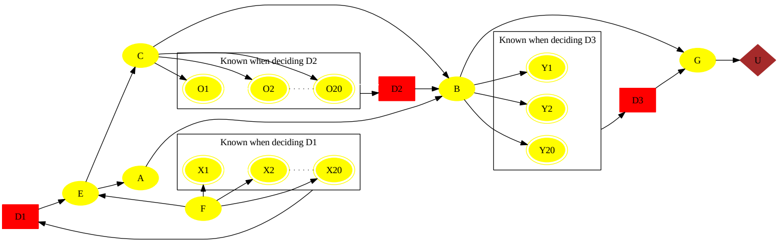
Approximate Representation of Optimal Strategies from Influence Diagrams

Finn V. Jensen

Department of Computer Science
Aalborg University

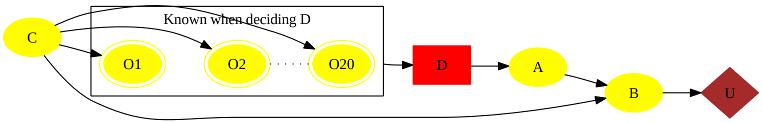
PGM, September 2008

An impossible influence diagram



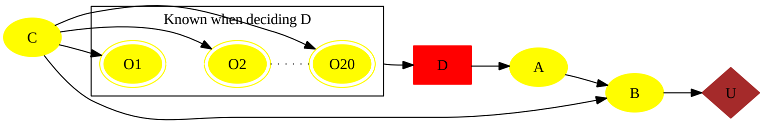
- The policies for D_3 and D_2 are intractably large

An influence diagram with one decision



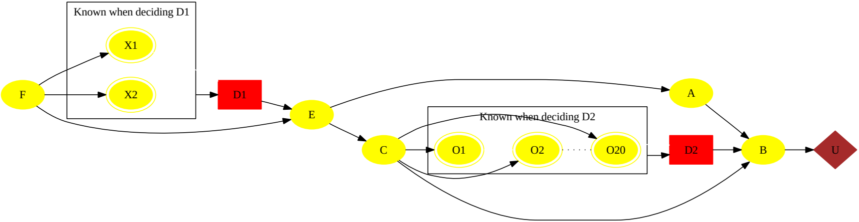
- Policy $\delta_D : O_1 \times \ldots \times O_{20} \longrightarrow D$

An influence diagram with one decision



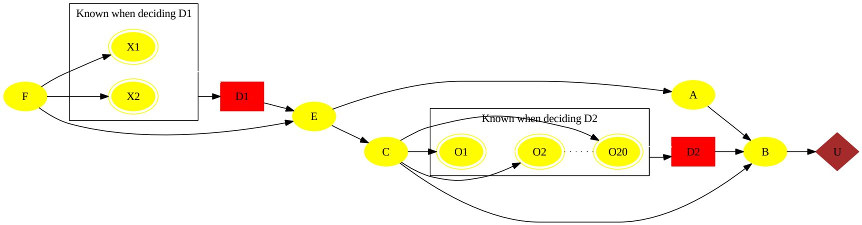
- Policy $\delta_D : O_1 \times \ldots \times O_{20} \rightarrow D$
- The ID itself is a very efficient representation of δ_D

Two decisions



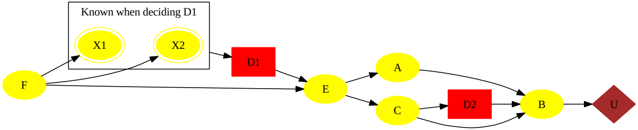
- The ID is a very efficient representation of $\delta_2(X_1, X_2, D_1, O_1, \ldots, O_{20})$
- To determine $\delta_1(X_1, X_2)$ we need $\delta_2(X_1, X_2, D_1, O_1, \ldots, O_{20})$.

Two decisions



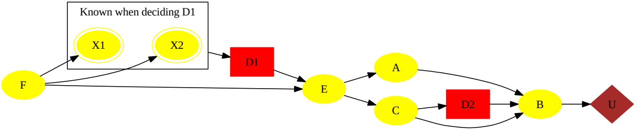
- The ID is a very efficient representation of $\delta_2(X_1, X_2, D_1, O_1, \ldots, O_{20})$
- To determine $\delta_1(X_1, X_2)$ we need $\delta_2(X_1, X_2, D_1, O_1, \ldots, O_{20})$.
- A direct representation of $\delta_2(X_1, X_2, D_1, O_1, \ldots, O_{20})$ is too costly
- The basic problem is that we need $P(O_1, \ldots, O_{20} | x_1, x_2, D_1)$ when deciding D_1

Overestimation of information



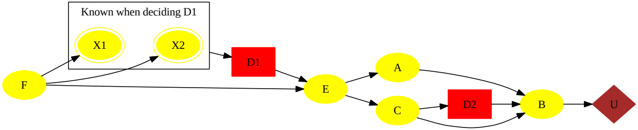
- Instead of 20 observations for C, we may assume that we know the state of C when deciding $D2$

Overestimation of information

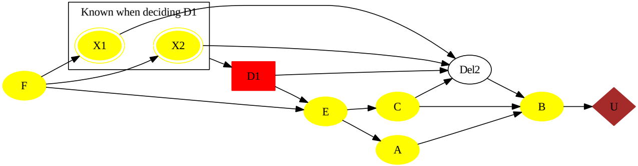


- Instead of 20 observations for C, we may assume that we know the state of C when deciding $D2$
- $\delta_2(X_1, X_2, D_1, O_1, \ldots, O_{20})$ is approximated by $\delta_2'(X_1, X_2, D_1, C)$

Overestimation of information

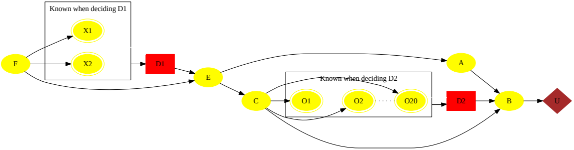


- Instead of 20 observations for C, we may assume that we know the state of C when deciding $D2$
- $\delta_2(X_1, X_2, D_1, O_1, \ldots, O_{20})$ is approximated by $\delta_2'(X_1, X_2, D_1, C)$



- And now we have an efficient (approximate) representation of $\delta_1(X_1, X_2)$

Sampling

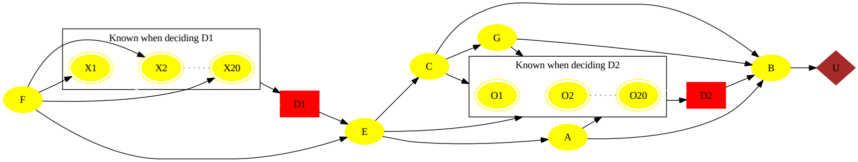


For each configuration (x_1, x_2, d_1) over X_1, X_2, D_1 **do N times**

- sample a configuration $\underline{c}$ over $(O_1, \ldots, O_{20}, x_1, x_2, d_1)$
- Solve the ID with $\underline{c}$ inserted (the result is d_2 with expected utility u)
- Construct the sample $\underline{s} = (x_1, x_2, d_1, u)$

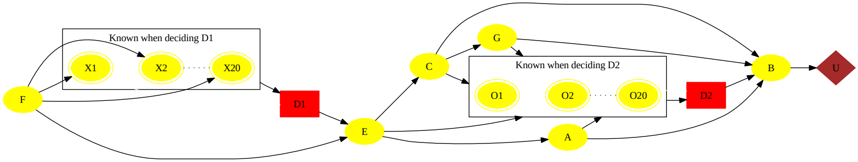
Use the N constructed samples to establish $EU(D_1|X_1, X_2)$.

A Nasty Influence Diagram



The previous techniques cannot cope.

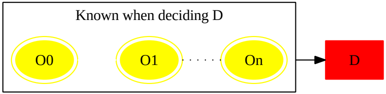
A Nasty Influence Diagram



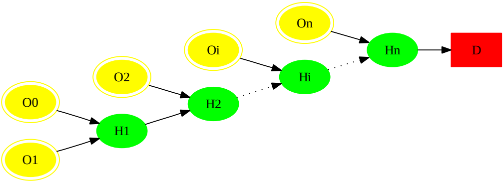
The previous techniques cannot cope.

Information abstraction: introduce latent variables connecting the information with the decision node.

An abstraction scheme: the conveyor belt

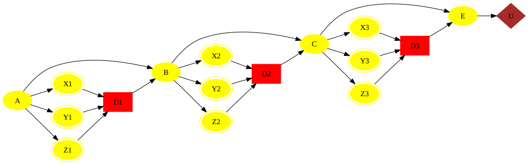


The information is abstracted down to one variable (with rather many states)



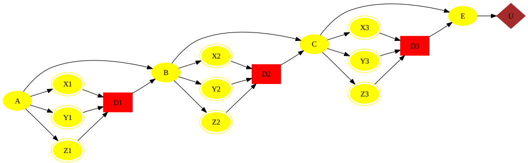
where $H_1, \ldots, H_n$ have an increasing number of states

Example: history variables

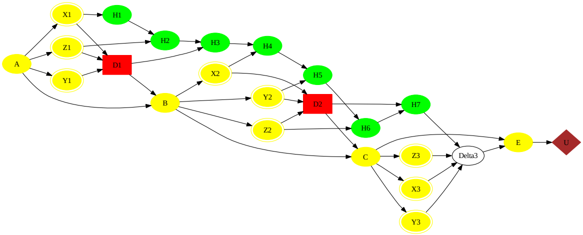


An ID with three decisions. A good representation of δ_3

Example: history variables

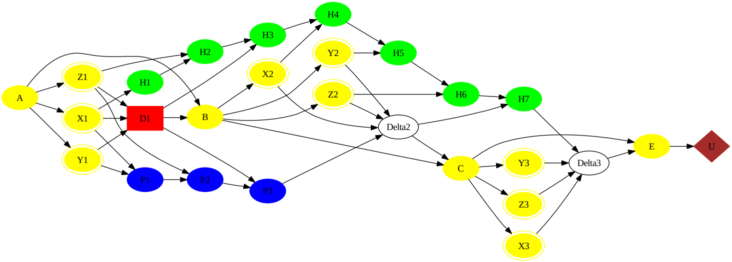


An ID with three decisions. A good representation of δ_3



An approximate representation of δ_2, where δ_3 is approximated through a belt of history variables

A representation of the first policy

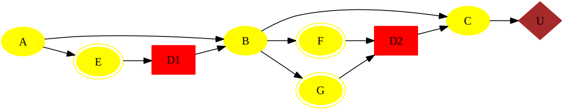


An approximate representation of δ_1 with history variables for both δ_2 and δ_3

Another abstraction scheme: conditional decomposition of the domain

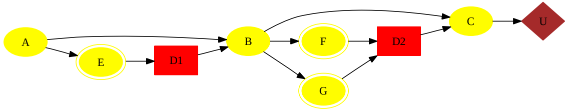
- The policy has the form: **if** $\phi(Z) = z_i$ **then** $f_i(X_i)$
  $i = 1, \ldots, m$,
  where Z and X_i are subsets of the variables of the policy domain
- $m = 2$: **if** $\phi(Z)$ **then** $f(X_1)$ **else** $g(X_0)$,
  where ϕ is a Boolean function
- ϕ may be an alert function ("Return for more fuel", "Your opponent is close to fullfilling her assignment")

Graphical representation of conditional decomposition of domains

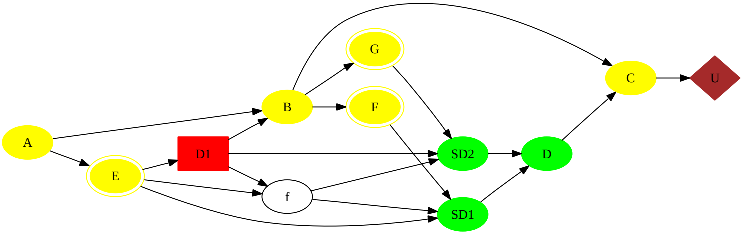


if $f(E, D1)$ **then** $SD1(E, F)$ **else** $SD2(D1, G)$,

Graphical representation of conditional decomposition of domains



if $f(E, D1)$ **then** $SD1(E, F)$ **else** $SD2(D1, G)$,



$SD1$ and $SD2$ have an extra state, na, but otherwise they hold only decisions relevant for $f = 1$ or 0, respectively.

Learning of information abstraction

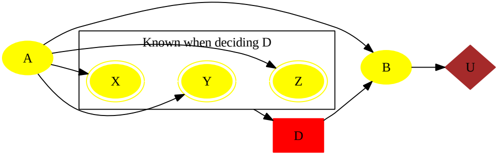
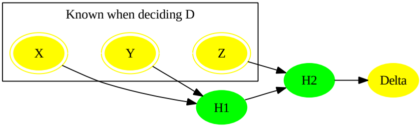
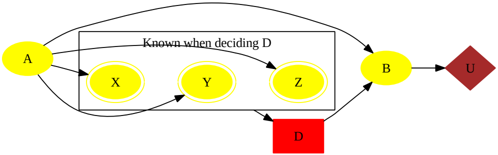


Figure: This is the ID

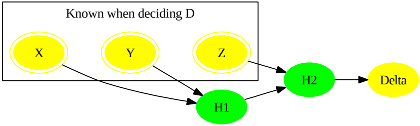


And we wish to learn the unknown parameters for this BN (knowing the number of states of $H1$ and $H2$).

Sampling and EM



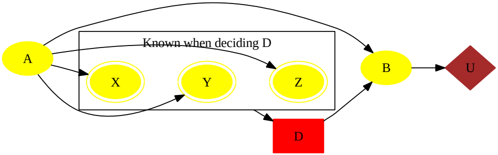
Sample the observed variables from the initial ID; for each sample, determine the optimal decision; hereby establish a database over observations and decisions.



Use the EM algorithm to learn the unknown parameters in the BN.

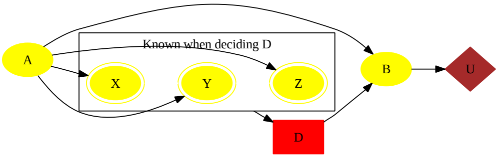
The poors man's sampling

It may be too time consuming to solve an ID for each sample

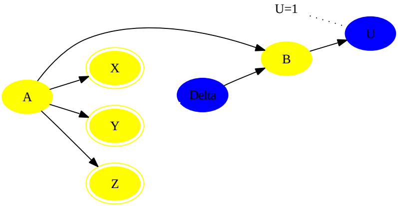


The poors man's sampling

It may be too time consuming to solve an ID for each sample



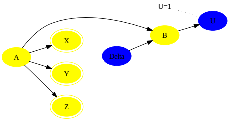
Convert the ID using Coper's trick (Cooper 1988)



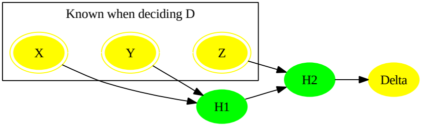
Insert $U = 1$ and sample $(X, Y, Z, Delta)$; use the EM algorithm as previously; possibly modify the CPT for *Delta* to be deterministic

Experiment, history variables

Poor man's sampling. 10.000 cases



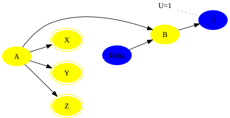
if $Z = y$ **then**
(**if** $X = y$ **then** $D = a_1$ **else** $D = a_2$)
else (**if** $Y = y$ **then** $D = a_3$ **else** $D = a_4$)



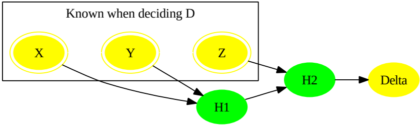
- $H1$ with three states and $H2$ with fours states

Experiment, history variables

Poor man's sampling. 10.000 cases



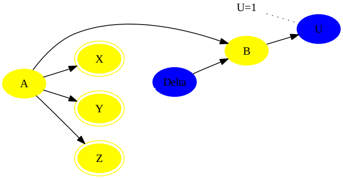
if $Z = y$ **then**
(**if** $X = y$ **then** $D = a_1$ **else** $D = a_2$)
else (**if** $Y = y$ **then** $D = a_3$ **else** $D = a_4$)

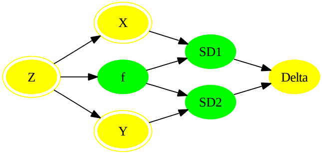


- $H1$ with three states and $H2$ with fours states
- For all eight scenarios the learned structure gave maximal probability to the correct decision.

Experiment, conditional decomposition

Poor man's sampling. 10.000 cases

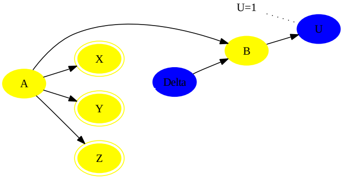


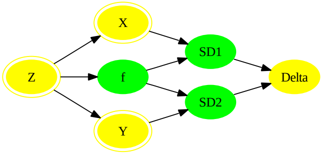


- The learned structure had all decisions correct

Experiment, conditional decomposition

Poor man's sampling. 10.000 cases





- The learned structure had all decisions correct
- Learning a policy over $f(X), SD1(Z), SD2(Y)$ resulted in a policy with 5 out of 8 decisions correct

Future work

- Experiments with real world IDs
- Library of abstraction schemes
- Alternative to (or combination with) LIMIDS (including single policy updating) - in particular for dynamic IDs.

Future work

- Experiments with real world IDs
- Library of abstraction schemes
- Alternative to (or combination with) LIMIDS (including single policy updating) - in particular for dynamic IDs.
  THANK YOU