# Combinatorial Optimization for Life Sciences

**Simon de Givry**
SaAB team, MIAT – INRA
Toulouse, France

November 1st 2013

Unité de Mathématiques et Informatique Appliquées Toulouse

**Direction**

**Directeur d'Unité : R. Sabbadin (DR)**

**Directeur d'Unité Adjoint : C. Cierco-Ayrolles (CR)**

**ÉQUIPE COLLECTIVE (ID & AU)**

**Secrétariat**

F. Ayrignac (TR) – N. Julliand (TR) - B. Nguyen Van Thanh (TR)

email : secretariat-miat@toulouse.inra.fr

**Informatique**

**A. Boukhairi (AI)** – M. Grialou (TR)

email : informatic-miat@toulouse.inra.fr

**MAD**

S. Couture (CR)
R. Faivre (DR)
S. Jasson (IR)
R. Martin-Clouaire (DR)
N. Peyrard (CR)
V. Picheny (CR)
G. Quesnel (CR)
**R. Sabbadin (DR)**

Doctorants :
J. Radoszycki

Post-doctorant:
W. Probert

**SaAB**

D. Allouche (IR)
H. Chiapello (IR)
C. Cierco-Ayrolles (CR)
M.-J. Cros (IE)
C. Gaspin (DR)
S. de Givry (CR)
G. Katsirelos (CR)
D. Leroux (IE)
A. Moisan (IE)
**T. Schiex (DR)**
M. Vignes (CR)
M. Zytnicki

Doctorants :
M. Champion
H. Nguyen

Post-doctorant
J. Davies

MCF en délégation
N. Villa-Vialaneix

ATER
J. Vandel

**Plateforme BIOINFORMATIQUE**

**C. Gaspin (DR)**
C. Hoede (IR)
**C. Klopp (IR)**
D. Laborie (IE)
J. Mariette (IE)
C. Noirot (IE)

CDD :
M.-S. Trotard (IE)

**GENOPOLE**

M. Falières (IE)

**Plateforme RECORD**

E. Casellas (IE)
P. Chabrier (IE)
**H. Raynal (IR)**
N. Rousse (IR)
R. Trepos (IR)

**Plateforme SIGENAE**

**C. Klopp (IR)**

CDD :
K. Barrilliot (IE)
A. Djari (IE)
A. Leleu (IE)
I. Nabihoudine (IE)
T. Raulet (IE)
O. Rué (IE)
P. Senin (IE)

# PLAN

Combinatorial Optimization expressed as
Weighted Constraint Satisfaction Problem (WCSP)

- Applications in Genetics
  - Mendelian error detection in complex pedigrees
  - *Haplotype reconstruction in half-sib families*
  - *TagSNP selection*
  - *Optimizing the reference population in a genomic selection design*
  - *Multipopulation integrated genetic and radiated hybrid mapping (carthagene)*

- Applications in Bioinformatics
  - Searching RNA motifs and their intermolecular contacts
  - Computational protein design
  - *Gene regulatory network reconstruction using bayesian networks*

# Mendelian error detection in complex pedigree

Simon de Givry, Marti Sanchez, Isabelle Palhière, Zulma Vitezica, and Thomas Schiex

INRA, Toulouse, France

WCB 2005, WCGALP 2006, Constraints 2008

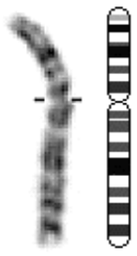
# Cleaning the data after genotyping

Today, about 1% errors remain after SNP genotyping

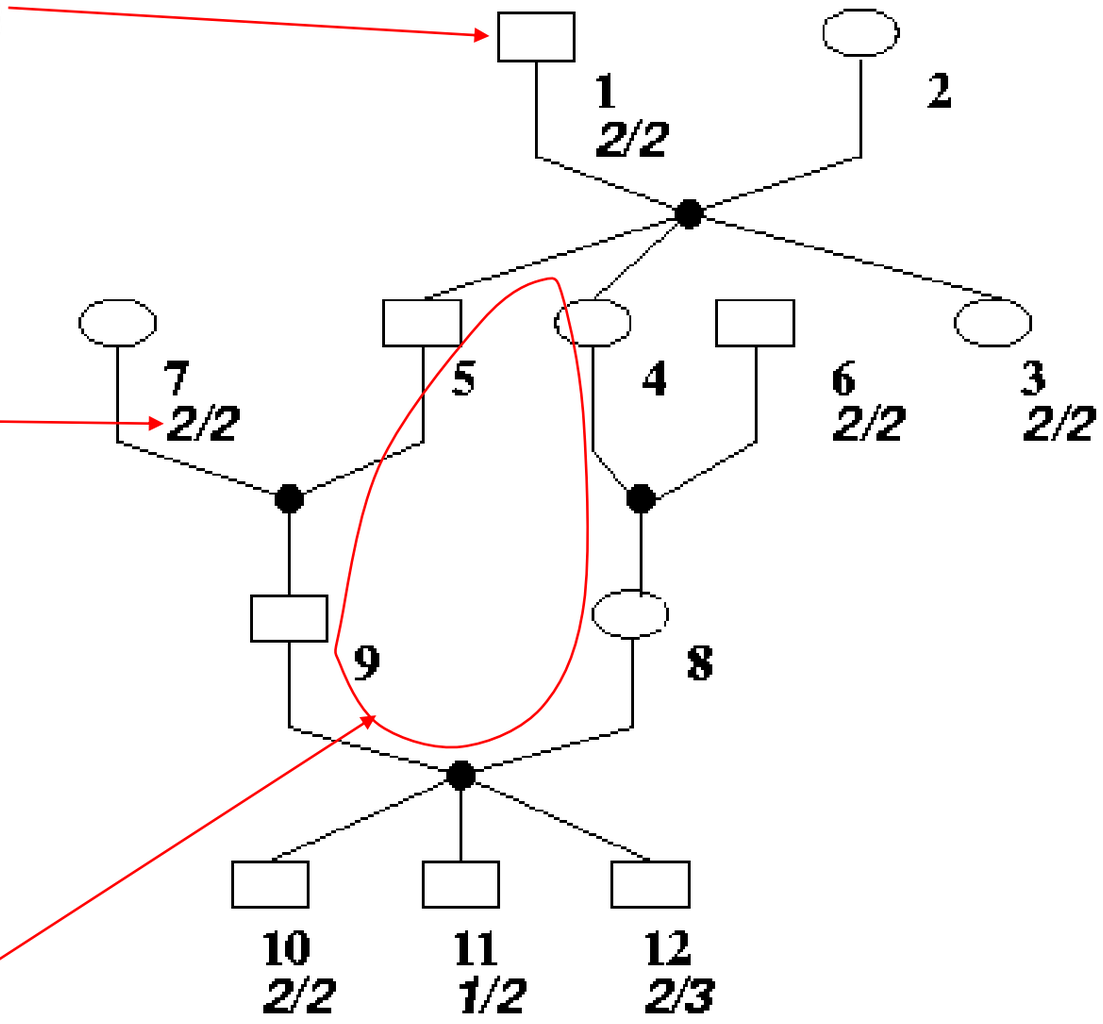
→ For each marker, detects Mendelian inheritance errors

Individual (founder)

Genotype: unordered pair of alleles (possibly unobserved)

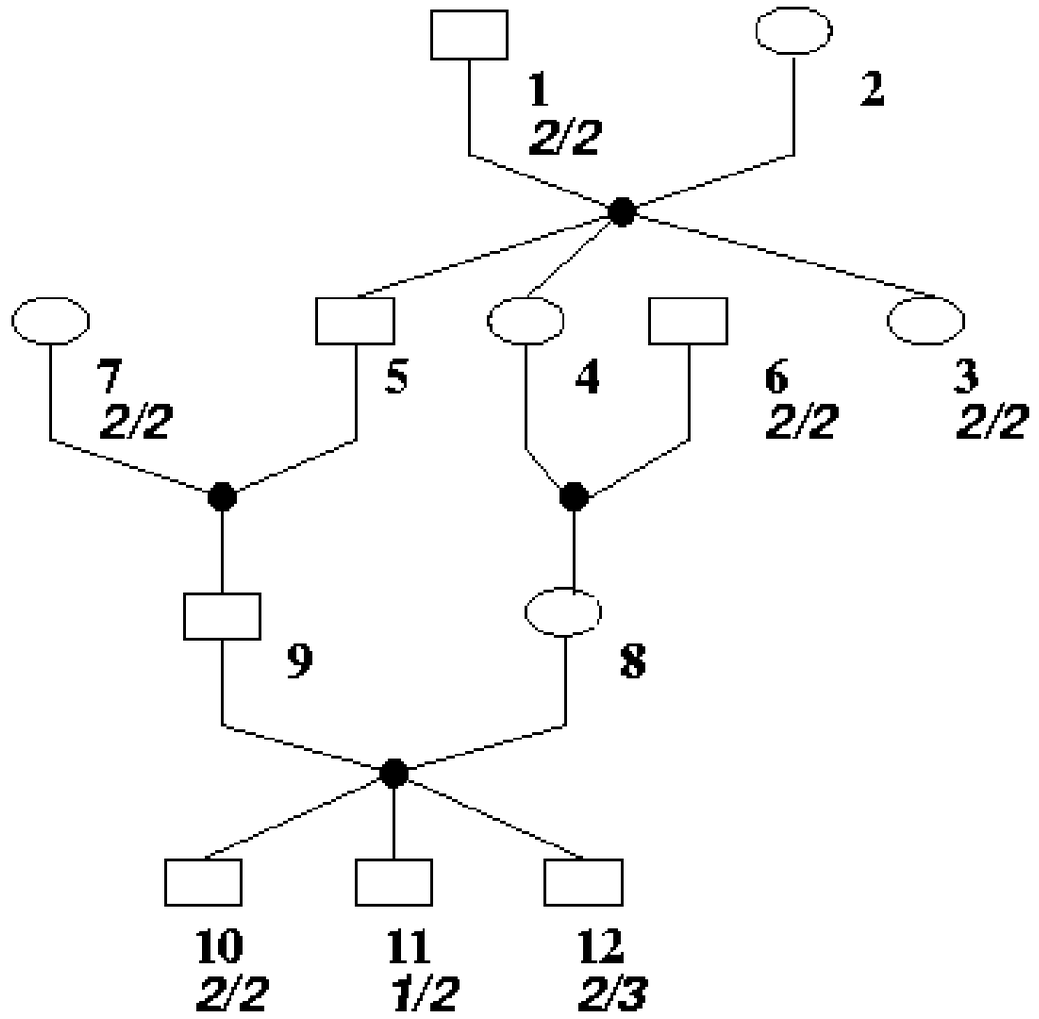


Marriage loop



# Task 1: Consistency Checking

- Assuming the pedigree is correct, checks if it exists a complete genotype assignment consistent with the observed genotypes and with the Mendelian laws of inheritance

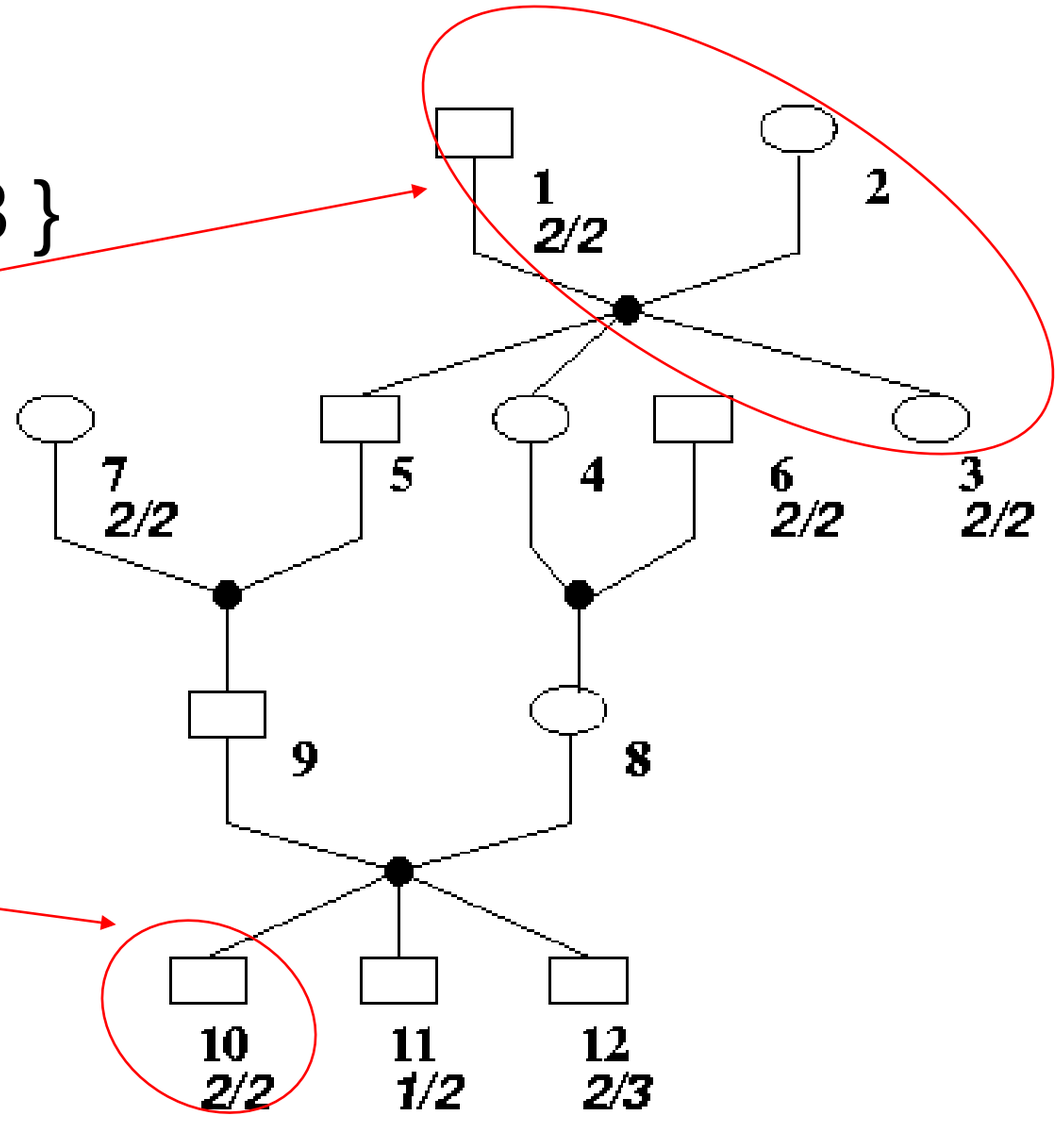



- Complexity results (Aceto et al., 2003)
  - NP-complete for a pedigree with loops and more than three alleles
  - Polynomial if no loops or just two alleles

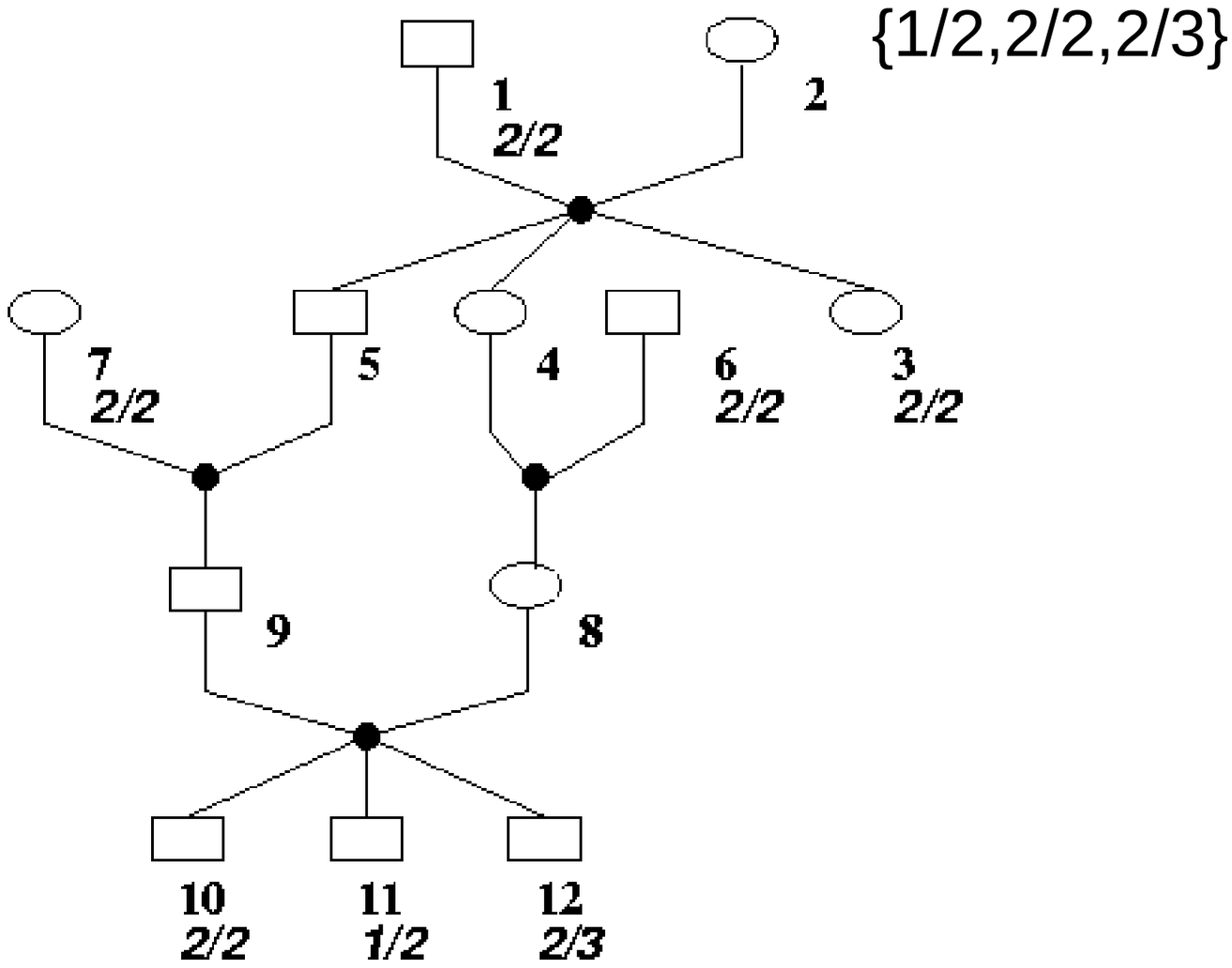
# Constraint Satisfaction Problem (X,D,C)

- **X**: one variable per individual
- **D**: domain of every variable is defined as the set of all possible genotypes

  Here: { 1/1, 1/2, 1/3, 2/2, 2/3, 3/3 }
- **C**:
  - Ternary constraints to encode Mendelian laws for any non founder
  - Unary constraints to encode genotyping data

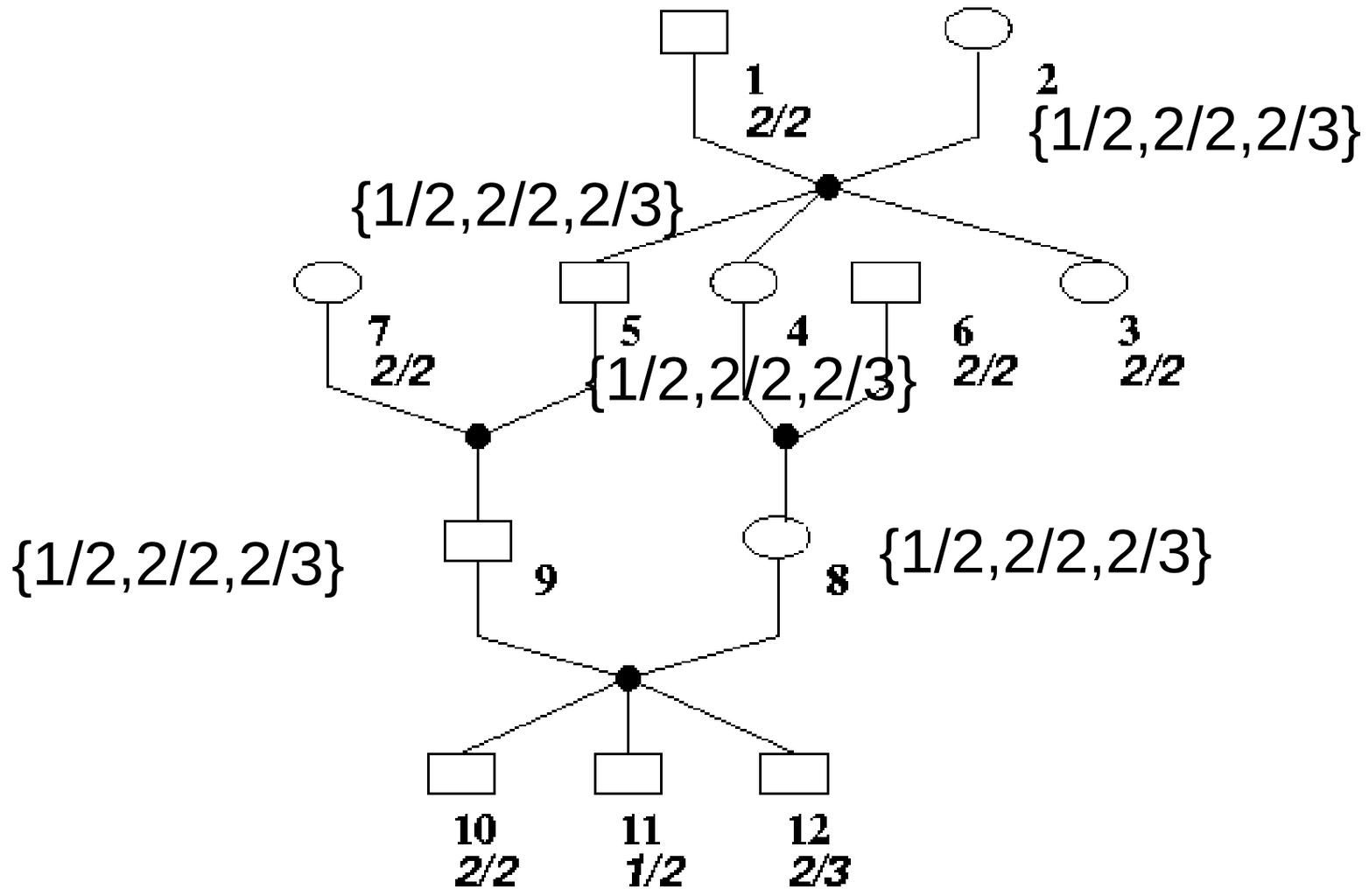


# Generalized Arc Consistency



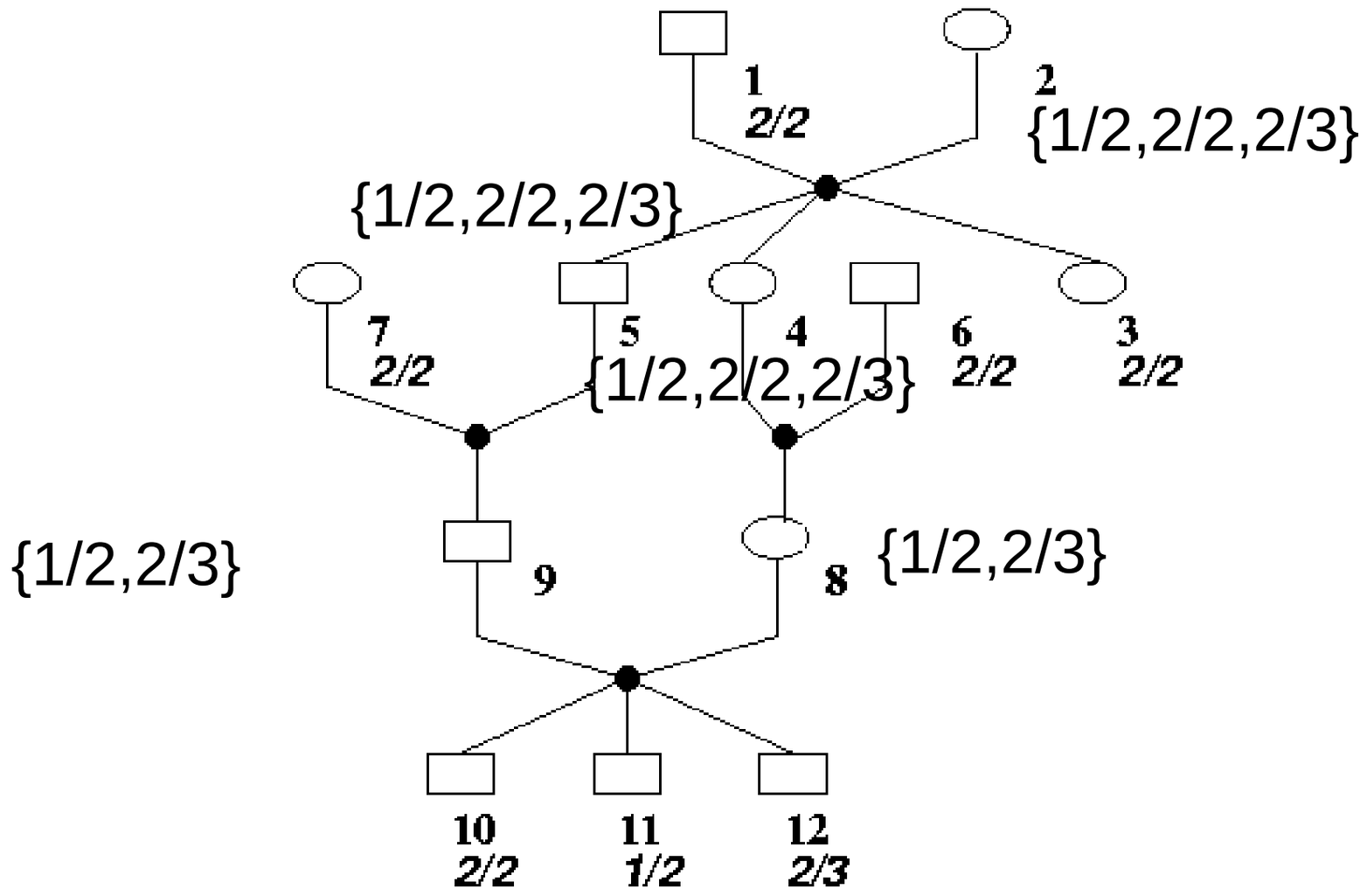

(Mackworth, AIJ 1977)

# Generalized Arc Consistency



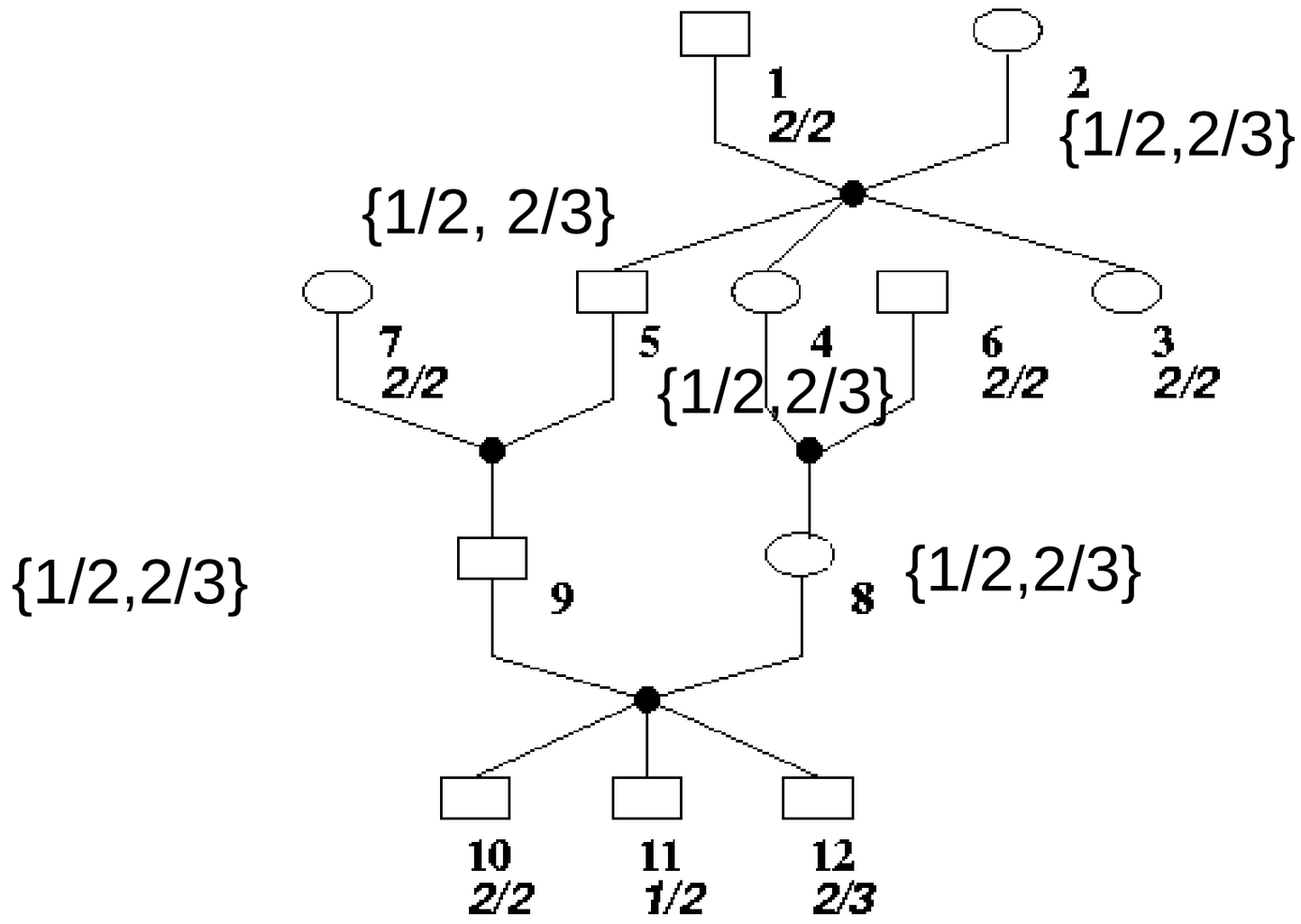

(Mackworth, AIJ 1977)

# Generalized Arc Consistency on nuclear family



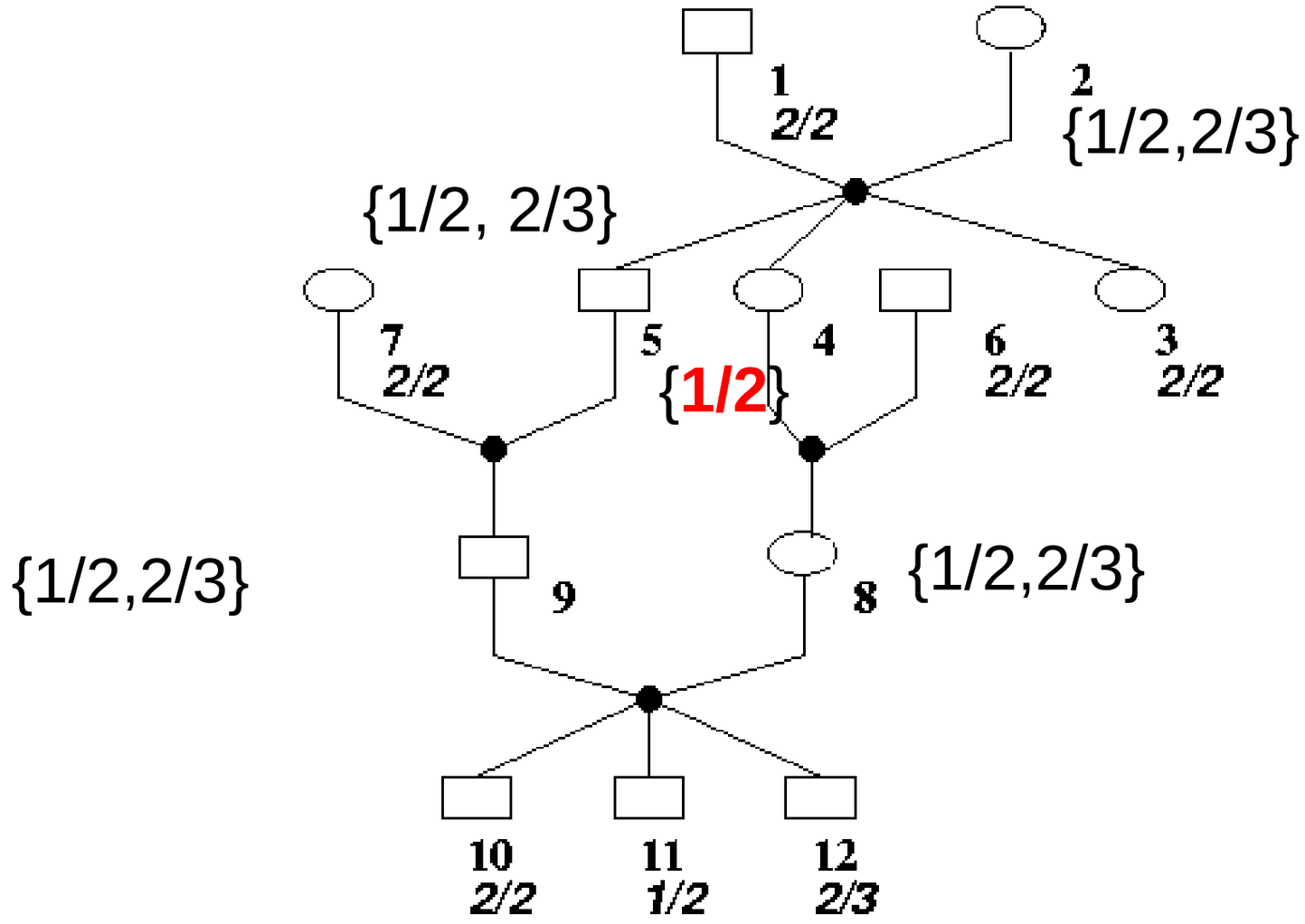

(Lange, Goradia, Am J Hum Genet 1987)

# Generalized Arc Consistency on nuclear family



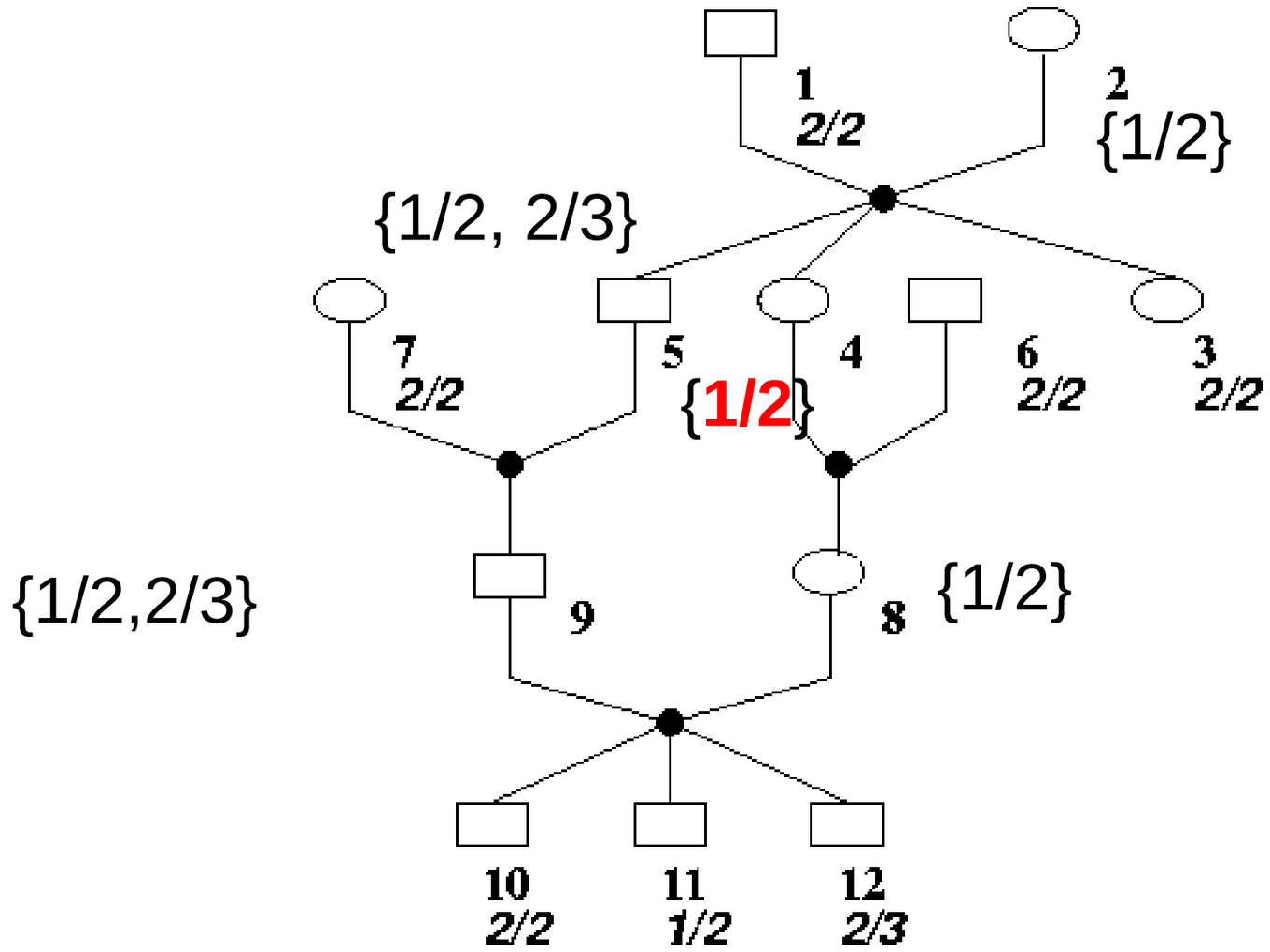

(Lange, Goradia, Am J Hum Genet 1987)

# Backtrack search on loop-breaker individuals



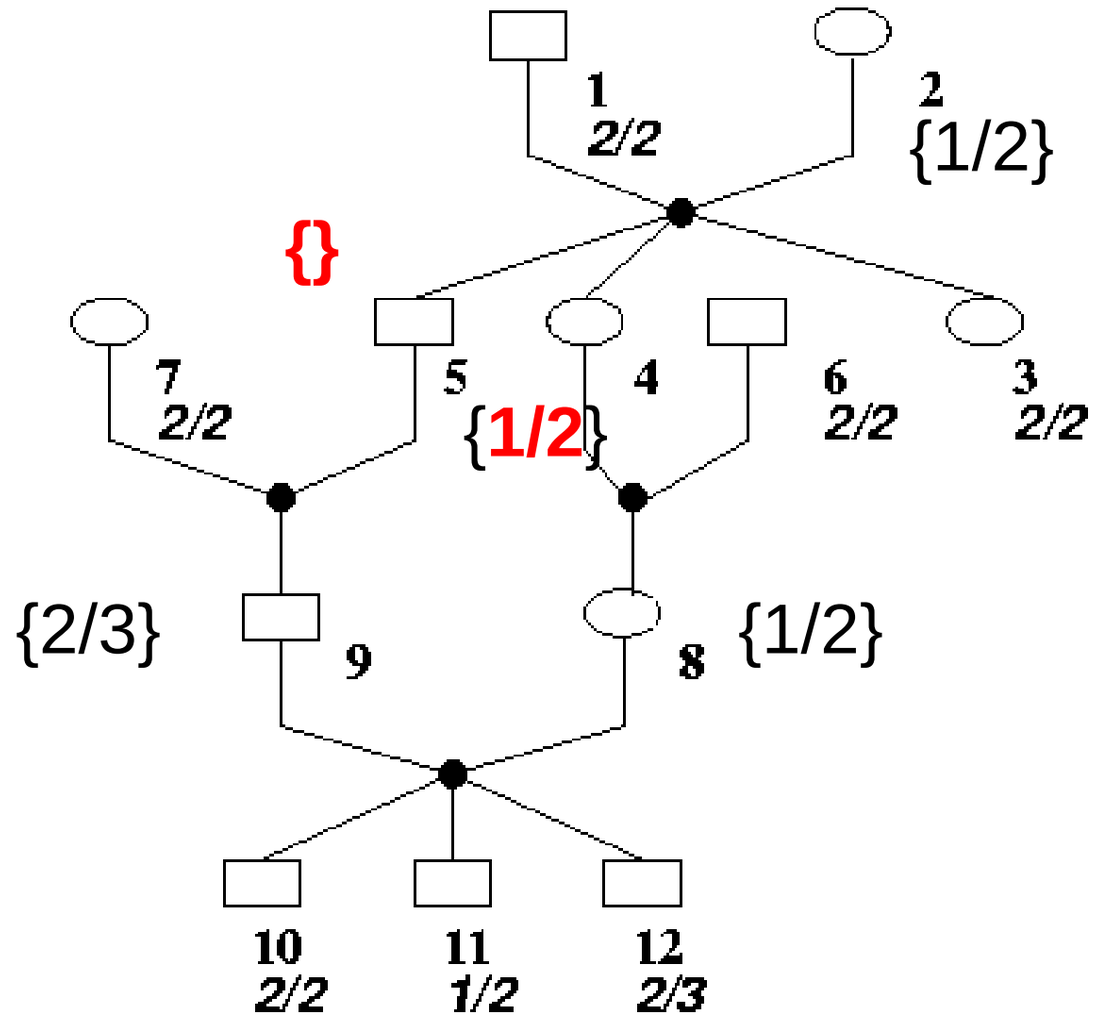

(O'Connell, Weeks, Am J Hum Genet 1997)
(Dechter, AIJ 1990)

# Backtrack search on loop-breaker individuals



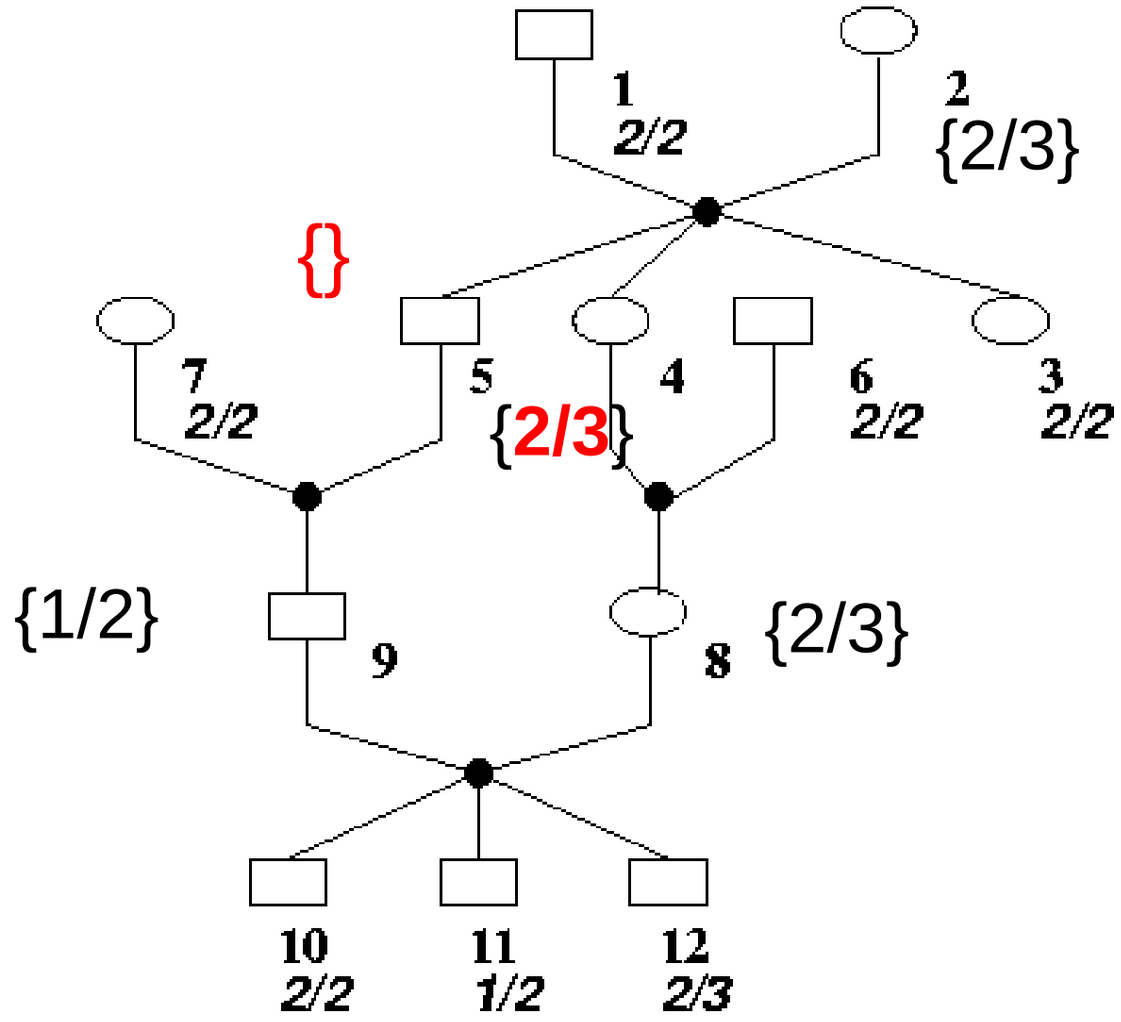

(O'Connell, Weeks, Am J Hum Genet 1997)
(Dechter, AIJ 1990)

# Backtrack search on loop-breaker individuals




(O'Connell, Weeks, Am J Hum Genet 1997)
(Dechter, AIJ 1990)

# Backtrack search on loop-breaker individuals

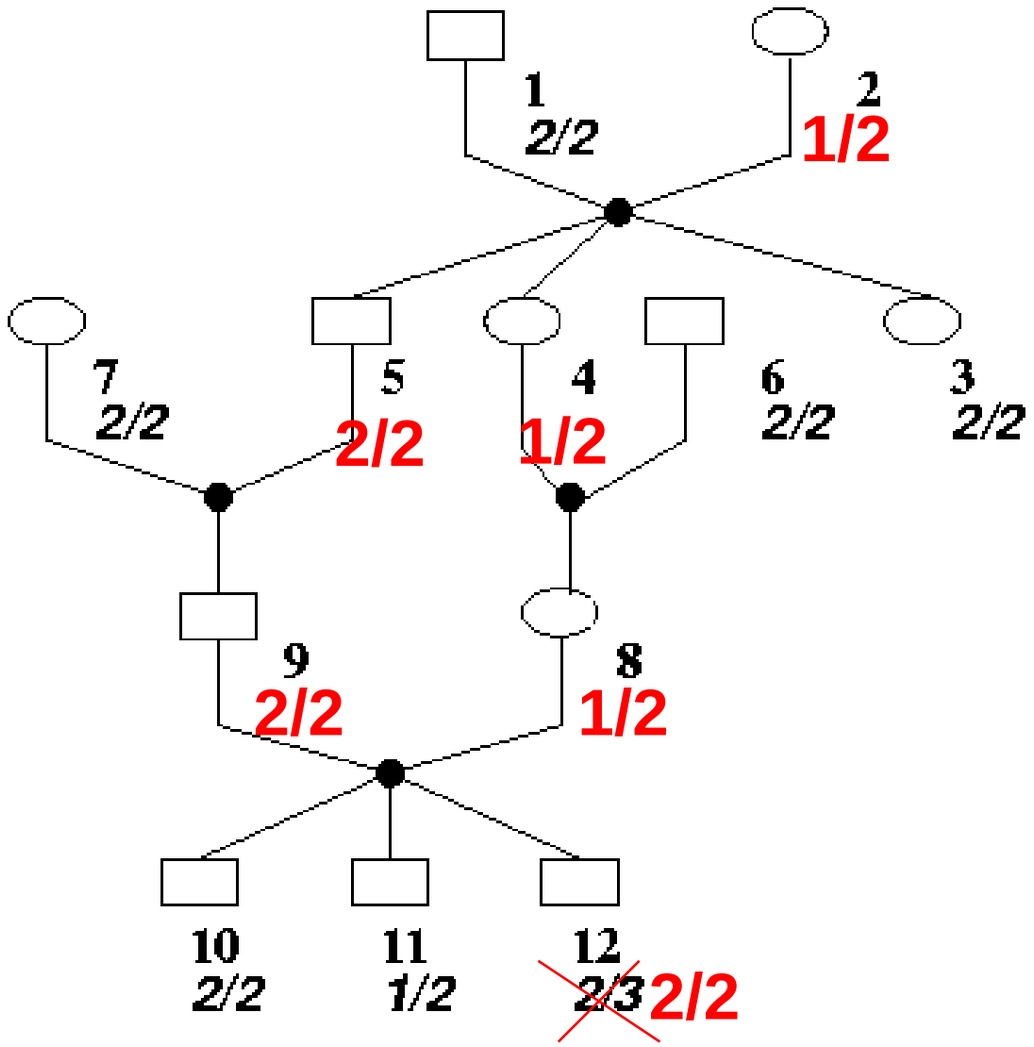



(O'Connell, Weeks, Am J Hum Genet 1997)
(Dechter, AIJ 1990)

# Task 2: Error Detection

- Finds a complete assignment with the minimum number of errors

→Follows the parsimony principle



# Weighted CSP (WCSP)

(Shapiro, Haralick, IEEE PAMI 81)
(Freuder, Wallace, AIJ 92)
(Schiex, Fargier, Verfaillie, IJCAI 95)

- $(X, D, F)$
  - $X=\{X_1,\ldots, X_n\}$ — ***n*** variables
  - $D=\{D_1,\ldots, D_n\}$ — ***n*** finite domains of maximum size ***d***
  - $F=\{f_{S_1},\ldots,f_{S_e}\}$, — ***e*** cost functions

    $f_{S_i}$ : associates a finite or infinite (***k***) positive integer to every tuple in $l(S_i)$
- Goal: find a complete assignment $A$ minimizing

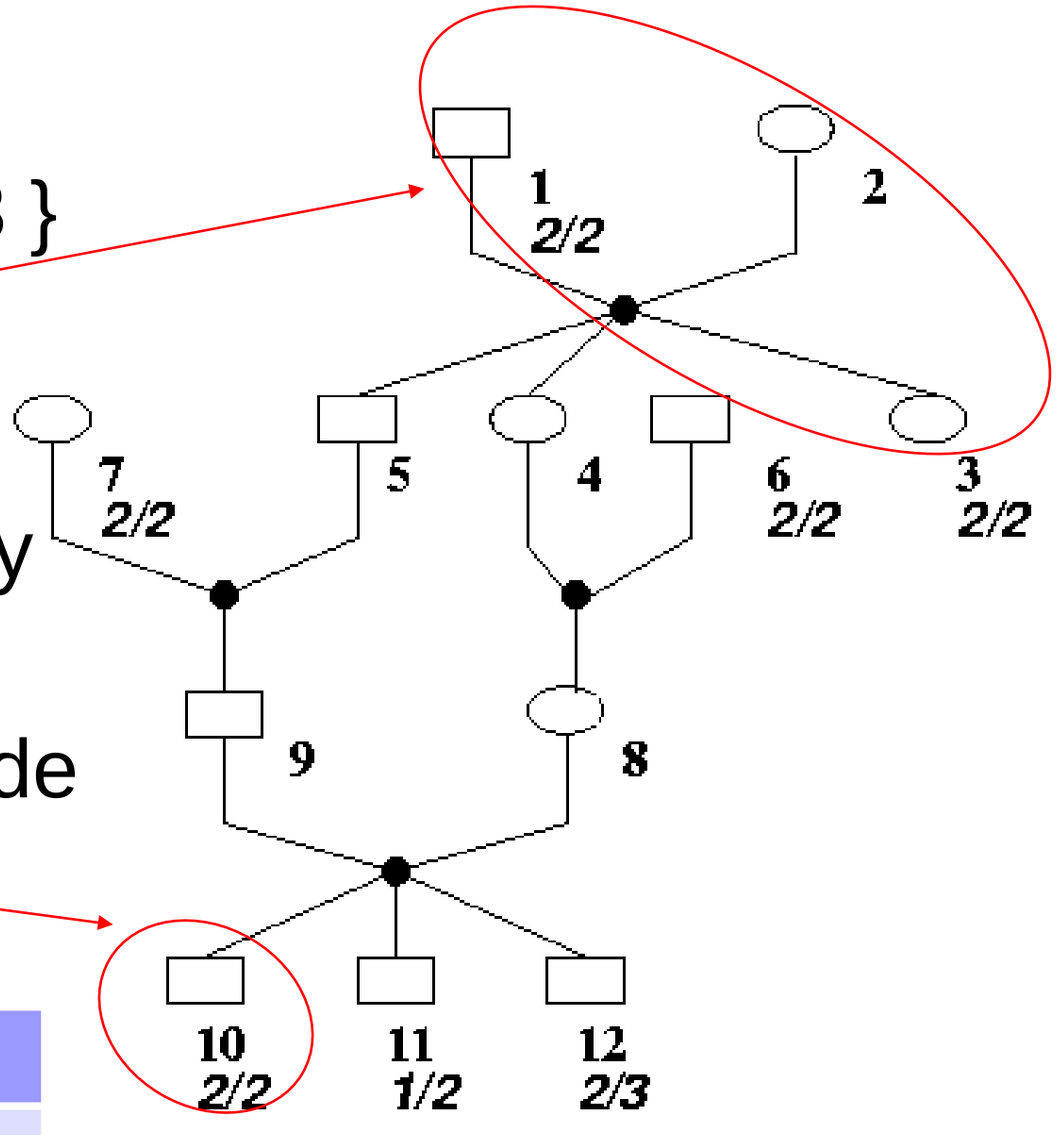
$$\sum_{f_S \in F} \; f_S ( A[S] )$$

NP-hard problem

# Weighted Constraint Satisfaction Problem (X,D,F)

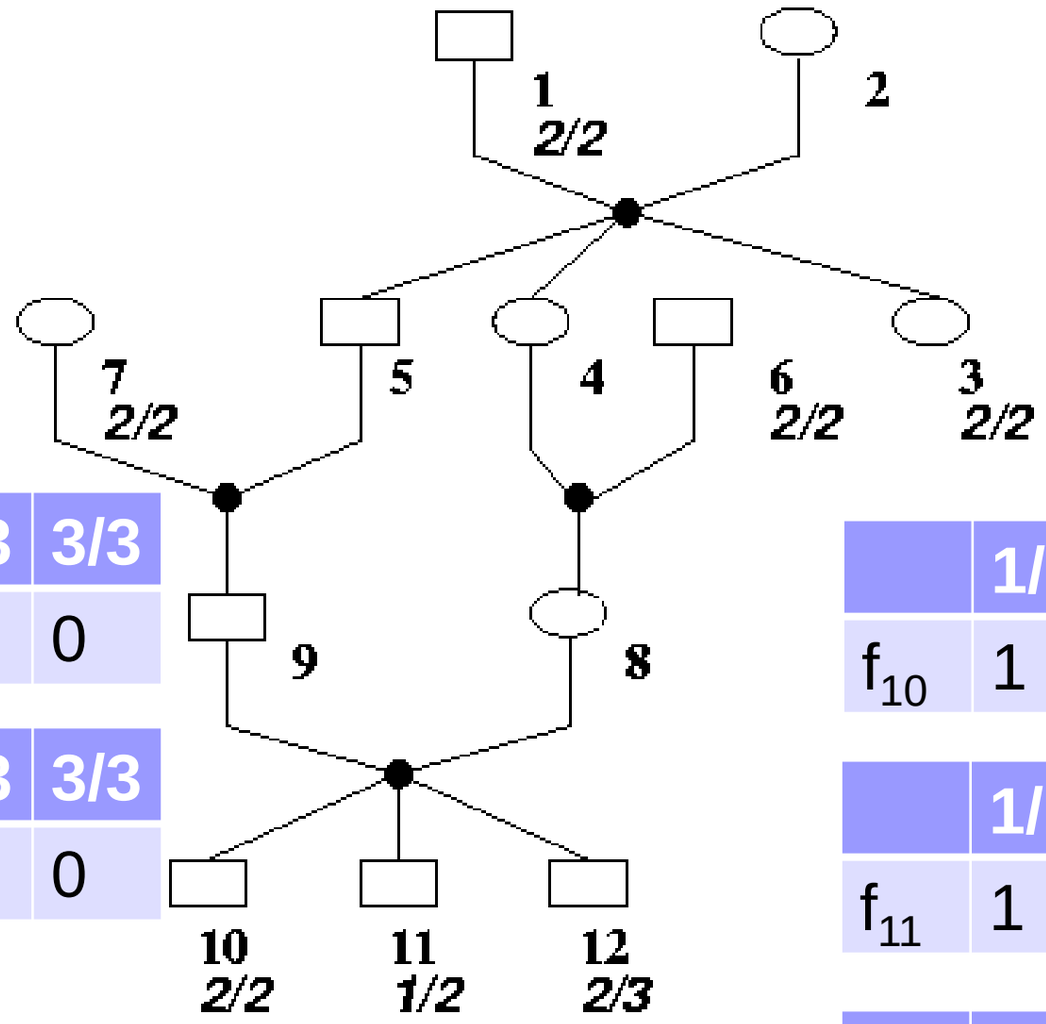
- **X**: one variable per individual
- **D**: domain of every variable is defined as the set of all possible genotypes

  Here: { 1/1, 1/2, 1/3, 2/2, 2/3, 3/3 }
- **F**:
  - Ternary hard constraints to encode Mendelian laws for any non founder
  - Unary soft constraints to encode genotyping data




| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_{10}$ | 1 | 1 | 1 | 0 | 1 | 1 |

# Generalized Soft Arc Consistency




| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_8$ | 0 | 0 | 0 | 0 | 0 | 0 |

| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_9$ | 0 | 0 | 0 | 0 | 0 | 0 |

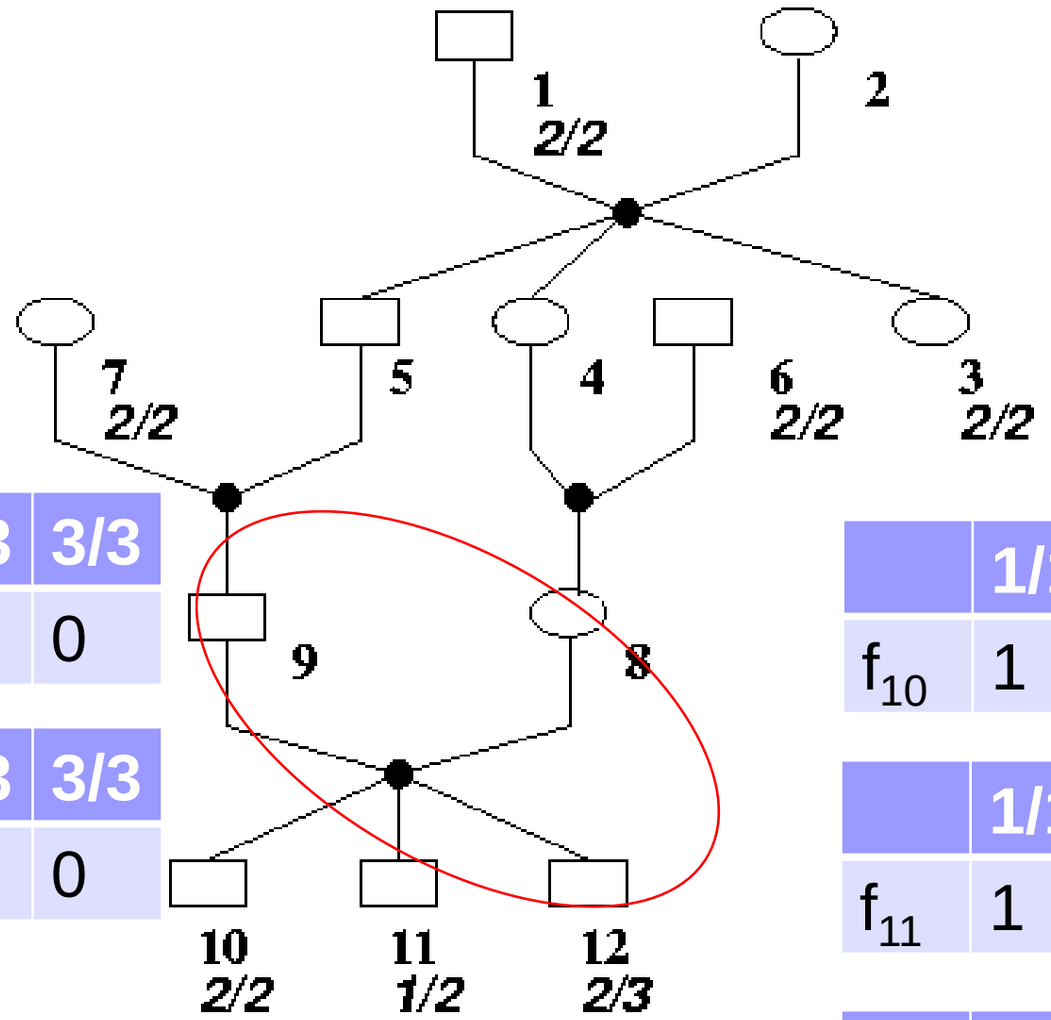
| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_{10}$ | 1 | 1 | 1 | 0 | 1 | 1 |

| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_{11}$ | 1 | 0 | 1 | 1 | 1 | 1 |

| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_{12}$ | 1 | 1 | 1 | 1 | 0 | 1 |

(Schiex, CP 2000), (Givry et al, Constraints 2008), (Lee & Leung, IJCAI 2009),…

# Generalized Soft Arc Consistency




| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_8$ | **1** | 0 | 0 | 0 | 0 | 0 |

| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_9$ | 0 | 0 | 0 | 0 | 0 | 0 |

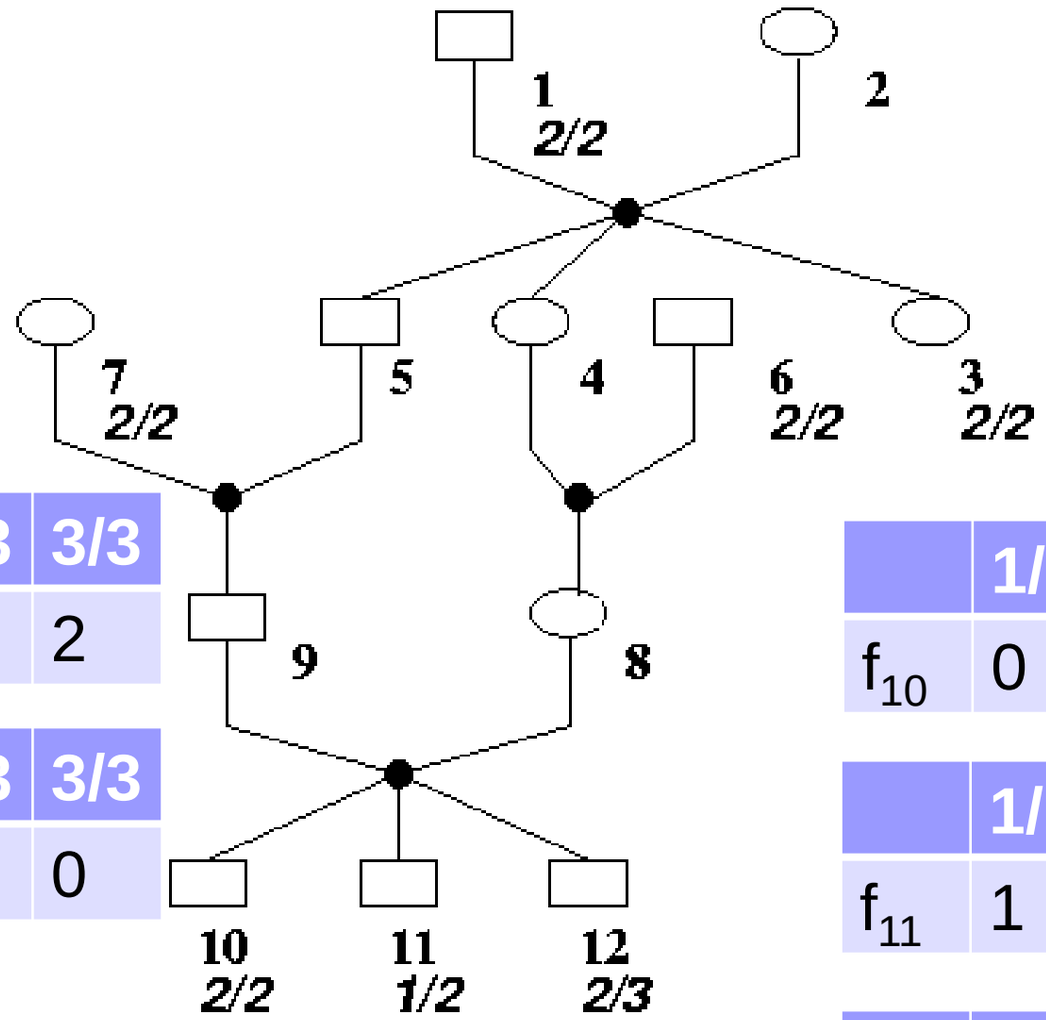
| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_{10}$ | 1 | 1 | 1 | 0 | 1 | 1 |

| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_{11}$ | 1 | 0 | 1 | 1 | 1 | 1 |

| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_{12}$ | **0** | **0** | **0** | 1 | 0 | 1 |

(Schiex, CP 2000), (Givry et al, Constraints 2008), (Lee & Leung, IJCAI 2009),…

# Generalized Soft Arc Consistency




| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_8$ | 2 | 0 | 1 | 0 | 0 | 2 |

| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_9$ | 0 | 0 | 0 | 0 | 0 | 0 |

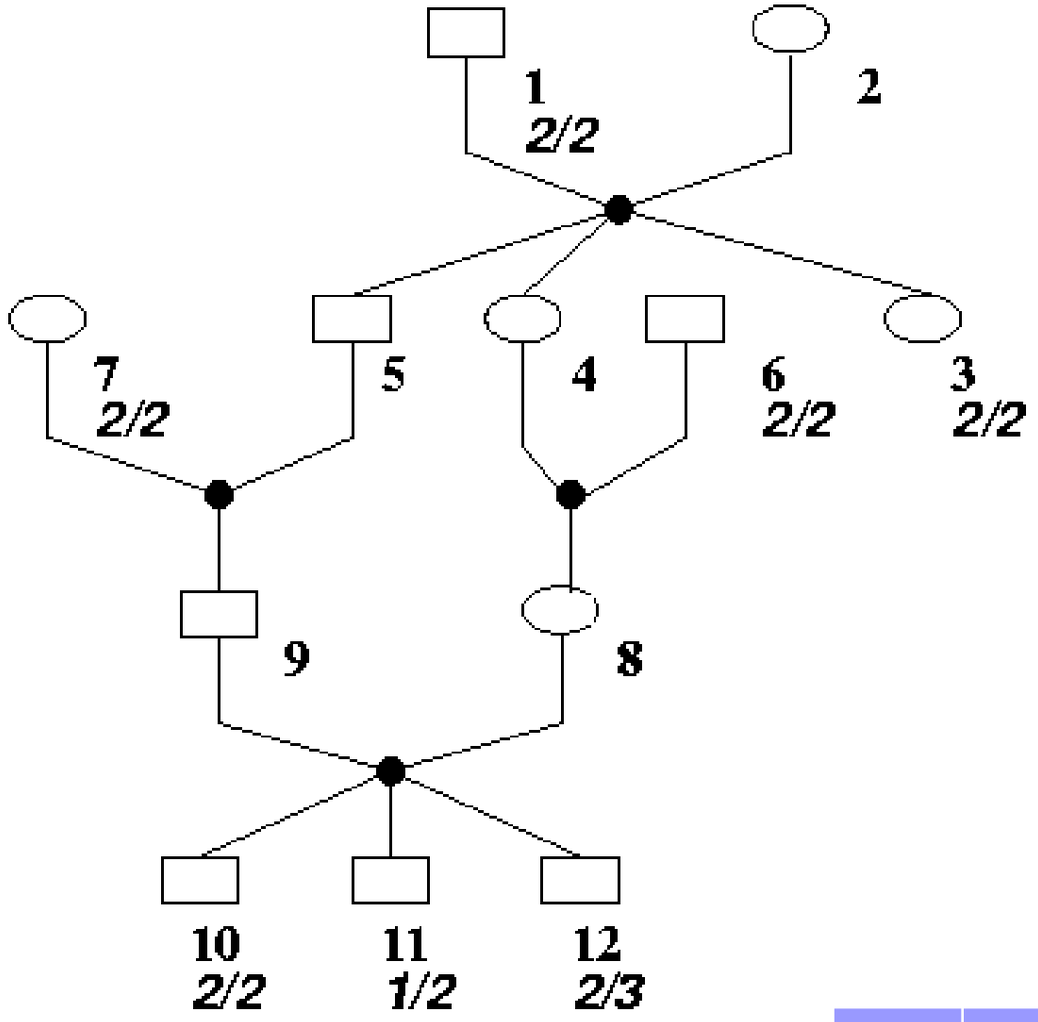
| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_{10}$ | 0 | 0 | 0 | 0 | 0 | 0 |

| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_{11}$ | 1 | 0 | 0 | 1 | 0 | 0 |

| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_{12}$ | 0 | 0 | 0 | 1 | 0 | 1 |

(Schiex, CP 2000), (Givry et al, Constraints 2008), (Lee & Leung, IJCAI 2009),…

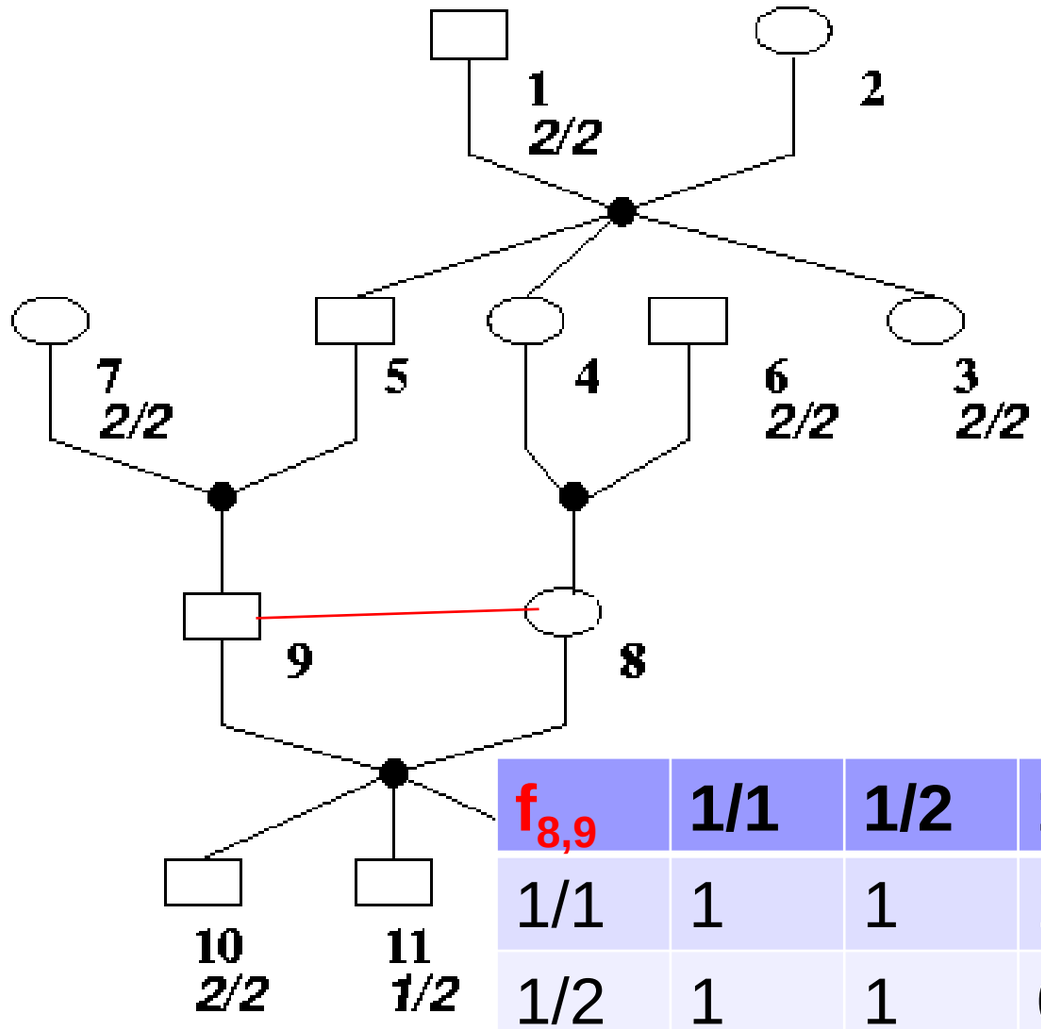
# Variable elimination




| | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| $f_{12}$ | 1 | 1 | 1 | 1 | 0 | 1 |

(Dechter, AIJ 1999)

# Variable elimination




| $f_{8,9}$ | 1/1 | 1/2 | 1/3 | 2/2 | 2/3 | 3/3 |
|---|---|---|---|---|---|---|
| 1/1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1/2 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1/3 | 1 | 0 | 1 | 0 | 0 | 1 |
| 2/2 | 1 | 1 | 0 | 1 | 0 | 0 |
| 2/3 | 1 | 0 | 0 | 0 | 0 | 0 |
| 3/3 | 1 | 0 | 1 | 0 | 0 | 1 |

(Dechter, AIJ 1999)

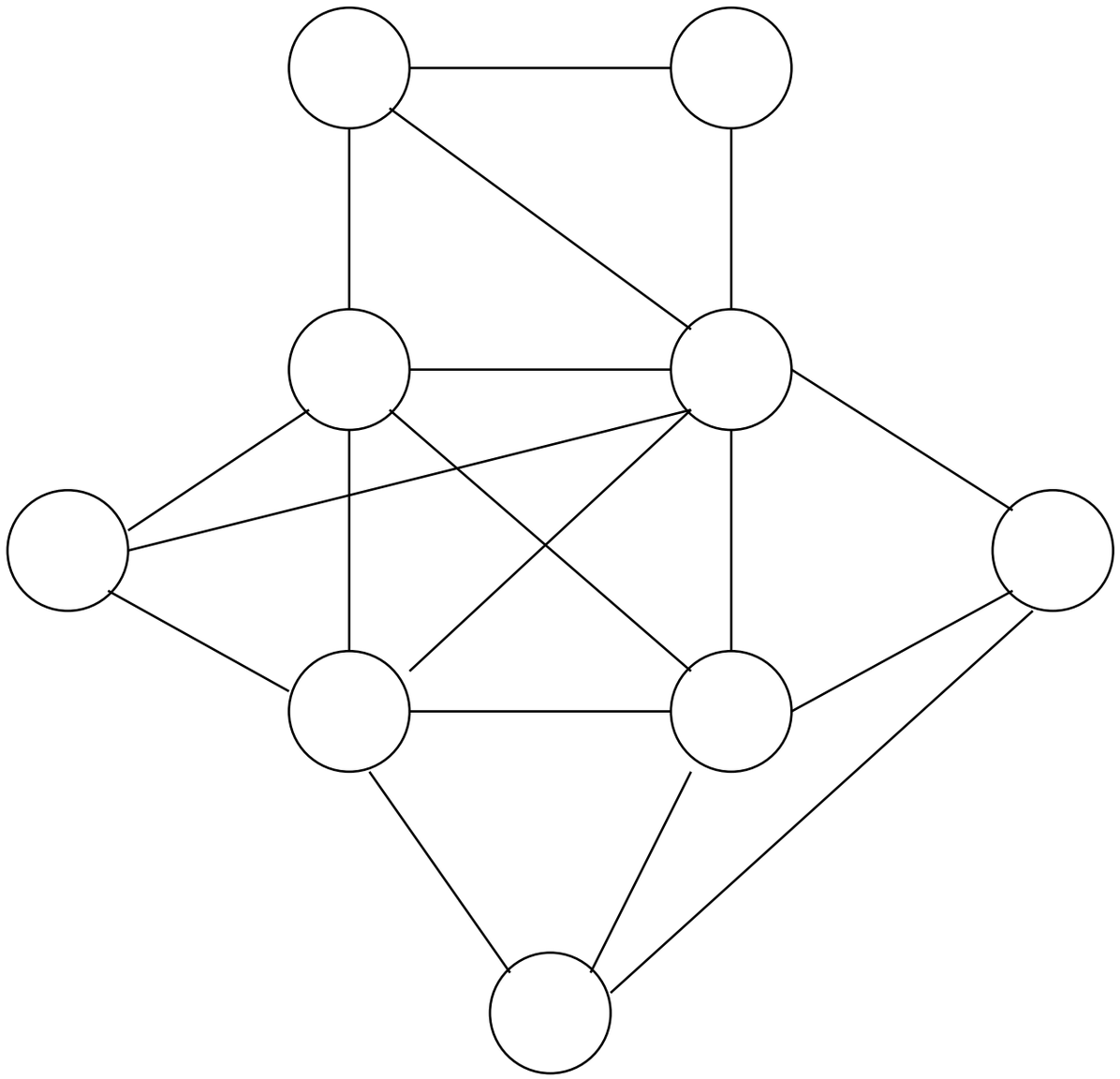
# Hybrid algorithm: Search & Variable Elimination

- Condition, condition, condition … and then only eliminate (*Cycle-Cutset*)

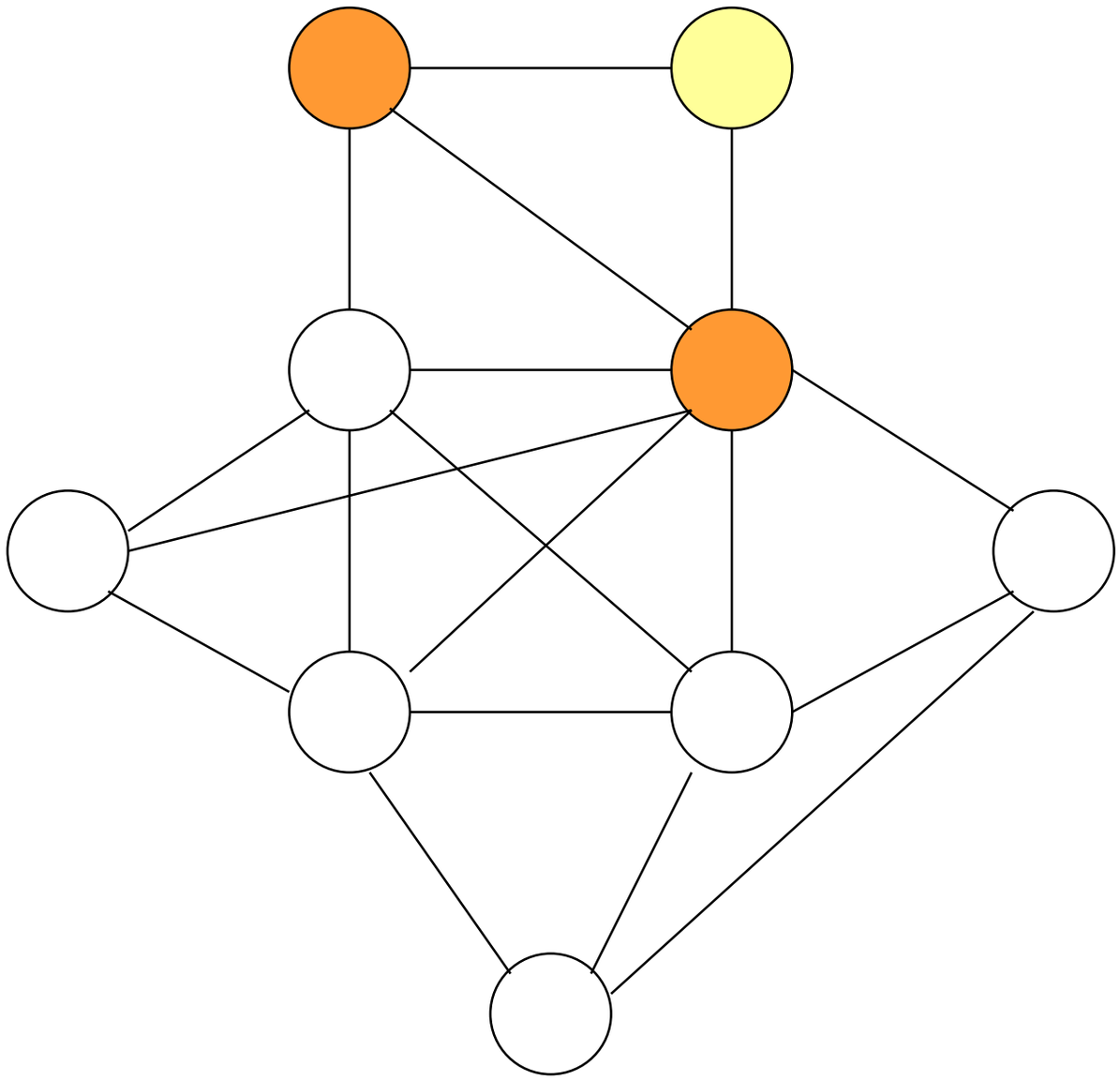
- Eliminate, eliminate, eliminate … and then only search

- Interleave conditioning and elimination

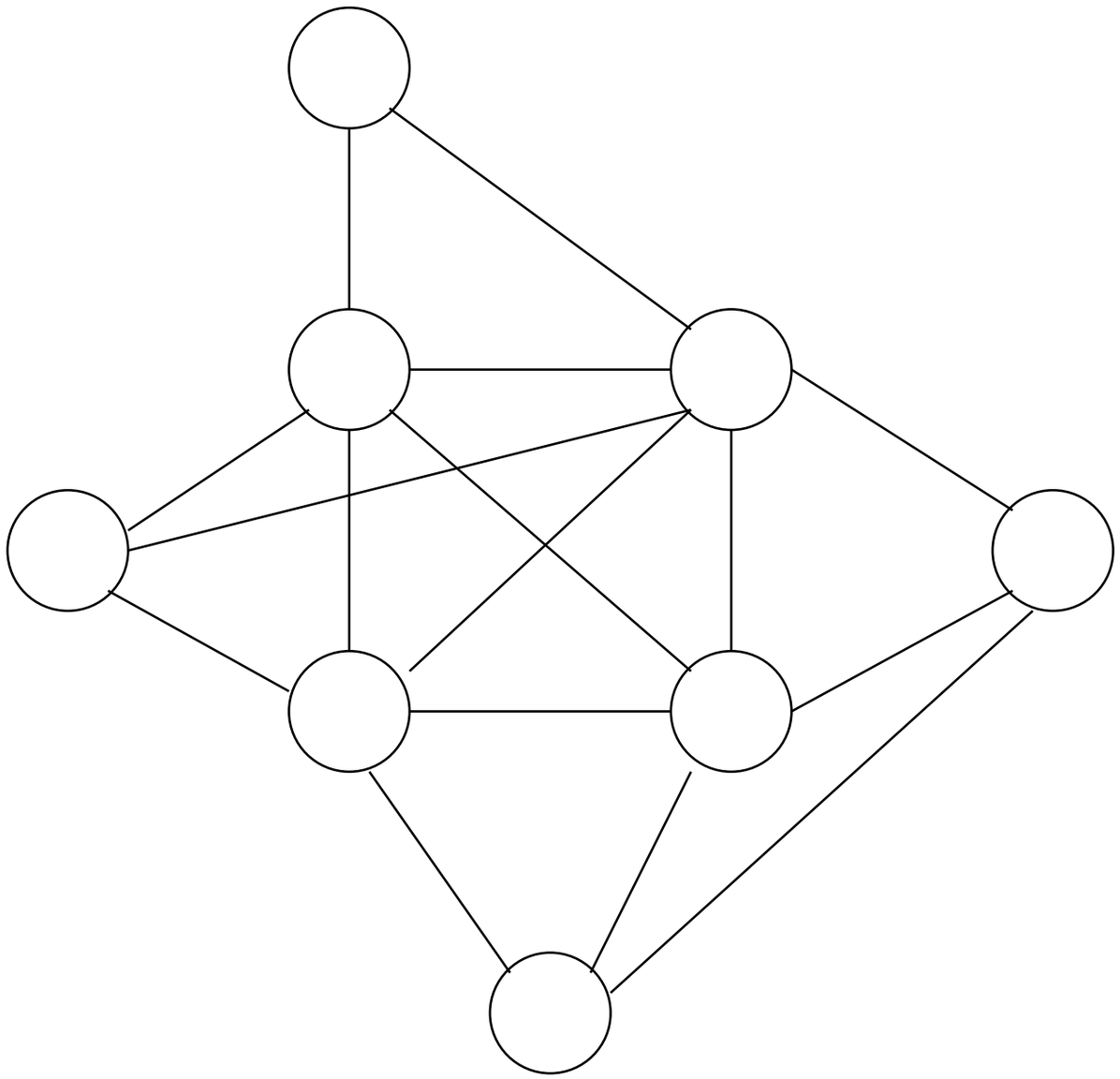
# Interleaving Conditioning and Elimination
# BB-VE(2)  (Larrosa & Dechter, CP 2000)



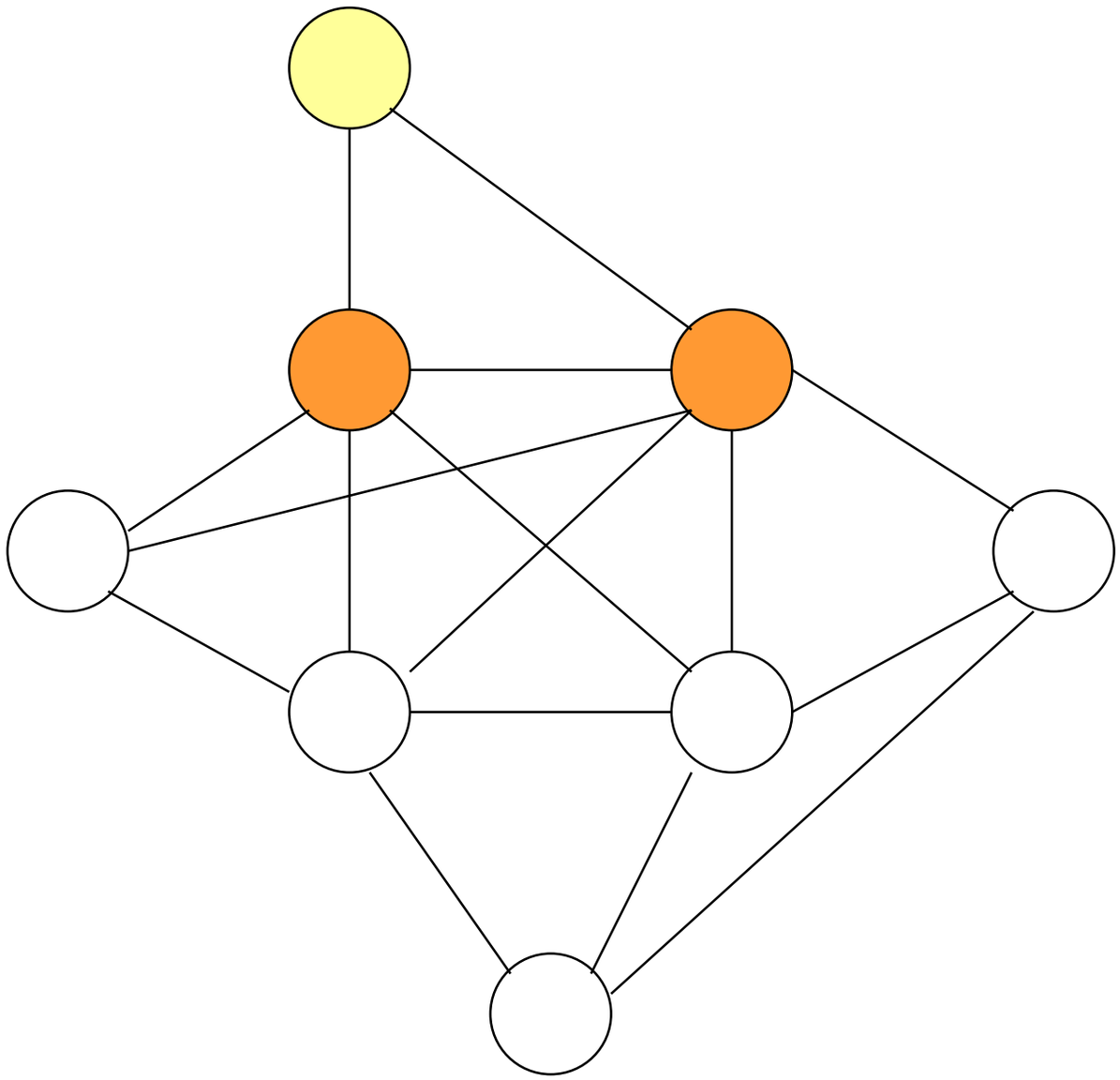
# Interleaving Conditioning and Elimination
# BB-VE(2)



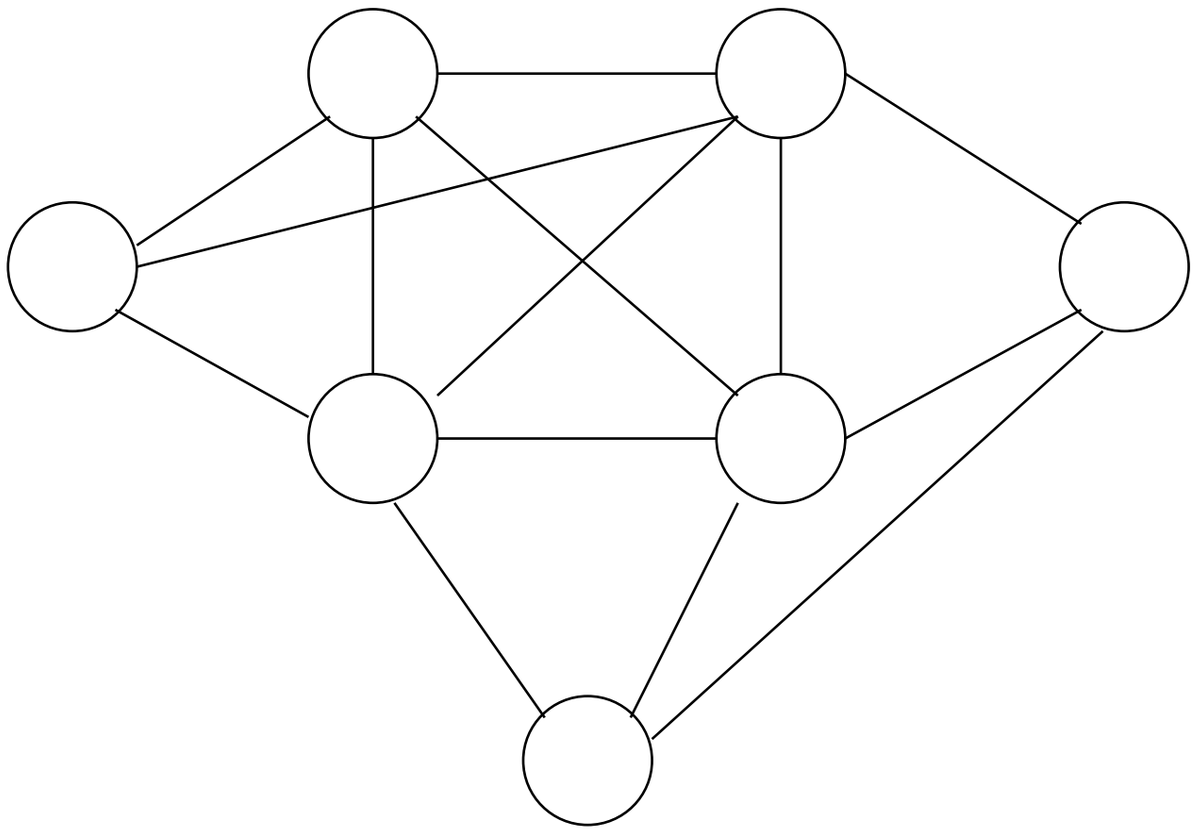
# Interleaving Conditioning and Elimination
## BB-VE(2)



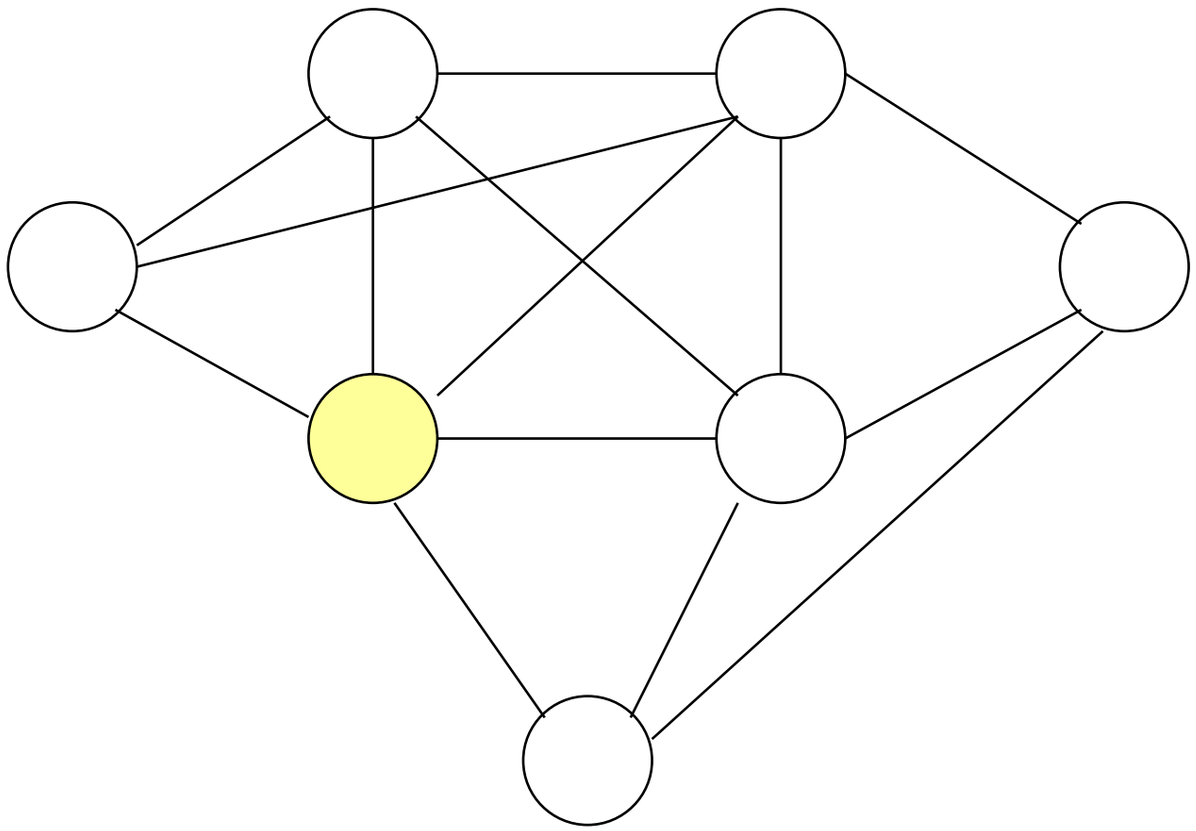
# Interleaving Conditioning and Elimination
## BB-VE(2)



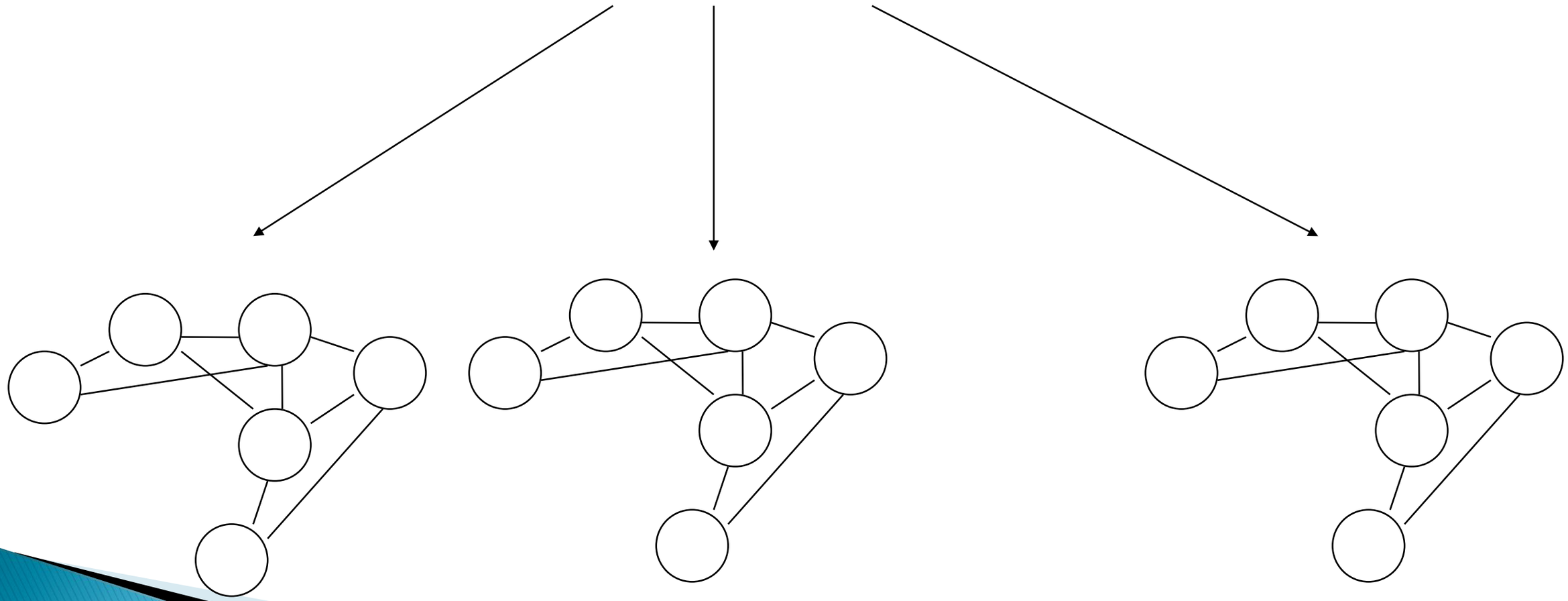
# Interleaving Conditioning and Elimination BB-VE(2)



# Interleaving Conditioning and Elimination
## BB-VE(2)

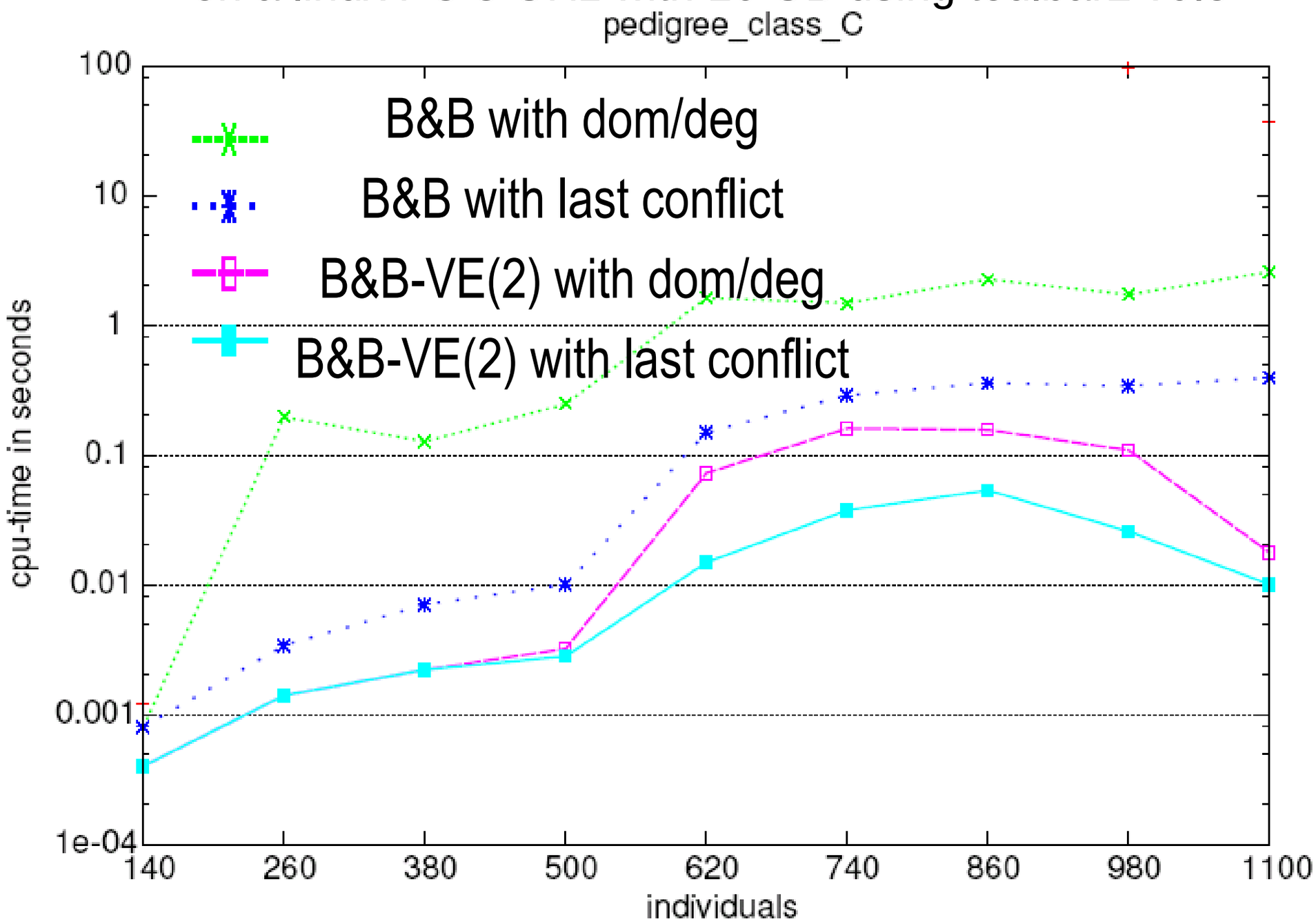


# Interleaving Conditioning and Elimination
## BB-VE(2)



# Simulated data

CPU time in seconds to find and prove optimality
on a linux PC 3 GHz with 16 GB using toulbar2 v0.5




(Givry et al, Constraints 2008)

# Real data

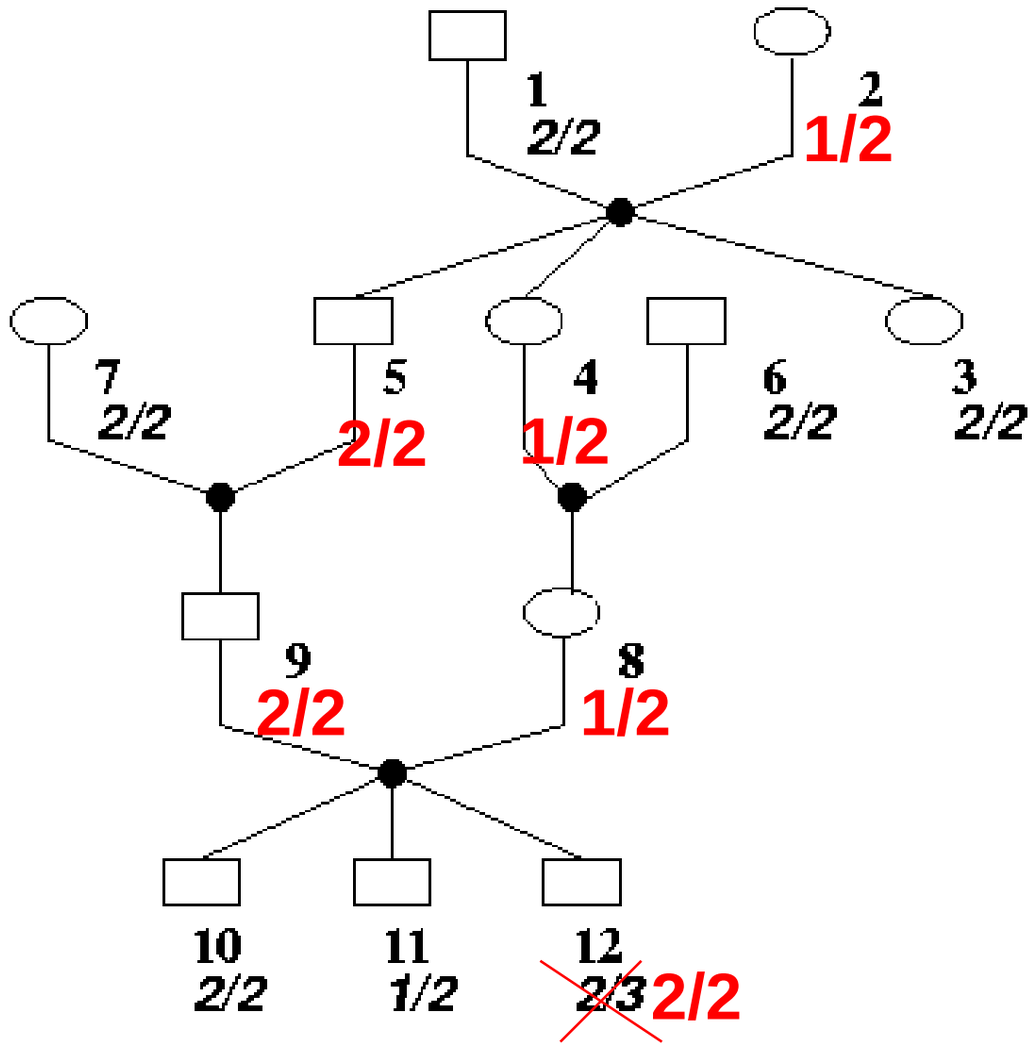
CPU time in seconds to find and prove optimality on a linux PC 3 GHz with 16 GB using toulbar2 v0.5

| | ind | *vars* | *genotyped* | alleles | *nf* | *ngen* | treewidth ub | B&B-VE(2) 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 |

# Task 3: Error Correction using Probabilistic Model

- Finds a complete assignment with maximum posterior probability

→Bayesian network

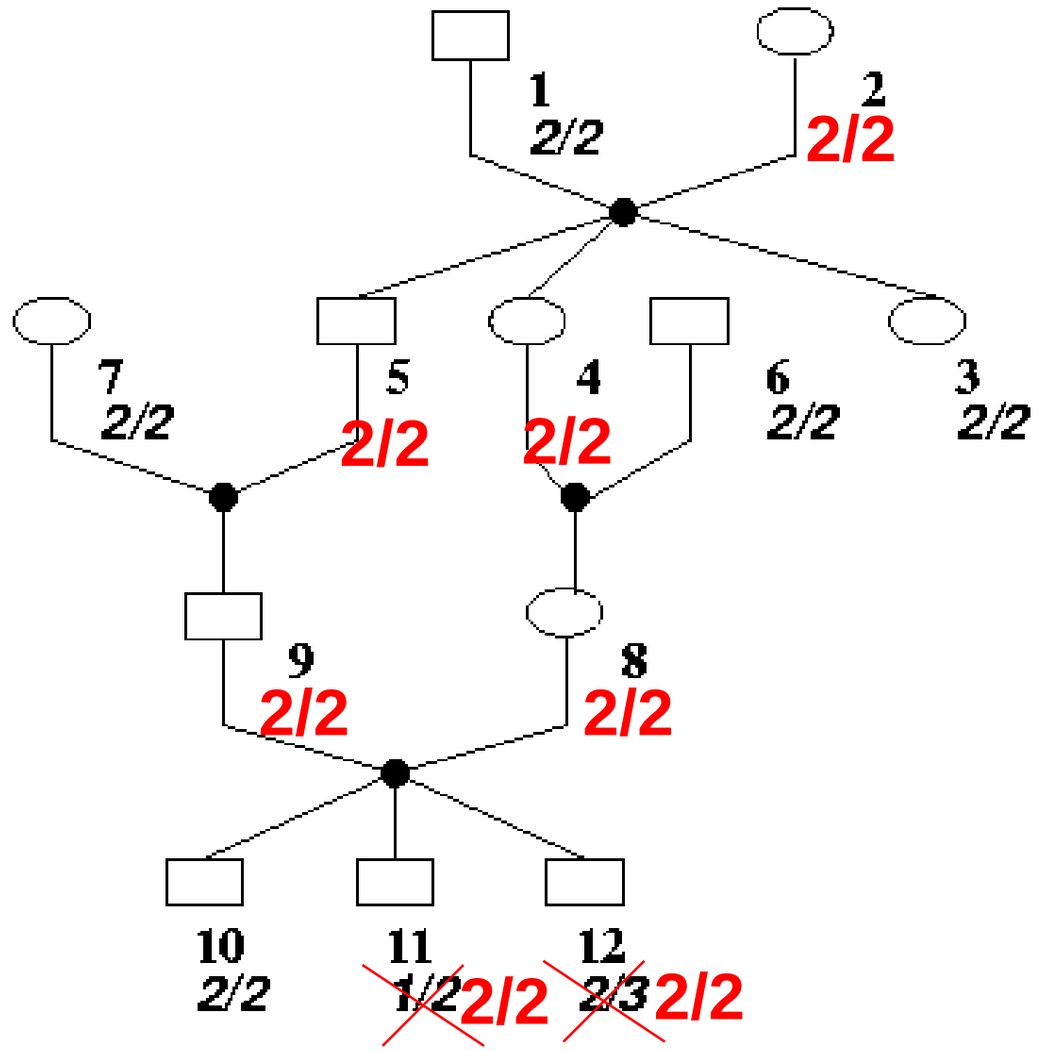



Prior on genotyping error: 1%
(and equifrequent alleles)

# Task 3: Error Correction using Probabilistic Model

- Finds a complete assignment with maximum posterior probability

→Bayesian network

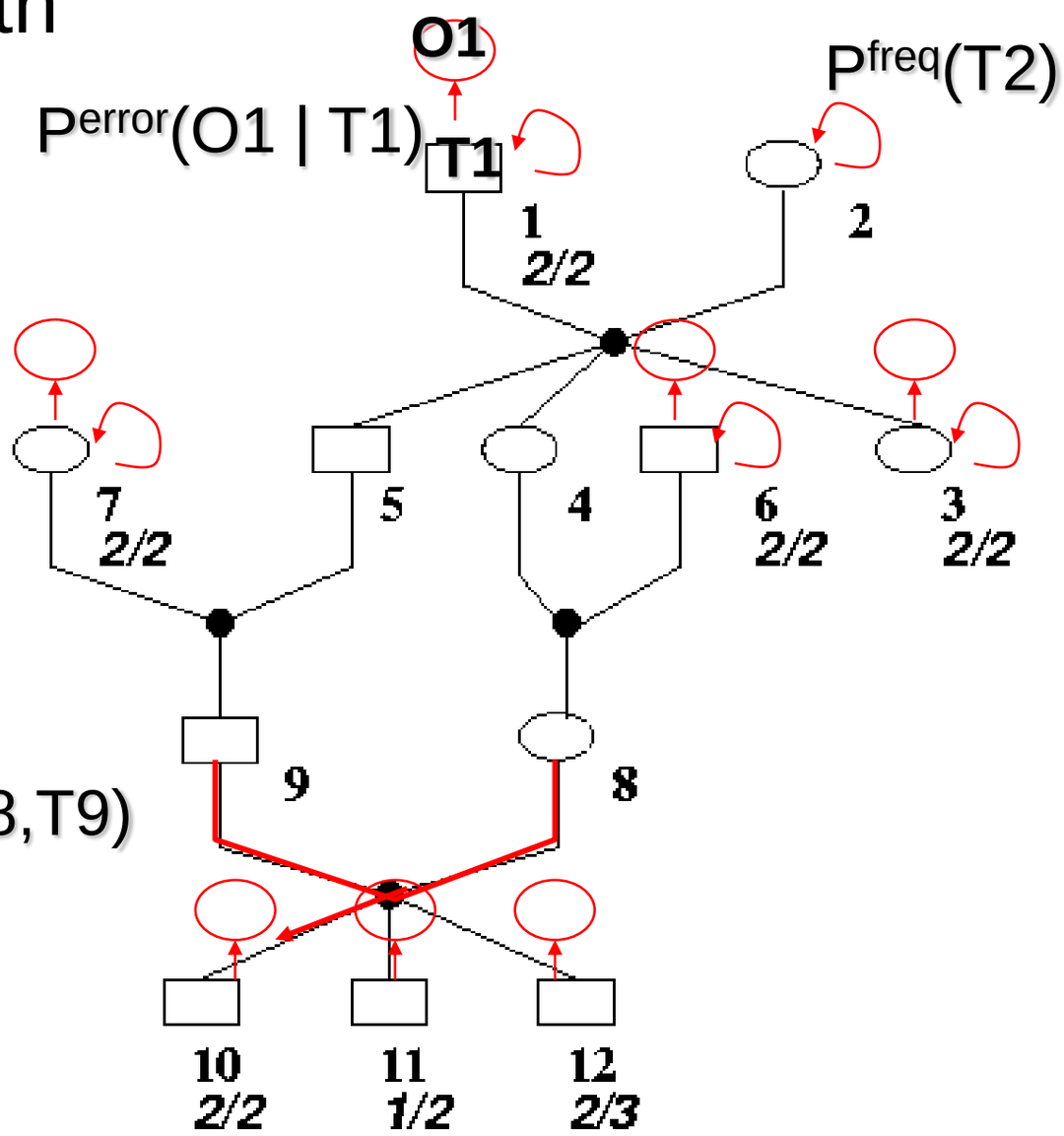



Prior on genotyping error: 10%
(and equifrequent alleles)

# Task 3: Error Correction using Probabilistic Model

- Finds a complete assignment with maximum posterior probability

→Bayesian network




$$P(O, T) = \prod P^{error}(O_i \mid T_i) \times \prod P^{mendel}(T_i \mid parents(i)) \times \prod P^{freq}(T_i)$$

# Graphical model (X,D,F)

- *X,* a set of ***n*** variables
- *D,* finite domains of maximum size ***d***
- $F=\{f_{S1},\ldots,f_{Se}\}$, *a set of* ***e*** functions with $S_i \subseteq X$

- Probabilistic models (Markov Net, Bayes Net,..)

$$P(X) \propto \prod_{i=1}^{e} f_i(S_i)$$

$$\mathrm{T} = 0$$

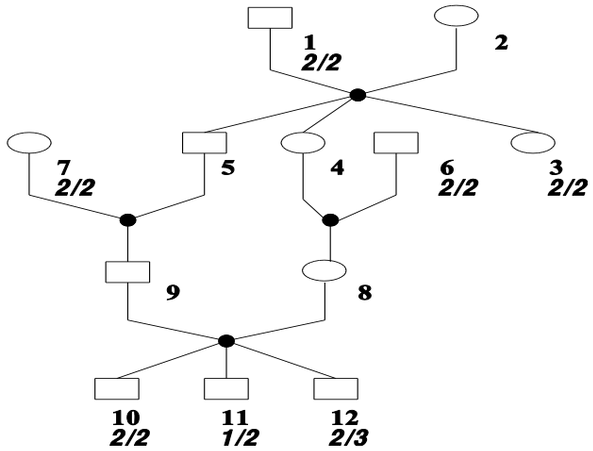
- Deterministic models (Max-SAT, Weighted CSP,…)

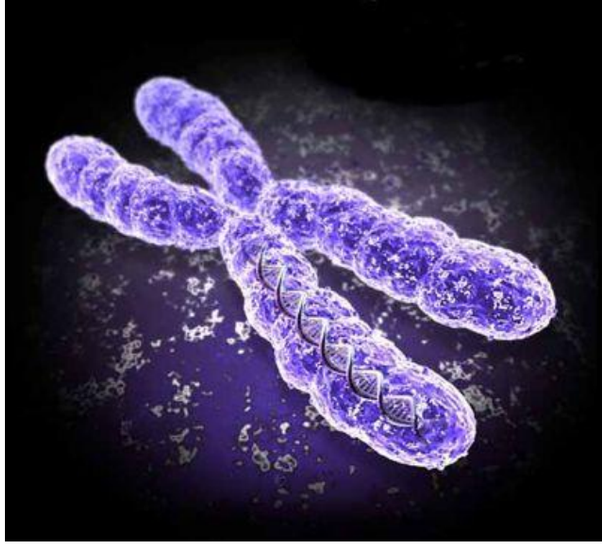
$$score(X) = \sum_{i=1}^{e} f_i(S_i)$$

$$\mathrm{T} = k$$

(finite or infinite positive integer)

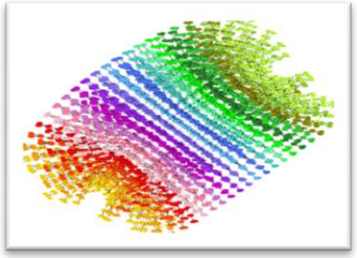
# Genetics



- Restoring consistency on large animal pedigrees
  - M. Sánchez, S. de Givry, and T. Schiex. Mendelian error detection in complex pedigrees using weighted constraint satisfaction techniques. Constraints, 13(1):130-154, 2008
  - MendelSoft http://www7.inra.fr/mia/T/MendelSoft/

- Optimal haplotype reconstruction in half-sib families
  - A. Favier, J-M. Elsen, S. de Givry, and A. Legarra. Optimal haplotype reconstruction in half-sib families. In ICLP-10 workshop on Constraint Based Methods for Bioinformatics, Edinburgh, UK, 2010



- Optimizing the reference population in a genomic selection design
  - J-M. Elsen, S. de Givry, G. Katsirelos, F. Shumbusho. Optimizing the reference population in a genomic selection design. In WCB'13, Uppsala, Sweden, 2013

# Combining constraint processing and pattern matching to describe and locate structured motifs in genomic sequences

P. Thébault, C. Gaspin, S. de Givry and T. Schiex

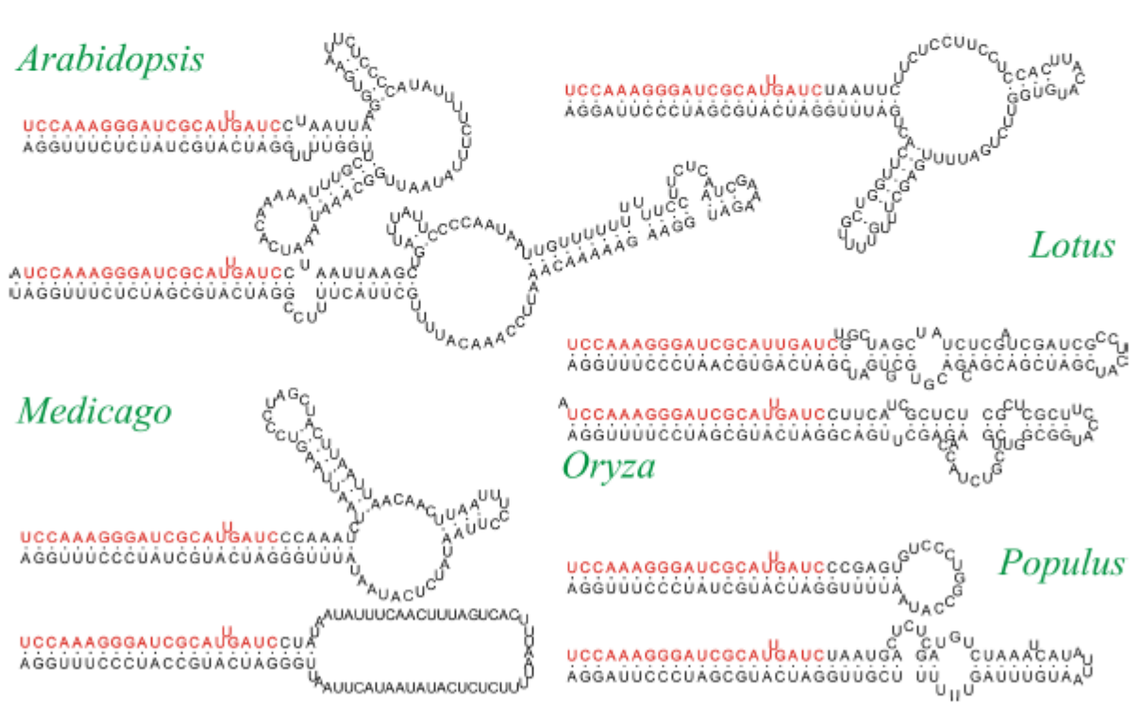
Thanks to Matthias Zytnicki

INRA BIA Toulouse

(IJCAI workshop 2005), (Bioinformatics 2006), (Constraints 2008)

# Motivation

- recent discovery of many important RNA gene families (snoRNA, siRNA, miRNA...)
- in interaction with other molecules



# RNA structure

One may define an RNA gene through:

- its primary structure,

Oriented sequence:

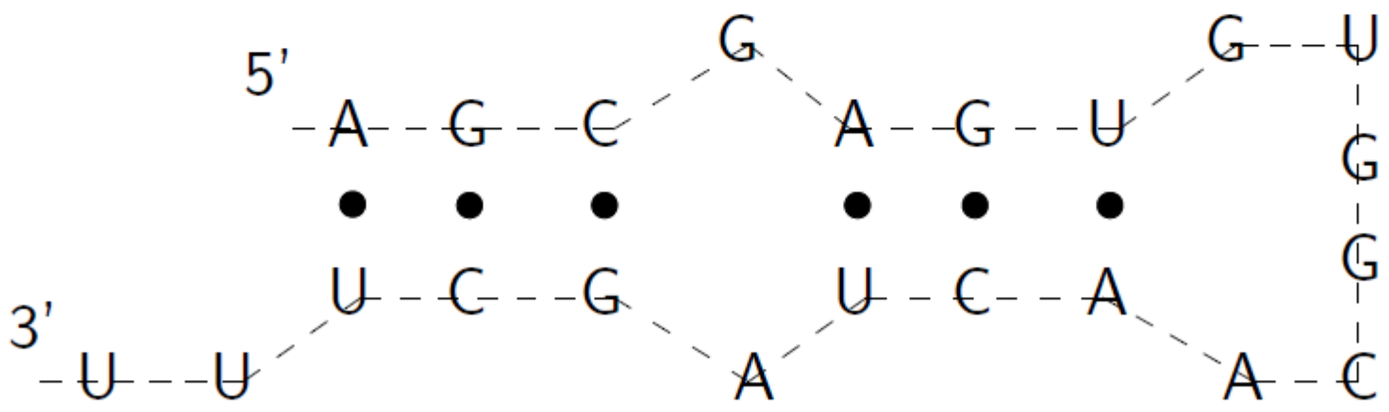
$^{5'}$ A G C G A G U G U G G C A A $^{3'}$

# RNA structure

One may define an RNA gene through:

- its primary structure,
- its secondary structure,

Watson-Crick G-C/A-U (or Wobble G-U) interactions

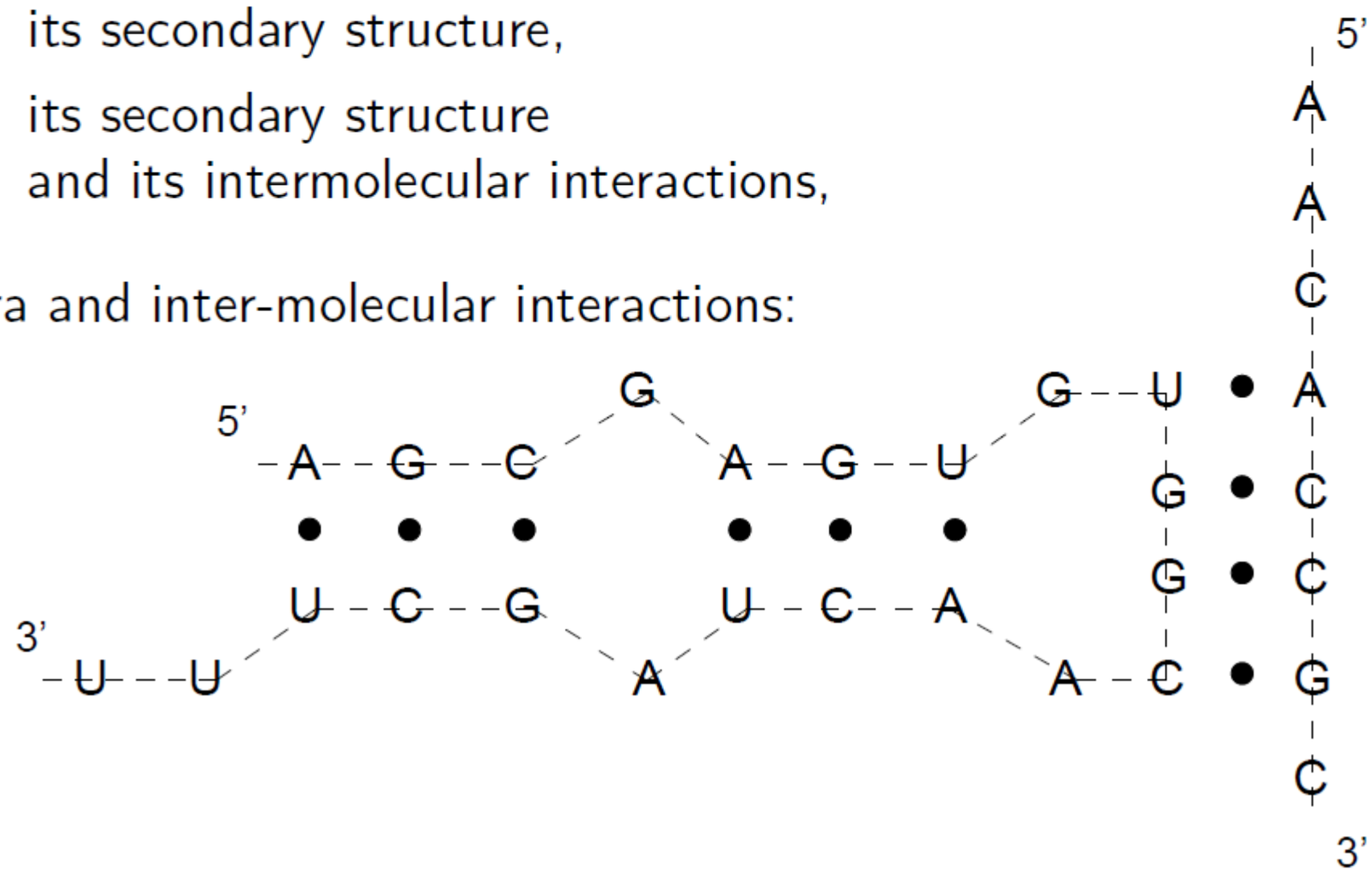


# RNA structure

One may define an RNA gene through:

- its primary structure,
- its secondary structure,
- its secondary structure
  and its intermolecular interactions,

Intra and inter-molecular interactions:

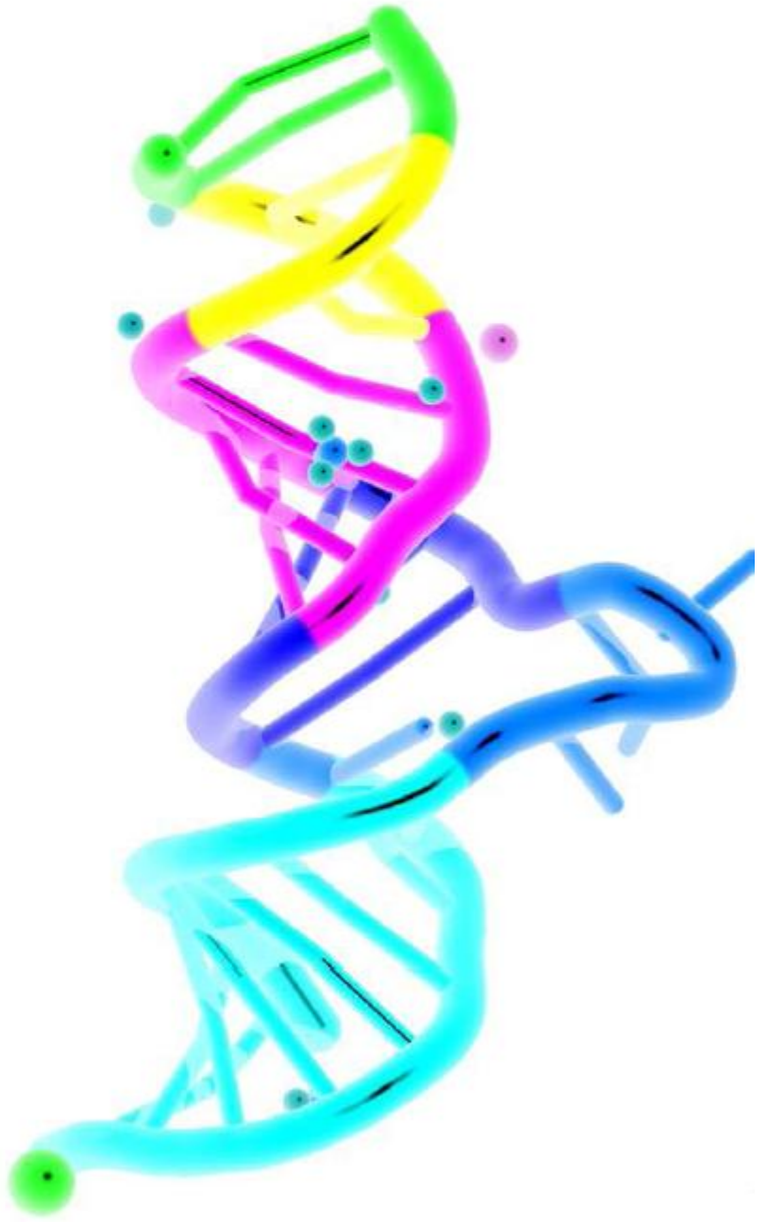


# RNA structure

One may define an RNA gene through:

- its primary structure,
- its secondary structure,
- its secondary structure and its intermolecular interactions,
- its tertiary structure.

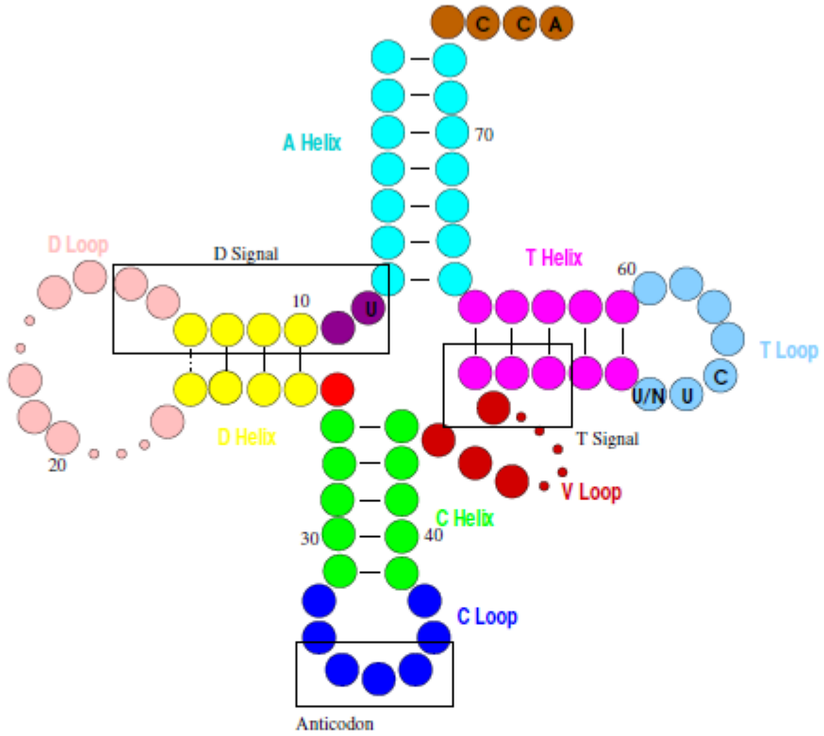
This determines the biological activity.



# The precise problem

- We define an RNA gene family as a set of RNA genes sharing a common function.
- Our aim is to identify all the elements of a given RNA gene family.
- For this, we could use a descriptor that relies on:
  - the tertiary structure? partly unknown, difficult to model,
  - the primary structure? poorly conserved, not very discriminating,
  - the secondary structure and the intermolecular interactions.

# A famous RNA gene family: tRNA



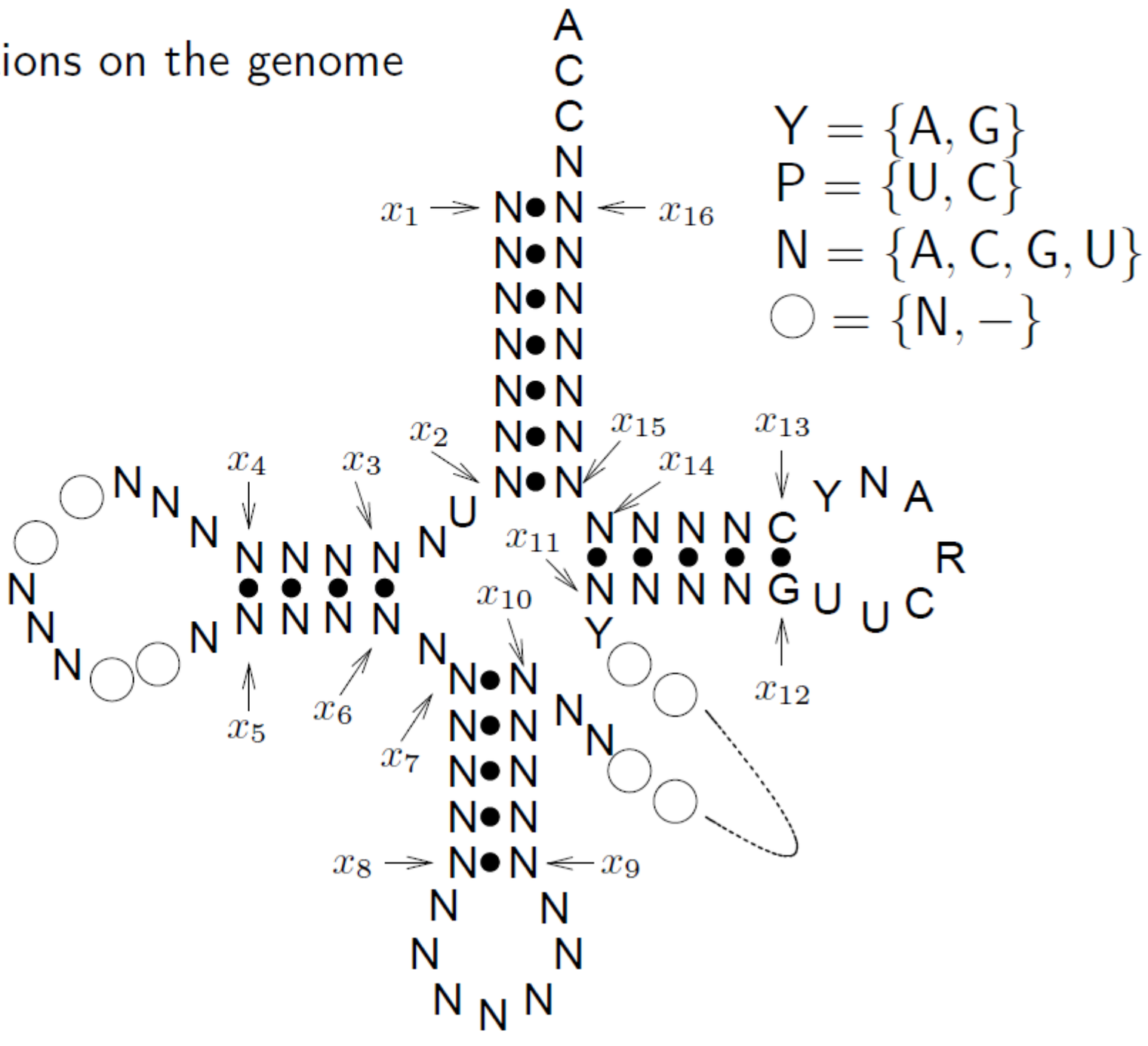

```
S1   GCGGGGG UG CCCG AGCCU.GGCCAA AGGG G UCGGG CUCAGGA CCCGA UGGCGUAGGCCUGC GUGGG UUCAAAU CCCAC CCCCCGCA
S2   GCCGCCG UA GCUC AGCC.CGGG..A GAGC G CCCGG CUGAAGA CCGGG UU.........GUC CGGGG UUCAAGU CCCCG CGGCGGCA
S3   GGGCCCG UA GCUU AGCUC.GGU..A GAGC G CUCGG CUCAUAA CCGAG UG.........GUC AGGGG UUCAAAU CCCCU CGGGCCCA
S4   GGGCCCG UC GUCU AGCC.UGGUU.A GGAC G CUGCC CUGACGC GGCAG AA.........AUC CUGGG UUCAAGU CCCAG CGGGCCCA
S5   GCGGCCG UC GUCU AGUCU.GGAUUA GGAC G CUGGC CUUCCAA GCCAG UA.........AUC CCGGG UUCAAAU CCCGG CGGCCGCA
S6   GCCGGGG UC GCCU AGCC.UGGUCAA GGGC G CCGGA CUCAUAA UCCGG UC.........UUC CCGGG UUCGAAU CCCGG CCCCGGCA
S7   GGGCCGG UA GUCU AGC...GGA..A GGAU G CCCGC CUCGCGC GCGGG AG.........AUC CCGGG UUCGAAU CCCGG CCGGUCCA
S8   GCGGGGG UG CCCG AGCCA.GGUCAA AGGG G CAGGG UUCAGGU CCCUG UGGCGUAGGCCUGC GUGGG UUCAAAU CCCAC CCCCCGCA
S9   GCGGGGG UG CCCG AGCCA.GGUCAA AGGG G CAGGG CUCAAGA CCCUG UGGCGUAGGCCUGC GUGGG UUCGAAU CCCAC CCCCCGCA
S10  GGGCUCG UA GCUC AGC...GGG..A GAGC G CCGCC UUUGCGA GGCGG AG.........GCC GCGGG UUCAAAU CCCGC CGAGUCCA
```

Helix 2 Helix 3 Helix 4

Helix 1

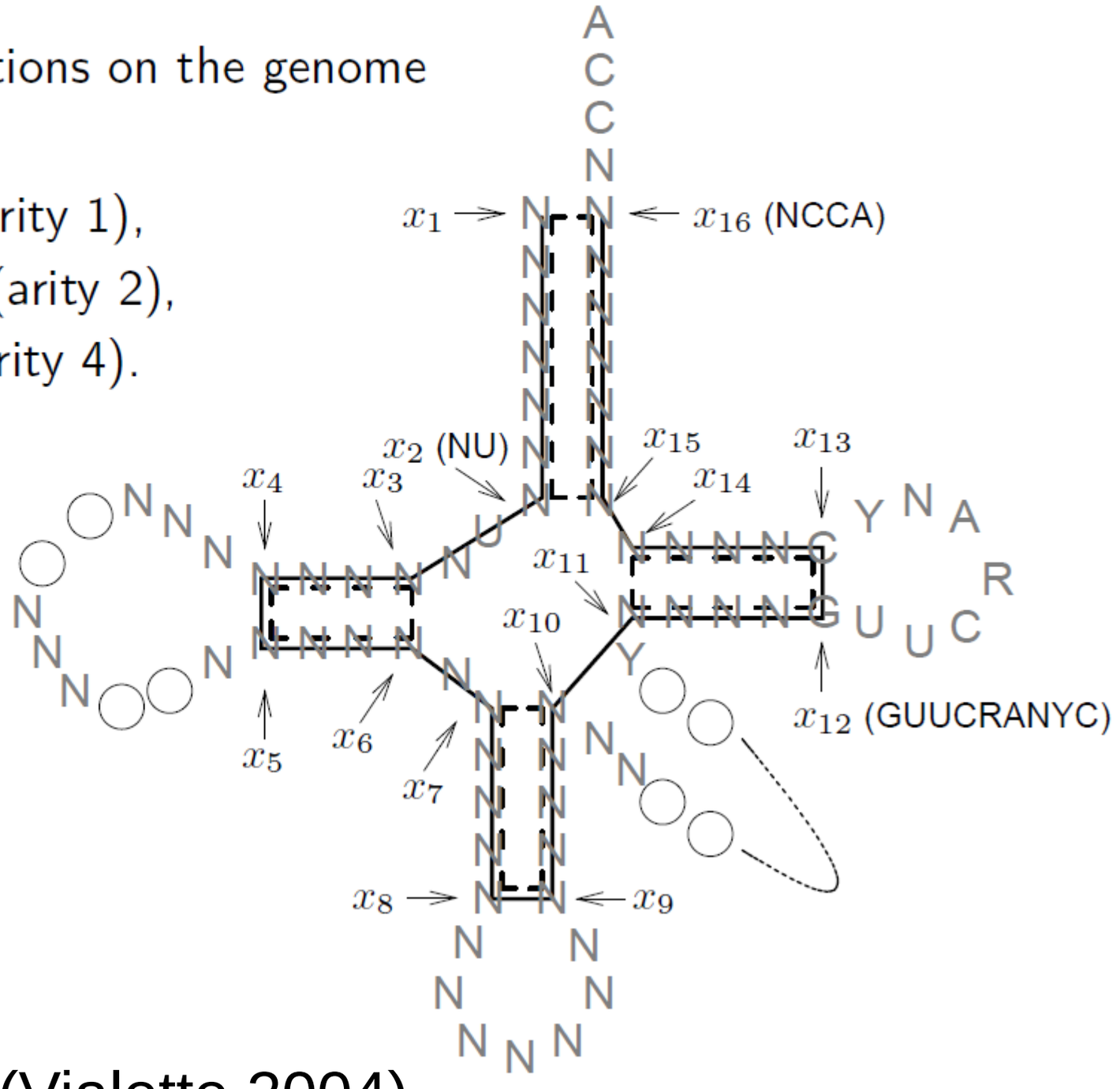
# A CSP model for RNA gene finding

- Variables: positions on the genome



# A CSP model for RNA gene finding

- Variables: positions on the genome
- Constraints:
  - the word (arity 1),
  - the spacer (arity 2),
  - the helix (arity 4).




NP-complete problem (Vialette 2004)

# Bound (Soft) Arc Consistency

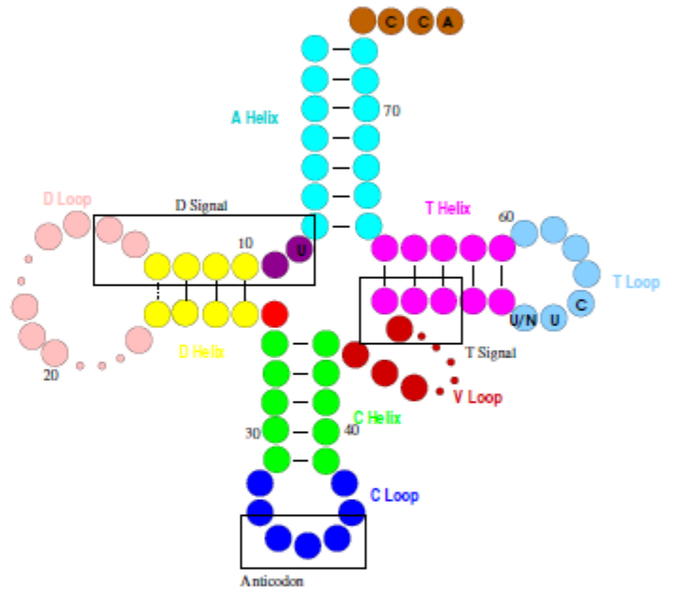
- CSP model
  - **spacer**: interval analysis (2-B consistency)
  - **word**: support search using Baeza-Yates/Manber algorithm. Allows for errors (extra sensitivity).
  - **helices (intra)**: support search by naive match search (locally bounded by spacers). Allows for errors (extra sensitivity).
  - **helices (inter)**: uses sophisticated suffix-trees ($k$-factor tree [Allali, Sagot 2004]). No error allowed.
- WCSP model
  - Minimizes sum of errors in word and helices (intra & inter)

| Size | 10k | 50k | 100k | 500k | 1M | 4.9M |
|---|---|---|---|---|---|---|
| # of solutions | 32 | 33 | 33 | 33 | 41 | 274 |
| AC* | | | | | | |
| Time | 1hour 25min. | 44 hours | - | - | - | - |
| # of backtracks | 93 | 101 | - | - | - | - |
| BAC | | | | | | |
| Time (sec.) | 0.016 | 0.036 | 0.064 | 0.25 | 0.50 | 2.58 |
| # of backtracks | 93 | 101 | 102 | 137 | 223 | 1159 |

# MilPat: efficiency

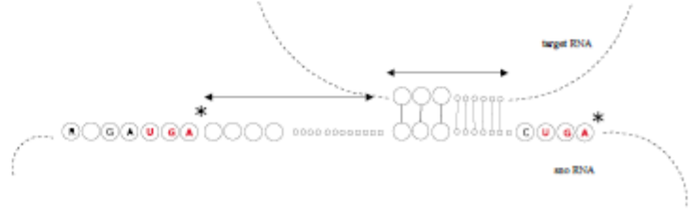
All tRNA in *S. cerevisiae* (12,07 Mb, 7,313,791 sol.)

| Tool | Time |
|---|---|
| PatScan | 1 h 40 |
| RNAMotif | 8 h 40 |
| RNAMot | 92 h |
| MilPat (order 1) | 1 h 52 |
| MilPat (order 2) | 20 mn 32 |




C/D snoRNAs in *Pyrococcus abyssi* (1,7 Mb, 59 known genes)

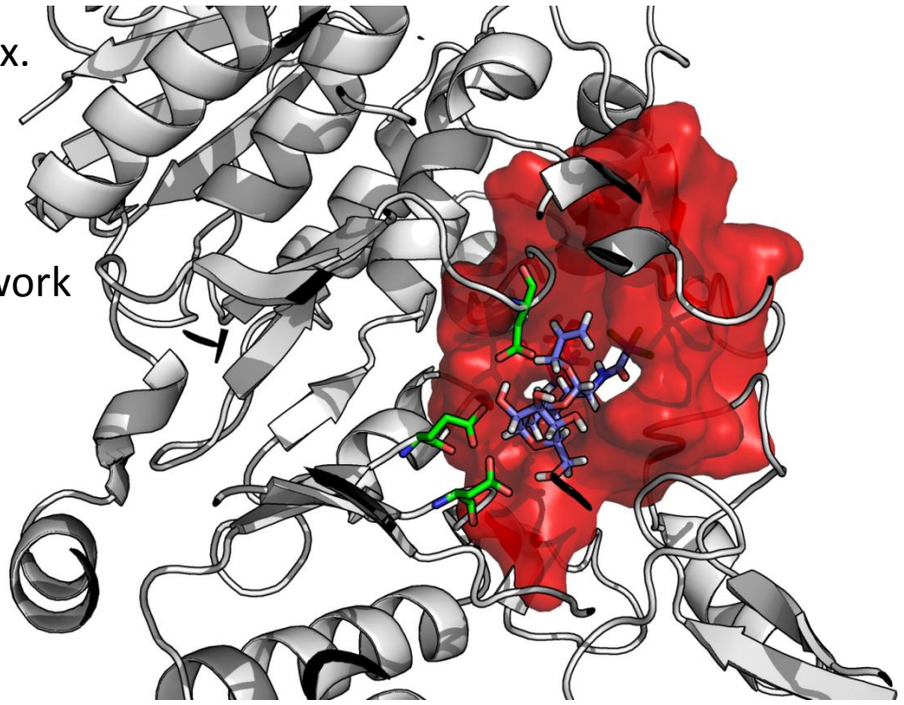
| Tool | # Sol. | True pos. | Time |
|---|---|---|---|
| SnoScan | 1611 | 27 | 20 min. |
| MilPat | 852 | 42 | 8 s. |



# RNA & protein structure

- RNA motif search in genomic sequences
  - P. Thébault, S. de Givry, T. Schiex, C. Gaspin. Combining constraint network processing and pattern matching to describe and locate structured motifs in genomic sequences. Bioinformatics, 22(17), 2006
  - M. Zytnicki, C. Gaspin, T. Schiex. DARN! A weighted constraint solver for RNA motif localization. Constraints, 13(1), 2008

- Computational Protein Design
  - D. Allouche, S. Traoré, I. André, S. de Givry, G. Katsirelos, S. Barbe, T. Schiex. Computational Protein Design as a Cost Function Network Optimization Problem. In CP'2012, Québec, Canada.
  - S. Traoré, D. Allouche, S. de Givry, G. Katsirelos, T. Schiex, S. Barbe. A New Framework for Computational Protein Design through Cost Function Network Optimization. Bioinformatics, 2013
  - toulbar2 https://mulcyber.toulouse.inra.fr/projects/toulbar2/



LISBP
Laboratoire d'Ingénierie
des Systèmes Biologiques
et des Procédés

Catalysis & Enzyme Molecular Engineering Group

# COMPUTATIONAL PROTEIN DESIGN

David Allouche, Jessica Davies, Simon de Givry, George Katsirelos, Thomas Schiex
*UBIA, UR-875, INRA, F-31320 Castanet Tolosan, France*
Isabelle André, Sophie Barbe, Seydou Traoré
*LISBP, INSA, UMR INRA 792/CNRS 5504, F-31400 Toulouse, France*
Steve Prestwich, Barry O'Sullivan
*Cork Constraint Computation Centre, University College Cork, Ireland*

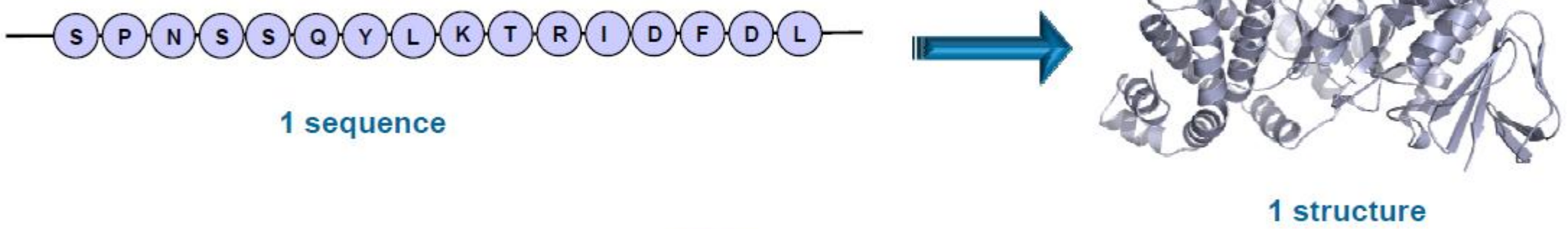
Meeting 30 Nov 2011

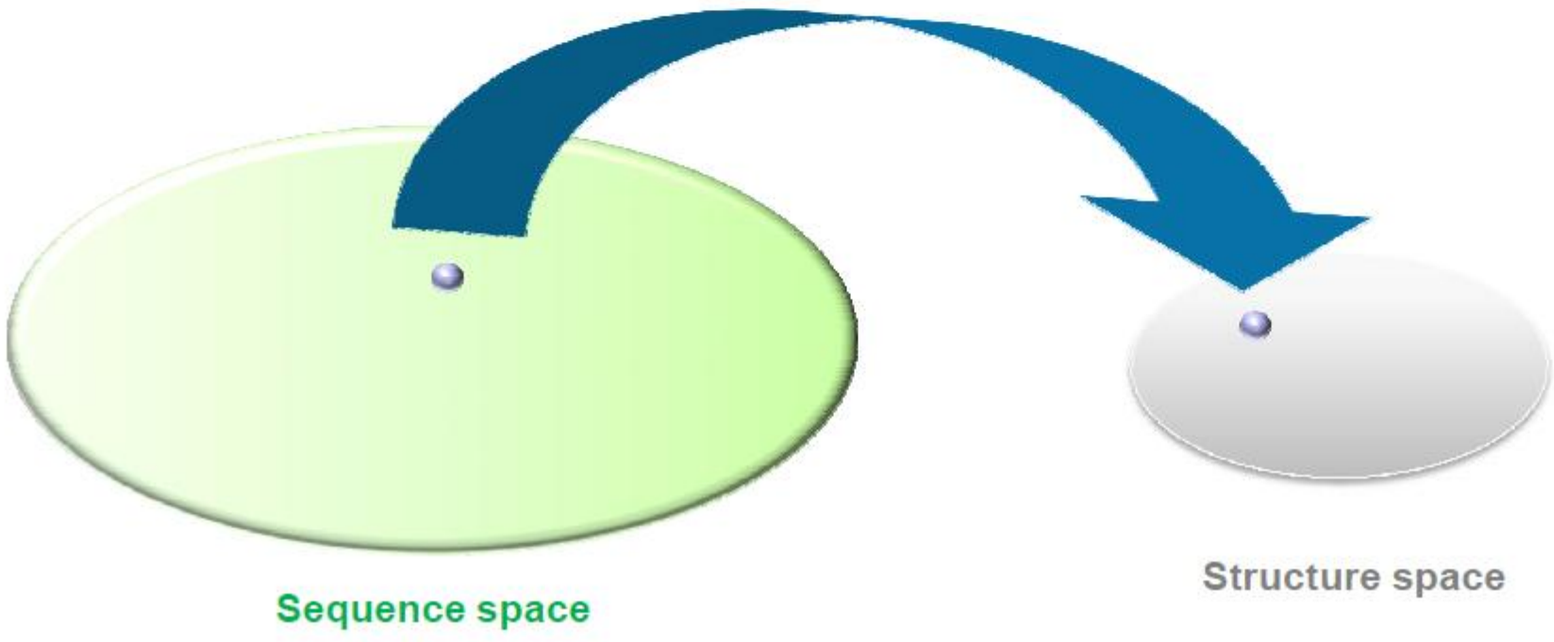
INRA cnrs INSA TOULOUSE Université de Toulouse

# PROTEIN FOLD PREDICTION VERSUS PROTEIN (RE)DESIGN

LISBP

- **PROTEIN FOLD PREDICTION**

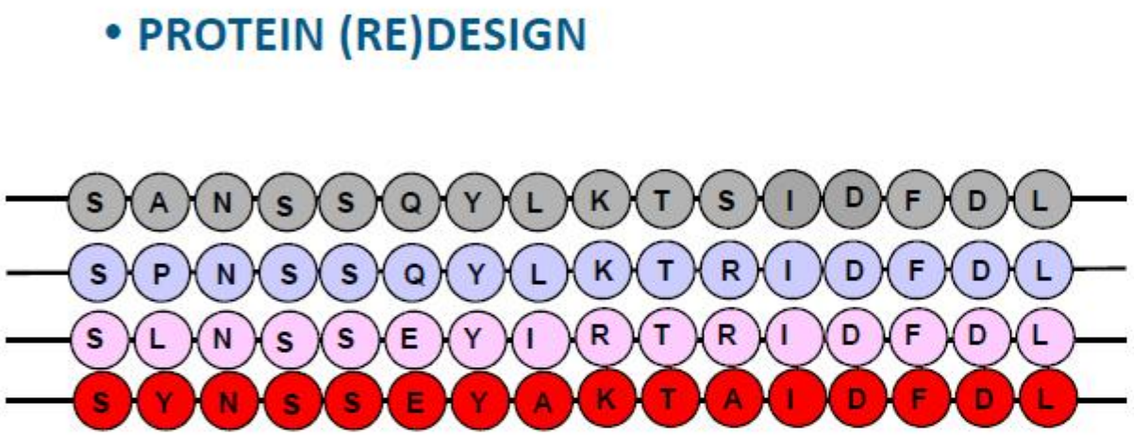


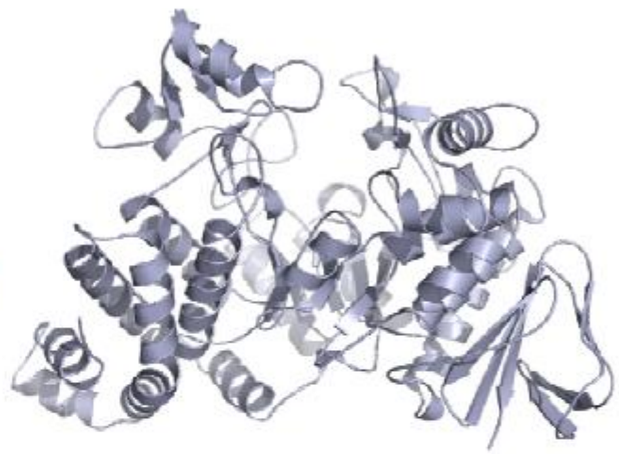
# PROTEIN FOLD PREDICTION VERSUS PROTEIN (RE)DESIGN

LISBP

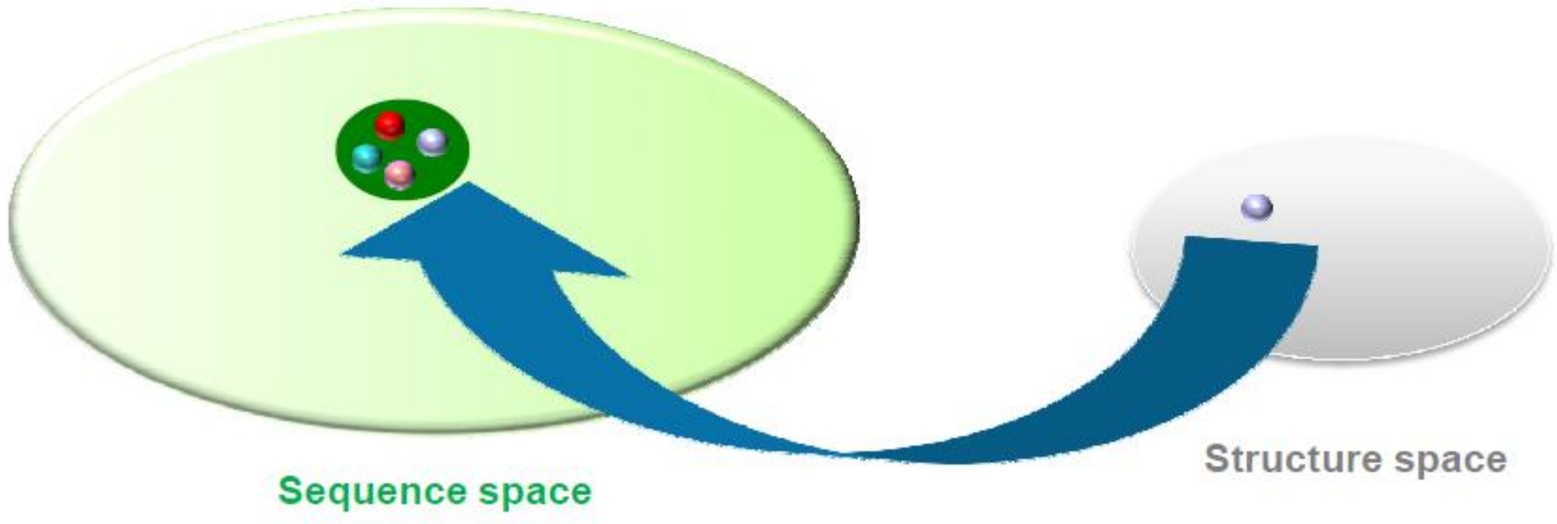
- **PROTEIN (RE)DESIGN**




**multiple sequences**

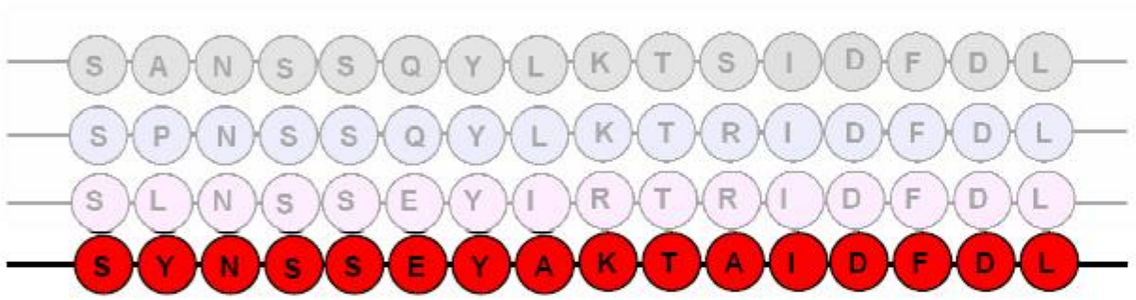


**1 structure**

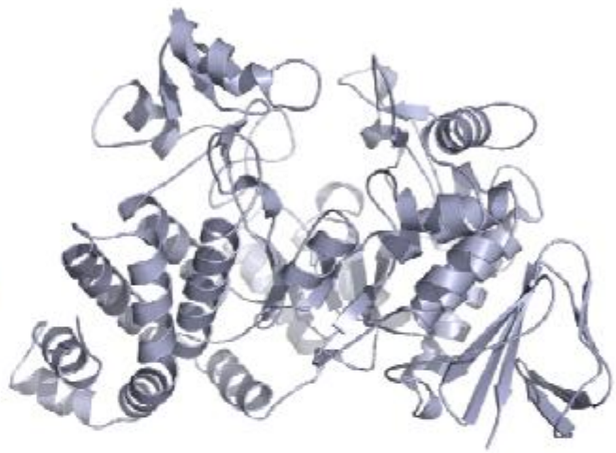


# PROTEIN FOLD PREDICTION VERSUS PROTEIN (RE)DESIGN

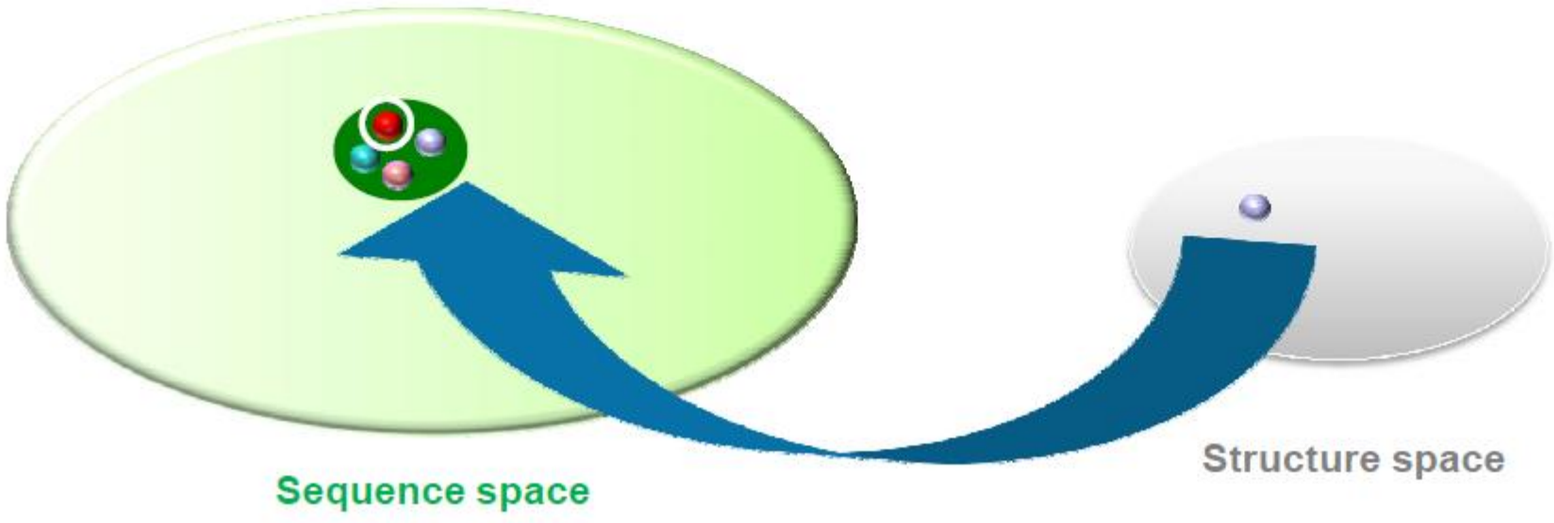
- **PROTEIN (RE)DESIGN**




« optimal » sequence



1 structure

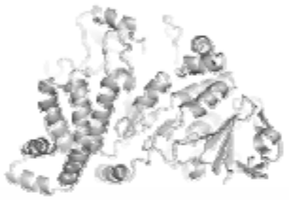


# GENERAL COMPONENTS OF A COMPUTATIONAL PROTEIN DESIGN (CPD) TOOL

LISBP

## INPUT

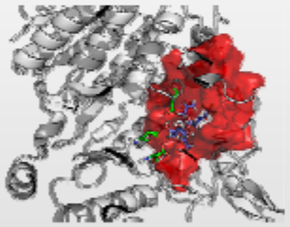
### 3D structures



Protein

Ligand

### Residues to redesign



### Conformational description

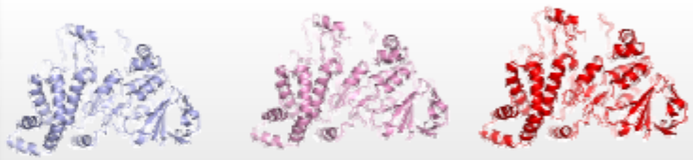
- Protein backbone
- Protein side-chains
- Ligand

## SEARCH

Algorithm efficiently samples vast sequence-conformation search space

## OUTPUT

Protein structures & sequences



## SCORE

Scoring function discriminates optimal from sub-optimal sequence-conformation

Amino acid sequences

Parental ……TPSDLRLDQFMKRWAPTLTEAEASAY……

Mutant ……TPSDLRLLQFMKQWAPTLTYAEASAY……

3 mutable residues

L

rotamer 1 rotamer 2

D

R

Q

Y

E

Side Chains

Backbone

3D Protein Structure

ARG amino acid rotamers

**Representation of a sequence-conformation model. A) Partial view of protein 3D structure showing as example three mutable positions. B) Zoom on a polypeptide segment. C) Illustration of accessible rotamers for an amino acid type.**

# Weighted Constraint Satisfaction Problem (X,D,F)

- **X**: one variable per mutable residue
- **D**: domain of every variable is defined as the set of all combinations of amino acids (20) and spatial conformations
- **F**: unary and binary soft constraints to encode the energy function for every pair of residues

$$E = E_{\varnothing} + \sum_{i} E(i_r) + \sum_{i} \sum_{j>i} E(i_r, j_s)$$

*Et' : Self-energy of the rigid region*

*E(ir) Interaction energy between rotamers and rigid region*

*E(ir, js) Interaction energy between rotamers*

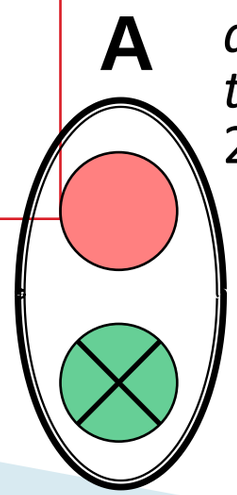
Table 1: For each instance: protein (PDB id.), number of mutable residues, maximum domain and CPU-time for solving using `maxhs`, `daoopt`, `osprey`, `cplex`, `mplp`, and `toulbar2`. A '-' indicates the corresponding solver did not prove optimality within the 9,000-second time-out. A '!' indicate solver stops with a SEGV signal.

| PDB id. | $n$ | $d$ | `maxhs` | `daoopt` | `osprey` | `cplex` | `mplp` | `toulbar2` |
|---|---|---|---|---|---|---|---|---|
| 2TRX | 11 | 44 | 4,086 | 268.6 | 31.5 | 2.6 | 2.8 | **0.1** |
| 1PGB | 11 | 45 | 5,209 | 300.4 | 135.3 | 3.6 | 0.5 | **0.1** |
| 1HZ5 | 12 | 45 | 5,695 | 350.2 | 75.0 | 7.6 | 16.7 | **0.1** |
| 1UBI | 13 | 45 | - | 826.9 | 2,812.6 | 139.2 | 37.3 | **0.2** |
| 1PGB | 11 | 148 | - | - | 8,695.2 | - | 1,291 | **4.3** |
| 1HZ5 | 12 | 148 | - | - | 2,398.3 | 1,555 | 1,217 | **2.4** |
| 1UBI | 13 | 148 | - | - | - | - | - | **1,557** |
| 2PCY | 18 | 44 | - | - | 1,281.1 | 26.9 | 14.5 | **0.2** |
| 2DHC | 14 | 148 | - | - | - | - | 5,388 | **14.1** |
| 1CM1 | 17 | 148 | - | - | 138.4 | 473.1 | 87.5 | **3.3** |
| 1MJC | 28 | 182 | 3,698 | 631.7 | 4.6 | 4.1 | 0.8 | **0.1** |
| 1CSP | 30 | 182 | - | - | 200.0 | 1,380 | 1,264 | **0.8** |
| 1BK2 | 24 | 182 | - | - | 93.2 | 125.0 | 114.9 | **0.6** |
| 1SHG | 28 | 182 | - | - | 138.0 | 39.4 | ! | **0.2** |
| 1CSK | 30 | 49 | - | - | 41.7 | 12.5 | 9.6 | **0.1** |
| 1SHF | 30 | 56 | - | - | 44.3 | 8.6 | 3.1 | **0.1** |
| 1FYN | 23 | 186 | - | - | 622.0 | 2,548 | 3,136 | **2.8** |
| 1PIN | 28 | 194 | - | - | - | - | - | **3.7** |
| 1NXB | 34 | 56 | - | - | 11.1 | 17.0 | 4.5 | **0.2** |
| 1TEN | 39 | 66 | - | - | 113.0 | 45.4 | 17.1 | **0.2** |
| 1POH | 46 | 182 | - | - | 77.9 | 29.0 | 13.1 | **0.3** |
| 2DRI | 37 | 186 | - | - | - | - | 4,458 | **42.8** |
| 1FNA | 38 | 48 | - | - | 3,310 | 124.9 | 121.2 | **0.5** |
| 1UBI | 40 | 182 | - | - | - | 2,572 | 979.4 | **2.4** |
| 1C9O | 43 | 182 | - | - | 2,310 | 1,635 | 155.7 | **1.8** |
| 1CTF | 39 | 56 | - | - | - | 263.2 | 549.2 | **0.7** |
| 2PCY | 46 | 56 | - | - | 2,080 | 54.0 | 20.3 | **0.4** |
| 1DKT | 46 | 190 | - | - | 5,420 | 1,254 | 3,103 | **2.5** |
| 2TRX | 61 | 186 | - | - | 487.0 | 765.0 | 344.1 | **0.9** |
| 1CM1 | 42 | 186 | - | - | - | - | - | **17.4** |
| 1BRS | 44 | 194 | - | - | - | - | - | **346.5** |
| 1CDL | 40 | 186 | - | - | - | - | - | **341.8** |
| 1LZ1 | 59 | 57 | - | - | - | 601.6 | 1,084 | **1.5** |
| 1GVP | 52 | 182 | - | - | - | - | - | **361.8** |
| 1RIS | 56 | 182 | - | - | - | - | 8,483 | **288.4** |
| 2RN2 | 69 | 66 | - | - | - | 480.8 | 565.2 | **1.2** |
| 1CSE | 97 | 183 | - | - | 367.0 | 172.9 | 60.9 | **0.7** |
| 1HNG | 85 | 182 | - | - | 5,590 | 2,360 | 5,934 | **2.8** |
| 3CHY | 74 | 66 | - | - | - | - | 8,691 | **59.6** |
| 1L63 | 83 | 182 | - | - | - | 1,480 | 1,779 | **2.9** |
| | | | 3 | 5 | 25 | 29 | 33 | **40** |

# Dominance rules in comb. optimization
# Value substitutability

- AI & OR
  - E Freuder. *Eliminating interchangeable values in constraint satisfaction problems*. In AAAI 1991
  - A Koster. *Frequency assignment: Models and Algorithms*. Ph.D. thesis, 1999
  - R Niedermeier, P Rossmanith. *New upper bounds for maximum satisfiability*. J. Algorithms 36(1), 2000
  - S Bistarelli, B Faltings, N Neagu. *Interchangeability in Soft CSPs*. In CP 2002
  - A Jouglet, J Carlier. *Dominance rules in combinatorial optimization problems*. EJOR 212(3), 2011
  - G Chu, P Stuckey. *A generic method for identifying and exploiting dominance relations*. In CP 2012
  - C Lecoutre, O Roussel, D Dehani. *WCSP Integration of Soft Neighborhood Substitutability*. In CP 2012

- Computational Protein Design
  - J Desmet, M Maeyer, B Hazes, I Lasters,. *The dead-end elimination theorem and its use in protein side-chain positioning*. Nature 356, 1992
  - R Goldstein. *Efficient rotamer elimination applied to protein side-chains and related spin glasses*. Biophysical Journal 66(5), 1994
  - N Pierce, J Spriet, J Desmet, S Mayo. *Conformational splitting: A more powerful criterion for dead-end elimination*. Journal of Computational Chemistry. 21(11), 2000
  - I Georgiev, R Lilien, B Donald. *Improved pruning algorithms and divide-and-conquer strategies for dead-end elimination, with application to protein design*. Bioinformatics 22(14), 2006



# Dead-End Elimination rules

Prune value (*x,b*), dominated by (*x,a*) if:

- **Rule 1**: (Desmet *et al*, Nature 1992)

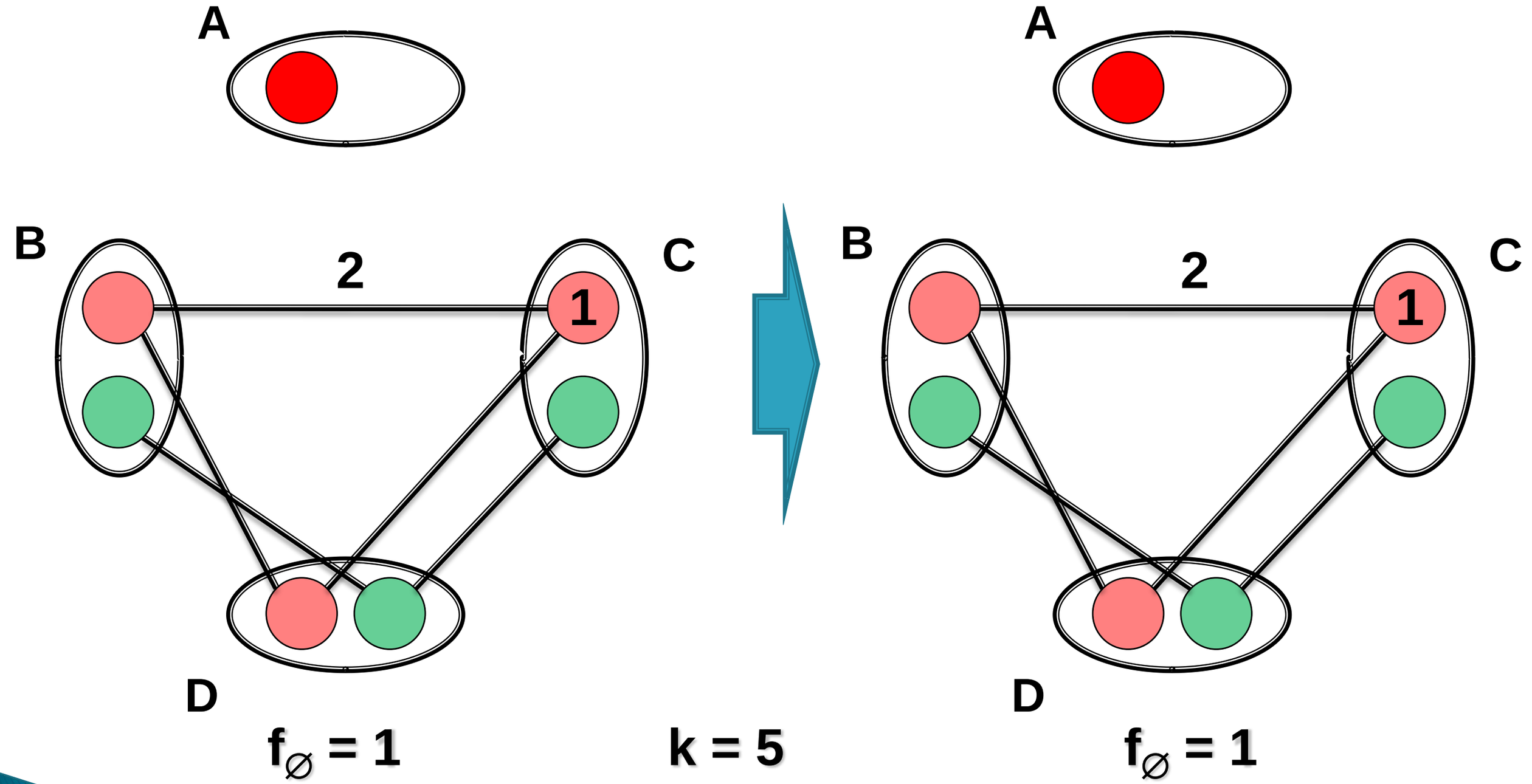
$$\sum_{f_S \in \Gamma(x)} \max_{t \in l(S \setminus \{x\})} f_S(t \cup \{(x,a)\}) \leq \sum_{f_S \in \Gamma(x)} \min_{t \in l(S \setminus \{x\})} f_S(t \cup \{(x,b)\})$$

- **Rule 2**: (Goldstein, Bio. J. 1994) (Koster, 1999)

$$\sum_{f_S \in \Gamma(x)} \max_{t \in l(S \setminus \{x\})} f_S(t \cup \{(x,a)\}) - f_S(t \cup \{(x,b)\}) \leq 0$$

Rule 2 is always stronger than rule 1 and it has been improved in (Givry et al, CP 2013)

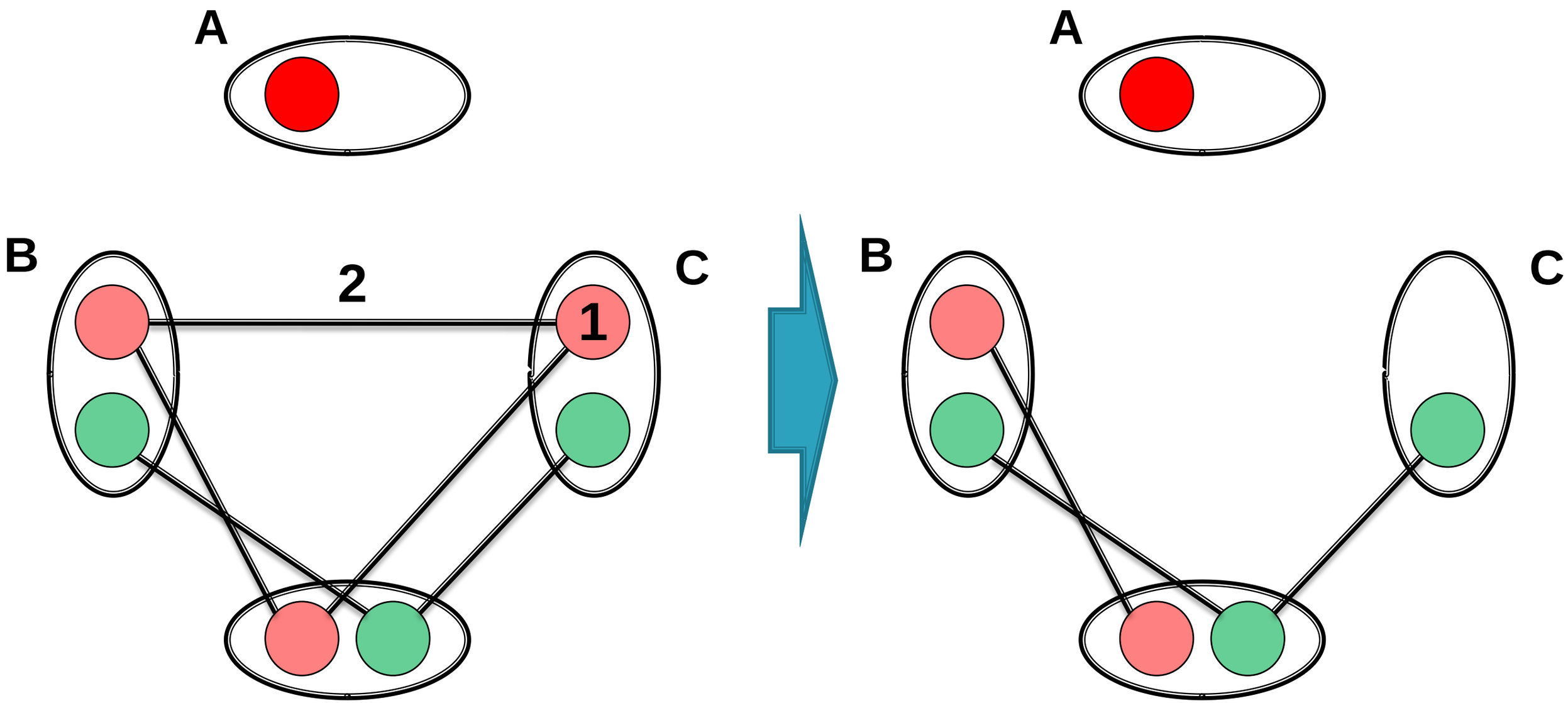
# Pruning by dominance




$f_{\varnothing} = 1$ $k = 5$ $f_{\varnothing} = 1$

Problem is EDAC

# Pruning by dominance

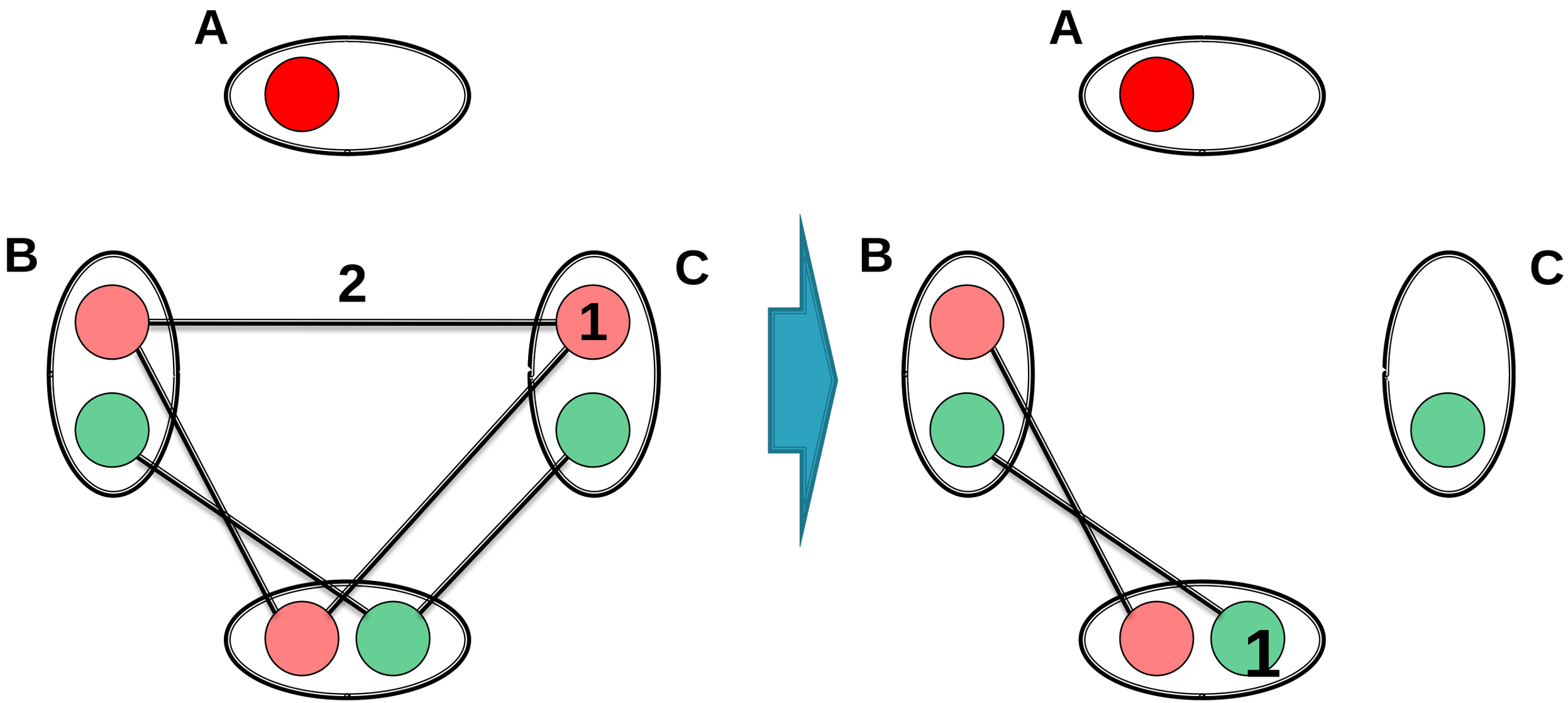



$f_{\varnothing} = 1$ **k = 5** $f_{\varnothing} = 1$

Problem is EDAC

Rule 1 and Rule 2(+) find (*C*,*red*) is dominated by (*C*,*green*)

# Pruning by dominance

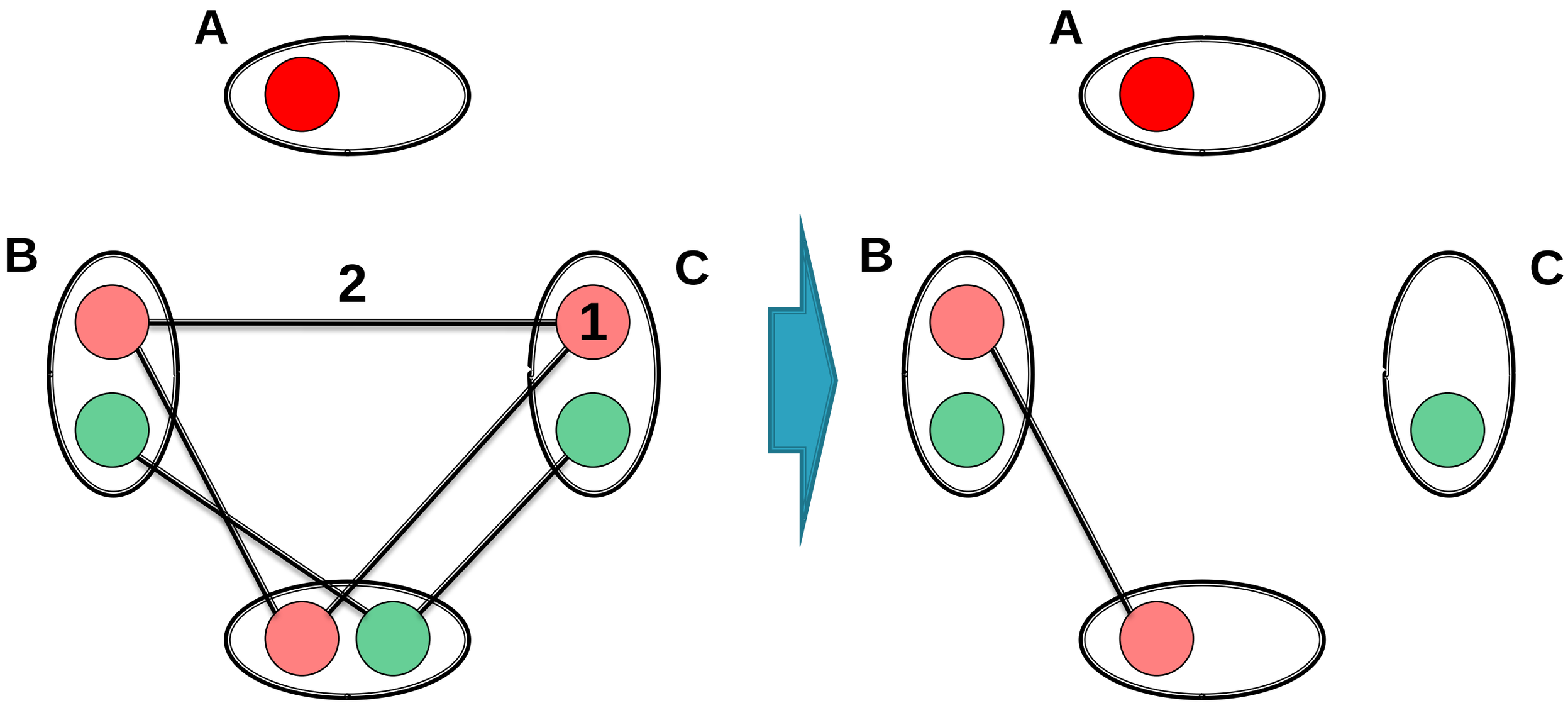



$f_{\varnothing} = 1$ **k = 5** $f_{\varnothing} = 1$

Problem is EDAC

Rule 1 and Rule 2(+) find (*C*,*red*) is dominated by (*C*,*green*)

# Pruning by dominance

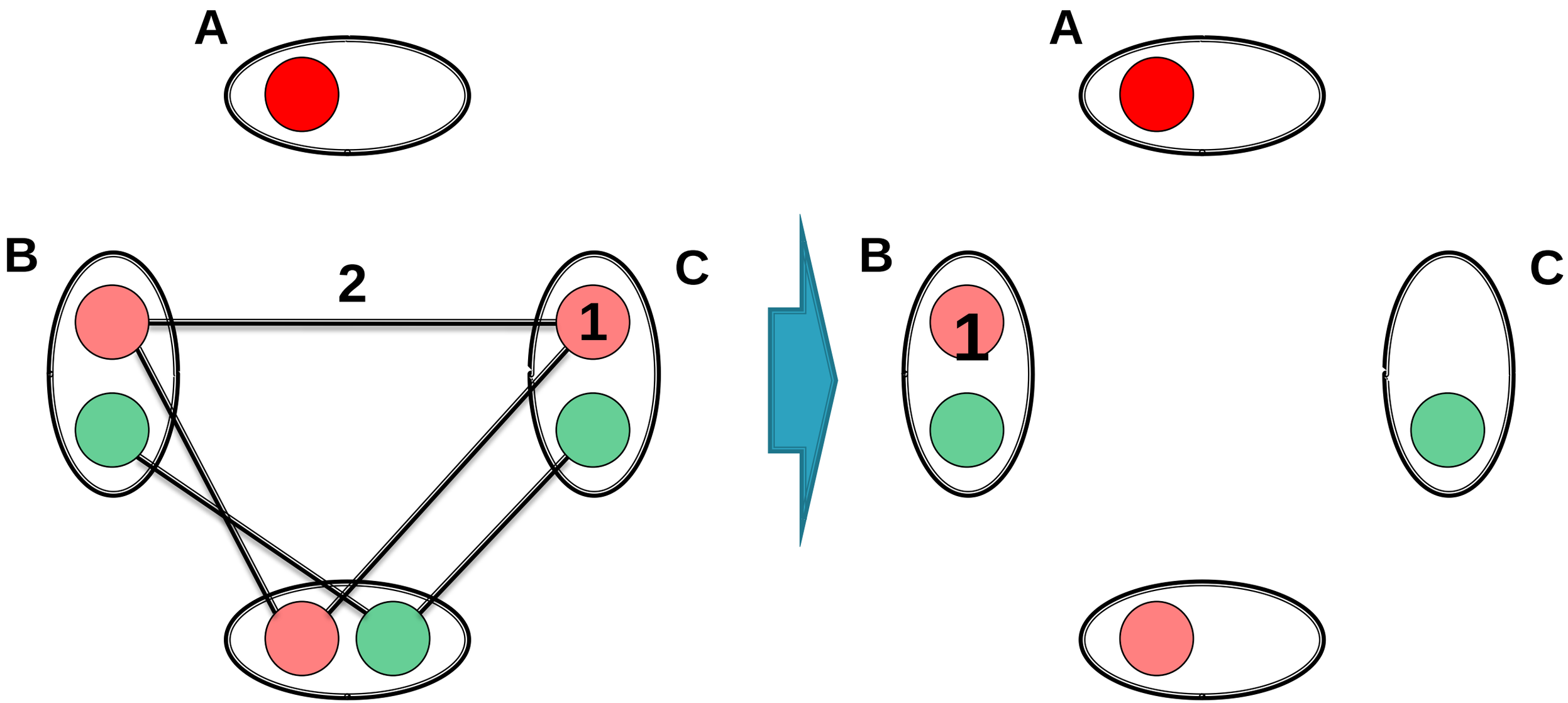



$f_{\varnothing} = 1$ **k = 5** $f_{\varnothing} = 1$

Problem is EDAC

Rule 1 and Rule 2(+) find (*C*,*red*) is dominated by (*C*,*green*)

# Pruning by dominance

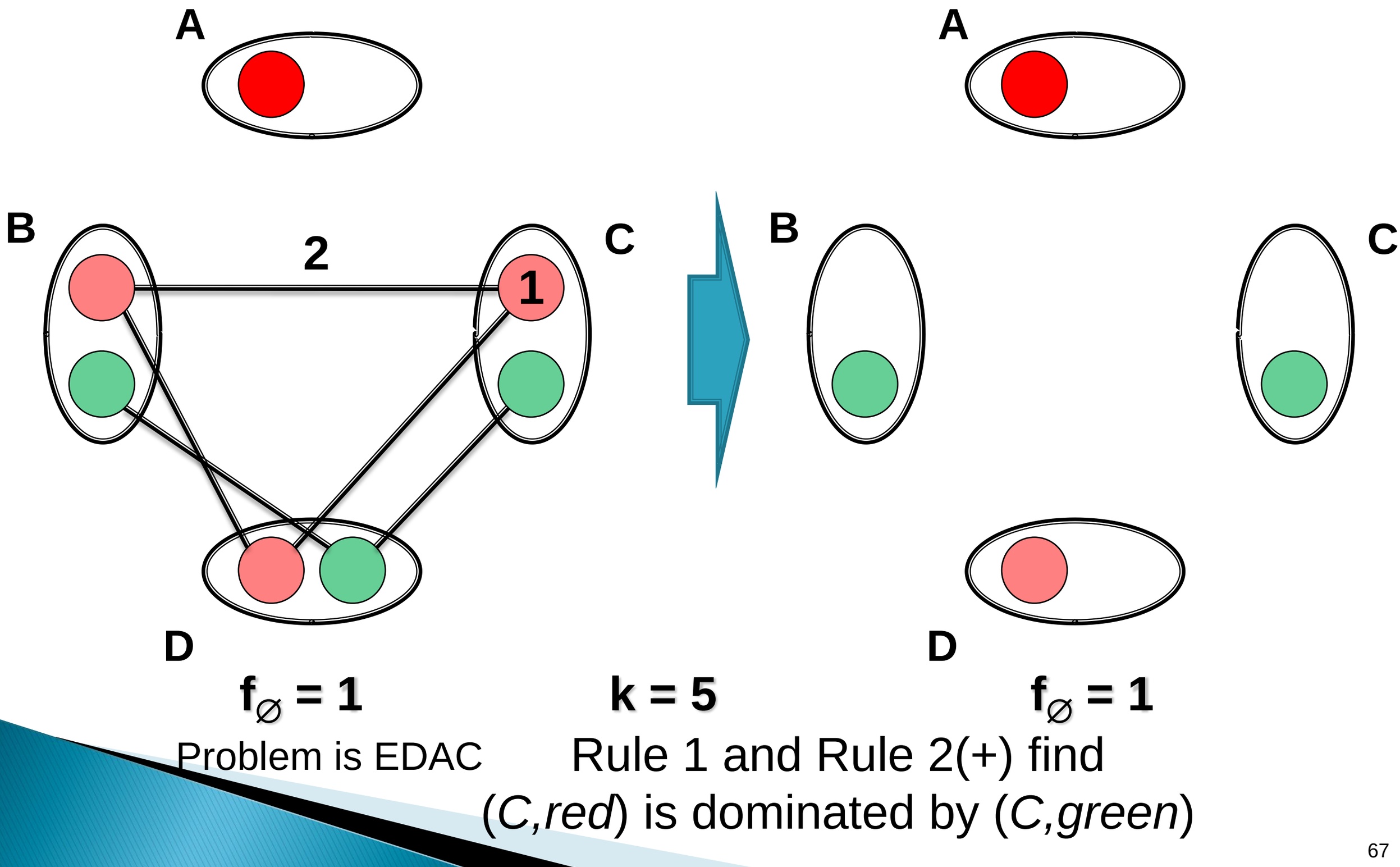



$f_{\varnothing} = 1$ **k = 5** $f_{\varnothing} = 1$

Problem is EDAC

Rule 1 and Rule 2(+) find (*C,red*) is dominated by (*C,green*)

# Pruning by dominance




$f_{\varnothing} = 1$ **k = 5** $f_{\varnothing} = 1$

Problem is EDAC

Rule 1 and Rule 2(+) find (*C*,*red*) is dominated by (*C*,*green*)

Table 2: For each instance: CPU-time for solving using `toulbar2` and different combinations of options for DVO and DEE. A '-' indicates that the corresponding solver did not prove optimality with the 9,000-second time-out. The `prev` column gives the results obtained using the vanilla toulbar2 for reference. The DVO corresponds to the activation of the new variable ordering heuristics described in Section 4.1. This option is kept activated in all the remaining columns. These columns correspond respectively to additionally maintaining DEE[1] during search, pre-processing using DEE, doing both, and maintaining DEE during search. The last line reports the number of times a method was faster than the others.

| PDB | $n$ | $d$ | `prev` | DVO | DEE[1] | DEE$_{pre}$ | DEE$_{pre}$ +DEE[1] | DEE |
|---|---|---|---|---|---|---|---|---|
| 2TRX | 11 | 44 | **0.1** | **0.1** | **0.1** | **0.1** | **0.1** | **0.1** |
| 1PGB | 11 | 45 | **0.1** | **0.1** | **0.1** | **0.1** | **0.1** | **0.1** |
| 1HZ5 | 12 | 45 | **0.1** | **0.1** | **0.1** | **0.1** | **0.1** | **0.1** |
| 1UBI | 13 | 45 | **0.2** | **0.2** | **0.2** | **0.2** | 0.3 | 0.5 |
| 1PGB | 11 | 148 | 4.3 | 3.8 | 3.4 | **3.1** | 8.6 | 15.1 |
| 1HZ5 | 12 | 148 | 2.4 | 2.4 | 2.3 | **2.2** | 3.1 | 3.5 |
| 1UBI | 13 | 148 | 1,557 | **1,068** | 1,736 | 1,133 | 1,162 | - |
| 2PCY | 18 | 44 | **0.2** | **0.2** | **0.2** | **0.2** | **0.2** | **0.2** |
| 2DHC | 14 | 148 | 14.1 | 8.0 | **7.0** | **7.0** | 14.5 | 52.0 |
| 1CM1 | 17 | 148 | 3.3 | **3.1** | 3.2 | **3.1** | **3.1** | **3.1** |
| 1MJC | 28 | 182 | **0.1** | **0.1** | **0.1** | **0.1** | **0.1** | **0.1** |
| 1CSP | 30 | 182 | 0.8 | 0.6 | **0.5** | 0.7 | 0.7 | 0.8 |
| 1BK2 | 24 | 182 | 0.6 | 0.6 | 0.6 | **0.5** | 0.7 | **0.5** |
| 1SHG | 28 | 182 | **0.2** | **0.2** | **0.2** | **0.2** | **0.2** | **0.2** |
| 1CSK | 30 | 49 | **0.1** | **0.1** | **0.1** | **0.1** | **0.1** | **0.1** |
| 1SHF | 30 | 56 | **0.1** | **0.1** | **0.1** | **0.1** | **0.1** | **0.1** |
| 1FYN | 23 | 186 | 2.8 | 2.9 | **2.6** | 3.0 | 3.2 | 3.8 |
| 1PIN | 28 | 194 | 3.7 | **3.0** | **3.0** | 4.8 | 6.2 | 12.0 |
| 1NXB | 34 | 56 | **0.2** | **0.2** | **0.2** | **0.2** | **0.2** | **0.2** |
| 1TEN | 39 | 66 | **0.2** | **0.2** | **0.2** | **0.2** | **0.2** | **0.2** |
| 1POH | 46 | 182 | **0.3** | **0.3** | **0.3** | 0.4 | 0.4 | 0.4 |
| 2DRI | 37 | 186 | 42.8 | 16.4 | 37.7 | **9.6** | 15.5 | 51.2 |
| 1FNA | 38 | 48 | 0.5 | 0.4 | **0.3** | 0.4 | 0.4 | 0.5 |
| 1UBI | 40 | 182 | 2.4 | 1.0 | **0.7** | 0.9 | 0.9 | 1.3 |
| 1C9O | 43 | 182 | 1.8 | **1.5** | 1.7 | 2.3 | 2.4 | 3.6 |
| 1CTF | 39 | 56 | 0.7 | 0.9 | **0.6** | **0.6** | 0.7 | 0.8 |
| 2PCY | 46 | 56 | **0.4** | **0.4** | **0.4** | **0.4** | **0.4** | **0.4** |
| 1DKT | 46 | 190 | 2.5 | 2.8 | **2.4** | 2.6 | 2.7 | 3.9 |
| 2TRX | 61 | 186 | **0.9** | **0.9** | **0.9** | 1.8 | 1.7 | 1.9 |
| 1CM1 | 42 | 186 | 17.4 | 11.8 | 13.2 | **8.6** | 11.6 | 20.0 |
| 1BRS | 44 | 194 | 346.5 | 241.4 | 135.4 | 70.4 | **60.1** | 129.0 |
| 1CDL | 40 | 186 | 341.8 | 198.1 | 159.0 | **79.6** | 128.8 | 286.4 |
| 1LZ1 | 59 | 57 | 1.5 | 1.1 | 1.0 | **0.9** | 1.0 | 1.1 |
| 1GVP | 52 | 182 | 361.8 | 248.5 | 408.2 | **38.3** | 66.8 | 163.5 |
| 1RIS | 56 | 182 | 288.4 | 147.4 | 77.9 | 37.8 | **28.8** | 122.8 |
| 2RN2 | 69 | 66 | 1.2 | **1.1** | **1.1** | 1.2 | **1.1** | 1.2 |
| 1CSE | 97 | 183 | 0.7 | 0.8 | **0.6** | **0.6** | **0.6** | **0.6** |
| 1HNG | 85 | 182 | 2.8 | 2.4 | **2.3** | 3.1 | 2.8 | 3.6 |
| 3CHY | 74 | 66 | 59.6 | 27.9 | 10.7 | **10.6** | 14.9 | 20.3 |
| 1L63 | 83 | 182 | 2.9 | 2.8 | **2.3** | 2.4 | 2.5 | 2.7 |
| | | | 14 | 19 | 26 | 25 | 16 | 14 |

Table 3: For each instance solved by both `cplex` and `toulbar2`, we report the number of nodes explored by each solver (with the number of backtracks in parentheses when available) and the number of nodes per minute developed. `toulbar2` is the vanilla version, `toulbar2`$^+$ uses the new variable ordering heuristics and DEE as pre-processing.

| PDB id. | $n$ | $d$ | `cplex` | | `toulbar2` | | `toulbar2`$^+$ | |
|---|---|---|---|---|---|---|---|---|
| | | | nodes | nd/min | nodes (bt) | nd/min | nodes (bt) | nd/min |
| 2TRX | 11 | 44 | 0 | - | 8 (0) | 6857 | 10 (1) | 7500 |
| 1PGB | 11 | 45 | 0 | - | 17 (1) | 11333 | 16 (1) | 10667 |
| 1HZ5 | 12 | 45 | 0 | - | 25 (5) | 15000 | 29 (7) | 17400 |
| 1UBI | 13 | 45 | 51 | 22.0 | 143 (61) | 39000 | 82 (31) | 24600 |
| 1HZ5 | 12 | 148 | 0 | - | 89 (34) | 2225 | 54 (16) | 1453 |
| 2PCY | 18 | 44 | 0 | - | 53 (6) | 13826 | 40 (9) | 10909 |
| 1CM1 | 17 | 148 | 0 | - | 14 (0) | 258 | 0 (0) | 0 |
| 1MJC | 28 | 182 | 0 | - | 22 (0) | 14667 | 2 (0) | 1714 |
| 1CSP | 30 | 182 | 547 | 23.8 | 540 (245) | 42078 | 42 (16) | 3877 |
| 1BK2 | 24 | 182 | 3 | 1.4 | 28 (3) | 2800 | 19 (3) | 2375 |
| 1SHG | 28 | 182 | 214 | 326 | 268 (101) | 69913 | 51 (16) | 19125 |
| 1CSK | 30 | 49 | 0 | - | 38 (5) | 20727 | 14 (3) | 7636 |
| 1SHF | 30 | 56 | 0 | - | 35 (4) | 17500 | 12 (0) | 6000 |
| 1FYN | 23 | 186 | 0 | - | 84 (20) | 1819 | 43 (8) | 863 |
| 1NXB | 34 | 56 | 0 | - | 30 (0) | 9474 | 14 (0) | 4667 |
| 1TEN | 39 | 66 | 0 | - | 75 (6) | 20455 | 22 (5) | 6600 |
| 1POH | 46 | 182 | 0 | - | 111 (5) | 21484 | 15 (0) | 2195 |
| 1FNA | 38 | 48 | 0 | - | 189 (47) | 25200 | 84 (28) | 12600 |
| 1UBI | 40 | 182 | 287 | 6.7 | 1,669 (766) | 41900 | 539 (228) | 36337 |
| 1C9O | 43 | 182 | 49 | 1.8 | 222 (57) | 7525 | 82 (16) | 2112 |
| 1CTF | 39 | 56 | 94 | 21.4 | 294 (95) | 24845 | 110 (33) | 10820 |
| 2PCY | 46 | 56 | 0 | - | 62 (5) | 10629 | 20 (0) | 3158 |
| 1DKT | 46 | 190 | 0 | - | 210 (36) | 5122 | 134 (24) | 3045 |
| 2TRX | 61 | 186 | 6 | 0.5 | 111 (14) | 7239 | 85 (16) | 2818 |
| 1LZ1 | 59 | 57 | 735 | 73.5 | 807 (308) | 31855 | 178 (53) | 11609 |
| 2RN2 | 69 | 66 | 0 | - | 105 (17) | 5385 | 110 (17) | 5500 |
| 1CSE | 97 | 183 | 0 | - | 94 (0) | 8418 | 9 (0) | 915 |
| 1HNG | 85 | 182 | 48 | 1.2 | 411 (110) | 8745 | 96 (21) | 1870 |
| 1L63 | 83 | 182 | 0 | - | 196 (17) | 4055 | 58 (3) | 1468 |

# Solvers and benchmarks

- toulbar2

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

  (Open source WCSP, MaxSAT, MPE solver in C++)
  - Contribution by UPC (J. Larrosa's team) & CSIC (P. Meseguer's team)
  - Contribution on soft global cost functions by CUHK (J. Lee's team)
  - Contribution on large neighborhood local search by Caen University (P. Boizumault's team)
- numberjack

  http://github.com/eomahony/Numberjack/tree/fzn
  - Multi-solver (Mistral,CPLEX,Gurobi,SCIP,Minisat,toulbar2) platform in python
  - minizinc reader
- Large set of benchmarks

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

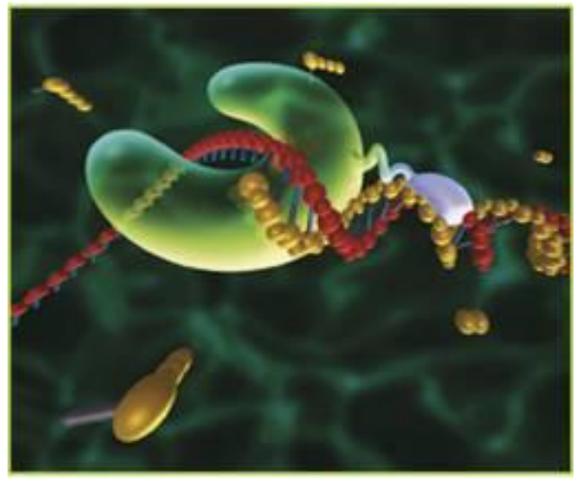
  http://www.costfunction.org

# WCB'14

- Workshop on Constraint-Based-Methods for Bioinformatics

  - Held with 20th International Conference on Principles and Practice of Constraint Programming (CP 2014)
  - Lyon, France
  - September 8th 2014
  - Organizers : Nicos Angelopoulos and Simon de Givry

# Gene regulatory network



- Structure learning (Bayesian net, GGM, RF)
  - M. Vignes, J. Vandel, D. Allouche, N. RamadanAlban, C. CiercoAyrolles, T. Schiex, B. Mangin, S. de Givry. Gene regulatory network reconstruction using bayesian networks, the Dantzig selector, the lasso and their meta-analysis. PLoS ONE, 6(12), 2011.
  - J Vandel, B Mangin, and S de Givry. New Local Move Operators for Bayesian Network Structure Learning. In Proc. of PGM-12, Granada, Spain, 2012.
  - DREAM5 System Genetics evaluation, 2010
  - GeneBayesNet

http://carlit.toulouse.inra.fr/genebayesnet