Local Computation Schemes with Partially Ordered Preferences

Hélène Fargier [1] and **Nic Wilson**[2]

Abstract. Many computational problems linked to reasoning under uncertainty can be expressed in terms of computing the marginal(s) of the combination of a collection of (local) valuation functions. Shenoy and Shafer showed how such a computation can be performed using only local computations. A major strength of this work is that it is based on an algebraic description: what is proved is the correctness of the local computation algorithm under a few axioms on the algebraic structure. The instantiations of the framework in practice make use of totally ordered scales. The present paper focuses on problems of optimization over partially ordered scales, including problems that do not rely on a semilattice, and examines how they can be cast in the Shafer-Shenoy framework so as to satisfy the axioms for local computation and thus benefit from local computation algorithms. It also provides many examples of preference relations, thus showing that each of the algebraic structures explored here has its own interest.

1 Introduction

Many computational problems linked to reasoning under uncertainty can be expressed in terms of computing the marginal(s) of the combination of a collection of (local) valuation functions. Shenoy and Shafer [15, 14] showed how such a computation can be performed using only local computations, operating using successive variable eliminations, based on a join tree representation (see also, in particular, [8]). A major strength of this work, is that it is based on an algebraic description: what is proved is the correctness of the local computation algorithm under a few axioms on the algebraic structure. Hence, the same algorithm of local computation may be used for computing the projection on a given variable of a joint probability distribution described by a Bayesian net, for making the fusion of several basic probability assignments with Dempster's rule of combination, or for computing the degree of consistency of a possibilistic knowledge base. The scope of Shenoy and Shafer's framework also encompasses several optimization problems, like the MAX CSP problem [5] or the VCSP problem [13].

All the above instantiations of the Shenoy-Shafer framework actually deal with totally ordered scales of scoring. The question is: can we use local computation in optimization problems over a partially ordered scale, and how? Indeed, AI has witnessed the emergence of frameworks based on such partial orders. Let us for instance cite semiring constraint satisfaction problems [1], order of magnitude reasoning [17], or belief revision [2, 6]; other examples are obviously provided by multicriteria decision making. The PFU framework [10] also makes use of variable elimination algorithms very close to local computation.

The main purpose of this paper is to show whether and how such partially ordered problems can be cast in Shenoy and Shafer's framework, so as to provide them with local computation algorithms. In addition, it helps comparing these problems according to their algebraic properties. We also give examples of preference relations in order to show that the algebraic structures explored here are of interest.

2 Axioms for local computation

We recall here some basics of the Shenoy-Shafer framework [15, 14, 8]. Consider a finite set $X = \{x_1, \ldots, x_n\}$ of variables, each x_i ranging over a finite state space (or "domain") D_i. D_S will denote the Cartesian product of the variables in S. For the sake of simplicity, considering a function over D_S we shall use the notation $f(d)$ for any d assigning a superset of S, so that $f(d) = f(proj(d, S))$, where $proj(d, S)$ is the projection of d to subset S. We also adopt the convention that the state space for the empty set $\emptyset$ consists of a single object $\blacklozenge$, i.e., $D_\emptyset = \{\blacklozenge\}$.

Given a set $S \subseteq X$ of variables, we assume a set V_S. The elements of V_S are called *valuations* and S is the *scope* of each $\sigma \in V_S$—let us denote it $scope(\sigma) = S$, which, intuitively, is the set of variables that σ gives information about. $\mathcal{V} = \bigcup_{S \subseteq X} V_S$ the set of valuations. Valuations are primitives in the Shenoy-Shafer framework and as such require no definition. They are simply entities that can be *combined* and *marginalized*:

- The combination of two valuations σ and τ, denoted $\sigma \boxtimes \tau$ is a valuation whose scope is $scope(\sigma) \cup scope(\tau)$.
- The marginalisation of one valuation σ to a set of variable $T \subseteq scope(\sigma)$ is a valuation whose scope is T. Let us denote it $\sigma^{\downarrow T}$.

$(\mathcal{V}, \boxtimes, \downarrow)$ is called a valuation algebra. A valuation network (VN) is a finite set $\Sigma = \{\tau_1, \ldots, \tau_m\} \subseteq \mathcal{V}$. The marginal of Σ over a (possibly empty) subset T of X is:

$$(\boxtimes\Sigma)^{\downarrow T} = (\tau_1 \boxtimes \ldots \boxtimes \tau_m)^{\downarrow T}$$

Bayesian nets are well known instances of VNs, where valuations are conditional probability distributions, combined by the product and marginalized using summation. These instances, among many others, satisfy the well known axioms for local computation:

Axiom A1: If $S \subseteq T \subseteq scope(\sigma)$, then $((\sigma)^{\downarrow T})^{\downarrow S} = \sigma^{\downarrow S}$

Axiom A2: $\boxtimes$ is associative and commutative

[1] IRIT-CNRS, 118 Route de Narbonne 31062 Toulouse Cedex, France. Helene.Fargier@irit.fr

[2] Cork Constraint Computation Centre, Department of Computer Science, University College Cork Cork, Ireland. n.wilson4c.ucc.ie

Axiom A3 (distributivity of $\downarrow$ over $\boxtimes$):
If $scope(\sigma) \subseteq T \subseteq scope(\sigma) \cup scope(\tau)$, then $(\sigma \boxtimes \tau)^{\downarrow T} = \sigma \boxtimes \tau^{\downarrow T \cap scope(\tau)}$

It is then shown that if the valuation algebra satisfies Axioms A1, A2, A3, then for any valuation network over X, the marginal of the network over set of variables Y can be computed by successive variable eliminations. More technically, given a VN network Σ, the basic procedure can be defined as:

$$Elim_T(\Sigma) = \Sigma_{\neg T} \cup \{(\boxtimes \Sigma_T)^{\downarrow X \setminus T}\}$$

where $\Sigma_{\neg T} = \{\sigma \in \Sigma, scope(\sigma) \cap T = \emptyset\}$ is the subset of valuations in Σ that do not bear on any variable in T and $\Sigma_T = \Sigma \setminus \Sigma_{\neg T}$ is the subset of valuations that do. If Axioms A1, A2, A3, hold, then it can be proved that:

$$(\boxtimes \Sigma)^{\downarrow X \setminus T} = \boxtimes Elim_T(\Sigma)$$

So, we can go from Σ to a new set of valuations, not bearing on T, by combining all the valuations that bear on T, computing its marginal over $X \setminus T$ and adding it to the set of valuations that do not bear on T. Applying the principle iteratively w.r.t. a sequence of variables $Y = (x_{p1} x_{p2} \dots x_{pk})$, the algorithm computes the marginal of the VN over $X \setminus Y$:

$$(\boxtimes \Sigma)^{\downarrow X \setminus \{x_{p1} \dots x_{pk}\}} = \boxtimes Elim_{\{x_{pk}\}}(Elim_{\{x_{p_{k-1}}\}}(\dots Elim_{\{x_{p1}\}}(\Sigma)\dots))$$

Axioms for local computation are sufficient conditions for the correctness of the sequential elimination procedure. They also ensure the correctness of algorithms of message passing in a join tree decomposition of the VN. What is important for the purpose of the present paper, is that it is granted that, when axioms A1, A2 and A3 hold, such algorithms are available.

Practical specializations of local computation focus on the case when optimization is made w.r.t. a total order. We will show that it applies to many other situations, which involve only partially ordered scales.

3 Optimization in utility structures

The problems we are interested in use a scoring scale L to evaluate alternatives.[3] $\preceq$ will denote the preference relation over scores. We use notation $\prec$ for the associated strict preference ($a \prec b$ iff $a \preceq b$ and not($b \preceq a$)) and $\sim$ for the corresponding indifference relation: $a \sim b$ holds if and only if both $a \preceq b$ and $b \preceq a$ hold. We adopt the convention that $a \preceq b$ means that the score a is as least as good as the score b, i.e., we are oriented toward minimization. Each alternative d receives a collection $\langle c_1(d), ..., c_m(d)\rangle$ of scores; the c_i can be criteria, soft constraints, formulas in a penalty logic, etc. The global score of d is the aggregation of all the $c_i(d)$ according $\otimes$. Most of these problems can be cast as problems of optimization in utility networks, as defined in [10].

Definition 1 *A utility structure is a triplet $\langle L, \preceq, \otimes\rangle$ which forms an ordered commutative monoid.*

That is to say, $\preceq$ is a reflexive, anti-symmetric and transitive relation over L (since $\preceq$ is antisymmetric, $a \sim b$ iff $a = b$) and L is equipped with an internal operator $\otimes$ which is associative, commutative operation and monotonic with respect to $\preceq$ ($a \preceq b \Rightarrow a \otimes c \preceq b \otimes c$.)

Definition 2 *Given a utility structure $\langle L, \preceq, \otimes\rangle$ and a set X of discrete variables:*

- *A local function is a function from the domain D_S of a set of variable $S \subseteq X$ into L.*
- *A utility network $\mathcal{C}$ is a set of local functions.*

Definition 3 *Given a utility network $\mathcal{C}$ on $\langle L, \preceq, \otimes\rangle$, the global score of d is $score_{\mathcal{C}}(d) = \bigotimes_{c_i \in \mathcal{C}} c_i(d)$. We shall also write $Scores(\mathcal{C}) = \{score_{\mathcal{C}}(d) : d \in D_X\}$.*

3.1 Five problems of optimization

3.1.1 *MAX CSP and VCSP*

In MAX CSPs (and respectively, VCSPs) [5, 13], the aim is to find a d that minimizes the number of violated constraints (resp. a combination, generally the sum, of the weight of the violated constraints). We shall use $L = \mathbb{N} \cup \{+\infty\}$. $\otimes$ is the addition of numbers and $\preceq = \leq$. In these examples, L is totally ordered, $\otimes$ admits a neutral element which is the best score is the scale (0).

3.1.2 *Cumulative prospect theory (CPT)*

CPT is an old attempt to take into account the positive and negative aspects of decision making [16]. In CPT, each c_i evaluates each possible decision d with a score that may be either a positive real (i is in favor of d) or a negative real (i is against decision d). The global score of d is the sum of the positive and negative scores and should be maximized. Here, $L = \mathbb{R} \cup \{-\infty\}$, $\otimes$ is the addition of numbers and $\preceq$ follows the classical comparison of reals: $a \preceq b$ iff $a \geq b$, since our convention minimizes while CPT maximizes. Notice that L is totally ordered, that $\otimes$ admits an annihilator ($-\infty$) and a neutral element (0). The main difference with MAX CSP is that the neutral element does not need to be the optimal element in L.

3.1.3 *Bi-attribute Pareto decision making*

In many multicriteria applications one has to simultaneously optimize several non commensurable quantities, like cost, time, security, etc. In the problem of bi-scaled shortest path for instance [7], each edge in a graph is labelled by a cost and a duration. The cost (resp. the duration) of a path is the sum of the costs (resp. durations) of its edges. For these problems, we can use $L = (\mathbb{N} \cup \{+\infty\}) \times (\mathbb{N} \cup \{+\infty\})$, the combination being the pointwise addition $(a, b) \otimes (a', b') = (a+a', b+b')$. Pairs are compared according to Pareto optimality, i.e. $(a, b) \preceq (a, b)$ iff $a \leq a'$ and $b \leq b'$. $\preceq$ is a partial order, e.g., $(3, 2)$ and $(2, 3)$ are incomparable.

3.1.4 *Order of Magnitude Reasoning*

In the system of order of magnitude reasoning described in [17], the elements of L are pairs $\langle s, r\rangle$ where $s \in \{+, -, \pm\}$, and $r \in \mathbb{Z} \cup \{\infty\}$. The system is interpreted in terms of "order of magnitude" values of utility, so, for example, $\langle -, r\rangle$ represents something which is negative and has order of magnitude K^r (for a large number K). Element $\langle \pm, r\rangle$ arises from the sum of $\langle +, r\rangle$ and $\langle -, r\rangle$. $\langle \pm, r\rangle$ can

[3] In the present paper, we often use the terms "scoring scale" for L and "score" for elements of L because its elements are not necessarily representing utilities (the framework encompasses also reasoning problems) and we are not maximizing. We also avoid the terms (e)valuation scale or (e)valuations in order because of potential confusion with the notion of valuation used in Valuation Networks; in VNs, a valuation is not an element of a scale, but, rather a function taking its values in a scoring scale.

be thought of as the interval between $\langle -, r\rangle$ and $\langle +, r\rangle$, since the sum of a positive quantity of order K^r and a negative quantity of order K^r can be either positive or negative and of any order less than or equal to r. Let $A_{oom} = \{\langle \pm, -\infty\rangle\} \cup \{\langle s, r\rangle\{s \in \{+, -, \pm\}, r \in \mathbb{Z} \cup \{+\infty\}\}$.

The interpretation leads to define $\otimes$ by: $\langle s, r\rangle \otimes \langle s', r'\rangle = \langle s, r\rangle$ if $r > r'$; it's equal to $\langle s', r'\rangle$ if $r < r'$; and is equal to $\langle s \vee s', r\rangle$ if $r = r'$, where $\vee$ is given by: $+ \vee + = +$ and $- \vee - = -$, and otherwise, $s \vee s' = \pm$. Operation $\otimes$ is commutative and associative with neutral element $\langle \pm, \infty\rangle$. $\preceq$ is defined by the following instances:[4] (i) for all r and s, $\langle +, r\rangle \preceq \langle -, s\rangle$; (ii) for all $s \in \{+, -, \pm\}$, and all r, r' with $r \geq r'$: $\langle +, r\rangle \preceq \langle s, r'\rangle \preceq \langle -, r\rangle$. The relation $\preceq$ is a partial order. However, there are incomparable elements, e.g. $\langle \pm, r\rangle$ and $\langle \pm, s\rangle$ when $r \neq s$.

3.1.5 *Tolerant Pareto*

The problem with a Pareto-based comparison is that the preference provided is often not decisive enough. For instance the two pairs $a = (a_{cost}, a_{time})$ and $b = (b_{cost}, b_{time})$ are incomparable as soon as $a_{cost} < b_{cost}$ and $b_{time} < a_{time}$, and this even if the difference between a_{cost} and b_{cost} is much greater than difference between b_{time} and a_{time}.

Consider our time/cost pair. The idea is to use indifference thresholds, say α_{cost} for the first dimension, and α_{time} for the second one. If $a_{cost} + \alpha_{cost} < b_{cost}$, we shall say that the cost dimension has a strong preference for a over b, and opposes a veto to the opposite preference. Then we decide that an alternative is better than the other iff it Pareto dominates, but with respect to the thresholds of tolerance. Formally decide:

$$a \prec b \text{ iff either } \begin{cases} b_{cost} - a_{cost} > \alpha_{cost} \text{ and} \\ b_{time} - a_{time} \geq -\alpha_{time}\text{; or} \\ b_{time} - a_{time} > \alpha_{time} \text{ and} \\ b_{cost} - a_{cost} \geq -\alpha_{cost} \end{cases}$$

So, when one dimension strongly prefers alternative a while the other does not oppose a veto we do not get an incomparability, like in the classical Pareto case, but a strict preference $a \prec b$. This decision rule is related to the Electre method (see e.g. [12]). It is not complete nor transitive: it may happen that $a \prec b$ and $b \prec c$ while a and c are not comparable (e.g. because the time dimension that does not oppose a veto to $a \prec b$ nor to $b \prec c$ is a vetoer for $a \prec c$). Nevertheless, $\prec$ is acyclic.

This example cannot be cast as a utility network stricto sensus, but its closure by transitivity can be, using pointwise addition as the combination. Let $\prec^*$ be the transitive closure of $\prec$. It can be shown that $a \prec^* b$ holds if and only if either (i) $b_{cost} - a_{cost} > 0$ and $b_{time} - a_{time} > 0$, or (ii) there exists $k \in \{1, 2, \ldots\}$ such that either (a) $b_{cost} - a_{cost} > k\alpha_{cost}$ and $b_{time} - a_{time} \geq -k\alpha_{time}$ or (b) $b_{time} - a_{time} > k\alpha_{time}$ and $b_{cost} - a_{cost}c \geq -k\alpha_{cost}$.

In this rule, the thresholds are considered as elementary units of strong preference. So, a is better than b when, going from b to a, the enhancement on one dimension (e.g. the cost dimension) is greater than the degradation in the other dimension, this enhancement (resp. degradation) being evaluated on a scale whose unit is α_{cost} (resp. α_{time}).

[4] This definition is slightly stronger than the original one, which doesn't allow $\langle +, r\rangle \preceq \langle \pm, r\rangle \preceq \langle -, r\rangle$; either order can be justified, but our choice is more discriminating.

3.2 A Further classification of the examples

Let us go back to the algebraic framework, $\langle L, \preceq \otimes\rangle$. Remark that in all the problems, the worst score annihilates $\otimes$. Indeed, for any ordered monoid, we can suppose without loss of generality that L contains a unique maximal (worst) element $\top$ and a unique minimal (best) element $\bot$, and that $\top$ annihilates $\otimes$.

- If $\bot$ is the neutral element, then it holds that $a \preceq a \otimes b$. $\langle L, \preceq, \otimes\rangle$ is then said to be negative.
- If there exists an associative and commutative operator $\oplus$ such that $a \preceq b \iff a \oplus b = a$, then we say that $\oplus$ represents $\preceq$. It is well known that such a $\oplus$ exists iff $\langle L, \preceq\rangle$ forms a meet semilattice.
- If $\preceq$ is a total order this operator necessarily exists ($\oplus = \min$).
- If $\forall a, b, \forall c \neq \bot, \top, a \prec b \Rightarrow a \otimes c \prec b \otimes c$ then $\langle L, \preceq, \otimes\rangle$ is said to be strictly monotonic.

Negative structures are well known in flexible constraint satisfaction. In semiring CSPs [1], the first two properties are assumed (semiring CSPs are essentially utility networks where $\langle L, \otimes, \oplus\rangle$ is a negative commutative semiring). If the completeness of $\preceq$ is moreover assumed, the network is a soft CSP in the sense of [3]

Let us go back to our examples. Max CSPs and VCSPs are instances of soft CSPs (and thus of semiring CSPs). Pure Pareto Cost/Time problems are semiring CSPs (just set $(a, b) \oplus (c, d) = (min(a, c), min(b, d))$). Both are based on a negative structure.

But there are utility networks that are not soft CSPs nor semiring CSPs, e.g.:

- In CPT and OOM, $\bot$ is not the neutral element
- In Tolerant Pareto problems, there exists no operator $\oplus$ encoding $\preceq$.

3.3 Optimality and Complexity

When the scale is totally ordered, as for CPT or VCSP, the usual optimization request is to compute the minimal value for $score_{\mathcal{C}}(d)$ (generally, together with the d leading to this score). When $\preceq$ is partial, there may be several optimal scores that are pairwise incomparable.

Definition 4 *$d \in D_X$ is an optimal solution for (C) if there is no d' in D_X such that $score_{\mathcal{C}}(d') \prec score_{\mathcal{C}}(d)$.*
a is an optimal score for $\mathcal{C}$ if $a = score_{\mathcal{C}}(d)$ for some optimal solution d.

For partial order $\preceq$ and any $A \subseteq L$, let us denote $Kernel_{\preceq}(A)$ (the kernel of A) as the set of $\preceq$-minimal elements of A, i.e., the set of elements $a \in A$ such that there exists no $b \in A$ with $b \prec a$. It is easy to see that the set of optimal scores is the Kernel of $Scores(\mathcal{C})$ w.r.t. $\preceq$:

Proposition 1 *$a \in Kernel_{\preceq}(Scores(\mathcal{C}))$ iff a is an optimal score for $\mathcal{C}$.*

So, if $\preceq$ is a total order, $Kernel_{\preceq}(Scores(\mathcal{C}))$ is the singleton set containing the optimal score for $\mathcal{C}$.

When compared to soft CSPs (resp. semiring CSPs), our utility networks relax the assumption of $\preceq$ being a total order (resp. a semilattice) as well as the requirement about the neutral element. However, this does not increase the complexity of the problem. Let

$\mathcal{L} = \langle L, \preceq, \otimes \rangle$ be a utility structure. We consider the following two problems:

[OPT$_\mathcal{L}$]: *Given a network $\mathcal{C}$ built on utility structure $\mathcal{L}$ and $a \in L$, does there exist an assignment d such that $score_\mathcal{C}(d) \prec a$.*
[FULLOPT$_\mathcal{L}$]: *Given a network $\mathcal{C}$ built on utility structure $\mathcal{L}$, and given $H \subseteq L$, does there exist an assignment d such that $\exists a \in H, score_\mathcal{C}(d) \prec a$.*

Proposition 2 *Let $\mathcal{L} = \langle L, \preceq, \otimes \rangle$ be a utility structure. Suppose that testing $a \preceq b$ is polynomial and that computing the combination of a multiset of elements of L is polynomial. Then OPT$_\mathcal{L}$ and FULLOPT$_\mathcal{L}$ are in* NP.

Furthermore they are NP-hard under very weak assumptions; see for example, Proposition 5 of [4].

So, the optimization problem, in its simple version (find an element of the Kernel) or its full version is at worst NP-complete. Branch and Bound algorithms can always be used for computing a single optimal solution or even for computing the Kernel. However, this analysis is a little biased, since the size of the Kernel is theoretically large. In the worst case, it is equal to the width of $\preceq$. The width of the Pareto comparison, for instance, is exponential, hence the weakness of the rule; the width of the OOM rule, on the contrary, is limited by the number of levels in the scale.

4 Casting utility networks in the local computation scheme

4.1 Direct encoding

Utility networks can be simply cast in terms of valuation algebras, letting $\mathcal{V} = \bigcup_{S \subseteq X} \{f : D_S \mapsto L\}$ and defining $\boxtimes$ in a pointwise fashion:

Definition 5 *Let $\langle L, \preceq, \otimes \rangle$ be a utility structure and σ, τ two functions for a subset of X to L. Define, for all $d \in D_{scope(\sigma) \cup scope(\tau)}$:*

$$(\tau \boxtimes \sigma)(d) = \tau(d) \otimes \sigma(d).$$

Then the global score function is simply the combination of the c_i in $\mathcal{C}$.

Proposition 3 *For any utility network $\mathcal{C}$ over $\langle L, \preceq, \otimes \rangle$, $score_\mathcal{C} = \boxtimes_{c_i \in \mathcal{C}} c_i$.*

Proposition 4 *$\boxtimes$ satisfies axiom $A2$ iff $\otimes$ is associative and commutative.*

This is trivial, but it gives a fundamental justification for having $\otimes$ associative and commutative. Now, the difficulties arise with the marginalisation operator. The only trivial case is when $\preceq$ is totally ordered. Then the min operator is well defined and we can set

$$\sigma^{\downarrow T}(d) = min_{d' \in D_S, d = proj(d',T)} \sigma(d')$$

It is easy to check that this definition ensures the satisfaction of A1 and A3, and that for any $\mathcal{C}$ built on $\langle L, \preceq, \otimes \rangle$, $(\boxtimes_{c \in \mathcal{C}} c)^{\downarrow \emptyset}$ is the optimal score for $\mathcal{C}$.

We can consider using the same technique when working on a semilattice, i.e. such that there exists an operator $\oplus$ such that $a \preceq b \iff a \oplus b = a$ and $\langle L, \otimes, \oplus \rangle$ is a semiring. It is always possible to define the marginalisation operator for $\oplus$:

Definition 6 *If there exists an operator $\oplus$ such as $a \preceq b \iff a \oplus b = a$ and $\langle L, \otimes, \oplus \rangle$ is a semiring, let us define:*

$$\forall \sigma, d : \sigma^{\downarrow T}(d) = \bigoplus_{d' \in D_S, d = proj(d',T)} \sigma(d').$$

Proposition 5 *Axioms A1, A2 and A3 are satisfied by $\boxtimes$ and $\downarrow$ as defined in Definitions 5 and 6.*

See [9]. $A2$ is satisfied thanks to Proposition 4. $A1$ is satisfied because if $\langle L, \otimes, \oplus \rangle$ is a semiring, then $\oplus$ is commutative and associative. Axiom $A3$ can also be proved fairly easily, since when $\langle L, \otimes, \oplus \rangle$ is a semiring, then $\otimes$ is distributive over $\oplus$.

The problem is that $\oplus$ and $\downarrow$ are not faithful to the notion of optimality in L. First of all because there may be more than one score in the kernel. Secondly, and maybe more importantly, because it may happen the score computed by this marginalisation is not achievable: it may happen that $(\boxtimes_{c \in \mathcal{C}} c)^{\downarrow \emptyset}$ *doesn't belong to the kernel at all*. More precisely, it holds that:

Proposition 6 *Given a utility structure $\langle L, \preceq, \otimes \rangle$, the following assertions are equivalent:*

- $\forall \mathcal{C}, (\boxtimes_{c \in \mathcal{C}} c)^{\downarrow \emptyset} \in Kernel_\preceq(Scores(\mathcal{C}))$
- *$\preceq$ is a total order.*

It is easy to see that the second statement implies the first. Conversely, suppose that $\preceq$ is not totally ordered; then there are two scores a and b that are incomparable, i.e. $a \oplus b \neq a, b$. Then for the trivial problem $\mathcal{C} = \{c\}$ with $c(1) = a$, $c(2) = b$, $Kernel_\preceq(Scores(\{c\})) = \{a, b\}$ while $c^{\downarrow \emptyset}$ provides a unique value $a \oplus b$, which is different from a and from b.

What local computation computes with this direct encoding is actually a lower bound of the Kernel:

Proposition 7 *If $\boxtimes$ and $\downarrow$ are defined according to Definitions 5 and 6, then $\forall a \in Kernel_\preceq(Scores(\mathcal{C}))$, we have $(\boxtimes_{c_i \in \mathcal{C}} c_i)^{\downarrow \emptyset} \preceq a$.*

But once again, it may happen that this score does not belong to the kernel. Proposition 6 is a rather negative result. Variable elimination is indeed potentially exponential in time and space. It may be worthwhile using it if it were providing the optimal score. But the spatial and temporal cost is too high for just an approximation of the result. However, we shall circumvent the difficulty, by working with another comparator. The first solution is to simply refine $\preceq$. This is the topic of the next section.

4.2 Set Encoding

There is a definitive way of using Shenoy's framework to optimize over a utility structure. The idea is to move from L to 2^L, the set of subsets of L. A score can then be a set of scores. Each c_i provides a singleton, nothing is really changed from this point of view. What changes, is the ability of computing a "min": when a and b are not comparable, we keep both when marginalizing. This transformation has been used in [11] in problems of multiobjective optimization problems based on a Pareto comparison. We show here that algebraic utility networks are rich enough to use this kind of transformation.

More formally, let $\mathbf{L} = \{A \subseteq L, A \neq \emptyset,\ s.t.\ A = Kernel_\preceq(A)\}$. Notice that a singleton is its own kernel, thus belongs to $\mathbf{L}$ and that $\mathbf{L}$ is stable w.r.t. the kernel based union : for any $A, B \in \mathbf{L}, Kernel_\preceq(A \cup B) \in \mathbf{L}$

For any constraint c, let $\mathbf{c}$ be the constraint taking it scores in $\mathbf{L}$ defined by:

$$\mathbf{c}(d) = \{c(d)\}$$

and denote $\mathbf{C} = \{\mathbf{c} : c \in \mathcal{C}\}$ the transformation of $\mathcal{C}$ by this "singletonization". Let us now define an operator $\oplus_s$ between sets of scores:

Definition 7 *For all non-empty subsets A and B of L, define: $A \oplus_s B = Kernel_{\preceq}(A \cup B)$.*

The main difference with the direct encoding, that provides an approximation, is that now the $\oplus_s$ operator does represent that preference order:

$a \preceq b \iff \{a\} \oplus_s \{b\} = \{a\}$

With the example of OOM, $\{4^+\} \oplus_s \{6^\pm\} = \{4^+, 6^\pm\}$. But when a score dominates another, one is eliminated, e.g. $\{4^+\} \oplus_s \{6^+\} = \{6^+\}$. This is due to the kernel reduction in the above definition.

The operation of aggregation now has to be able to handle sets of scores.

Definition 8 $\forall A, B \subseteq L, A \otimes_s B = Kernel(\{a \otimes b, a \in A, b \in B\})$.

With the OOM example again, if $A = \{4^+, 6^\pm\}$ and $B = \{5^+\}$ then $A \otimes_s B = Kernel(\{4^+ \otimes 5^+, 6^\pm \otimes 5^+\}) = Kernel(\{5^+, 6^\pm\}) = \{5^+, 6^\pm\}$. If $A = \{4^+, 6^\pm\}$ and $B = \{8^+\}$ then $A \otimes_s B = Kernel(\{4^+ \otimes 8^+, 6^\pm \otimes 8^+\}) = Kernel(\{8^+, 8^\pm\}) = \{8^+\}$.

Proposition 8 $\langle \mathbf{L}, \otimes_s, \oplus_s \rangle$ *is a (commutative) semiring.*

Proposition 5 then implies:

Proposition 9 *Axioms A1, A2 and A3 are satisfied by $\boxtimes$ and $\downarrow$ as defined in Definitions 5 and 6 from the set operations $\otimes_s$ and $\oplus_s$ provided by Definitions 7 and 8.*

The following is the key result that shows that the set of optimal elements $Kernel_{\preceq}(Scores(\mathcal{C}))$ can be expressed as the projection of a combination, which can be computed using local computation because of Proposition 9.

Proposition 10 $(\boxtimes_{\mathbf{c} \in \mathbf{C}} c)^{\downarrow \emptyset} = Kernel_{\preceq}(Scores(\mathcal{C}))$.

A direct consequence of these propositions is that local computation can be used to compute the set of optimal values of any utility network, i.e., variable elimination is possible for *any* utility network.

Now, the theoretical application of local computation must not overshadow its practical range of application. It is known that variable elimination is in the worst case exponential w.r.t. the treewidth of the constraint graph. This is the case if we consider that size of the score sets is 1. Depending on how decisive $\preceq$ is, we may get a larger score set at some point in the computation. The worst case complexity of variable elimination, in time and space, must thus be multiplied by the size of the largest subset of L that contains elements that are pointwisely incomparable with respect to $\preceq$. Mathematically, this number is known as the width of $\preceq$. It is relatively small for some of our examples:

- Its value is 1, obviously, for the total orders (Max CSP and CPT);
- For Pareto comparison on n criteria, the width is exponentially large in the number of criteria. This is an additional reason to prefer refinements when meaningful.
- For the OOM case, the largest kernel is $\{\alpha_1^\pm, \ldots, \alpha_k^\pm\}$, $\{\alpha_1, \ldots, \alpha_k\}$ being the set of possible values for the order of magnitude—typically, reduced to a small selection of qualitative values: "null", negligible", "weak", "significant", "high", "very high".

In practice the width of the order can be a minor issue in comparison to the original complexity of the variable elimination procedure—which is exponential in the tree width of the elimination sequence. If variable elimination is affordable, variable elimination may well also be over partially ordered scales.

5 Conclusion and perspectives

This paper mainly focused on the ways of embedding utility networks into Shenoy and Shafer's framework, showing how it is always possible to benefit from the local computation machinery. But as it is the case for the tolerant Pareto example, there are meaningful structures of preferences that are not captured by utility networks. However, we can often reason with the transitive closure of a non-transitive relation (such as the Tolerant Pareto relation), since if a relation is monotonic then so is its transitive closure. Further research will be developed around the algebraic study of such structures.

REFERENCES

[1] S. Bistarelli, U. Montanari, and F. Rossi, 'Constraint solving over semirings', in *IJCAI'95*, pp. 624–630, (1995).

[2] Gerhard Brewka, 'Preferred subtheories: An extended logical framework for default reasoning', in *IJCAI'89*, pp. 1043–1048, (1989).

[3] Martin Cooper and Thomas Schiex, 'Arc consistency for soft constraints', *Artif. Intelligence*, **154**(1-2), 199–227, (2004).

[4] H. Fargier and N. Wilson, 'Algorithmic schemes for semiring csps', in *Proc. ECSQARU'07*, pp. 623–634, (2007).

[5] Eugene Freuder and Richard Wallace, 'Partial constraint satisfaction', *Artificial Intelligence*, **58**(1-3), 21–70, (1992).

[6] Nir Friedman and Joseph Halpern, 'Plausibility measures and default reasoning', in *AAAI'96*, pp. 1297–1304, (1996).

[7] M.I. Henig, 'The shortest path problem with two objective functions', *EJOR*, **25**, 281–291, (1985).

[8] J. Kohlas, *Information Algebras: Generic Structures for Inference*, Springer-Verlag, 2003.

[9] J. Kohlas and N. Wilson, 'Local computation in semiring induced valuation algebras: Exact and approximate methods', *to appear in Artificial Intelligence*, (2008).

[10] Cédric Pralet, Gérard Verfaillie, and Thomas Schiex, 'Decision with uncertainties, feasibilities, and utilities: Towards a unified algebraic framework', in *ECAI'06*, pp. 427–431, (2006).

[11] Emma Rollon and Javier Larrosa, 'Bucket elimination for multiobjective optimization problems', *J. Heuristics*, **12**(4-5), 307–328, (2006).

[12] Bernard Roy, 'The outranking approach and the foundations of ELECTRE methods', *Theory and Decision*, **31**(1), 49–73, (1991).

[13] Thomas Schiex, Hélène Fargier, and Gerard Verfaillie, 'Valued constraint satisfaction problems: Hard and easy problems', in *IJCAI'95*, pp. 631–637, Montreal, (1995).

[14] Prakash P. Shenoy, 'Valuation-based systems for discrete optimisation', in *UAI'90*, pp. 385–400, (1990).

[15] Prakash P. Shenoy and Glenn Shafer, 'Axioms for probability and belief-function proagation', in *UAI*, pp. 169–198, (1988).

[16] A. Tversky and D. Kahneman, 'Advances in prospect theory: Cumulative representation of uncertainty', *Journal of Risk and Uncertainty*, **5**, 297–323, (1992).

[17] Nic Wilson, 'An order of magnitude calculus', in *UAI'95*, pp. 548–555, (1995).