Local Computation Schemes with Partially Ordered Preferences

Hélène Fargier* and Nic Wilson+

*IRIT, 118 route de Narbonne 31062 Toulouse Cedex, France
+Cork Constraint Computation Centre, University College Cork, Ireland
fargier@irit.fr, n.wilson@4c.ucc.ie

Abstract. Many computational problems linked to uncertainty and preference management can be expressed in terms of computing the marginal(s) of a combination of a collection of valuation functions. Shenoy and Shafer showed how such a computation can be performed using a local computation scheme. 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 the use of partially ordered scales and examines how such scales can be cast in the Shafer-Shenoy framework and thus benefit from local computation algorithms. It also provides many examples of such scales, thus showing that each of the algebraic structures explored here is of interest.

Keywords: Soft CSP; Dynamical programming; Valuation networks/algebra.

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 [16, 15] showed how such a computation can be performed using only local computation (see also, in particular, [9]). 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 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, 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 [14].

But in practice, the all the actual applications of the Shenoy-Shafer framework rely on totally ordered scales of scoring. On the other hand, AI has witnessed the emergence of frameworks based on partial orders. Let us for instance cite semiring constraint satisfaction problems [1], order of magnitude reasoning [18], or belief revision [2, 6]; other examples are obviously provided by multicriteria decision making. The 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. 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 [16, 15, 9]. 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 domains of variables in S. For the sake of simplicity, considering a function f over D_S we shall extend the notation $f(d)$ to tuples d assigning a superset of S, i.e., $f(d) = f(d')$ where $d' = proj(d, S)$, the projection/restriction of d to S. We also adopt the convention that the state space for the empty set $\emptyset$ consists of a single object $\Diamond$: $D_\emptyset = \{\Diamond\}$.

Given a set $S \subseteq X$ of variables there is 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 $scope(\sigma) = S$. $\mathcal{V} = \bigcup_{S \subseteq X} V_S$ 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 over a set of variables $T \subseteq scope(\sigma)$ is a valuation whose scope is T. Let us denote it $\sigma^{\downarrow T}$.

Call $(\mathcal{V}, \boxtimes, \downarrow)$ 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 subset T of X is:

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

Bayesian nets are instances of VNs, where valuations are conditional probability distributions, combined by the product and marginalized using summation. These instances, among many others, satisfy the Shenoy-Shafer 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

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 in [16, 15, 9] that if the valuation algebra satisfies Axioms A1, A2, A3, then for any valuation network over X and for any $Y \subseteq X$, the marginal of the network over Y can be computed by successive variable eliminations. More technically, given a VN network Σ, the basic procedure can be defined as:

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

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 valuation 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 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.

Applications of local computation focus on the case when optimization is made w.r.t. a total order (though see [12, 10]). We will show that it applies to many other situations, which involve only partially ordered scales.

3 Optimization in utility structures

3.1 Utility structures

Let L be a scale on which alternatives, state of the world, possible choices (the interpretation depends on the application) are scored[1] and let $\preceq$ 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. We adopt the convention that $a \preceq b$ means that the score a is at 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, etc. The global score of d is the aggregation of all the $c_i(d)$ according to $\otimes$.

Definition 1 *A utility structure is a triplet $\langle L, \preceq, \otimes\rangle$ which forms an ordered commutative monoid. Its neutral element will be denoted denoted* $\mathbf{1}$.

[1] We cautiously avoid the term"(e)valuation" because of the potential confusion with the notion of valuation used in VNs; in VNs, a valuation is not an element of a scale L, but (often) a function taking its values in L.

That is to say, $\preceq$ is a partial order: a reflexive, anti-symmetric and transitive relation over L (hence $a \sim b$ iff $a = b$), and L is equipped with an internal operation $\otimes$ which is associative, commutative and monotonic w.r.t. $\preceq$ ($a \preceq b \implies a \otimes c \preceq b \otimes c$) and such that $a \otimes \mathbf{1} = a$ for all a.

Before going in more details about the possible property of utility structures, let us present a large class of examples that can be captured by the framework:

• *MAX CSP and VCSP* . In the MAX CSP [5] and (resp. VCSPs [14]) framework, 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 (0) which is the best score is L.

• *Cumulative prospect theory (CPT)* is an old attempt to take into account the positive and negative aspects of decision making [17]. In CPT, each c_i evaluates each possible decision d with a score 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 $\preceq$ is 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.

• *Bi-attribute Pareto decision making:* In many multicriteria problems 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 labeled 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 $L = (\mathbb{N} \cup \{+\infty\}) \times (\mathbb{N} \cup \{+\infty\})$, $\boxtimes$ being the pointwise addition $(a, b) \otimes (a', b') = (a + a', b + b')$. Pairs are compared according to Pareto's rule: $(a, b) \preceq (a, b)$ iff $a \leq a'$ and $b \leq b'$. $\preceq$ is a partial order, e.g., $(3, 2)$ and $(2, 3)$ are incomparable.

• *Order Of Magnitude (OOM) Reasoning* In the system of order of magnitude reasoning described in [18], 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 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 is 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:[2] (i) for all r and s, $\langle +, r\rangle \preceq \langle -, s\rangle$; (ii) for all $s \in \{+, -, \pm\}$, and all r, r' with $r \geq r'$: $\langle +, r\rangle \preceq \langle s, r'\rangle \preceq \langle -, r\rangle$. $\preceq$ is a partial order. However, there are incomparable elements, e.g. $\langle \pm, r\rangle$ and $\langle \pm, s\rangle$ when $r \neq s$.

• *Discrimax comparison* In the application described by [11] one has to satisfy n agents, each of them expressing her preferences by weighted formulas of propositional logic. The "disutilty" of an agent is then the combination, normally the sum (resp. the max), of those of her formulae that are not satisfied: in other terms, $L = (\mathbb{N} \cup \{+\infty\})^n$ and $\otimes$ is the pointwise addition (resp. maximum) of the vectors. In this application, decision making must be fair and egalitarist. So, the disutility of the most unsatisfied agent is minimized: for two vectors a and b, $a \preceq b$ iff $max_{j=1,n,a_j \neq b_j} a_j < max_{j=1,n,a_j \neq b_j} b_j$ or $a = b$. The restriction $a_j \neq b_j$ allows one not to consider the agents that are equally satisfied by a and b (the comparison is made Ceteris Paribus). Hence the name "Discrimax". Other uses of discrimax comparison include belief revision [2] and multi criteria optimisation [8]. The preference relation is only a partial order. For instance, with two agents, $\langle 0, 5\rangle$ and $\langle 5, 0\rangle$ are incomparable vectors. Both $\otimes = +$ and $\otimes = \max$ are associative, commutative, with a neutral element $\langle 0, \ldots, 0\rangle$, but $\otimes = +$ is not monotonic. With $\otimes = \max$, monotonicity holds.

• *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. [13]). It yields a preference relation that 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

[2] 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

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}).

Let us return 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 CSP [1], the first two properties are assumed (semiring CSP are utility structures 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]. 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 structures that cannot be captured by soft CSPs nor semiring CSPs: in the CPT and OOM examples, $\bot$ is not the neutral element; in the Tolerant Pareto example, there exist no operator $\oplus$ encoding $\preceq$. The reason of the last assertion is that in these two cases, $\preceq$ is not a meet semilattice. Intuitively, in these cases, there may be several candidates for $a \oplus b$.

3.2 Optimisation in utility networks

Let us now use utility structure in combinatorial optimisation problems, thus defining utility networks:

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

- *A local function is a function from the domain D_S of some $S \subseteq X$ 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 $Scores(\mathcal{C}) = \{score_{\mathcal{C}}(d) : d \in D_X\}$.

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 $\exists a \in H, score_{\mathcal{C}}(d) \prec a$.*

There problems are easily seen to be in NP. Furthermore they are NP-hard under very weak assumptions, as shown by the following result which is similar to Proposition 5 of [4].

Proposition 2 *Let $\mathcal{L} = \langle L, \preceq, \otimes \rangle$ be a utility structure. Suppose that testing $a \preceq b$ is polynomial, that computing the combination of a multiset of elements of L is polynomial, and that L contains some element a such that $a \succ \mathbf{1}$. Then $OPT_{\mathcal{L}}$ and $FULLOPT_{\mathcal{L}}$ are* NP*-complete.*

So, the optimization problem in its simple version (find an element of the Kernel) or its full version (find the Kernel) is not harder in the case of a partially ordered scale than in the case of a totally ordered one. Branch and Bound algorithms can always be used for computing a single optimal solution or even for computing the Kernel. But this analysis is somewhat 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

In the following, we focus on the ways of embedding utility networks into Shenoy and Shafer's framework in order to benefit from the local computation machinery. First, we show that a direct encoding of the utility structure is inadequate. Two alternative ways are then investigated: the use of a a refinement of the original preference order (this provides one of the optimal scores, provided that such a refinement exists) and the use of a set encoding of the utility structure (this is always possible and provides all the the optimal scores).

4.1 Direct encoding

Utility networks can be simply cast as a problem of combination of valuations, 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. For any $d \in D_{scope(\sigma) \cup scope(\tau)}$, define $(\tau \boxtimes \sigma)(d) = \tau(d) \otimes \sigma(d)$*

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 $\langle L, \preceq, \otimes \rangle$, $Score = \boxtimes_{c_i \in \mathcal{C}} c_i$.*

Also, $\boxtimes$ satisfies axiom $A2$ iff $\otimes$ is associative and commutative, which gives a fundamental justification for having $\otimes$ associative and commutative in preference structures. 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

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

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 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:

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 $\downarrow$ as:*

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

Proposition 4 *Axioms A1, A2 and A3 are satisfied by* $\boxtimes$ *and* $\oplus$ *as defined in Definitions 5 and 6.*

See [10], Theorem 2. 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: $(\boxtimes_{c \in \mathcal{C}} c)^{\downarrow \emptyset}$ *does not necessarily belong to the kernel at all.* More precisely, it holds that:

Proposition 5 *Given a utility structure* $\langle L, \preceq, \otimes \rangle$, *and if* $\boxtimes$ *and* $\downarrow$ *are defined according to Definitions 5, 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.*

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

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

But once again, it may happen that this score does not belong to the kernel. Proposition 5 is a rather negative result. A variable elimination approach is indeed potentially exponential in time and space. It may be worthwhile using it if it were providing the optimal score. But the computational cost is too high for just an approximation of the result. We shall circumvent the difficulty, by working with another comparator. The first solution is to simply refine $\preceq$.

4.2 Refining $\preceq$

A classical approach in Pareto-based multicriteria optimisation problems is to optimize a linear combination of the criteria. The important idea here is that one optimizes according to a new comparator, say $\trianglelefteq$, such that $a \preceq b$ implies $a \trianglelefteq b$: if a is preferred to b according to the original relation, then it is still the case with the new one. But $\trianglelefteq$ can rank scores that are incomparable w.r.t. $\preceq$. Such a relation is called a refinement of the original relation.

Definition 7 $\trianglelefteq$ *refines* $\preceq$ *if and only if* $a \preceq b$ *implies* $a \trianglelefteq b$.

In the Cost/Time Pareto case, we shall decide $a \trianglelefteq b$ iff $a_{cost} + \beta \ . \ a_{time} \leq b_{cost} + \beta \ . \ b_{time}$. $\trianglelefteq$ is complete and if β is high enough, there are no ties, i.e. $\trianglelefteq$ is a total order.

Optimizing with respect to a refinement leads to solutions that are optimal with respect to the original relation. More precisely:

Proposition 7 *If* $\trianglelefteq$ *refines* $\preceq$, *then whatever* A, $Kernel_{\trianglelefteq}(A) \subseteq Kernel_{\preceq}(A)$

Now, if there exists a totally ordered refinement $\trianglelefteq$ of $\preceq$ such that $< L, \trianglelefteq, \otimes >$ is a monotonic utility structure, it is then possible to define $\oplus$ as the min of two scores according to $\trianglelefteq$. Then Definitions 5 and 6 can be applied from $< L, \trianglelefteq, \otimes >$ and Axioms A1, A2 and A3 are satisfied, thanks to Proposition 4.

Like the approach described in Section 4.1, the present one provides the user with a unique score among the optimal ones, but this one has the advantage of being reached by one of the optimal solutions.

Unfortunately, such a totally ordered refinement does not necessarily exist. Consider for instance the case where L is the set of integers, $\otimes = \times$ and any $\preceq$ making $\langle L, \preceq, \otimes \rangle$ a utility structure (e.g., $a \preceq b \iff a = b$). Since $\trianglerighteq$ is total, we have either $1 \trianglerighteq -1$ or $-1 \trianglerighteq 1$. $1 \trianglerighteq -1$ implies by monotonicity $1 \otimes -1 \trianglerighteq -1 \otimes -1$, and so $-1 \trianglerighteq 1$. Similarly, $-1 \trianglerighteq 1$ implies $1 \trianglerighteq -1$, so in either case we have $1 \trianglerighteq -1 \trianglerighteq 1$, contradicting antisymmetry. The following result gives sufficient conditions for an appropriate refinement to exist. (a^1 is defined to be a, and, for $k \geq 1$, $a^{k+1} = a^k \otimes a$.)

Theorem 1. *Let $\langle L, \preceq, \otimes \rangle$ be a utility structure with unit element* $\mathbf{1}$*, which also satisfies the following two properties:*

(i) for all $a, b \in L$ with $a \neq b$ and all $k > 0$ we have $a^k \neq b^k$;
(ii) $a \otimes c \preceq b \otimes c \Rightarrow a \preceq b$ for all $a, b, c \in L$.

Then there exists a total order $\trianglelefteq$ on L extending $\preceq$ and such that for all $a, b, c \in L$, $a \otimes c \trianglelefteq b \otimes c \iff a \trianglelefteq b$, and so, in particular, $\langle L, \trianglelefteq, \otimes \rangle$ is a utility structure.

4.3 Set Encoding

There is a definitive way of using Shafer and 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 by Rollon and Larrosa [12] 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 8 *For all non-empty subsets A and B of L, define: $A \oplus_s B = Kernel_{\preceq}(A \cup B)$.*

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

Definition 9 *$\forall A, B \subseteq L, A \otimes_s B = Kernel(\{a \otimes b, a \in A, b \in B\})$.*

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

Proposition 4 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 8 and 9.*

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 discriminating $\preceq$ 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 treewidth of the elimination sequence. If variable elimination is affordable, it may work well 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. More precisely, we have shown that (i) a direct encoding is not always sound w.r.t. optimality, (ii) the definition of a refinement, widely used in multi criteria optimization, can be applied in certain cases (though not always); (iii) it always possible to benefit from the local computation machinery by using a set encoding.

But as it is the case for the tolerant Pareto example, there are meaningful structures of preferences that are not captured by utility networks. Other examples include preorders and semiorders, that allow richer indifference relations. Further research will be developed around the algebraic study of such structures.

Acknowledgements

The work of the second author has been supported by the Science Foundation Ireland under Grant No. 00/PI.1/C075 and Grant No. 08/PI/I1912.

References

1. S. Bistarelli, U. Montanari, and F. Rossi. Constraint solving over semirings. In *IJCAI'95*, pages 624–630, 1995.
2. G. Brewka. Preferred subtheories: An extended logical framework for default reasoning. In *IJCAI'89*, pages 1043–1048, 1989.
3. M. Cooper and T. Schiex. Arc consistency for soft constraints. *Artif. Intelligence*, 154(1-2):199–227, 2004.
4. H. Fargier and N. Wilson. Algebraic structures for bipolar constraint-based reasoning. In *Proc. ECSQARU'07*, pages 623–634, 2007.
5. E. Freuder and R. Wallace. Partial constraint satisfaction. *Artificial Intelligence*, 58(1-3):21–70, 1992.
6. N. Friedman and J. Halpern. Plausibility measures and default reasoning. In *AAAI'96*, pages 1297–1304, 1996.
7. M.I. Henig. The shortest path problem with two objective functions. *EJOR*, 25:281–291, 1985.
8. U. Junker. Preference-based search and multi-criteria optimization. In *AAAI'02*, pages 34–40, 2002.
9. J. Kohlas. *Information Algebras: Generic Structures for Inference*. Springer-Verlag, 2003.
10. J. Kohlas and N. Wilson. Semiring induced valuation algebras: Exact and approximate local computation algorithms. *Artificial Intelligence*, 172:1360–1399, 2008.
11. C. Lafage and J. Lang. Logical representation of preferences for group decision making. In *KR'00*, pages 457–468, 2000.
12. E. Rollon and J. Larrosa. Bucket elimination for multiobjective optimization problems. *J. Heuristics*, 12(4-5):307–328, 2006.
13. B. Roy. The outranking approach and the foundations of ELECTRE methods. *Theory and Decision*, 31(1):49–73, 1991.
14. T. Schiex, H. Fargier, and G. Verfaillie. Valued constraint satisfaction problems: Hard and easy problems. In *IJCAI'95*, pages 631–637, Montreal, 1995.
15. P. P. Shenoy. Valuation-based systems for discrete optimisation. In *UAI'90*, pages 385–400, 1990.
16. P. P. Shenoy and G. Shafer. Axioms for probability and belief-function proagation. In *UAI*, pages 169–198, 1988.
17. A. Tversky and D. Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, 5:297–323, 1992.
18. N. Wilson. An order of magnitude calculus. In *UAI'95*, pages 548–555, 1995.