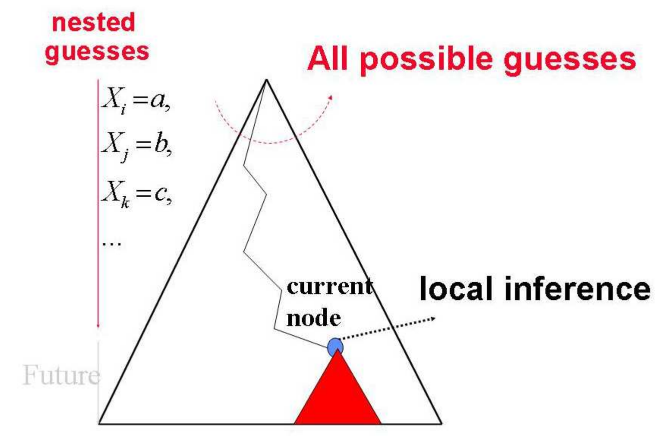
Improving Tree Decomposition Methods With Function Filtering

Marti Sanchez, Pedro Meseguer
IIIA - Artificial Intelligence Research Institute
CSIC - Spanish Council for Scientific Research
Campus UAB, 08193 Bellaterra, Catalonia, Spain.
{marti,pedro}@iiia.csic.es

Javier Larrosa
LSI - Lenguajes y Sistemas Informaticos
UPC - Universitat Politecnica de Catalunya
Jordi Girona, 08028 Barcelona, Spain
{larrosa}@lsi.upc.es

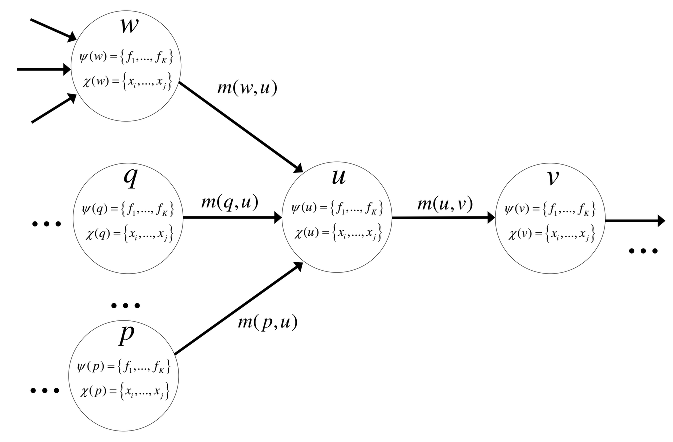
Search



- Main step: guessing
- Bottleneck: exponential search tree traversal
- Time Complexity: $O(d^n)$
- Space Complexity: $O(n.d)$
- Average Time Complexity: better than worst case

Inference

- Main step: message passing
- Bottleneck: memory storage
- Time Complexity: $O(exp(tw))$. tw tree width
- Space Complexity: $O(exp(tw)$
- Average Space Complexity: close to worst



Cluster and Mini Cluster Tree Elimination

WCSP and Valuation Structures

Definition 1 A valuation structure $S(k) = \langle [0,1,...,k], \oplus, \geq \rangle$ *where:*

- $k \in [1, \ldots, \infty]$,
- $a \oplus b = min\{k, a+b\}$, *and*
- $\geq$ *is the standard order among naturals.*

Definition 2 *A* Weighted CSP *(WCSP) is* $\langle X, D, C, S(k) \rangle$ *where:*

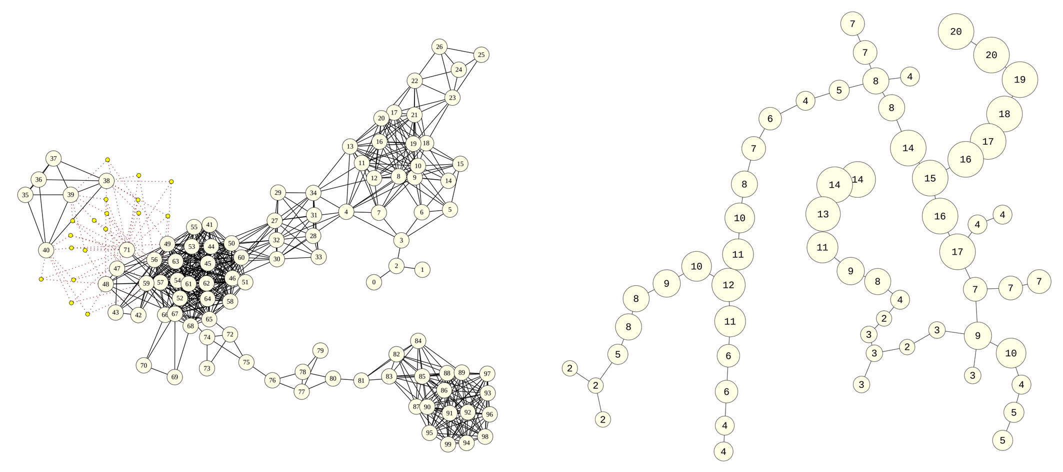
- $X = \{x_1, ..., x_n\}$ *and* $D = \{D_1, ..., D_n\}$ *as in CSP*
- C *is a finite set of cost functions:*

$$f(t) = \begin{cases} 0 & \text{if } t \text{ is allowed} \\ [1 \ldots k\text{-}1] & \text{if } t \text{ is partially allowed} \\ k & \text{if } t \text{ is totally forbidden} \end{cases}$$

Tree Decomposition

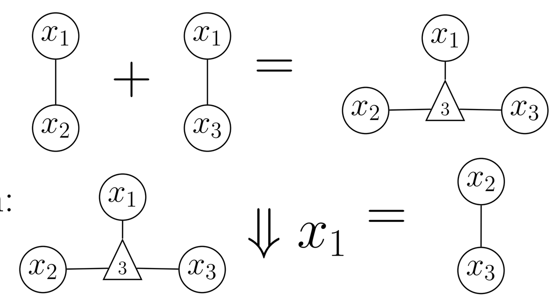
Definition 3 *A tree decomposition* $\langle T, \chi, \psi \rangle$*, where* $T = \langle V, E \rangle$ *is a tree.* $\chi(v) \subseteq X$ *and* $\psi(v) \subseteq C$ *satisfy:*

1. *for all* $f \in C$*, there is exactly one vertex* $v \in V$ *s.t.* $f \in \psi(v)$ *and* $var(f) \subseteq \chi(v)$
2. *for all* $x \in X$*, the set* $\{v \in V | x \in \chi(v)\}$ *induces a connected subtree of* T

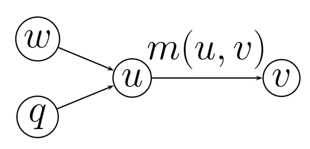


Operation on Functions

- Sum:
  $(f+g)(t \cup t') = f(t) \oplus g(t')$
- Projecting out by minimization:
  $f_{\Downarrow x}(t) = \min_{a \in D_x}(f(a \cup t))$



CTE and MCTE



$$m(u,v) = \sum_{i, i \neq v} m(i,u) + \sum_{f \in \Psi(v)} f$$

- CTE solves WCSP by sending msgs $m(u, v)$ along the edges of a tree decomposition.
- CTE time and space complexity are $O(exp(tw))$.
- MCTE(r) approximates CTE limiting the arity: $M(u, v) \leq m(u, v)$

CTEf and MCTEf

Function Filtering

Definition 4 *The* function filtering *operation applied to a function* f *from a set of functions* H*, noted* $\overline{f}^H$ *is:*

$$\overline{f}^H(t) = \begin{cases} f(t) & \text{if } \left(\bigoplus_{h \in H} h(t)\right) \oplus f(t) < k \\ k & \text{otherwise} \end{cases}$$

Property 1 $\overline{f}^G + \overline{g}^F = f + g$

Property 2 $\overline{f+g}^H = \overline{\overline{f}^H + \overline{g}^H}^H$

A Filtering Tree decomposition

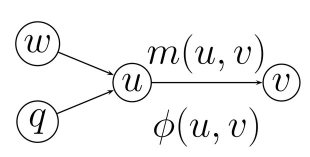
Definition 5 *A filtering tree-decomposition of a WCSP is a tuple* $\langle T, \chi, \psi, \phi \rangle$ *where:*

- $\langle T, \chi, \psi \rangle$ *is a tree-decomposition as in definition3.*
- $\phi(u, v)$ *is a set of functions associated to edge* $(u, v) \in E$*, s.t.*

$$\phi(u, v) \leq m_{(u,v)}$$

Main Idea: delete tuples that will become inconsistent using functions in $\phi(u, v)$.

CTEf and MCTEf



- CTEf sends $m(u, v)$ filtering with functions in $\phi(u, v)$.

$$m(u,v) = \overline{\sum_{i, i \neq v} m(i,u)}^{\phi(u,v)} + \overline{\sum_{f \in \Psi(v)} f}^{\phi(u,v)}$$

Iterative MCTEf

IMCTE

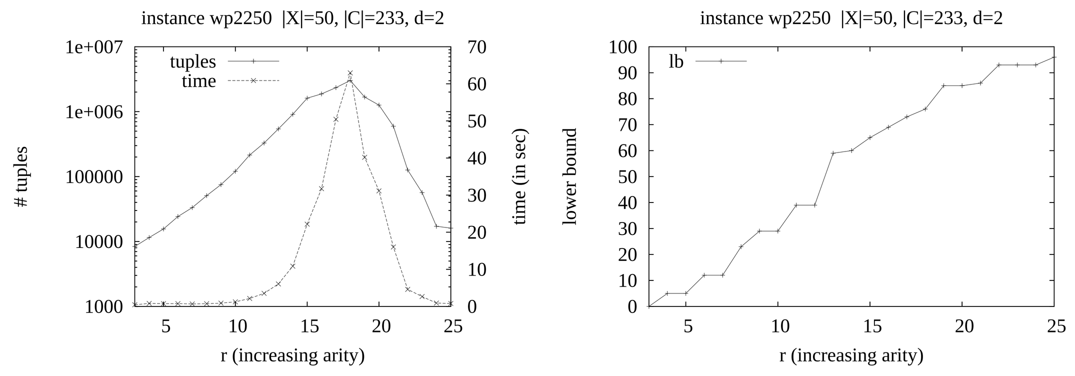
$\phi(u, v) = M(v, u)$ the approximated MCTEf message of a previous execution (one with minor r)

```
procedure IMCTEf(⟨X, D, C, k⟩, ⟨⟨V, E⟩, χ, ψ⟩)
INPUT:  P = ⟨X, D, C, k⟩ is a WCSP instance
        TD = ⟨⟨V, E⟩, χ, ψ⟩ is a tree decomposition.
for each (u, v) ∈ E do φ(u, v) := {∅};
r := 1;
repeat
   MCTEf(r);
   for each (u, v) ∈ E do φ(u, v) := M_(u,v);
   r := r + 1;
until exact solution or exhausted resources
```

Experiments

Experiments are focused in two aspects:

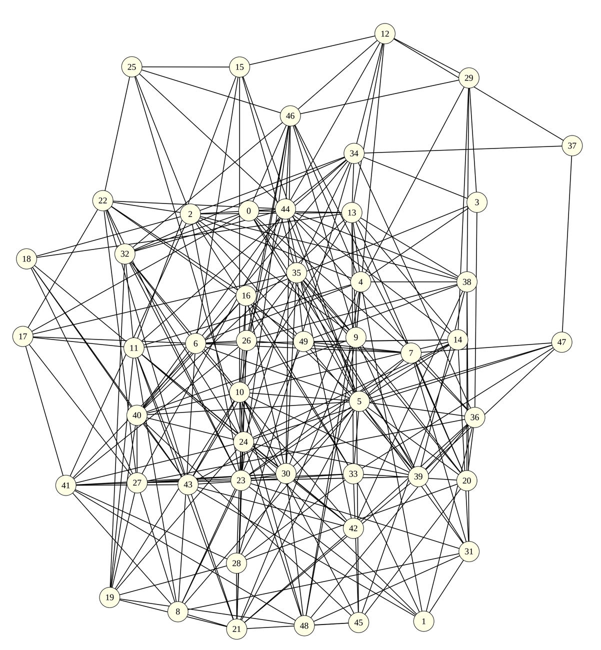
1. Showing that CTEf versus state of the art CTE uses less tuples to find the exact solution.
2. Inside an approximation schema we show that MCTEf(r), exhausts resources at a smaller r and finds worst LB than the iterative version IMCTEf where the previous messages of MCTEf(r) execution are used as filters.



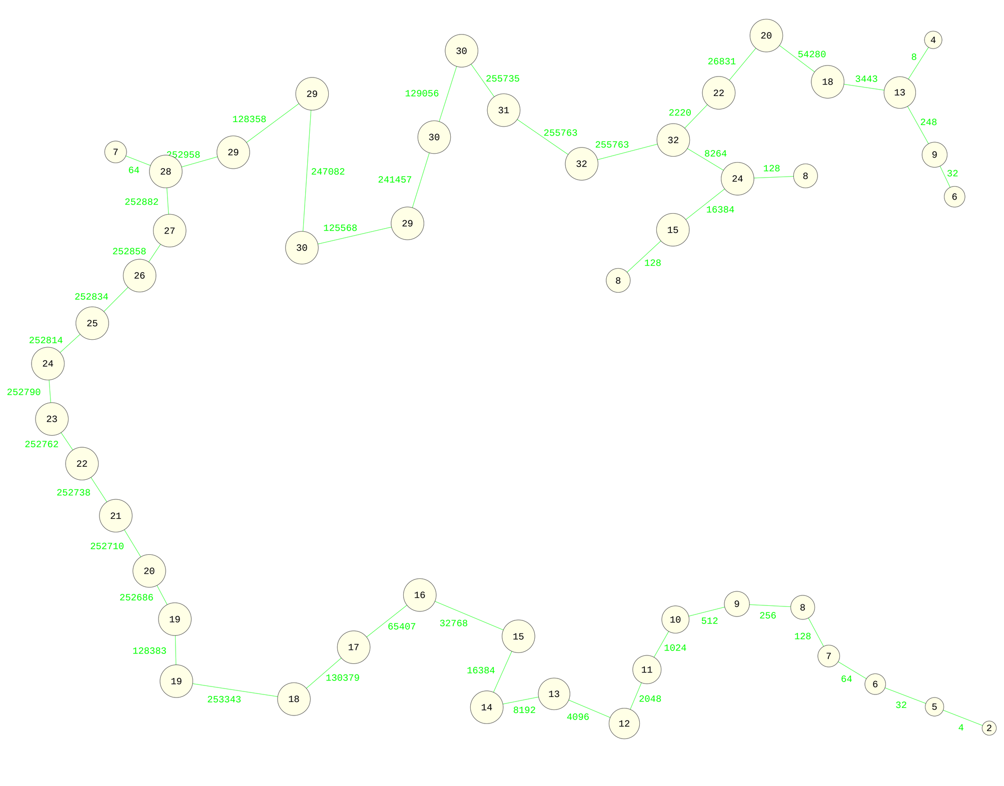
	$\lvert X \rvert$	$\lvert C \rvert$	d	sep	CTE	CTEf	MCTEf(r) r	MCTEf(r) LB	IMCTEf r	IMCTEf LB	UB
dubois100	75	200	2	3	3k	2k					1*
wp2100	50	95	2	9	6k	1k					16*
wp2150	50	138	2	15	302k	40k					34*
wp2200	50	186	2	19	-	733k					69*
wp2250	50	233	2	24	-	-	23	71	25	96	96*
wp2300	50	261	2	26	-	-	22	84	26	132	132*
wp2350	50	302	2	30	-	-	21	129	21	159	212
wp2400	50	340	2	30	-	-	20	70	20	137	212
wp2450	50	378	2	31	-	-	20	130	20	187	257
wp2500	50	418	2	34	-	-	20	168	20	251	318
spot54	67	271	4	11	754k	16k					37*
spot29	82	462	4	14	-	63k					8059*
spot503	143	635	4	8	-	34k					11113*
spot404	100	710	4	20	-	306k					115*
spot505	240	2242	4	22	-	-	12	8044	15	19217	21254
spot42	190	1394	4	26	-	-	13	116001	15	127050	155051

Conclusions

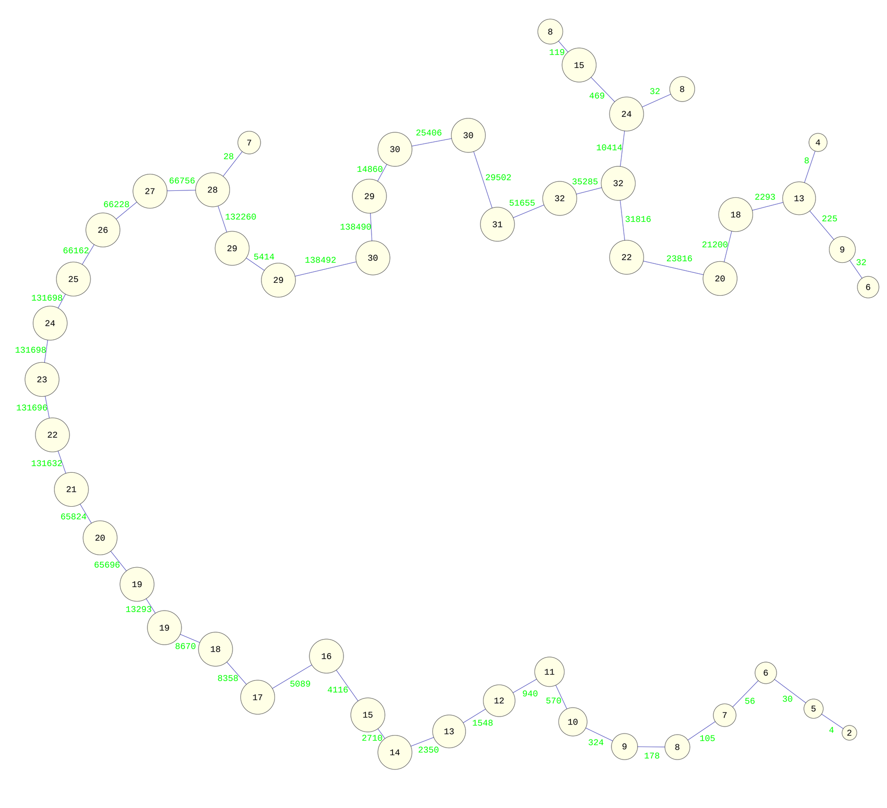
- Main Idea: Delete tuples from functions that we predict will become inconsistent in the future.
- Filtering is a way of using other functions of other clusters, parts of functions of other clusters, sums and approximations of functions of other clusters.
- So is a way of going behind the "exactly one" imposition of the tree decomposition definition.
- Allows to use Upper Bounds and Lower Bounds to delete tuples
- An elegant extension IMCTE uses the messages computed in previous iterations to delete tuples.
- Memory storage is reduced significantly.



wp2300 instance



$r = 19$



$r = 19$ with $r17$ previously computed