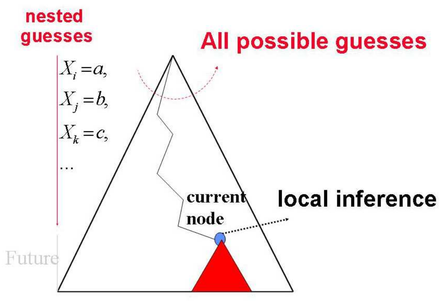
Improving the Applicability of Adaptive Consistency: Preliminary Results

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

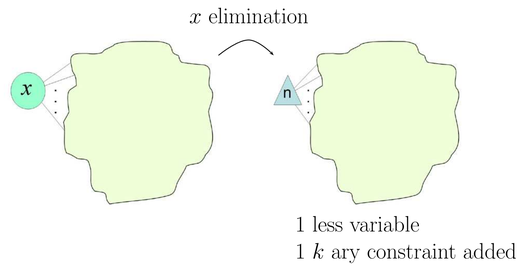
Javier Larrosa
LSI - Lenguajes y Sistemas Informaticos
UPC - Universitat Politecnica de Catalunya
Jordi Girona, 08028 Barcelona, Spain
{larrosa}@lsi.upc.es

Search vs. Inference



- 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

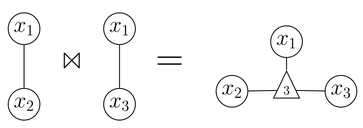
- Main step: variable elimination
- Bottleneck: memory storage
- Time Complexity: $O(n.(2d)^{w*+1})$
- Space Complexity: $O(n.d^{w*})$
- Average Space Complexity: close to worst



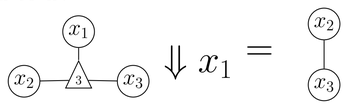
Adaptive Consistency

ADC

- ADC is the reference algorithm to solve CSP by complete inference
- ADC is a specialization of bucket elimination, a more general algorithm used in optimization
- The time and space complexity of ADC are exponential on the problem induced width $w*$.
- ADC follows a Dynamic Programming schema and computes all solutions
- Two basic operations:
  - Join. $c \bowtie r$ is a new constraint with scope $var(c) \cup var(r)$ that includes all the tuples permitted by both constraints when values of common variables coincide.



  - Projecting out. $c \Downarrow x$ eliminates x from a constraint c keeping the tuples that can be extended to x.

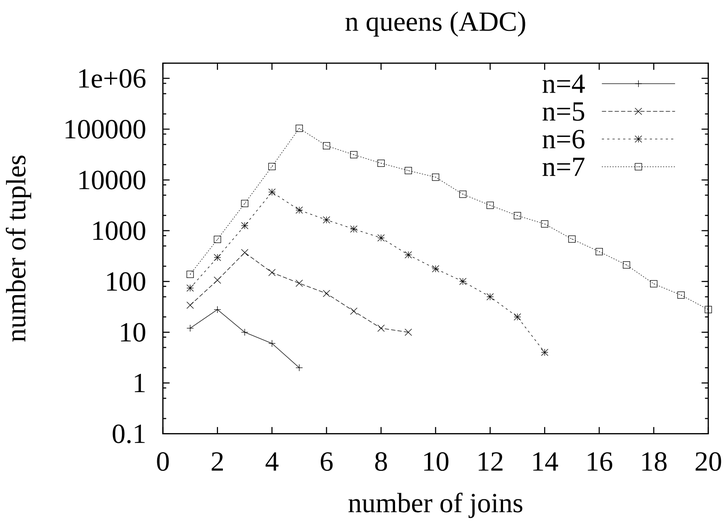


Variable Elimination

- The process of variable elimination consists in joining all the constraints linked to a variable and then projecting out the variable:

$$(\bowtie_{c \in C | x \in var(c)} c) \Downarrow x$$

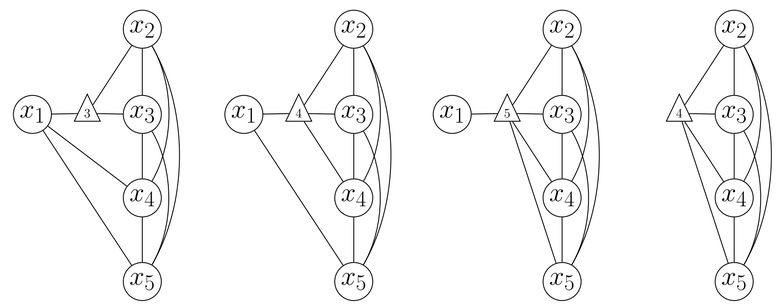
Experiments



- Out of memory after n = 7
- Peaks in every line correspond to number of tuples of the first variable elimination
- End points correspond to total number of solutions
- The vertical distance between each peak and the end point is the amount of memory used to store intermediate joins

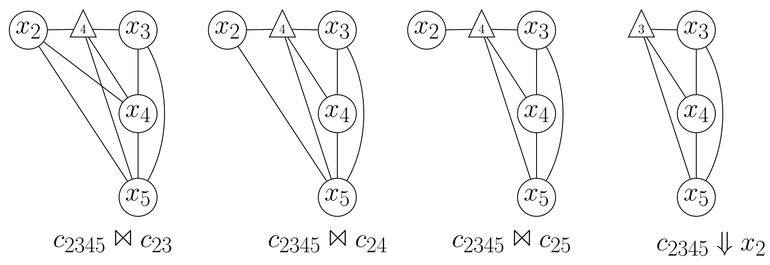
ADC on the 5-queens

Elimination of variable x_1



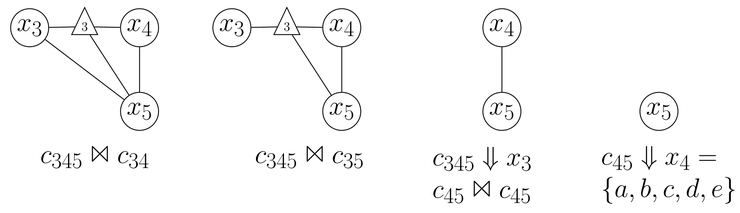
$c_{12} \bowtie c_{13}$ | $c_{123} \bowtie c_{14}$ | $|c_{1234} \bowtie c_{15}|$ = 340 tuples | $c_{12345} \Downarrow x_1$

Elimination of variable x_2



$c_{2345} \bowtie c_{23}$ | $c_{2345} \bowtie c_{24}$ | $c_{2345} \bowtie c_{25}$ | $c_{2345} \Downarrow x_2$

Elimination of variable x_3 and x_4



$c_{345} \bowtie c_{34}$ | $c_{345} \bowtie c_{35}$ | $c_{345} \Downarrow x_3$ $c_{45} \bowtie c_{45}$ | $c_{45} \Downarrow x_4 = \{a, b, c, d, e\}$

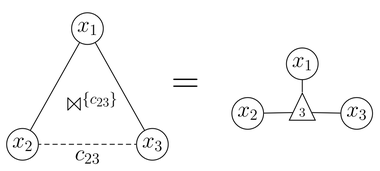
ADC with Delayed Variable Elimination and Filtering

Delayed Variable Elimination

- We are not forced to eliminate one variable in one shoot. One can start joining two constraints in one bucket, continue joining in another bucket, etc.
- As soon as one variable is mentioned by one constraint only, this variable can be eliminated.
- Variable elimination is no longer seen as an atomic operation

Join with filtering

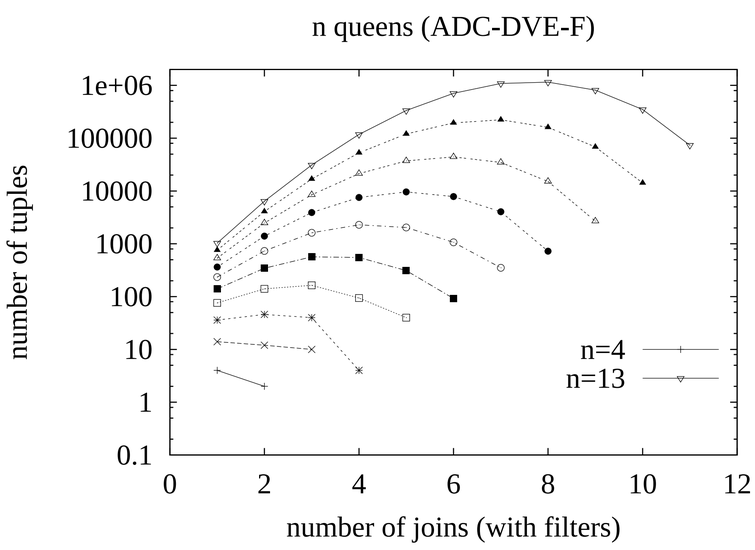
- Given two constraints c and r and a set of constraints F where $\forall_{p \in F}(var(p) \subset (var(c) \cup var(r)))$, $c \bowtie^F r$, is the join where tuples forbidden by any $p \in F$ do not appear in the resulting join.



- If $var(p) \subseteq var(c) \cup var(r)$ $\quad c \bowtie^{\{p\}} r = (c \bowtie r) \bowtie p$
- The complexity of adding a filter set $|F|$ in the join is $O(|c||r||F|)$ where $|F|$ is the number of constraints in the filter. This is equivalent to performing $|F|$ joins
- The advantage of using filters is that intermediate tuples that will be filtered out are never stored.

ADC-DVE-F Algorithm

function ADC-DVE-F(X, D, C)
1 **if** $X = \emptyset$ **then return** λ
2 **else if** $\exists x \in X$ only linked to one constraint
3 $\quad t :=$ADC-DVE-F$(X - \{x\}, D - \{D_x\}, C \cup \{c \Downarrow x\} - \{c\})$
4 $\quad a :=$ feasible extension of t w.r.t. c
5 $\quad$ **return** $t \cdot (x, a)$
6 **else**
7 $\quad x :=$selectVar(X)
8 $\quad B := \{c \in C |\ x \in var(c)\}$
9 $\quad \{p, q\} :=$selectTwo(B)
10 $\quad F := \{c \in C |\ var(c) \subset var(p) \cup var(q)\}$
11 $\quad r := p \bowtie^F q$
12 $\quad$ **if** $r = \emptyset$ **then return** NIL
13 $\quad$ **else return** ADC-DVE-F$(X, D, C \cup \{r\} - (\{p, q\} \cup F))$

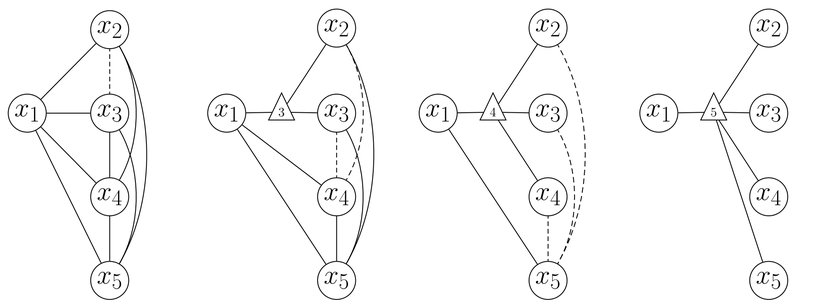


Experiments

- Out of memory after n = 13, an increase of 6 dimensions with respect to ADC.
- The vertical distance between the highest point and the end point of every line correspond to the amount of memory used to store intermediate joins, which never exceed one order of magnitude.

ADC-DVE-F on the 5-queens

Elimination of variable x_1



$c_{12} \bowtie^F c_{13}$ $F = \{c_{23}\}$ $= \{ace, adb, aeb, bda, bea, bec, cad, ceb, dac, dae, dbe, ead, ebd, eca\}$

$c_{123} \bowtie^F c_{14}$ $F = \{c_{24}, c_{34}\}$ $= \{cadb, dace, daeb, dbec, ebda, aceb, ecad, adbe, bdac, bead, beca, cebd\}$

$c_{1234} \bowtie^F c_{15}$ $F = \{c_{25}, c_{35}, c_{45}\}$ $= \{bdace, ecadb, adbec, cebda, becad, daceb, cebda, cebda, acebd, dbeca\}$

- At this point the problem is completely solved and no more eliminations are needed
- The number of stored tuples are 15, 12, 10 respectively. ADC needed 340 in the first elimination