
– **TOTAL**	**(577)**	**(220)**	**(357)**	**(323)**

Approximate PE

- 4 Solvers:
  - *All problems:*
    - *irvine-samplesearch, irvine-vec, ucla-edbp-pe*
  - Binary-variable problems only:
    - *upf-pe*

Approximate PE

- 4 Solvers:
  - *All problems:*
    - *irvine-samplesearch*
      - *Special importance sampling technique for problems with zero probabilities*
    - *irvine-vec,*
      - *variable elimination+conditioning based solver*
    - *ucla-edbp-pe*
      - *Generalized Belief propagation based solver*
      - *Choi and Darwiche 2007*
  - *Binary problems only*
    - *upf-pe*
      - *Belief propagation style solver*
      - *Truncated loop series (Gomez et al., 2007)*

Approximate PE Score

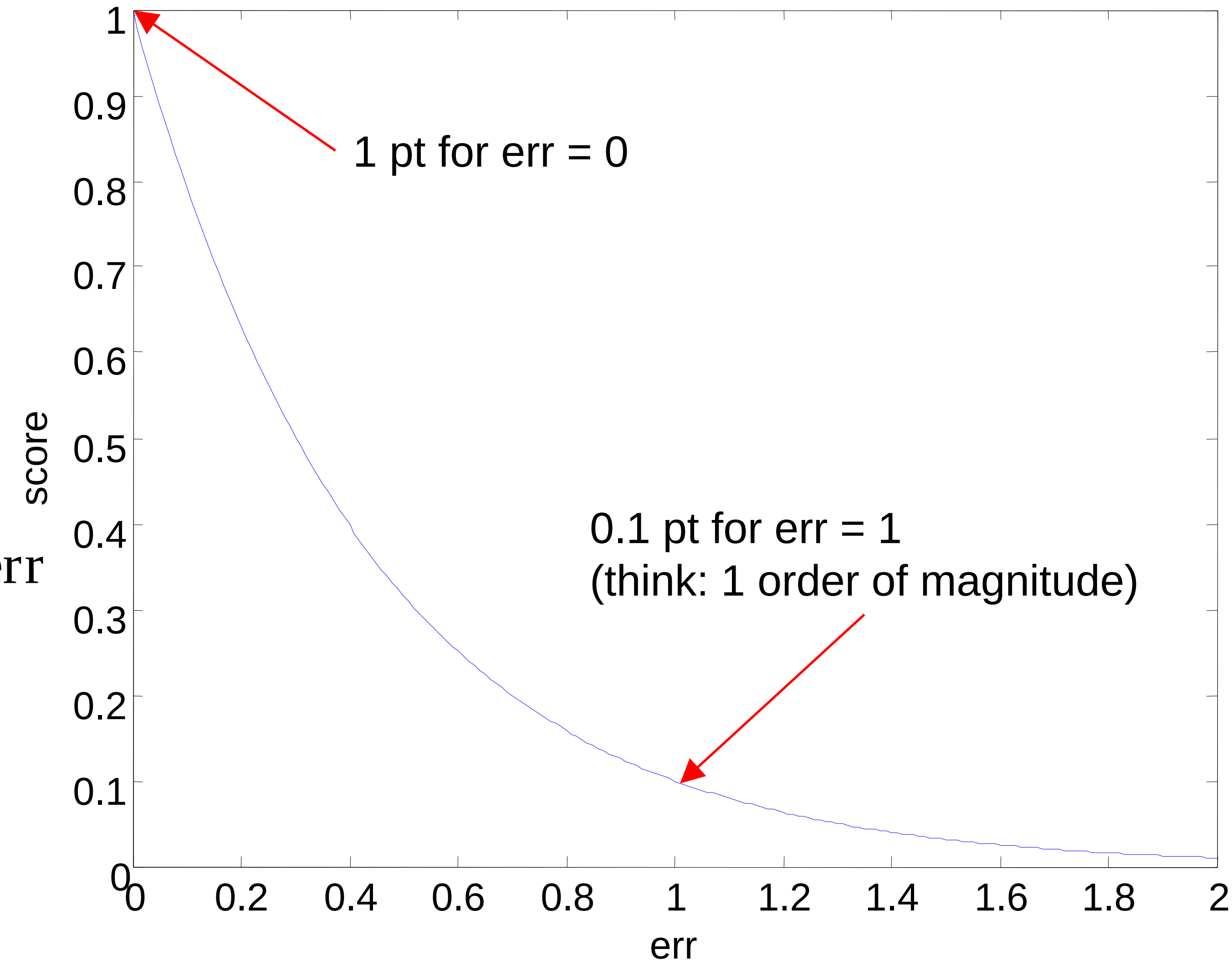
- No solution
  - score = 0
- Otherwise, given the relative error:

$$\mathrm{err} = \left| \frac{\log Z_{\mathrm{sol}} - \log Z_{\mathrm{exact}}}{\log Z_{\mathrm{exact}}} \right|$$

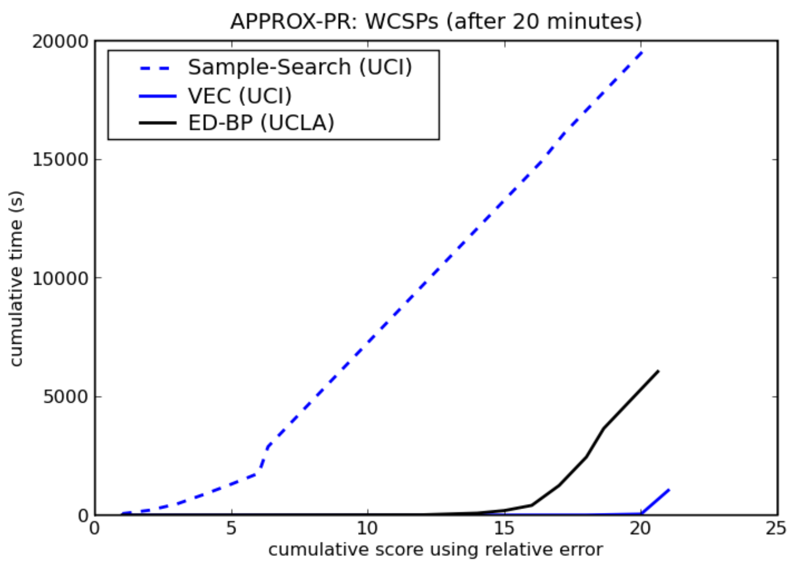
- Compute the score:

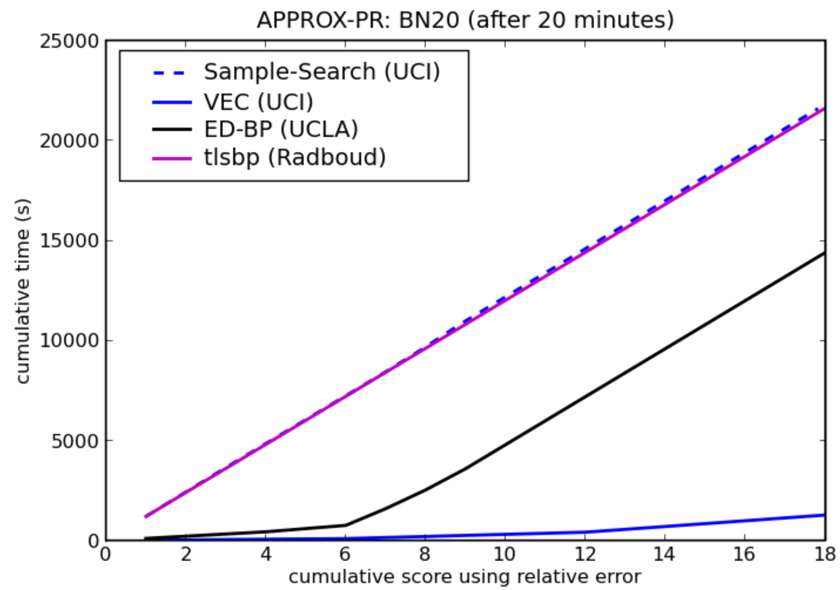
$$\mathrm{score} = 10^{-\mathrm{err}}$$

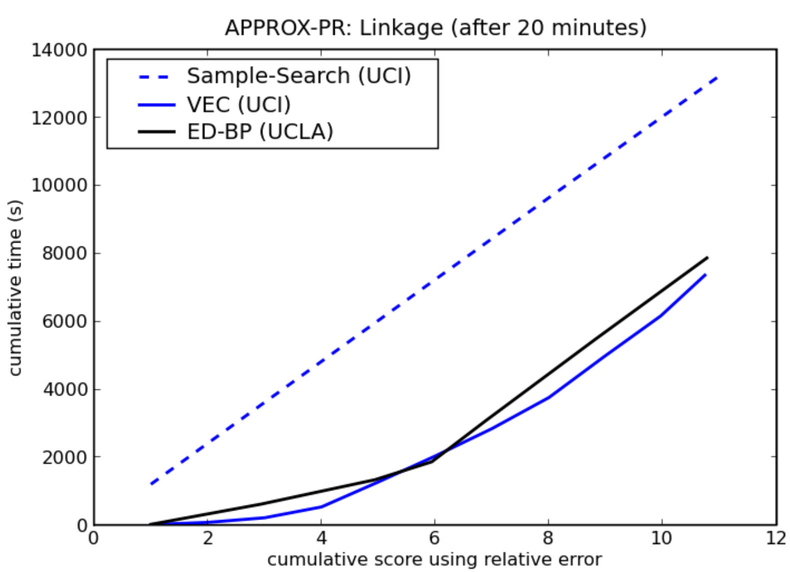
$$\text{score} = 10^{-\text{err}}$$

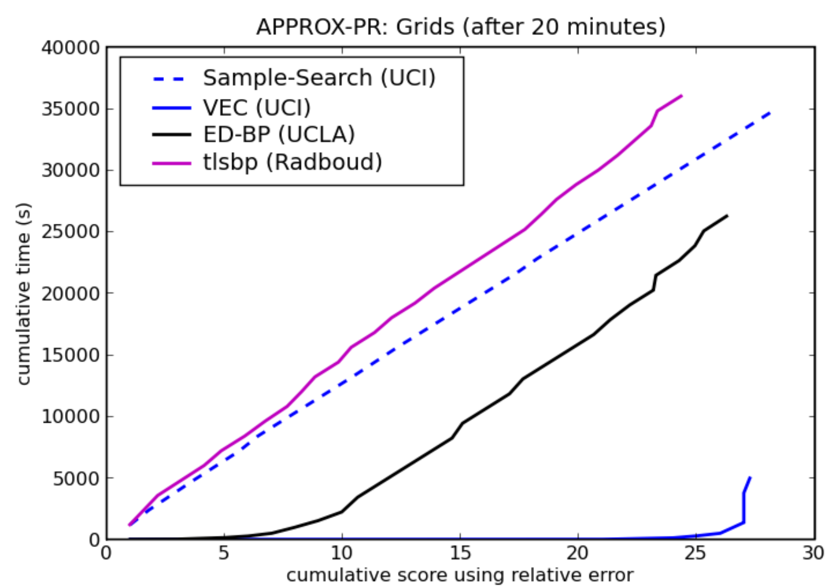


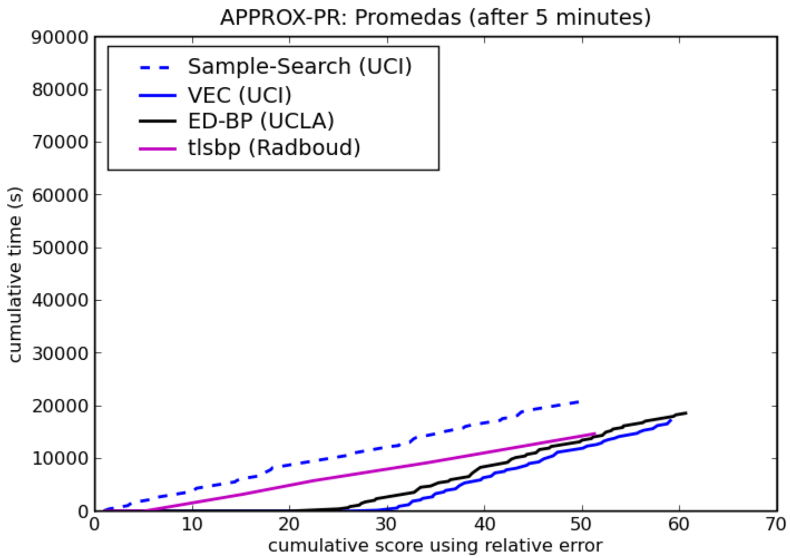
$$\text{err} = \left| \frac{\log_{10} Z_{\text{sol}} - \log_{10} Z_{\text{exact}}}{\log_{10} Z_{\text{exact}}} \right|$$

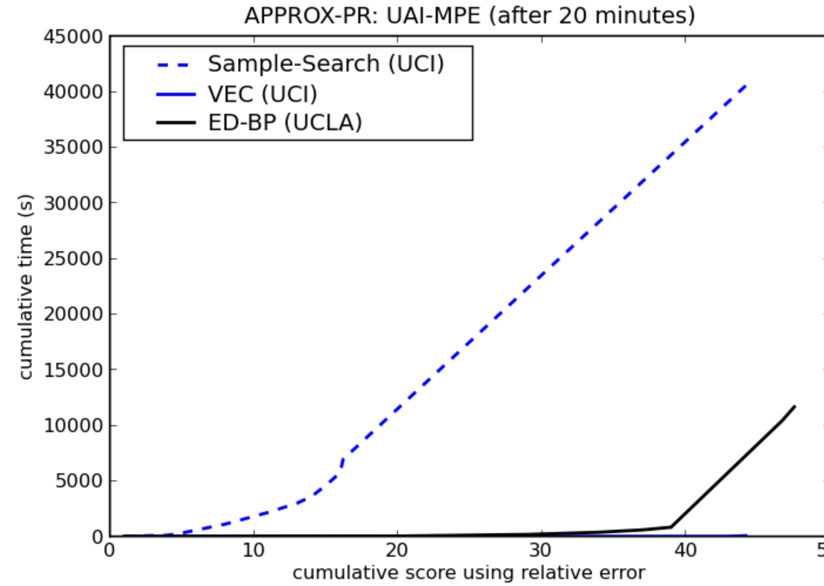

APPROX-PR: WCSPs (after 20 minutes)
Sample-Search (UCI)
VEC (UCI)
ED-BP (UCLA)
cumulative time (s)
cumulative score using relative error
0
5000
10000
15000
20000
5
10
15
20
25

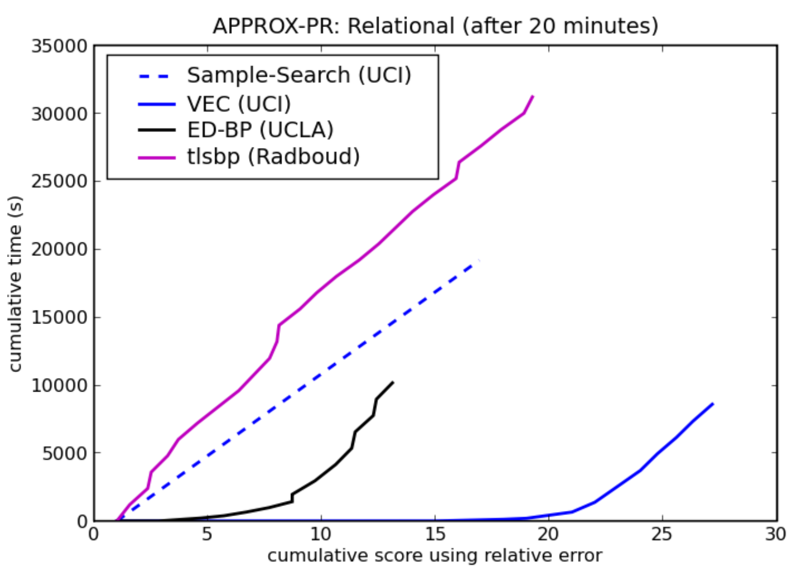

APPROX-PR: BN20 (after 20 minutes)
Sample-Search (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using relative error
0
5000
10000
15000
20000
25000
2
4
6
8
10
12
14
16
18

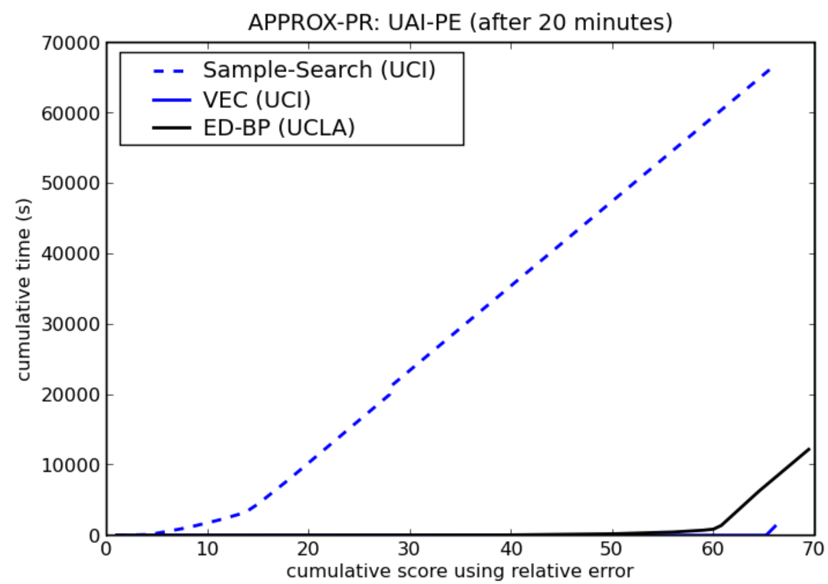

APPROX-PR: Linkage (after 20 minutes)
Sample-Search (UCI)
VEC (UCI)
ED-BP (UCLA)
cumulative time (s)
cumulative score using relative error
0
2000
4000
6000
8000
10000
12000
14000
2
4
6
8
10
12

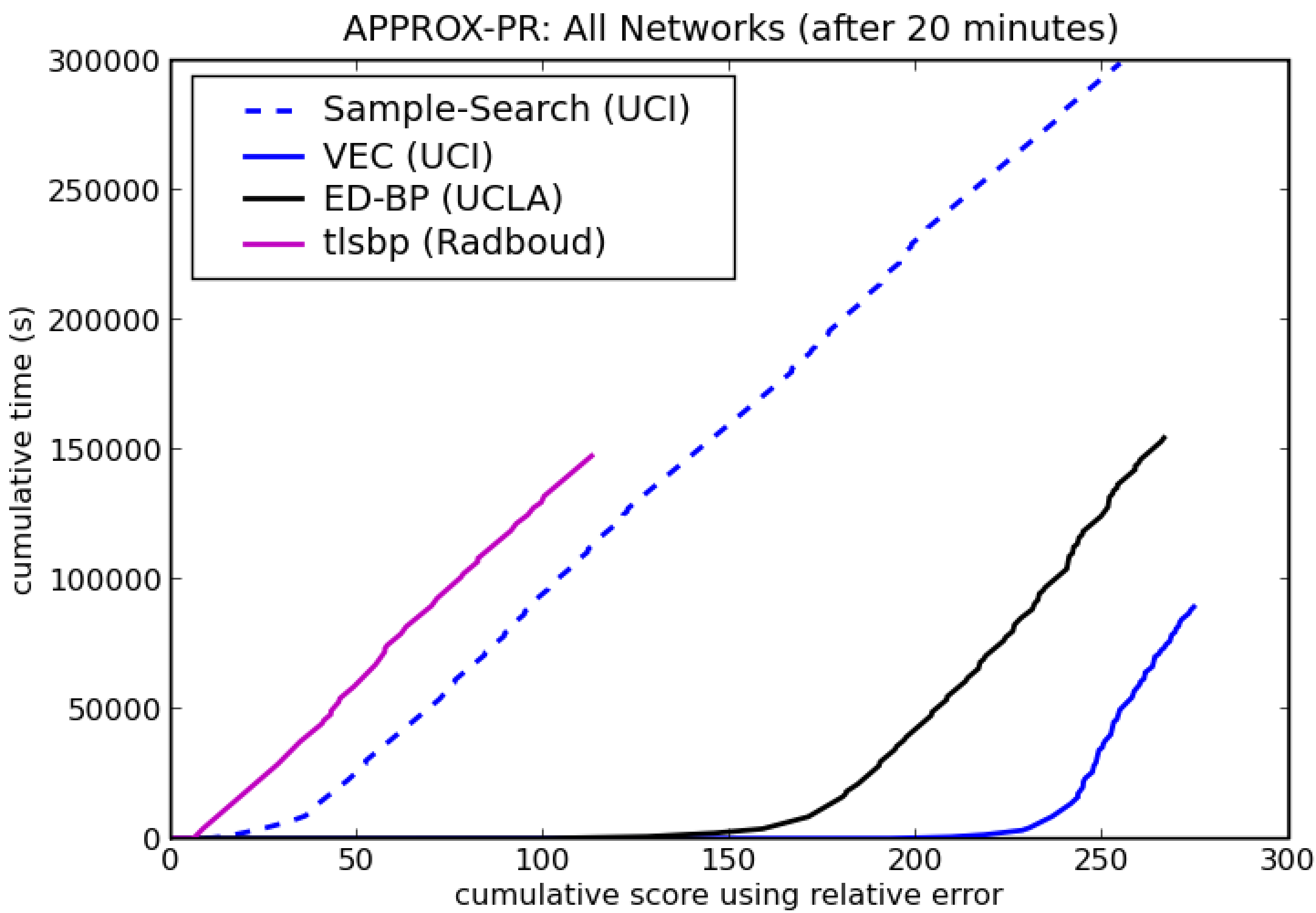

APPROX-PR: Grids (after 20 minutes)
Sample-Search (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using relative error
0
5000
10000
15000
20000
25000
30000
35000
40000
5
10
15
20
25
30


APPROX-PR: Promedas (after 5 minutes)
Sample-Search (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using relative error

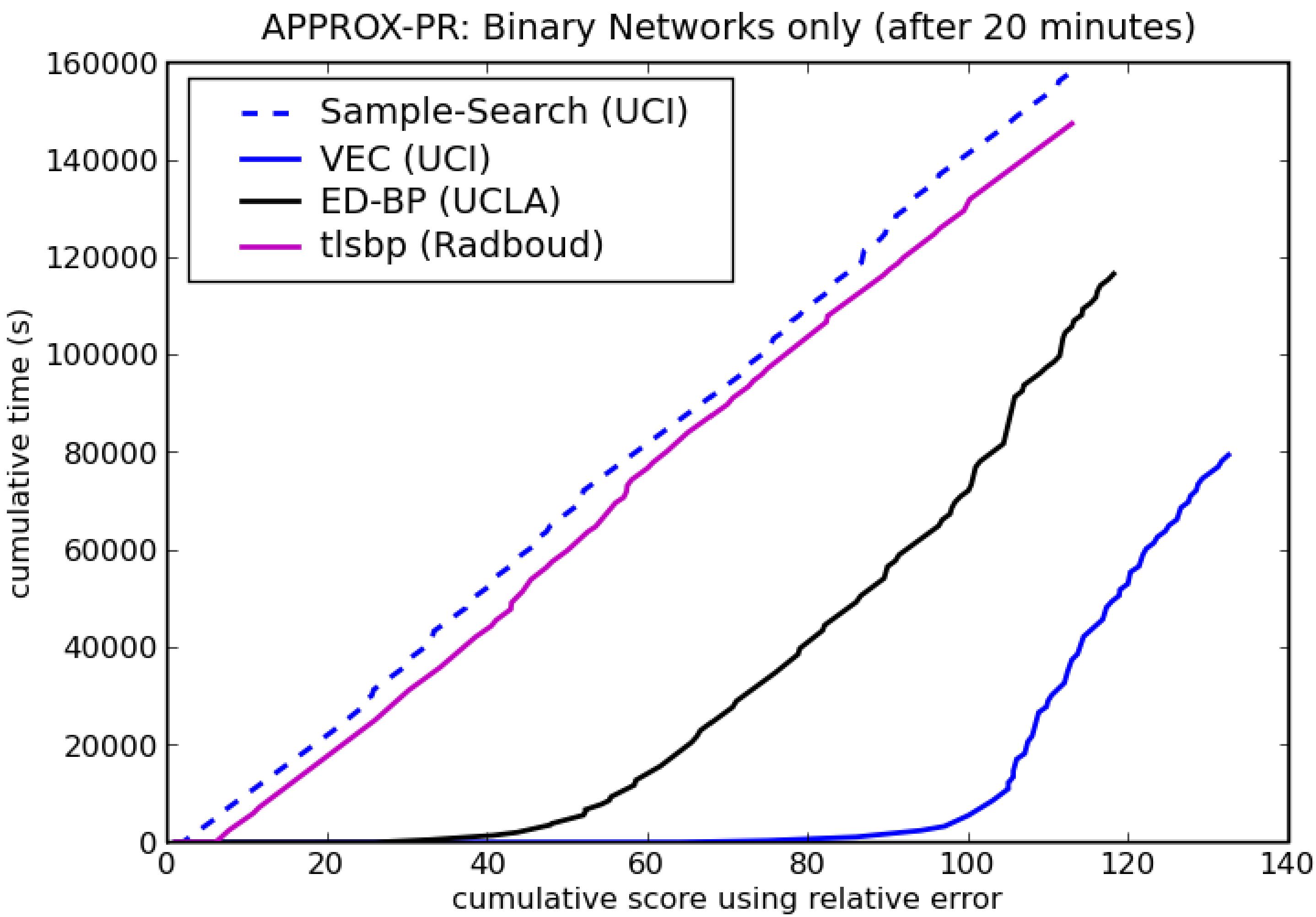

APPROX-PR: UAI-MPE (after 20 minutes)
Sample-Search (UCI)
VEC (UCI)
ED-BP (UCLA)
cumulative time (s)
cumulative score using relative error


APPROX-PR: Relational (after 20 minutes)
Sample-Search (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using relative error


APPROX-PR: UAI-PE (after 20 minutes)
Sample-Search (UCI)
VEC (UCI)
ED-BP (UCLA)
cumulative time (s)
cumulative score using relative error



APPROX-PR: All Networks (after 20 minutes)
Sample-Search (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using relative error
0
50000
100000
150000
200000
250000
300000
0
50
100
150
200
250
300



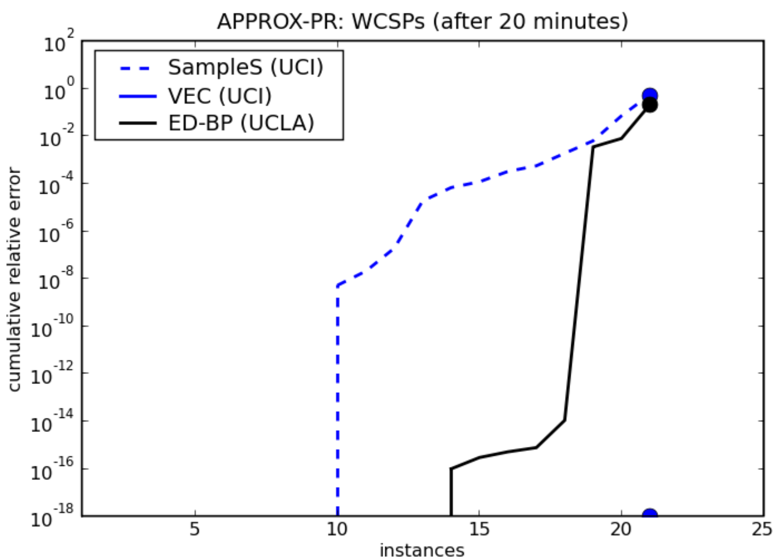
APPROX-PR: Binary Networks only (after 20 minutes)
Sample-Search (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using relative error
0
20000
40000
60000
80000
100000
120000
140000
160000
0
20
40
60
80
100
120
140

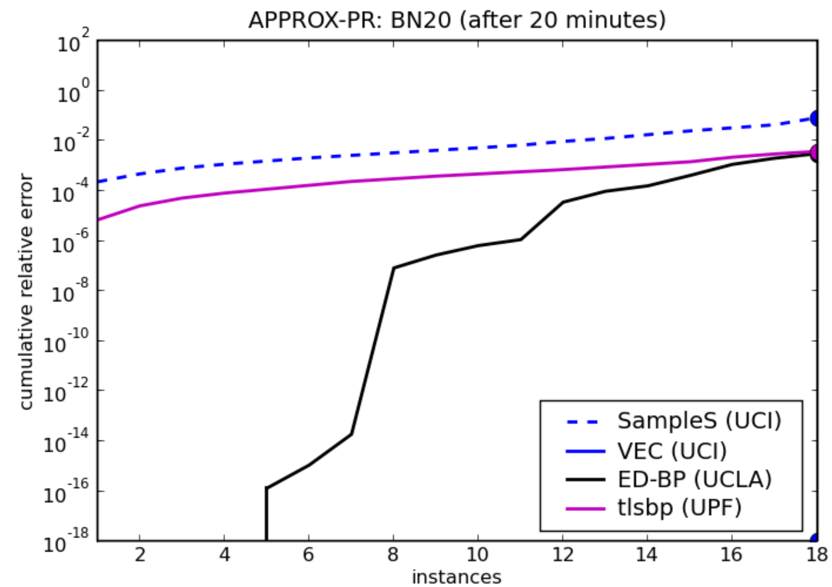
Approximate PE Error plots

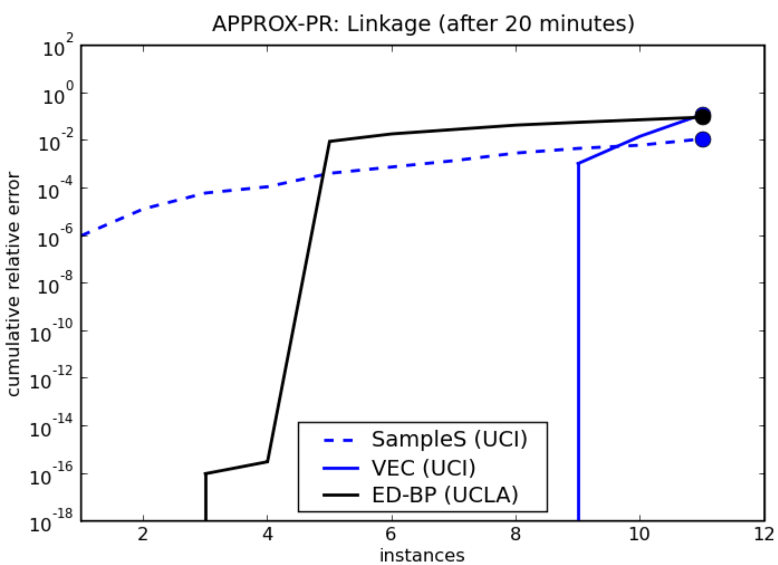
- Relative error:

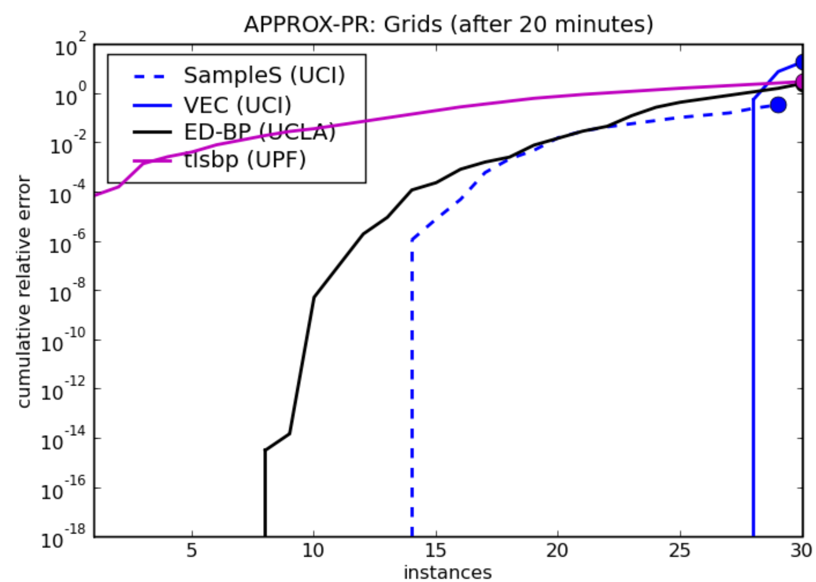
$$\mathrm{err} = \left| \frac{\log Z_{\mathrm{sol}} - \log Z_{\mathrm{exact}}}{\log Z_{\mathrm{exact}}} \right|$$

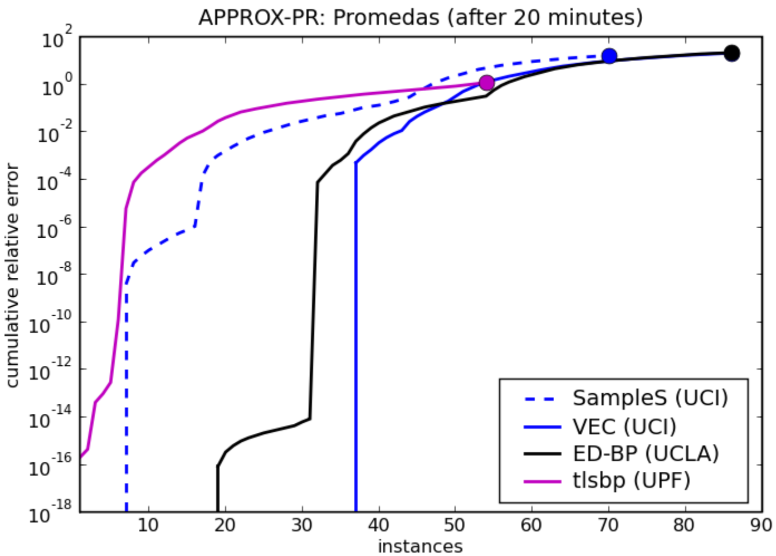
- Sort instances based on err
- Plot cumu error vs instances

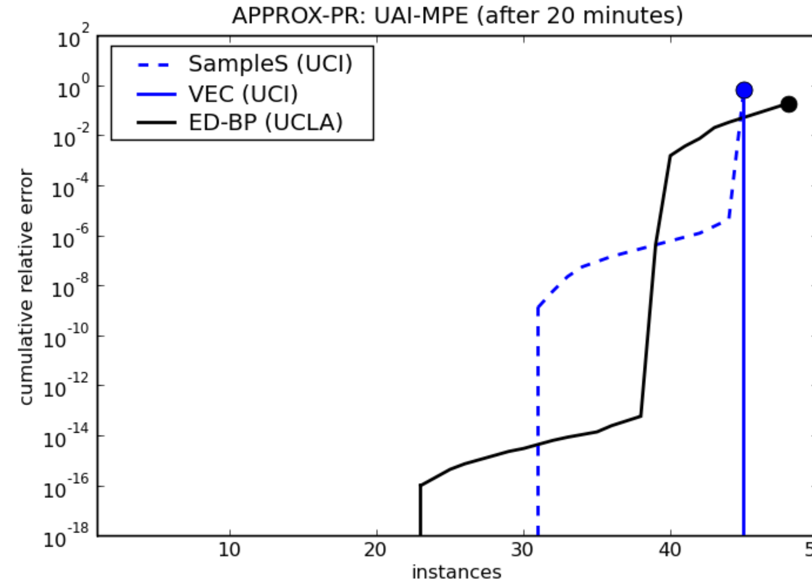

APPROX-PR: WCSPs (after 20 minutes)
SampleS (UCI)
VEC (UCI)
ED-BP (UCLA)
cumulative relative error
instances

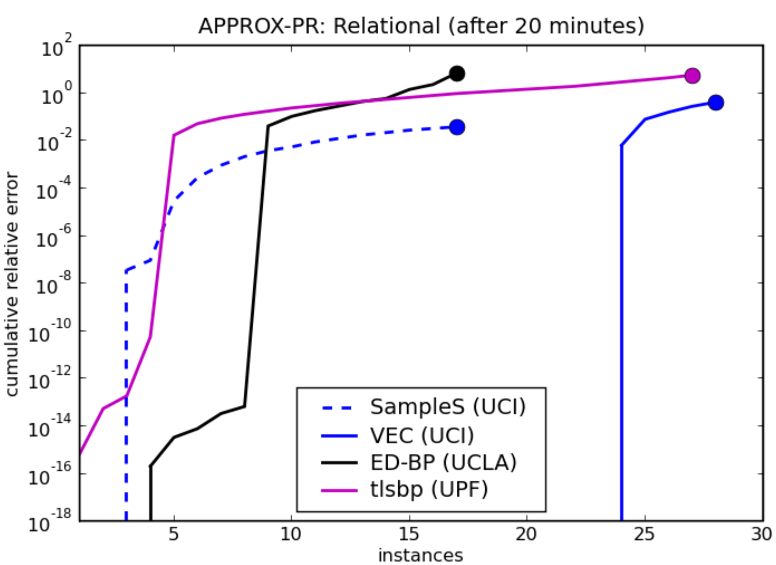

APPROX-PR: BN20 (after 20 minutes)
SampleS (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (UPF)
cumulative relative error
instances

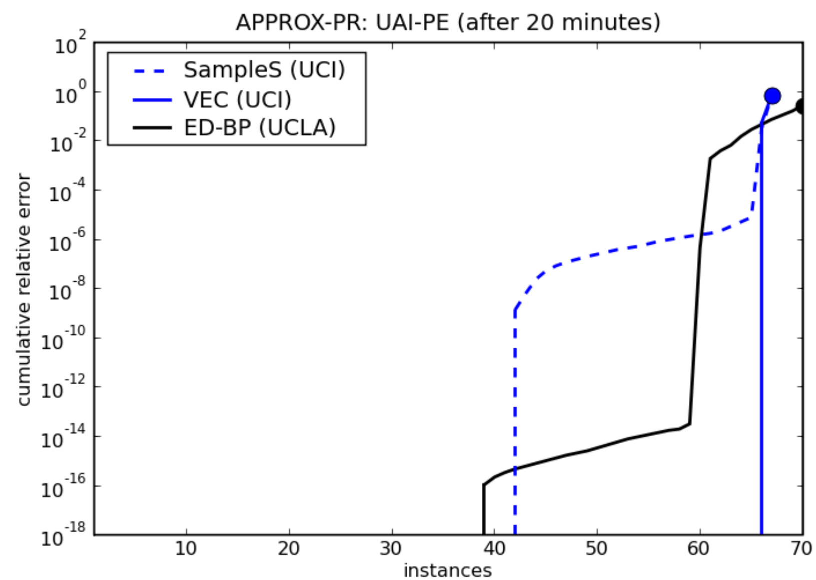

APPROX-PR: Linkage (after 20 minutes)
SampleS (UCI)
VEC (UCI)
ED-BP (UCLA)
cumulative relative error
instances

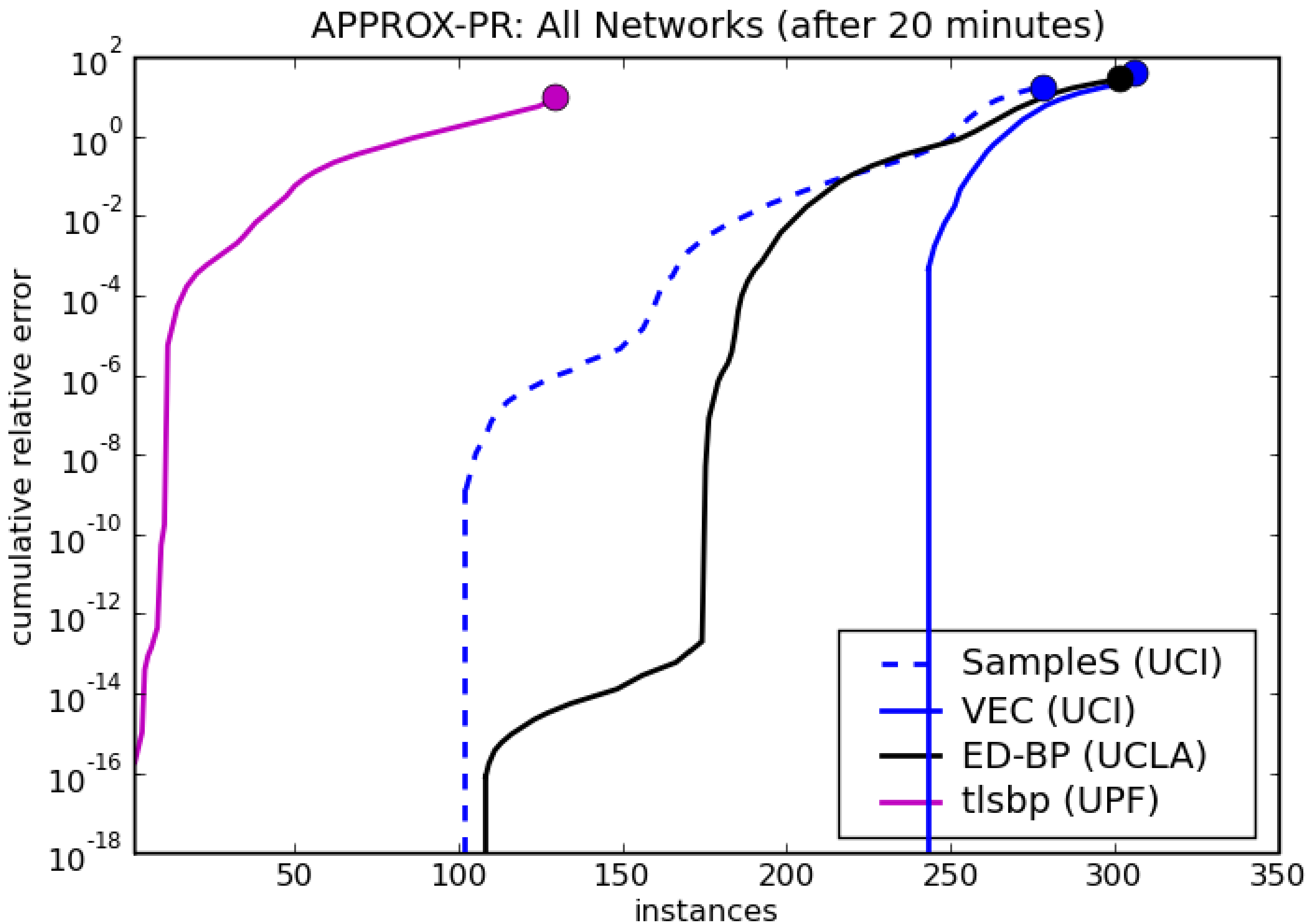

APPROX-PR: Grids (after 20 minutes)
SampleS (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (UPF)
cumulative relative error
instances


APPROX-PR: Promedas (after 20 minutes)
cumulative relative error
instances
SampleS (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (UPF)

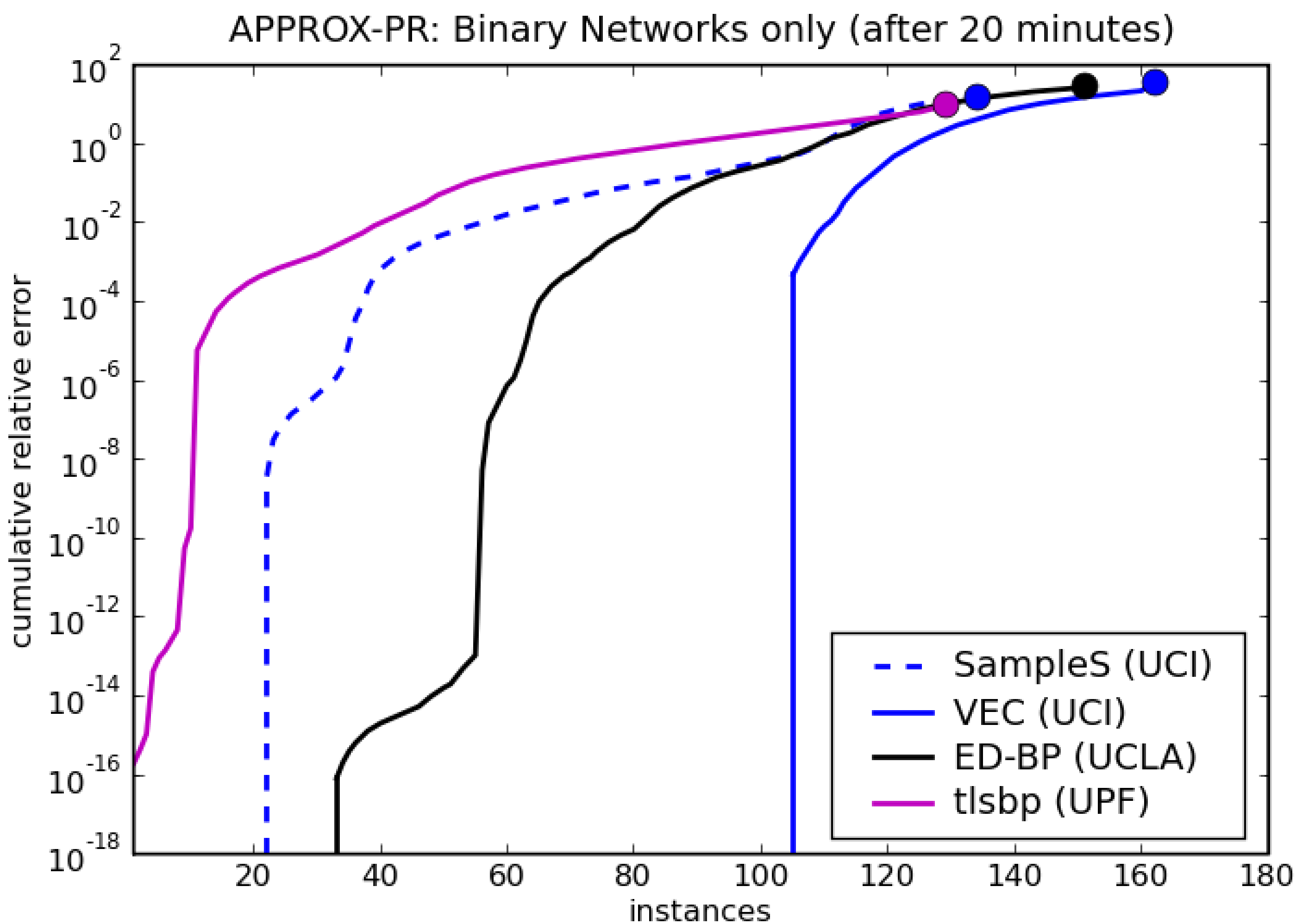

APPROX-PR: UAI-MPE (after 20 minutes)
cumulative relative error
instances
SampleS (UCI)
VEC (UCI)
ED-BP (UCLA)


APPROX-PR: Relational (after 20 minutes)
cumulative relative error
instances
SampleS (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (UPF)

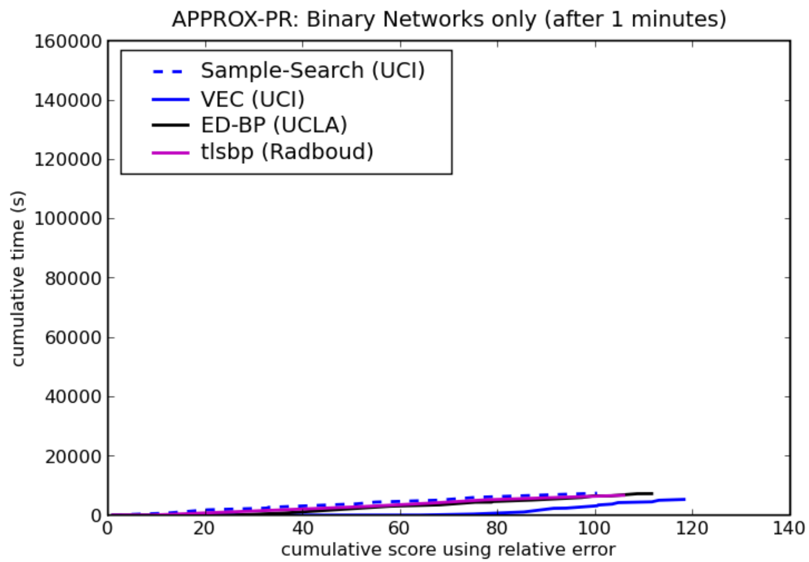

APPROX-PR: UAI-PE (after 20 minutes)
cumulative relative error
instances
SampleS (UCI)
VEC (UCI)
ED-BP (UCLA)

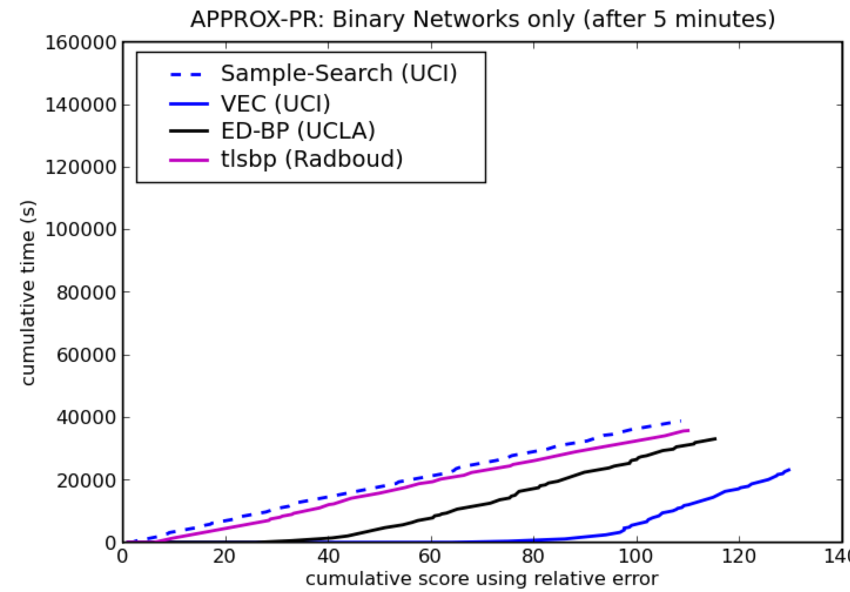


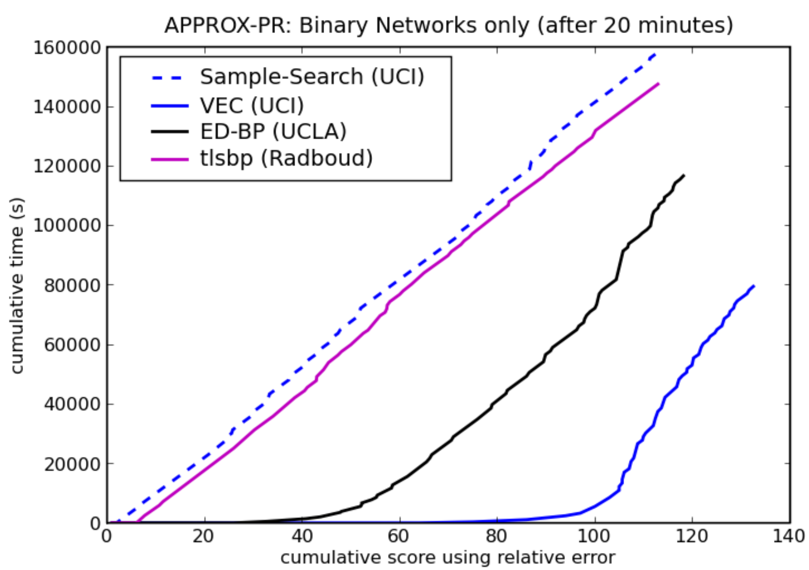
APPROX-PR: All Networks (after 20 minutes)
cumulative relative error
instances
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
10^{-16}
10^{-18}
50
100
150
200
250
300
350
SampleS (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (UPF)

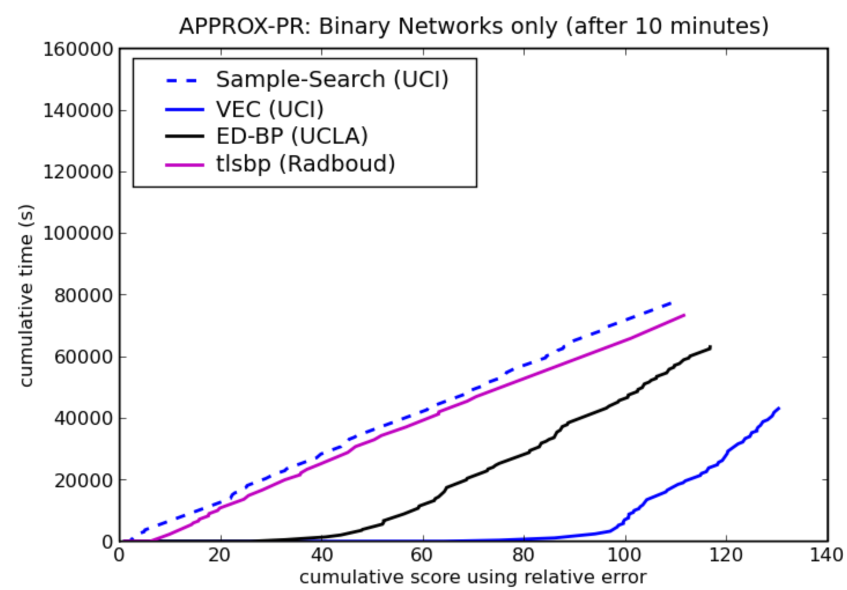


APPROX-PR: Binary Networks only (after 20 minutes)
cumulative relative error
instances
SampleS (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (UPF)


APPROX-PR: Binary Networks only (after 1 minutes)
Sample-Search (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using relative error


APPROX-PR: Binary Networks only (after 5 minutes)
Sample-Search (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using relative error


APPROX-PR: Binary Networks only (after 20 minutes)
Sample-Search (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using relative error


APPROX-PR: Binary Networks only (after 10 minutes)
Sample-Search (UCI)
VEC (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using relative error

Evaluating Approximate MAR Solvers

Benchmark Summary (mar)

- 9 sets:

		Bys	Mkv	bin
– weighted-CSP	(97)		•	
– bn2o (diagnosis)	(18)	•		•
– hand-built	(0/100)	•		
– grids	(32/320)	•		•
– linkage	(22)		•	
– Promedas	(238)		•	•
– UAI-06 (MPE)	(57)	•		
– UAI-06 (PE)	(78)	•		
– relational	(35/251)	•		•
– **TOTAL**	**(577)**	**(220)**	**(357)**	**(323)**

Approximate MAR

- 5 Solvers:
  - *All problems:*
    - *irvine-SampleSearch (Gogate and Dechter, 2007)*
      - *importance sampling based solver with emphasis on zero probabilities*
    - *irvine-ijgp (Dechter et al., 2002)*
      - *generalize belief propagation based solver*
    - *ucla-edbp (Choi and Darwiche, 2007)*
      - *generalized belief propagation based solver*
  - Binary-variable problems only:
    - *upf-mar (Gomez et al, 2007)*
  - *Bayesian networks only*
    - *pitt-epis (Changhe and Druzdzel, 2004)*
      - *adaptive importance sampling based solver*

Approx MAR Score (KL error)

- No solution
  - score = 0
- Otherwise, given the average error:

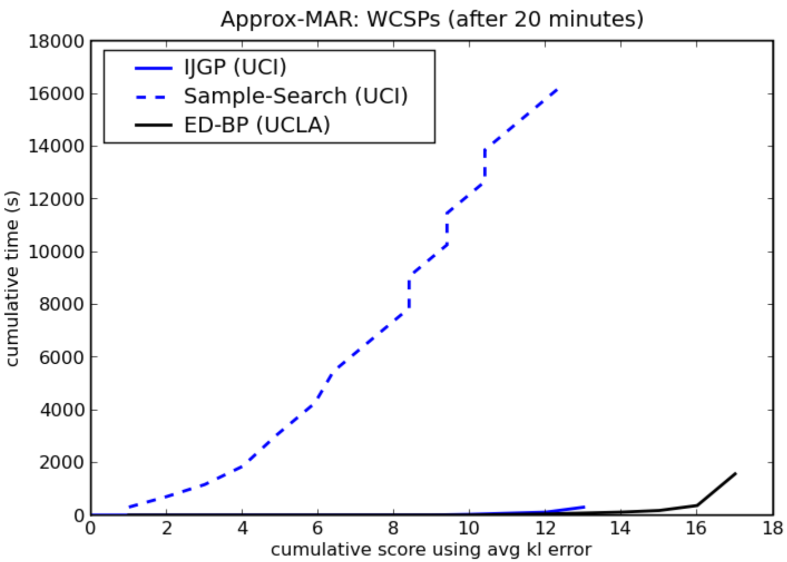
$$err = \frac{1}{N} Avg_X [KL(\Pr_{exact}(X), \Pr_{sol}(X))]$$

- Compute the score:

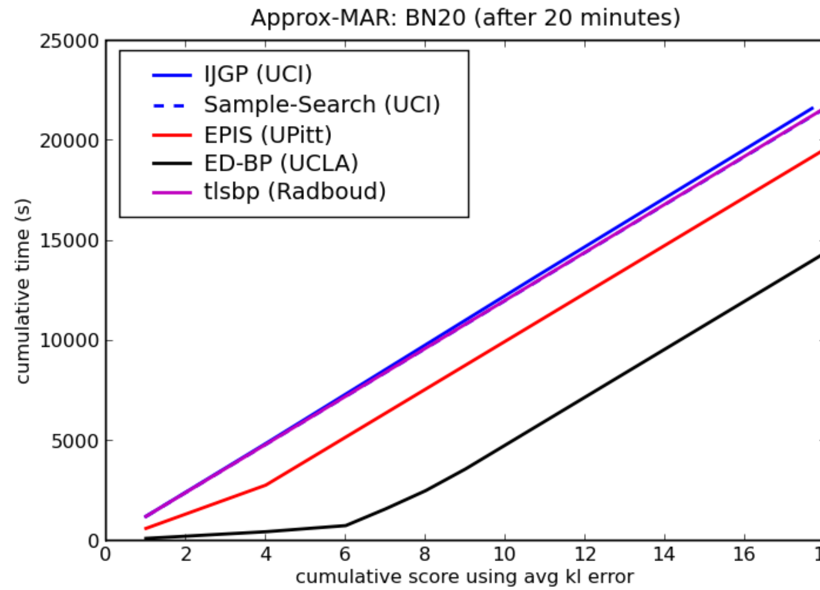
$$\text{score} = 10^{\text{-err}}$$

1 point for exact

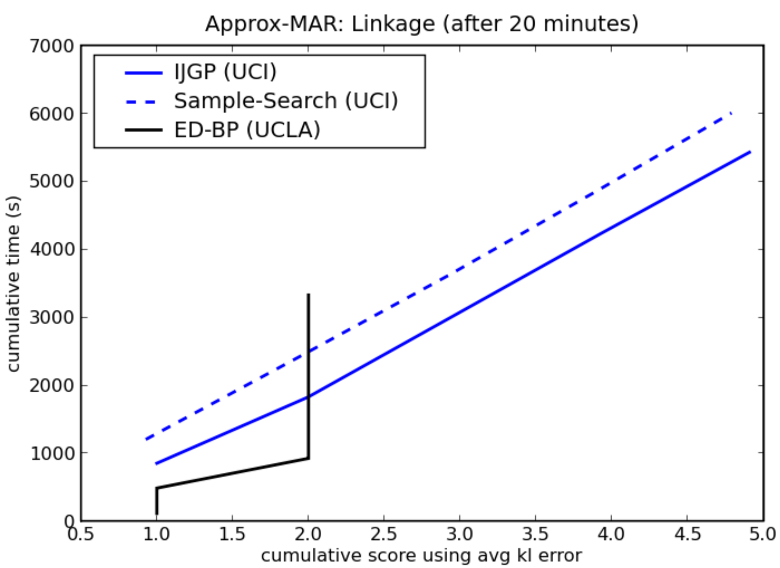
over minutes:
1, 2, 5, 10, 20



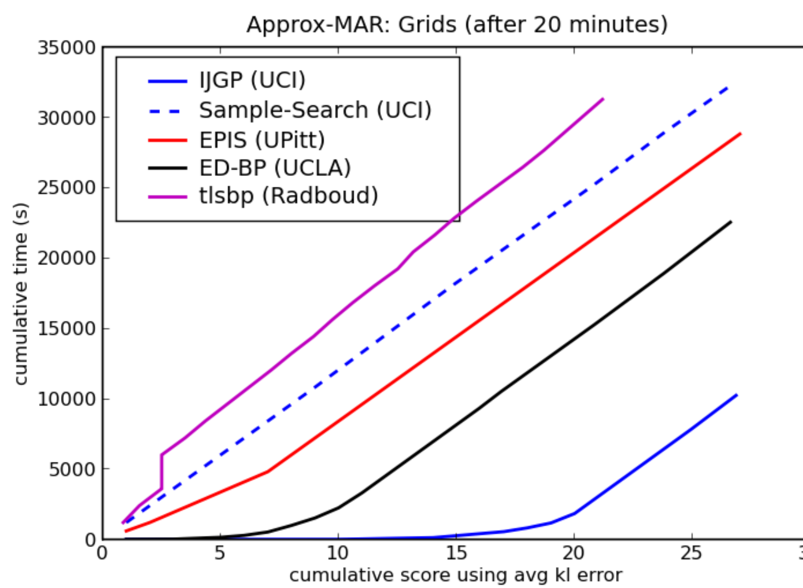
Approx-MAR: WCSPs (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
ED-BP (UCLA)
cumulative time (s)
cumulative score using avg kl error



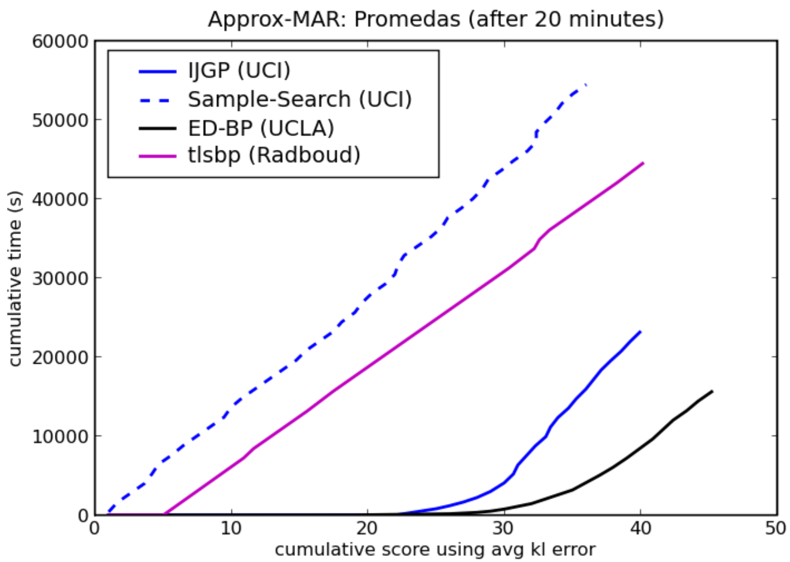
Approx-MAR: BN20 (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
EPIS (UPitt)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using avg kl error



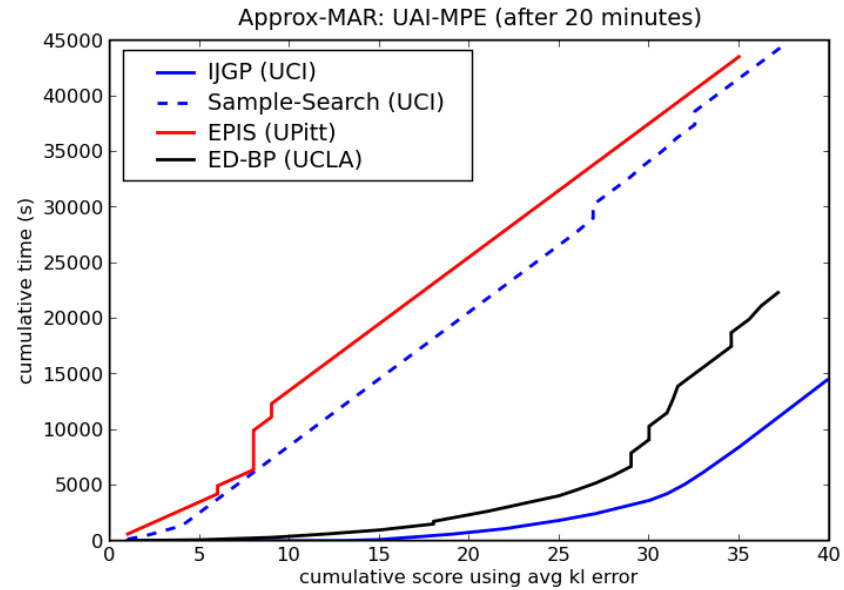
Approx-MAR: Linkage (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
ED-BP (UCLA)
cumulative time (s)
cumulative score using avg kl error



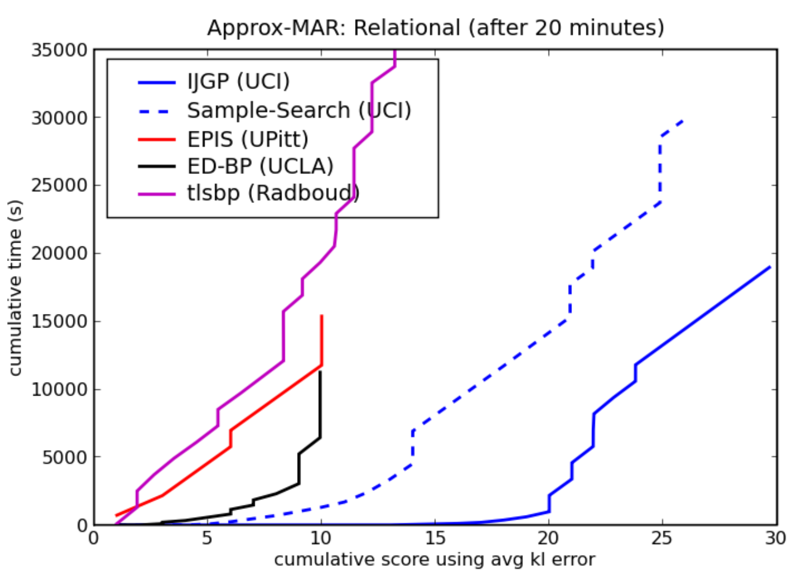
Approx-MAR: Grids (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
EPIS (UPitt)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using avg kl error



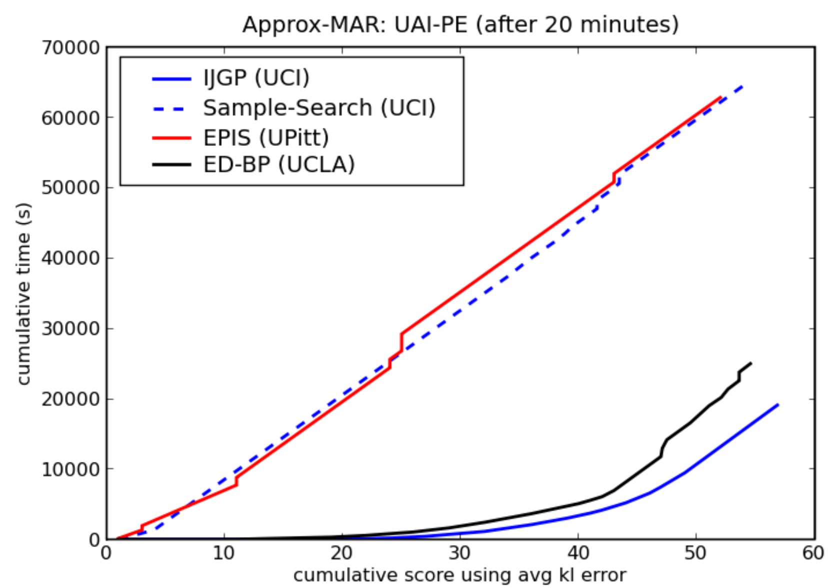
Approx-MAR: Promedas (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using avg kl error



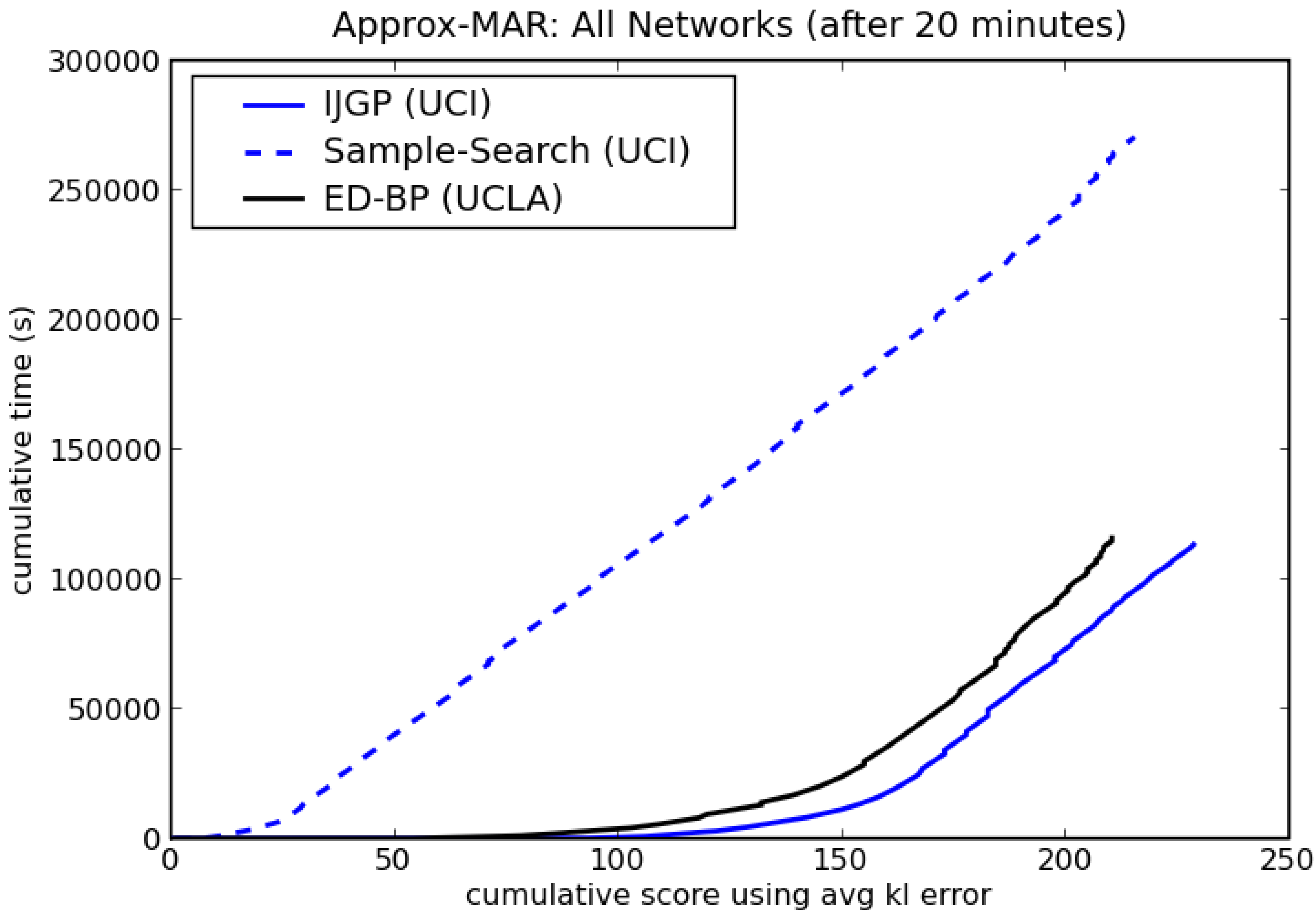
Approx-MAR: UAI-MPE (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
EPIS (UPitt)
ED-BP (UCLA)
cumulative time (s)
cumulative score using avg kl error



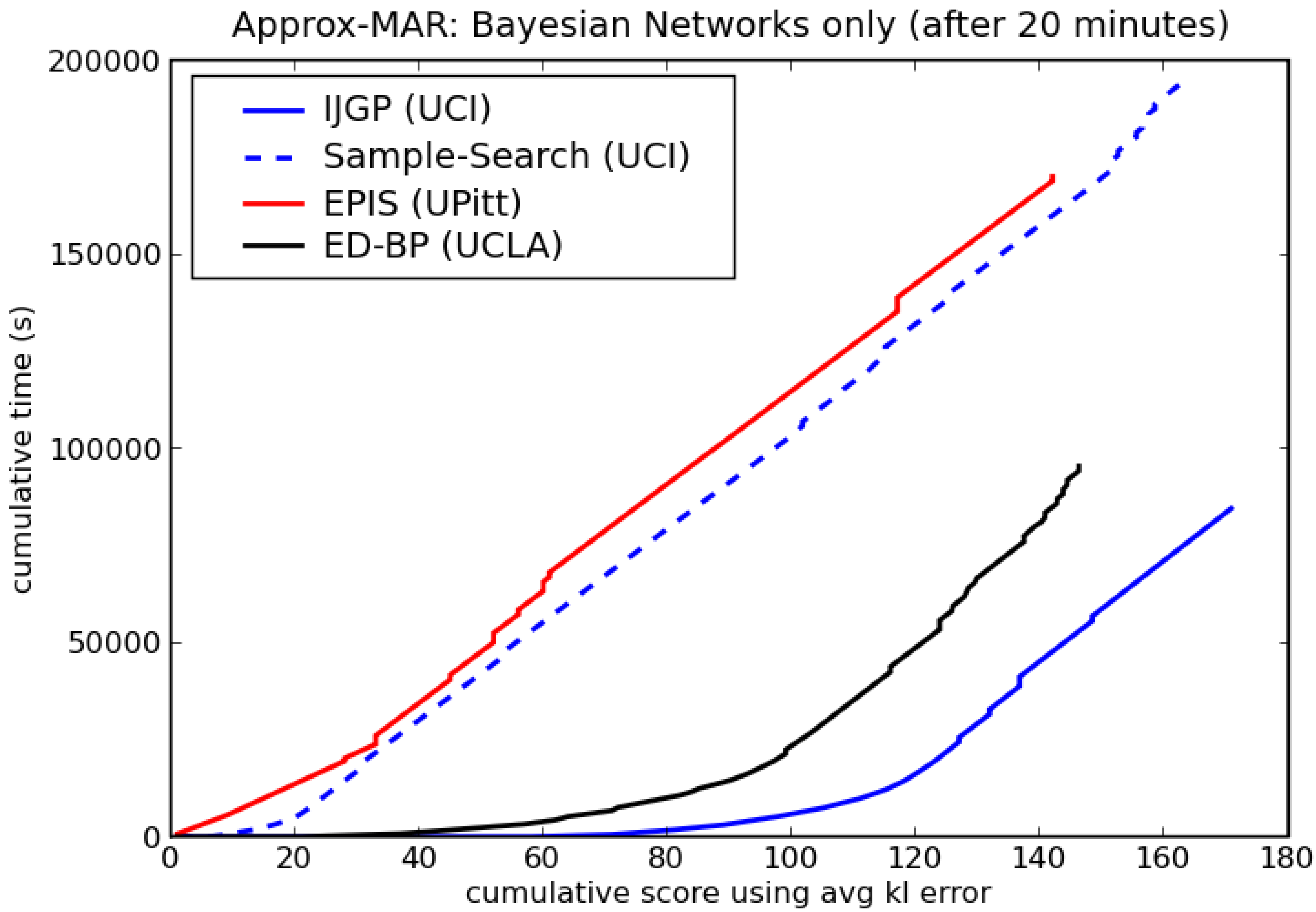
Approx-MAR: Relational (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
EPIS (UPitt)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using avg kl error



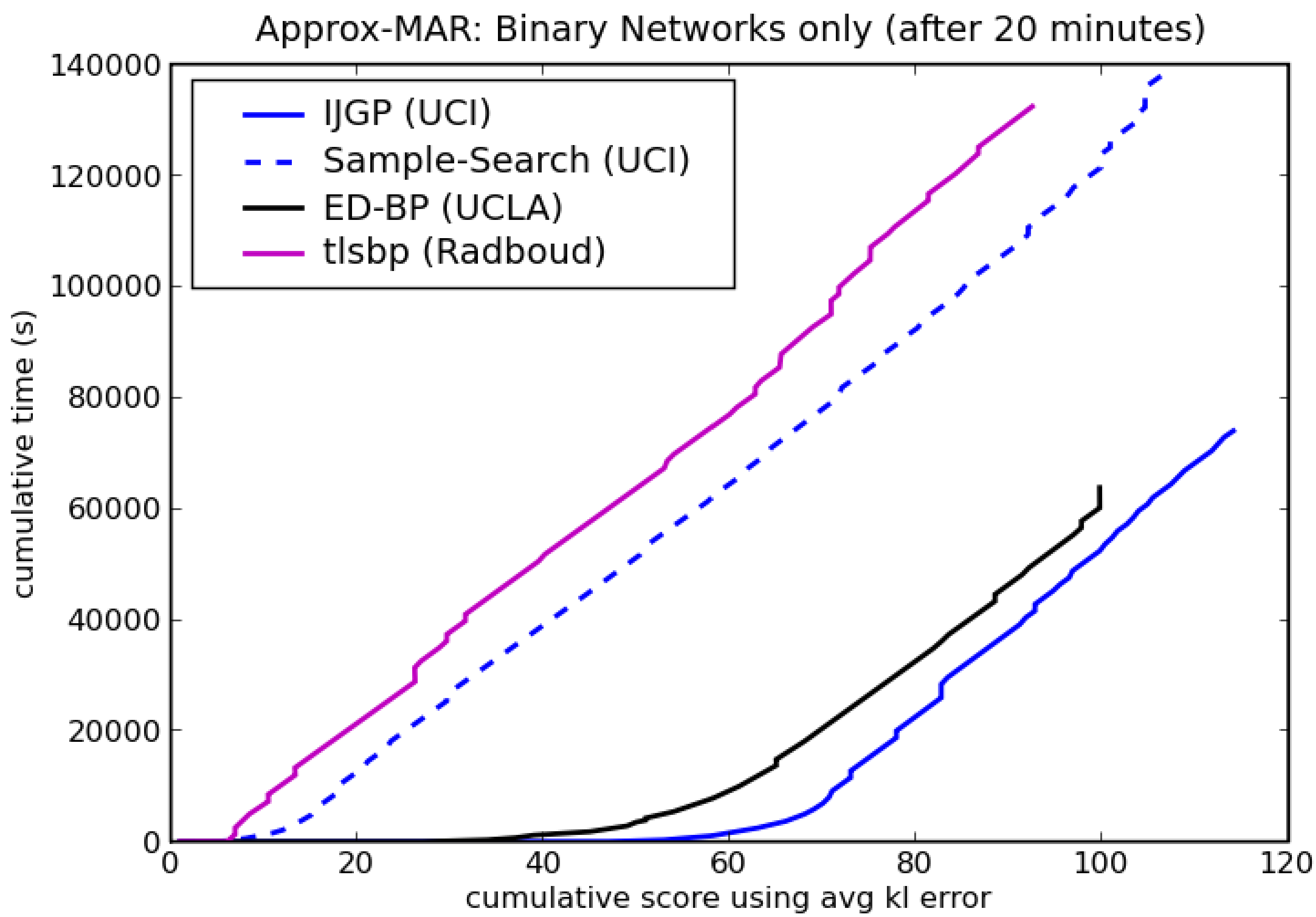
Approx-MAR: UAI-PE (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
EPIS (UPitt)
ED-BP (UCLA)
cumulative time (s)
cumulative score using avg kl error



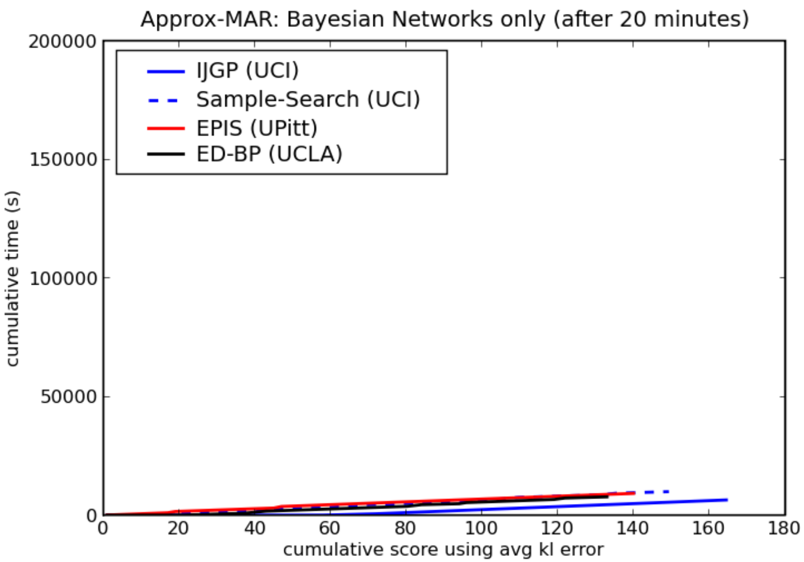
Approx-MAR: All Networks (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
ED-BP (UCLA)
cumulative time (s)
cumulative score using avg kl error
0
50000
100000
150000
200000
250000
300000
0
50
100
150
200
250

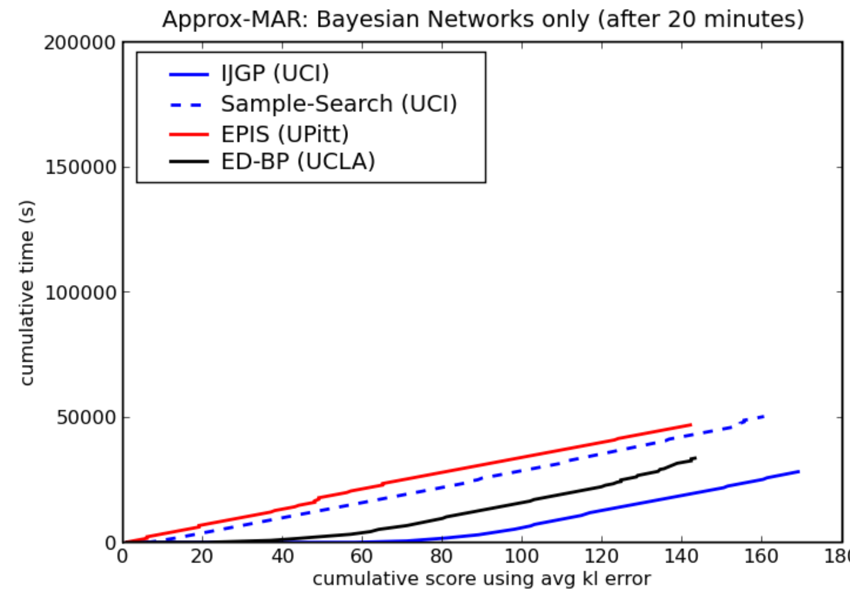


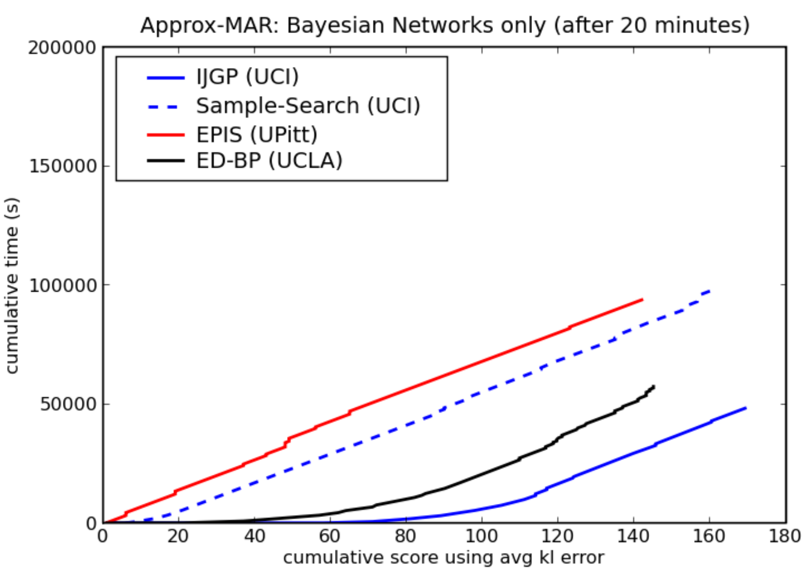
Approx-MAR: Bayesian Networks only (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
EPIS (UPitt)
ED-BP (UCLA)
cumulative time (s)
cumulative score using avg kl error
0
50000
100000
150000
200000
0
20
40
60
80
100
120
140
160
180

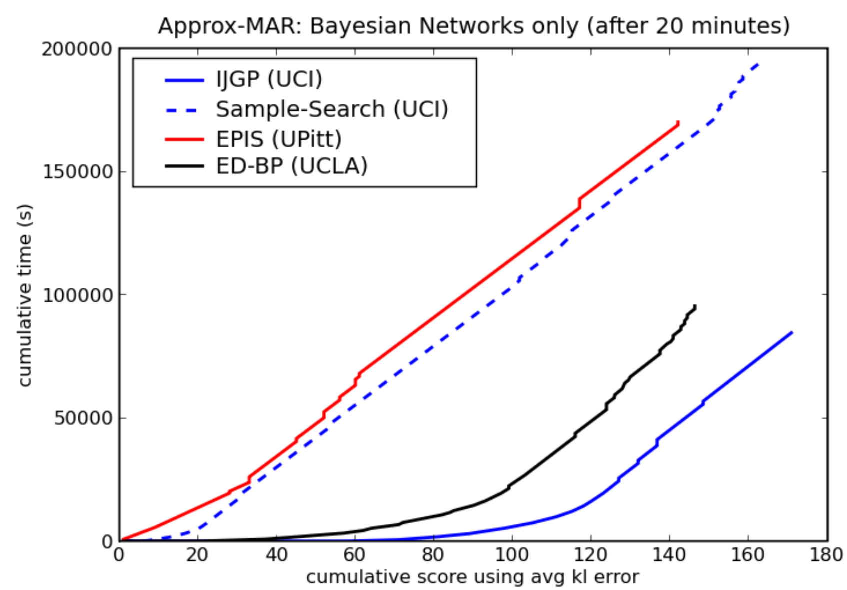


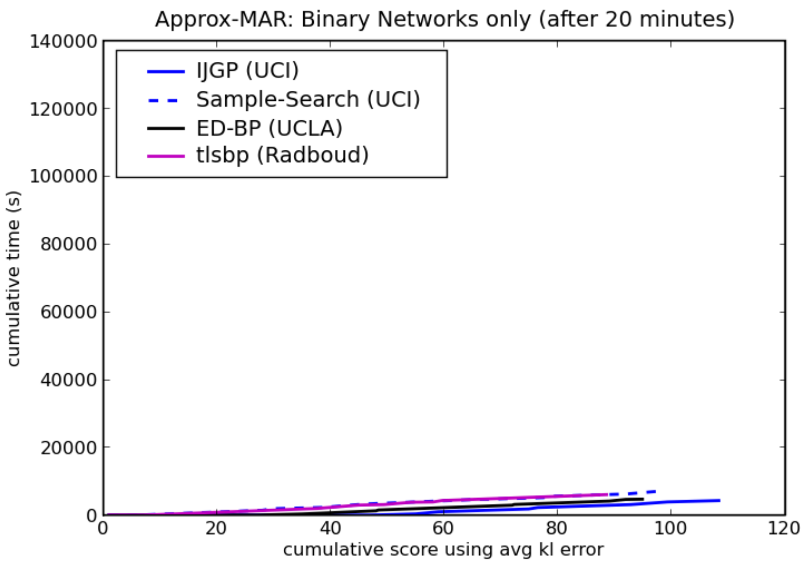
Approx-MAR: Binary Networks only (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using avg kl error
0
20000
40000
60000
80000
100000
120000
140000
0
20
40
60
80
100
120

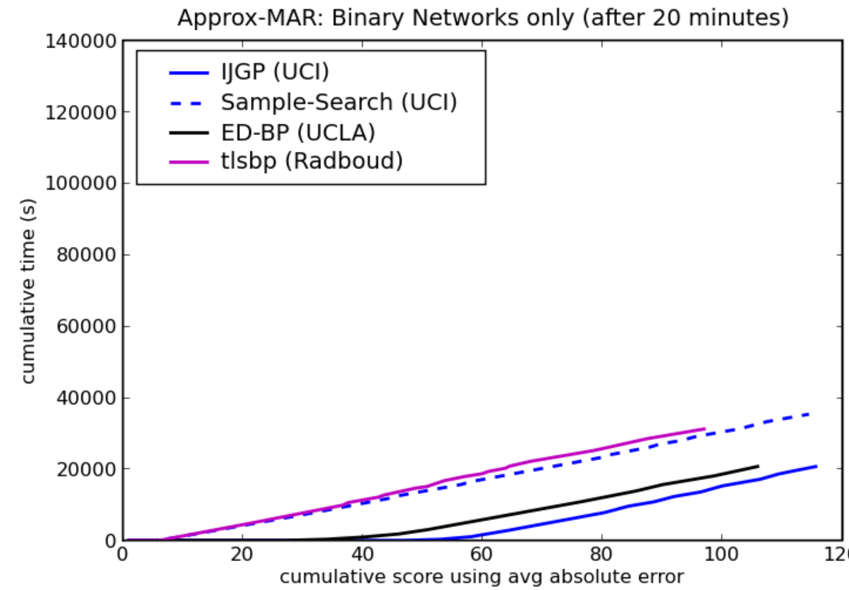

Approx-MAR: Bayesian Networks only (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
EPIS (UPitt)
ED-BP (UCLA)
cumulative time (s)
cumulative score using avg kl error

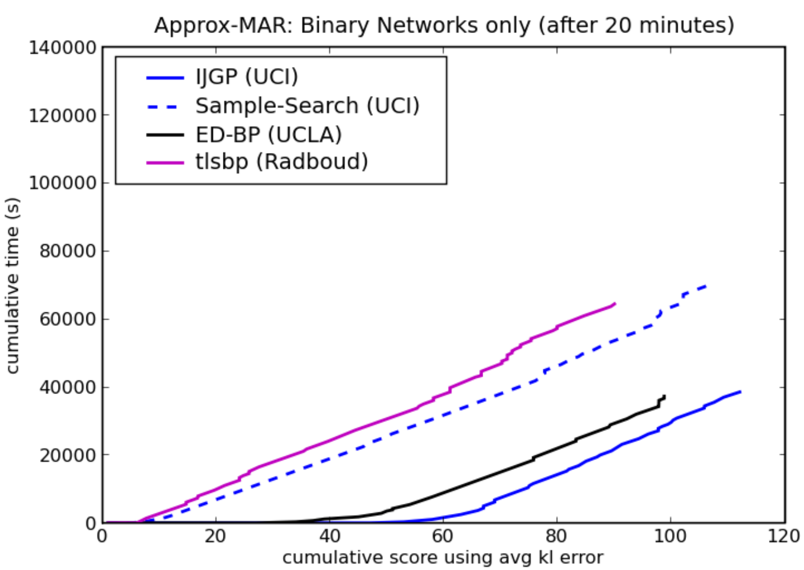

Approx-MAR: Bayesian Networks only (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
EPIS (UPitt)
ED-BP (UCLA)
cumulative time (s)
cumulative score using avg kl error

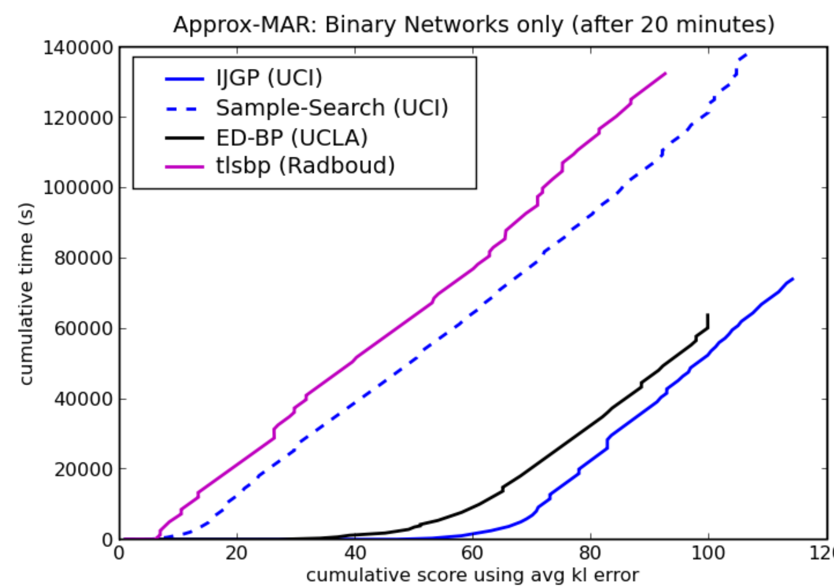

Approx-MAR: Bayesian Networks only (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
EPIS (UPitt)
ED-BP (UCLA)
cumulative time (s)
cumulative score using avg kl error

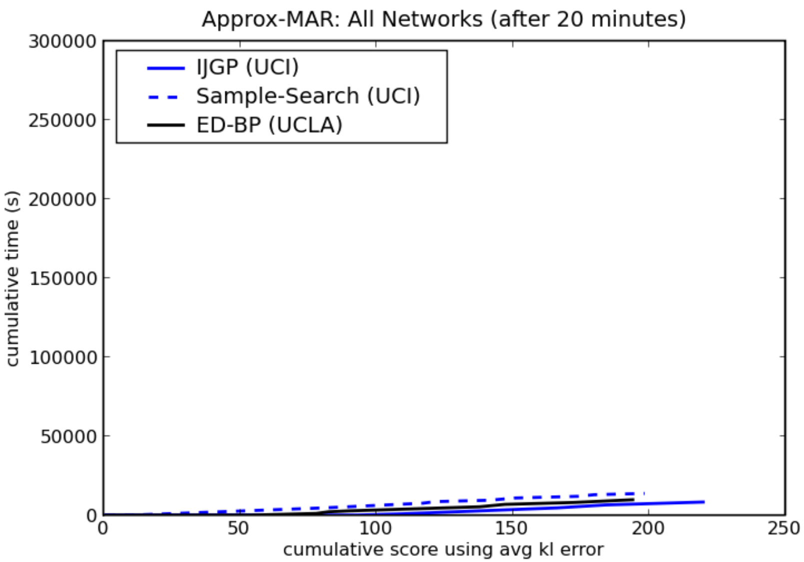

Approx-MAR: Bayesian Networks only (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
EPIS (UPitt)
ED-BP (UCLA)
cumulative time (s)
cumulative score using avg kl error

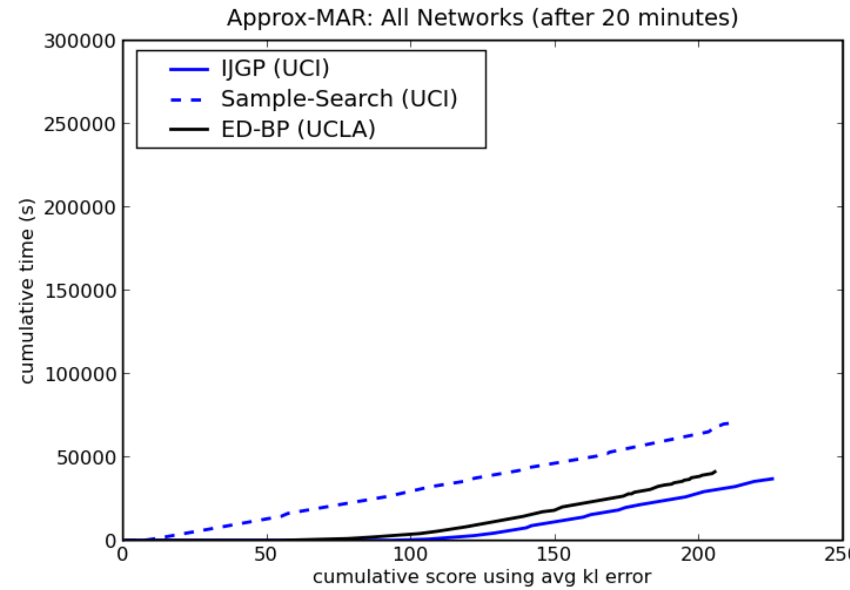

Approx-MAR: Binary Networks only (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using avg kl error

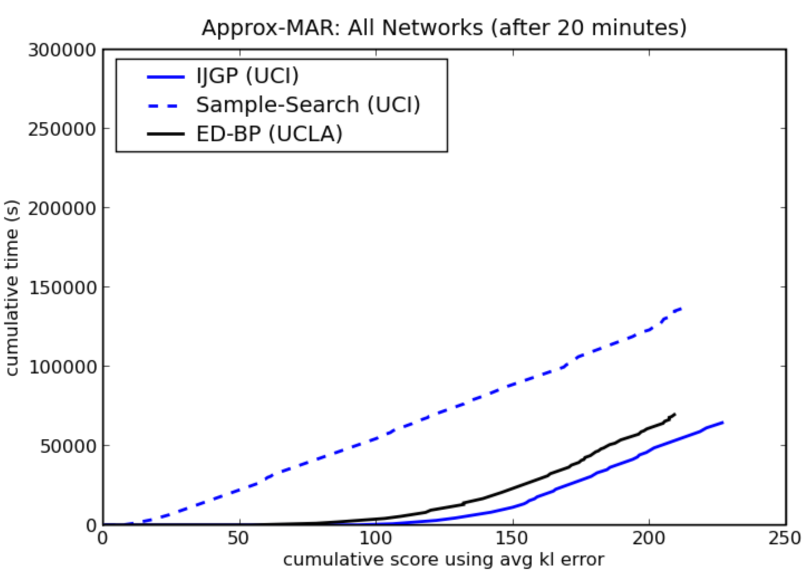

Approx-MAR: Binary Networks only (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using avg absolute error

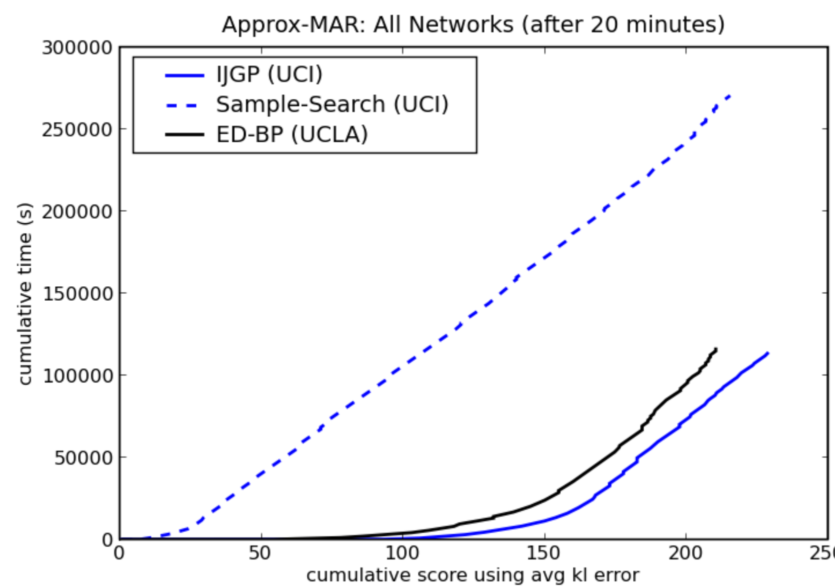

Approx-MAR: Binary Networks only (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using avg kl error


Approx-MAR: Binary Networks only (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
ED-BP (UCLA)
tlsbp (Radboud)
cumulative time (s)
cumulative score using avg kl error


Approx-MAR: All Networks (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
ED-BP (UCLA)
cumulative time (s)
cumulative score using avg kl error
300000
250000
200000
150000
100000
50000
0
0
50
100
150
200
250


Approx-MAR: All Networks (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
ED-BP (UCLA)
cumulative time (s)
cumulative score using avg kl error
300000
250000
200000
150000
100000
50000
0
0
50
100
150
200
250


Approx-MAR: All Networks (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
ED-BP (UCLA)
cumulative time (s)
cumulative score using avg kl error
300000
250000
200000
150000
100000
50000
0
0
50
100
150
200
250


Approx-MAR: All Networks (after 20 minutes)
IJGP (UCI)
Sample-Search (UCI)
ED-BP (UCLA)
cumulative time (s)
cumulative score using avg kl error
300000
250000
200000
150000
100000
50000
0
0
50
100
150
200
250

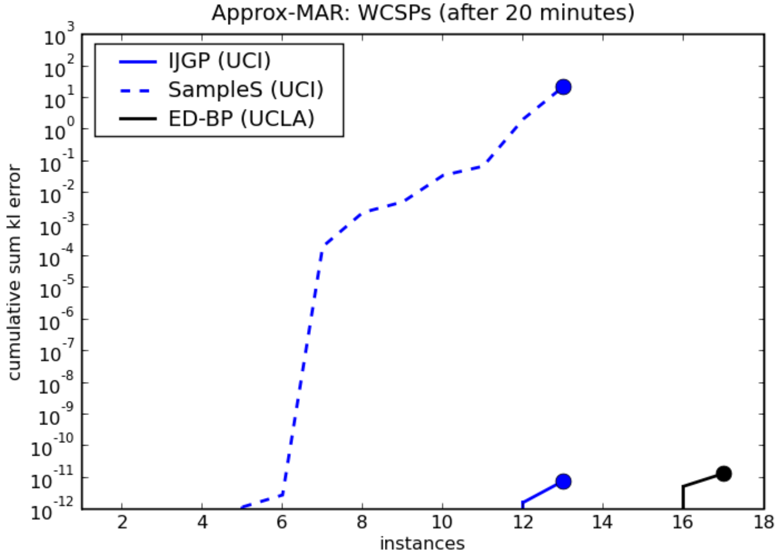
Approx MAR Error plots (KL error)

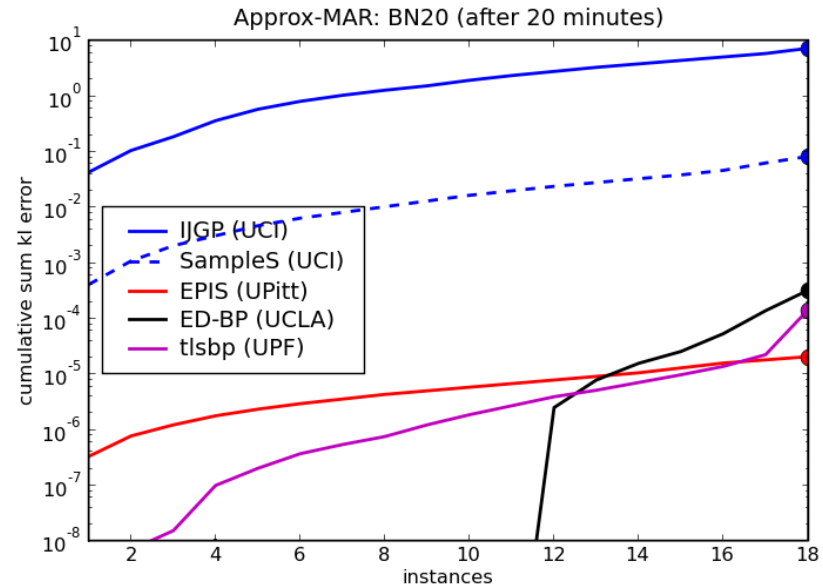
- For each instance compute Average error:

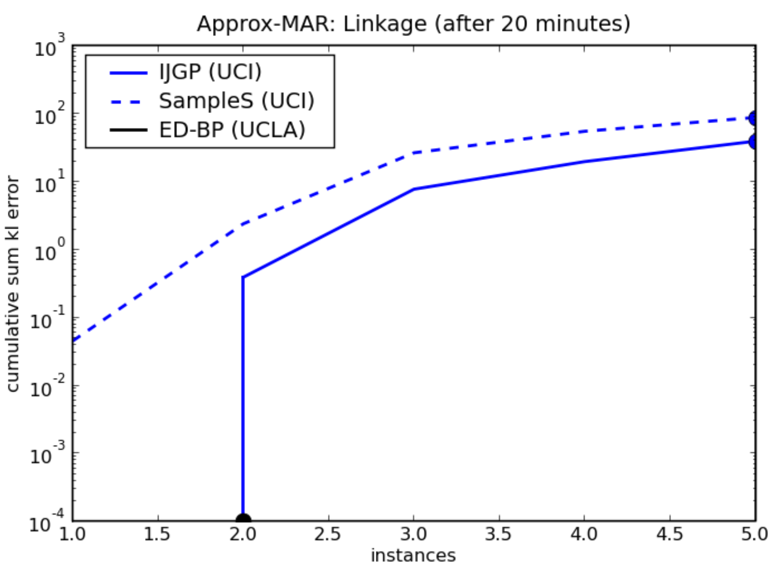
$$err = \frac{1}{N} Avg_X [KL(\Pr_{exact}(X), \Pr_{sol}(X))]$$

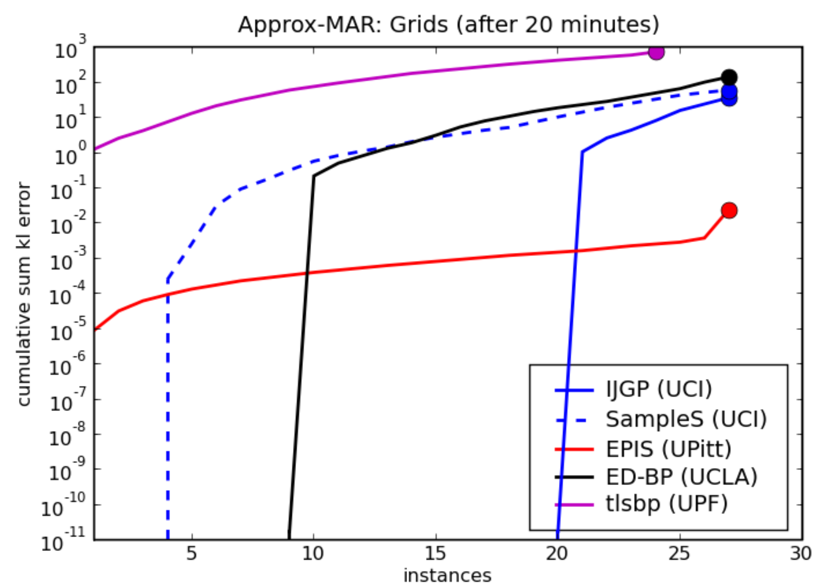
- Sort instances based on err
- Plot cumulative err vs instances

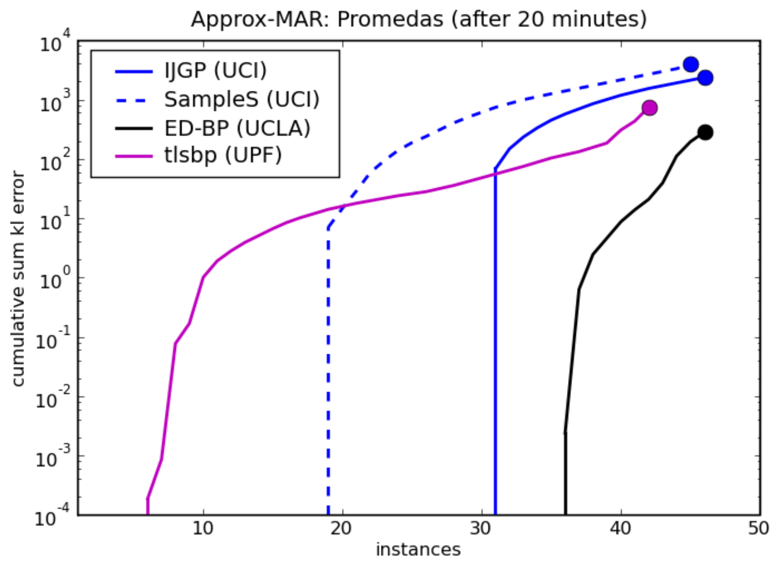
over minutes:
1, 2, 5, 10, 20


Approx-MAR: WCSPs (after 20 minutes)
cumulative sum kl error
instances
IJGP (UCI)
SampleS (UCI)
ED-BP (UCLA)

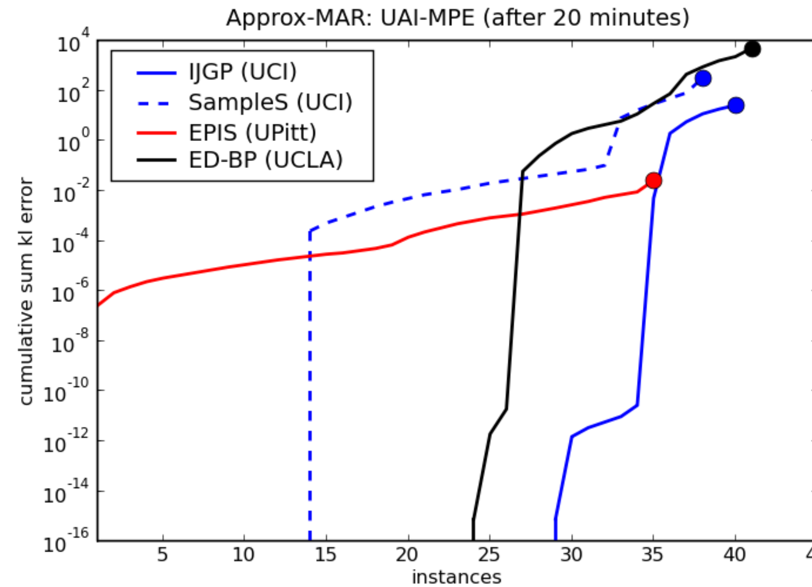

Approx-MAR: BN20 (after 20 minutes)
cumulative sum kl error
instances
IJGP (UCI)
SampleS (UCI)
EPIS (UPitt)
ED-BP (UCLA)
tlsbp (UPF)


Approx-MAR: Linkage (after 20 minutes)
cumulative sum kl error
instances
IJGP (UCI)
SampleS (UCI)
ED-BP (UCLA)

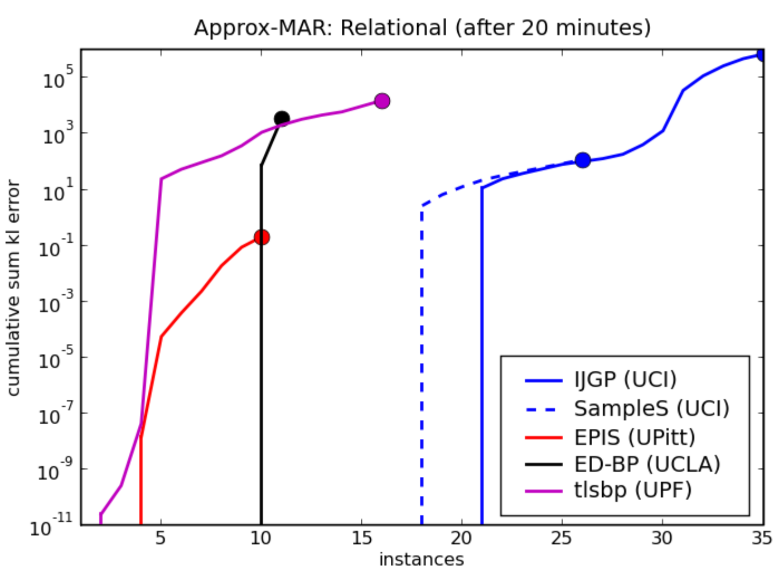

Approx-MAR: Grids (after 20 minutes)
cumulative sum kl error
instances
IJGP (UCI)
SampleS (UCI)
EPIS (UPitt)
ED-BP (UCLA)
tlsbp (UPF)



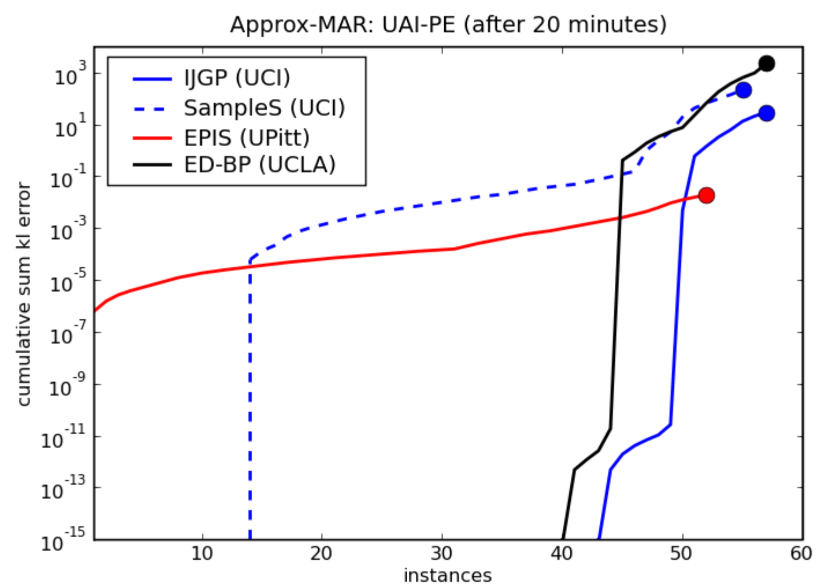
Approx-MAR: Promedas (after 20 minutes)
IJGP (UCI)
SampleS (UCI)
ED-BP (UCLA)
tlsbp (UPF)
cumulative sum kl error
instances



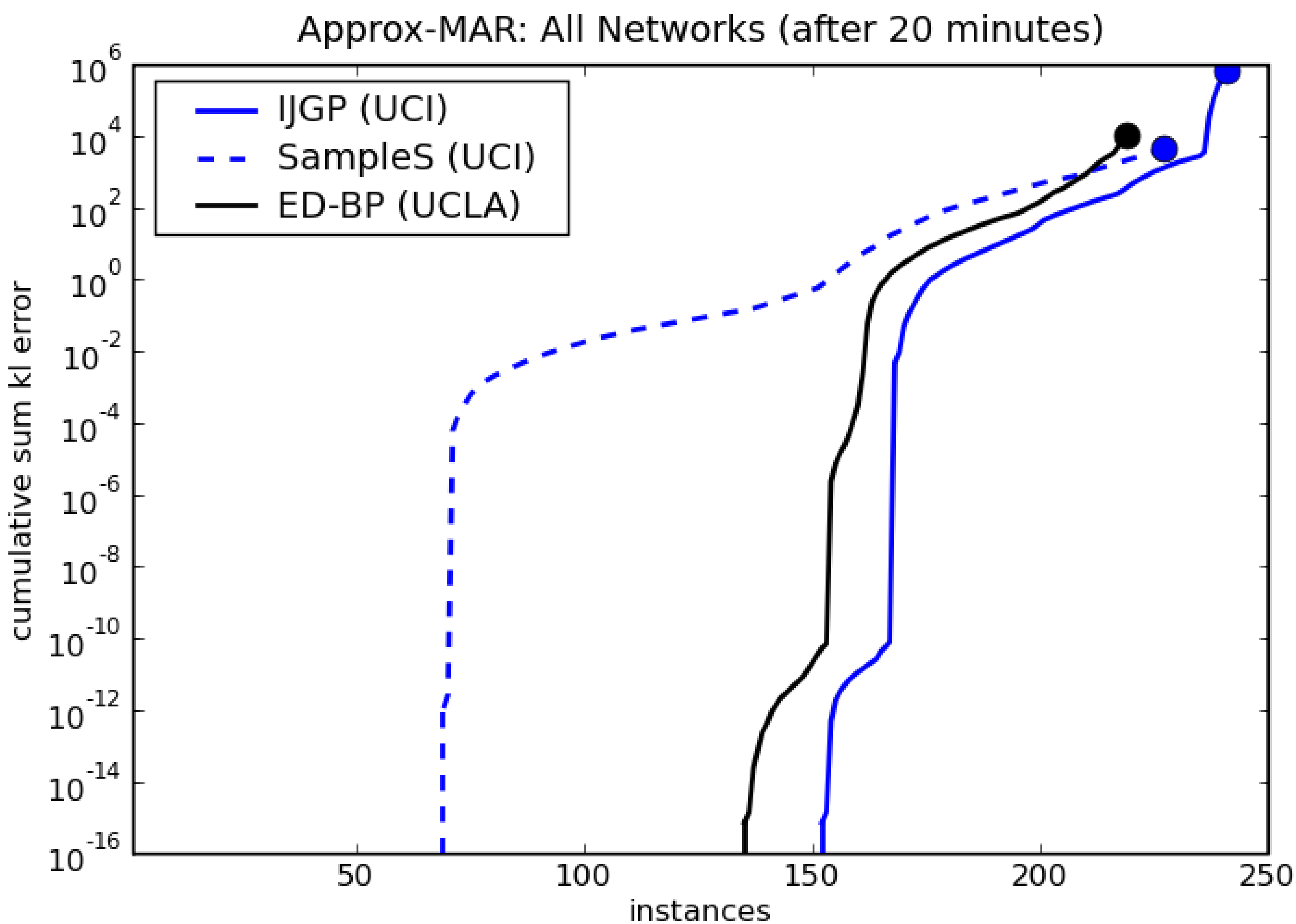
Approx-MAR: UAI-MPE (after 20 minutes)
IJGP (UCI)
SampleS (UCI)
EPIS (UPitt)
ED-BP (UCLA)
cumulative sum kl error
instances



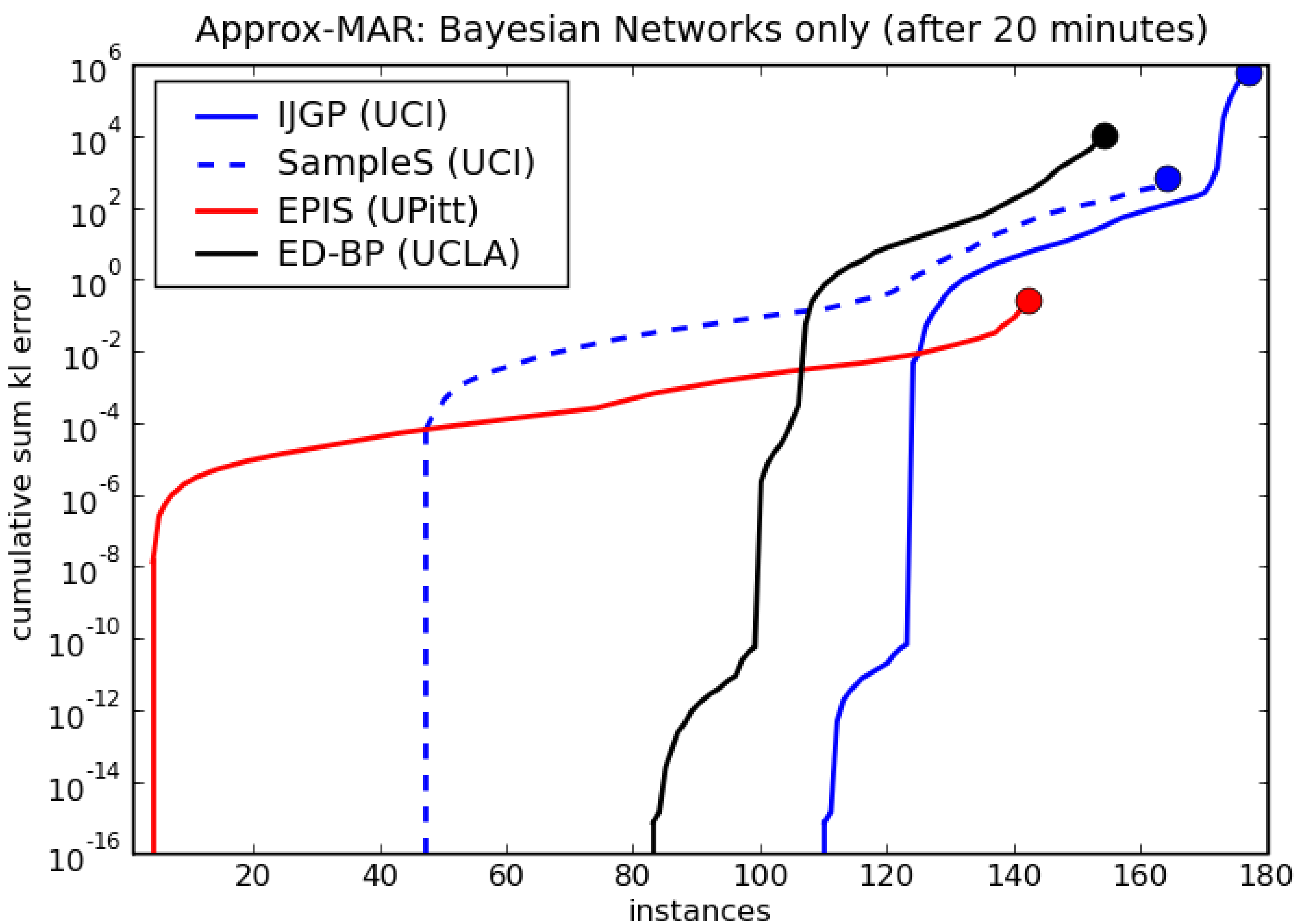
Approx-MAR: Relational (after 20 minutes)
IJGP (UCI)
SampleS (UCI)
EPIS (UPitt)
ED-BP (UCLA)
tlsbp (UPF)
cumulative sum kl error
instances

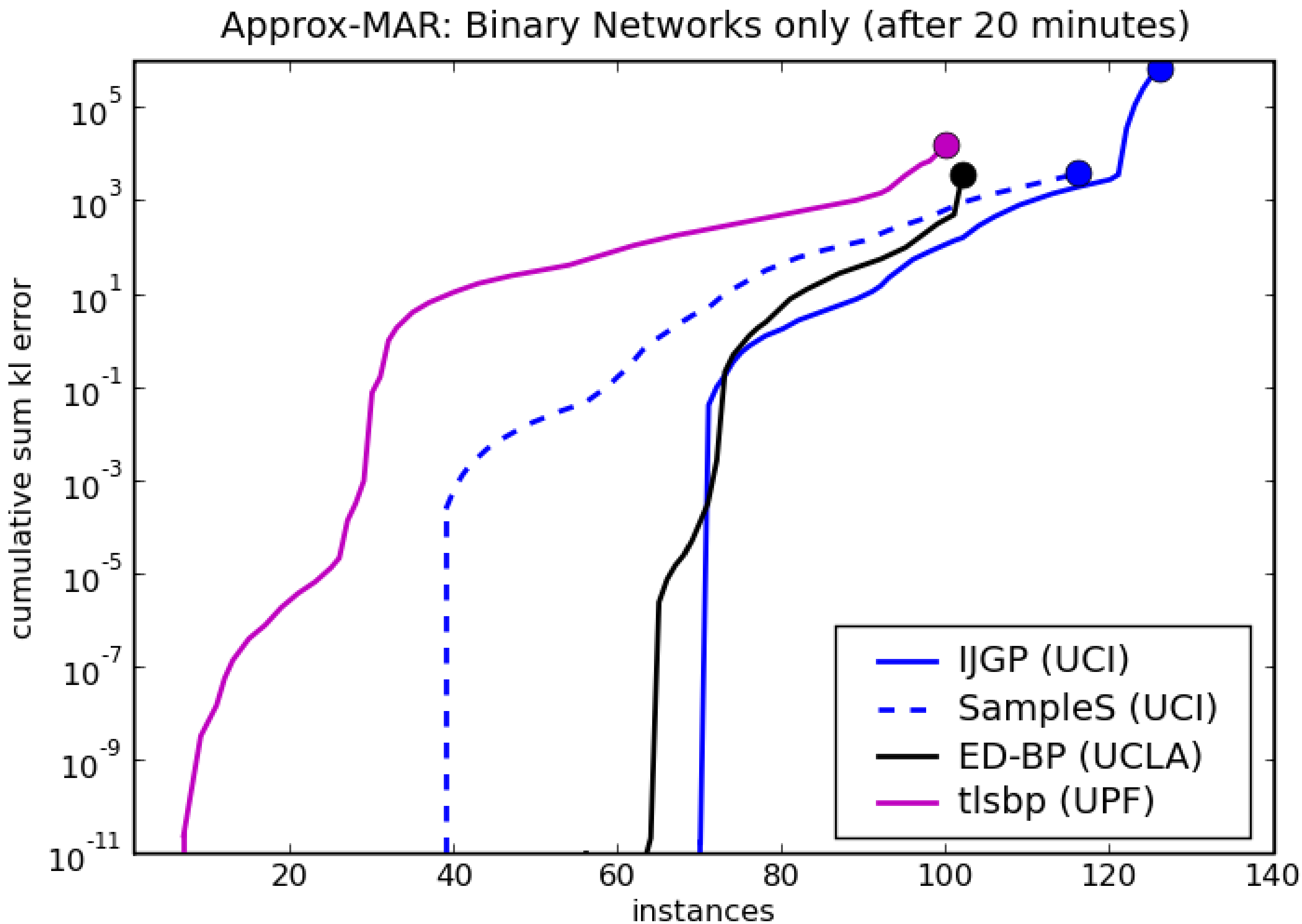


Approx-MAR: UAI-PE (after 20 minutes)
IJGP (UCI)
SampleS (UCI)
EPIS (UPitt)
ED-BP (UCLA)
cumulative sum kl error
instances



Approx-MAR: All Networks (after 20 minutes)
IJGP (UCI)
SampleS (UCI)
ED-BP (UCLA)
cumulative sum kl error
10^{6}
10^{4}
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
10^{-16}
50
100
150
200
250
instances


Approx-MAR: Bayesian Networks only (after 20 minutes)
IJGP (UCI)
SampleS (UCI)
EPIS (UPitt)
ED-BP (UCLA)
cumulative sum kl error
instances
10^{6}
10^{4}
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
10^{-16}
20
40
60
80
100
120
140
160
180



Approx-MAR: Binary Networks only (after 20 minutes)
cumulative sum kl error
instances
IJGP (UCI)
SampleS (UCI)
ED-BP (UCLA)
tlsbp (UPF)

What we plotted for Approx mar

- Cumulative Score vs Cumulative Time
- Cumulative Error vs Instances
  - KL, Hellinger, MSE, Absolute and Relative (5)
  - Sum and Average (2)
  - After 1, 2, 5, 10 and 20 minutes (5)
  - For each benchmark and each overall type (11)
  - Total plots (5 * 2*5*11 = 550 plots)

Concluding Remarks

- Need more difficult networks (with answers)
- Markov networks harder than Bayesian networks
- More time for evaluation
- Further investigations into the performance measures of approximate solvers and how they can be combined
- Computational clusters
- Benchmark normalization
- Split exact PE/MAR into two tracks?
- Post-workshop report