Partial Valuation Structures for Qualitative Soft Constraints*

Alexander Schiendorfer, Alexander Knapp, Jan-Philipp Steghöfer, Gerrit Anders, Florian Siefert, and Wolfgang Reif

Institute for Software and Systems Engineering, University of Augsburg, Germany
{schiendorfer, knapp, steghoefer, anders, siefert, reif}@isse.de

Abstract. Soft constraints have proved to be a versatile tool for the specification and implementation of decision making in adaptive systems. A plethora of formalisms have been devised to capture different notions of preference. Wirsing et al. have proposed partial valuation structures as a unifying algebraic structure for several soft constraint formalisms, including quantitative and qualitative ones, which, in particular, supports lexicographic products in a broad range of cases. We demonstrate the versatility of partial valuation structures by integrating the qualitative formalism of constraint relationships as well as the hybrid concept of constraint hierarchies. The latter inherently relies on lexicographic combinations, but it turns out that not all can be covered directly by partial valuation structures. We therefore investigate a notion for simulating partial valuation structures not amenable to lexicographic combinations by better suited ones. The concepts are illustrated by a case study in decentralized energy management.

1 Introduction

Adaptive systems consisting of a large number of interacting components as treated in Organic Computing [26] or Ensembles [14] rely on formalisms to specify models of their complex behavior. Equipped with adequate abstract goal models that describe a corridor of correct behavior, these systems become amenable to formal verification [20] as well as testing [11]. Modeling both the concrete and the abstract components' behavior in terms of relations of their system variables representing input and output naturally leads to the framework of constraint programming. If these models are also used by the system at runtime to actually implement the decision-making, constraint satisfaction and optimization techniques can be applied. Clearly, problems can become overconstrained. Hence, constraint satisfaction has been extended to soft constraints [19].

In constraint hierarchies [8], users qualitatively put constraints into layers represented by a family of sets of constraints $(H_i)_{i \in I}$ where a constraint in layer H_j is valued less important than a constraint in layer H_i if $j > i$. A lexicographic ordering is then established by prioritizing the satisfaction degree of more important layers. This satisfaction degree is evaluated on an assignment and may include metric real-valued

* This research is partly sponsored by the German Research Foundation (DFG) in the project "OC-Trust" (FOR 1085).

error functions for constraints. So-called *comparators* define the ordering over assignments. By definition, constraint hierarchies tend to ignore all constraints on higher levels which leads to a strongly hierarchical evaluation. One satisfied constraint is possibly worth more than a whole set of other, violated constraints.

More recently, constraint relationships [21] have been proposed to capture qualitative statements over soft constraints such as "Prefer a solution violating constraint a to one that violates b" without having to express this fact numerically. This allows for flexible use especially with problems changing at runtime [22,17] as faced with dynamically reconfiguring groups of power plants as described in Sect. 2. However, constraint relationships only consider predicates in lieu of using error metrics. Problems from distributed energy management [2] call for both those formalisms. If a problem admits metric real-valued error functions, one may want to use constraint hierarchies. If, on the other hand, the solution quality is measured by the number and importance of satisfied boolean properties, constraint relationships provide a less restrictive framework.

A broad variety of soft constraint approaches have been captured by the generalizing algebraic formalisms of *c-semirings* and *valued constraints*. That way, users may specify their preferences in the most suitable formalism for the task at hand and rely on a well-defined algebraic underpinning. C-semirings [5] include a set of satisfaction degrees, one operator to combine and one to compare (i.e., calculating a supremum) them as well as a minimal and maximal element to express total dissatisfaction and satisfaction. Frameworks and algorithms based on c-semirings have been devised to build a general theory of soft constraints as well as to provide common solvers [18,10]. Valued constraints [24], on the other hand, use *valuation structures*, i.e., totally ordered monoids instead of the partial order implied by the comparison operator in a c-semiring. The theoretical connection between c-semirings and valued constraints is well understood for totally ordered c-semirings [6]. Recently, the totality in valuation structures was relaxed in [12] following earlier work by Hölzl, Meier and Wirsing [13] to form *partial valuation structures* that also admit lexicographic products for many instances – as opposed to c-semirings. This combination operator offers to specify one's preferences in a more structured way to capture different criteria of descending priority. More complicated partial valuation structures can be formed from elementary ones – allowing for modular implementations and (re)combinations at runtime. These considerations pave the way for the further development of common constraint propagators [9] and search algorithms based on partial valuation structures [13].

As a unifying effort we first represent constraint relationships as partial valuation structures using an algebraically free construction in Sect. 4. For constraint hierarchies, Hosobe established that a reasonable class can be expressed as c-semirings [15]. It remained, however, unclear how to properly draw the boundary between expressible and non-expressible hierarchies. Using Wirsing's results, we can now exploit the lexicographic ordering in constraint hierarchies by mapping layers to partial valuation structures in Sect. 5. Using the insight that certain elements of monoidal soft constraints can be *collapsing* [12], i.e., making comparable elements equal when used with the combination operator, we can give necessary conditions on the partial valuation structures representing layers. In Sect. 5.2 we show that in particular idempotent comparators such as the worst-case comparator in constraint hierarchies cannot directly be represented as a collapse-free partial valuation structure and thus not used in a lexicographical product. However, in Sect. 6 we introduce a notion of *simulation* where another partial valuation structure reasonably mimics the behavior of the worst-case comparator by using a suitable p-norm to induce a collapse-free partial valuation structure.

2 Soft Constraints in Distributed Energy Management

We first give elementary definitions in the realm of classical constraint programming that are then exemplified by a real world application in distributed energy management.

A *constraint domain* (X, D) is given by a set X of *variables* and a family $D = (D_x)_{x \in X}$ of *variable domains* where each D_x is a set representing the possible values for variable x. An *assignment* for a constraint domain (X, D) is a dependent map $v \in \Pi x \in X \,.\, D_x$, i.e., $v(x) \in D_x$; we abbreviate $\Pi x \in X \,.\, D_x$ by $[X \to D]$. A *constraint* c over a constraint domain (X, D), or (X, D)-*constraint*, is given by a map $c : [X \to D] \to \mathbb{B}$. We also write $v \models c$ for $c(v) = tt$.

A *constraint satisfaction problem* (CSP) consists in finding an assignment that yields true for a set of constraints, i.e., a *solution*, and a *constraint satisfaction optimization problem* (CSOP) further seeks to optimize an objective [28] among all solutions. Classical hard constraints are generalized to soft constraints by removing the restriction of constraints to map to true or false [19] but rather an ordered domain. We call these evaluations *gradings* of assignments. In particular, we consider CSOPs that search for maximal gradings in terms of soft constraints.

Such problems occur in many adaptive systems. They are particularly interesting if individual constraint problems must be combined. Adaptive power management provides us with an illustrative example. The main task in power management systems is to maintain the balance between energy production and consumption to avoid instabilities leading to blackouts. Since the prosumers' ability to change their prosumption is subject to physical inertia (e.g., limited ramping rates), the prosumption of controllable prosumers has to be stipulated beforehand as schedules for future points in time.[1]

The concept of *Autonomous Virtual Power Plants* (AVPPs) [27] has been presented as an approach to deal with scalability issues in future smart grids. Each AVPP represents a self-organizing group of two or more prosumers of various types and has to satisfy a fraction of the overall consumption. To accomplish this task, each AVPP autonomously and periodically calculates schedules for its prosumers. Due to uncertainties such as weather conditions, AVPPs can change their composition at runtime to remain manageable. Moreover, the rising complexity with increasing numbers of controlled prosumers motivates the formation of a hierarchical structure of AVPPs following a system-of-systems approach in which hierarchy levels are dynamically created and dissolved. Hence, each AVPP controls less prosumers (including AVPPs) compared to the non-hierarchical case, resulting in shorter scheduling times for each AVPP.

When creating schedules, AVPPs not only have to respect the physical models – in terms of hard constraints – but also their prosumers' individual preferences concerning "good" schedules. For example, a baseload power plant might be reluctant to be

[1] We use the term "prosumer" to refer to producers and consumers, and the term "prosumption" to refer to production and consumption.

switched on and off frequently, whereas a peaking power plant is designed for exactly that purpose. Certainly, prosumers should be free to use whatever specific formalism is most adequate to model their real-life preferences. Consequently, the dynamics of this self-organizing system calls for the treatment and combination of heterogeneous preference specifications at runtime.

To illustrate these considerations, we regard a concrete example of an AVPP consisting of three prosumers: A garbage incineration plant as a thermal power plant where steam drives a generator (`thermal`), a biogas power plant using an engine to produce power (`biogas`), and an electric vehicle that can be used as a power storage when connected to the power grid (`EV`). Each of these prosumers is described by a relational model restricting its physically and economically feasible behavior. These individual models are combined and define the space of feasible schedules [22], ordered by the organizational goal, i.e., to keep mismatches between demand and production (violations) low and the combined preferences of the prosumers. Since blackout prevention is critical, the organizational goal is compared "pessimistically", i.e., by a schedule's worst anticipated violation over a set of future time steps. For instance, two schedules with violations $(0, 0, 3)$ and $(3, 3, 3)$, respectively, for three time steps would be esteemed equal due to the worst violation. For this process of combining shared and individual aspects [23], a common constraint domain (X, D) is used consisting of the smallest set of shared variables, e.g., those for scheduled prosumptions p_t^a for the prosumed power by prosumer a at time step t. Since we are particularly concerned with soft constraints, we deliberately omit the prosumers' hard constraints.

In addition, each prosumer defines its own set of soft constraints in a formalism of its choice over the common constraint domain and additional individual variables. As shown in Fig. 1, `biogas` and `EV` use constraint relationships while `thermal` uses constraint hierarchies. The overall model lexicographically arranges the organizational preference ($\texttt{violation}_{\texttt{org}}$) similar to constraint hierarchies where $\texttt{violation}_{\texttt{org}}$ is put at a higher level H_1^{org} than the individual soft constraints placed on level H_2^{org} since the AVPP's primary objective is arguably to reduce the probability of blackouts. This is regulated by a limitation `maxVio` of the absolute value of the difference between demand d_t and produced power $\sum_{a \in A} p_t^a$. As indicated before, this reflects the semantics of the worst case comparator in constraint hierarchies which could not be expressed by c-semirings in [15]. We provide an explanation for this as well as a solution in Sect. 6.

The electric vehicle can also consume power to load its batteries in which case p_t^a is negative. With regard to the time horizon T schedules are created for, the error function $\mathsf{e}_{\texttt{violation}_{\texttt{org}}}$ associated with the constraint $\texttt{violation}_{\texttt{org}}$ maps to the maximum value by which the threshold `maxVio` is exceeded:

$$\texttt{violation}_{\texttt{org}} \equiv \forall t \in T \,.\, \left| d_t - \textstyle\sum_{a \in A} p_t^a \right| \leq \texttt{maxVio}$$
$$\mathsf{e}_{\texttt{violation}_{\texttt{org}}} \equiv \max_{t \in T} \max\{0, \left| d_t - \textstyle\sum_{a \in A} p_t^a \right| - \texttt{maxVio}\}$$

The model for `biogas` specifies preferences regarding the use of its gas storage tank. It is advisable that this tank is not entirely filled and that the plant runs upon a certain filling threshold since inflow can not be regulated ($\texttt{gasFull}_{\texttt{bio}}$). The plant has to run if the tank is full. Furthermore, the power plant has an economic "sweet spot" which optimizes the ratio of fuel consumption to power production ($\texttt{ecoSweet}_{\texttt{bio}}$) and it should

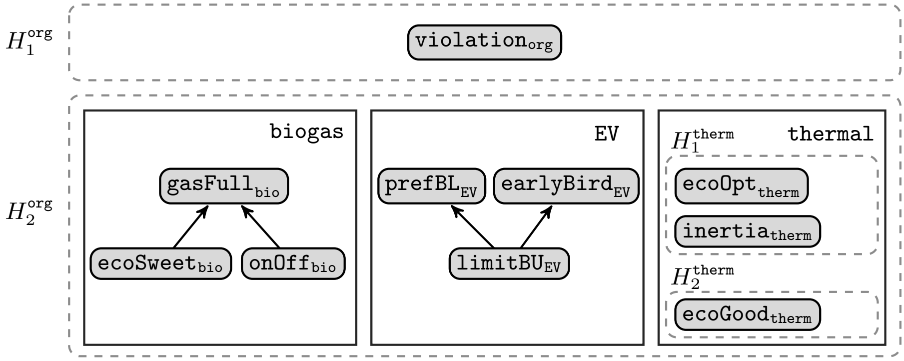

Fig. 1. Case study depicting individual and organizational preference specifications in context

not be frequently switched on and off to minimize maintenance cost ($\texttt{onOff}_{\texttt{bio}}$). Both $\texttt{ecoSweet}_{\texttt{bio}}$ and $\texttt{onOff}_{\texttt{bio}}$ are desirable but deemed less important than $\texttt{gasFull}_{\texttt{bio}}$. No statement regarding their importance is however made. It need not hold that satisfying $\texttt{gasFull}_{\texttt{bio}}$ is worth violating the two others in a strict hierarchical sense. Therefore, constraint relationships are used (see Sect. 4 for details on how an order over assignments is thereby induced).

The preferences of `EV` address its battery status. A preferred battery level should be maintained to allow for emergency trips ($\texttt{prefBL}_{\texttt{EV}}$). To reduce the charging cycles, a soft constraint prescribes that the amount of energy taken out of the battery should not exceed a certain threshold within a specific time frame ($\texttt{limitBU}_{\texttt{EV}}$). Finally, a higher battery charge is required in the morning to assure the trip to work ($\texttt{earlyBird}_{\texttt{EV}}$). Dually to `biogas`, $\texttt{limitBU}_{\texttt{EV}}$ is considered less important than the other constraints.

Finally, `thermal` restricts both the production ranges and the changes in power production due to inertia. The former limitation ensures economically reasonable assignments similar to `biogas` and the latter ensures that `thermal` does not have to be cooled down and heated up all the time at high costs due to energy-intensive processes. As metric error functions are easily found for these constraints, a constraint hierarchy is employed which puts constraints for economical optimality ($\texttt{ecoOpt}_{\texttt{therm}}$) and inertia-based change limits ($\texttt{inertia}_{\texttt{therm}}$) on level $H_1^{\texttt{therm}}$ and constraints for economically still good ranges ($\texttt{ecoGood}_{\texttt{therm}}$) on level $H_2^{\texttt{therm}}$.

Concluding, this example presents three challenges to a soft constraint framework: Adaptive heterogeneous systems need 1) different preference formalisms, 2) combinations of such preference specifications at runtime, and 3) algorithms to solve the resulting soft constraint problems in a general manner.

3 Partial Valuation Structures as a Unifying Formalism

As presented, heterogeneous preference formalisms can show up in soft constraint based systems. Yet, algorithms to find feasible and high quality solutions need some structure to perform constraint propagation or apply branch-and-bound techniques. Seminal work in unifying formalisms has been done in the frameworks of valuation structures and c-semirings. Our following constructions rely on partial valuation structures [12] that turn out to generalize valuation structures [24] by dropping the restriction that the ordering has to be total. Connections with c-semirings are discussed in [16]. First soft constraint solvers based on partial valuation structures using branch-and-bound and constraint propagation have been presented in [13] and [17][2].

3.1 Partial Valuation Structures

Partial valuation structures (also called ic-monoids [13] or meet monoids [16]) capture essential operations for specifying gradings for assignments: Besides providing the set of gradings, they show an associative, commutative multiplication for combining gradings, a partial ordering on gradings such that the multiplication is monotone w.r.t. this ordering, and a top element w.r.t. the partial ordering capturing the best grade, i.e., total satisfaction, that simultaneously is the neutral element for the multiplication.

Definition 1. *A* partial valuation structure $M = (X, \cdot, \varepsilon, \leq)$ *is given by an* underlying set X*, an associative and commutative* multiplication *operation* $\cdot : X \times X \to X$*, a* neutral *element* $\varepsilon \in X$ *for* $\cdot$*, and a partial ordering* $\leq \subseteq X \times X$ *such that the multiplication* $\cdot$ *is monotone in both arguments w.r.t. to* $\leq$*, i.e.,* $m_1 \cdot m_2 \leq m_1' \cdot m_2'$ *if* $m_1 \leq m_1'$ *and* $m_2 \leq m_2'$*, and* ε *is the top element w.r.t.* $\leq$*.*

We write $m_1 < m_2$ *if* $m_1 \leq m_2$ *and* $m_1 \neq m_2$*, and* $m_1 \parallel m_2$ *if neither* $m_1 \leq m_2$ *nor* $m_2 \leq m_1$*. We write* $|M|$ *for the underlying set and* $\cdot_M$*,* ε_M*, and* $\leq_M$ *for the other parts of* M*.* □

Intuitively, $m \leq n$ says that grading m is "worse than" n, so ε will be the top (and best) element of the ordering. In fact, requiring that ε is top is equivalent to requiring that $m \cdot n \leq m$. An illustrative example is the partial valuation structure $(\mathbb{N}, +, 0, \geq)$ used in weighted CSP [19]. The natural numbers represent penalties for violating constraints, with 0 representing satisfaction, and the goal is to minimize the sum of penalties. Another example (previously considered in [4] as a c-semiring) is an inclusion-based partial valuation structure $(\mathfrak{P}(A), \cup, \emptyset, \supseteq)$, where smaller sets are considered better, i.e., $\emptyset$ being best. The sets could, e.g., represent violated constraints.

3.2 Soft Constraints

Classical CSPs are turned into soft CSPs by means of soft constraints mapping assignments to arbitrary gradings instead of $\mathbb{B}$. For a partial valuation structure M, an *M-soft constraint* over a constraint domain (X, D), or *(X, D)-M-soft constraint*, is given by a map $\mu : [X \to D] \to |M|$. The *solution degree* of an assignment w.r.t. a finite set of (X, D)-M-soft constraints M is obtained by combining all gradings using $\cdot_M$, i.e., $\mathsf{M}(v) = \prod_M \{\mu(v) \mid \mu \in \mathsf{M}\}$. This gives rise to the assignment comparison $\lesssim_{\mathsf{M}} \subseteq [X \to D] \times [X \to D]$ with $w \lesssim_{\mathsf{M}} v \iff \mathsf{M}(w) \leq_M \mathsf{M}(v)$, where w is considered worse. The *maximum solution degrees* and the *maximum solutions* of M, which are the goal for solving algorithms, are given by

[2] See `http://git.io/mH_pOg` for this solver.

$$\mathsf{M}^* = \mathrm{Max}^{\leq_M}\{\mathsf{M}(v) \mid v \in [X \to D]\} ,$$
$$\mathrm{Max}^{\lesssim_\mathsf{M}}[X \to D] = \{v \in [X \to D] \mid \mathsf{M}(v) \in \mathsf{M}^*\} .$$

In the process of searching maximum solutions, a vital question is to ask whether the problem formulation actually admits optima. Consider, for example, the constraint domain (X, D) with $X = \{\mathrm{x}\}$, $D_\mathrm{x} = [0, 1]$, and the partial valuation structure $M = ([0, 1], \max, 0, \geq)$ with $\geq$ the usual ordering on real numbers. Let $\mu : [X \to D] \to |M|$ be defined by $\mu(\{\mathrm{x} \mapsto r\}) = r$ if $r > 0$, and $\mu(\{\mathrm{x} \mapsto 0\}) = 1$, and let $\mathsf{M} = \{\mu\}$. Then $\mathsf{M}^* = \emptyset$ since the set of solution degrees is the open interval $(0, 1]$, i.e., no maximum solution degrees and no maximum solutions exist.

Definition 2. *A set of (X, D)-M-soft constraints is* admissible *if* M *is finite and for each $v \in [X \to D]$ there is an $m \in \mathsf{M}^*$ such that $\mathsf{M}(v) \leq_M m$.* □

Sufficient conditions for the finite set M of (X, D)-M-soft constraints to be admissible are that X and $\bigcup_{x \in X} D_x$ are finite, or that $<_M$ has no infinite ascending chains.

3.3 Product Operators for Partial Valuation Structures

For runtime combinations of different soft constraint formulations as are prevalent in adaptive systems, partial valuation structures admit finite (direct) products but also lexicographic products, as shown by Gadducci, Hölzl, Monreale, and Wirsing [12].

First, let us consider the direct product that is defined component-wise obviously yielding a partial valuation structure:

Definition 3. *Let M and N be partial valuation structures. Let*
- $P = |M| \times |N|$,
- $\cdot_P : P \times P \to P$ *given by* $(m_1, n_1) \cdot_P (m_2, n_2) = (m_1 \cdot_M m_2, n_1 \cdot_N n_2)$,
- $\varepsilon_P = (\varepsilon_M, \varepsilon_N)$,
- $\leq_P \subseteq P \times P$ *given by* $(m_1, n_1) \leq_P (m_2, n_2) \iff m_1 \leq_M m_2 \wedge n_1 \leq_N n_2$.

The (direct) product *of M and N, written as $M \times N$, is given by the partial valuation structure $(P, \cdot_P, \varepsilon_P, \leq_P)$.* □

This product leaves many combinations incomparable. Let us thus turn our attention to lexicographic products introduced by [12] useful in situations where a preference is composed of multiple criteria of decreasing priority. The *lexicographic ordering* $\leq_{M \ltimes N} \subseteq |M \times N| \times |M \times N|$ on the direct product distinguishes first by $\leq_M$ and then by $\leq_N$ if the first comparison yields equality:

$$(m_1, n_1) \leq_{M \ltimes N} (m_2, n_2) \iff (m_1 <_M m_2) \vee (m_1 = m_2 \wedge n_1 \leq_N n_2) .$$

However, for $\cdot_{M \times N}$ still to be monotone now w.r.t. $\leq_{M \ltimes N}$, we would have to show that $(m_1, n_1) \cdot_{M \times N} (m, n) \leq_{M \ltimes N} (m_2, n_2) \cdot_{M \times N} (m, n)$ holds if $(m_1, n_1) \leq_{M \ltimes N} (m_2, n_2)$. But this fails, if there are $m_1, m_2, m \in |M|$ such that $m_1 <_M m_2$ and at the same time $m_1 \cdot_M m = m_2 \cdot_M m$. In this case, order-preservation w.r.t. $\leq_N$ does not hold, if $m_1 <_M m_2$ but $n_1 >_N n_2$, since we would have $(m_1, n_1) \cdot_{M \times N} (m, n) >_{M \ltimes N} (m_2, n_2) \cdot_{M \times N} (m, n)$, clearly violating monotonicity.

First, the notion of *collapsing elements* [12] captures the objectionable elements of M as the set

$$\mathfrak{C}(M) = \{m \in |M| \mid \exists m_1, m_2 \in |M| \,.\, m_1 <_M m_2 \wedge m_1 \cdot_M m = m_2 \cdot_M m\} \,.$$

All idempotent elements w.r.t. $\cdot_M$ different from ε_M are collapsing: if $m \cdot_M m = m$, we have $m <_M \varepsilon_M$ but $m \cdot_M m = \varepsilon \cdot_M m = m$. On the other hand, $\varepsilon_M \notin \mathfrak{C}(M)$ since $m_1 <_M m_2$ implies $m_1 \cdot_M \varepsilon_M <_M m_2 \cdot_M \varepsilon_M$. Furthermore, $|M| \setminus \mathfrak{C}(M)$ is closed under $\cdot_M$, and thus $(|M| \setminus \mathfrak{C}(M), \cdot_M, \varepsilon_M, \leq_M)$ forms a partial valuation structure.

Second, the notion of bounded partial valuation structures [12] allows to avoid the comparison of pairs (m, n) with $m \in \mathfrak{C}(M)$ by requiring that then n must be the smallest element of N: A partial valuation structure N is *bounded* if $|N|$ has a smallest element $\bot_N$ w.r.t. $\leq_M$. Then $\bot_N$ is unique and annihilating for $\cdot_N$, i.e., $n \cdot_N \bot_N = \bot_N$ for all $n \in |N|$. We can always *lift* a partial valuation structure M into a bounded partial valuation structure $M_\bot = (|M| \cup \{\bot\}, \cdot_{M_\bot}, \varepsilon_M, \leq_{M_\bot})$ by using a fresh $\bot$ and extending $\cdot_M$ and $\leq_M$ by $m \cdot_{M_\bot} \bot = \bot$ and $\bot \leq_{M_\bot} m$ for all $m \in |M| \cup \{\bot\}$.

Equipped with these concepts, we can define the lexicographic product of partial valuation structures. The well-definedness of this construction, i.e., that it indeed yields a partial valuation structure, has been shown in [12].

Definition 4. *Let M be a partial valuation structure and let N be a bounded partial valuation structure. Let*

- $L = ((|M| \setminus \mathfrak{C}(M)) \times |N|) \cup (\mathfrak{C}(M) \times \{\bot_N\})$,
- $\cdot_L : L \times L \to L$ *given by* $(m_1, n_1) \cdot_L (m_2, n_2) = (m_1 \cdot_M m_2, n_1 \cdot_N n_2)$,
- $\varepsilon_L = (\varepsilon_M, \varepsilon_N)$,
- $\leq_L \subseteq L \times L$ *given by* $(m_1, n_1) \leq_L (m_2, n_2) \iff (m_1 <_M m_2) \vee (m_1 = m_2 \wedge n_1 \leq_N n_2)$.

The lexicographic product *of M and N, written as $M \ltimes N$, is given by the partial valuation structure $(L, \cdot_L, \varepsilon_L, \leq_L)$.* □

Consequently, all collapsing elements have to be ignored for the lexicographic product. However, idempotent operators such as a worst case combination found in constraint hierarchies (and present in our case study in Sect. 2 when evaluating an assignment based on the worst violation over several time steps) or fuzzy and possibilistic constraints [19] necessarily lead to collapsing elements – an issue we address in Sect. 6.

However, using combinations of partial valuation structures by means of direct and lexicographic products, we are able to model the scenario depicted in Sect. 2 and also reuse them to present constraint hierarchies as partial valuation structures. But first we consider constraint relationships as a representative.

4 Constraint Relationships as Partial Valuation Structures

Partial valuation structures enable us to give an algebraic structure capable of representing preferences specified with constraint relationships. We revisit this construction first presented in [17] and [16], where we describe how to lift a quantitative preference specification over constraints to sets of violated constraints (representing assignments).

4.1 Constraint Relationships

A *directed acyclic graph*, or *DAG*, $G = (|G|, \to_G)$ is given by a set $|G|$ and a binary relation $\to_G \subseteq |G| \times |G|$ such that $\to_G^+$ is irreflexive. If $x \to_G y$, then x is a *predecessor* of y, and y is a *successor* of x. We obtain a partial order $PO\langle G \rangle = (|G|, \to_G^*)$ from G by taking the reflexive, transitive closure of $\to_G$, and write $g \leq_{PO\langle G \rangle} h$ if $g \to_G^* h$.

A *constraint relationship* over a constraint domain (X, D), or (X, D)*-constraint relationship*, is given by a DAG C with $|C|$ a finite set of (X, D)-constraints. We think of a constraint $c' \in |C|$ as *more important* than another constraint $c \in |C|$ if $c \to_C c'$.

For $V, W \subseteq |C|$, which we think of being sets of *violated* constraints by (X, D)-assignments v and w (i.e., $V = \{c \in |C| \mid v \not\models c\}$ and similarly for W), we want to express that W is *worse* than V w.r.t. C. We describe two kinds of liftings of the partial ordering induced by the DAG C to an ordering over subsets of $|C|$, using two *dominance properties* p: *single-predecessor* dominance ($p =$ SPD) and *transitive-predecessors* dominance ($p =$ TPD) as originally defined in [21]. Intuitively, dominance properties denote how much more important a constraint is compared to its predecessors to the quality of a solution. In SPD, a constraint can dominate only one less important one; in TPD, a single constraint is deemed more important than a whole set of predecessors.

We write $V \leadsto_C^p W$ for "V worsens to W for dominance property p over C". Both dominance properties share the following worsening rule, expressing that violating strictly more constraints is worse ($V_1 \uplus V_2$ denotes the union of V_1 and V_2 simultaneously requiring that V_1 and V_2 are disjoint):

$$V \leadsto_C^p V \uplus \{c\} \qquad \text{if } c \in |C| \qquad \text{(W)}$$

The remaining rules for SPD and TPD express which constraint violations can be "traded" under the *ceteris paribus* assumption represented by $\uplus$:

$$V \uplus \{c\} \leadsto_C^{\mathrm{SPD}} V \uplus \{c'\} \qquad \text{if } c \to_C c' \qquad \text{(SPD)}$$

$$V \uplus \{c_1, \ldots, c_k\} \leadsto_C^{\mathrm{TPD}} V \uplus \{c'\} \qquad \text{if } \forall i \,.\, c_i \to_C^+ c' \qquad \text{(TPD)}$$

These worsening relations induce partial orderings $\leq_C^p$ over sets of (violated) constraints for $p \in \{\mathrm{SPD}, \mathrm{TPD}\}$, when defining $W <_C^p V$ if, and only if, $V\ (\leadsto_C^p)^+\ W$ (meaning repeated sequential application of the rules); this is to be read as "W is worse than V". Note that, by definition, the empty set is the *top* element w.r.t. to these orderings, meeting the intuition that "no violations" should be considered optimal since $\emptyset \leadsto_C^p V \neq \emptyset$. By abuse of notation, for assignments we also write $w <_C^p v$ if $\{c \in |C| \mid w \not\models c\} <_C^p \{c \in |C| \mid v \not\models c\}$, also read as "$w$ is worse than v".

4.2 From Constraint Relationships to Partial Valuation Structures

When abstracting from constraint relationships and casting them as a partial valuation structure, one might be tempted to start from the inclusion-based structure and extending it to accept an ordering over the constraints. The empty set, representing the fact that no constraints are violated, is the top element and simultaneously the neutral element for the union. But set union is idempotent. Consider an exemplary constraint relationship C with $|\mathrm{C}| = \{\mathrm{a}, \mathrm{b}\}$ and $\mathrm{b} \to_{\mathrm{C}} \mathrm{a}$. Then $\{\mathrm{a}\} <_{\mathrm{C}}^{\mathrm{SPD}} \{\mathrm{b}\}$ holds. Multiplying on both sides with $\{\mathrm{a}\}$, i.e., taking the union, would result in $\{\mathrm{a}\} \leq_{\mathrm{C}}^{\mathrm{SPD}} \{\mathrm{a}, \mathrm{b}\}$ by the required monotonicity of the multiplication. Hence, violating a only would be worse than violating both a and b, contradicting (W). However, we can patch this defect by not considering sets and their union but multisets and the multiset union as hinted by the disjointness assumptions in (SPD) and (TPD). Incidentally, when equipping multisets with an appropriate ordering induced by the partial order from the constraint relationship, the *free* partial valuation structure over the constraint relationship is obtained.

We denote the set of finite multisets over a set S by $\mathfrak{M}_{\text{fin}}(S)$, and the multiset union by $\uplus$. For a partial order $P = (|P|, \leq_P)$, we define the *upper* or *Smyth ordering*[3] on $\mathfrak{M}_{\text{fin}}\,|P|$ as the binary relation $\Subset^P \subseteq (\mathfrak{M}_{\text{fin}}\,|P|) \times (\mathfrak{M}_{\text{fin}}\,|P|)$ given by the transitive closure of

$$T \supseteq U \text{ implies } T \Subset^P U\ ,$$
$$p \leq_P q \text{ implies } T \uplus \wr p \int \Subset^P T \uplus \wr q \int\ .$$

This relation is indeed a partial ordering on $\mathfrak{M}_{\text{fin}}\,|P|$ and $PVS\langle P\rangle = (\mathfrak{M}_{\text{fin}}\,|P|, \uplus, \wr\int, \Subset^P)$ indeed a partial valuation structure. Moreover, $PVS\langle P\rangle$ is the *free* partial valuation structure over the partial order P in the sense of universal algebra. Thus, we have (for a detailed proof, see [16, §12]):

Lemma 1. *Let P be a partial order. Then $PVS\langle P\rangle = (\mathfrak{M}_{\text{fin}}\,|P|, \uplus, \wr\int, \Subset^P)$ is the free partial valuation structure over P.* □

The upper ordering, when employed for sets, exactly corresponds to $\leq_{C^{-1}}^{\mathrm{SPD}}$ for a constraint relationship C: We need to invert C, i.e., consider $PVS\langle PO\langle C^{-1}\rangle\rangle$, as violating more important constraints has to lead to worse solutions. We get the corresponding set of (X, D)-$PVS\langle PO\langle C^{-1}\rangle\rangle$-soft constraints $\mathsf{P} = \{\varphi_c \mid c \in |C|\}$ where $\varphi_c(v) = \wr c \int$ if $v \not\models c$ and $\wr\int$ otherwise for $v \in [X \to D]$. However, the transitive-predecessors dominance can only be achieved by using a more specialized ordering.

This partial valuation structure can now be used to capture the preferences issued by the prosumers `EV` and `biogas` from our case study, see Fig. 1. For `biogas` we have the DAG $\mathrm{C} = (\{\mathtt{onOff}_{\mathtt{bio}}, \mathtt{gasFull}_{\mathtt{bio}}, \mathtt{ecoSweet}_{\mathtt{bio}}\}, \{\mathtt{onOff}_{\mathtt{bio}} \to_{\mathrm{C}} \mathtt{gasFull}_{\mathtt{bio}}, \mathtt{ecoSweet}_{\mathtt{bio}} \to_{\mathrm{C}} \mathtt{gasFull}_{\mathtt{bio}}\})$. Assume we were to choose between the assignments v_1 and v_2 with $v_1 \not\models \{\mathtt{gasFull}_{\mathtt{bio}}, \mathtt{ecoSweet}_{\mathtt{bio}}\}$, $v_2 \not\models \{\mathtt{onOff}_{\mathtt{bio}}, \mathtt{ecoSweet}_{\mathtt{bio}}\}$. In $PVS\langle PO\langle \mathrm{C}^{-1}\rangle\rangle$, v_1 is graded as $\mathsf{P}(v_1) = \wr \mathtt{gasFull}_{\mathtt{bio}}, \mathtt{ecoSweet}_{\mathtt{bio}} \int$ and v_2 is graded as $\mathsf{P}(v_2) = \wr \mathtt{onOff}_{\mathtt{bio}}, \mathtt{ecoSweet}_{\mathtt{bio}} \int$. Thus we get that $\mathsf{P}(v_1) \Subset^{PO\langle \mathrm{C}^{-1}\rangle} \mathsf{P}(v_2)$, i.e., $\mathsf{P}(v_1)$ is worse than $\mathsf{P}(v_2)$ since $\mathtt{gasFull}_{\mathtt{bio}} \to_{\mathrm{C}^{-1}} \mathtt{onOff}_{\mathtt{bio}}$ and therefore $\mathtt{gasFull}_{\mathtt{bio}} \leq_{PO\langle \mathrm{C}^{-1}\rangle} \mathtt{onOff}_{\mathtt{bio}}$. This meets our intuition as $\mathtt{gasFull}_{\mathtt{bio}}$ is denoted more important (and thus more detrimental if violated) than $\mathtt{onOff}_{\mathtt{bio}}$.

5 Expressing Constraint Hierarchies as Lexicographic Products

As motivated by Sect. 2, constraint relationships provide the ability to combine unrelated preferences without introducing bias, as would occur if categorizing unrelated

[3] This multiset ordering mimics the eponymous ordering used in powerdomain constructions [1, Ch. 9], where partial orders are lifted to semi-lattices with an idempotent multiplication.

constraints into more or less equivalent layers in constraint hierarchies. However, constraint hierarchies are more appropriate when metric error functions are available or a clear dominance of one layer over others exists – as might be the case in relating organizational vs. individual goals. Using lexicographic combinations, both approaches can be seamlessly combined.

We first recast the original definitions of constraint hierarchies [8] to position them within the scope of partial valuation structures. In particular, we represent a constraint hierarchy as a lexicographical product of partial valuation structures in place of the layers. We discuss the existing propositions of weighting functions but increase the generality of the approach as arbitrary partial valuation structures could eventually be lexicographically combined to form hierarchies. The presence of collapsing elements gives us a criterion that algebraic structures defining a combination operation for gradings such as partial valuation structures or c-semirings representing layers in a constraint hierarchy need to show in order to be used in lexicographic combinations: Soft constraints in all but the least important layer should not map to collapsing elements to preserve all gradings. All constraint hierarchies classified as "rational" in [15] (and thus expressible as c-semirings) are void of collapsing elements.

Formally, a *constraint hierarchy* $H = (C_k)_{1 \leq k \leq n}$ over a constraint domain (X, D), or (X, D)*-constraint hierarchy*, is given by a family of sets C_k of (X, D)-constraints. The constraints in *level* $1 \leq k \leq n$ are considered as *strictly more important* than the ones in level $k + 1$. An (X, D)-constraint hierarchy is *finite* if $\bigcup_{1 \leq k \leq n} C_k$ is finite.

Let $H = (C_k)_{1 \leq k \leq n}$ be a finite (X, D)-constraint hierarchy, let $W = (M_k)_{1 \leq k \leq n}$ be a corresponding family of partial valuation structures M_k representing the individual layers, and let for each $1 \leq k \leq n$ and for each $c \in C_k$, μ_c be the associated (X, D)-M_k-soft constraint. We call $\mathsf{H} = (\mathsf{M}_k)_{1 \leq k \leq n}$ with $\mathsf{M}_k = \{\mu_c \mid c \in C_k\}$ for $1 \leq k \leq n$ a (X, D)-W*-soft constraint hierarchy*. For a $v \in [X \to D]$ the *solution degree* for $(\mathsf{M}_k)_{1 \leq k \leq n}$ of v is defined to be $(\mathsf{M}_k(v))_{1 \leq k \leq n}$. Define a binary relation $<_{\mathsf{H}} \subseteq [X \to D] \times [X \to D]$ by

$$
\begin{aligned}
w <_{\mathsf{H}} v \iff \exists 1 \leq k \leq n \, . \quad & (\forall 1 \leq i \leq k-1 \, . \, \mathsf{M}_i(w) = \mathsf{M}_i(v)) \\
& \wedge \mathsf{M}_k(w) <_{M_k} \mathsf{M}_k(v) \; ,
\end{aligned}
$$

saying that the assignment w is *strictly worse* than the assignment v if ties up to a certain level $k - 1$ (or no ties if $k = 1$) are resolved by a strict inequality in k. This corresponds to the lexicographic order on the set $\{(\mathsf{M}_k(v))_{1 \leq k \leq n} \mid v \in [X \to D]\}$, i.e.,

$$
w <_{\mathsf{H}} v \iff (\mathsf{M}_k(w))_{1 \leq k \leq n} <_{M_1 \ltimes \ldots \ltimes M_n} (\mathsf{M}_k(v))_{1 \leq k \leq n}
$$

if, on the one hand, every M_k is a bounded partial valuation structure for all $2 \leq k \leq n$, and, on the other hand, $\mathsf{M}_k(v), \mathsf{M}_k(w) \notin \mathfrak{C}(M_k)$ for all $1 \leq k \leq n$, or, equivalently, if $\mu_c(v), \mu_c(w) \notin \mathfrak{C}(M_k)$ for each $c \in C_k$, $1 \leq k \leq n$. The first requirement, that each M_k is bounded, can be achieved by moving from M_k to its lifted variant $(M_k)_\perp$. The second hinges on the selected partial valuation structure, guaranteeing order equivalence *only if* no collapsing elements are present. In practice, this requires that no soft constraint maps to any collapsing element.

5.1 Locally Predicate Better

In the literature, many different variants are used for the comparison of solution degrees of individual layers. A straightforward approach requests that an assignment is considered worse if it is Pareto-dominated in terms of soft constraints, i.e., it violates a strict superset of constraints of another assignment's violation set. Consider a single level k of a finite (X, D)-constraint hierarchy $H = (C_k)_{1 \le k \le n}$, and let $C = C_k$. The so-called *locally-predicate-better* (LPB)-*comparator* [8] for C corresponds to requiring

$$w <_C^{\mathrm{LPB}} v \iff \{c \in C \mid w \not\models c\} \supset \{c \in C \mid v \not\models c\} .$$

This is expressed by the partial valuation structure $M = (\mathfrak{P}_{\mathrm{fin}}(C), \cup, \emptyset, \supseteq)$ where $\mathfrak{P}_{\mathrm{fin}}(C)$ stands for finite subsets of C and the set of (X, D)-M-soft constraints $\mathsf{M} = \{\mu_c \mid c \in C\}$ with $\mu_c(v) = \{c\}$ if $v \not\models c$ and $\mu_c(v) = \emptyset$ otherwise, for each $c \in C$. However, all elements of M are idempotent, and thus the collapsing elements of M are $\mathfrak{P}_{\mathrm{fin}}(C) \setminus \{\emptyset\}$. Hence, M is not suitable for a lexicographic product. Choosing instead the partial valuation structure $N = (\mathfrak{M}_{\mathrm{fin}}(C), \uplus, \wr\!\int, \sqsupseteq)$, where $\mathfrak{M}_{\mathrm{fin}}(C)$ denotes finite multisets over C, N has no collapsing elements and the set of (X, D)-N-soft constraints $\mathsf{N} = \{\nu_c \mid c \in C\}$ with $\nu_c(v) = \wr c \int$ if $v \not\models c$ and $\nu_c(v) = \wr\!\int$ otherwise, for each $c \in C$, deviates this situation, since we have for all $v, v' \in [X \to D]$ that

$$\mathsf{M}(v) \le_M \mathsf{M}(v') \iff \mathsf{N}(v) \le_N \mathsf{N}(v')$$

as any $\nu(c)$ adds at most one occurrence of c to the combined grading.

One may think of the ordering over $[X \to D]$ induced by M as preference *specification* that is *implemented* by N which is applicable to lexicographic products due to the absence of collapsing elements. More specifically, from a user's point of view, the used structure is not relevant as long as the intended ordering is preserved. We can generalize this idea of substituting a specifying partial valuation structure by another implementing collapse-free counterpart:

Definition 5. *A finite set of* (X, D)-M*-soft constraints* M *and a finite set of* (X, D)-N*-soft constraints* N *are* optima equivalent*, written as* $\mathsf{M} \approx \mathsf{N}$*, if* $\mathrm{Max}^{\lesssim_{\mathsf{M}}}[X \to D] = \mathrm{Max}^{\lesssim_{\mathsf{N}}}[X \to D]$. □

5.2 Globally Weighted Better

The locally predicate better comparator, however, leaves us with various incomparable assignments due to the proper subset relation. Moreover, predicate evaluations may be too strict and metric error functions can take their role. Additionally, constraints may be weighted. Thus, a more general approach does not consider constraints at the individual level but maps a layer to one aggregated value (corresponding to $\mathsf{M}_k(v)$). Borning called these comparators *global* [8]. That way, we can also treat locally predicate (and metric) better as special cases.

Formally, a *weighting* for a set C of (X, D)-constraints is given by a function $g : C \times [X \to D] \to \mathbb{R}_{\ge 0}$ with $g(c, v) = 0$ iff $v \models c$ for $v \in [X \to D]$ and $c \in C$. This function subsumes both the metric aspects and weights. Traditionally, the following combinations of weights have been considered, where a valuation is deemed worse than another if its combined weight is greater than the combined weight of the other.

– *Weighted sum*: $W_1(v) = \sum_{c \in C} g(c, v)$.
– *Least squares*: $W_2(v) = \sqrt{\sum_{c \in C} g(c, v)^2}$.
– *Worst case*: $W_\infty(v) = \max\{g(c, v) \mid c \in C\}$.

These comparators can be recast as partial valuation structures based on the real numbers where the ordering is just $\geq$:

Definition 6. *A* real *partial valuation structure R has $0 \in |R| \subseteq \mathbb{R}_{\geq 0}$ for its underlying set, 0 as its neutral element and the (inverted) usual ordering on the real numbers $\geq$ as its ordering.* □

The following real partial valuation structures capture the global comparators; the notation R_∞ is justified by the well-known fact that $\lim_{p \to \infty}(r^p + s^p)^{1/p} = \max\{r, s\}$:[4]

– *Weighted sum*: $\mathrm{R}_1 = (\mathbb{R}_{\geq 0}, \cdot_1, 0, \geq)$ with $r \cdot_1 s = r + s$;
– *Least squares*: $\mathrm{R}_2 = (\mathbb{R}_{\geq 0}, \cdot_2, 0, \geq)$ with $r \cdot_2 s = \sqrt{r^2 + s^2}$;
– *p-norm* for $p > 0$: $\mathrm{R}_p = (\mathbb{R}_{\geq 0}, \cdot_p, 0, \geq)$ with $r \cdot_p s = (r^p + s^p)^{1/p}$;
– *Worst case*: $\mathrm{R}_\infty = (\mathbb{R}_{\geq 0}, \cdot_\infty, 0, \geq)$ with $r \cdot_\infty s = \max\{r, s\}$.

Given a real partial valuation structure R and a weighting $g : C \times [X \to D] \to |R| \subseteq \mathbb{R}_{\geq 0}$, the (R, g)-*weighting* of a $v \in [X \to D]$ is now given by $W^g_R(v) = \prod_R \{g(c, v) \mid c \in C\}$. Each such weighting W induces a relation $\lesssim^W_C \subseteq [X \to D] \times [X \to D]$ on assignments with $w \lesssim^W_C v$ denoting w is worse than v, defined by

$$w \lesssim^W_C v \iff W(w) \geq W(v) \ .$$

Let us now turn to the question how to use these real partial valuation structures in a lexicographic product. All real partial valuation structures R with $\cdot_R = \cdot_p$ for some $p > 0$ are appealing as they have no collapsing elements, since $r \cdot_p s = (r^p + s^p)^{1/p}$ is strictly monotonic in both arguments. The choices of weighted-sum-better and least-squares-better are thus readily applicable to lexicographic products. For real partial valuation structures with $\cdot_R = \cdot_\infty$, however, $\mathfrak{C}(R) = |R| \setminus \{0\}$, since $\cdot_\infty$ is idempotent. Consequently, one cannot use them to mimic the ordering of a (X, D)-W-soft constraint hierarchy using a lexicographic product since the resulting partial valuation structures would degrade to $(\{0\}, \cdot_\infty, 0, \geq)$. Assume, e.g., that C has three different constraints c_1, c_2, and c_3; that there are assignments v_1 violating only c_1, v_2 violating only c_2, v_{13} violating exactly c_1 and c_3, and v_{23} violating exactly c_2 and c_3; and that the weightings are independent of the valuation, i.e., $g(c_1, v_1) = g(c_1, v_{13})$ and $g(c_2, v_2) = g(c_2, v_{23})$ and $g(c_3, v_{13}) = g(c_3, v_{23})$. Also assume that the weightings for v_1, v_2, v_{13}, and v_{23} are related by

$$\begin{aligned} W^g_{\mathrm{R}_\infty}(v_1) &= g(c_1, v_1) > g(c_2, v_2) = W^g_{\mathrm{R}_\infty}(v_2) \ , \\ W^g_{\mathrm{R}_\infty}(v_{13}) &= \max\{g(c_1, v_{13}), g(c_3, v_{13})\} = \\ &\qquad \max\{g(c_2, v_{23}), g(c_3, v_{23})\} = W^g_{\mathrm{R}_\infty}(v_{23}) \ . \end{aligned}$$

[4] The choice of $\cdot_R$ for a real partial valuation structure is somewhat limited by the following theorem by Bohnenblust [7]: If $|R| = \mathbb{R}_{\geq 0}$ and $(t \cdot r) \cdot_R (t \cdot s) = t \cdot (r \cdot_R s)$ holds in the real partial valuation structure R for all $r, s, t \in \mathbb{R}_{\geq 0}$ (where $\cdot$ is the usual multiplication), then either $1 \cdot_R 1 = 1$ and $r \cdot_R s = \max\{r, s\}$ for all $r, s \in \mathbb{R}_{\geq 0}$, or $1 \cdot_R 1 > 1$ and $r \cdot_R s = (r^p + s^p)^{1/p}$ for all $r, s \in \mathbb{R}_{\geq 0}$ for some $p > 0$.

Intuitively, $g(c_3, v)$ is greater than $g(c_1, v)$ and $g(c_2, v)$ if $v \not\models c_3$, but as only the worst case is considered, all other gradings do not contribute to the distinction. Therefore, previously comparable assignments become equal when combined with $g(c_3, v)$. Any set of (X, D)-M-soft constraints $\mathsf{M} = \{\mu_c \mid c \in C\}$ reflecting the ordering induced by $W^g_{\mathrm{R}_\infty}$ on assignments, i.e., $\mathsf{M}(v) \leq_M \mathsf{M}(v') \iff W^g_{\mathrm{R}_\infty}(v) \geq W^g_{\mathrm{R}_\infty}(v')$, would thus have μ_{c_3} mapping to a collapsing element in M. To still implement a partial valuation structure that meets our preference specifications originally stated in R_∞, we have to abandon the search for optima equivalence (see Def. 5) for a less restrictive property.

6 Simulating Partial Valuation Structures

A variety of application scenarios, however, motivate the evaluation of assignments based on the worst criterion including our examples in Sect. 2. To still be able to use "worst case" as a valid comparator for lexicographic products, we first relax our notion of optima equivalence to the asymmetric optima simulation. A similar effort was made by Bistarelli, Codognet, and Rossi, who discuss abstractions of c-semiring-based soft constraint problems by means of Galois connections [3]. The problem can also be seen in the context of viewpoints in model reformulation [25] in the sense that we seek an alternative partial valuation structure that reflects the same underlying user preferences.

Definition 7. *A finite set of (X, D)-N-soft constraints* N optima simulates *a finite set of (X, D)-M-soft constraints* M*, written as* $\mathsf{N} \preccurlyeq \mathsf{M}$*, if for each* $v_\mathsf{M} \in \mathrm{Max}^{\lesssim_\mathsf{M}}[X \to D]$ *there is a* $v_\mathsf{N} \in \mathrm{Max}^{\lesssim_\mathsf{N}}[X \to D]$ *with* $\mathsf{M}(v_\mathsf{M}) = \mathsf{M}(v_\mathsf{N})$*, and, vice versa, if for each* $v_\mathsf{N} \in \mathrm{Max}^{\lesssim_\mathsf{N}}[X \to D]$ *there is a* $v_\mathsf{M} \in \mathrm{Max}^{\lesssim_\mathsf{M}}[X \to D]$ *with* $\mathsf{M}(v_\mathsf{M}) = \mathsf{M}(v_\mathsf{N})$. □

Intuitively, our definition of optima simulation allows that assignments in the same equivalence class w.r.t. M are *further* distinguished in N as long as each equivalence class in $\mathrm{Max}^{\lesssim_\mathsf{M}}[X \to D]$ is represented in $\mathrm{Max}^{\lesssim_\mathsf{N}}[X \to D]$ (we do not "lose" optima) and no assignment suboptimal in M is considered optimal in N. Then, N is a reasonable candidate for substituting M, constituting a kind of refinement. Obviously, $\mathsf{M} \approx \mathsf{N}$ if, and only if, $\mathsf{N} \preccurlyeq \mathsf{M}$ and $\mathsf{M} \preccurlyeq \mathsf{N}$. We can furthermore give sufficient criteria for the relations of assignments evaluated in M and N to check if $\mathsf{N} \preccurlyeq \mathsf{M}$ holds, provided that both M and N are admissible:

Lemma 2. *Let (X, D) be a constraint domain, and let* M *and* N *be admissible sets of M- and N-soft constraints over (X, D), respectively, such that for all $v, v' \in [X \to D]$*

$$\mathsf{M}(v) <_M \mathsf{M}(v') \quad \textit{implies} \quad \mathsf{N}(v) <_N \mathsf{N}(v')$$
$$\mathsf{M}(v) \parallel_M \mathsf{M}(v') \quad \textit{implies} \quad \mathsf{N}(v) \parallel_N \mathsf{N}(v')$$

Then $\mathsf{N} \preccurlyeq \mathsf{M}$.

Proof. Let first $v_1 \in \mathrm{Max}^{\lesssim_\mathsf{M}}[X \to D]$. Let $v_1 \notin \mathrm{Max}^{\lesssim_\mathsf{N}}[X \to D]$. Then, since N is admissible, there is a $v_2 \in \mathrm{Max}^{\lesssim_\mathsf{N}}[X \to D]$ with $\mathsf{N}(v_1) <_N \mathsf{N}(v_2)$. Moreover, there is a $v_1' \in \mathrm{Max}^{\lesssim_\mathsf{M}}[X \to D]$ with $\mathsf{M}(v_2) \leq_M \mathsf{M}(v_1')$, since M is admissible. But $\mathsf{M}(v_2) <_M \mathsf{M}(v_1')$ is impossible, since then also $\mathsf{N}(v_2) <_N \mathsf{N}(v_1')$ contradicting $\mathsf{N}(v_2) \in \mathsf{N}^*$.

Thus $\mathsf{M}(v_2) = \mathsf{M}(v_1')$. Moreover, either $\mathsf{M}(v_1) \parallel_M \mathsf{M}(v_1')$ or $\mathsf{M}(v_1) = \mathsf{M}(v_1')$ since both $\mathsf{M}(v_1)$ and $\mathsf{M}(v_1')$ are elements of M^*. But $\mathsf{M}(v_1) \parallel_M \mathsf{M}(v_1')$ is impossible, since we would have $\mathsf{M}(v_1) \parallel_M \mathsf{M}(v_2) = \mathsf{M}(v_1')$ and $\mathsf{N}(v_1) <_N \mathsf{N}(v_2)$. Thus $\mathsf{M}(v_2) = \mathsf{M}(v_1') = \mathsf{M}(v_1)$. — Now let $v_2 \in \mathrm{Max}^{\lesssim_\mathsf{N}}[X \to D]$. If $v_2 \notin \mathrm{Max}^{\lesssim_\mathsf{M}}[X \to D]$, there would be, since M is admissible, a $v_1 \in \mathrm{Max}^{\lesssim_\mathsf{M}}[X \to D]$ such that $\mathsf{M}(v_2) <_M \mathsf{M}(v_1)$, i.e. $\mathsf{N}(v_2) <_N \mathsf{N}(v_1)$, contradicting $\mathsf{N}(v_2) \in \mathsf{N}^*$. □

The requirements of the lemma prove helpful in finding a collapse-free simulating partial valuation structure for a real partial valuation structure using $\cdot_\infty$. In particular, we investigate the use of $\cdot_p$ as a substitute for $\cdot_\infty$, since this directly avoids collapsing elements. For that purpose, for a $0 \in V \subseteq \mathbb{R}_{\geq 0}$, let, for each $p > 0$, V_p be the real partial valuation structure $(\langle V \rangle_p, \cdot_p, 0, \geq)$ with $\langle V \rangle_p$ the smallest subset of $\mathbb{R}_{\geq 0}$ with $r \cdot_p s \in \langle V \rangle_p$ if $r, s \in \langle V \rangle_p$; and let V_∞ denote the real partial valuation structure $(V, \cdot_\infty, 0, \geq)$. The second requirement of the lemma for moving from a V_∞ to some V_p is trivially satisfied for real partial valuation structures, since $\geq$ is total. For the first requirement we have the following characterization:

Lemma 3. *Let* $0 \in V \subseteq \mathbb{R}_{\geq 0}$*, and* $p > 0$*. Then for each* $n \geq 1$

$$\textstyle\prod_\infty \vec{r} < \prod_\infty \vec{s} \quad \textit{implies} \quad \prod_p \vec{r} < \prod_p \vec{s} \quad \textit{for all } \vec{r}, \vec{s} \in V^n \qquad (*_p)$$

if, and only if,

$$r < s \quad \textit{implies} \quad n^{1/p} \cdot r < s \quad \textit{for all } r, s \in V. \qquad (**_p)$$

Proof. Let first $(*_p)$ hold and let $r, s \in V$ with $r < s$. Choose $r_1 = \ldots = r_n = r$, $s_1 = \ldots = s_{n-1} = 0$, and $s_n = s$. Then $\prod_\infty (r_i)_{1 \leq i \leq n} = r < s = \prod_\infty (s_i)_{1 \leq i \leq n}$, and thus $n^{1/p} \cdot r = \prod_p (r_i)_{1 \leq i \leq n} < \prod_p (s_i)_{1 \leq i \leq n} = s$. — Now, let $(**_p)$ hold and let $r = \prod_\infty (r_i)_{1 \leq i \leq n} < \prod_\infty (s_i)_{1 \leq i \leq n} = s$. Define $r_1' = \ldots = r_n' = r$ and $s_1' = \ldots = s_{n-1}' = 0$, $s_n' = s$. Then $\prod_p (r_i)_{1 \leq i \leq n} \leq \prod_p (r_i')_{1 \leq i \leq n} = n^{1/p} \cdot r$, since $r_i \leq r$ for all $1 \leq i \leq n$, and $s = \prod_p (s_i')_{1 \leq i \leq n} \leq \prod_p (s_i)_{1 \leq i \leq n}$, since $0 \leq s_i$ for all $1 \leq i \leq n$. Then $\prod_p (r_i)_{1 \leq i \leq n} \leq n^{1/p} \cdot r < s \leq \prod_p (s_i)_{1 \leq i \leq n}$. □

The lemma shows that $r < n^{1/p} \cdot r < s$ is required for all $0 \neq r < s \in V$. But this is only satisfiable if there is no $t \in V$ with $r < t \leq n^{1/p} \cdot r$, since by $(**_p)$ we would get $r < n^{1/p} \cdot r < t$. In particular, $V = \mathbb{R}_{\geq 0}$ cannot be simulated.

We call a $0 \in V \subseteq \mathbb{R}_{\geq 0}$ *δ-separated* for some $\delta > 1$ if $s/r \geq \delta$ for all $0 \neq r < s \in V$. For each δ-separated V and $n \geq 1$, $(**_p)$ holds if $p > \ln n / \ln \delta$, i.e. $n^{1/p} < \delta$: Let $r < s$ for $r, s \in V$. Then either $r = 0$, and thus $n^{1/p} \cdot r = 0 = r < s$, or $r \neq 0$, and thus $n^{1/p} \cdot r < \delta \cdot r \leq s$. Moreover, not only does δ-separation provide us with a suitable p, a set $0 \in V \subseteq \mathbb{R}_{\geq 0}$ must be δ-separated for $(**_p)$ to hold: If $0 \in V \subseteq \mathbb{R}_{\geq 0}$ for each $\delta > 1$ shows $0 \neq r < s \in V$ with $s/r < \delta$, then $(**_p)$ is violated for each $p > 0$, since we can choose $0 \neq r < s \in V$ with $s/r < n^{1/p}$, and then $n^{1/p} \cdot r > s$.

Example 1. (1) Let $0 \in V \subseteq \mathbb{R}_{\geq 0}$ be finite. Then there is a $\varepsilon > 0$ such that $|r_1 - r_2| \geq \varepsilon$ for all $r_1 \neq r_2 \in V$. Let $0 \neq r < s \in V$. Then $s/r \geq (r + \varepsilon)/r = 1 + \varepsilon/r \geq 1 + \varepsilon/\max V$. Thus V is $(1 + \varepsilon/\max V)$-separated.

(2) Let $c \in \mathbb{R}$ with $c > 1$ and let $V^c = \{c^n \mid n \in \mathbb{N}\} \cup \{0\}$. If $0 \neq r < s \in V^c$, then there are $m < n$ with $r = c^m$ and $s = c^n$. Then $c^n/c^m = c^{n-m} \geq c$ holds. Thus, V^c is c-separated and unbounded.
(3) Let $d \in \mathbb{R}$ with $d > 1$ and let $V^d = \{d^{-n} \mid n \in \mathbb{N}\} \cup \{0\}$. If $0 \neq r < s \in V^d$, then there are $m < n$ with $r = d^{-n}$ and $s = d^{-m}$. Then $d^{-m}/d^{-n} = d^{-m+n} > d$ holds. In addition, $0 < d^{-n} \leq d$ for all $n \in \mathbb{N}$. Hence, V^d is d-separated and bounded. □

Wrapping up, we can define a suitable simulating partial valuation structure for V_∞ by means of a p-norm to deal with preference specifications requiring the worst case.

Proposition 1. *Let (X, D) be a constraint domain, $0 \in V \subseteq \mathbb{R}_{\geq 0}$ δ-separated, M_∞ an admissible set of (X, D)-V_∞-soft constraints, and $p > \ln |\mathsf{M}_\infty| / \ln \delta$. Define $\tau_p : |V_\infty| \to |V_p|$ by $\tau_p(r) = r$ and the finite set of (X, D)-V_p-soft constraints M_p by $\mathsf{M}_p = \{\tau_p \circ \mu \mid \mu \in \mathsf{M}_\infty\}$. If M_p is admissible, then $\mathsf{M}_p \preccurlyeq \mathsf{M}_\infty$.*

Proof. The claim that $\mathsf{M}_p \preccurlyeq \mathsf{M}_\infty$ follows from Lem. 2 by the choice of p and the totality of the order in V_∞. □

This construction gives us a tool for practical scenarios requiring a worst-case comparator that are, as we showed, not directly expressible in lexicographic products of partial valuation structures or c-semirings [15] due to the presence of collapsing elements. For a finite set of (X, D)-constraints C and for a weighting $g : C \times [X \to D] \to \mathbb{R}_{\geq 0}$, let $V = \{g(c, v) \mid c \in C,\ v \in [X \to D]\} \cup \{0\}$. It has now to be checked that V is δ-separated for some $\delta > 1$. Classical CSPs are dealing with finite domains, and thus δ-separatedness is readily applicable in these scenarios. For real-valued possible error values, one could turn to discretizing and bounding a domain according to Ex. 1(1). We may then choose $p > \ln |C| / \ln \delta$. For each $c \in C$, we have the (X, D)-V_∞-soft constraint c^g_∞ by $c^g_\infty(v) = g(c, v)$, and the (X, D)-V_p-soft constraint c^g_p by $c^g_p(v) = g(c, v)$. Then, since C is finite, we obtain $\{c^g_p \mid c \in C\} \preccurlyeq \{c^g_\infty \mid c \in C\}$, provided that both $\{c^g_\infty \mid c \in C\}$ and $\{c^g_p \mid c \in C\}$ are admissible. But if V is finite, this is guaranteed as observed in Sect. 3.2. Example 1(3), however, shows that δ-separatedness alone does not imply admissibility.

Let us apply the construction to the organizational preferences of Sect. 2. Assume that the possible error values representing violations are given by $V = \{0, 1, 2, 3\}$ and that the finite set of V_x-soft constraints $\mathsf{V}_x = \{v_t \mid t \in T\}$ represent the violations at time step $t \in T = \{1, 2, 3\}$ (with $x = \infty$ or $x = p$). Assume two assignments w_1 and w_2 such that $v_1(w_1) = 3$ and $v_t(w_1) = 0$ if $t > 1$; and further $v_t(w_2) = 2$ for all $t \in T$. Since V is finite, by Ex. 1(1) we get that it is δ-separated with $\delta = 1 + 1/3 \approx 1.3$. In fact, it is also 1.5-separated since 3 cannot take the role of r in Ex. 1(1). With $n = |\mathsf{V}| = 3$, we get that we have to choose p greater than $\ln 3 / \ln 1.5 \approx 2.71$ for $\mathsf{V}_p \preccurlyeq \mathsf{V}_\infty$ to work. Indeed, we get that while $\mathsf{V}_\infty(w_1) = 3 > \mathsf{V}_\infty(w_2) = 2$, $p = 2$ is not high enough to preserve this ordering as $\mathsf{V}_2(w_1) = 3 < \mathsf{V}_2(w_2) \approx 3.46$, leading to an incorrect preference decision. But choosing $p = 3$ already preserves the ordering correctly as $\mathsf{V}_3(w_1) = 3 > \mathsf{V}_3(w_2) = 2.88$ and we thus have $\mathsf{V}_3 \preccurlyeq \mathsf{V}_\infty$.

This makes the construction applicable for a lexicographic product with its controlled prosumers' preferences. These are in turn also given as partial valuation structures: for `biogas` and `EV`, we use the free partial valuation structure over the partial order induced by their constraint relationships calling them $P_{\texttt{biogas}}$ and $P_{\texttt{EV}}$; for thermal we use p-norms to either directly translate the desired comparators or also use simulation to get a worst-case comparator and a lexicographic product to obtain a partial valuation structure $P^1_{\texttt{thermal}} \ltimes P^2_{\texttt{thermal}}$. Since no prosumer is considered more important than the others, we combine their preferences with a direct product. In accordance with Fig. 1 we thus get the partial valuation structure

$$V_3 \ltimes (P_{\texttt{biogas}} \times P_{\texttt{EV}} \times (P^1_{\texttt{thermal}} \ltimes P^2_{\texttt{thermal}}))$$

for the overall soft constraint problem where $P_{\texttt{biogas}}$ and $P_{\texttt{EV}}$ are partial valuation structures originating in constraint relationships and $(P^1_{\texttt{thermal}} \ltimes P^2_{\texttt{thermal}})$ represents a constraint hierarchy.

7 Conclusions

Based on results of Wirsing et al., we showed how to express different qualitative and quantitative preference formalisms as partial valuation structures. First we expressed the representation of constraint relationships as partial valuation structures by a free construction. Second, the lexicographical product associated with partial valuation structures allowed us to reformulate constraint hierarchies to position them in a soft constraint framework. This process also led to the negative result that a direct translation of the worst-case comparator necessarily leads to partial valuation structures with collapsing elements. This fact hindered previous attempts at expressing constraint hierarchies as c-semirings. However, it is possible to look for collapsing-free partial valuation structures that fulfill several qualities regarding the assignment ordering. We therefore introduced the notion of optima simulation and provided an example of a real-valued partial valuation structure implemented with p-norms which can be used to order assignments in lieu of the original collapsing worst-case comparator. We have also demonstrated by means of a small case study that adaptive and organic computing applications can benefit from the presented ideas since reconfiguration and clustering call for compositionality which the more conventional c-semirings do not offer to the same extent.

However, our simulation result for the worst-case comparator still is burdened by some computational effort for the involved p-norms. In fact, it seems that a more general construction for optima simulation at least for totally-ordered partial valuation structures (i.e., valuation structures) is reachable, that may avoid this effort. Furthermore, based on these constructions, efficient solving and optimization algorithms and propagators need to be devised to make them available to problems of practical interest.

Dedication. The authors express their gratitude to Martin Wirsing for his encouraging style in research and teaching, displaying a kind and appreciative attitude towards the work of colleagues as well as motivating to connect rigorous methods with software engineering.

References

1. Amadio, R.M., Curien, P.L.: Domains and Lambda-Calculi. Cambridge Tracts in Theoretical Computer Science, vol. 46. Cambridge University Press (1998)
2. Anders, G., Schiendorfer, A., Steghöfer, J.P., Reif, W.: Robust Scheduling in a Self-Organizing Hierarchy of Autonomous Virtual Power Plants. In: Stechele, W., Wild, T. (eds.) Proc. 2nd Int. Wsh. Self-optimisation in Organic and Autonomic Computing Systems (SAOS 2014), pp. 1–8 (2014)
3. Bistarelli, S., Codognet, P., Rossi, F.: Abstracting Soft Constraints: Framework, Properties, Examples. Artif. Intell. 139, 175–211 (2002)
4. Bistarelli, S., Frühwirth, T., Marte, M., Rossi, F.: Soft Constraint Propagation and Solving in Constraint Handling Rules. Computational Intelligence 20(2), 287–307 (2004)
5. Bistarelli, S., Montanari, U., Rossi, F.: Semiring-based Constraint Satisfaction and Optimization. J. ACM 44(2), 201–236 (1997)
6. Bistarelli, S., Montanari, U., Rossi, F., Schiex, T., Verfaillie, G., Fargier, H.: Semiring-Based CSPs and Valued CSPs: Frameworks, Properties, and Comparison. Constraints 4(3), 199–240 (1999)
7. Bohnenblust, H.F.: An Axiomatic Characterization of L_p-spaces. Duke Math. J. 6, 627–640 (1940)
8. Borning, A., Freeman-Benson, B., Wilson, M.: Constraint Hierarchies. LISP Symb. Comp. 5, 223–270 (1992)
9. Cooper, M., Schiex, T.: Arc Consistency for Soft Constraints. Artificial Intelligence 154(1), 199–227 (2004)
10. Delgado, A., Olarte, C.A., Pérez, J.A., Rueda, C.: Implementing Semiring-Based Constraints Using Mozart. In: Van Roy, P. (ed.) MOZ 2004. LNCS, vol. 3389, pp. 224–236. Springer, Heidelberg (2005)
11. Eberhardinger, B., Seebach, H., Knapp, A., Reif, W.: Towards Testing Self-organizing, Adaptive Systems. In: Merayo, M.G., de Oca, E.M. (eds.) ICTSS 2014. LNCS, vol. 8763, pp. 180–185. Springer, Heidelberg (2014)
12. Gadducci, F., Hölzl, M., Monreale, G.V., Wirsing, M.: Soft constraints for lexicographic orders. In: Castro, F., Gelbukh, A., González, M. (eds.) MICAI 2013, Part I. LNCS, vol. 8265, pp. 68–79. Springer, Heidelberg (2013)
13. Hölzl, M., Meier, M., Wirsing, M.: Which Soft Constraints do you Prefer? In: Proc. 7th Int. Wsh. Rewriting Logic and its Applications (WRLA 2008). Electronic Notes in Theoretical Computer Science, vol. 238(3), pp. 189–205 (2009)
14. Hölzl, M., Wirsing, M.: Towards a System Model for Ensembles. In: Agha, G., Danvy, O., Meseguer, J. (eds.) Formal Modeling: Actors, Open Systems, Biological Systems. LNCS, vol. 7000, pp. 241–261. Springer, Heidelberg (2011)
15. Hosobe, H.: Constraint Hierarchies as Semiring-Based CSPs. In: Proc. 21st Int. Conf. Tools with Artificial Intelligence (ICTAI 2009), pp. 176–183 (2009)
16. Knapp, A., Schiendorfer, A.: Embedding Constraint Relationships into C-Semirings. Tech. Rep. 2014-03, Institute for Software and Systems Engineering, University of Augsburg (2014), `http://opus.bibliothek.uni-augsburg.de/opus4/frontdoor/index/index/docId/2684`
17. Knapp, A., Schiendorfer, A., Reif, W.: Quality over Quantity in Soft Constraints. In: Proc. 26th Int. Conf. Tools with Artificial Intelligence (ICTAI 2014), pp. 453–460 (2014)
18. Leenen, L., Anbulagan, A., Meyer, T., Ghose, A.K.: Modeling and Solving Semiring Constraint Satisfaction Problems by Transformation to Weighted Semiring Max-SAT. In: Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS (LNAI), vol. 4830, pp. 202–212. Springer, Heidelberg (2007)

19. Meseguer, P., Rossi, F., Schiex, T.: Soft Constraints. In: Rossi, F., van Beek, P., Walsh, T. (eds.) Handbook of Constraint Programming, ch. 9 (2006)
20. Nafz, F., Seebach, H., Steghöfer, J.P., Anders, G., Reif, W.: Constraining Self-organisation Through Corridors of Correct Behaviour: The Restore Invariant Approach. In: Müller-Schloer, C., Schmeck, H., Ungerer, T. (eds.) Organic Computing – A Paradigm Shift for Complex Systems. Autonomic Systems, vol. 1, pp. 79–93. Springer (2011)
21. Schiendorfer, A., Steghöfer, J.P., Knapp, A., Nafz, F., Reif, W.: Constraint Relationships for Soft Constraints. In: Bramer, M., Petridis, M. (eds.) Proc. 33rd SGAI Int. Conf. Innovative Techniques and Applications of Artificial Intelligence (AI 2013), pp. 241–255. Springer (2013)
22. Schiendorfer, A., Steghöfer, J.P., Reif, W.: Synthesis and Abstraction of Constraint Models for Hierarchical Resource Allocation Problems. In: Proc. 6th Int. Conf. Agents and Artificial Intelligence (ICAART 2014), vol. 2, pp. 15–27. SciTePress (2014)
23. Schiendorfer, A., Steghöfer, J.P., Reif, W.: Synthesised Constraint Models for Distributed Energy Management. In: Proc. 3rd Int. Wsh. Smart Energy Networks & Multi-Agent Systems (SEN-MAS 2014), pp. 1529–1538 (2014)
24. Schiex, T., Fargier, H., Verfaillie, G.: Valued Constraint Satisfaction Problems: Hard and Easy Problems. In: Proc. 14th Int. Joint Conf. Artificial Intelligence (IJCAI 1995), vol. 1, pp. 631–639. Morgan Kaufmann (1995)
25. Smith, B.M.: Modelling. In: Rossi, F., van Beek, P., Walsh, T. (eds.) Handbook of Constraint Programming, ch. 11. Elsevier (2006)
26. Steghöfer, J.-P., et al.: Trustworthy Organic Computing Systems: Challenges and Perspectives. In: Xie, B., Branke, J., Sadjadi, S.M., Zhang, D., Zhou, X. (eds.) ATC 2010. LNCS, vol. 6407, pp. 62–76. Springer, Heidelberg (2010)
27. Steghöfer, J.P., Anders, G., Siefert, F., Reif, W.: A System of Systems Approach to the Evolutionary Transformation of Power Management Systems. In: Wsh. Proc. 43th Nat. Conf. GI Jahrestagung (INFORMATIK 2013). Lect. Notes Inf., vol. P-220, Bonner Köllen Verlag (2013)
28. Tsang, E.P.K.: Foundations of Constraint Satisfaction. Computation in Cognitive Science 289. Academic Press (1993)