Quantified cost functions for reasoning under uncertainty

Cédric Pralet, Thomas Schiex, Gérard Verfaillie

Decision problems, SAT and CSP

Constraint network

- a set $X = \{x_1, \ldots, x_n\}$ of variables with boolean/finite domains.
- a set $C = \{c_1, \ldots, c_n\}$ of boolean functions (clauses, constraints) with a scope (subset of X).

Query

Satisfiability: $\exists x_1 \ldots x_n (c_1 \wedge \cdots \wedge c_n)$

Optimisation: MaxSAT/CSP, Valued/Semiring CSP

Cost function network

- a set $X = \{x_1, \ldots, x_n\}$ of variables with boolean/finite domains.
- a set $C = \{c_1, \ldots, c_n\}$ of cost functions over a set E (combination operator $\otimes$) with a scope.

Query

Optimisation: $\max x_1 \ldots \max x_n (c_1 \otimes \cdots \otimes c_n)$.

Addition: an utility set E, a combination op. $\otimes$, an elimination op. max, axioms.

Example

John has three doors (A, B, C) in front of him. Behind one, there is a treasure, and behind another a gangster. John may choose one door to open. He must pay 4€ to the gangster and the treasure amounts to 10€.

The network

- Variables: ga, $tr \in \{A, B, C\}$ and $do \in \{A, B, C, nd\}$.
- Cost functions: $U_1 : do = ga(-4)$, $U_2 : do = tr(10)$.

No uncertainty: $ga = A$, $tr = C$. $do = C$ (we get 10€).

Adding uncertainty (plausibility)

Non determinism

The treasure and the gangster are not behind the same door.
Extra constraint: $P_1 : ga \neq tr$.
Pessimistic query: $\exists do \forall ga, tr, (P_1) \rightarrow (do \neq ga \wedge do = tr)$?

Quantitative (probabilistic) uncertainty

Situations are all as likely. Cost function: $P_2 : \frac{1}{6}$ (normalization).
Query: maximize expected utility

$$\max_{do} \sum_{tr,ga} ((\prod_{i=1}^{2} P_i) \times (\sum_{i=1}^{2} U_i))$$

Two types of cost functions (P, U), specific combination operators, an operator to combine P and U. Multiple elimination (quantification) operators: max (decision), $+$ (environment).

Observations

John can listen to one door and try to detect the gangster. The probability of hearing is 0.8 *if John listens to the ganster door but still* 0.4 *for a door next to this one. His friend Peter can do the same.*

Network extension

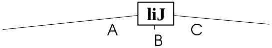
Variables: li_J, li_P, domain $\{A, B, C\}$, he_J, he_P $\{y, n\}$
Plausibilities: $P_3 : P(he_J|ga, li_J)$, $P_4 : P(he_P|ga, li_P)$ (norm)

Query: decision tree

Query: decision rules **maximizing the expected utility**, if first Peter and John listen in, and then John decides to open a door knowing what has been heared?

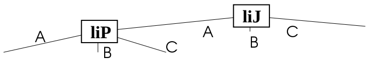
Query: decision tree

Query: decision rules **maximizing the expected utility**, if first Peter and John listen in, and then John decides to open a door knowing what has been heared?



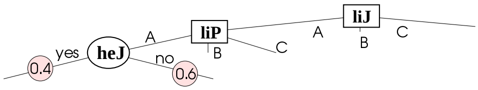
Query: decision tree

Query: decision rules **maximizing the expected utility**, if first Peter and John listen in, and then John decides to open a door knowing what has been heared?



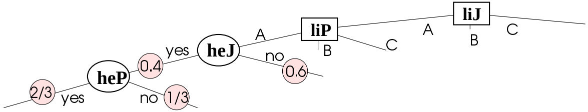
Query: decision tree

Query: decision rules **maximizing the expected utility**, if first Peter and John listen in, and then John decides to open a door knowing what has been heared?



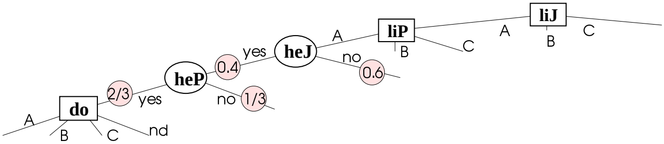
Query: decision tree

Query: decision rules **maximizing the expected utility**, if first Peter and John listen in, and then John decides to open a door knowing what has been heared?



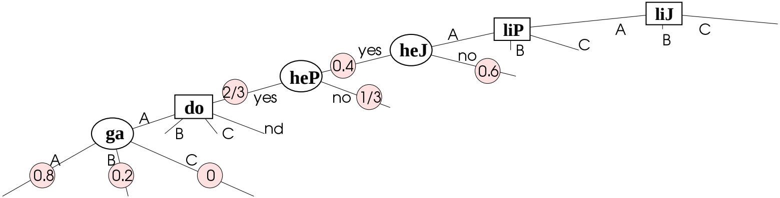
Query: decision tree

Query: decision rules **maximizing the expected utility**, if first Peter and John listen in, and then John decides to open a door knowing what has been heared?



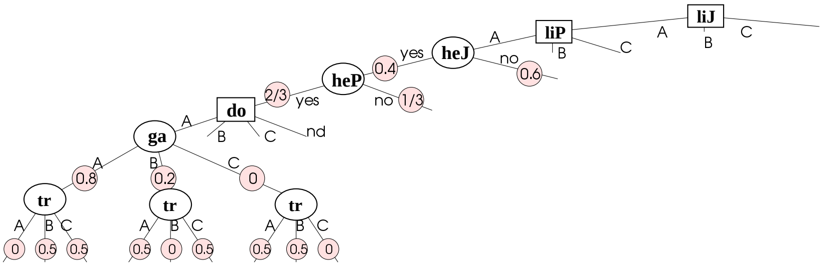
Query: decision tree

Query: decision rules **maximizing the expected utility**, if first Peter and John listen in, and then John decides to open a door knowing what has been heared?



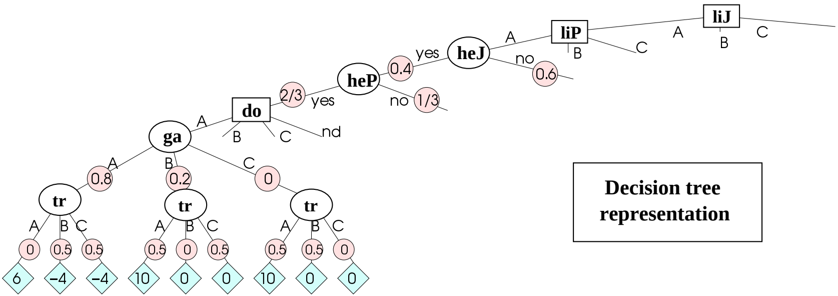
Query: decision tree

Query: decision rules **maximizing the expected utility**, if first Peter and John listen in, and then John decides to open a door knowing what has been heared?



Query: decision tree

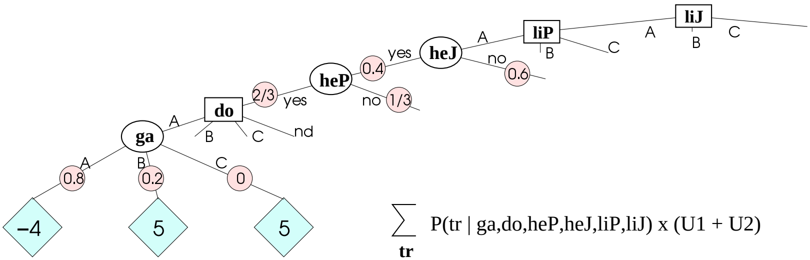
Query: decision rules **maximizing the expected utility**, if first Peter and John listen in, and then John decides to open a door knowing what has been heared?



Decision tree representation

Query: decision tree

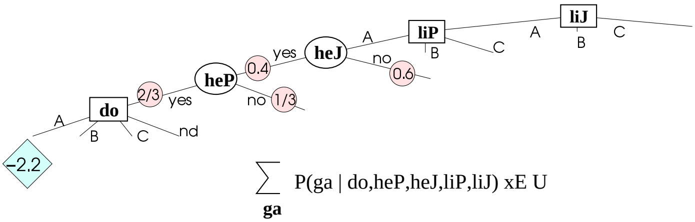
Query: decision rules **maximizing the expected utility**, if first Peter and John listen in, and then John decides to open a door knowing what has been heared?



$$\sum_{\mathbf{tr}} P(tr \mid ga,do,heP,heJ,liP,liJ) \times (U1 + U2)$$

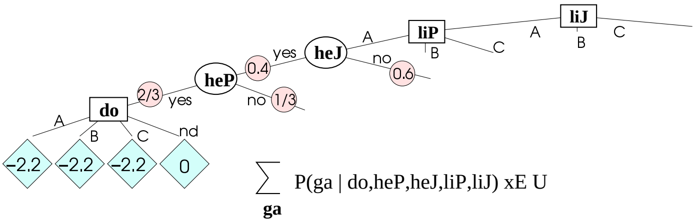
Query: decision tree

Query: decision rules **maximizing the expected utility**, if first Peter and John listen in, and then John decides to open a door knowing what has been heared?



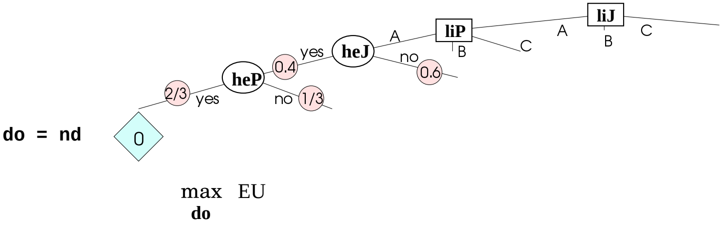
Query: decision tree

Query: decision rules **maximizing the expected utility**, if first Peter and John listen in, and then John decides to open a door knowing what has been heared?



Query: decision tree

Query: decision rules **maximizing the expected utility**, if first Peter and John listen in, and then John decides to open a door knowing what has been heared?



Query: decision tree

Query: decision rules **maximizing the expected utility**, if first Peter and John listen in, and then John decides to open a door knowing what has been heared?

$$\left[\begin{array}{c}\text{Solve the query}\\ \text{with a tree}\\ \text{exploration}\end{array}\right] \Leftrightarrow \left[\begin{array}{c}\text{Record optimal decision rules for the quantity}\\ \max_{li_J, li_P} \sum_{he_J, he_P} \max_{do} \sum_{ga, tr} \left(\prod_{i \in [1,4]} P_i\right) \times \left(\sum_{i \in [1,2]} U_i\right)\end{array}\right]$$

Other possible queries

Opposite aims

If John thinks Peter is a traitor and let him choose a door to listen in first (**pessimistic attitude concerning the other agent**):

$$\min_{li_P} \max_{li_J} \sum_{he_P, he_J} \max_{do} \sum_{ga,tr} \left(\prod_{i \in [1,4]} P_i \right) \times \left(\sum_{i \in [1,2]} U_i \right)$$

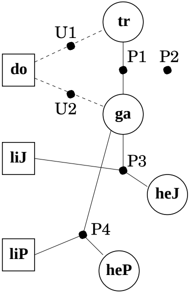
Observabilities

If Peter does not even tell John what he has heard (John does not **observe** he_P):

$$\min_{li_P} \max_{li_J} \sum_{he_J} \max_{do} \sum_{he_P} \sum_{ga,tr} \left(\prod_{i \in [1,4]} P_i \right) \times \left(\sum_{i \in [1,2]} U_i \right)$$

Adding feasibilities

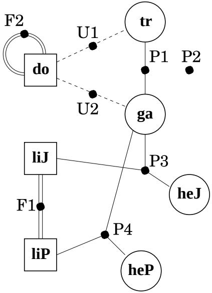
John and Peter cannot listen in to the same door, door A is locked!



Adding feasibilities

John and Peter cannot listen in to the same door, door A is locked!

Two **local feasibility functions**: $F_1 : li_J \neq li_P$, $F_2 : do \neq A$



Feasibility operators

Feasibilities is not unacceptibility. It restricts elimination domains.

- $\mathit{false} \star \alpha = \lozenge$, $\mathit{true} \star \alpha = \alpha$
- $\lozenge$ is ignored by elimination operators $(\max(\lozenge, u) = \min(\lozenge, u) = \lozenge + u = u)$

Computation of **optimal decision rules** with:

$$\min_{li_P} \max_{li_J} \sum_{he_J} \max_{do} \sum_{he_P} \sum_{ga,tr} \left(\left(\bigwedge_{i \in [1,2]} F_i \right) \star \left(\prod_{i \in [1,4]} P_i \right) \times \left(\sum_{i \in [1,2]} U_i \right) \right)$$

Many frameworks, many related algorithms

CSP
Valued CSP
Quantified CSP
Mixed and probabilistic CSP
Stochastic CSP

Finite−horizon Markov decision processes (MDP)
probabilistic/possibilistic/with kappa rankings
competely or partially (POMDP) observable
factored or not

SAT
Quantified boolean formulas
Stochastic SAT
Extended stochastic SAT

Bayesian networks
Hybrid networks
Markov Random Fields
Chain graphs

Probability computation
MPE (Most Probable Explanation)
MAP (Maximum A Posteriori hyp.)

Influence diagrams
Valuation networks

Finite−horizon
STRIPS planning,
Conformant planning,
Probabilistic planning

Three key components

1. the **algebraic structure**, defining properties satisfied by the combination and the elimination operators
2. the **network of local functions**, with normalization conditions on plausibilities and feasibilities
3. the **query** (sequence of eliminations).

An algebraic structure for plausibilities, utilities...

Extension of Halpern and Chu work to sequential decisions.

1. **plausibilities**: $(E_p, \preceq_p, \oplus_p, \otimes_p)$ commutative semiring with monotonic operators.
2. **utilities**: $(E_u, \preceq_u, \otimes_u)$ commutative monoid.
3. **expected utility**: $(E_p, E_u, \oplus_u, \otimes_{pu})$ $(E_u, \oplus_u, \otimes_{pu})$ semi-module on plausibilities:
   - $\otimes_{pu}$ distributive wrt $\oplus_p$ and $\oplus_u$.
   - $\forall p_1, p_2 \in E_p, \forall u \in E_u, p_1 \otimes_{pu} (p_2 \otimes_{pu} u) = (p_1 \otimes_p p_2) \otimes_{pu} u$.
   - $\forall u \in E_u, (0_p \otimes_{pu} u = 0_u) \wedge (1_p \otimes_{pu} u = u)$.

$\sum_j (p_j \times u_j)$ becomes $\bigoplus_{u\,j} (p_j \otimes_{pu} u_j)$

Examples

	E_p	$\oplus_p$	$\otimes_p$	E_u	$\otimes_u$	$\oplus_u$	$\otimes_{pu}$
1	$\mathbb{R}^+$	$+$	$\times$	$\mathbb{R} \cup \{-\infty\}$	$+$	$+$	$\times$
2	$\mathbb{R}^+$	$+$	$\times$	$\mathbb{R}^+$	$\times$	$+$	$\times$
3	$[0,1]$	max	min	$[0,1]$	min	max	min
4	$[0,1]$	max	min	$[0,1]$	min	min	$\max(1-p, u)$
5	$\mathbb{N} \cup \{\infty\}$	min	$+$	$\mathbb{N} \cup \{\infty\}$	$+$	min	$+$
6	$\{t, f\}$	$\vee$	$\wedge$	$\{t, f\}$	$\wedge$	$\vee$	$\wedge$
7	$\{t, f\}$	$\vee$	$\wedge$	$\{t, f\}$	$\wedge$	$\wedge$	$\rightarrow$

(1) probabilistic expected utility, (2) probabilistic expected satisfaction, (3/4) optimistic/pessimistic possibilistic utilities, (5) κ-rankings with positive utilities (6/7) optimistic/pessimistic expected boolean satisfaction

Locality of functions

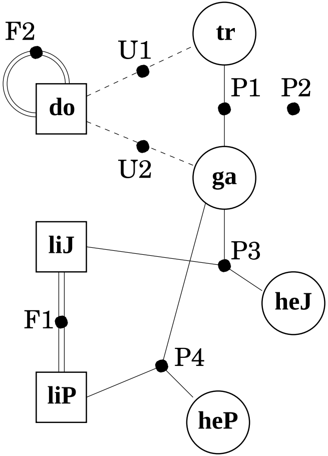
Definition: a **PFU network** is a tuple (V, G, P, F, U) where:

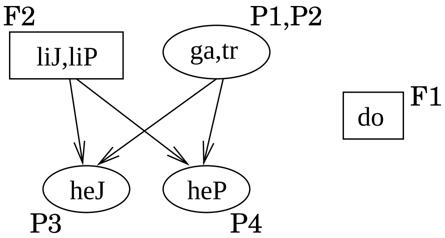
- V: finite **set of variables**, partitionned between V_D the set of **decision variables**, and V_E, the set of **environment variables**. Each is partitioned in clusters.
- G a **a cluster DAG representing normalization/independance.**
- $P = \{P_1, P_2, \ldots\}$ a finite set of **local plausibility functions**
- $F = \{F_1, F_2, \ldots\}$ a finite set of **local feasibility functions**
- $U = \{U_1, U_2, \ldots\}$ a finite set of **local utility functions**

Example

PFU network of the treasure problem:

[b]





Query Q on a PFU network $\mathcal{N}$

A pair **$(\mathcal{N}, Sov)$** where:

- $\mathcal{N}$: PFU network
- Sov is a **sequence of operator/variable(s) pairs**. Operators are min, max, or $\oplus_u$ and act as **quantifiers**.

Ex.: $(\mathcal{N}, \max_{x_1,x_2} \min_{x_3} \oplus_{u\, x_4,x_5} \max_{x_6})$

Meaning:

- the order specifies the **order in which decisions are made** and **environment variables are observed**
- min and max operators applied on decision variables specify if **an optimistic or a pessimistic** attitude is adopted for decisions.

Correct queries

1. **Adequation between an eliminated variable and its elimination operator** (decision variable $\longrightarrow$ eliminated with min or max; environment variables $\longrightarrow$ eliminated with $\oplus_u$).
   Ex: if $Sov = \ldots \sum_{do} \ldots$, *the query is not correct*
2. **Respect of causality**: conditions (imposed by the DAG) on the order of elimination
   Ex: if $Sov = \ldots \sum_{he_J} \ldots \max_{li_J} \ldots$, *the query is not correct.*

Value of a query

Answer to a query Q

The answer $Ans(Q)$ to a correct query Q can be defined based on **decision trees**.

- Advantage: has a **clear semantic foundation**.
- Drawback: **weights associated with some edges in the tree** must be computed, which can take an exponential time.

Property

The definition is **equivalent** to the following operational one:

$$Ans(Q) = Sov\left(\left(\bigwedge_{F_i \in F} F_i\right) \star \left(\bigotimes_{p\, P_i \in P} P_i\right) \otimes_{pu} \left(\bigotimes_{u\, U_i \in U} U_i\right)\right)$$

What is covered

Theorem: the PFU framework can be used to model and solve queries asked on:

CSP
Valued CSP
Quantified CSP
Mixed and probabilistic CSP
Stochastic CSP

What is covered

Theorem: the PFU framework can be used to model and solve queries asked on:

CSP
Valued CSP
Quantified CSP
Mixed and probabilistic CSP
Stochastic CSP

SAT
Quantified boolean formulas
Stochastic SAT
Extended stochastic SAT

What is covered

Theorem: the PFU framework can be used to model and solve queries asked on:

CSP
Valued CSP
Quantified CSP
Mixed and probabilistic CSP
Stochastic CSP

SAT
Quantified boolean formulas
Stochastic SAT
Extended stochastic SAT

Bayesian networks
Hybrid networks
Markov Random Fields
Chain graphs

Probability computation
MPE (Most Probable Explanation)
MAP (Maximum A Posteriori hyp.)

What is covered

Theorem: the PFU framework can be used to model and solve queries asked on:

CSP
Valued CSP
Quantified CSP
Mixed and probabilistic CSP
Stochastic CSP

SAT
Quantified boolean formulas
Stochastic SAT
Extended stochastic SAT

Bayesian networks
Hybrid networks
Markov Random Fields
Chain graphs

Probability computation
MPE (Most Probable Explanation)
MAP (Maximum A Posteriori hyp.)

Influence diagrams
Valuation networks

What is covered

Theorem: the PFU framework can be used to model and solve queries asked on:

CSP
Valued CSP
Quantified CSP
Mixed and probabilistic CSP
Stochastic CSP

Finite−horizon Markov decision processes (MDP)
probabilistic/possibilistic/with kappa rankings
competely or partially (POMDP) observable
factored or not

SAT
Quantified boolean formulas
Stochastic SAT
Extended stochastic SAT

Bayesian networks
Hybrid networks
Markov Random Fields
Chain graphs

Probability computation
MPE (Most Probable Explanation)
MAP (Maximum A Posteriori hyp.)

Influence diagrams
Valuation networks

What is covered

Theorem: the PFU framework can be used to model and solve queries asked on:

CSP
Valued CSP
Quantified CSP
Mixed and probabilistic CSP
Stochastic CSP

Finite−horizon Markov decision processes (MDP)
probabilistic/possibilistic/with kappa rankings
competely or partially (POMDP) observable
factored or not

SAT
Quantified boolean formulas
Stochastic SAT
Extended stochastic SAT

Bayesian networks
Hybrid networks
Markov Random Fields
Chain graphs

Probability computation
MPE (Most Probable Explanation)
MAP (Maximum A Posteriori hyp.)

Influence diagrams
Valuation networks

Finite−horizon
STRIPS planning,
Conformant planning,
Probabilistic planning

Not covered

- non distributional uncertainty (Dempster-Shafer)
- really qualitative utility (CP-nets) or partially ordered utilities (Queries).

Algorithms

The operational definition gives us a **generic tree search algorithm** (and PSPACE membership).

Variable elimination: additional axioms needed (sufficient).

1. (Ax1) semiring:
   $((E_p, \oplus_p, \otimes_p) = (E_u, \oplus_u, \otimes_u)) \wedge (\otimes_{pu} = \otimes_u) \wedge (\preceq_p = \preceq_u)$
   (2,3,5,6)
2. (Ax2) vector space-like: $\oplus_u = \otimes_u$ on E_u (1,4,7).

Todo...

1. combine existing algorithms: branch and bound, local consistency (csp, hard information), cost function consistency (soft constraints), tree decomposition based (recursive conditioning).
2. variable elimination/CTE: constrained treewidth and quantified cost functions: connection between the query and the network structure. Do query optimization to lower contrained treewidth.
3. islands of tractability/NP-completeness and beyond (multi-chotomy ?) in quantified cost-functions.