Search methods exploiting the problem structure

Tutorial – Kiev 09/2010

Simon de Givry
INRA, Toulouse, France

with contributed slides by Thomas Schiex (INRA, France), Javier Larrosa (UPC, Spain),
M. Sanchez, D. Allouche & A. Favier (INRA, France), R. Dechter (UCI, USA),
R. Marinescu (4C, Ireland), P. Jeavons (Univ. Oxford, UK), and M. Cooper (IRIT, France)

Framework: Weighted CSP

- (X,D,F,**k**)
  - $X=\{X_1,..., X_n\}$ **n** variables
  - $D=\{D_1,..., D_n\}$ **n** finite domains of maximum size **d**
  - $F=\{f_1,…,f_m\}$, a set of **m** cost functions
    - f_S with scope *S*, assigns a positive cost to any assignment of *S*
    - **$f_\varnothing$** problem lower bound (obtained by enforcing local consistency by *Equivalence Preserving Transformations*)
  - **k**, maximum cost and global upper bound

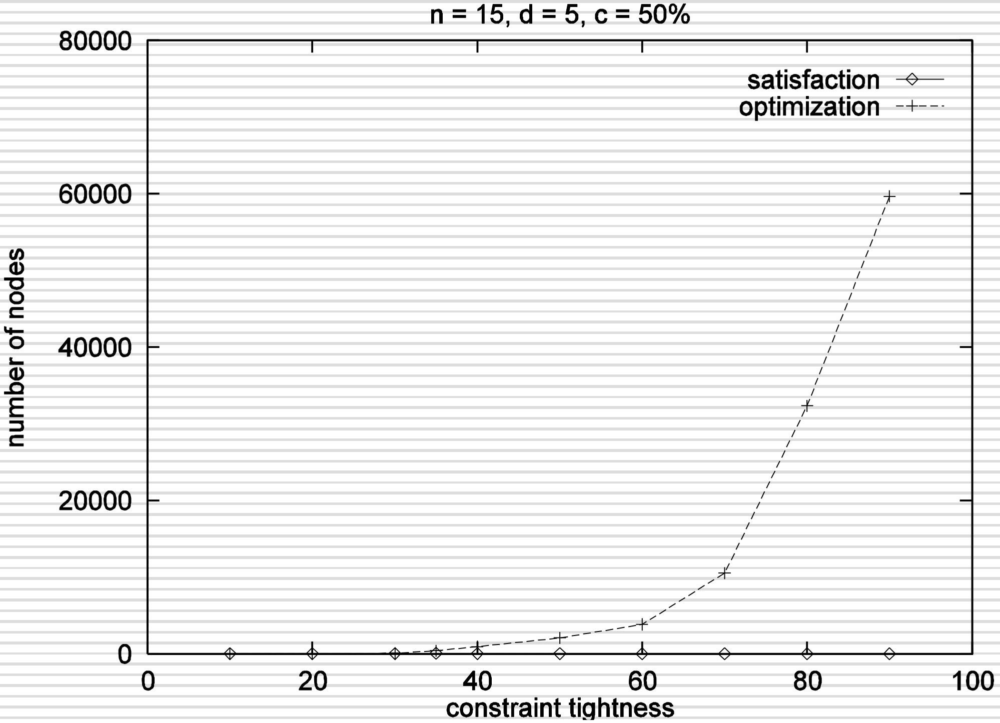
- Find an assignment A of all the variables which **minimizes**

$$\sum_{f_S \in F} f_S (A[S])$$

NP-hard problem

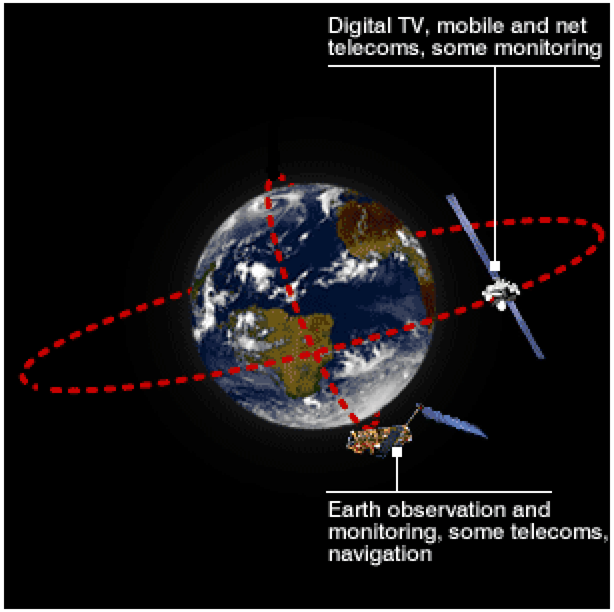
Observation

- ☐ Optimization problems are harder than satisfaction problems



CSP vs. Max-CSP

Combinatorial Optimization



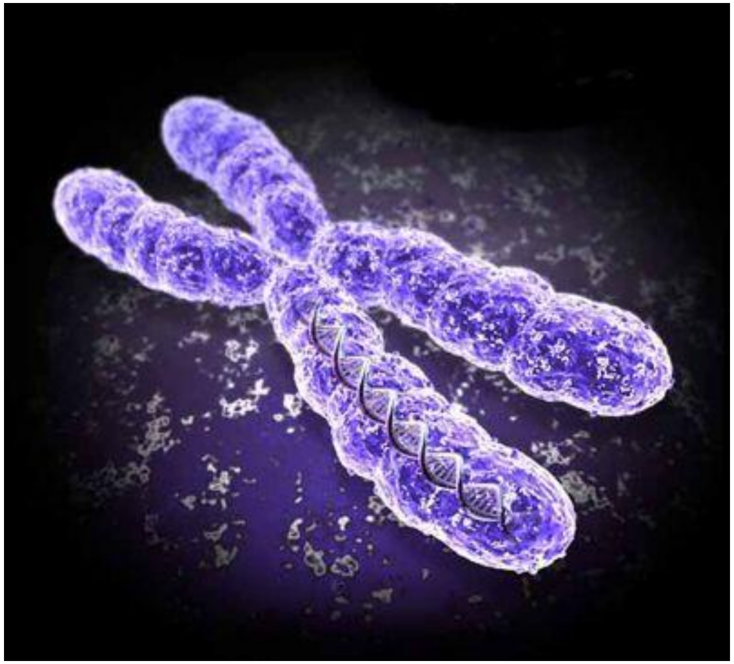
Find a schedule for the satellite that maximizes the number of photographs taken, subject to the on-board recording capacity

n ≤ 364, e ≤ 10,108

Assign frequencies to a set of radio links such that interferences are minimized

n ≤ 458, e ≤ 5,000

Combinatorial Optimization



Find a minimum number of erroneous genotypes such that their removal restores consistency in a complex pedigree

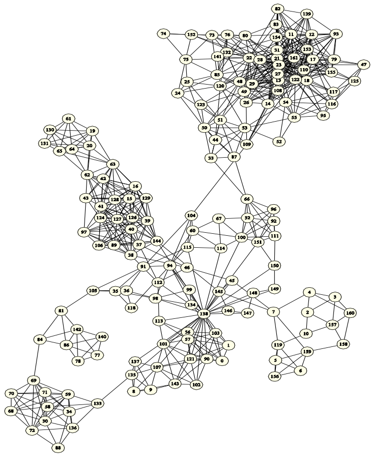
n ≤ 20,000, e ≤30,000

Select a minimum number of markers such that they cover all the markers and secondly, maximize weighted coverage sum of unselected markers and maximize dispersion between selected markers

n ≤ 1,500, e ≤ 150,000

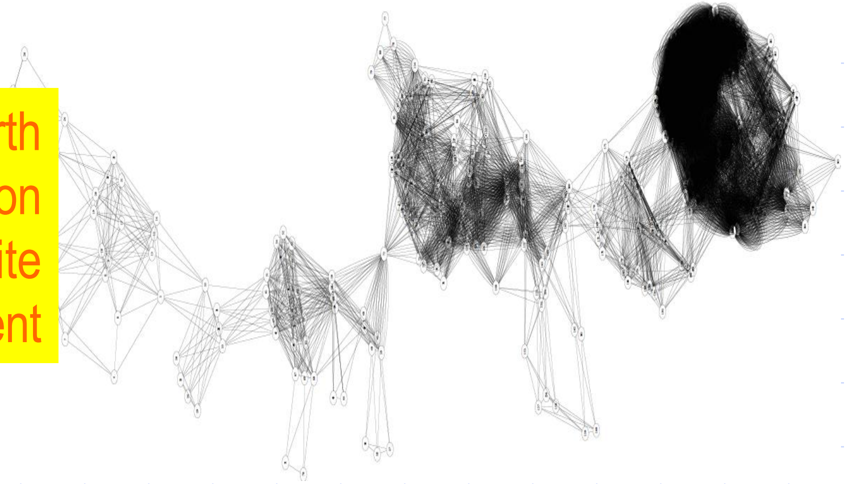
Many real applications have a structured network

Radio
Link
Frequency
Assignment



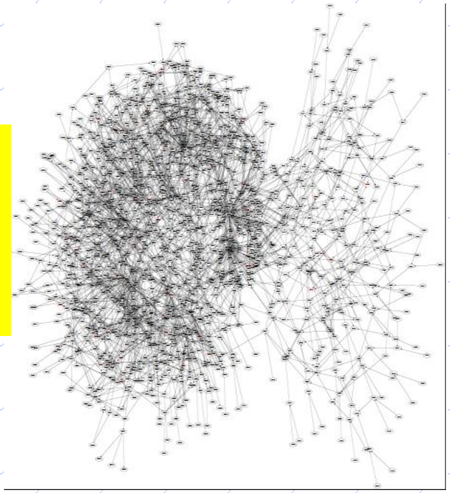
CELAR SCEN-07r
(*Constraints* 4(1), 1999)

Earth
Observation
Satellite
Management



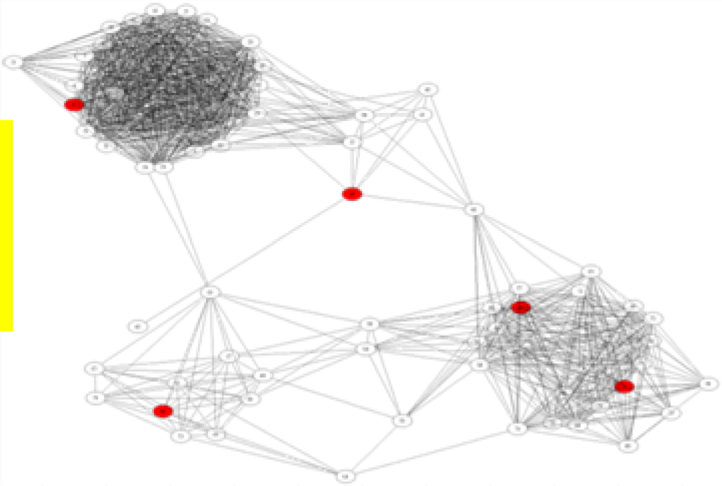
SPOT5 #509 (*Constraints* 4(3), 1999)

Mendelian
Error
Detection



langladeM7 sheep pedigree
(*Constraints* 13(1), 2008)

Tag
SNP
Selection



HapMap chr01 $r^2 \geq 0.8$ #14481
(*Bioinformatics* 22(2), 2006)

CELAR 6 results since 1993

n. of vars: n=100, domain size: d=44, n. of cost functions: e=1222

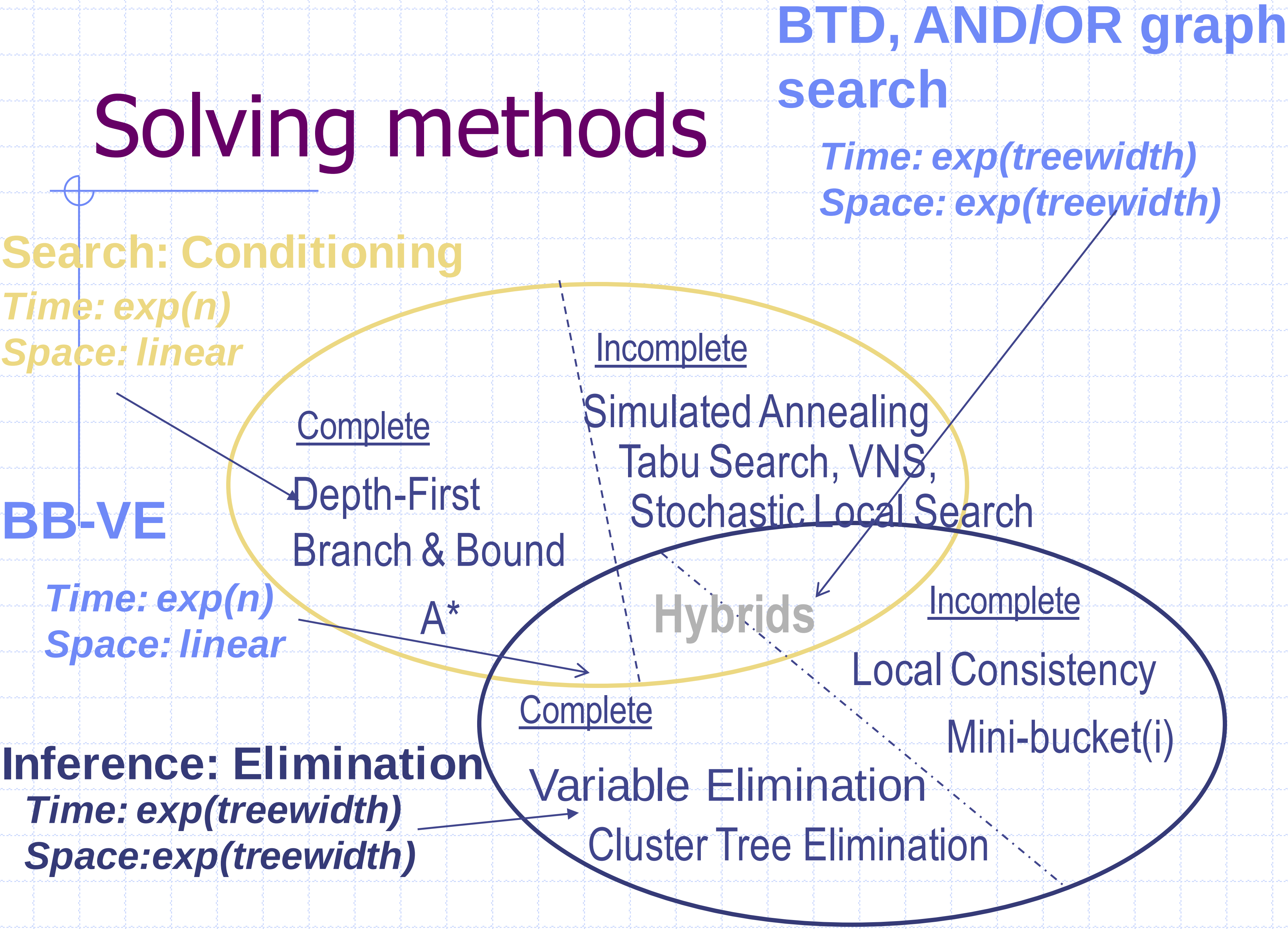
Time of optimality proof	Method(s) used	Publication
26 days (SUN UltraSparc 167 MHz)	Ad-hoc problem decomposition & Russian Doll Search *(22 vars only)*	(de Givry, Verfaillie, Schiex, CP 1997)
3 days (SUN Sparc 2)	Ad-hoc problem decomposition & PFC-MRDAC *(22 vars only)*	(Larrosa, Meseguer, Schiex, AIJ 1998)
8 hours (DEC Alpha 500MP)	Preprocessing rules & Cluster Tree Elimination	(Koster PhD thesis, 1999)
3 hours (PC 2.4 GHz)	B&B with EDAC & tree decomposition (BTD)	(de Givry, Schiex, Verfaillie, AAAI 2006)
1′ 26″ **25000x** (PC 2.5 GHz) **16 x**	BTD & variable ordering heuristisc &dicho branching	(Sanchez, Allouche, de Givry,Schiex, IJCAI 2009)

CELAR 7 (n=200) solved in 4.5 days (Sanchez et al, IJCAI 2009)

CELAR 8 (n=458) solved in 127 days (Allouche et al, CP 2010)

All CELAR and GRAPH instances are closed!

Solving methods



BTD, AND/OR graph search

Time: exp(treewidth)

Space: exp(treewidth)

Search: Conditioning

Time: exp(n)

Space: linear

Incomplete

Simulated Annealing

Tabu Search, VNS,

Stochastic Local Search

Complete

Depth-First

Branch & Bound

A*

BB-VE

Time: exp(n)

Space: linear

Hybrids

Incomplete

Local Consistency

Mini-bucket(i)

Complete

Variable Elimination

Cluster Tree Elimination

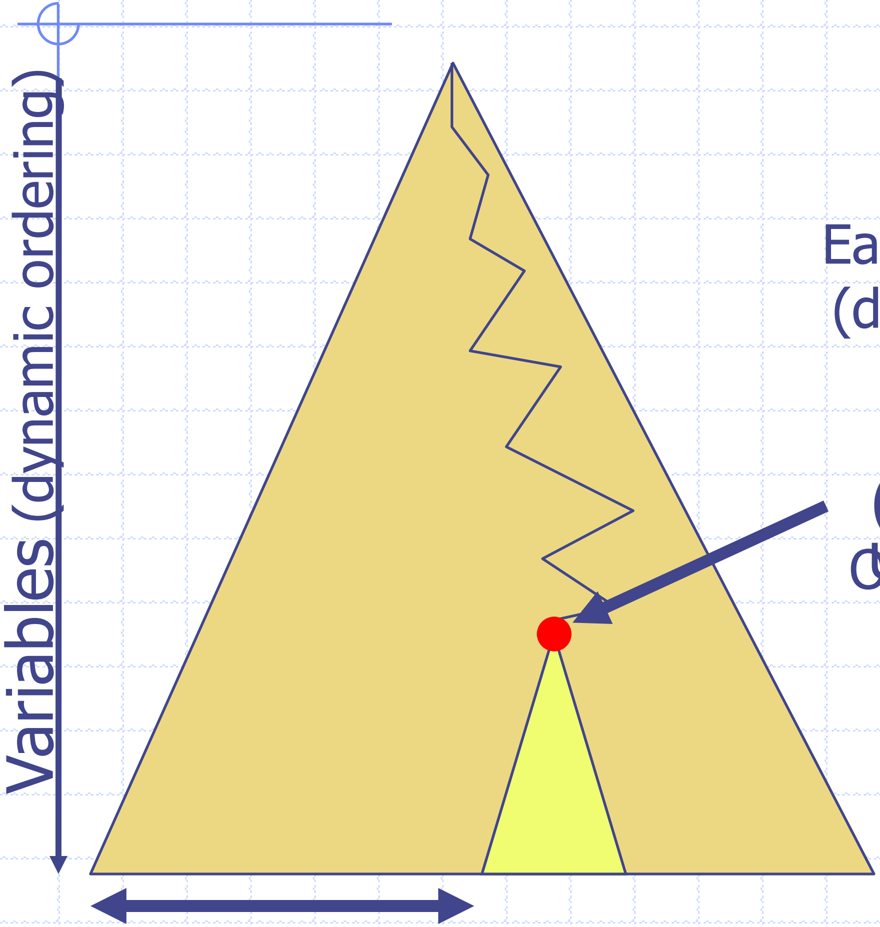
Inference: Elimination

Time: exp(treewidth)

Space:exp(treewidth)

(slide from IJCAI'09 tutorial)

Depth-First Branch and Bound (DFBB)



Each node is a WCSP subproblem
(defined by current conditioning)

(LB) Lower Bound = $f_\emptyset$

under estimation of the
best solution
in the sub-tree

Obtained by enforcing local consistency

If LB ≥ *k* then prune

If $f_\emptyset$ ≥ *k* then prune

(UB) Upper Bound
= best solution found so far = ***k***

Combination of cost functions

A	B	f(A,B)
b	b	6
b	g	0
g	b	0
g	g	6

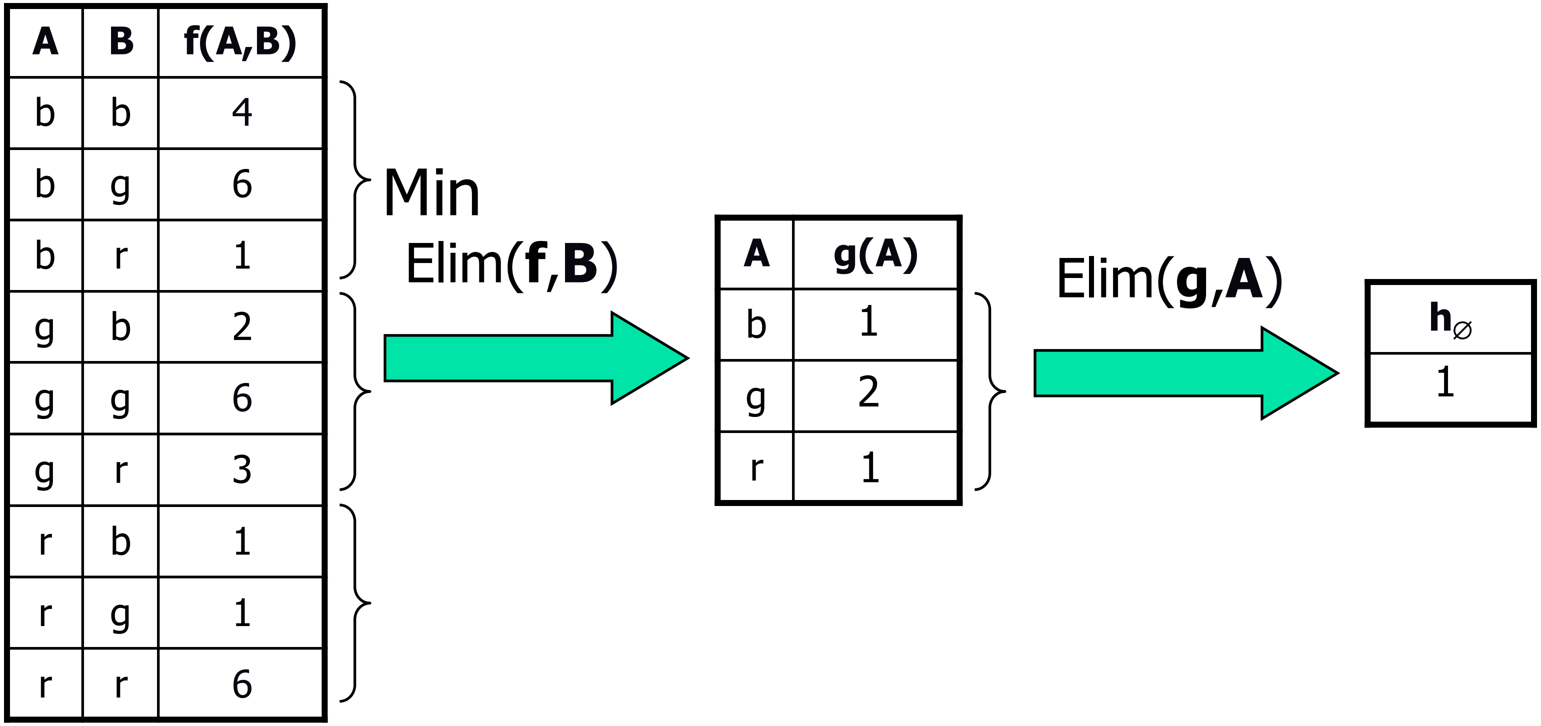
+

B	C	f(B,C)
b	b	6
b	g	0
g	b	0
g	g	6

A	B	C	f(A,B,C)
b	b	b	12
b	b	g	6
b	g	b	0
b	g	g	6
g	b	b	6
g	b	g	0
g	g	b	6
g	g	g	12

= 0 + 6

Elimination in a cost function



A	**B**	**f(A,B)**
b	b	4
b	g	6
b	r	1
g	b	2
g	g	6
g	r	3
r	b	1
r	g	1
r	r	6

Min

Elim(**f**,**B**)

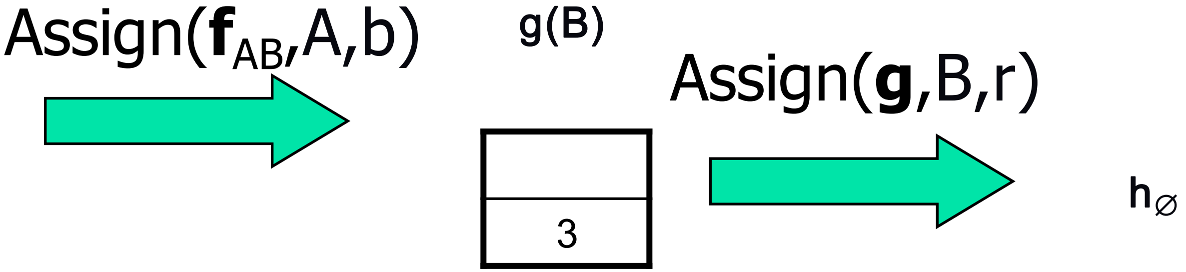
A	**g(A)**
b	1
g	2
r	1

Elim(**g**,**A**)

$\mathbf{h}_\varnothing$
1

Conditioning a cost function

A	B	f(A,B)
b	b	6
b	g	0
b	r	3
g	b	0
g	g	6
g	r	0
r	b	0
r	g	0
r	r	6



Assign($\mathbf{f}_{AB}$,A,b)

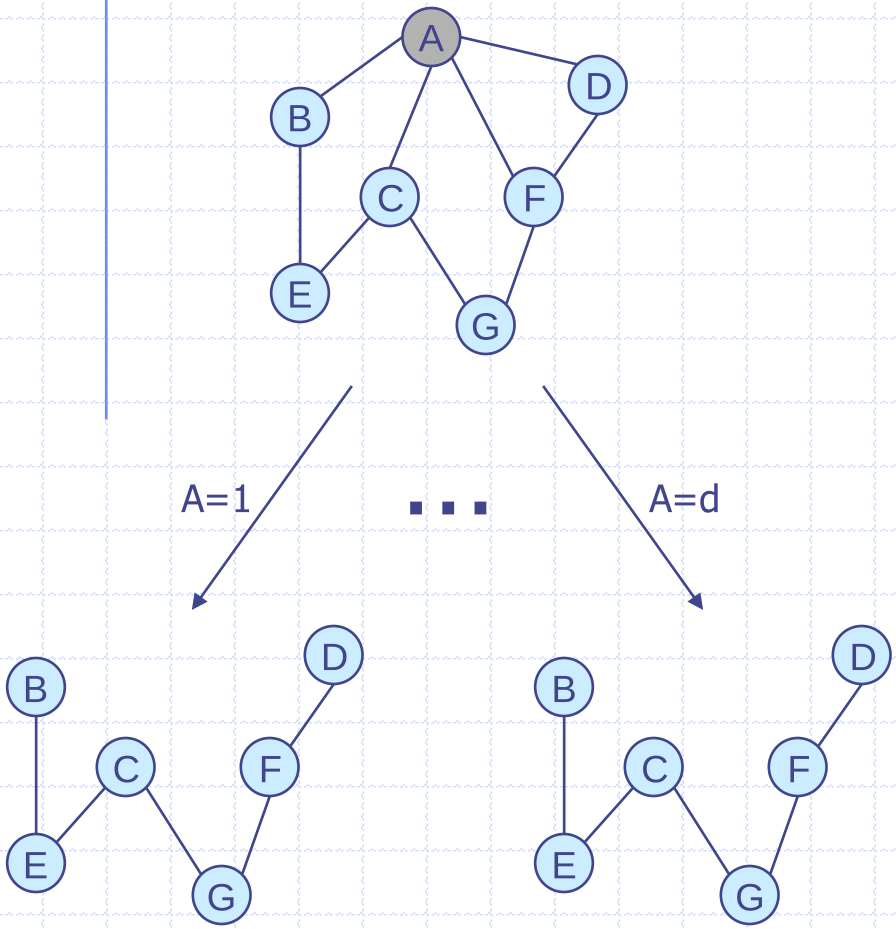
g(B)
3

Assign(**g**,B,r)

$\mathbf{h}_{\emptyset}$

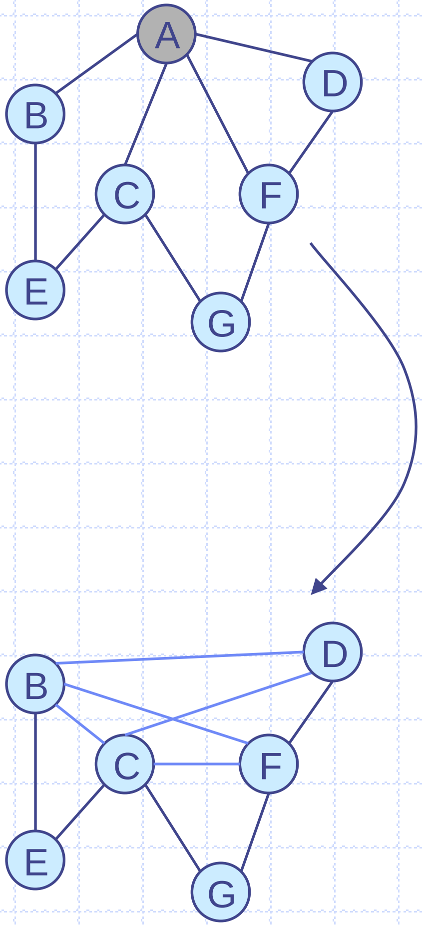
Conditioning vs. Elimination

Conditioning (search)



d “sparser” problems

Elimination (inference)



1 “denser” problem

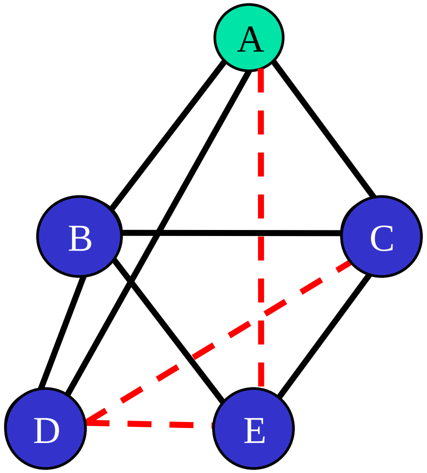
(slide from IJCAI’09 tutorial)

Variable elimination (aka bucket elimination)

- Eliminate Variables one by one.
- When all variables have been eliminated, the **problem is solved**
- Optimal solutions of the original problem can be recomputed

- Complexity: exponential in the *induced width* (maximum degree found during the elimination process)
- Minimizing induced width is NP-hard.

Computing the Optimal Cost Solution



Constraint graph

$$\textbf{OPT} = \min_{e=0,d,c,b} \underbrace{f(a,b)}+f(a,c)+f(a,d)+f(\underbrace{b,c)+f(b,d)+f(b,e)}+f(c,e)$$

Combination

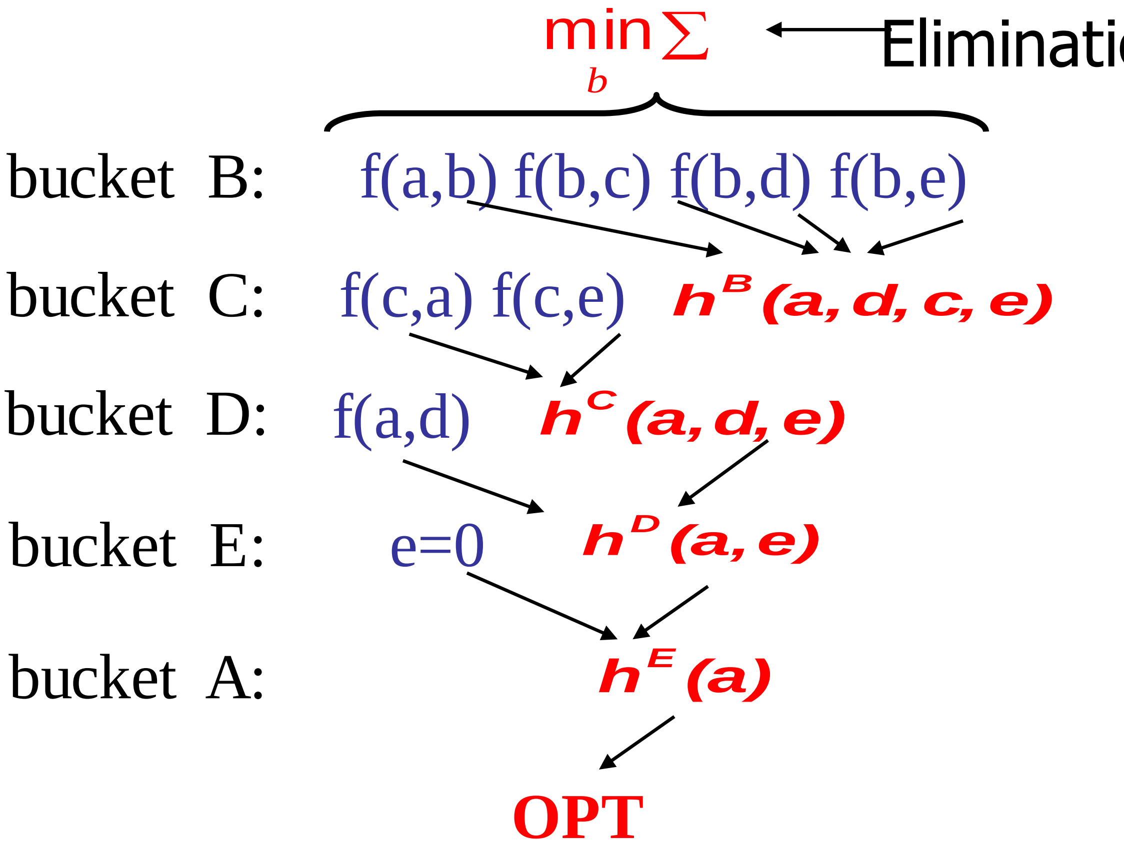
$$\min_{e=0}\min_{d} f(a,d) + \min_{c} f(a,c)+f(c,e) + \min_{b} \underbrace{f(a,b)+f(b,c)+f(b,d)+f(b,e)}_{h^{B}(a,d,c,e)}$$

Variable Elimination

Finding $OPT = \min_{X_1,\ldots,X_n} \sum_{j=1}^{r} f_j(X)$

Algorithm **elim-opt** (Dechter, 1996)
Non-serial Dynamic Programming (Bertele & Briochi, 1973

$$OPT = \min_{a,e,d,c,b} F(a,b) + F(a,c) + F(a,d) + F(b,c) + F(b,d) + F(b,e) + F(c,e)$$



$\overbrace{}^{\min_b \sum}$ ← Elimination operator

bucket B: $f(a,b)$ $f(b,c)$ $f(b,d)$ $f(b,e)$

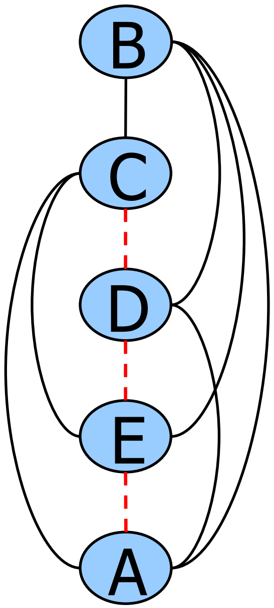
bucket C: $f(c,a)$ $f(c,e)$ $h^B(a,d,c,e)$

bucket D: $f(a,d)$ $h^C(a,d,e)$

bucket E: $e=0$ $h^D(a,e)$

bucket A: $h^E(a)$

OPT



Generating the Optimal Assignment

$$5.\ \ b' = \arg\min_{b} f(a',b) + f(b,c') + f(b,d') + f(b,e')$$

$$4.\ \ c' = \arg\min_{c} f(c,a') + f(c,e') + h^{B}(a',d',c,e')$$

$$3.\ \ d' = \arg\min_{d} f(a',d) + h^{C}(a',d,e')$$

$$2.\ \ e' = 0$$

$$1.\ \ a' = \arg\min_{a} h^{E}(a)$$

B: $f(a,b)\ f(b,c)\ f(b,d)\ f(b,e)$

C: $f(c,a)\ f(c,e)$ $\quad h^{B}(a,d,c,e)$

D: $f(a,d)$ $\quad h^{C}(a,d,e)$

E: $e=0$ $\quad h^{D}(a,e)$

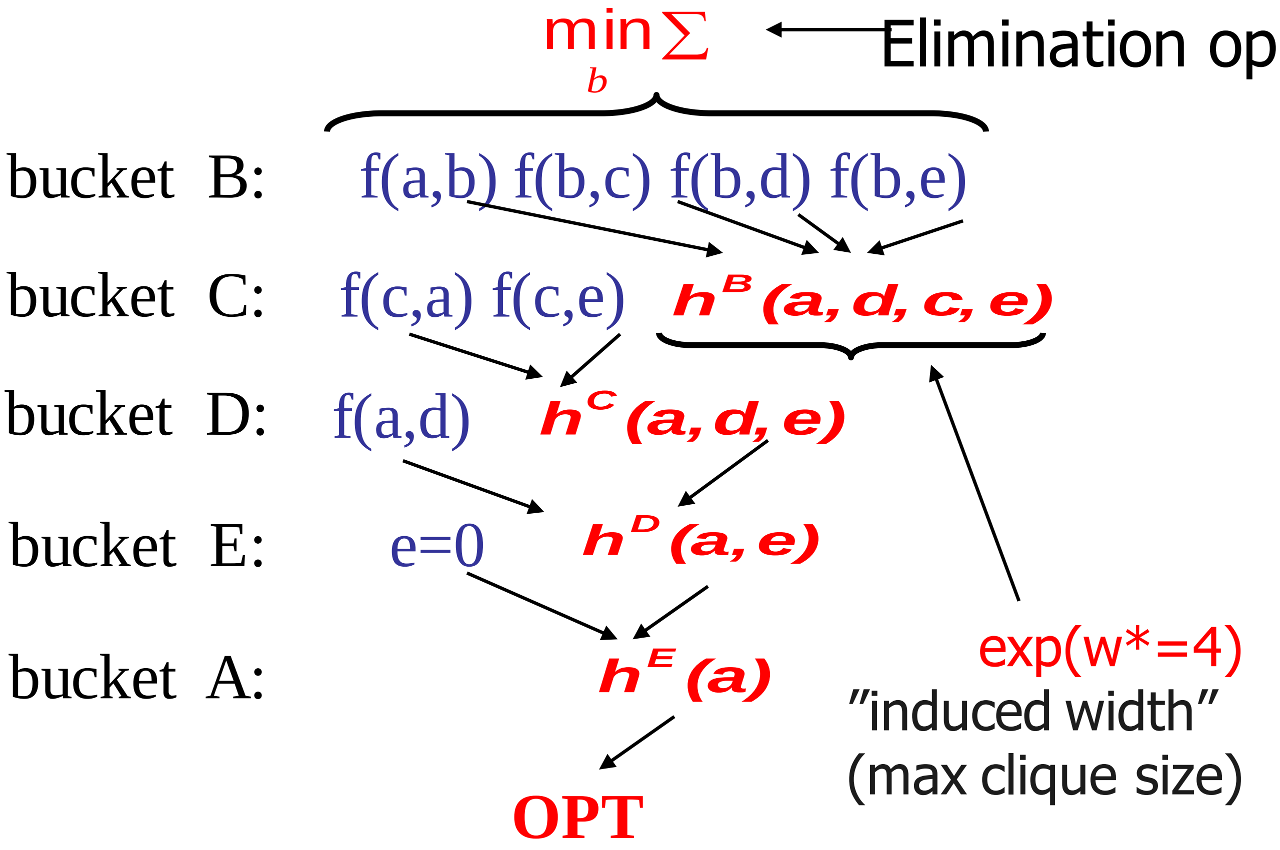
A: $\quad h^{E}(a)$

Return (a', b', c', d', e')

Complexity

Algorithm **elim-opt** (Dechter, 1996)
Non-serial Dynamic Programming (Bertele & Briochi, 1973

$$OPT = \min_{a,e,d,c,b} F(a,b) + F(a,c) + F(a,d) + F(b,c) + F(b,d) + F(b,e) + F(c,e)$$



$\min_{b} \sum$ ← Elimination operator

bucket B: $\underbrace{f(a,b)\ f(b,c)\ f(b,d)\ f(b,e)}$

bucket C: f(c,a) f(c,e) $\underbrace{h^B(a,d,c,e)}$

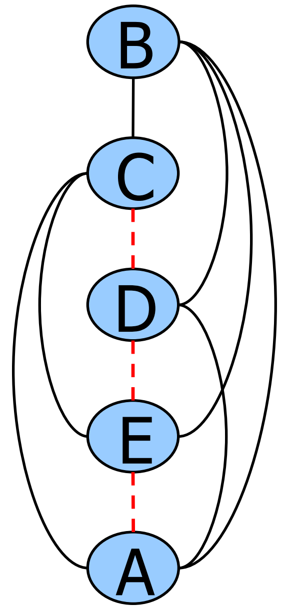
bucket D: f(a,d) $h^C(a,d,e)$

bucket E: e=0 $h^D(a,e)$

bucket A: $h^E(a)$

OPT

exp(w*=4)
"induced width"
(max clique size)



Complexity of Bucket Elimination

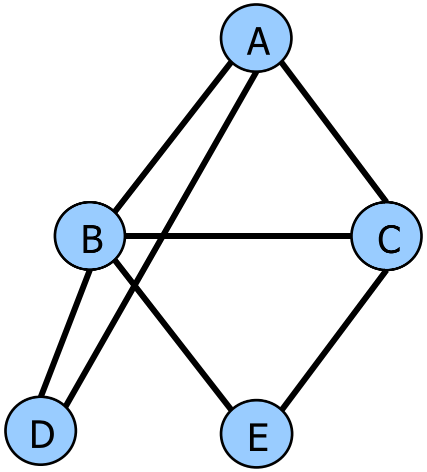
Bucket-Elimination is **time** and **space**

r = number of functions

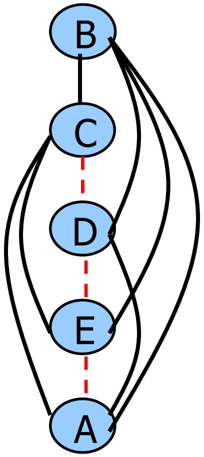
$$O(r \exp(w^*(d))$$

$w^*(d)$ – the induced width of the primal graph along ordering d

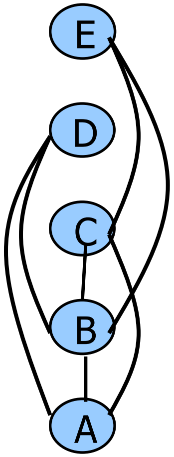
The effect of the ordering:



constraint graph



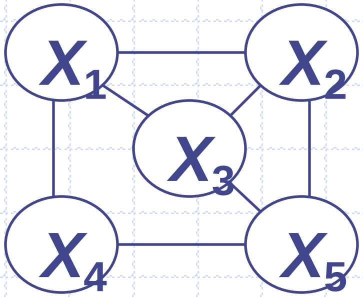
$w^*(d_1) = 4$



$w^*(d_2) = 2$

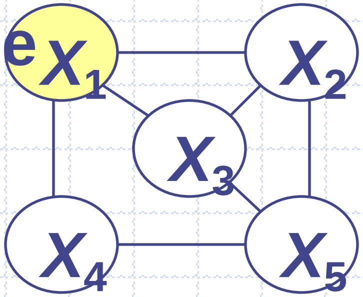
Finding smallest induced-width is hard!

Search Basic Step: Conditioning



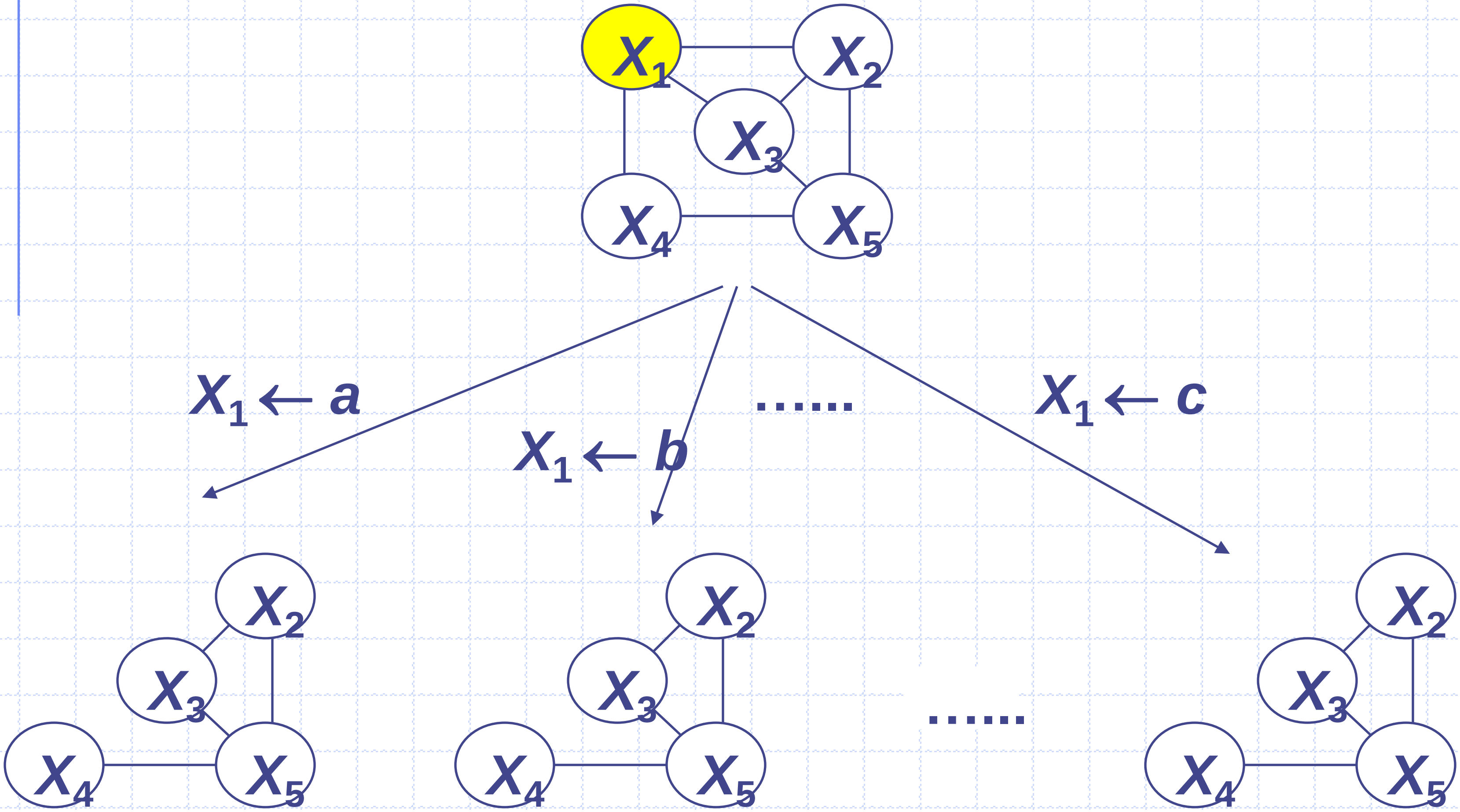
Search Basic Step: Conditioning

- **Select a variable**



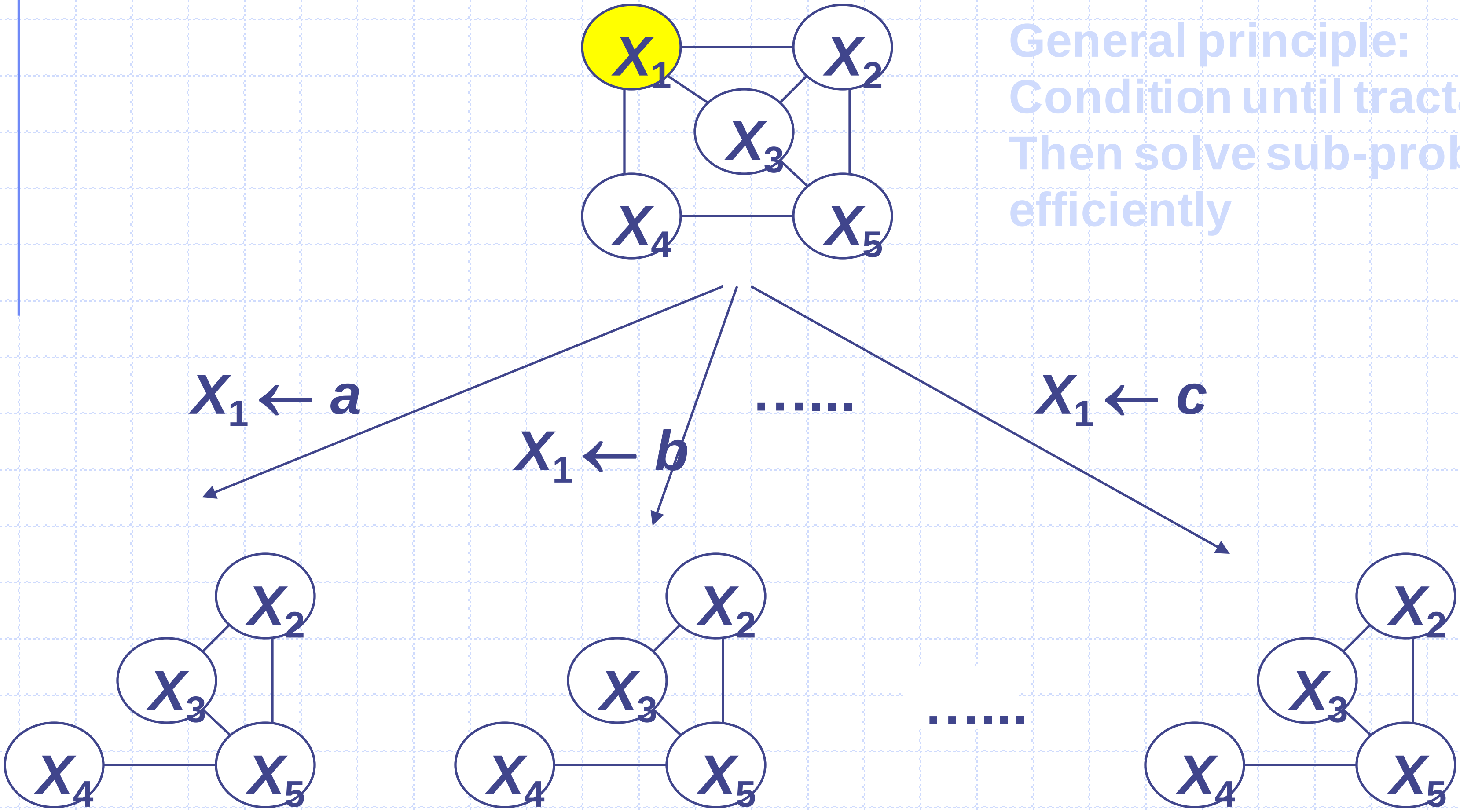
(slide from IJCAI'09 tutorial)

Search Basic Step: Conditioning



(slide from IJCAI'09 tutorial)

Search Basic Step:
Variable Branching by Conditioning

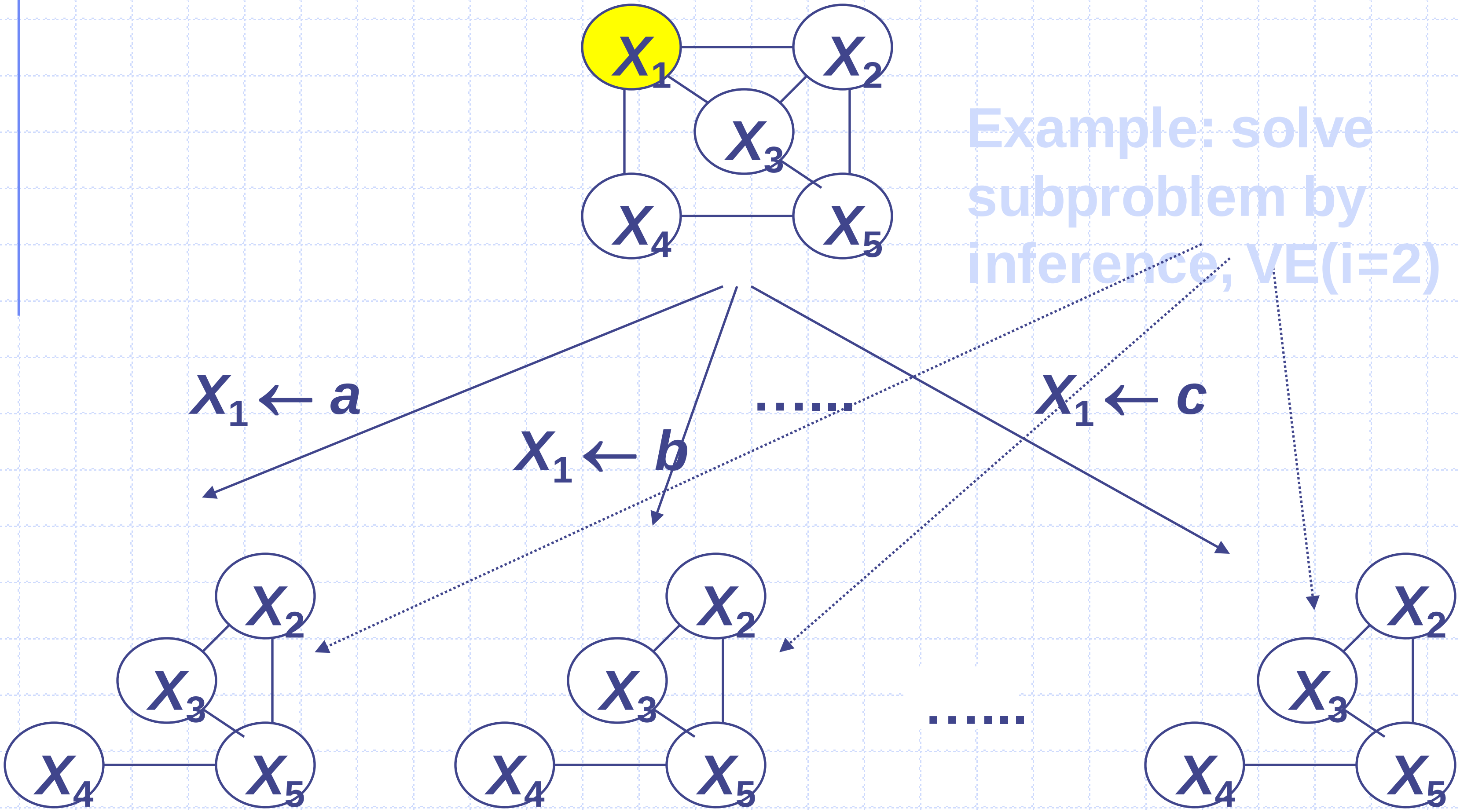


General principle:
Condition until tractable
Then solve sub-problems efficiently

(slide from IJCAI'09 tutorial)

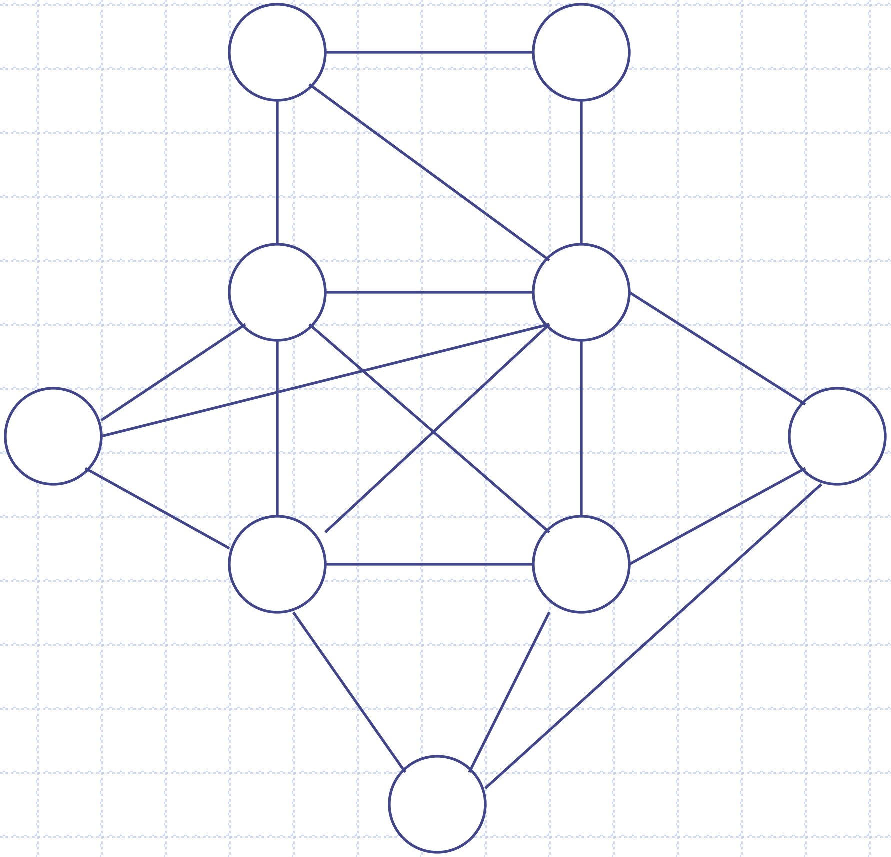
Search Basic Step:

Variable Branching by Conditioning

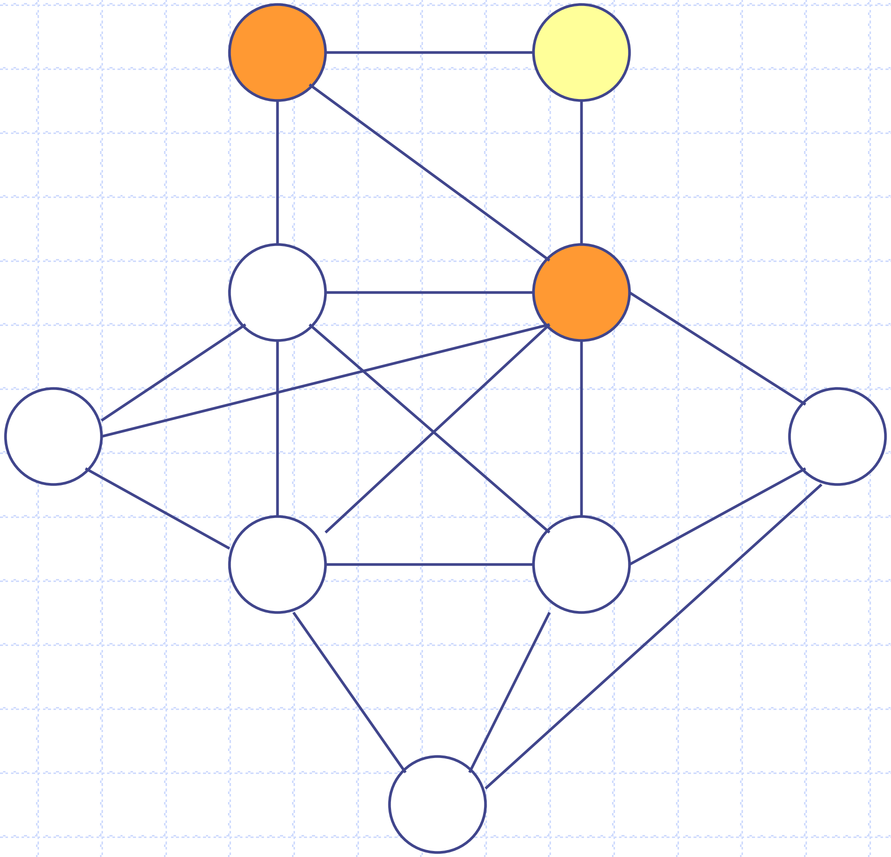


(slide from IJCAI'09 tutorial)

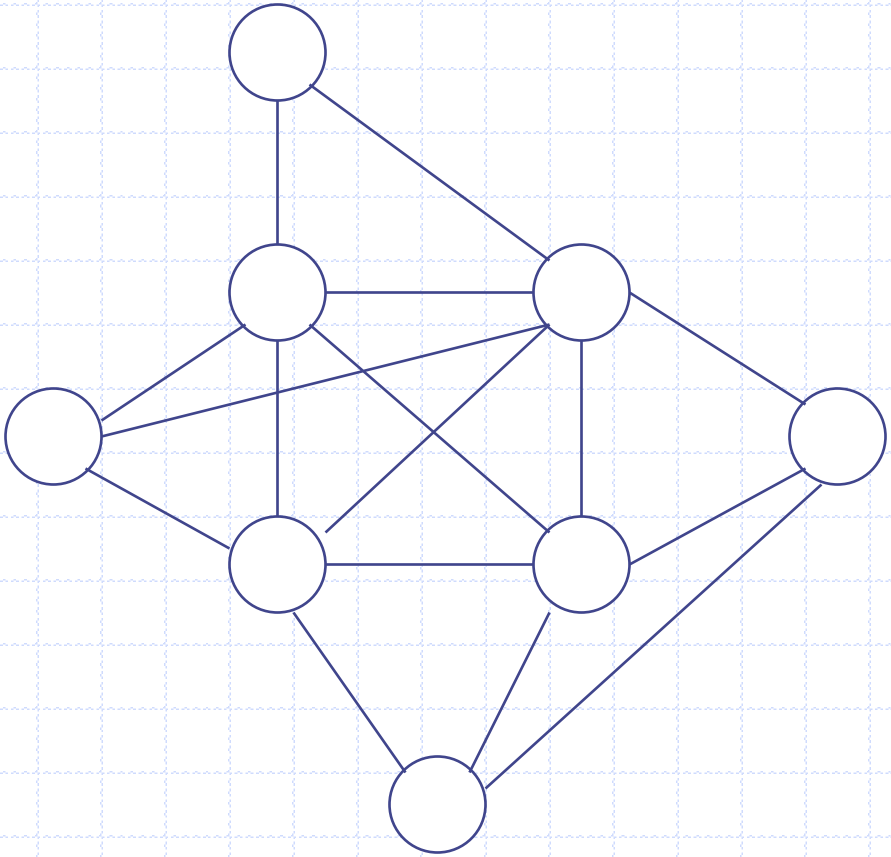
Eliminate First



Eliminate First

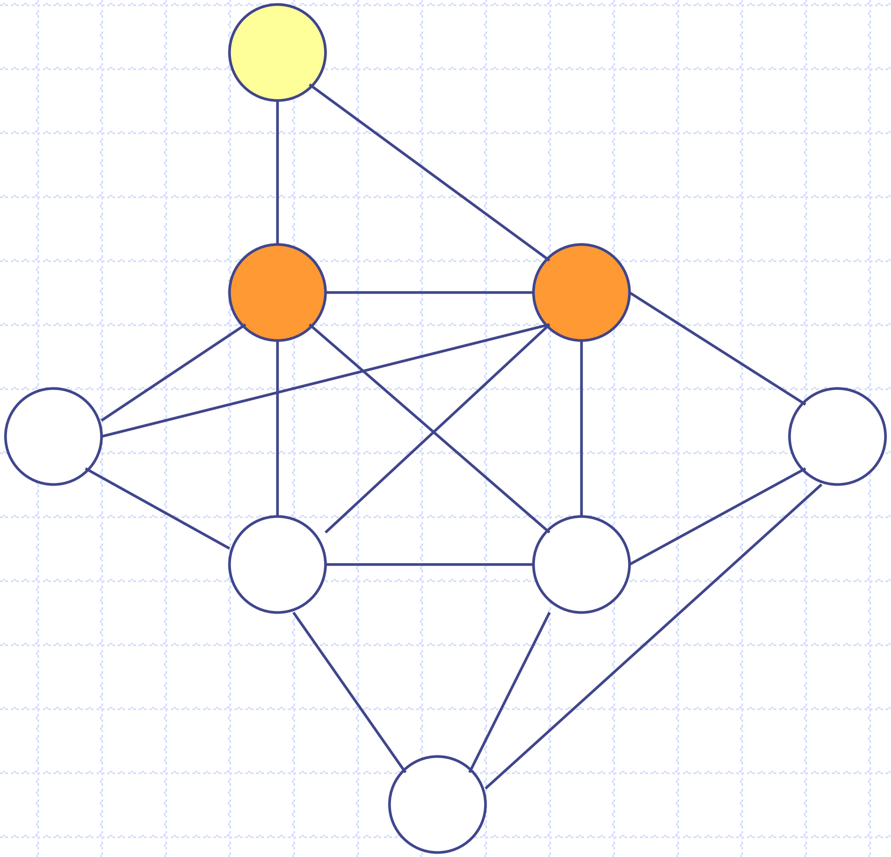


Eliminate First



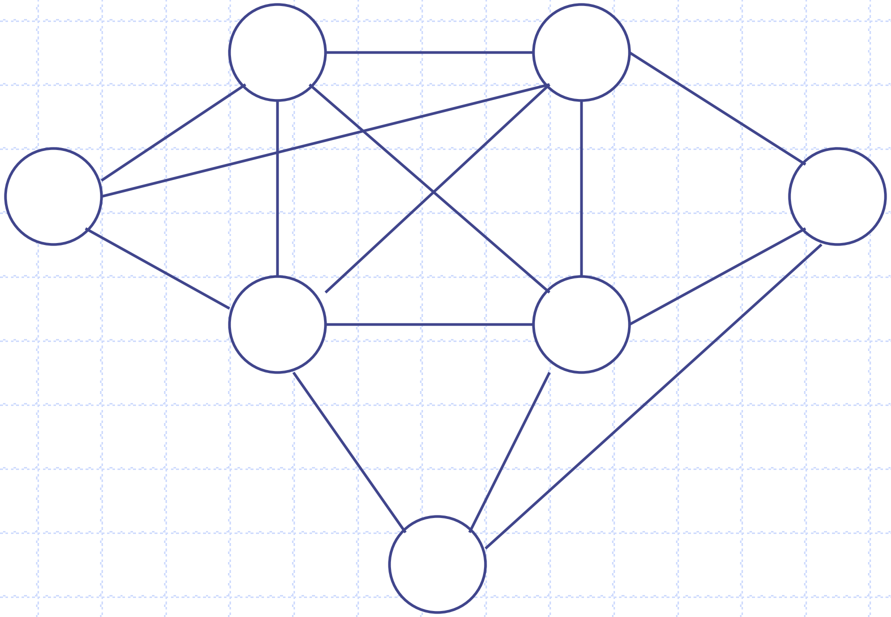
(slide from IJCAI’09 tutorial)

Interleaving Conditioning and Elimination



Interleaving Conditioning and Elimination

Solve the rest of the problem by any means



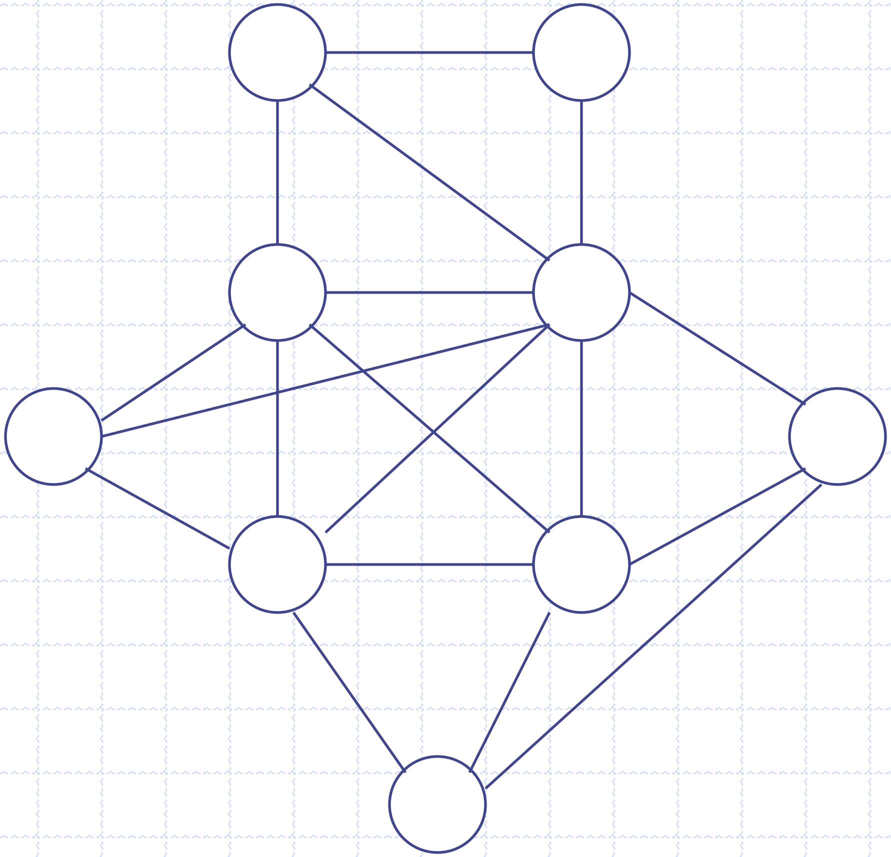
(slide from IJCAI'09 tutorial)

First hybrids: Search & Variable Elimination

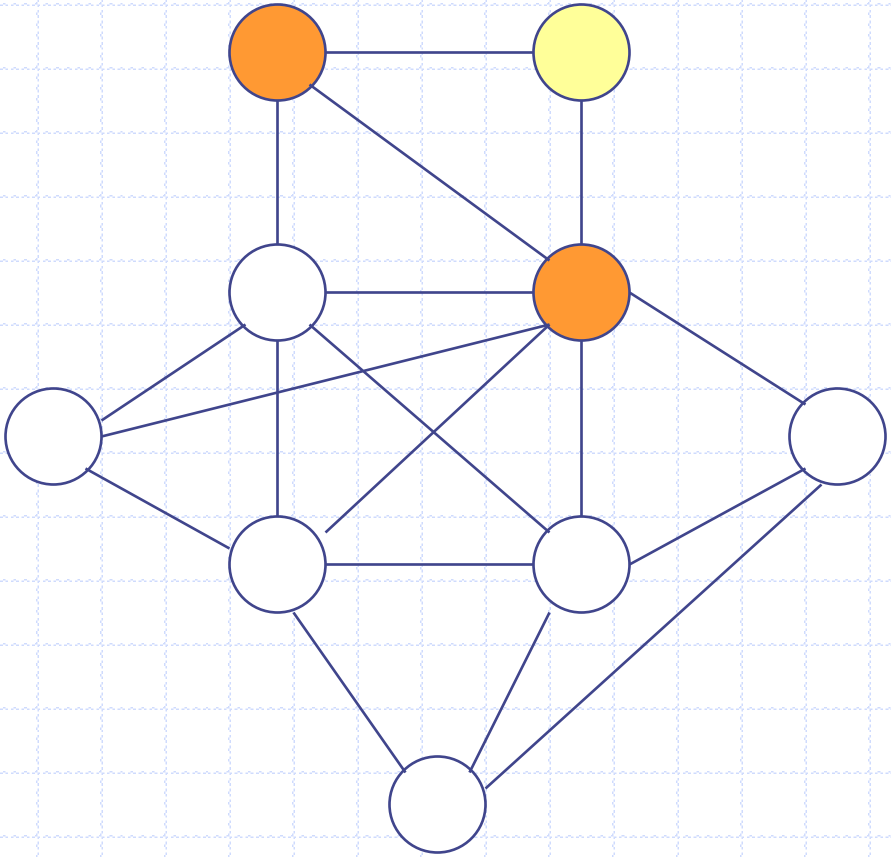
- Condition, condition, condition … and then only eliminate (*Cycle-Cutset*)
- Eliminate, eliminate, eliminate … and then only search
- Interleave conditioning and elimination

(slide from IJCAI'09 tutorial)

Interleaving Conditioning and Elimination
BB-VE(2) (Larrosa & Dechter, CP 2002)

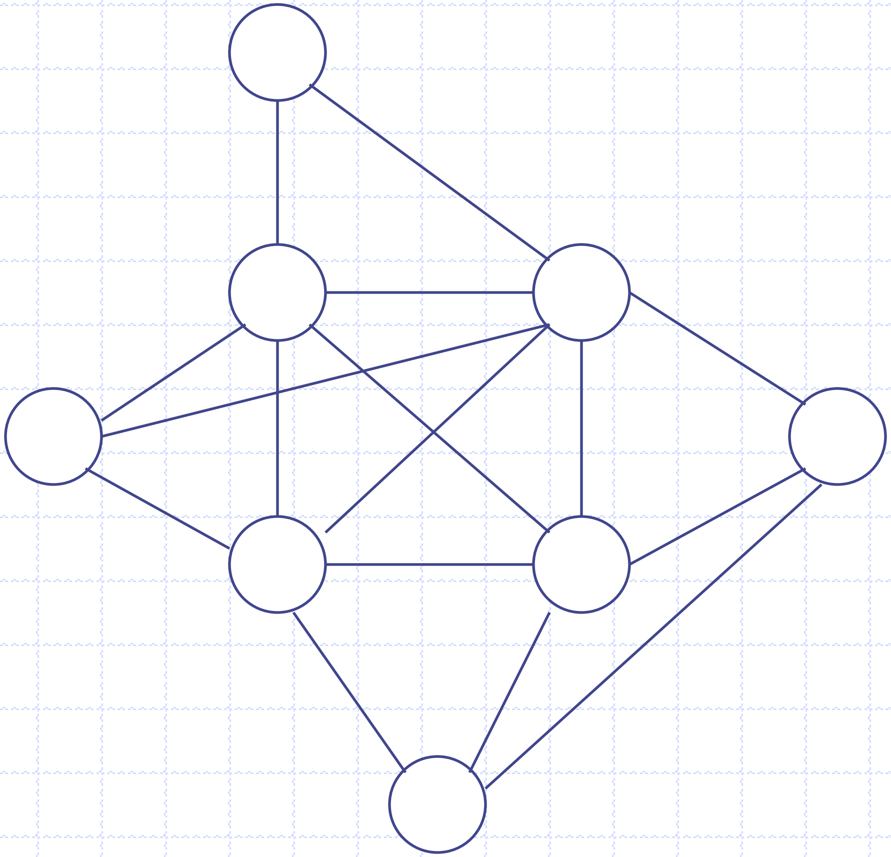


Interleaving Conditioning and Elimination
BB-VE(2)



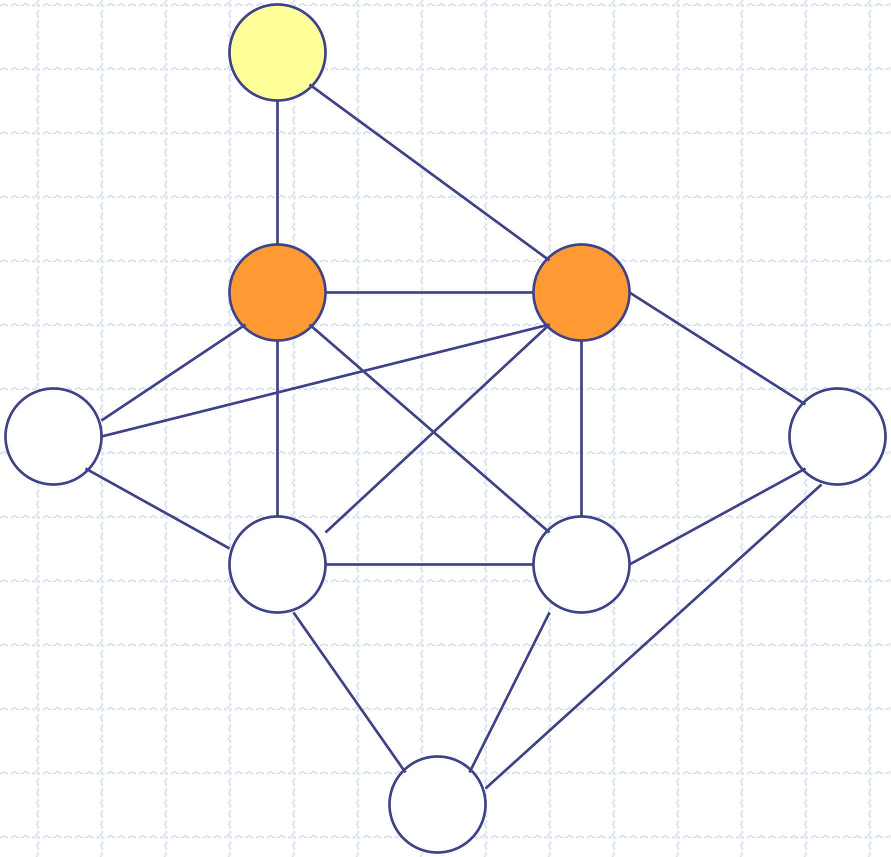
(slide from IJCAI'09 tutorial)

Interleaving Conditioning and Elimination BB-VE(2)

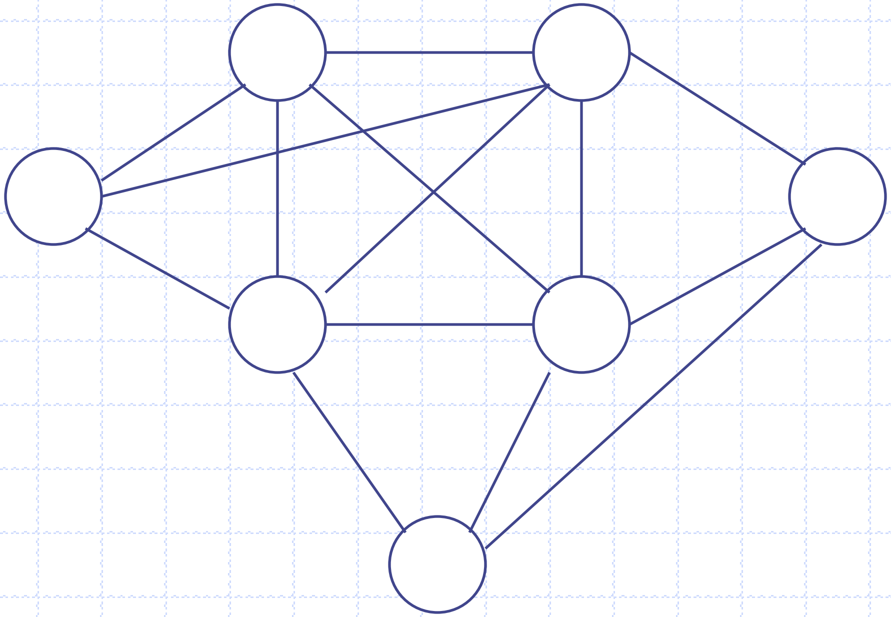


(slide from IJCAI'09 tutorial)

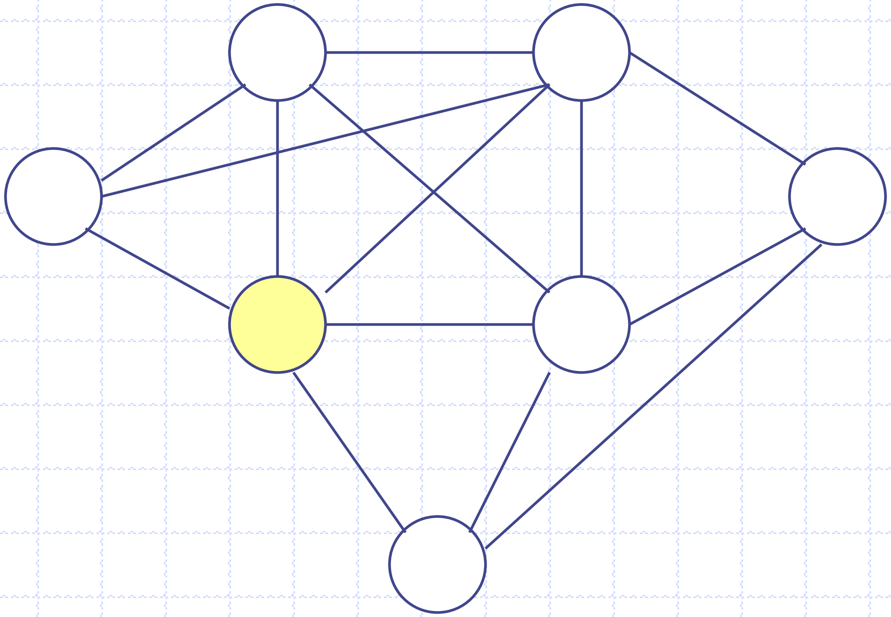
Interleaving Conditioning and Elimination BB-VE(2)



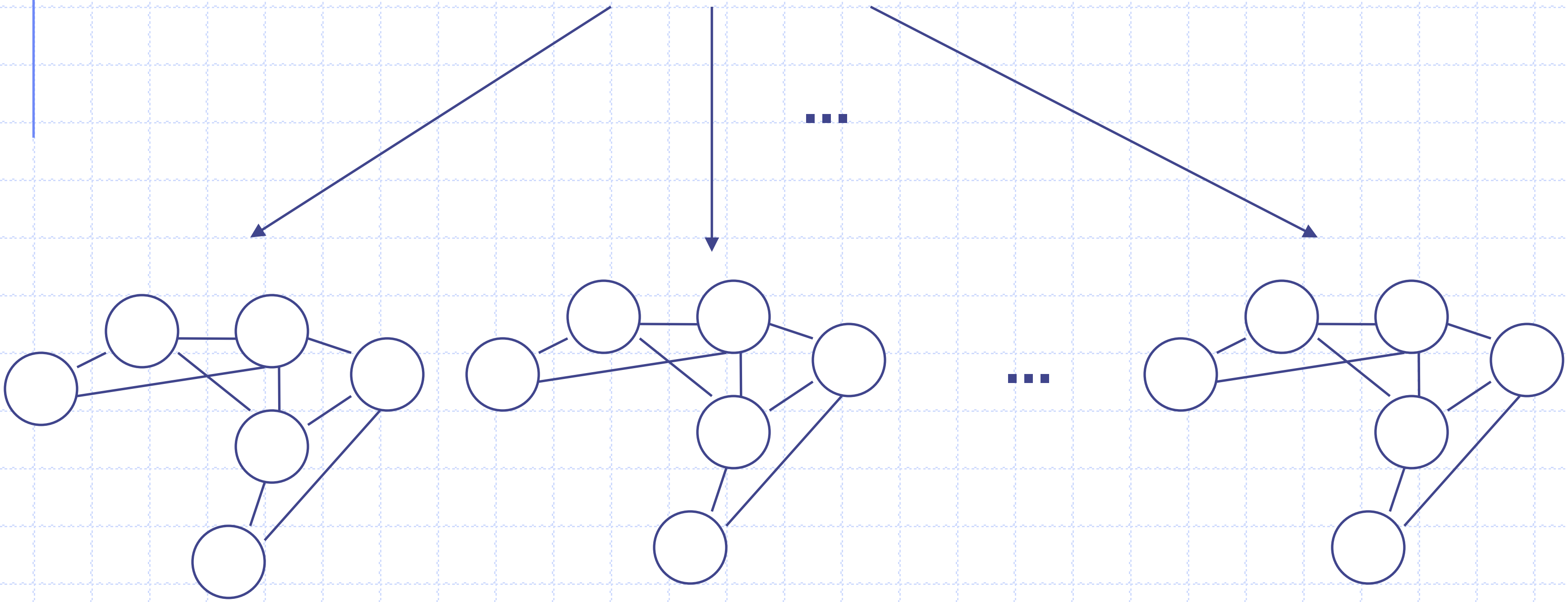
Interleaving Conditioning and Elimination BB-VE(2)



Interleaving Conditioning and Elimination BB-VE(2)

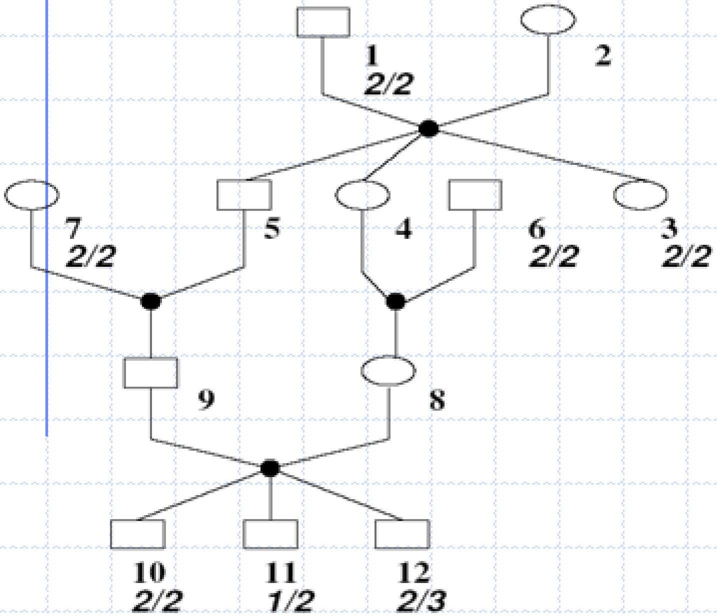


Interleaving Conditioning and Elimination BB-VE(2)



(Sanchez et al, *Constraints* 2008)

Mendelian error detection



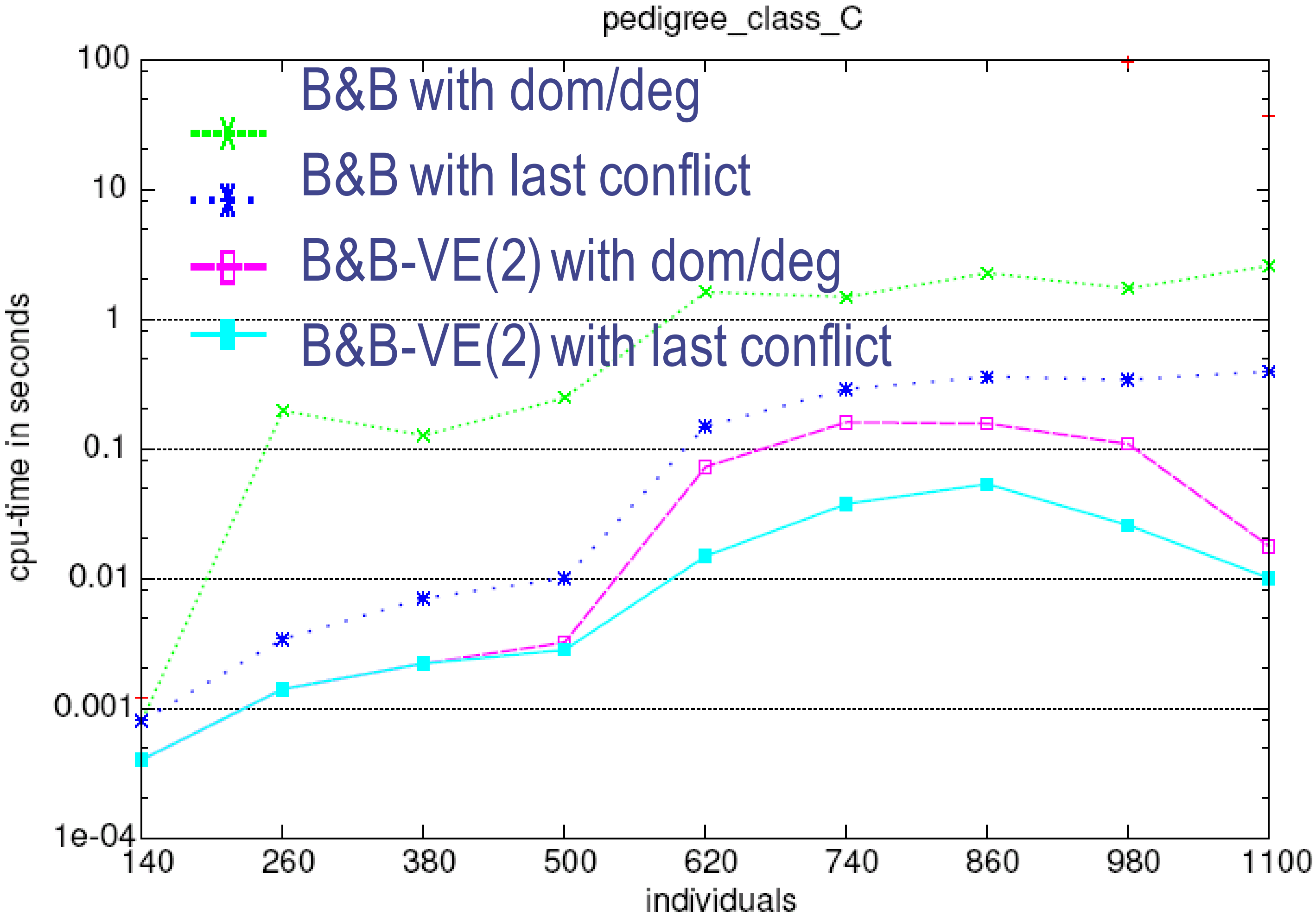
- Given a pedigree and partial observations (genotypings)
- Find the erroneous genotypings, such that their removal restores consistency

- Checking consistency is NP-complete (Aceto et al., *Comp. Sci. Tech.* 2004)
- Minimize the number of genotypings to be removed
- Maximize the joint probability of the true genotypes (MPE)

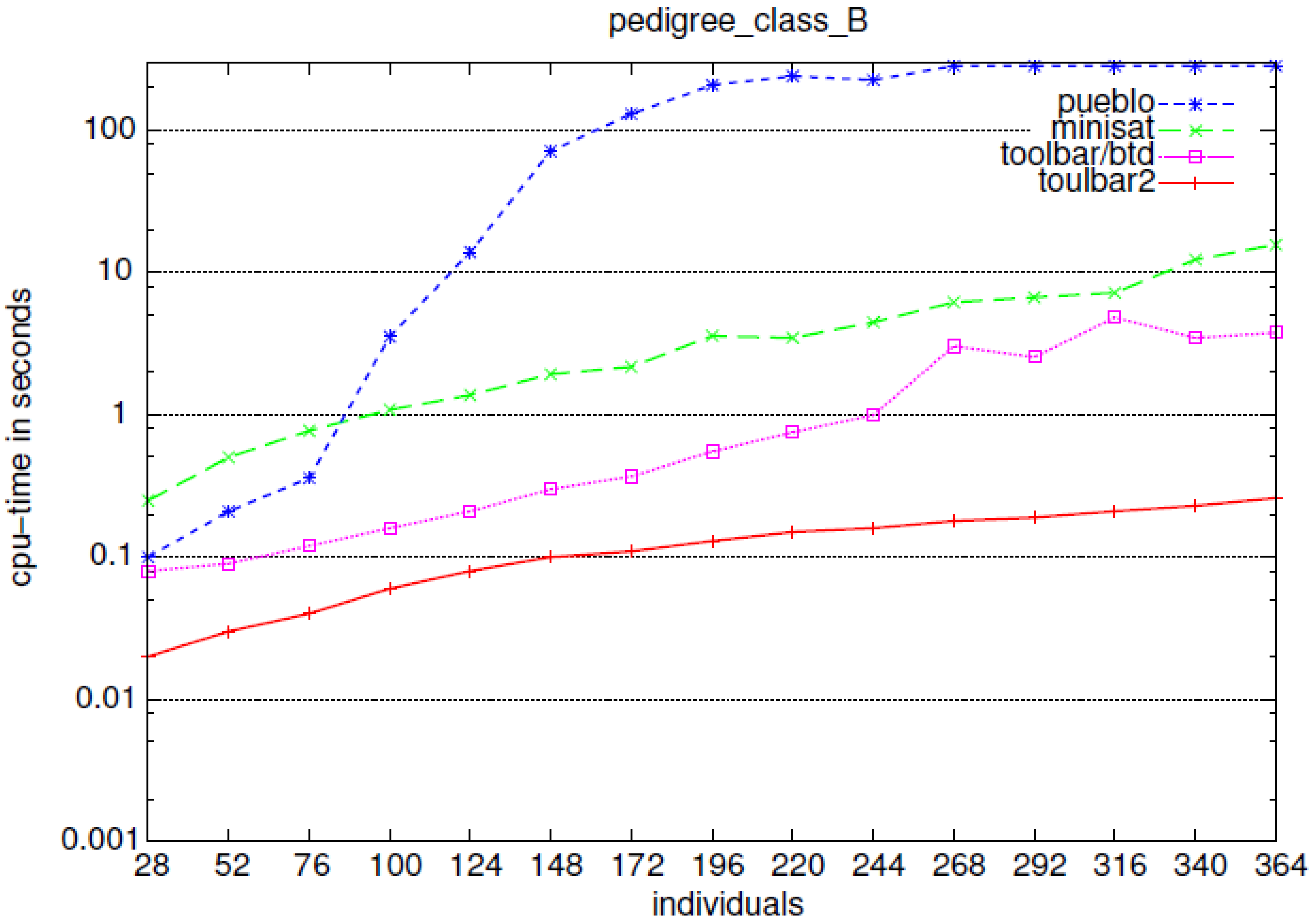
Pedigree *problem size:*
n≤20,000 ; d=3—66 ; e(3)≤30,000

Pedigree

- toulbar2 v0.5 with EDAC3 and binary branching
- Minimize the number of genotypings to be removed
- CPU time in seconds to find and prove optimality on a linux PC 3 GHz with 16 GB



Pedigree


pedigree_class_B
cpu–time in seconds
individuals
100
10
1
0.1
0.01
0.001
28
52
76
100
124
148
172
196
220
244
268
292
316
340
364
pueblo
minisat
toolbar/btd
toulbar2

Pedigree

- toulbar2 with last conflict
- Minimize the number of genotypings to be removed
- CPU time in seconds to find and prove optimality

								B&B-VE(2)		
	ind	*vars*	*genotyped*	alleles	*nf*	*ngen*	treewidth ub	errors	time	nodes
eye	36	36	28	6	11	4	2	1	0.02	0
cancer	49	48	37	8	18	5	2	1	0.21	0
parkinson	37	34	13	4	7	7	5	0	0	6
berrichon$_{1nc}$	129516	9947	2448	4	8821	17	262	2	4.73	8805
berrichon$_{1}$	129516	10017	2483	4	8786	17	330	23	5.81	8384
berrichon$_{2nc}$	27255	19337	10215	4	4719	19	-	41	5.89	6170
berrichon$_{2}$	27255	19562	10215	4	2381	19	-	106	17.23	15445
langlade$_{1}$	1355	1209	711	9	298	13	84	38	12.28	391
langlade$_{2}$	1355	1223	715	7	298	13	82	89	60.56	17857
langlade$_{3}$	1355	1258	787	5	298	13	85	39	14.19	6731
langlade$_{4}$	1355	1186	672	8	298	13	83	43	59.7	3520
moissac$_{1}$	283	260	183	2	81	5	6	0	0	5
moissac$_{2}$	283	244	167	7	81	5	6	0	0.51	6
moissac$_{3}$	283	225	151	3	81	5	6	0	0	4
moissac$_{4}$	283	256	179	2	81	5	6	0	0	5
moissac$_{5}$	283	237	161	8	81	5	6	0	1.02	5
moissac$_{6}$	283	201	131	11	81	5	5	0	5.64	6

Solving methods

BTD, AND/OR graph search

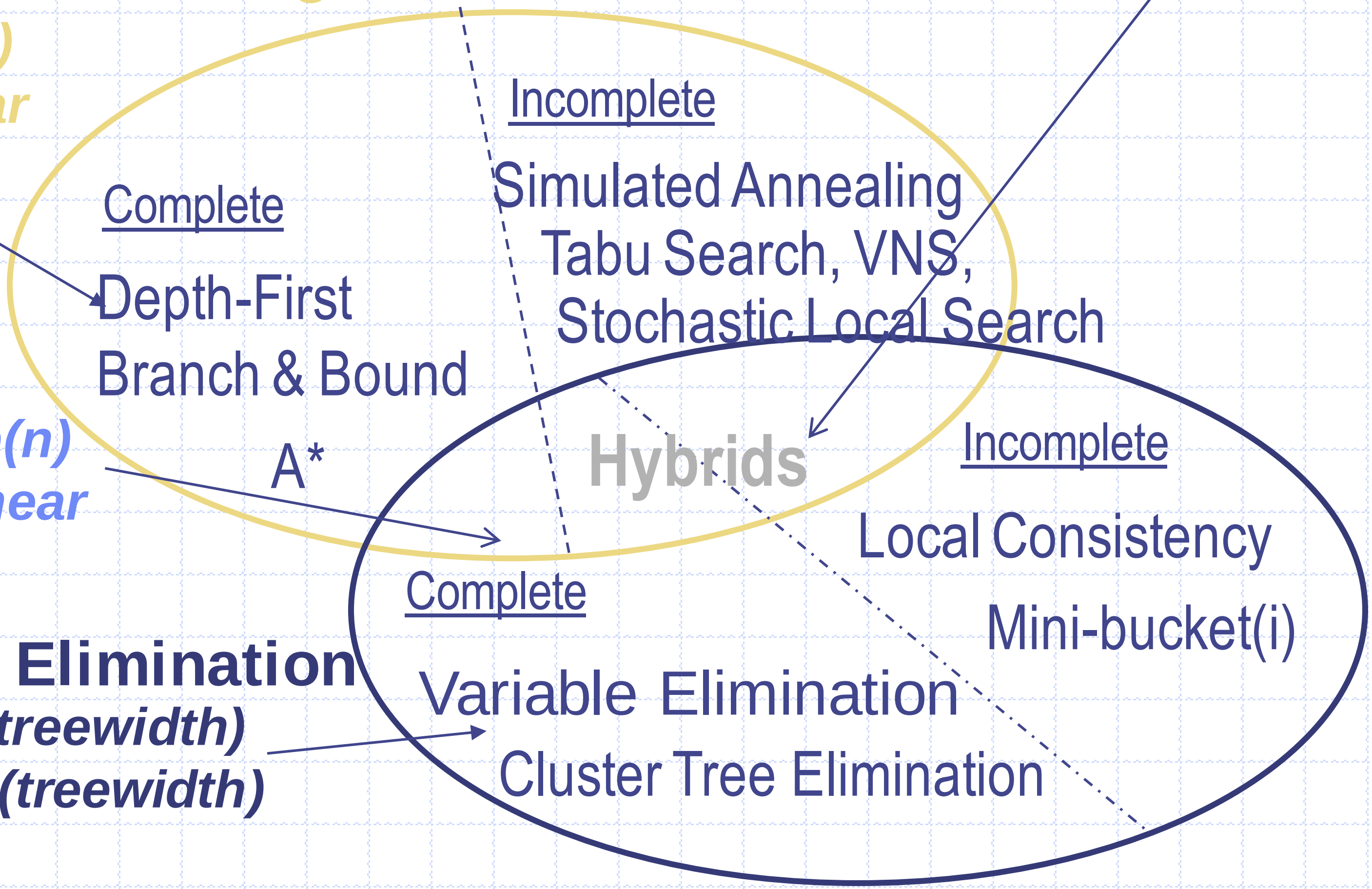
Time: exp(treewidth)

Space: exp(treewidth)

Search: Conditioning

Time: exp(n)

Space: linear



Complete

Depth-First

Branch & Bound

A*

Incomplete

Simulated Annealing

Tabu Search, VNS,

Stochastic Local Search

Hybrids

Incomplete

Local Consistency

Mini-bucket(i)

Complete

Variable Elimination

Cluster Tree Elimination

BB-VE

Time: exp(n)

Space: linear

Inference: Elimination

Time: exp(treewidth)

Space:exp(treewidth)

(slide from IJCAI'09 tutorial)

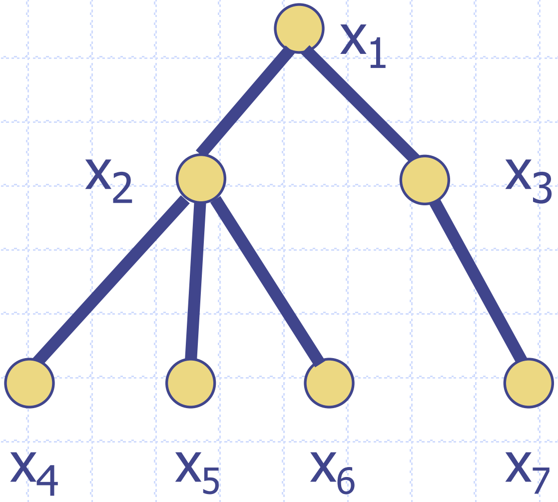
Second hybrids: Search & Cluster Tree Elimination

- Depth-First Branch and Bound exploiting a tree decomposition with:
  - A restricted variable ordering
  - Graph-based backjumping
  - Graph-based learning

  ⇒ Lazy elimination of subproblems using search

Structural tractability

Tree-structured binary WCSPs are tractable



Time complexity $O(e\ d^2)$
Space complexity $O(n\ d)$

n: number of variables
d: maximum domain size
e: number of cost functions

Proceed from the leaf nodes to a chosen root node

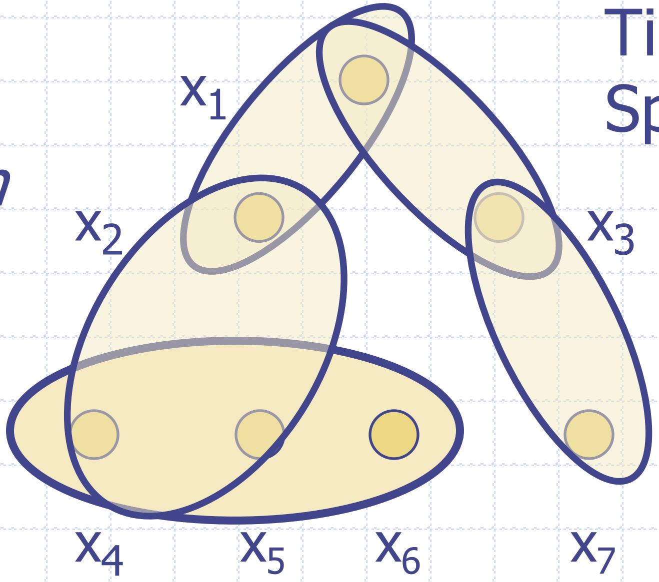
Project out leaf nodes by minimising over possible assignments

e.g. add $PROJ_{x2}(f_{x4} + f_{x2,x4})$ to f_{x2}

Structural tractability

- Acyclic r-ary WCSPs are tractable

Each cost function is represented by an hyperedge



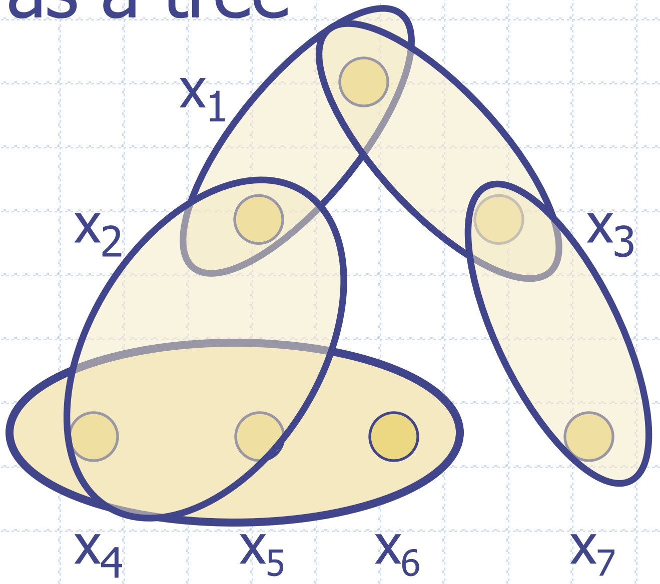
Time complexity in $O(e\ d^r)$
Space complexity in $O(n\ d^s)$

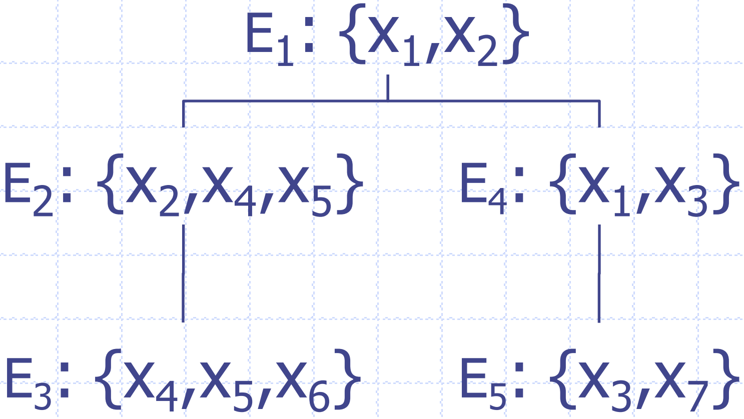
r: bounded max arity
s: $\max\{|U \cap V| : f_U \neq f_V\}$

Proceed from the *leaf* hyperedges to a chosen root hyperedge
e.g. add $PROJ_{x4,x5}(f_{x6} + f_{x4,x5,x6})$ to $f_{x4,x5}$

Tree decomposition

- A collection of subsets of X (clusters) organized as a tree




E_1: $\{x_1,x_2\}$

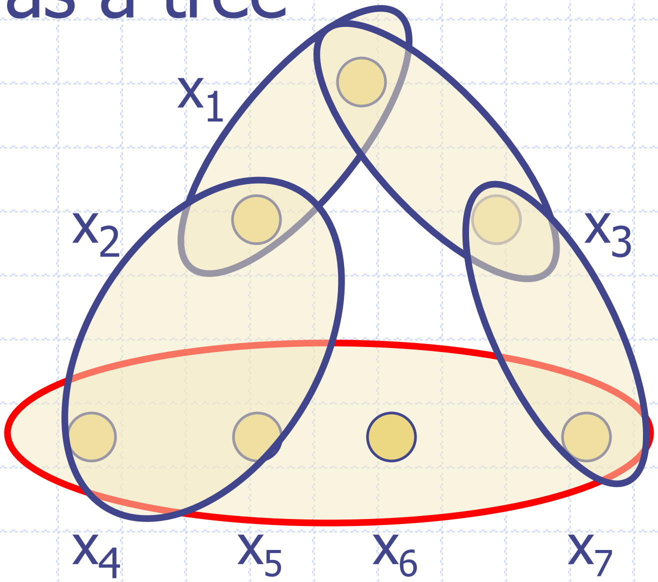
E_2: $\{x_2,x_4,x_5\}$ E_4: $\{x_1,x_3\}$

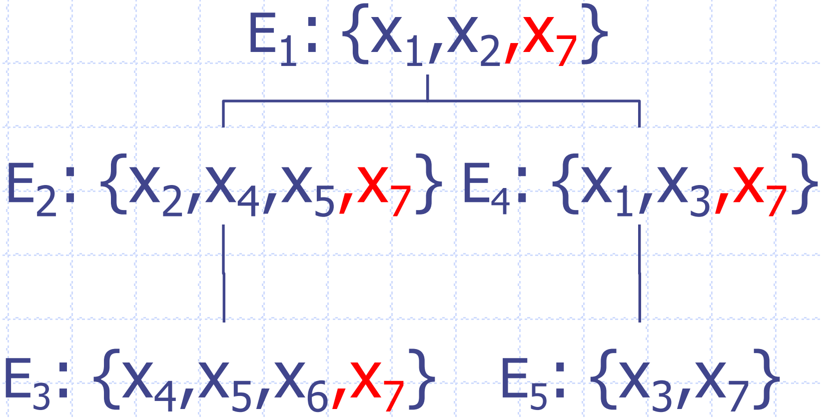
E_3: $\{x_4,x_5,x_6\}$ E_5: $\{x_3,x_7\}$

Properties:
- $\cup E_i = X$
- $\forall f_S \in F, \exists E_i$ such that $S \in E_i$
- $\forall x \in X$, $\{E_i : x \in E_i\}$ forms a connected subtree

Tree decomposition

- A collection of subsets of X (clusters) organized as a tree





E_1: $\{x_1,x_2,x_7\}$

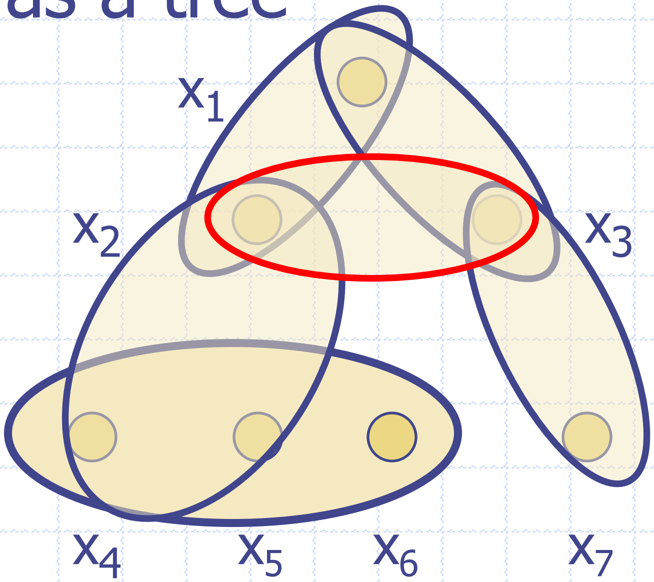
E_2: $\{x_2,x_4,x_5,x_7\}$ E_4: $\{x_1,x_3,x_7\}$

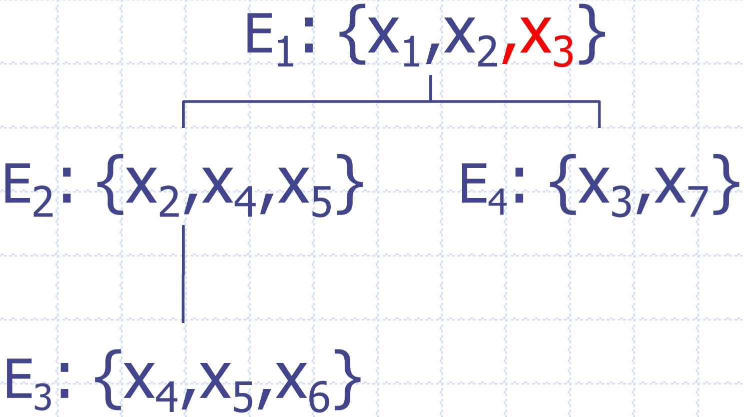
E_3: $\{x_4,x_5,x_6,x_7\}$ E_5: $\{x_3,x_7\}$

Properties:
- $\cup E_i = X$
- $\forall f_S \in F, \exists E_i$ such that $S \in E_i$
- $\forall x \in X, \{E_i : x \in E_i\}$ forms a connected subtree

Tree decomposition

- A collection of subsets of X (clusters) organized as a tree





E_1: $\{x_1,x_2,x_3\}$

E_2: $\{x_2,x_4,x_5\}$

E_4: $\{x_3,x_7\}$

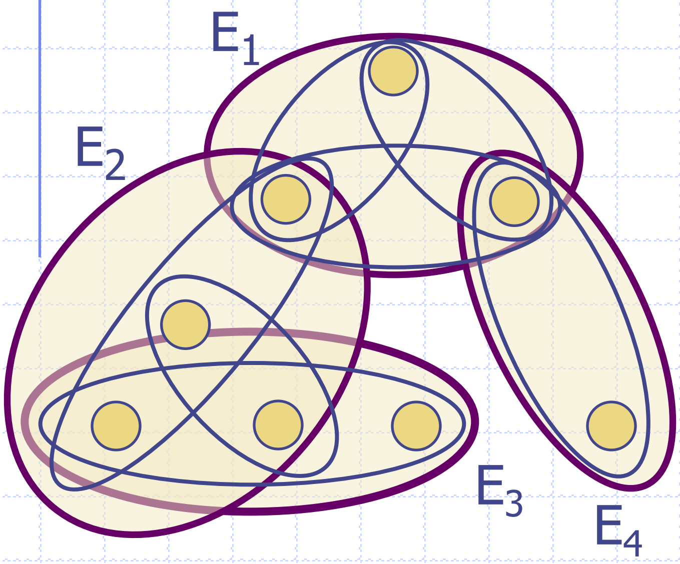
E_3: $\{x_4,x_5,x_6\}$

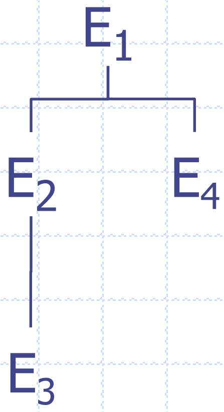
Properties:

- $\cup E_i = X$
- $\forall c_S \in P, \exists E_i$ such that $S \in E_i$
- $\forall x \in X, \{E_i : x \in E_i\}$ forms a connected subtree

Tree decomposition

Bounded treewidth VCSPs are tractable





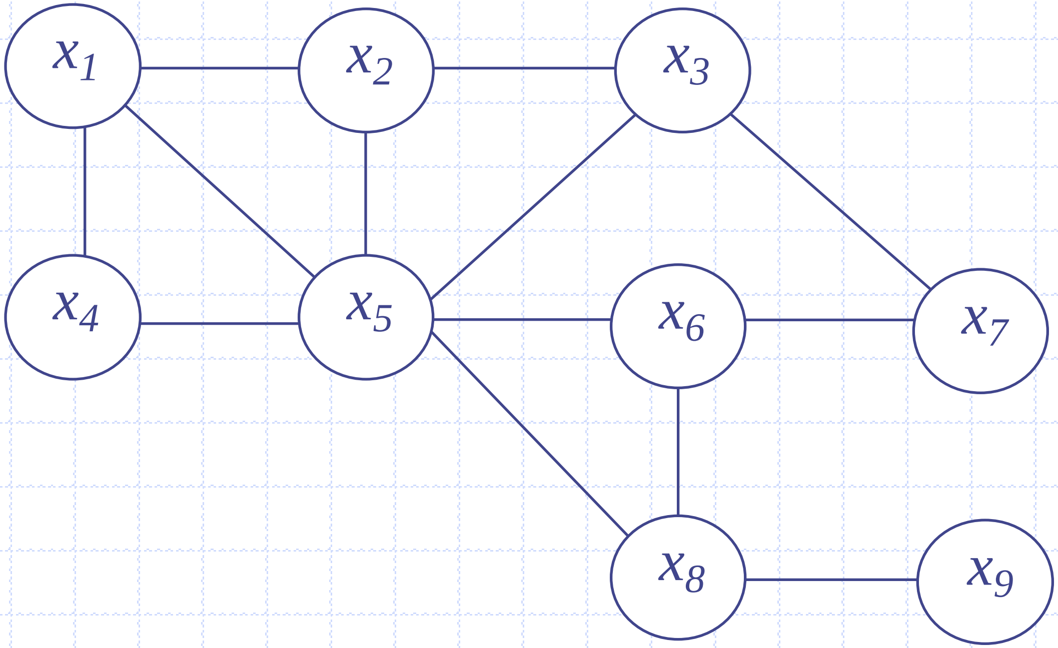
Time complexity $O(e\, d^{w+1})$
Space complexity $O(n\, d^{s})$

w: bounded treewidth
= max |Ei| - 1

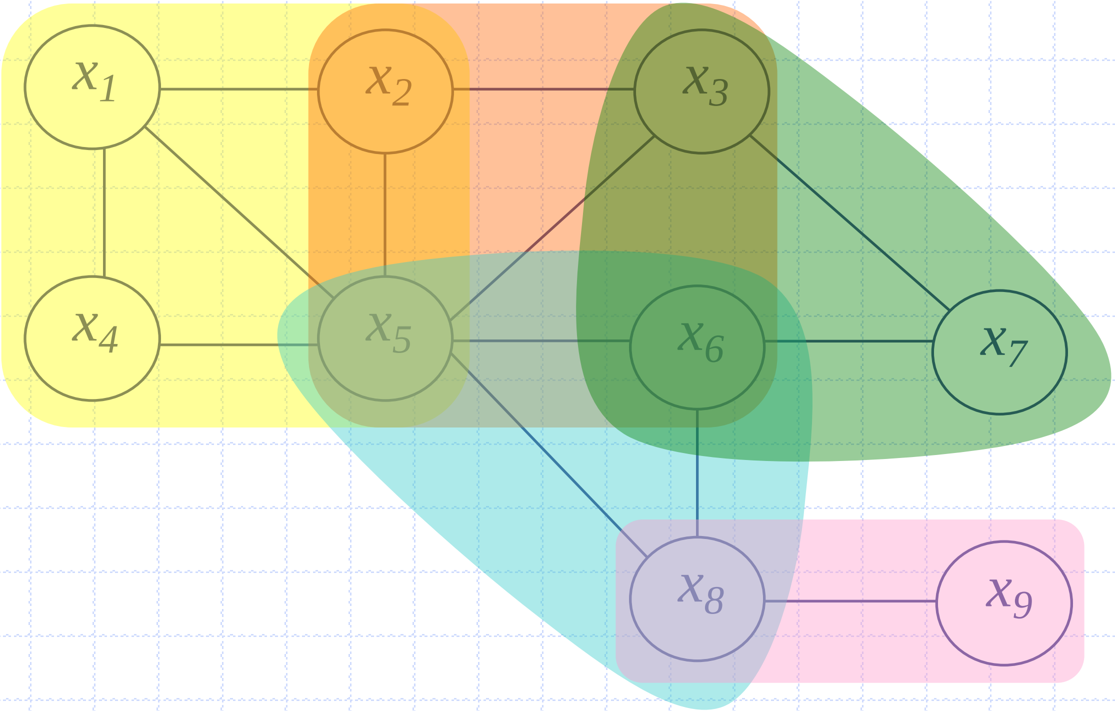
s: max $\{|E_i \cap E_j|: i \neq j\}$

Finding a tree decomposition with minimum w* is NP-hard!

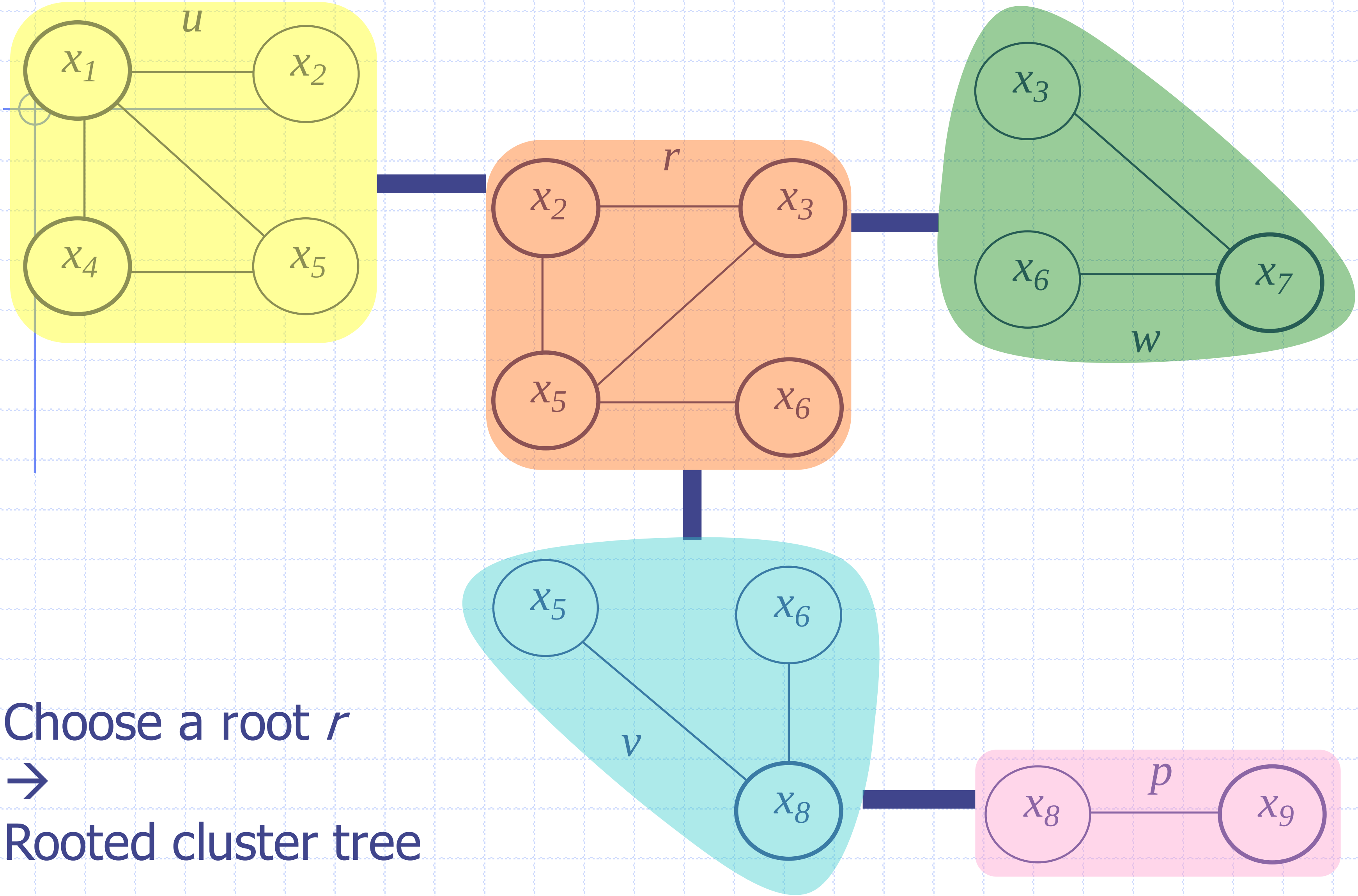
Cluster tree elimination



Cluster tree elimination



Tree decomposition

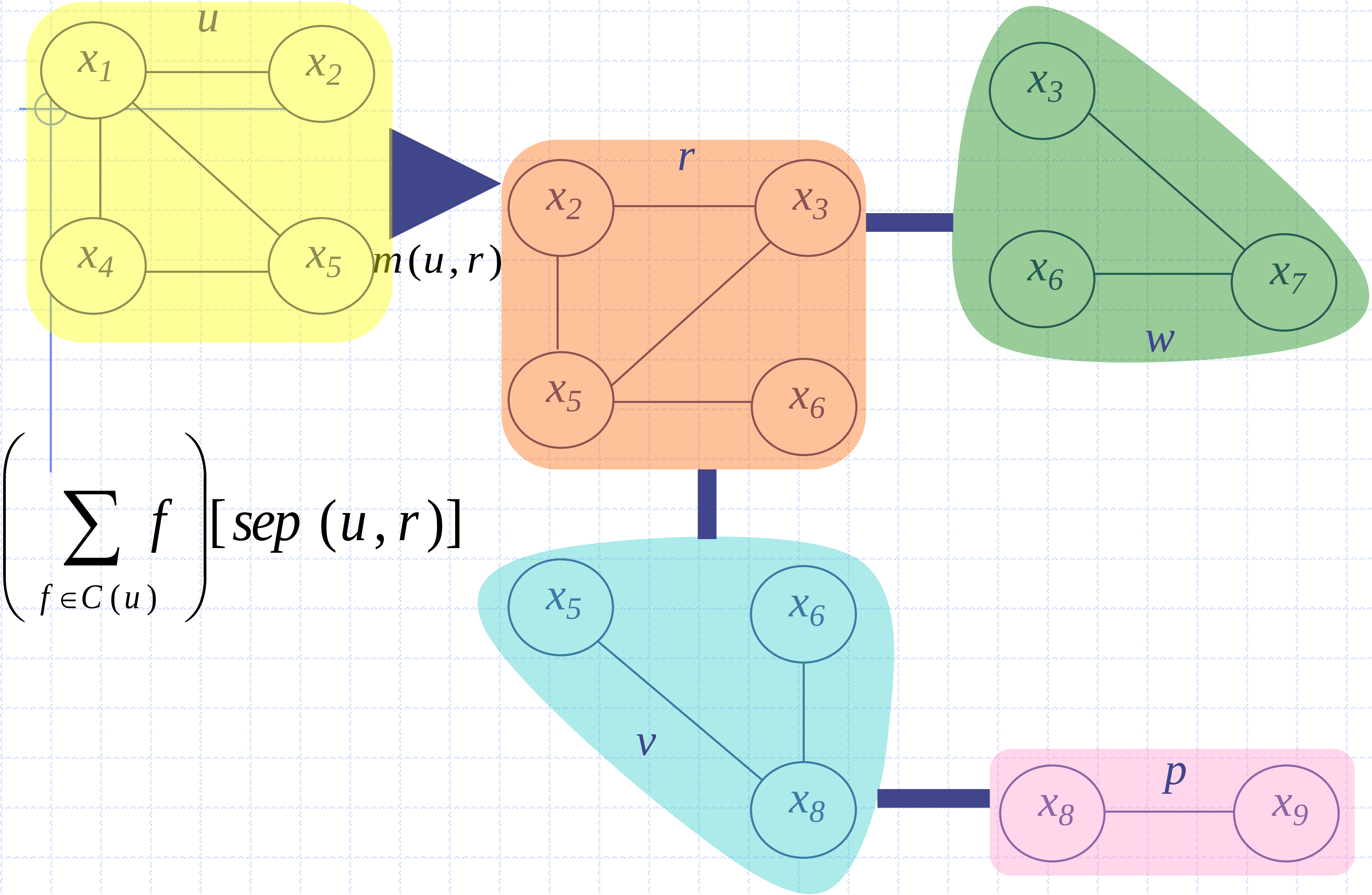


Choose a root *r*

→

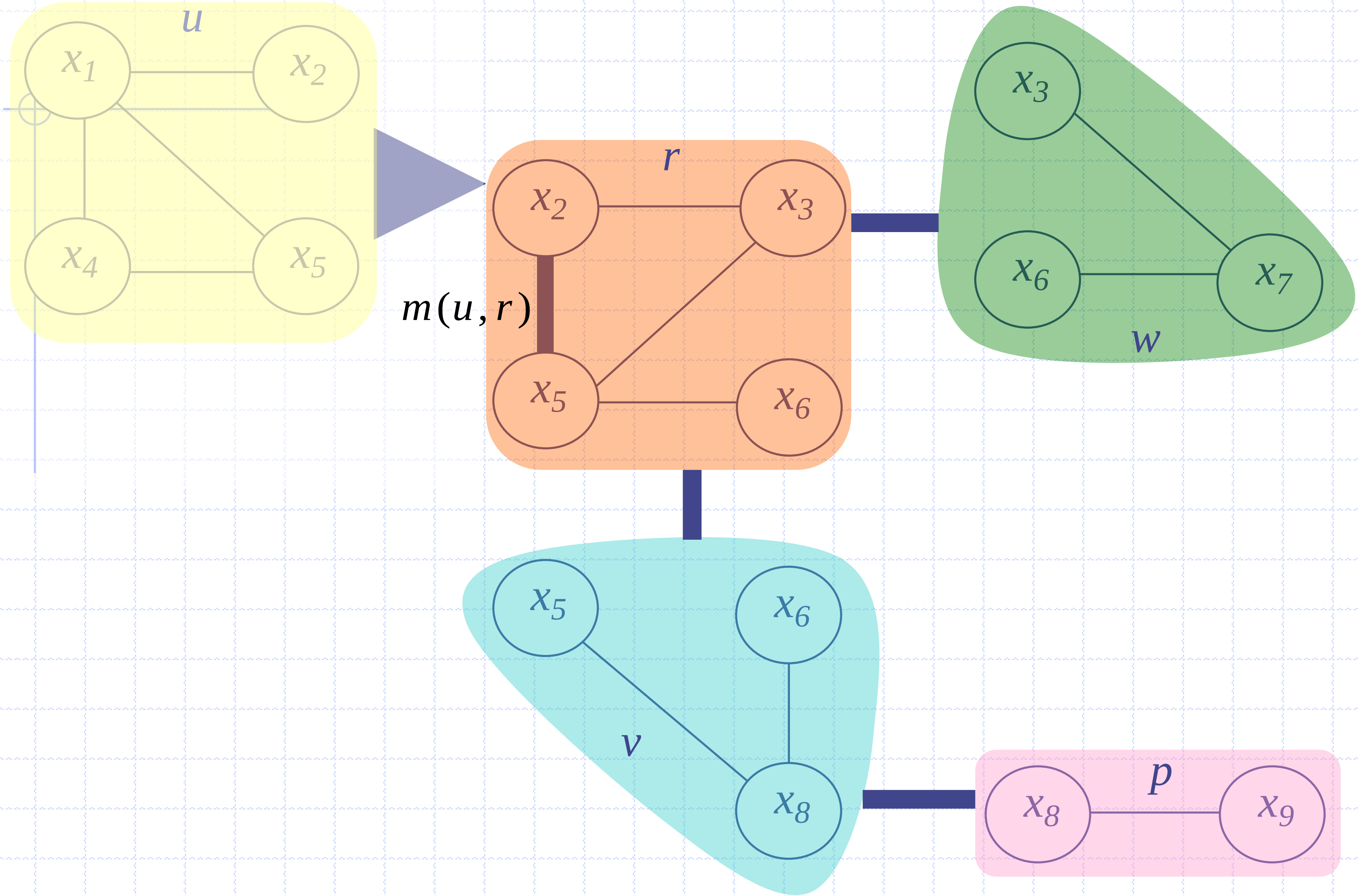
Rooted cluster tree

Cluster Tree Elimination

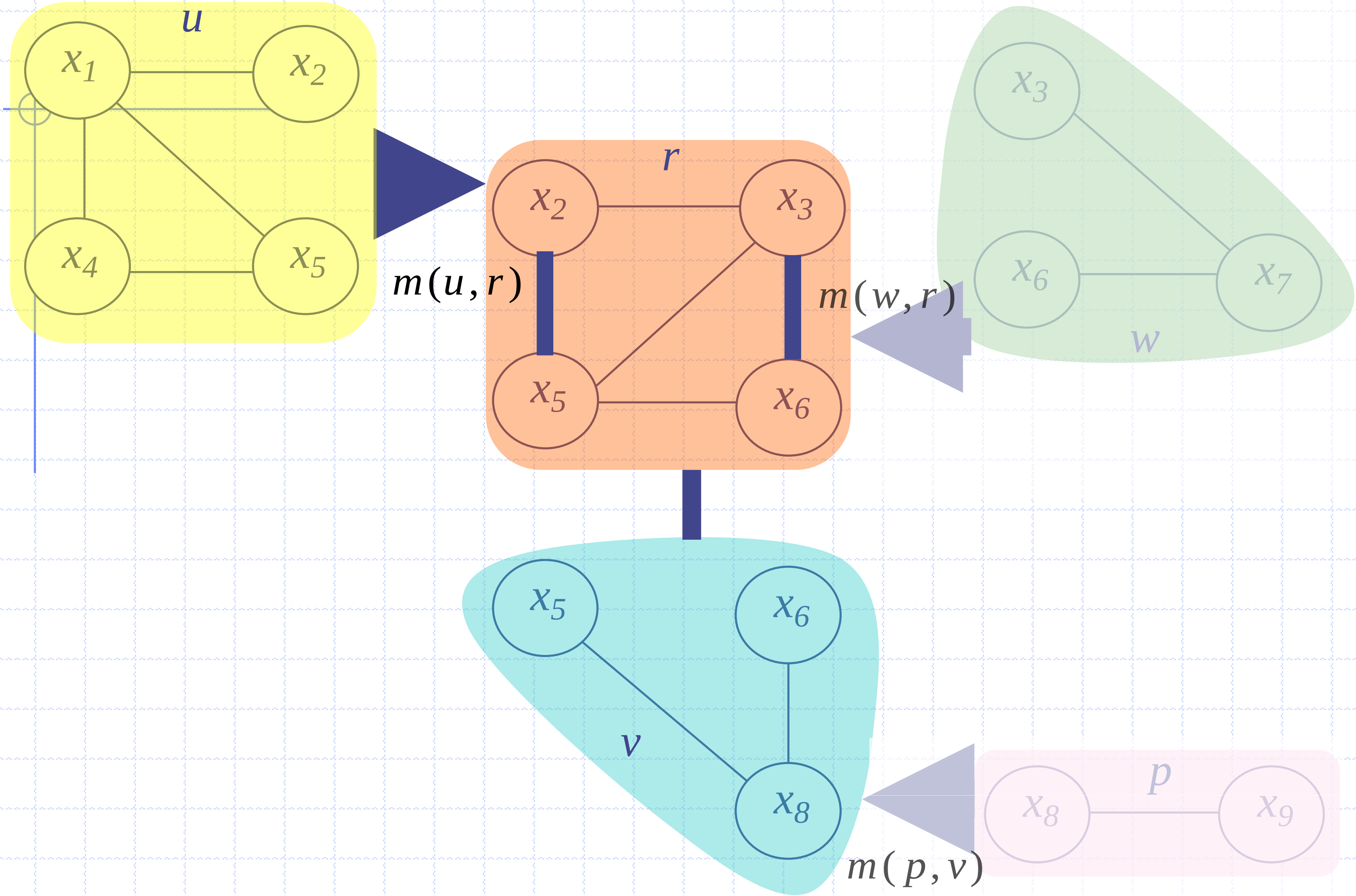


$$\left(\sum_{f \in C(u)} f \right) [sep(u,r)]$$

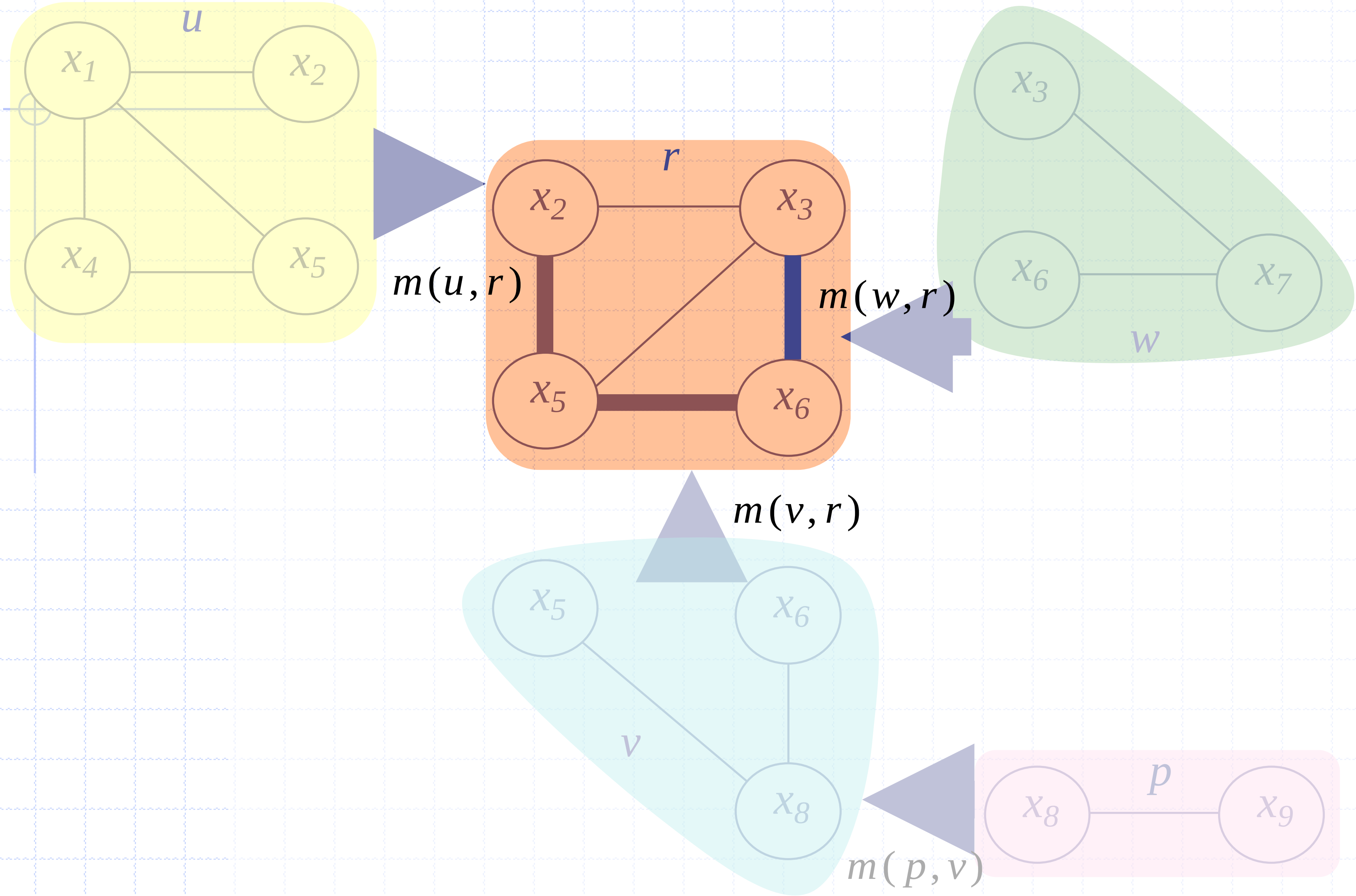
Cluster Tree Elimination



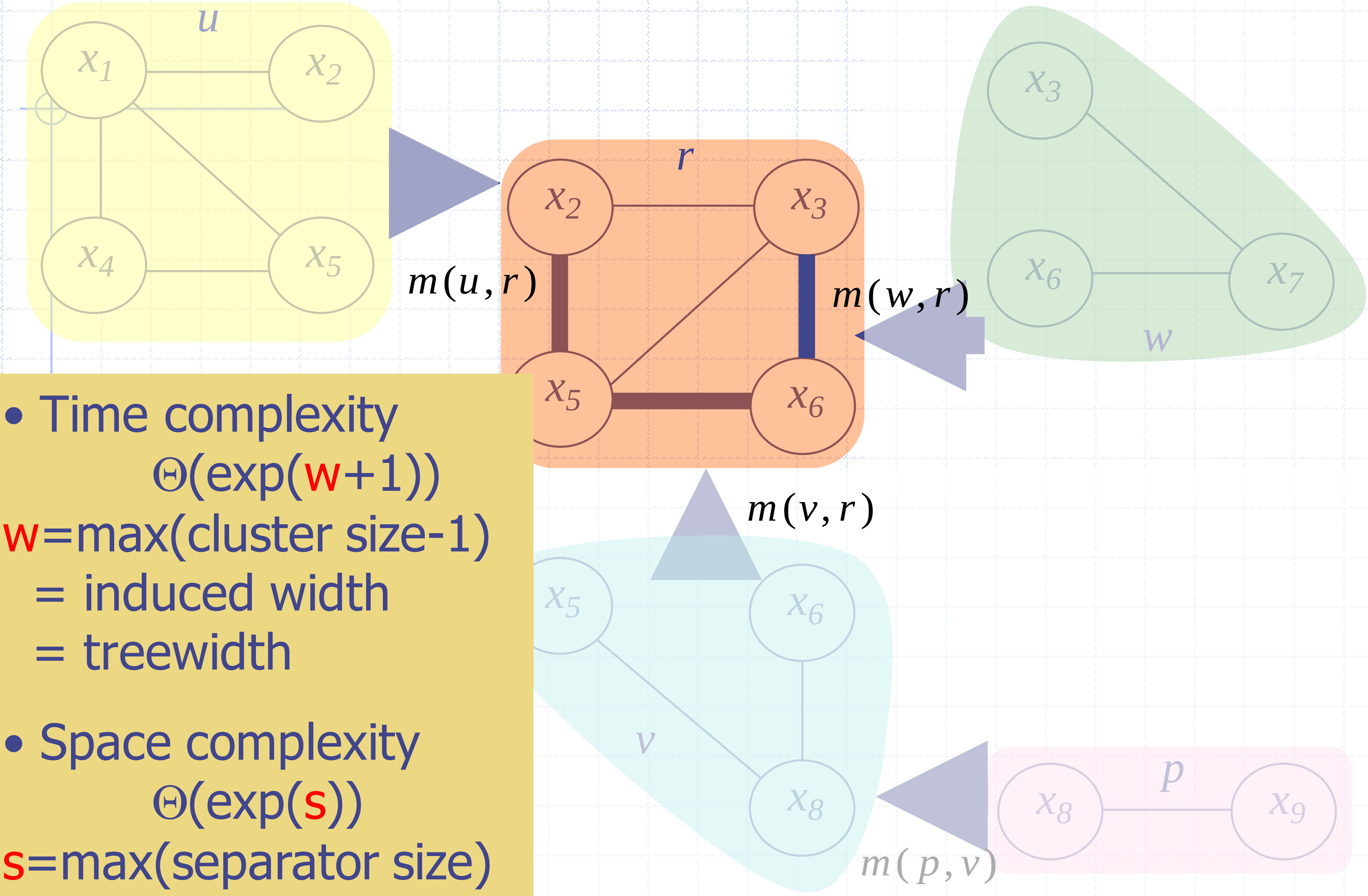
Cluster Tree Elimination



Cluster Tree Elimination



Cluster Tree Elimination



- Time complexity

  Θ(exp(w+1))

w=max(cluster size-1)

= induced width

= treewidth

- Space complexity

  Θ(exp(s))

s=max(separator size)

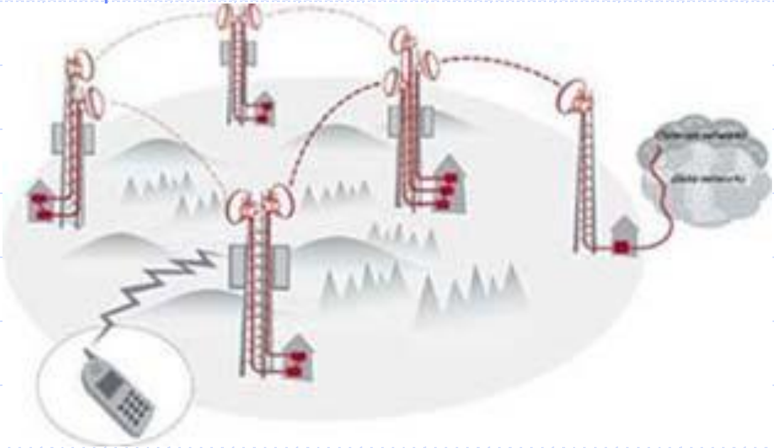
≤ w

History / terminology

- Operations Research: Non serial **dynamic programming** (Bertelé Brioschi, 72)
- Databases: Acyclic DB (Beeri et al 1983)
- Bayesian nets: Join-tree (Pearl 88, Lauritzen et Spiegelhalter 88)
- Constraint nets: Adaptive Consistency (Dechter and Pearl 88)
- Genetics: Peeling (Elston and Stewart 71)

Radio Link Frequency Assignment Problem

(Cabon et al., *Constraints* 1999) (Koster *et al.*, *4OR* 2003)



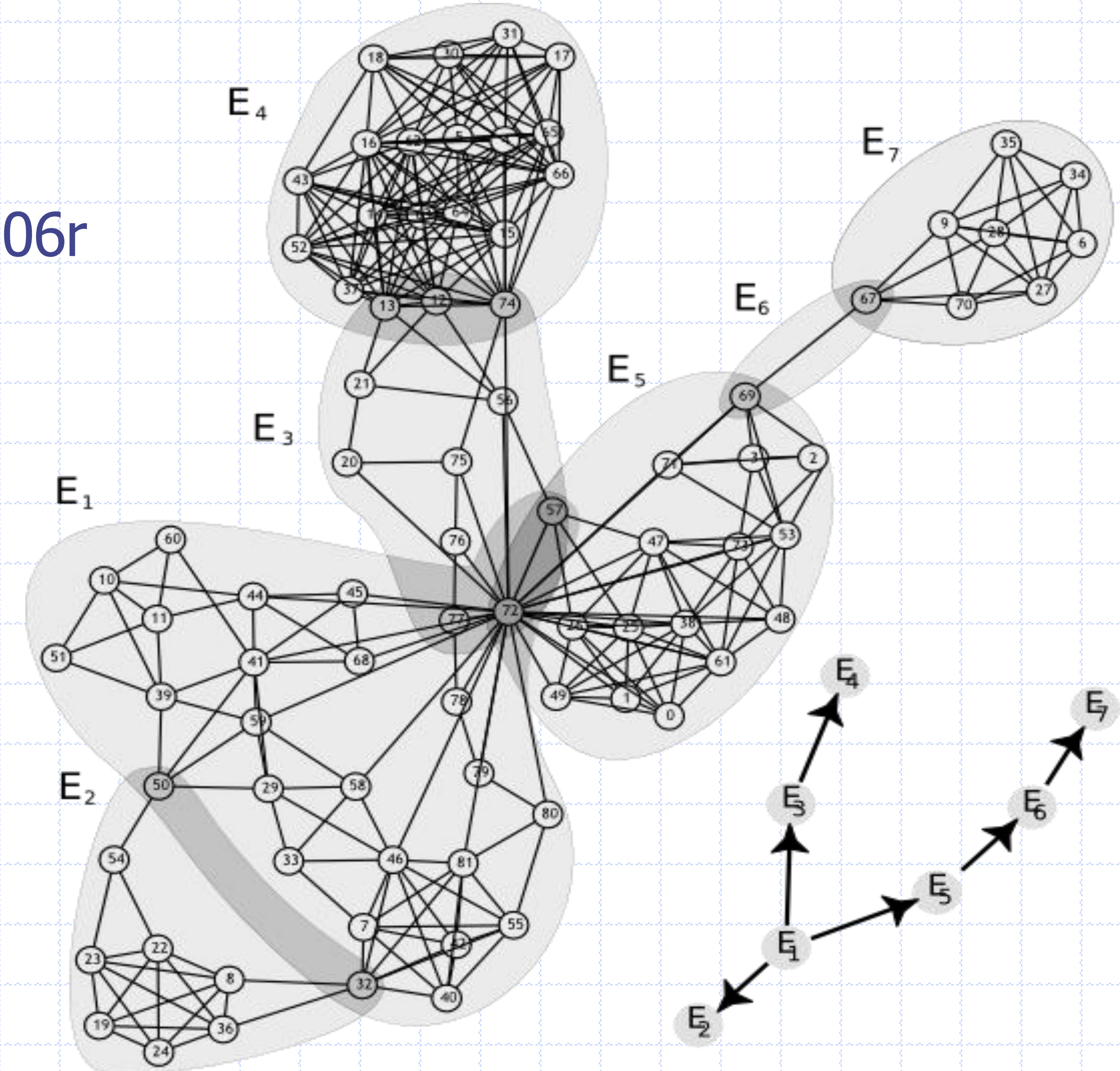
- Given a telecommunication network
- …find the best frequency for each communication link, avoiding interferences

- Best can be:
  - Minimize the maximum frequency, no interference (max operator)
  - **Minimize the global interference (sum operator)**
- Generalizes graph coloring problems: $|f_i - f_j| \geq a$

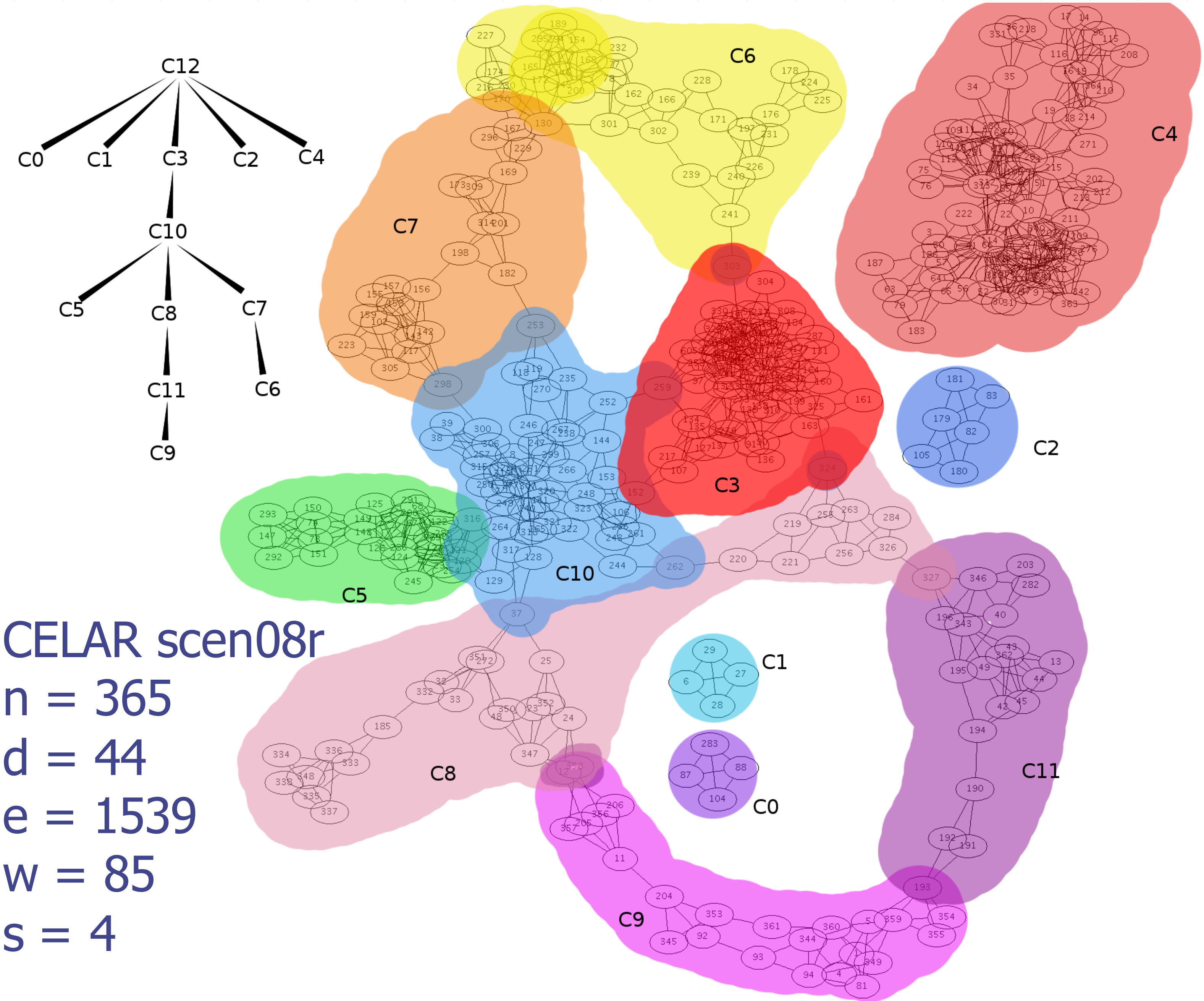
***CELAR** problem size: n=100—458 ; d=44 ; e=1,000—5,000*

Tree decomposition example

CELAR scen06r
n = 82
d = 44
e = 327
w = 26
s = 3



(Allouche et al, CP 2010)



CELAR scen08r

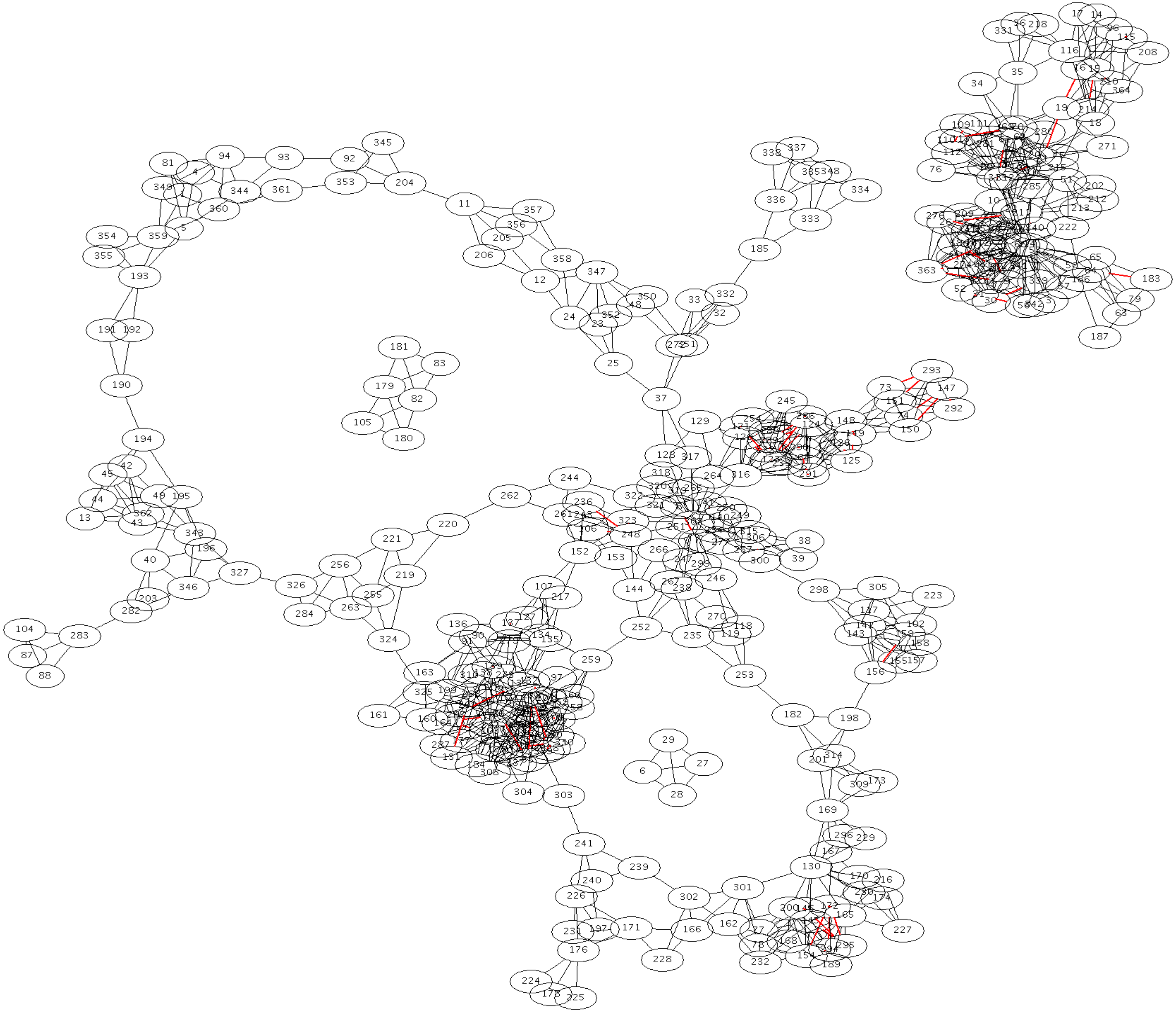
n = 365

d = 44

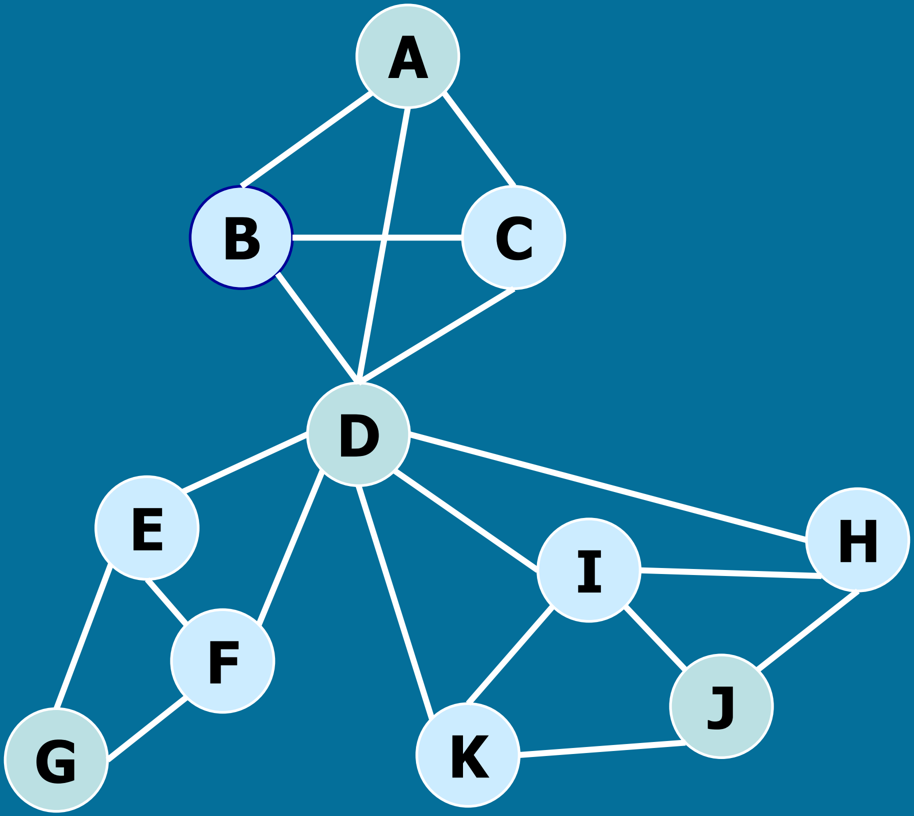
e = 1539

w = 85

s = 4

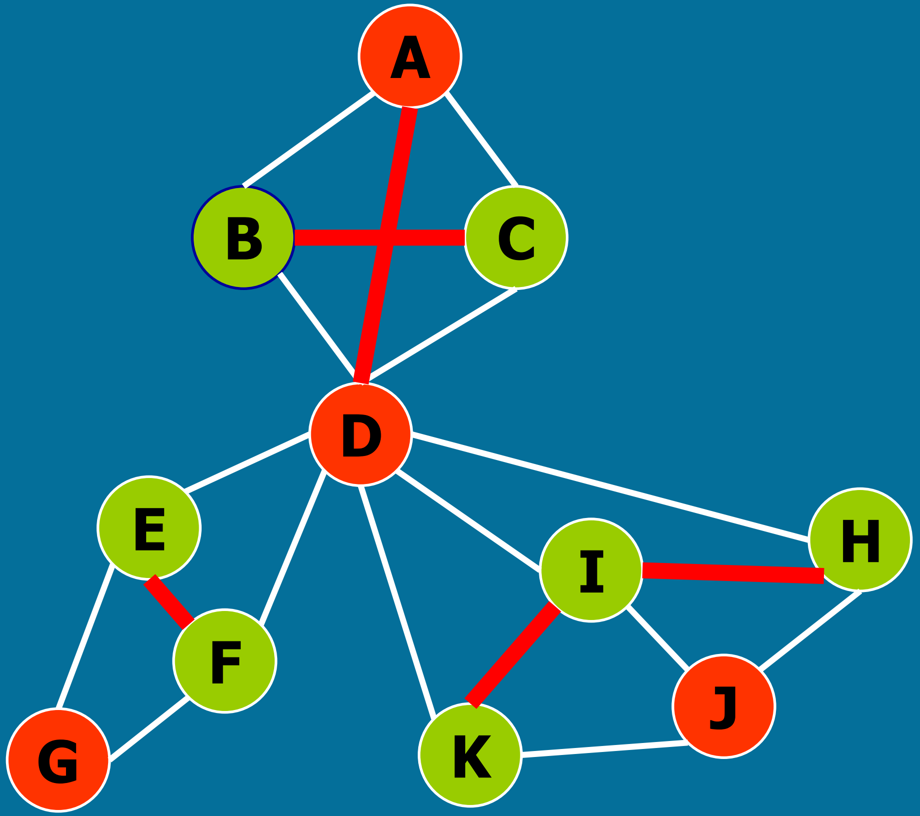

17
14
331
36
218
96
115
116
208
35
16
15
34
210
364
19
214
18
109
111
280
271
110
112
76
215
51
202
212
10
213
276
209
222
65
363
58
64
183
186
52
30
79
63
187
339
57
342
345
92
204
93
94
81
4
349
1
344
361
353
360
5
354
359
355
193
191
192
190
194
11
357
356
205
206
358
347
12
350
48
352
24
23
25
33
332
32
273
251
37
338
337
335
348
334
336
333
185
181
83
179
82
105
180
45
42
49
195
44
362
13
43
343
196
40
327
203
346
282
104
283
87
88
262
244
236
322
261
323
106
248
220
221
219
256
326
255
284
263
324
152
153
266
107
217
144
252
235
253
270
118
119
246
267
238
247
299
38
39
300
306
315
129
128
317
318
320
319
321
264
316
245
254
286
124
148
149
125
291
73
293
147
151
74
292
150
298
305
223
117
141
102
143
159
188
155
157
156
182
198
201
314
309
173
169
296
229
167
130
170
216
230
174
227
301
302
200
172
165
145
162
166
77
78
168
295
154
189
232
228
171
197
231
176
226
240
241
239
224
178
225
303
304
308
184
131
287
164
160
161
163
325
199
310
97
259
136
137
127
134
135
91
90
29
27
6
28

Soft 2-Coloring example



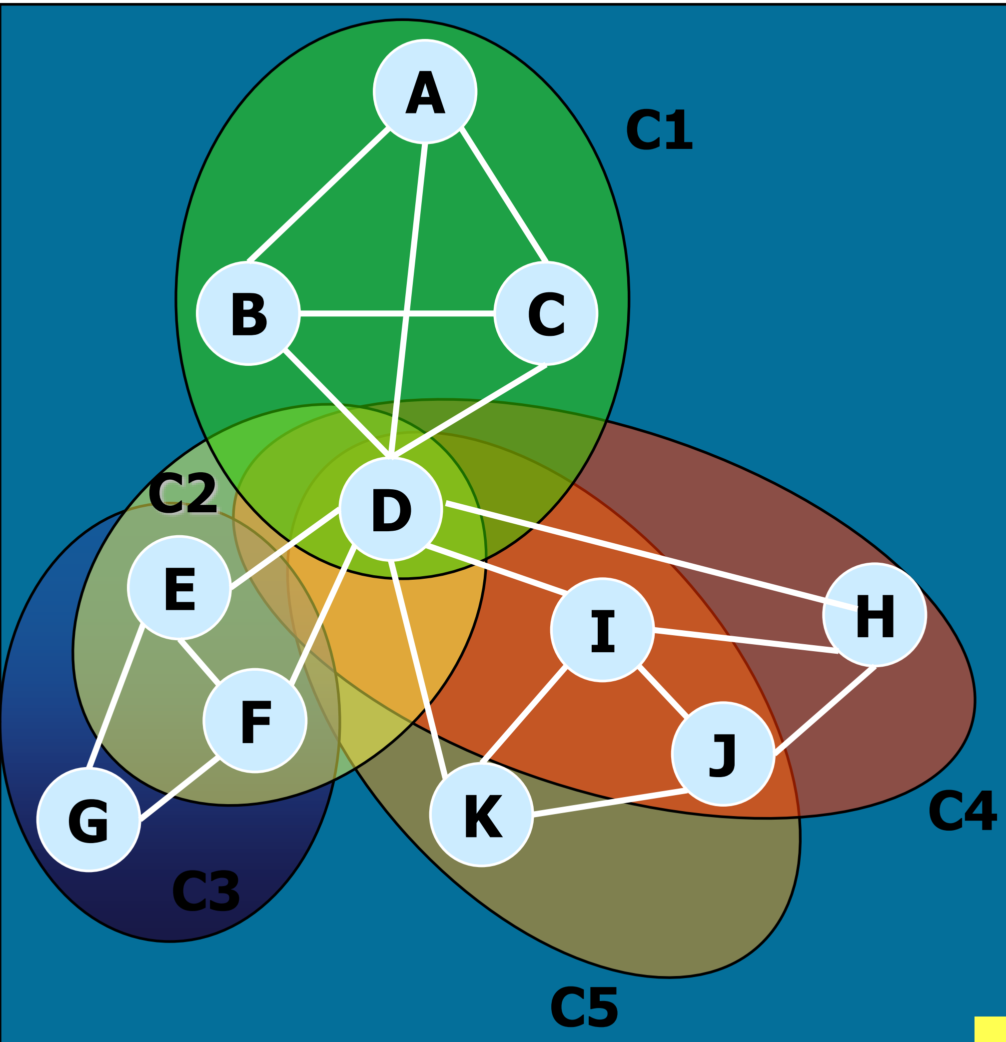
X_i	X_j	c_{ij}
🟥	🟥	1
🟥	🟩	0
🟩	🟥	0
🟩	🟩	1

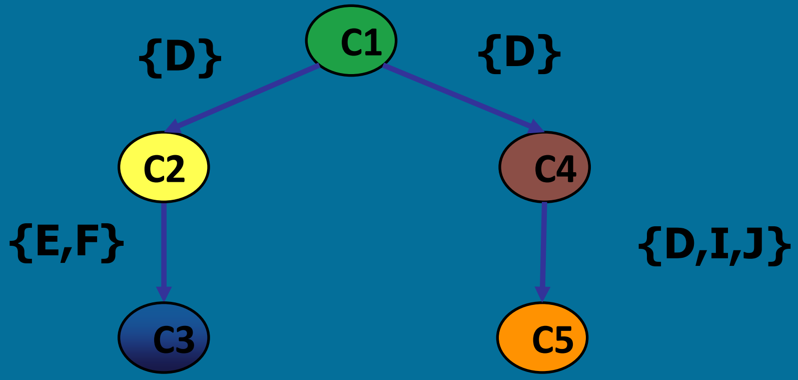
Soft 2-Coloring example



Optimal solution with a cost of 5

Tree Decomposition

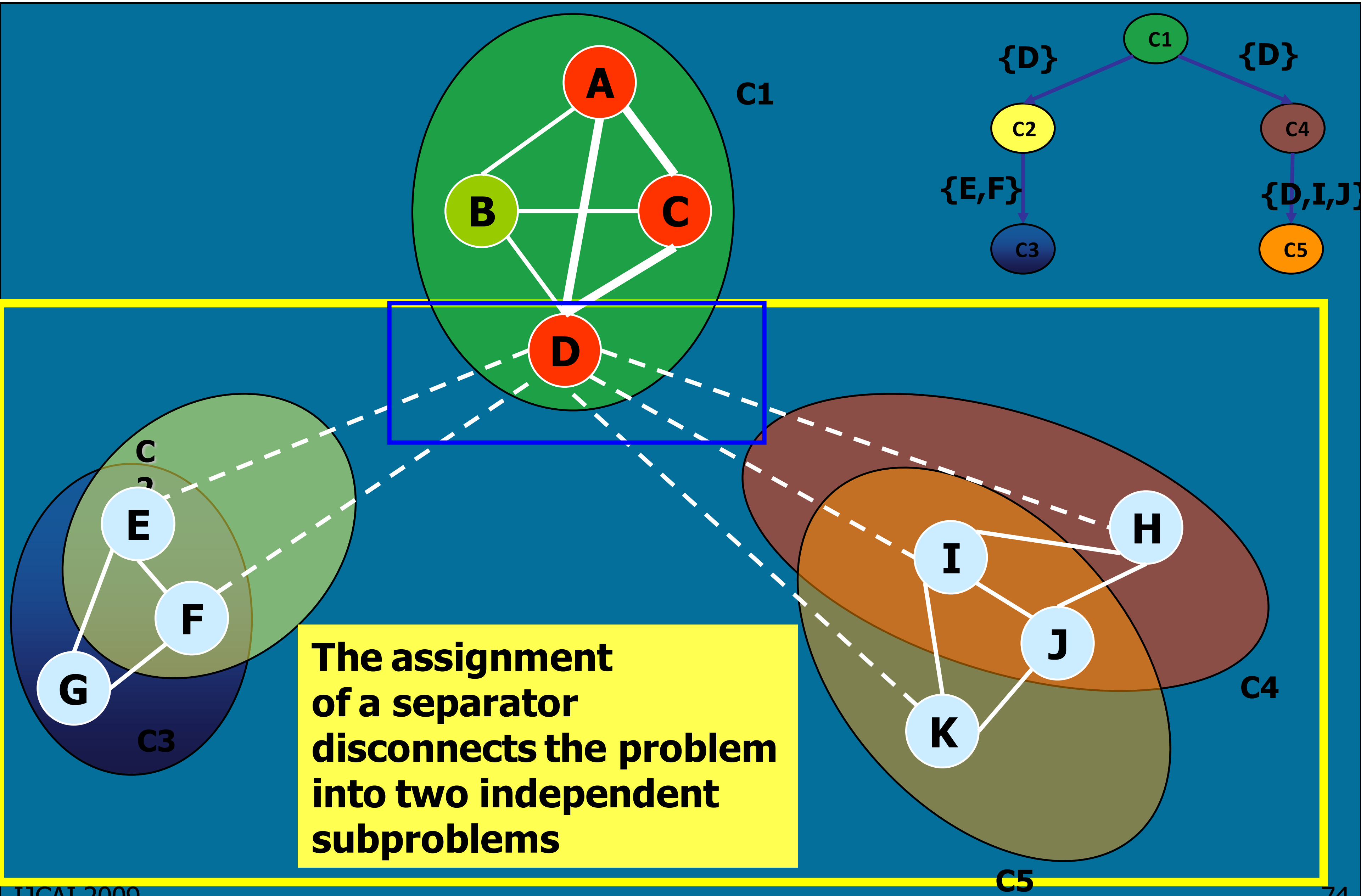




The set of clusters covers
the set of variables and
the set of cost functions

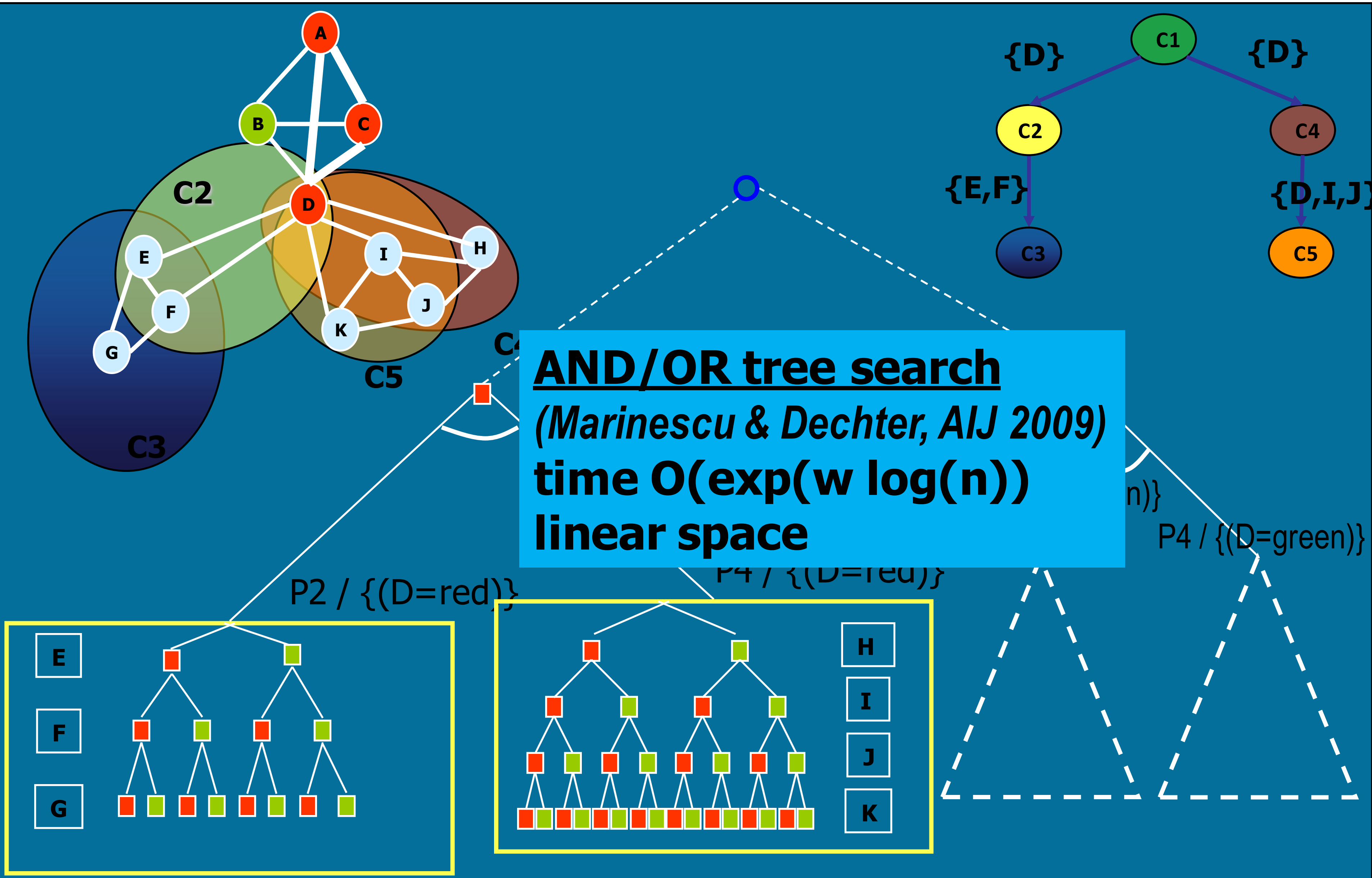
**Separator = intersection between
two connected clusters**

Search with Tree Decomposition



The assignment of a separator disconnects the problem into two independent subproblems

Search with Tree Decomposition



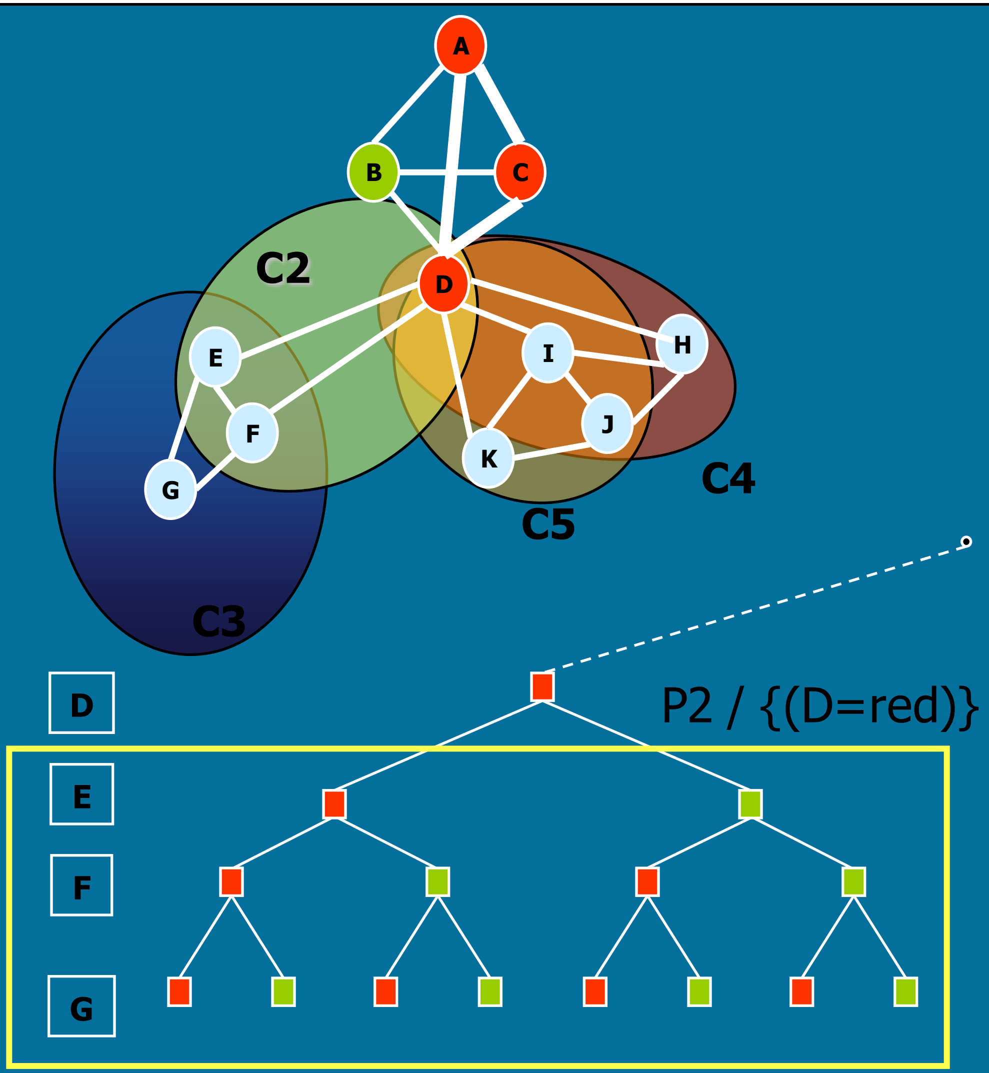
AND/OR tree search

(Marinescu & Dechter, AIJ 2009)

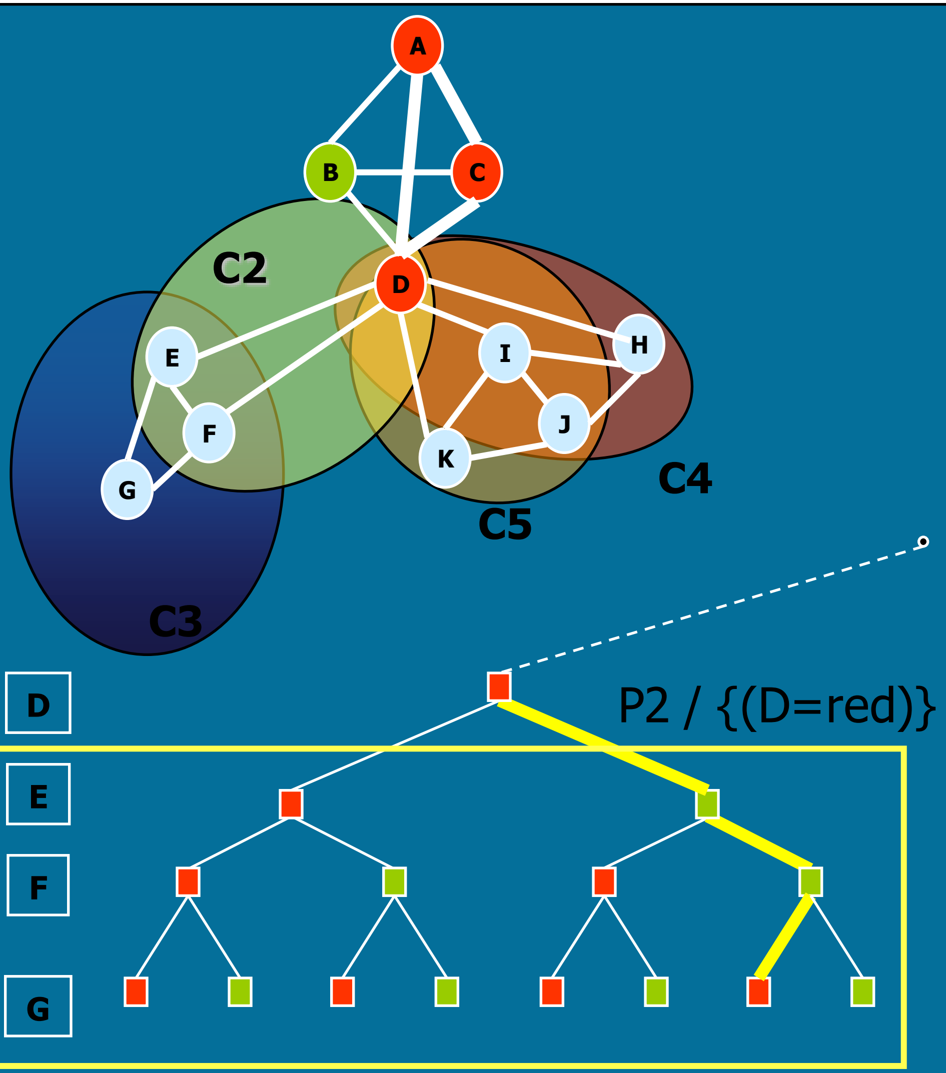
time O(exp(w log(n))

linear space

Search with Tree Decomposition



Search with Tree Decomposition



Record the optimum of P2 / {(D=red)}

<u>AND/OR graph search</u>

(Marinescu & Dechter, AIJ 2009)

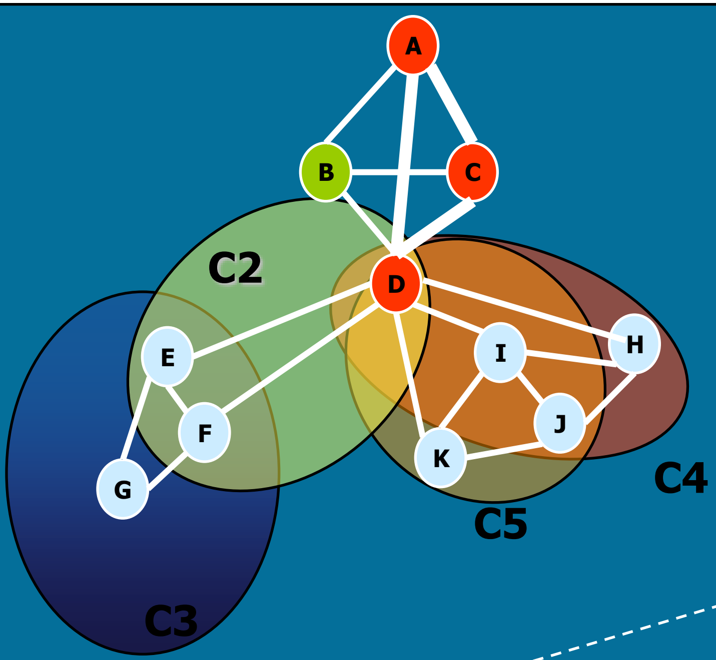
time O(exp(w))

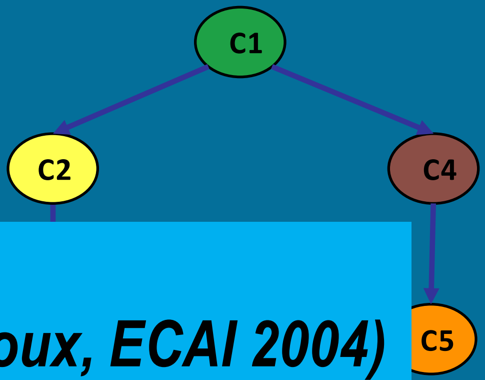
space O(exp(w))

bound k = 5.

It may be useless to compute the optimum of P2 / {(D=red)}, only a lower bound is needed!

Backtrack bounded by Tree Decomposition





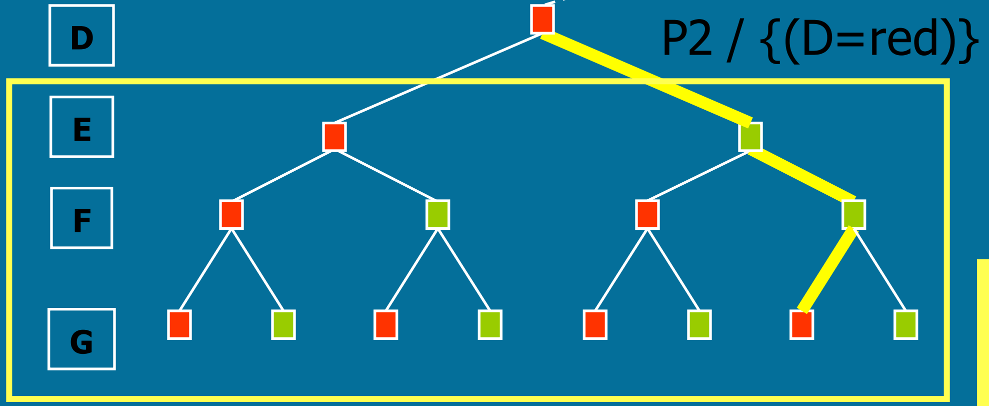
BTD

(Jégou & Terrioux, ECAI 2004)

(de Givry et al., AAAI 2006)

time O(k*exp(w))

space O(exp(w))



per

It may be useless to compute the optimum of P2,
only a lower bound is needed!

Add a local upper bound:

$$UB_{P2/\{(D=red)\}} = k - 3 - LB_{P4/\{(D=red)\}}$$

$$UB_{P2/\{(D=red)\}} = k - 3 - \max(\underbrace{f_{\varnothing}^{C4} + f_{\varnothing}^{C5}}_{\text{Maintaining local consistency}}, \underbrace{LB_{P4/\{(D=red)\}}}_{\text{Recorded during search}})$$

Russian Doll Search with Tree Decomposition

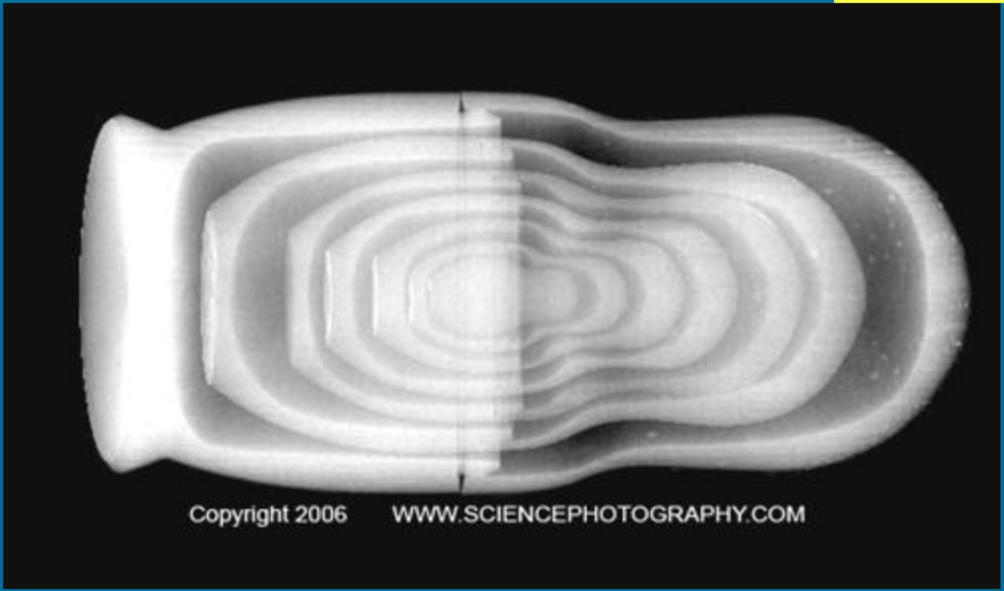
(Sanchez et al, IJCAI 2009)

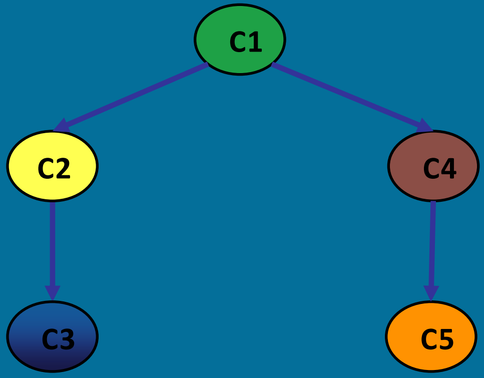
$$UB_{P2\,/\,\{(D=red)\}} = k - 3 - \max(\underbrace{f_{\varnothing}^{C4} + f_{\varnothing}^{C5}, LB_{P4\,/\,\{(D=red)\}}}_{\text{Found during search}}, \underbrace{LB_{P4}^{RDS}}_{\text{Obtained in preprocessing}})$$

RDS-BTD lower bound is a maximum between

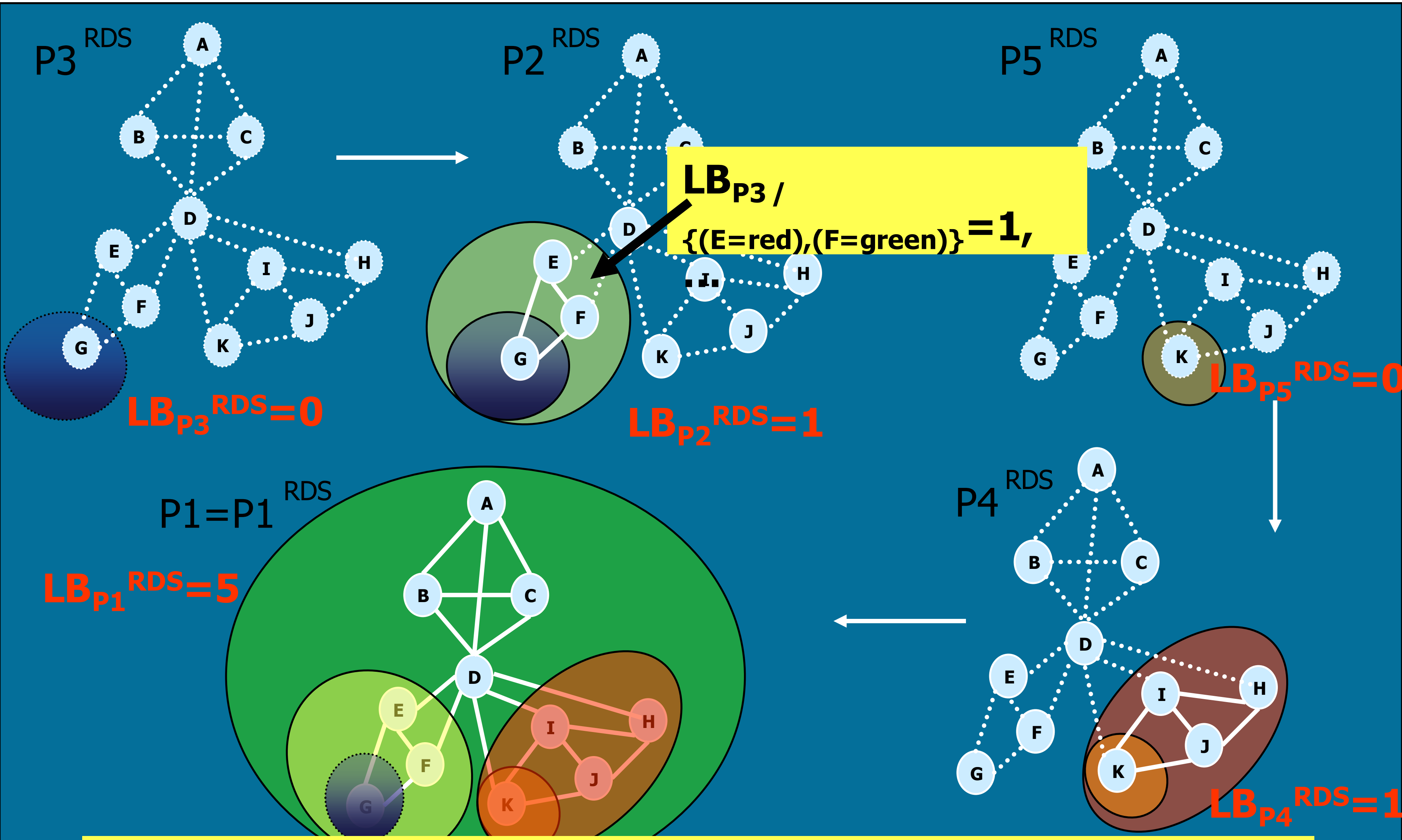
- Local Consistency
- Recorded lower bounds
- RDS lower bounds

The optimum of Pi^{RDS} is a lower bound of Pi / A for any separator assignment A





Russian Doll Search with Tree Decomposition

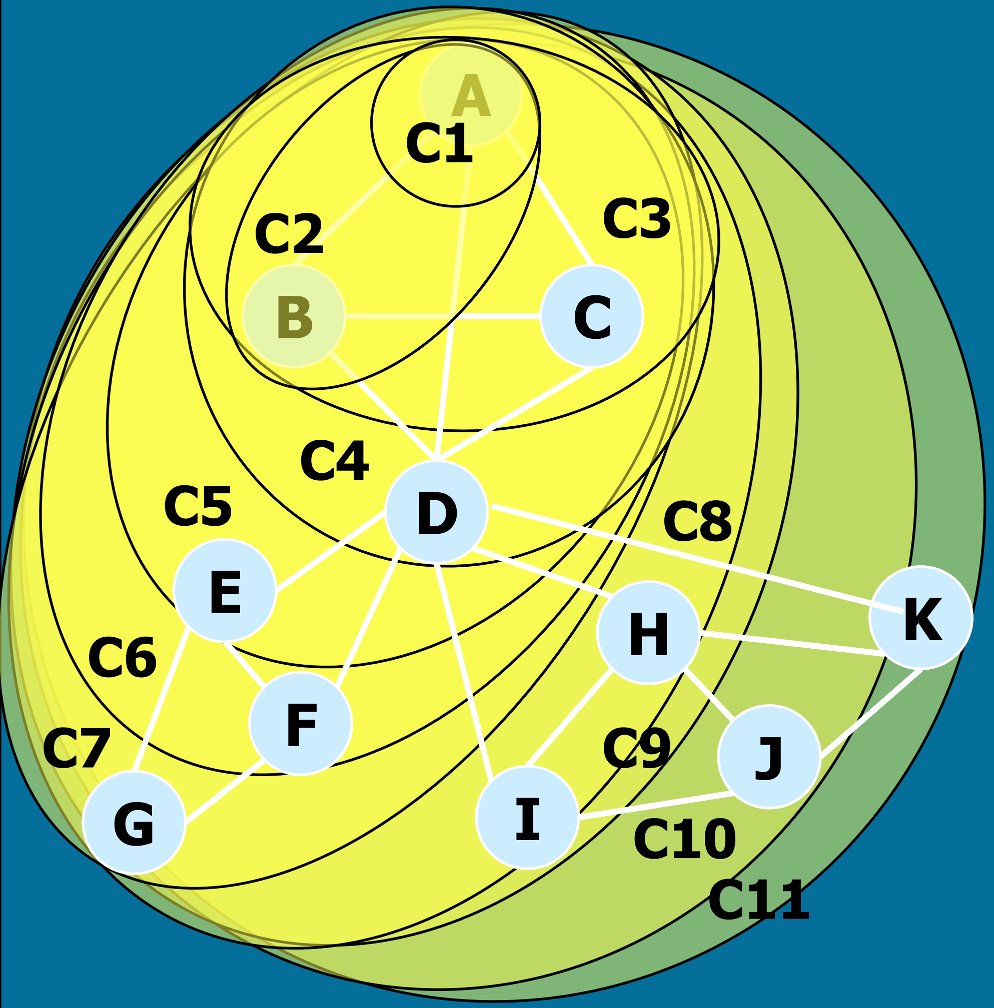


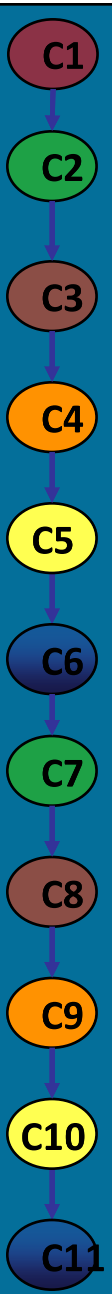
LB_{P3} / $_{\{(E=red),(F=green)\}}$ =1,

Same complexities as BTD: Time: *O(exp(w+1))* Space: *O(exp(s))*

w = tree-width = largest cluster size –1, s = largest separator size

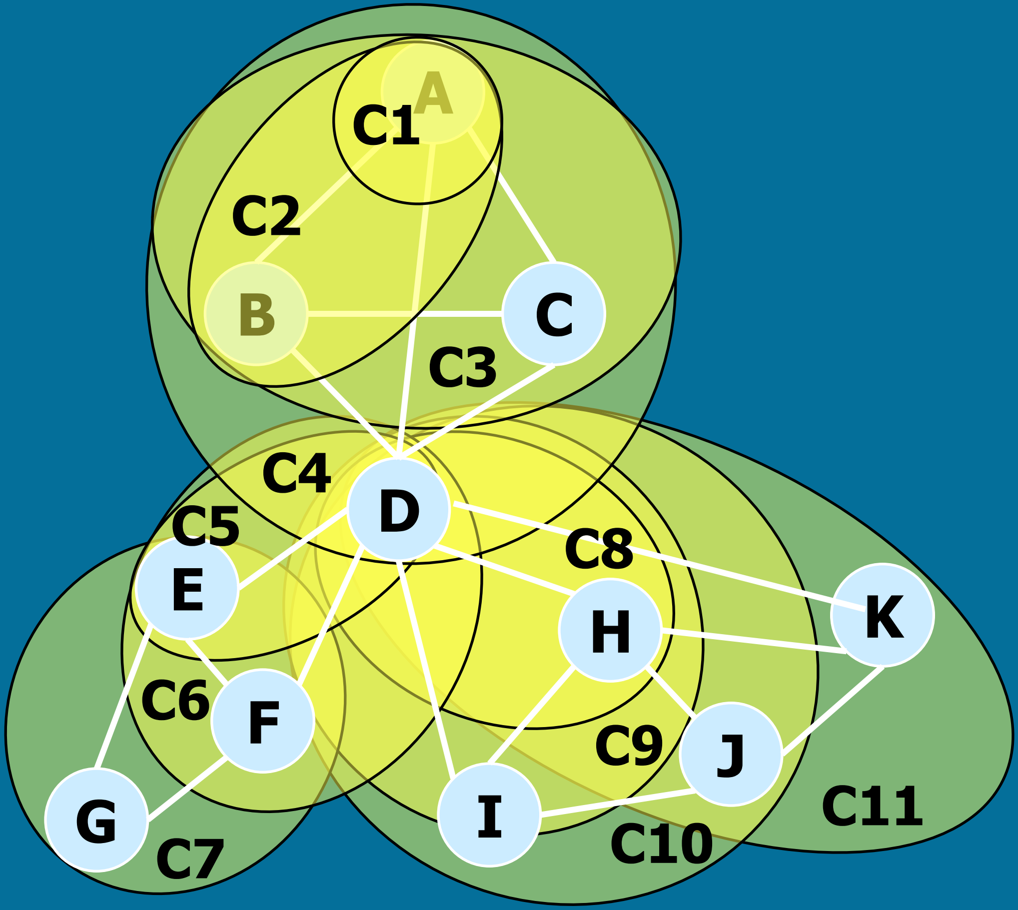
Related Work

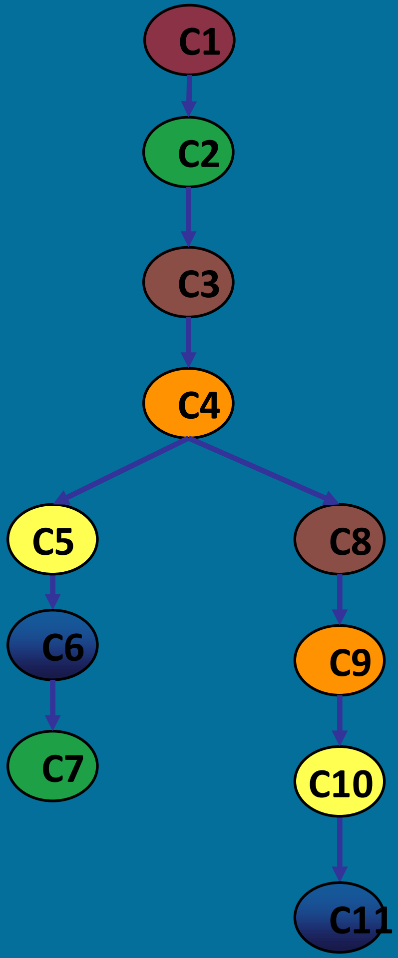




***RDS* (Verfaillie *et al*, AAAI 1996)**

Related Work



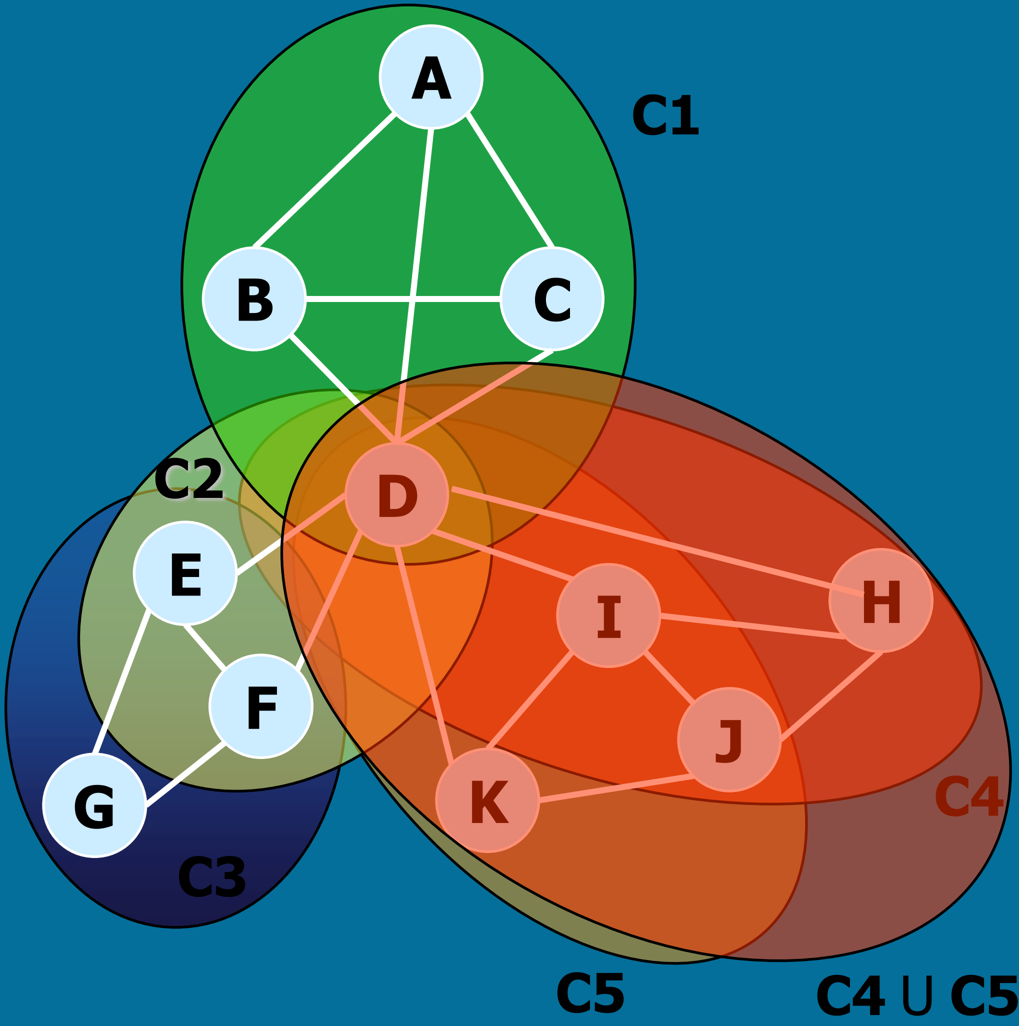


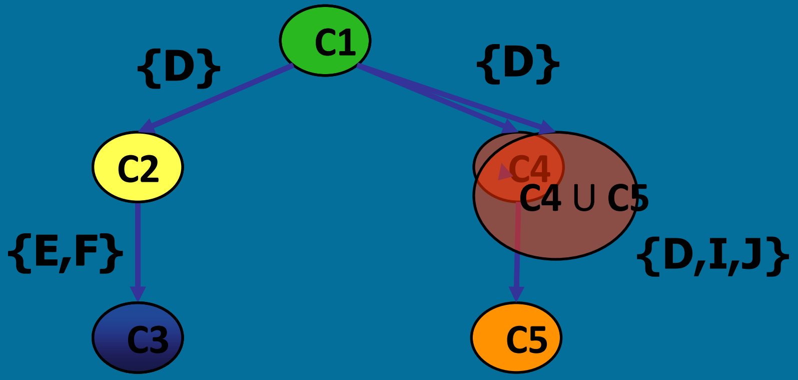
Pseudo-Tree RDS **(Larrosa *et al*, ECAI 2002)**

AND/OR graph search **(Marinescu & Dechter, AAAI 2006)**

Exploiting small separators only

(Jégou et al., CP 2007)



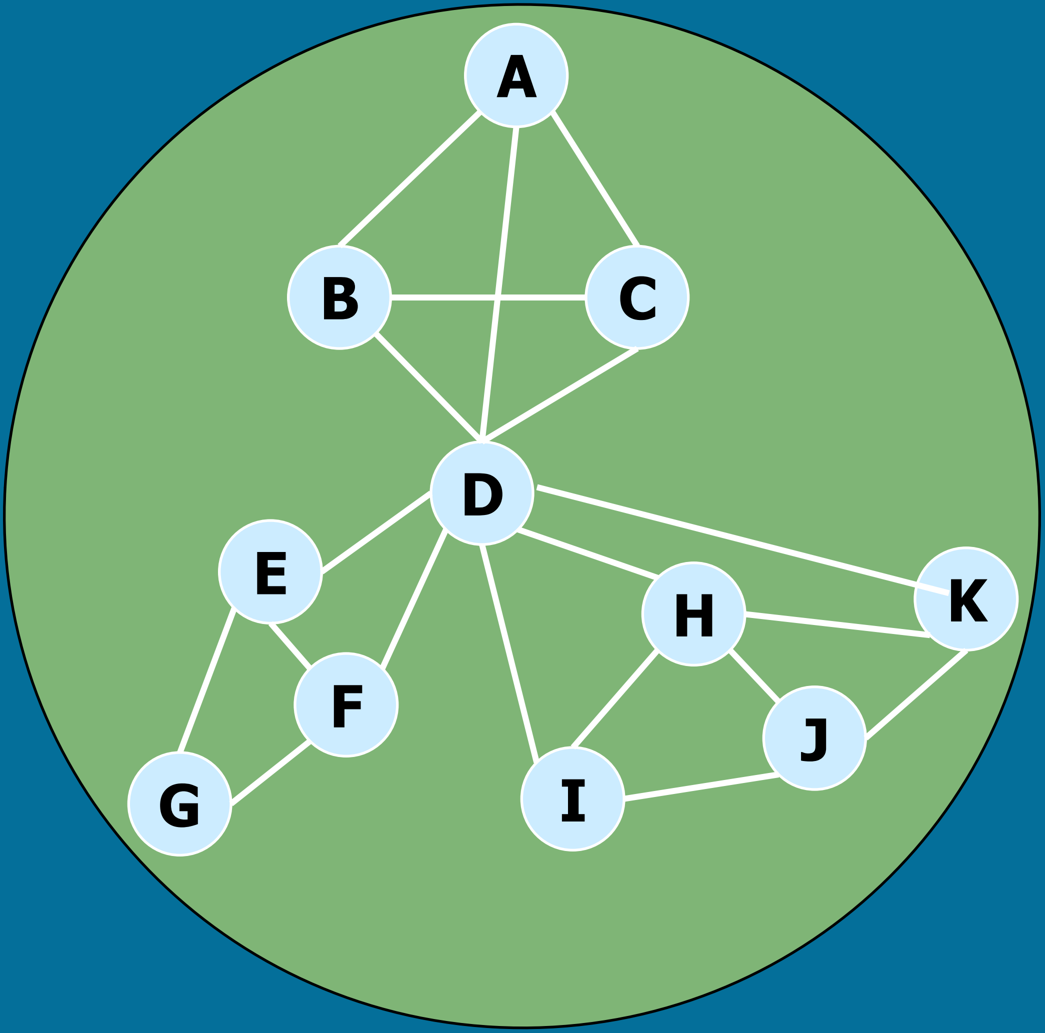


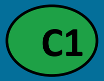
p = 2

Merge clusters i and father(i) if $|sep_{i,father(i)}| > \mathbf{p}$

→ Gives more freedom for the dynamic variable ordering heuristic

Exploiting small separators only





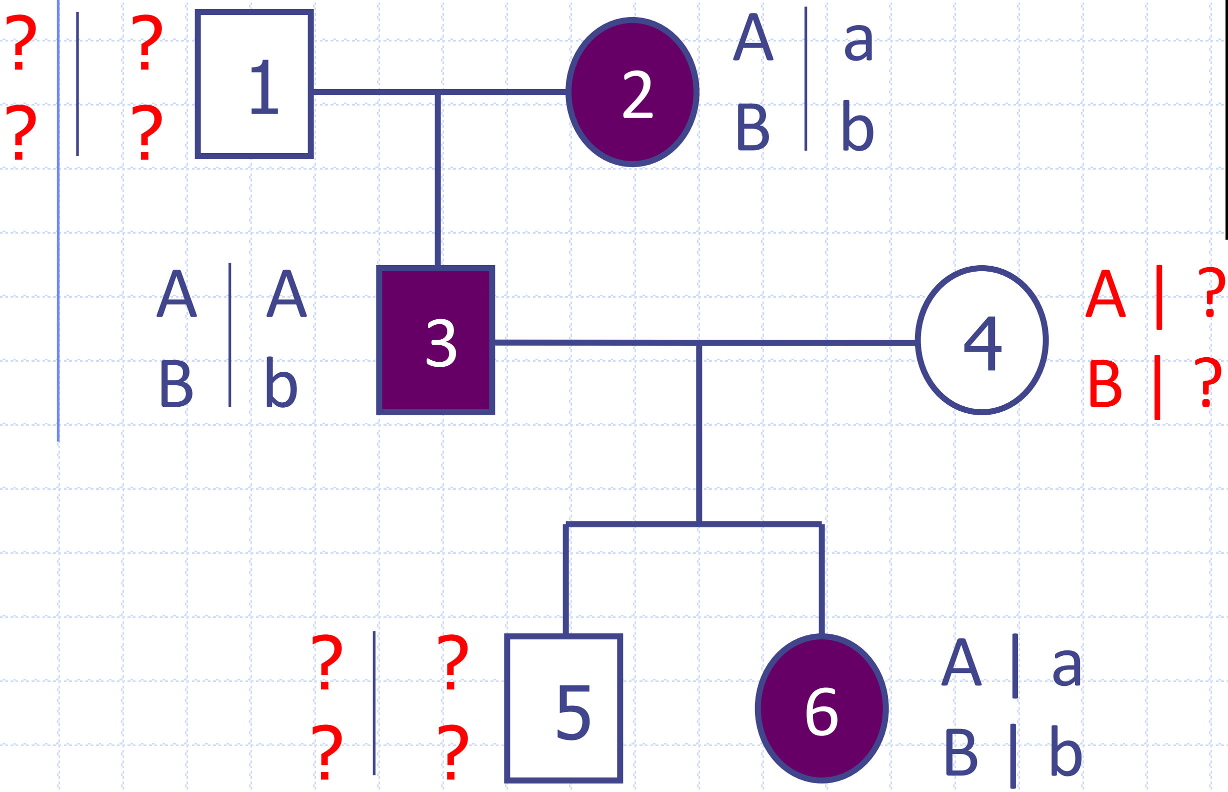
p = 0

DFBB

Some practical observations

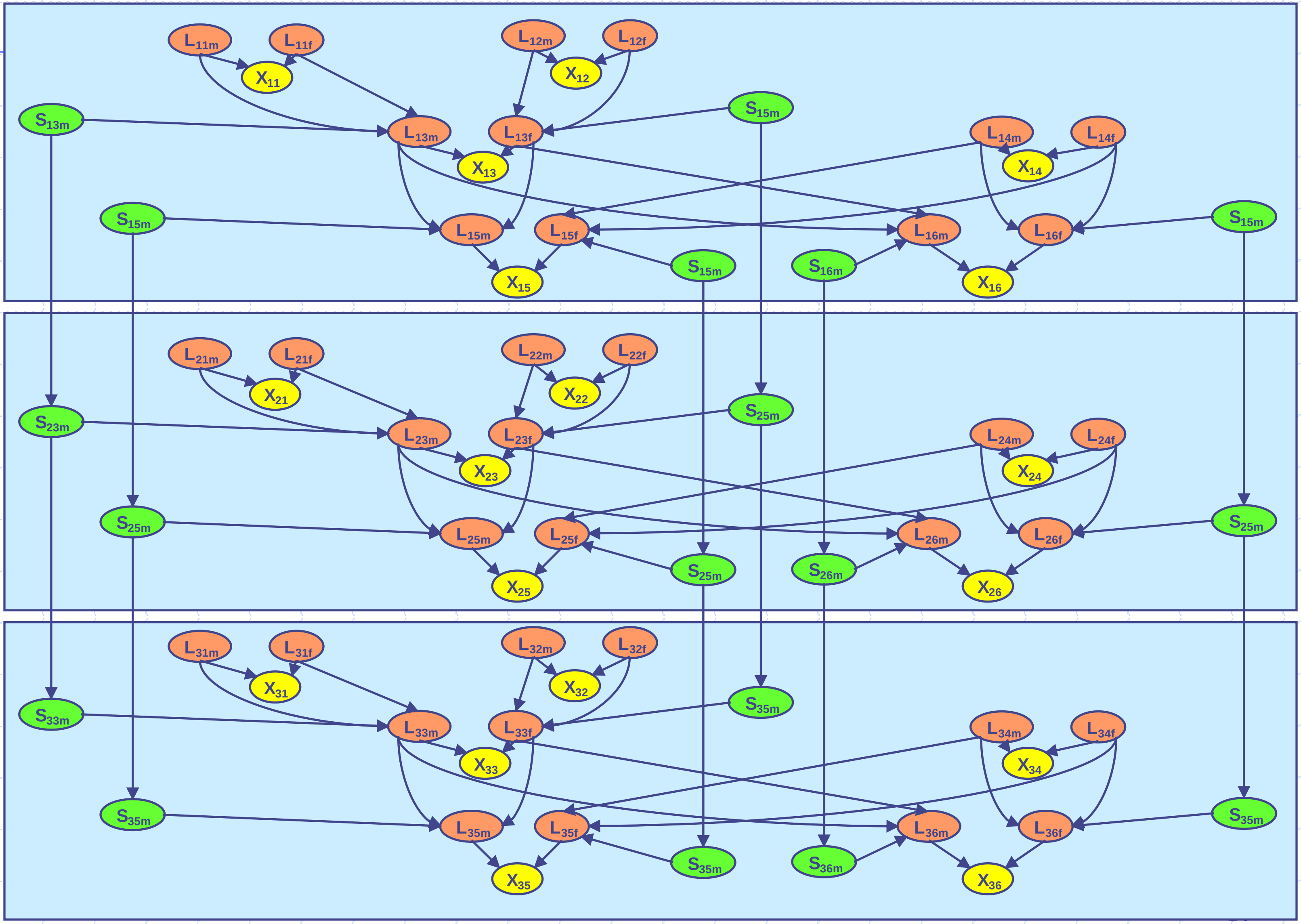
- BTD may use much less memory than Variable Elimination thanks to pruning
- Impact of root cluster
  - ⇒ Choose the largest / most costly cluster as root
- Exploit small separators only (Jégou, Ndiaye & Terrioux, CP 2007)
  - ⇒ Give more freedom for the dynamic variable ordering heuristic
  - ⇒ Tuning based on treewidth versus separator size
- BB-VE(2) often faster than BTD

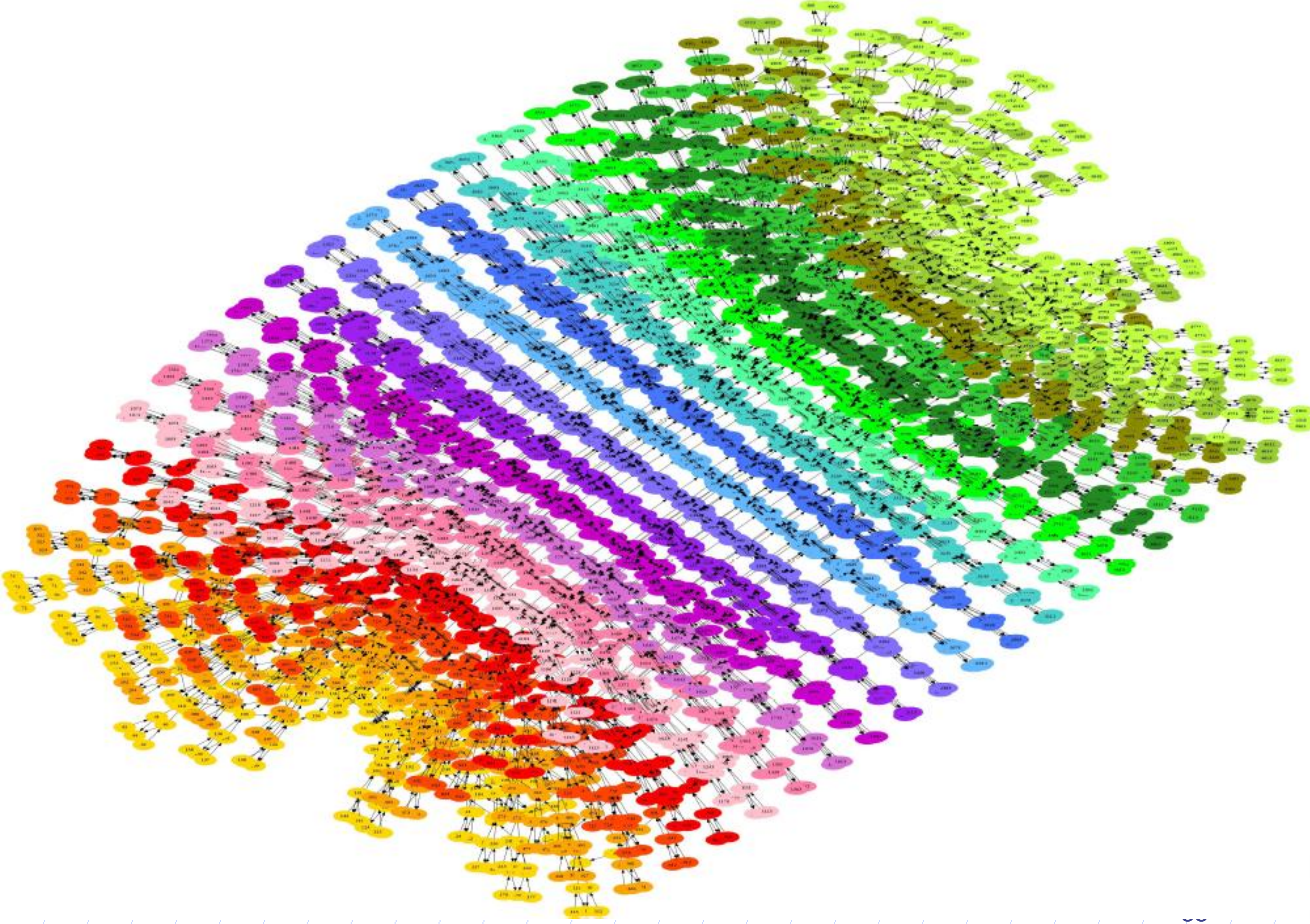
Linkage Analysis



- 6 individuals
- Haplotype: {2, 3}
- Genotype: {6}
- Unknown

Pedigree: 6 people, 3 markers





Genetic Linkage Analysis

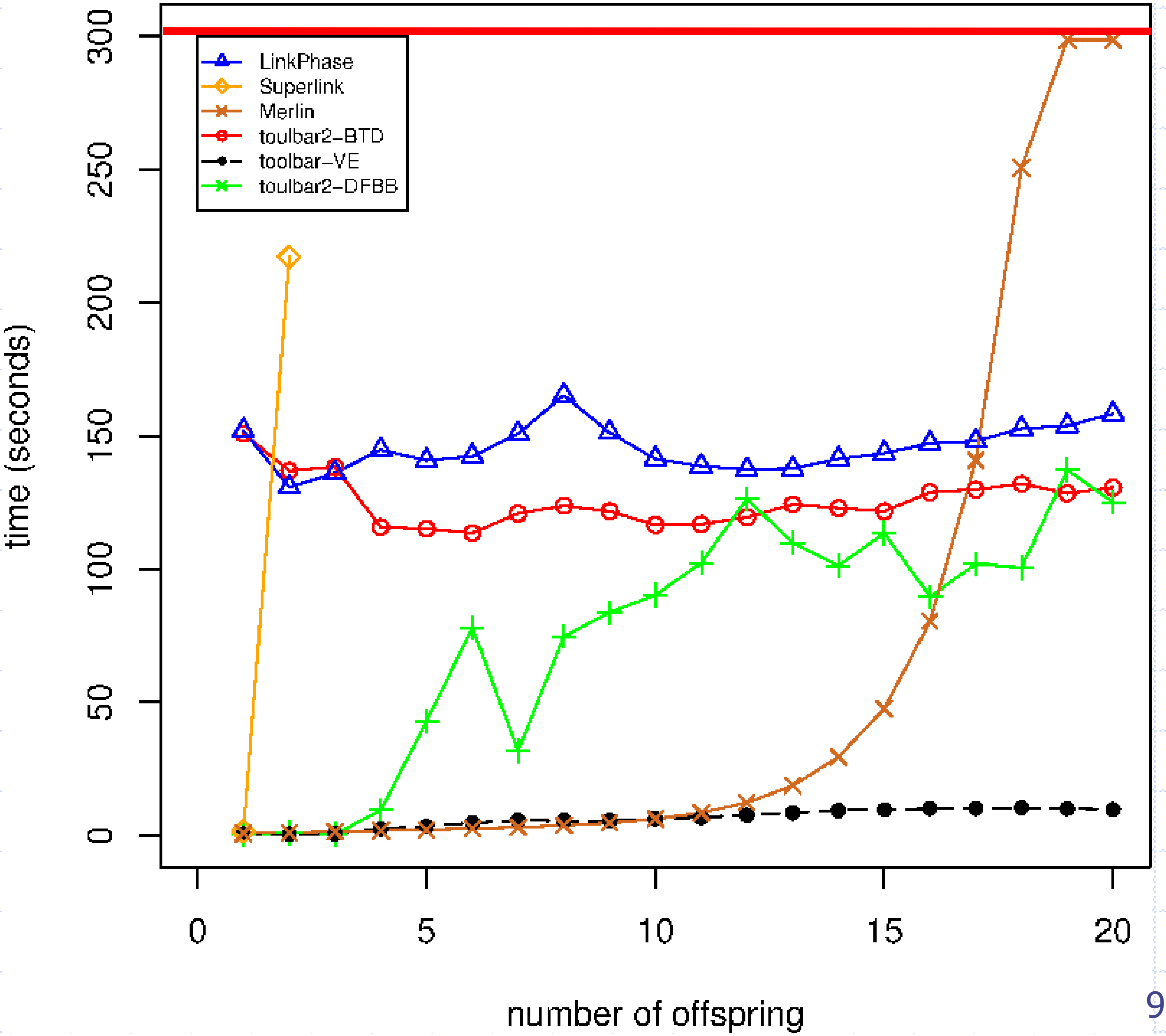
(Fishelson et al, J. Hum. Heredity 05)

pedigree (n, d) (w*, h)	**Superlink** v. 1.6	**SamIam** v. 2.3.2	**MBE(i)** **BB+SMB(i)** **AOBB+SMB(i)** i=12		**MBE(i)** **BB+SMB(i)** **AOBB+SMB(i)** i=16		**MBE(i)** **BB+SMB(i)** **AOBB+SMB(i)** i=20	
	time	time	time	nodes	time	nodes	time	nodes
ped18			0.51		4.59		19.30	
(1184, 5)	139.06	157.05	-	-	-	-	-	-
(21, 119)			-	-	270.96	2,555,078	**20.27**	7,689
ped25			0.34		3.20		33.42	
(994, 5)	-	out	-	-	-	-	-	-
(29, 53)			-	-	-	-	**1894.17**	11,709,153
ped30			0.31		2.66		24.88	
(1016, 5)	13095.83	out	-	-	-	-	-	-
(25, 51)			5563.22	63,068,960	1811.34	20,275,620	**82.25**	588,558
ped33			0.41		5.28		51.24	
(581, 5)	-	out	-	-	-	-	-	-
(26, 48)			2335.28	32,444,818	**62.91**	807,071	76.47	320,279
ped39			0.52		8.41		81.27	
(1272, 5)	322.14	out	-	-	-	-	-	-
(23, 94)			-	-	4041.56	52,804,044	141.23	407,280

Min-fill pseudo tree. Time limit 3 hours.

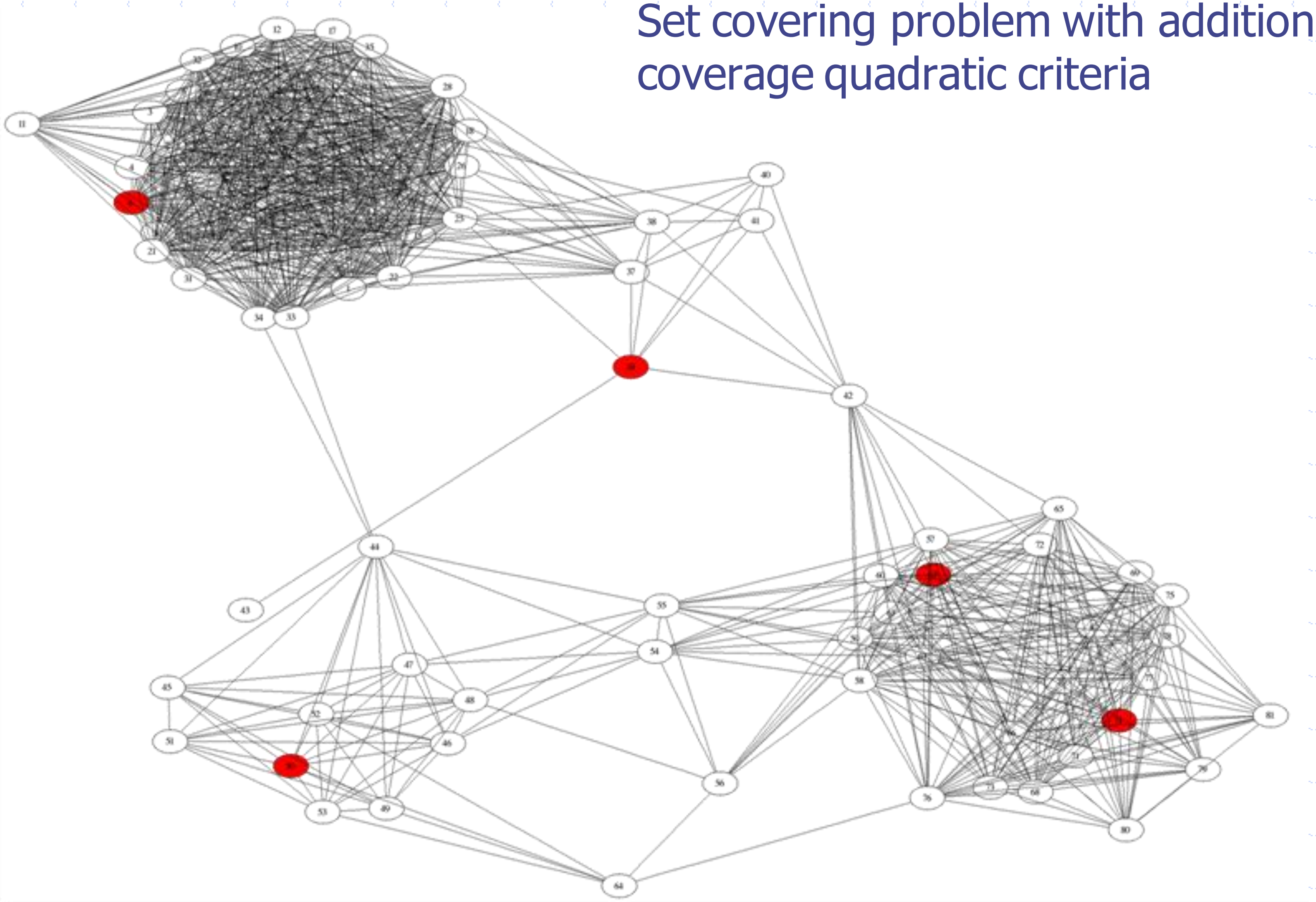
Haplotype reconstruction in half-sib pedigrees (Favier et al, WCB'10)

Read dataset HAPMAP Chr.X with 36,000 markers treewidth < 15



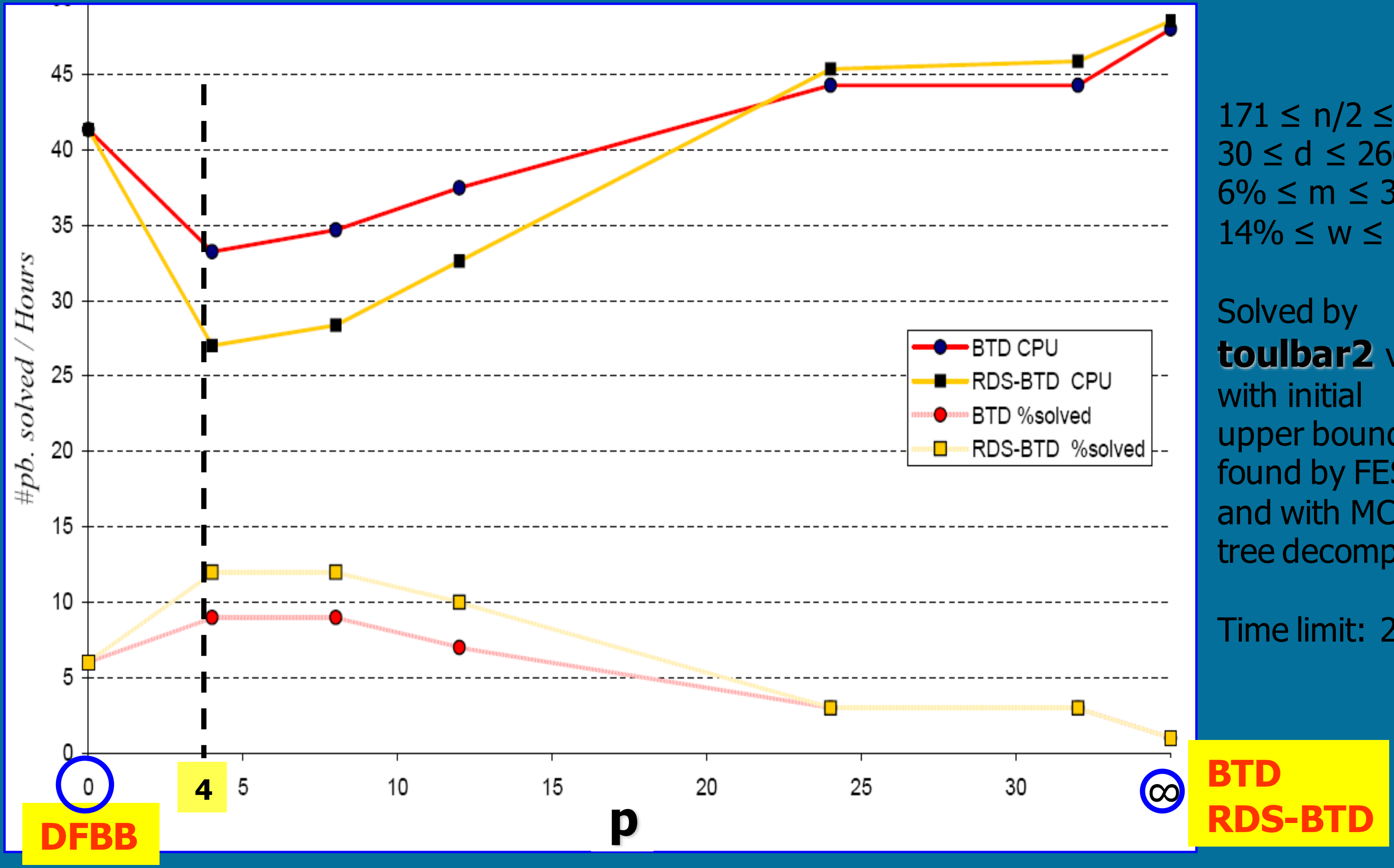
Tag SNP Selection

Set covering problem with additional coverage quadratic criteria



Tag SNP Selection

25 largest instances from HapMap human chr01 ($r^2 \geq 0.5$)

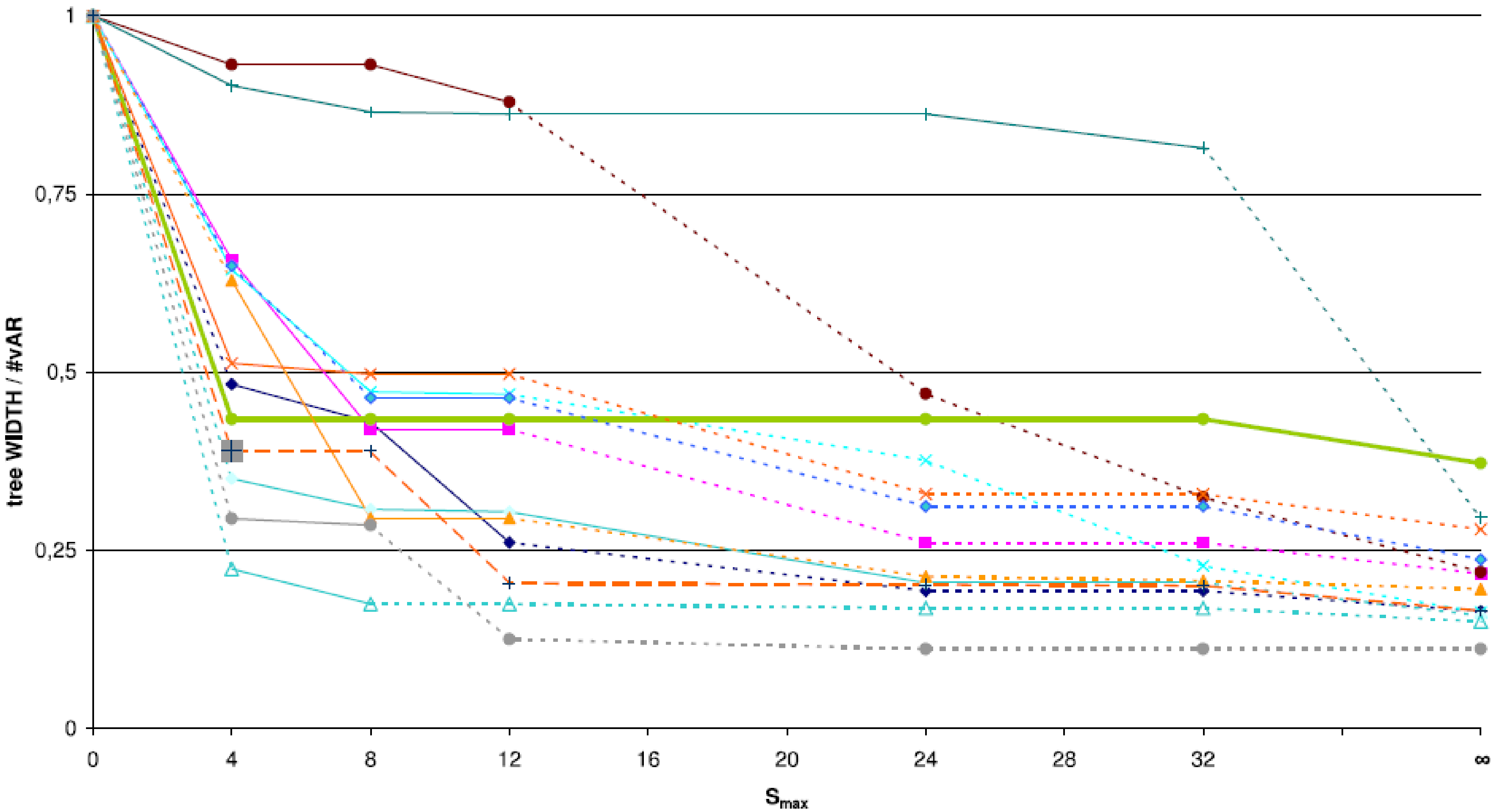


$171 \leq n/2 \leq 777$
$30 \leq d \leq 266$
$6\% \leq m \leq 37\%$
$14\% \leq w \leq 23\%$

Solved by **toulbar2** v.0.8 with initial upper bound found by FESTA and with MCS tree decomposition

Time limit: 2 hours

Tuning the separator size limit for the Tag SNP problem (Allouche et al., WCB'09)



Evolution of treewidth normalized by variable number for different maximum separator size thresholds

Bibliography

- For hybrids of search and inference, see the chapter 10 in *Constraint Processing*, Dechter, Morgan Kaufmann, 2003.
- For exploiting tree decomposition, see
  - "*Exploiting Tree Decomposition and Soft Local Consistency in Weighted CSP*", de Givry, Schiex & Verfaillie , AAAI 2006.
  - "*Memory intensive AND/OR search for combinatorial optimization in graphical models (Part I&II)*", Marinescu & Dechter, AIJ 2009.

http://carlit.toulouse.inra.fr/cgi-bin/awki.cgi/SoftCSP

Applications / benchmarks

toulbar2, aolib

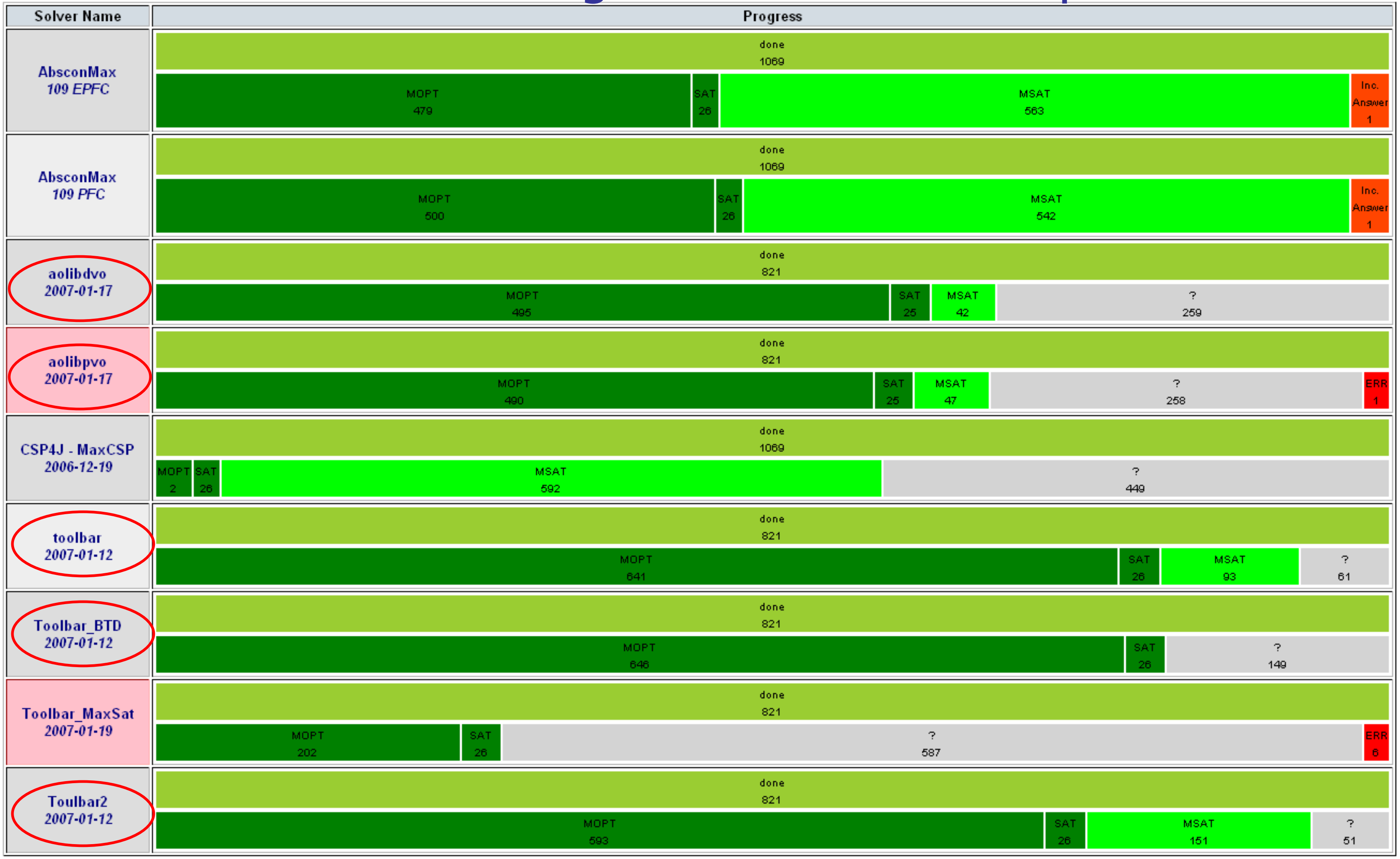
- Resource Allocation
  - Frequency assignment (Allouche et al, CP 2010) CTE, BTD, VAC
    $n \leq 458$, $d=44$, $e(2) \leq 5{,}000$
  - Satellite management (Verfaillie et al, AAAI 1996) RDS, RDS-BTD
    $n \leq 364$, $d=4$, $e(2\text{-}3) \leq 10{,}108$
  - Uncapacitated warehouse location (Zytnicki et al, IJCAI 2005) EDAC,VAC, *ILP0/1*
    $n \leq 1{,}100$, $d \leq 300$, $e(2)$ $100{,}000$
- Bioinformatics
  - Genetic linkage analysis (Marinescu & Dechter, AAAI 2006) AND/ORsearch
    $n \leq 1{,}200$, $d \leq 7$, $e(2\text{-}5) \leq 2{,}000$
  - Mendelian error detection (Sanchez et al, Constraints 2008) EDAC3,BB-VE
    $n \leq 20{,}000$, $d \leq 66$, $e(3) \leq 30{,}000$
  - RNA gene finding (Zytnicki et al, Constraints 2008) BAC
    $n \approx 20$, $d > 100$ million, $e(4) \approx 10$
  - Tag SNP selection (Sanchez et al, IJCAI 2009) RDS-BTD, *ILP0/1*
    $n \leq 1{,}500$, $d \leq 266$, $e(2) \leq 150{,}000$

CP’06 Competitors

- Solvers
  - AbsconMax (ie, DFBB+MRDAC)
  - **aolibdvo** (ie, AOBB+EDAC+DVO solver)
  - **aolibpvo** (ie, AOBB+EDAC+PVO solver)
  - CSP4J-MaxCSP
  - **Toolbar** (ie, DFBB+EDAC)
  - **Toolbar_BTD** (ie, BTD+EDAC+VE)
  - **Toolbar_MaxSAT** (ie, DPLL+specific EPT rules)
  - **Toulbar2** (ie, DFBB+EDAC+VE+LDS)

CP’06 Results

Overall ranking on all selected competition benchmar



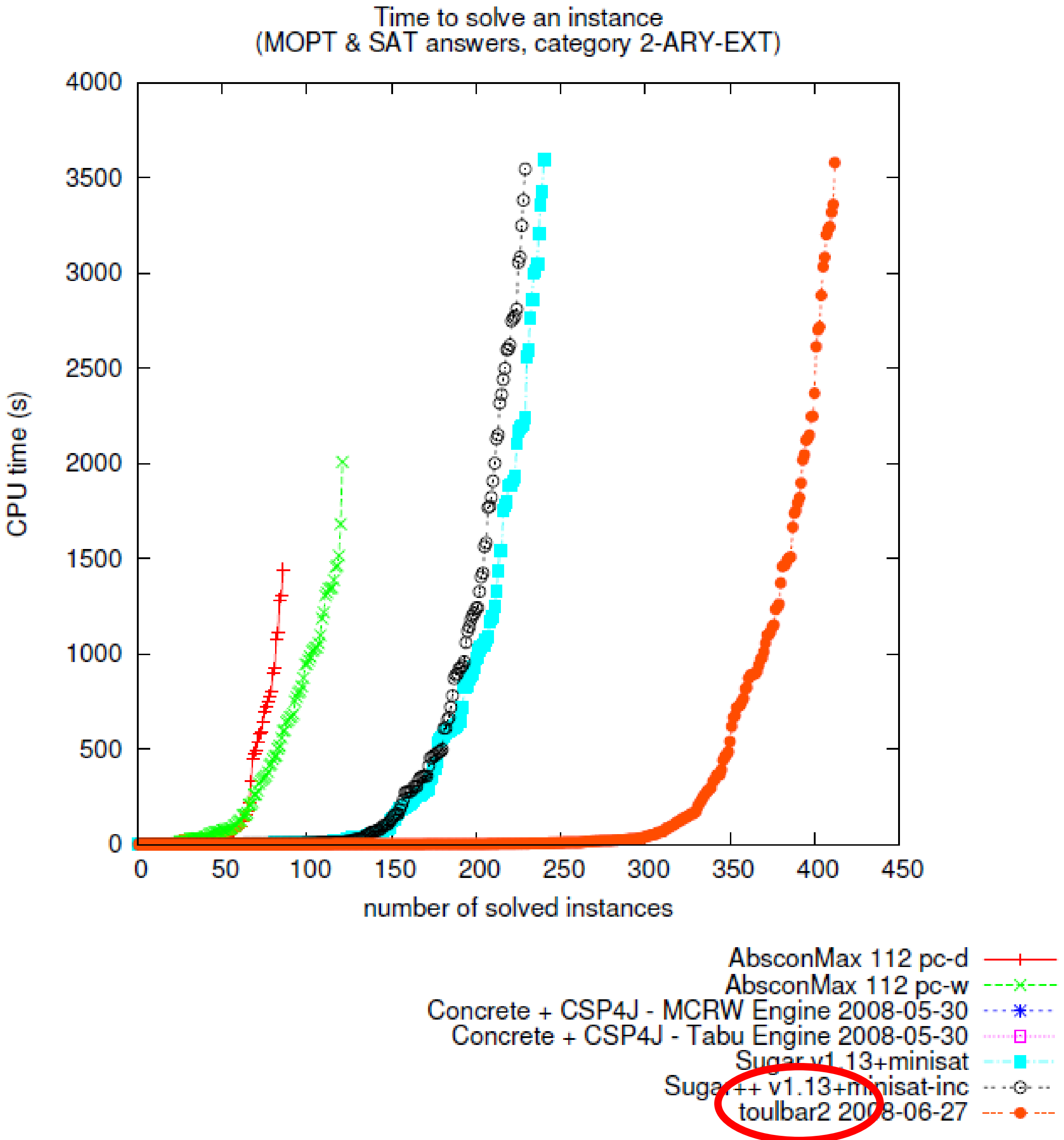
	Solver Name	Progress
	AbsconMax *109 EPFC*	done 1069 MOPT 479 SAT 26 MSAT 563 Inc. Answer 1
	AbsconMax *109 PFC*	done 1069 MOPT 500 SAT 26 MSAT 542 Inc. Answer 1
4	aolibdvo *2007-01-17*	done 821 MOPT 495 SAT 25 MSAT 42 ? 259
5	aolibpvo *2007-01-17*	done 821 MOPT 490 SAT 25 MSAT 47 ? 258 ERR 1
	CSP4J - MaxCSP *2006-12-19*	done 1069 MOPT 2 SAT 26 MSAT 592 ? 449
2	toolbar *2007-01-12*	done 821 MOPT 641 SAT 26 MSAT 93 ? 61
1	Toolbar_BTD *2007-01-12*	done 821 MOPT 646 SAT 26 ? 149
	Toolbar_MaxSat *2007-01-19*	done 821 MOPT 202 SAT 26 ? 587 ERR 6
3	Toulbar2 *2007-01-12*	done 821 MOPT 593 SAT 26 MSAT 151 ? 51

The longest dark green bar wins

CP'08 Competitors

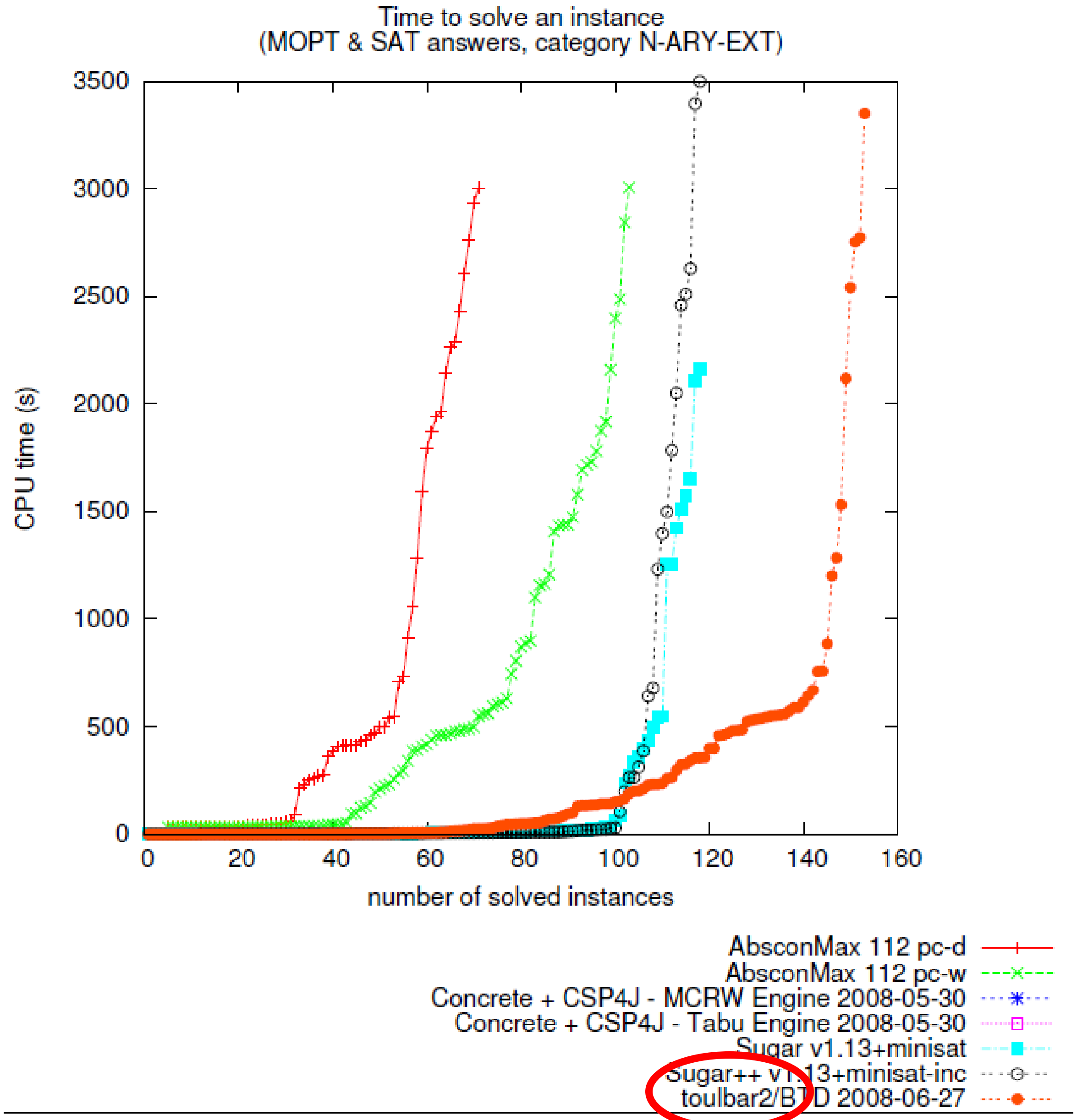
- Solvers
  - AbsconMax (ie, DFBB+MRDAC)
  - CSP4J-MaxCSP
  - Sugar (SAT-based solver)
  - **Toulbar2** (ie, BTD+EDAC+VE)

CP'08 Results



1

CP’08 Results

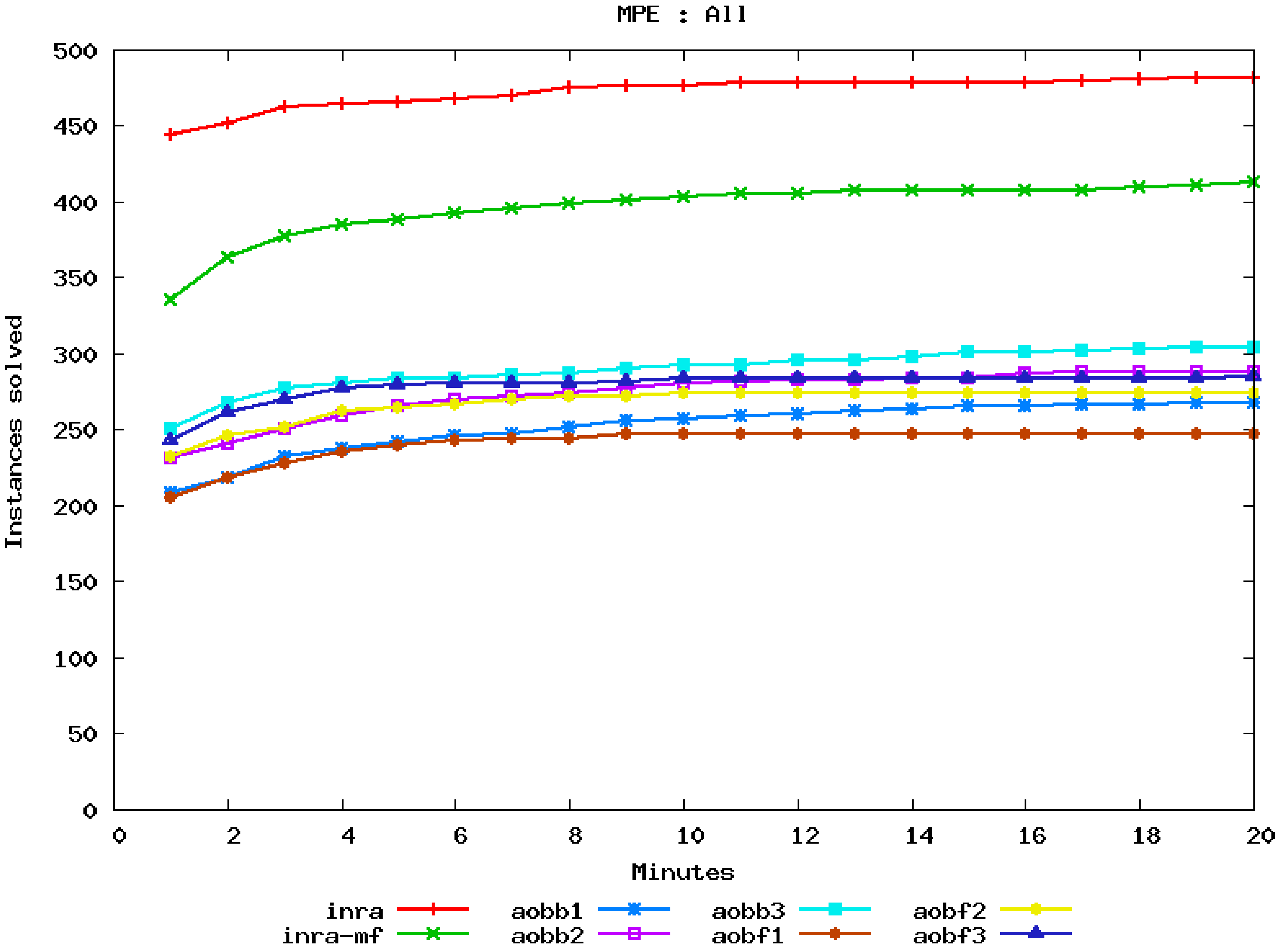


1

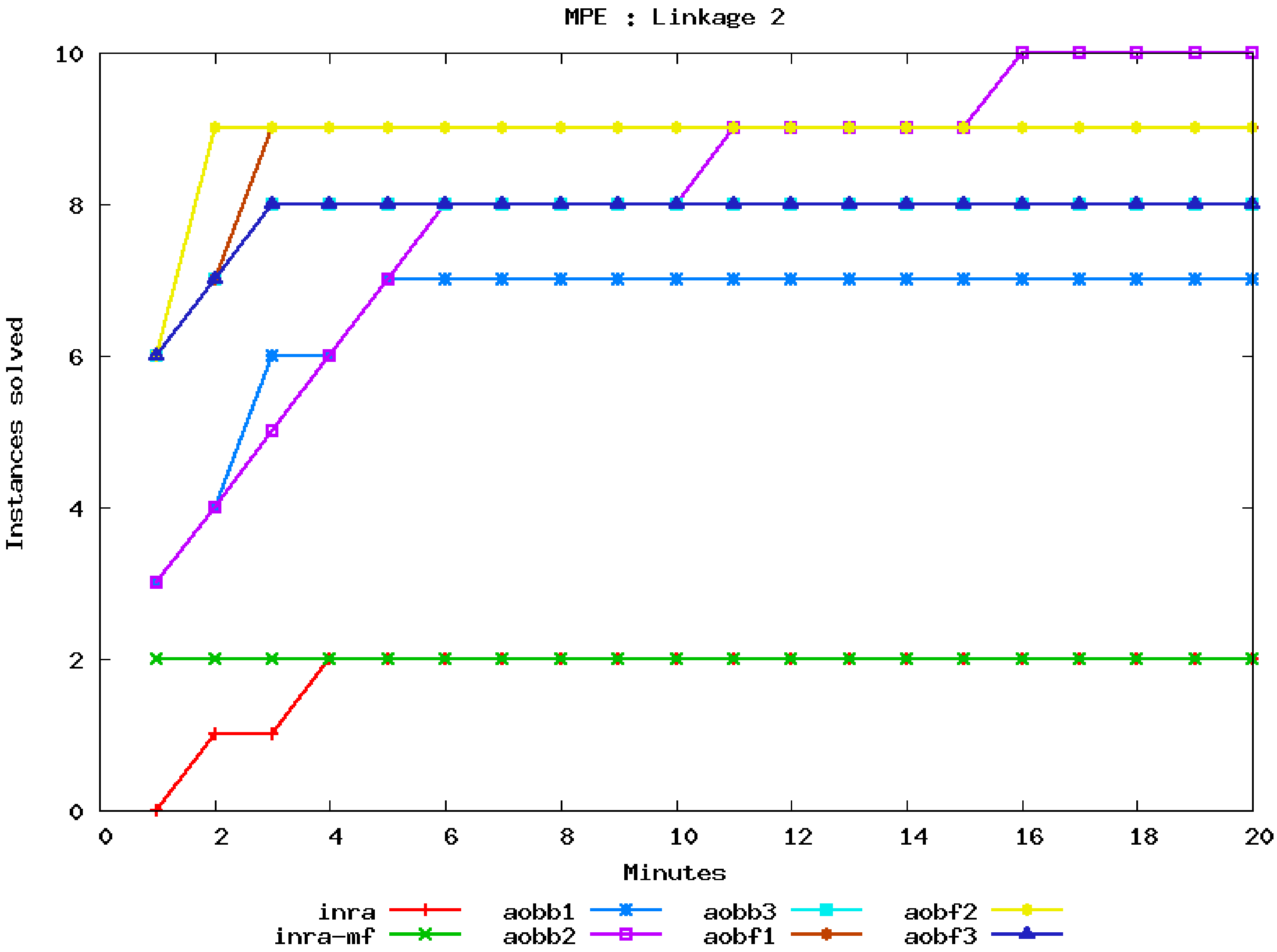
UAI’08 Competition

- **AOBB-C+SMB(i) – (i = 18, 20, 22)**
  - AND/OR Branch-and-Bound with pre-compiled mini-bucket heuristics (i-bound), full caching, static pseudo-trees, constraint propagation
- **AOBF-C+SMB(i) – (i = 18, 20, 22)**
  - AND/OR Best-First search with pre-compiled mini-bucket heuristics (i-bound), full caching, static pseudo-trees, no constraint propagation
- **Toulbar2**
  - OR Branch-and-Bound, dynamic variable/value orderings, EDAC consistency for binary and ternary cost functions, variable elimination of small degree (2) during search
- **Toulbar2/BTD**
  - DFBB exploiting a tree decomposition (AND/OR), same search inside clusters as toulbar2, full caching (no cluster merging), combines RDS and EDAC, and caching lower bounds

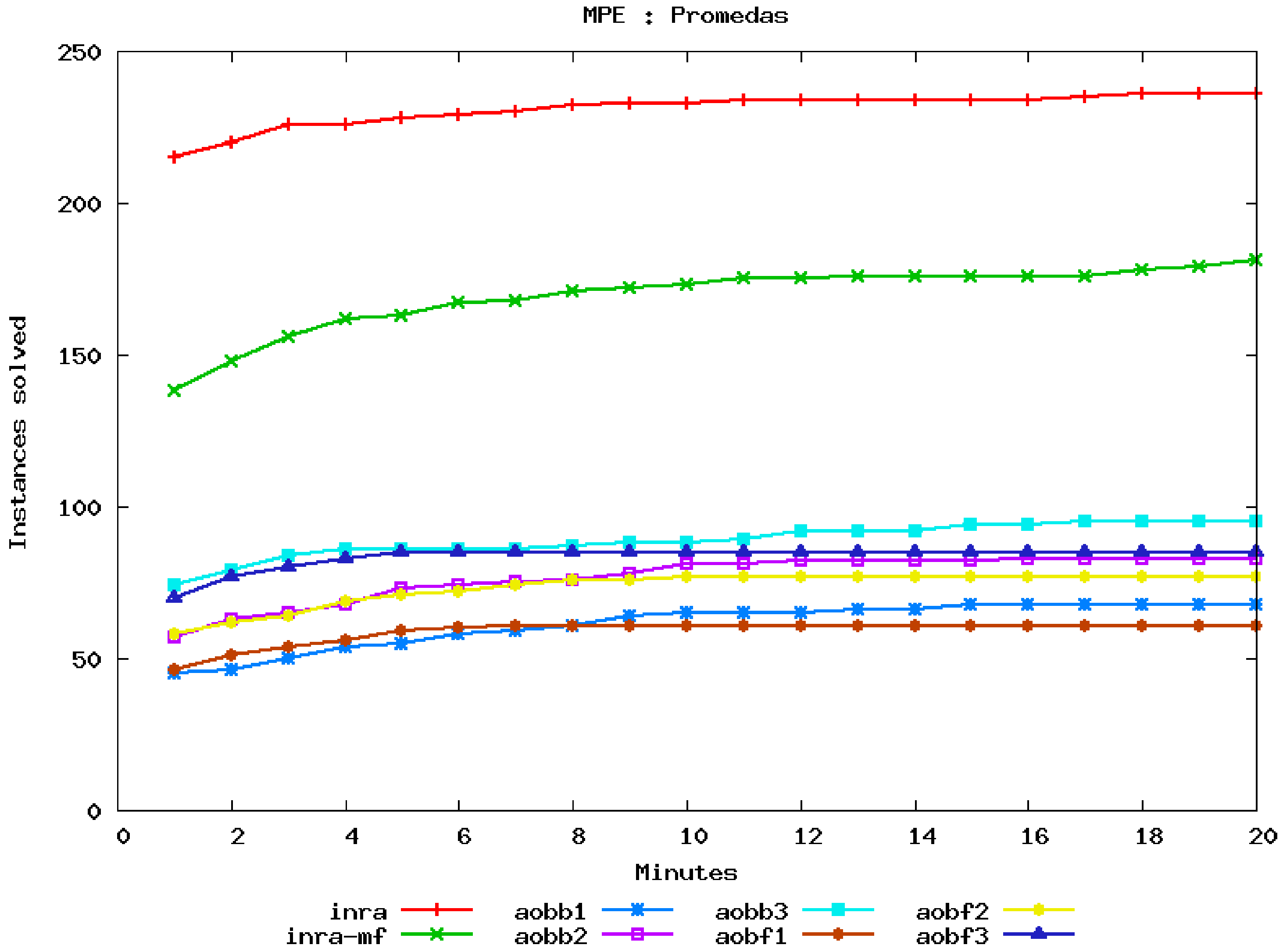
UAI'08 Competition Results



UAI'08 Competition Results (II)



UAI'08 Competition Results (III)



2010 Approximate Inference Evaluation (results given at UAI’10)

Networks – by domain (1 hour)

Network	PR	MAR	MPE
CSP	8	8	55
Grids	20	20	40
Image Alignment			10
Medical Diagnosis	26	26	
Object Detection	96	96	92
Pedigree	4	4	
Protein Folding			21
Protein-Protein Interaction			8
Segmentation	50	50	50

Summary of the results

Seconds	PR	MAR	MPE
20	Arthur Choi (UCLA)	Arthur Choi (UCLA)	Joris Mooij (Max Planck)
1200	Vibhav Gogate (UW+UCI)	Vibhav Gogate (UW+UCI)	Thomas Schiex (INRA)
3600	Vibhav Gogate (UW+UCI)	Vibhav Gogate (UW+UCI)	Joris Mooij (Max Planck)

toulbar2 was also first at UAI'08 Evaluation, MaxCSP'06,'08 Competition

Winning Teams

- (MAR) **IJGP** by Vibahv Gogate (UW), Andrew Gefland, Natasha Flerova and Rina Dechter (UCI): **Anytime** iterative GBP based algorithm
- (PR) **Vgogate** by Vibahv Gogate, Pedro Domingos (UW),, Andrew Gefland and Rina Dechter (UCI): Formula based importance sampling
- (PR+MAR) **EDBP** by Arthur Choi, Adnan Darwiche, with support from Glen Lenker and Knot Pipatsrisawat (UCLA): **Anytime** BP based anytime thickening of structure
- (MAP) **libDAI** by Joris Mooij (Max Planck): junction tree, LBP/MP, double-loop GBP, Gibbs, decimation
- (MAP) **toulbar2** by Thomas Schiex et al (INRA) **Anytime** branch and bound weighted CSP solver

73% instances solved exactly

Toulbar2 & aolib & benchmarks

- toulbar2

  http://mulcyber.toulouse.inra.fr/projects/toulbar2

  (Open source WCSP, MPE solver in C++ and counting/PR)

- aolib

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

  (WCSP, MPE, ILP solver in C++, inference and counting)

- Large set of benchmarks

  http://carlit.toulouse.inra.fr/cgi-bin/awki.cgi/SoftCSP

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

  http://www.cril.univ-artois.fr/~lecoutre/benchmarks.html