

Quality over quantity in soft constraints

Alexander Knapp, Alexander Schiendorfer, Wolfgang Reif

Angaben zur Veröffentlichung / Publication details:

Knapp, Alexander, Alexander Schiendorfer, and Wolfgang Reif. 2014. "Quality over quantity in soft constraints." In *2014 IEEE 26th International Conference on Tools with Artificial Intelligence, 10-12 November 2014, Limassol, Cyprus*, edited by Andreas Andreou, 453–60. Los Alamitos, CA: IEEE. <https://doi.org/10.1109/ictai.2014.75>.

Nutzungsbedingungen / Terms of use:

licgercopyright

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:

Deutsches Urheberrecht

Weitere Informationen finden Sie unter: / For more information see:

<https://www.uni-augsburg.de/de/organisation/bibliothek/publizieren-zitieren-archivieren/publiz/>



Quality over Quantity in Soft Constraints

Alexander Knapp, Alexander Schiendorfer, Wolfgang Reif

Institute for Software & Systems Engineering, University of Augsburg, Germany

Email: {knapp, alexander.schiendorfer, reif}@informatik.uni-augsburg.de

Abstract—Partial constraint satisfaction and soft constraints enable to deal with over-constrained problems in practice. Constraint relationships have been introduced to provide a qualitative approach to specifying preferences over the constraints that should be satisfied. In contrast to quantitative approaches like weighted or fuzzy CSPs, the preferences just rely on a directed acyclic graph. The approach is particularly aimed at scenarios where soft-constraint problems stemming from several independently modeled agents have to be aggregated into one problem in a multi-agent system. Existing transformations into weighted CSP introduce unintended, additional preference decisions. We first illustrate the application of constraint relationships in a case study from energy management along with deficiencies of existing work. We then show how to embed constraint relationships into the soft-constraint frameworks of partial valuation structures and further c-semirings by means of free constructions. We finally provide a prototypical implementation of heuristics for the well-known branch-and-bound algorithm along with an empirical evaluation.

I. CONSTRAINTS AND PREFERENCES

Constraint programming (CP) is a versatile paradigm for efficiently modeling and solving various combinatorial problems such as scheduling, resource allocation [1], or fault diagnosis [2]. A *constraint satisfaction problem* (CSP) is formulated in terms of *variables* that are assigned values from their *domains*, and *constraints* that restrict the validity of assignments. Traditionally, CSP treats hard constraints that must not be violated. However, practical problems can become over-constrained, i.e., no assignment satisfying all constraints (a solution) exists. In order to handle such cases, *soft* constraints do not express hard requirements but rather desired properties.

Several formalisms to *grade* solutions have been devised (see, e.g., [3] for an overview): *Partial constraint satisfaction* (PCSP [4]) allows that some constraints may be dropped if a problem is infeasible. The goal is then to search for a *relaxed* problem that is as close as possible to the original problem. In particular, *Max-CSP* seeks to find a solution that satisfies as many constraints of the original problem as possible. However, the search for a feasible problem is agnostic towards the individual relevance of a dropped constraint. Users may be reluctant to ignore an important constraint even if that forces a higher number of violated constraints. Weighted CSPs [5] allow to annotate constraints with weights for that purpose, and hence take a *quantitative* approach. However, finding appropriate weights is notoriously hard in practice, in particular when composing constraint problems [6].

Alternatively, in constraint hierarchies [7], users put constraints into layers represented by a family of sets of constraints $H = (H_i)_{i \in I}$ where H_0 contains hard constraints and a constraint in layer H_j is considered less important than a constraint in layer H_i if $j > i$. A lexicographic order is then

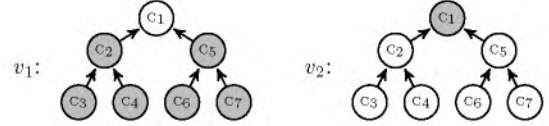


Figure 1. Two assignments v_1 and v_2 satisfying white and violating grey constraints. $c \rightarrow c'$ means that constraint c' is more important to be satisfied than c . Constraint hierarchies (with a locally predicate better comparator [7]) would deem v_1 better since it satisfies the most important constraint, whereas partial CSP (in particular Max-CSP) prefers v_2 as it violates only c_1 .

established based on the satisfaction degree of more important layers. An assignment v_1 performs better than v_2 if they are equal up to some level k and v_1 is strictly better with regard to the constraints in level k . By definition, constraint hierarchies ignore all constraints on levels greater than k , leading to a strict evaluation for particular cases, as illustrated in Fig. 1. Such a “hierarchical” evaluation seems reasonable when violating important constraints is absolutely detrimental but might not be adequate for problems requiring a more “egalitarian” stance.

We therefore argued that there ought to be more flexibility in specification and extended the idea of partial CSP in [8] by introducing *constraint relationships*, i.e., *qualitatively* specifying a set of preferences over constraints represented by a directed acyclic graph (dag). This graph induces a partial order on assignments such that less important constraints may be dropped in favor of more important ones. Although a partial order leads to a less decisive framework and reduces possible pruning due to the existence of incomparable elements, it is useful to leave indifference as an option during the elicitation phase of constraints and preferences [9]. That way, users can be informed about unspecified preferences if two solutions are not comparable by the formalism and eventually add these preferences explicitly. On the other hand, constraint relationships can also express a subclass of constraint hierarchies [8].

Quantitative formalisms such as weighted or fuzzy CSP share common features: A set of values reflects the quality of an assignment which is graded by each constraint. The overall grading of an assignment is then found by combining the constraints’ individual gradings. This very general principle has been formalized in the frameworks of c-semirings [10] and valued constraints [11] to abstract from concrete formalisms and to provide a common ground for algorithms and theoretical considerations (see, e.g., [2], [12], [13]). Recently, [14] proposes to use an even more general formalism, viz., partially ordered monoids to represent soft constraints which are useful for lexicographical combinations.

In this paper, we show how to construct a c-semiring from a qualitative constraint relationship specification as an algebraically free construction over the induced partial order.

Therein, the notion of *partial valuation structures* [14] emerges naturally, again as a free construction over partial orders. An additional free construction from partial valuation structures to c-semirings then enables compliance with existing c-semiring based algorithms and proofs. In previous work [8], we presented constraint relationships for specification along with a mapping to weighted CSP. While this mapping allows us to solve the resulting optimization problems with off-the-shelf constraint solvers, it also totalizes the induced partial order, introducing unintended extra preference relations. Additionally, algorithmic techniques from soft constraints can already be formulated on the level of partial valuation structures. The underlying dag suggests efficient heuristics for computing maximum solution degrees.

For combining different sources into one constraint problem, as used in multi-agent systems (see, e.g., [1]), quantitative approaches to preferences would have to aggregate gradings that are possibly defined on different scales. By contrast, qualitative formalisms like CP-nets [15] avoid such explicit aggregation and keep the underlying rationales behind the relative importance settings of varying goals [16]. Constraint relationships also follow this proposed research direction (see [17] and more recently [9]) and offer an alternative approach to qualitative soft constraints since we showed in [8] that constraint relationships are formally incomparable to CP-nets. Combining constraint relationships from different sources just amounts to taking the union of their dags.

In Sect. II, we illustrate the use of constraint relationships for the constraint combination problem in multi-agent systems with a case study taken from energy management, highlighting the differences between quantitative and qualitative approaches. After summarizing constraint programming with constraint relationships formally in Sect. III, we present our contributions for the connection of constraint relationships and c-semirings: We give algebraically *free* constructions of, first, a partial valuation structure from a dag that qualitatively specifies preferences over constraints without enforcing a total order; and, second, of a c-semiring from a partial valuation structure (see Sect. IV). Then we present heuristics specific to constraint relationships that integrate with the well-known branch-and-bound algorithm that calculates the maximum solution degrees of a partial valuation structure-based soft-constraint problem (see Sect. V). Moreover, we evaluate these heuristics for constraint relationships in a prototypical solver with the source code being available online (see Sect. VI). Finally, we conclude with an outlook to future work.

II. CASE STUDY: POWER PLANT SCHEDULING

We illustrate the qualitative constraint relationships approach and its use in multi-agent systems with a simplified example taken from scheduling power plants [1], showing issues with quantitative approaches when it comes to comparing unrelated soft constraints: Power plants are modeled as agents that dynamically group to organizations, forming *Autonomous Virtual Power Plants* (AVPP). The main task is to distribute the energy demand to power plants. Feasible trajectories of power plants are described by CSPs that consider *hard* constraints like minimal and maximal productions or physically limited rates of change as well as *soft* constraints such as economically better ranges or limited – yet technically feasible – rates

of change. We assume *qualitative* preferences over soft constraints: e.g., “produce more than 500 MW” is more important than “change the output no faster than 5 MW in 15 minutes” elicited from an operator’s experience. Both constraints may be dropped if necessary.

An AVPP is then composed of a set of CSPs for each of its subordinate power plants and has to create feasible schedules to meet the given output demand as accurately as possible. In addition, it should consider the individual preferences specified by each power plant over its own decision variables. Assume, e.g., an AVPP A composed of two physical power plants, p and q. We regard the *output* of each plant for *two* time steps from $\mathcal{T} = \{1, 2\}$ and thus have the variables $\{a[t] \mid a \in \{p, q\}, t \in \mathcal{T}\}$ with domains $D_p = \{0, 1, \dots, 10\}$, $D_q = \{0, 1, \dots, 20\}$ for $t \in \mathcal{T}$, where we abbreviate $D_{a[t]}$ by D_a since we have identical domains for all time steps. Constraints include soft, yet important, organizational constraints $\text{org}[t] : p[t] + q[t] = \mathcal{D}[t]$ for each $t \in \mathcal{T}$, where $\mathcal{D}[t]$ denotes the *demand* at time step t which is given as a constant. Individual power plant models provide additional constraints:

- 1) Power plants prefer economically better subranges.
- 2) Rates of change may be further limited.

Assume, for power plant p, that the best range was $\{7, 8\}$, followed by a good range $\{5, \dots, 9\}$ and an acceptable range $\{4, \dots, 10\}$. Values in $\{0, \dots, 3\}$ are technically feasible but less desirable for p. We denote constraints representing these ranges by $\text{best}[t]$, $\text{good}[t]$, and $\text{acc}[t]$ with $t \in \mathcal{T}$. All qualitative constraint preference statements are captured in the *constraint relationship* C of p: $\text{acc}[t] \rightarrow_C \text{good}[t] \rightarrow_C \text{best}[t]$ for $t \in \mathcal{T}$. Here, $c \rightarrow_C c'$ indicates that constraint c “precedes” c' in C, thus c' is “more important” in C than c .

The other power plant, q, prefers to run *either* in a low range $\{1, \dots, 5\}$ (constraints $\text{low}[t]$ for $t \in \mathcal{T}$) *or* in a high range $\{15, \dots, 20\}$ (constraints $\text{high}[t]$), but most importantly wants to avoid high changes. This is reflected by the constraint $\text{diff} : |q[1] - q[2]| \leq 5$. There is however no preference among the two subranges.

Enhancing Existing Soft Constraint Approaches

When combining a set of independently modeled preferences, i.e., p and q into A in our case study, we need to define how to compare constraints originating from different power plants to address questions like “Is a solution violating only p’s constraints better than one violating only q’s constraints?”. More concretely, if we take the union of the constraint relationships and map the resulting problem to weighted CSPs (or any quantitative, totally ordered formalism), constraints from different power plants become comparable even though no relative importance was specified between them. Figure 2 shows this for the exemplary constraint relationships (constraints for economical sub-ranges exist for both time steps and we consider $\text{org}[1]$ and $\text{org}[2]$ to be more important than, e.g., $\text{best}[1]$ and $\text{best}[2]$ individually). We highlight our analysis by focusing on time step 1. Given that the demand is 14, we have to violate $\text{high}[1]$ (which would force $q[1] \geq 15$) in order to satisfy $\text{org}[1]$. Assume two assignments $v_1 = \{p[1] \mapsto 9, q[1] \mapsto 5\}$ and $v_2 = \{p[1] \mapsto 8, q[1] \mapsto 5\}$. Thus, v_1 violates $\{\text{best}[1], \text{high}[1]\}$ and v_2 violates $\{\text{org}[1], \text{high}[1]\}$, leading to a preference of v_1 as it violates a less important constraint.

$X = \{a[t] \mid a \in \{p, q\}, t \in \mathcal{T}\}$, $D = \langle 14, 14 \rangle$
 $D_p = \{0, \dots, 10\}$, $D_q = \{0, \dots, 20\}$, $D_A = \{0, \dots, 30\}$

- Organizational: $\text{org}[t] : \mathcal{D}[t] = p[t] + q[t]$
- Power plant p: $\text{best}[t] : p[t] \in \{7, 8\}$
 $\text{good}[t] : p[t] \in \{5, \dots, 9\}$
 $\text{acc}[t] : p[t] \in \{4, \dots, 10\}$
- Power plant q: $\text{diff} : |q[1] - q[2]| \leq 5$
 $\text{low}[t] : q[t] \in \{1, \dots, 5\}$
 $\text{high}[t] : q[t] \in \{15, \dots, 20\}$

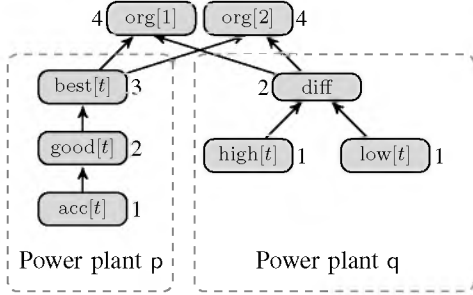


Figure 2. Case study with domains, constraints and combined constraint relationship C with annotated possible weights.

When using weighted CSPs [5], the constraints $\text{best}[t]$ are implicitly weighted higher than diff (since p needs to distinguish $\text{good}[t]$ from $\text{acc}[t]$) even though both are modeled more important than two other constraints and have no explicit notion of importance relative to each other. With constraint hierarchies, one could design a hierarchy level including $\text{high}[t]$, $\text{low}[t]$, and $\text{best}[t]$ with $\text{acc}[t]$ on the next level. But there is no obvious reason that $\text{high}[t]$ or $\text{low}[t]$ (stemming from power plant q) should be considered more important than $\text{acc}[t]$ (from the other power plant, p). Alternatively, we could categorize $\text{good}[t]$ and $\text{acc}[t]$ on the same hierarchy level and weigh them accordingly on that level – forcing again a total order.

Consequently, we search a construction of a c-semiring that only adds the *required* structure enforced by a preference dag and nothing more – a *free* construction [18].

III. CONSTRAINT RELATIONSHIPS

Based on our case study, we now summarize constraint programming with constraint relationships on a more formal level. We restate the notion of constraint relationships originally given in [8], starting with classical constraint satisfaction.

A *constraint domain* (X, D) is given by a finite set X of *variables* and a family $D = (D_x)_{x \in X}$ of *variable domains* where each D_x is a finite 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 $[X \rightarrow D]$. A *constraint* c over a constraint domain (X, D) , or (X, D) -*constraint*, is given by a map $c : [X \rightarrow D] \rightarrow \mathbb{B}$. We also write $v \models c$ for $c(v) = \text{tt}$. In our case study, we have used the constraint domain (X, D) with $X = \{a[t] \mid a \in \{p, q\}, t \in \mathcal{T}\}$ and $D_x = D_a$ for $x = a[t]$. An assignment $v = \{p[t] \mapsto 5, q[t] \mapsto 0 \mid t \in \mathcal{T}\}$ violates $\text{best}[t]$, $\text{low}[t]$, and $\text{high}[t]$ for $t \in \mathcal{T}$ if $D = \langle 5, 5 \rangle$.

A *directed acyclic graph*, or *dag*, $G = (|G|, \rightarrow_G)$ is given

by an *underlying set*¹ $|G|$ and a binary relation $\rightarrow_G \subseteq |G| \times |G|$ such that \rightarrow_G^+ is irreflexive. If $x \rightarrow_G y$, then x is a *predecessor* of y , and y is a *successor* of x . We obtain a partial order $PO(G) = (|G|, \rightarrow_G^*)$ from G by taking the reflexive, transitive closure of \rightarrow_G , and write $g \leq_{PO(G)} h$ if $g \rightarrow_G^* h$.

A *constraint relationship* over a constraint domain (X, D) , or (X, D) -*constraint relationship*, is given by a dag C , where $|C|$ is a finite set of (X, D) -constraints. We think of a constraint $c \in |C|$ if $c \rightarrow_C c'$. For two sets $V, W \subseteq |C|$, which we think of being sets of *violated* constraints representing (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 order induced by the dag C to an order over subsets of $|C|$, represented by two *dominance properties* π : *single-predecessor dominance* ($\pi = \text{SPD}$) and *transitive-predecessors dominance* ($\pi = \text{TPD}$, note that this dominance corresponds to the hierarchical idea illustrated in Fig. 1); we write $V \rightsquigarrow_C^\pi W$ for “ W is worse than V for dominance property π 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):

$$V \rightsquigarrow_C^\pi V \uplus \{c\} \quad \text{if } c \in |C| \quad (\text{W})$$

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

$$V \uplus \{c\} \rightsquigarrow_C^{\text{SPD}} V \uplus \{c'\} \quad \text{if } c \rightarrow_C c' \quad (\text{SPD})$$

$$V \uplus \{c_1, \dots, c_k\} \rightsquigarrow_C^{\text{TPD}} V \uplus \{c'\} \quad \text{if } \forall i. c_i \rightarrow_C^+ c' \quad (\text{TPD})$$

These worsening relations induce partial orders \leq_C^π over sets of (violated) constraints for $\pi \in \{\text{SPD}, \text{TPD}\}$, when defining $W <_C^\pi V$ if, and only if, $V (\rightsquigarrow_C^\pi)^+ W$ (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. By abuse of notation, for assignments we also write $w <_C^\pi v$ if $\{c \in |C| \mid w \not\models c\} <_C^\pi \{c \in |C| \mid v \not\models c\}$, again to be read as “ w is worse than v ”.

In our example, this partial order does not introduce the bias shown in Fig. 2 between $\text{best}[t]$ and diff since there is no edge between those two constraints. Consider two assignments v and w with $v \not\models \text{diff}$ and $w \not\models \text{best}[1]$. Using weights, v would be valued with 2 and w with 3, and thus $w <_{\text{WCSP}} v$ (meaning “ w is worse than v ” in weighted CSP) whereas neither $v <_C^\pi w$ nor $w <_C^\pi v$ holds.

IV. FROM CONSTRAINT RELATIONSHIPS TO C-SEMRINGS

The embedding of constraint relationships into the established soft constraint framework of c-semirings using the special c-semiring of weighted CSP [8] has the drawback of forcing a total order on the assignments. As illustrated in Sect. II,

¹We write $|G|$ in reminiscence of a *forgetful functor*, mapping an object G in a *concrete* category to its underlying set.

assignments become comparable which would not be deemed comparable considering solely the constraint relationships. To overcome this bias, while still keeping the connection with c-semirings, we search for the “most general” c-semiring that preserves the orderings prescribed by the constraint relationships. This c-semiring should not show more orderings than are required by the constraint relationships and the c-semiring axioms. This task amounts to constructing the *free* c-semiring over a partial order (from a constraint relationship) in the sense of universal algebra [18, Ch. 4] and category theory [19, Ch. 3].

Categorically speaking, given two categories \mathcal{A} and \mathcal{B} and a functor $U : \mathcal{B} \rightarrow \mathcal{A}$, the *free object* $F(A)$ in \mathcal{B} over an object A of \mathcal{A} is characterized by a *unit* morphism $\eta_A : A \rightarrow U(F(A))$ in \mathcal{A} such that for every \mathcal{A} -morphism $f : A \rightarrow U(B)$ with B an object of \mathcal{B} , there is a unique *lifting* \mathcal{B} -morphism $f^\sharp : F(A) \rightarrow B$ satisfying $U(f^\sharp) \circ \eta_A = f$; the free object $F(A)$ is unique up to isomorphism [19, Ch. 3].

It turns out to be advantageous to perform the free construction from partial orders to c-semirings via an intermediate, again freely constructed, type of algebraic structure, the *partial valuation structures* by Gadducci et al. [14], where the total ordering of valuation structures [11] is relaxed to a partial ordering. We first recapitulate the notion of partial valuation structures and describe the free partial valuation structure over a partial order. Then we recall the notion of c-semirings and describe the free c-semiring over a partial valuation structure. By combining these free constructions, we obtain the free c-semiring over a partial order and thus the connection to constraint relationships. The proofs that the free partial valuation structure and the free c-semiring are indeed the correct notions can be found in the accompanying technical report [20].

A. The Free Partial Valuation Structure Over a Partial Order

Like valuation structures, partial valuation structures [14] capture essential operations for grading assignments: They show an associative and 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. In contrast to valuation structures, the ordering is not forced to be total. In contrast to c-semirings, partial valuation structures do, in general, not show the suprema required by c-semiring addition.

Constraint relationships and the induced orderings of Sect. III lifting the dag to sets of (violated) constraints apparently fit quite naturally to such a monoidal structure, where the multiplication should be the union. 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 the exemplary constraint relationship C in Fig. 2 including the constraints $\text{org}[2]$ and diff with $\text{diff} \rightarrow_C \text{org}[2]$. Then $\{\text{org}[2]\} <_C^{\text{SPD}} \{\text{diff}\}$. Multiplying on both sides with $\{\text{org}[2]\}$, i.e., taking the union, would result in $\{\text{org}[2]\} \leq_C^{\text{SPD}} \{\text{org}[2], \text{diff}\}$ by the required monotonicity of the multiplication. Hence, violating $\text{org}[2]$ only would be worse than violating both $\text{org}[2]$ and diff , contradicting (W). However, we can patch this defect by not considering sets and their union but multisets and the multiset union; the disjointness assumptions in (SPD) and (TPD) indicate this necessity.

Incidentally, when equipping multisets with an appropriate ordering induced by the partial order from the constraint relationship, the free partial valuation structure is obtained.

Formally, a *partial valuation structure* $M = (|M|, \cdot, \varepsilon, \leq)$ is given by an *underlying set* $|M|$, an associative and commutative *multiplication* operation $\cdot : |M| \times |M| \rightarrow |M|$, a *neutral element* $\varepsilon \in |M|$ for \cdot , and a partial ordering $\leq \subseteq |M| \times |M|$ 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 . Requiring that ε is top is equivalent to requiring that $m_1 \cdot m_2 \leq m_1$. We add subscripts to \cdot, ε and \leq when different partial valuation structures arise. A *partial valuation structure morphism* $\varphi : M \rightarrow N$ is given by a structure-preserving function $\varphi : |M| \rightarrow |N|$, i.e., $\varphi(m_1 \cdot_M m_2) = \varphi(m_1) \cdot_N \varphi(m_2)$, $\varphi(\varepsilon_M) = \varepsilon_N$, and $\varphi(m_1) \leq_N \varphi(m_2)$ if $m_1 \leq_M m_2$.

Each partial valuation structure M induces a partial order $PO(M) = (|M|, \leq_M)$ and each partial valuation structure morphism $\varphi : M \rightarrow N$ induces a partial order morphism (i.e., an ordering-preserving or monotone function) $PO(\varphi) = \varphi : PO(M) \rightarrow PO(N)$. With this functor PO from partial valuation structures to partial orders, we now consider the free partial valuation structure $PVS\langle P \rangle$ over a partial order P .

We denote the set of finite multisets over the underlying set $|P|$ of P by $\mathcal{M}_{\text{fin}} |P|$, its elements by $\{p_1, \dots, p_k\}$, and multiset union by \cup . We define the *upper* or *Smyth ordering* on $\mathcal{M}_{\text{fin}} |P|$ as the binary relation $\subseteq^P \subseteq (\mathcal{M}_{\text{fin}} |P|) \times (\mathcal{M}_{\text{fin}} |P|)$ given by the transitive closure of

$$\begin{aligned} T \supseteq U \text{ implies } T &\subseteq^P U, \\ p \leq_P q \text{ implies } T \cup \{p\} &\subseteq^P T \cup \{q\}. \end{aligned}$$

This relation is indeed a partial ordering on $\mathcal{M}_{\text{fin}} |P|$ and it is easy to check that $PVS\langle P \rangle = (\mathcal{M}_{\text{fin}} |P|, \cup, \{ \}, \subseteq^P)$ is a partial valuation structure. Moreover, $PVS\langle P \rangle$ is the *free* partial valuation structure over the partial order P : Define the *unit* partial order morphism $\eta_P^{\text{PVS}} : P \rightarrow PO(PVS\langle P \rangle)$ by $\eta_P^{\text{PVS}}(p) = \{p\}$. Then every partial order morphism $\varphi : P \rightarrow PO(M)$ for some partial valuation structure M can be uniquely *lifted* into a partial valuation structure morphism $\varphi^{\sharp \text{PVS}} : PVS\langle P \rangle \rightarrow M$ with $\varphi^{\sharp \text{PVS}}(\{p_1, \dots, p_k\}) = \varphi(p_1) \cdot_M \dots \cdot_M \varphi(p_k)$ such that $PO(\varphi^{\sharp \text{PVS}})(\eta_P^{\text{PVS}}(p)) = \varphi(p)$. Thus, we have (for a detailed proof, see [20, §12]):

PROPOSITION. *Let P be a partial order. Then $PVS\langle P \rangle = (\mathcal{M}_{\text{fin}} |P|, \cup, \{ \}, \subseteq^P)$ is the free partial valuation structure over P .*

The upper ordering on finite multisets mimics the eponymous ordering used in powerdomain constructions [21, Ch. 9]. There, partial orders are lifted to semi-lattices where multiplication is idempotent. On the other hand, our upper ordering, when employed for sets, exactly corresponds to $\leq_{C^{-1}}^{\text{SPD}}$ 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 – e.g., $\text{diff} \leq_{PO\langle C \rangle} \text{org}[2]$ but $\{\text{org}[2]\}$ should be worse than $\{\text{diff}\}$. Further reconsidering our discussion based on the constraint relationship C presented in Fig. 2 with $\text{diff} \rightarrow_C \text{org}[2]$, we find that multiplying on both sides with $\{\text{org}[2]\}$ by now taking the multiset union, which is not idempotent, results in $\{\text{org}[2], \text{org}[2]\} \leq_{PVS\langle PO\langle C^{-1} \rangle \rangle}$

$\{\text{org}[2], \text{diff}\}$. This relation clearly holds, but the other direction does not as $\{\text{org}[2], \text{org}[2]\}$ does not worsen to $\{\text{org}[2], \text{diff}\}$ since we cannot “trade” the remaining occurrence of $\text{org}[2]$ for diff . However, the transitive-predecessors dominance presented in (TPD) can only be achieved by using a more specialized ordering (see Sect. V-B).

The free partial valuation structure over a partial order P does not show suprema of finite multisets, in general: Consider the partial order P given by $(\{g, h, i, j\}, \{g < h, g < i, h < j, i < j\})$, which has suprema. In $PVS(P)$, however, we have

$$\begin{aligned} \{g\} \subseteq^P \{h\} \text{ and } \{g\} \subseteq^P \{i\}, \\ \{h, i\} \subseteq^P \{h\} \text{ and } \{h, i\} \subseteq^P \{i\} \end{aligned}$$

and no $T \in \mathcal{M}_{\text{fin}}[P]$ exists with $\{g\}, \{h, i\} \subseteq^P T \subseteq^P \{h\}, \{i\}$ since, e.g., for $\{g\} \subseteq^P T$, T can only be $\{g\}$ by the first rule (with $\{g\}, \{h, i\} \subseteq^P \{g\}$), or $\{h\}, \{i\}$, or $\{j\}$ by the second rule (where $\{h\}, \{i\} \subseteq^P \{j\}$); but $\{h\}$ and $\{i\}$ are incomparable w.r.t. \subseteq^P . We therefore need to introduce additional structure to obtain a c-semiring which requires suprema.

B. The Free C-Semiring Over a Partial Valuation Structure

A c-semiring [10] $A = (|A|, \oplus, \otimes, 0, 1)$ is given by an underlying set $|A|$, a binary addition $\oplus : |A| \times |A| \rightarrow |A|$, a binary multiplication $\otimes : |A| \times |A| \rightarrow |A|$, a zero $0 \in |A|$, and a one $1 \in |A|$ such that

- 1) \oplus is associative and commutative, has 1 as annihilator and 0 as neutral element;
- 2) \otimes is associative and commutative, has 0 as annihilator and 1 as neutral element;
- 3) \otimes distributes over \oplus .

In particular, \oplus is idempotent (since $a \oplus a = (a \otimes 1) \oplus (a \otimes 1) = a \otimes (1 \oplus 1) = a \otimes 1 = a$). Thus the relation $\leq \subseteq |A| \times |A|$ defined by $a_1 \leq a_2$ iff $a_1 \oplus a_2 = a_2$ is a partial ordering on $|A|$, i.e., reflexive, transitive, and antisymmetric. In fact, $a_1 \oplus a_2$ yields the supremum of a_1 and a_2 with respect to \leq . Moreover, 0 is the least and 1 the greatest element in $|A|$ w.r.t. \leq . A c-semiring morphism $\varphi : A \rightarrow B$ from a c-semiring A to a c-semiring B is given by a structure-preserving function $\varphi : |A| \rightarrow |B|$, i.e., $\varphi(a_1 \oplus a_2) = \varphi(a_1) \oplus \varphi(a_2)$, $\varphi(a_1 \otimes a_2) = \varphi(a_1) \otimes \varphi(a_2)$, $\varphi(0_A) = 0_B$, and $\varphi(1_A) = 1_B$.

An arbitrary c-semiring A gives rise to the partial valuation structure $PVS(A) = (|A|, \otimes_A, 1_A, \leq_A)$, and a c-semiring morphism $\varphi : A \rightarrow B$ induces a partial valuation structure morphism $PVS(\varphi) = \varphi : PVS(A) \rightarrow PVS(B)$. With this functor PVS from c-semirings to partial valuation structures, we now consider the free c-semiring $cSRng\langle M \rangle$ over a partial valuation structure M .

To obtain a valid \oplus -operation yielding suprema that do not necessarily exist in a partial valuation structure, we move from the set $|M|$ to the finite subsets of $|M|$ only containing pairwise incomparable elements w.r.t. \leq_M . We denote these by $\mathcal{I}_{\text{fin}}^{\leq M}(|M|)$. Define $cSRng\langle M \rangle = (\mathcal{I}_{\text{fin}}^{\leq M}[M], \tilde{\cup}_M, \tilde{\cdot}_M, \emptyset, \{\varepsilon_M\})$ with

$$\begin{aligned} I_1 \tilde{\cdot}_M I_2 &= \text{Max}^{\leq M} \{m_1 \cdot_M m_2 \mid m_1 \in I_1, m_2 \in I_2\}, \\ I_1 \tilde{\cup}_M I_2 &= \text{Max}^{\leq M} (I_1 \cup I_2). \end{aligned}$$

It can be shown that $cSRng\langle M \rangle$ is indeed a c-semiring (see [20, §29]). Moreover, it is the free c-semiring over M : We have the partial valuation structure morphism $\eta_M^{cSRng} : M \rightarrow PVS(cSRng\langle M \rangle)$ with $\eta_M^{cSRng}(m) = \{m\}$, serving as a unit, such that if A is an arbitrary c-semiring and $\varphi : M \rightarrow PVS(A)$ some partial valuation structure morphism, then there is the unique lifting c-semiring morphism $\varphi^{\sharp cSRng} : cSRng\langle M \rangle \rightarrow A$ defined by $\varphi^{\sharp cSRng}(\{m_1, \dots, m_k\}) = \varphi(m_1) \oplus_A \dots \oplus_A \varphi(m_k)$ such that $\eta_M^{cSRng} \circ PVS(\varphi^{\sharp cSRng}) = \varphi$ is satisfied. We thus obtain (for a detailed proof, see [20, §29])

PROPOSITION. *Let M be a partial valuation structure. Then $cSRng\langle M \rangle = (\mathcal{I}_{\text{fin}}^{\leq M}[M], \tilde{\cup}_M, \tilde{\cdot}_M, \emptyset, \{\varepsilon_M\})$ is the free c-semiring over M .*

This construction of the free c-semiring over a partial valuation structure shares the definition of addition and multiplication with the set-based encoding of “preference degree structures” as a semiring by Fargier et al. [22, Sect. 4.3]. However, in order to obtain a c-semiring proper (rather than a plain semiring), we had to rely on the neutral element being the top element for the ordering. A similar construction has been suggested by Bistarelli et al. [23], although starting from a c-semiring. The freeness of the resulting (c-)semiring (which corresponds to the lower or Hoare powerdomain construction [21, Ch. 9]) has not been considered by either previous works.

Reconsidering our previous counterexample for the existence of suprema in partial valuation structures, no supremum for $\{g\}$ and $\{h, i\}$ exists in $PVS(P)$ but in the free c-semiring $cSRng\langle PVS(P) \rangle$, the supremum of $\{\{g\}\}$ and $\{\{h, i\}\}$ is $\{\{g\}, \{h, i\}\}$. Indeed, $\{\{g\}, \{h, i\}\} \leq_{cSRng\langle PVS(P) \rangle} \{\{x\}\}$ holds for both $x = h$ and $x = i$. Considering $\{\{h\}\}$, we get $\{\{g\}, \{h, i\}\} \tilde{\cup}_{cSRng\langle PVS(P) \rangle} \{\{h\}\} = \text{Max}^{\leq_P}(\{\{g\}, \{h, i\}, \{h\}\}) = \{\{h\}\}$ since $\{g\} \subseteq^P \{h\}$ and $\{h, i\} \subseteq^P \{h\}$ – analogously for $\{\{i\}\}$.

The composition of two free constructions again yields a free construction [19]. Thus we obtain the free c-semiring over a partial order P as $cSRng\langle PVS\langle P \rangle \rangle$. For a constraint relationship C this means to consider $cSRng\langle PVS\langle PO\langle C^{-1} \rangle \rangle \rangle$.

V. SOLVING CONSTRAINT RELATIONSHIP PROBLEMS

Solving a c-semiring based soft constraint satisfaction problem generally results in a set of several optimal solutions due to the partiality of the induced order. We show how to apply the well-known branch-and-bound algorithm (see, e.g., [24], [12]) to soft-constraint problems specified using partial valuation structures as this algorithm does not rely on the \oplus -operation of a c-semiring (such as, e.g., dynamic programming based approaches [10]) but works directly with the induced order. Consequently, we can use the free partial valuation structure from a constraint relationship to obtain a solver for constraint relationship problems.

An M -soft constraint over a constraint domain (X, D) for a partial valuation structure M , or (X, D) - M -soft constraint, is given by a map $\mu : [X \rightarrow D] \rightarrow |M|$. For a constraint relationship C over (X, D) and a partial valuation structure M , we convert each $c \in |C|$ into an M -soft constraint via an embedding partial order morphism $\varphi : PO\langle C^{-1} \rangle \rightarrow PO(M)$:

We define $c_{M,\varphi} : [X \rightarrow D] \rightarrow |M|$ by

$$c_{M,\varphi}(v) = \begin{cases} \varphi(c) & \text{if } v \not\models c \\ \varepsilon_M & \text{otherwise} \end{cases}.$$

We leave φ as a parameter of $c_{M,\varphi}$ to allow for different (monotonic) representations of a constraint as element of $|M|$ (e.g., a violated constraint could embed itself along with its predecessors to accommodate alternative assignment orderings). Inverting the constraint relationship C is motivated by the upper (Smyth) ordering used in the free partial valuation structure (see Sect. IV-A). In particular, we can define the embedding of C into the free partial valuation structure $PVS\langle PO\langle C^{-1} \rangle \rangle = (\mathcal{M}_{\text{fin}}[C], \sqcup, \sqcap, \sqsubseteq^{PO\langle C^{-1} \rangle})$ by $\varphi(c) = \eta_{PO\langle C^{-1} \rangle}^{\text{PVS}}(c) = \sqcap c$ – corresponding to the *single-predecessor dominance lifting*.

Let M be a finite set of (X, D) - M -soft constraints. For an assignment $v \in [X \rightarrow D]$ the *solution degree* for M of v , written as $M(v)$, is obtained by combining the gradings with respect to \cdot_M , i.e., $M(v) = \prod_M \{\mu(v) \mid \mu \in M\}$. Then the *maximum solution degrees* of M are given by

$$M^* = \text{Max}^{\leq_M} \{M(v) \mid v \in [X \rightarrow D]\}.$$

A. Branch and Bound

For the branch-and-bound algorithm for computing the maximum solution degrees over a partial valuation structure, we now move from assignments to partial assignments. For a constraint domain (X, D) we fix an *extended* constraint domain $(X, D^?)$ setting $D^? = (D_x^?)_{x \in X}$ with $D_x^? = D_x \sqcup \{?\}$, where $?$ is fresh, to represent a *partial assignment* which is a dependent map $p \in \Pi x \in X. D_x^? = [X \rightarrow D^?]$. The *domain of definition* $\text{def}(p)$ of a partial assignment p for (X, D) is the set $\{x \in X \mid p(x) \neq ?\}$. For $p, q \in [X \rightarrow D^?]$, we write $p \sqsubseteq q$ if $x \in \text{def}(p)$ implies $x \in \text{def}(q)$ and $q(x) = p(x)$ for each $x \in X$; by $p \uparrow$ we denote the set $\{v \in [X \rightarrow D] \mid p \sqsubseteq v\}$ of (X, D) -assignments. The *scope* of a constraint c over a constraint domain (X, D) is given by the set of variables it depends on, i.e.,

$$\text{sc}(c) = \{x \in X \mid \exists v \in [X \rightarrow D], d_1 \neq d_2 \in D_x. \\ c(v\{x \mapsto d_1\}) \neq c(v\{x \mapsto d_2\})\}.$$

For a partial assignment $p \in [X \rightarrow D^?]$, we write $p \not\models c$ if $\text{sc}(c) \subseteq \text{def}(p)$ and $v \not\models c$ for some $v \in p \uparrow$ (which is well defined, since then c only depends on variables that are in the domain of definition of p).

For our algorithm, we use a “best case/worst case” *bounding pair* that maps to the best (worst) possible solution degree of a partial assignment. In branch-and-bound, we usually only need a best case to compare it to already found solution degrees. If, however, a *worst-case* estimation of extending a $p \in [X \rightarrow D^?]$ by $p\{x \mapsto d_1\}$ is *better* than the *best-case* estimation for $p\{x \mapsto d_2\}$, we should prune the path following $p\{x \mapsto d_2\}$ before finding a (complete) assignment.

Formally, a *bounding pair* (α, ζ) for a finite set of (X, D) - M -soft constraints M is given by two maps $\alpha, \zeta : [X \rightarrow D^?] \rightarrow |M|$, with α representing the worst case and ζ the best case estimate satisfying

- 1) $\alpha(p) \leq_M M(v) \leq_M \zeta(p)$ for all $p \in [X \rightarrow D^?]$ and $v \in p \uparrow$;
- 2) $M(v) = \zeta(v)$ for all $v \in [X \rightarrow D]$.

Concretely, for a constraint relationship C and an embedding φ , we define $\alpha_{M,\varphi}, \zeta_{M,\varphi} : [X \rightarrow D^?] \rightarrow |M|$ by

$$\alpha_{M,\varphi}(p) = \prod_M \{\varphi(c) \mid c \in |C|, \text{sc}(c) \subseteq \text{def}(p), p \not\models c\} \cdot_M \\ \prod_M \{\varphi(c) \mid \text{sc}(c) \not\subseteq \text{def}(p)\}, \\ \zeta_{M,\varphi}(p) = \prod_M \{\varphi(c) \mid c \in |C|, \text{sc}(c) \subseteq \text{def}(p), p \not\models c\}.$$

Intuitively, $\alpha_{M,\varphi}$ represents the worst case by assuming all partially defined constraints to be violated, and thus satisfies the first part of condition (1) for bounding pairs. Analogously, $\zeta_{M,\varphi}$ idealizes that no additional constraints will be violated, thus satisfying the second part of (1) and (2).

The following branch-and-bound algorithm for partial valuation structures using bounding pairs finds the maximum solution degrees of a soft constraint problem given a partial assignment and a set of known lower bounds:

Assume: – (X, D) constraint domain
– M partial valuation structure
– M finite set of (X, D) - M -soft constraints
– (α, ζ) bounding pair for M
In: – $p \in [X \rightarrow D^?]$ partial valuation for (X, D)
– $L \subseteq |M|$ finite and pairwise incomparable w.r.t. \leq_M
Return: $\text{Max}^{\leq_M} (L \cup \{M(v) \mid v \in p \uparrow\})$

maxSolDeps $_{(\alpha,\zeta)}(p, L) \equiv$
if $\forall x \in X. p(x) \neq ?$ — Is p already an assignment?
then return $\text{Max}^{\leq_M} (L \cup \{M(p)\})$ **fi**
 $x \leftarrow \text{choose } \{x \in X \mid p(x) = ?\}$
— Add worst-case estimates
 $L \leftarrow \text{Max}^{\leq_M} (L \cup \{\alpha(p\{x \mapsto d\}) \mid d \in D_x\})$
for $d \in D_x$ — Test against best-case estimates
do if $\neg \exists l \in L. \zeta(p\{x \mapsto d\}) \leq_M l$
then $L \leftarrow \text{maxSolDeps}_{(\alpha,\zeta)}(p\{x \mapsto d\}, L)$ **fi od**
return L

Starting the algorithm with a partial assignment mapping all variables to $?$ and the empty set of known lower bounds, we get from the return assertion that

$$\text{maxSolDeps}_{(\alpha,\zeta)}(\lambda x \in X. ?, \emptyset) = \\ \text{Max}^{\leq_M} \{M(v) \mid v \in [X \rightarrow D]\} = M^*.$$

The correctness proof can be found in [20, §34, §36].

B. Extensions to the Free Partial Valuation Structure

Practice suggests additional structure to the free partial valuation structure. First, constraint relationships only consider *soft* constraints by design. The free partial valuation structure $PVS\langle PO\langle C^{-1} \rangle \rangle$ does not offer an element for complete dissatisfaction, i.e., $\sqsubseteq^{PO\langle C^{-1} \rangle}$ has no smallest element.

Using the terminology of [14], a partial valuation structure M is *bounded* if $|M|$ has a smallest element w.r.t. \leq_M ; we denote this element by \perp_M if it exists. In a bounded partial valuation structure M it holds that $m \cdot_M \perp_M = \perp_M$ for all $m \in |M|$, i.e., \perp_M is an absorbing element. Each partial valuation

structure M which is not bounded can be *lifted* into a bounded partial valuation structure [14].

Consequently, we can use \perp_M to represent the violation of hard constraints in a bounded partial valuation structure M by using $c_{M,\varphi}^s = c_{M,\varphi}$ for all soft constraints and $c_{M,\varphi}^h$ for the conversion of hard constraints defined as $c_{M,\varphi}^h(v) = \perp_M$ if $v \not\models c$ and $c_{M,\varphi}^h(v) = \varepsilon_M$ otherwise.

Furthermore, for a subclass of constraint hierarchies [8], we also consider the *transitive-predecessors dominance lifting* $\rightsquigarrow_C^{\text{TPD}} \subseteq (\mathcal{M}_{\text{fin}}|C|) \times (\mathcal{M}_{\text{fin}}|C|)$ of C to the finite multisets $\mathcal{M}_{\text{fin}}|C|$ over the elements of C , given by

$$T \rightsquigarrow_C^{\text{TPD}} T \cup \{c\}, \\ c_1, \dots, c_n \rightarrow_C^+ c \text{ implies } T \cup \{c_1, \dots, c_n\} \rightsquigarrow_C^{\text{TPD}} T \cup \{c\},$$

which is in one-to-one correspondence to the rules (W) and (TPD) in Sect. III. Writing \leq_C^{TPD} for $((\rightsquigarrow_C^{\text{TPD}})^*)^{-1}$ and using the fact that \leq_C^{TPD} is monotonic w.r.t. multiset union, we have that $(\mathcal{M}_{\text{fin}}|C|, \cup, \bigcup, \leq_C^{\text{TPD}})$ is a partial valuation structure. Using the same embedding φ leaves us with a TPD-constraint relationship problem.

C. Variable Ordering and Local Consistency Heuristics

It is well known that the success of systematic tree search methods is greatly influenced by the variable and value ordering and several heuristics have been devised for classical constraint problems [25]. This not the case for weighted CSPs or soft constraint problems in general. Successful heuristics typically take into account the domain cardinality or the effect assignments make [26]. We propose heuristics for the investigated soft-constraint problems over constraint relationships:

- 1) *Most important first* (MIF) variable ordering: A topological ordering of a constraint relationship lists most important constraints first and assigns values to their variables first.
- 2) *Local consistency* (LC): We improve lower and upper bounds inspired by soft-arc consistency [27]. For each partially assigned (binary) constraint, we include constraints that are violated irrespective of the value of the remaining variable into the best-case ζ and exclude those that are certainly true from the worst-case α .

VI. EVALUATION

In a first test-bed, we implemented the branch-and-bound algorithm presented in Sect. V in Java. For replicability of our experiments, the source code is available online.² The solver is only in a prototypical state and serves to investigate the influence of the proposed heuristics and modeling aspects instead of being directly applied to the (in reality more complex) problems presented in our case study. In particular, we wanted to compare our heuristics to a naïve branch-and-bound implementation that is intractable for a larger number of variables.

We test several parameters regarding algorithmics (variable orderings, local consistency, and bounding pairs) and modeling (using SPD or TPD semantics) on randomly generated CSP instances and measure running time as well as the number of recursive branch-and-bound calls. Constraints are simple

Table I. Modeling influence: Comparison of mean runtimes in seconds and recursive calls for SPD vs. TPD and weighted CSP vs. constraint relationship-based CSP with standard deviations given in parentheses.

	WCSP-RT	WCSP-RC	CRCSP-RT	CRCSP-RC
SPD	0.86 (2.79)	7183.78 (22617.88)	2.08 (5.77)	13323.36 (32050.35)
TPD	0.83 (2.81)	6888.38 (22622.73)	2.00 (5.73)	12721.54 (31996.98)

arithmetical expressions including $+$, $-$, $*$, $/$, and $<$, \leq , $=$, \geq , $>$ as relational operators. In each experiment, n random problems were solved by differently parametrized solvers.

A. Modeling Influence

First, we investigate how the selected dominance property and choice of partial valuation structure (using constraint relationships directly or weights according to [8]) affect the solver's efficiency in order to select the formalism for which to evaluate the proposed heuristics in more detail. We fixed the used bounding pairs (as defined in Sect. V-A), the variable ordering to use MIF (most important first), and the usage of a local consistency check (LC). We compare weighted CSPs (WCSP) and constraint relationship-based CSPs (CRCSP) with regard to runtime (RT) and recursive calls (RC). The concrete weights for a constraint relationship C are computed recursively, as suggested in [8], by

$$w_C^{\text{SPD}}(c) = 1 + \max\{w_C^{\text{SPD}}(c') \mid c' \in C : c' \rightarrow_C c\}, \\ w_C^{\text{TPD}}(c) = 1 + \sum_{c' \in C : c' \rightarrow_C c} (2 \cdot w_C^{\text{TPD}}(c') - 1).$$

This experiment consists of 50 runs with 7 variables having domains of 5 to 15 values and 10 constraints arranged in a random constraint relationship.

As expected, Table I shows that weighted CSPs require less runtime and recursive calls to find the maximum solution degrees. This is mostly due to the presence of incomparable elements in constraint relationships which reduce the possible pruning and to the more complicated evaluation of the upper (Smyth) ordering. However, the stronger dominance property TPD only improved the pruning by about 5% in the constraint relationship case and 4% in the weighted CSP case. This is still a considerable gain but does not completely rule out SPD in terms of performance. Consequently, we focus on weighted CSP instances to examine the proposed solver heuristics.

B. Algorithmic Efficiency

With a weighted CSP based on constraint relationships and TPD semantics, we investigate how variable ordering using the most-important-first heuristic, bounding pairs, as well as local consistency affect the performance. The CSPs for this experiment had 6 variables, 10 constraints and domains with 10 to 30 values, and 150 runs were made. We compare values for random order (RO) vs. most-important-first (MIF) with flags for worst-case boundaries (WCB) and local consistency (LC).

For clarity, we discuss the number of recursive calls to measure the pruning performance of the algorithm. Table II clearly shows that using the most-important-first heuristic significantly improves the performance. Similar improvements can be achieved using local consistency checks for improved

²<https://github.com/Alexander-Schiendorfer/constraint-relationships-csemiring>

Table II. Algorithmic efficiency: Comparison of mean number of recursive calls ($\cdot 10^3$) for different combinations of using bounding pairs, local consistency and the most-important-first variable heuristic for a fixed weighted CSP with standard deviations given in parentheses.

	\neg WCB + \neg LC	\neg WCB + LC	WCB + \neg LC	WCB + LC
RO	1,914 (1,359)	472 (993)	1,914 (1,359)	472 (993)
MIF	1,264 (1,290)	218 (649)	1,264 (1,289)	217 (649)

boundaries. Using a worst-case boundary only does not lead to better results in our experiments. These results hint that variable ordering heuristics using “important” variables should be researched further in conjunction with traditional ordering heuristics based on, e.g., the cardinality of the domains.

VII. CONCLUSIONS AND FUTURE WORK

We have motivated constraint relationships to qualitatively specify soft constraint preferences. In a case study, abstracted from a realistic scenario, we have illustrated the benefits for decision making in multi-agent systems. We have also provided a connection of constraint relationships to c-semirings via free constructions, first from the dag over constraints to partial valuation structures, second from partial valuation structures to c-semirings. This connection suggests partial valuation structures as a direct means for soft constraints and we have evaluated different heuristics for finding the maximum solution degrees.

Partial valuation structures offer lexicographical products [14] with which we plan to revisit the encoding of constraint hierarchies as c-semirings [28]. Due to the conceptual simplicity constraint relationships offer towards users, we also plan to use this formalism for preference elicitation and preference learning. A process of asking preference decisions over solutions to elicit constraint relationships is envisioned.

ACKNOWLEDGMENT

We thank Jan-Philipp Steghöfer and Florian Nafz for their initial involvement in constraint relationships. This research is partly sponsored by the German Research Foundation (DFG) in the project “OC-Trust” (FOR 1085).

REFERENCES

- [1] A. Schiendorfer, J.-P. Steghöfer, and W. Reif, “Synthesised Constraint Models for Distributed Energy Management,” in *Proc. 3rd Int. Wsh. Smart Energy Networks & Multi-Agent Systems (SEN-MAS’14)*, 2014.
- [2] M. Sachenbacher and B. C. Williams, “Diagnosis as Semiring-based Constraint Optimization,” in *Proc. 16th Europ. Conf. Artificial Intelligence (ECAI’04)*, R. L. de Mántaras and L. Saitta, Eds. IOS Press, 2004, pp. 873–877.
- [3] P. Meseguer, F. Rossi, and T. Schiex, “Soft Constraints,” in *Handbook of Constraint Programming*, F. Rossi, P. van Beek, and T. Walsh, Eds. Elsevier, 2006, ch. 9.
- [4] E. C. Freuder and R. J. Wallace, “Partial Constraint Satisfaction,” *Artif. Intell.*, vol. 58, no. 1–3, pp. 21–70, 1992.
- [5] C. Ansótegui, M. L. Bonet, J. Levi, and F. Manyà, “The Logic Behind Weighted CSP,” in *Trends in Constraint Programming*. ISTE, 2010, pp. 303–316.
- [6] G. R. Santhanam, S. Basu, and V. Honavar, “Representing and Reasoning with Qualitative Preferences for Compositional Systems,” *J. Artif. Int. Res.*, vol. 42, no. 1, pp. 211–274, 2011.
- [7] A. Borning, B. Freeman-Benson, and M. Wilson, “Constraint Hierarchies,” *LISP Symb. Comp.*, vol. 5, pp. 223–270, 1992.
- [8] A. Schiendorfer, J.-P. Steghöfer, A. Knapp, F. Nafz, and W. Reif, “Constraint Relationships for Soft Constraints,” in *Proc. 33rd SGAI Int. Conf. Innovative Techniques and Applications of Artificial Intelligence (AI’13)*, M. Bramer and M. Petridis, Eds. Springer, 2013, pp. 241–255.
- [9] F. Rossi, “Collective Decision Making: A Great Opportunity for Constraint Reasoning,” *Constraints*, vol. 19, no. 2, pp. 186–194, 2014.
- [10] S. Bistarelli, U. Montanari, and F. Rossi, “Semiring-based Constraint Satisfaction and Optimization,” *J. ACM*, vol. 44, no. 2, pp. 201–236, 1997.
- [11] T. Schiex, H. Fargier, and G. Verfaillie, “Valued Constraint Satisfaction Problems: Hard and Easy Problems,” in *Proc. 14th Int. Joint Conf. Artificial Intelligence (IJCAI’95)*, Vol. 1. Morgan Kaufmann, 1995, pp. 631–639.
- [12] L. Leenen and A. K. Ghose, “Branch and Bound Algorithms to Solve Semiring Constraint Satisfaction Problems,” in *Proc. 10th Pacific Rim Int. Conf. Artificial Intelligence (PRICAI’08)*, ser. Lect. Notes Comp. Sci. 5351, T.-B. Ho and Z.-H. Zhou, Eds. Springer, 2008, pp. 991–997.
- [13] L. Leenen, Anbulagan, T. Meyer, and A. K. Ghose, “Modeling and Solving Semiring Constraint Satisfaction Problems by Transformation to Weighted Semiring Max-SAT,” in *Proc. 20th Australian Joint Conf. Artificial Intelligence (ACAI’07)*, ser. Lect. Notes Comp. Sci. 4830, M. A. Orgun and J. Thornton, Eds. Springer, 2007, pp. 202–212.
- [14] F. Gadducci, M. M. Hölzl, G. V. Monreale, and M. Wirsing, “Soft Constraints for Lexicographic Orders,” in *Proc. 12th Mexican Int. Conf. Artificial Intelligence (MICA’13)*, ser. Lect. Notes Comp. Sci. 8265, F. Castro, A. Gelbukh, and M. González, Eds. Springer, 2013, pp. 68–79.
- [15] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole, “CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements,” *J. Artif. Intell. Res.*, vol. 21, pp. 135–191, 2004.
- [16] J. Doyle and M. McGeachie, “Exercising Qualitative Control in Autonomous Adaptive Survivable Systems,” in *Proc. 2nd Int. Conf. Self-adaptive Software: Applications (IWSAS’01)*, ser. Lect. Notes Comp. Sci. 2614. Springer, 2003, pp. 158–170.
- [17] F. Rossi, K. B. Venable, and T. Walsh, “Preferences in Constraint Satisfaction and Optimization,” *AI Mag.*, vol. 29, no. 4, pp. 58–68, 2008.
- [18] G. Grätzer, *Universal Algebra*, 2nd ed. Springer, 2008.
- [19] D. Sannella and A. Tarlecki, *Foundations of Algebraic Specification and Formal Software Development*, ser. EATCS Monographs in Theoretical Computer Science. Springer, 2012.
- [20] A. Knapp and A. Schiendorfer, “Embedding Constraint Relationships into C-Semirings,” Institute for Software and Systems Engineering, University of Augsburg, Tech. Rep. 2014-03, 2014, <http://opus.bibliothek.uni-augsburg.de/opus4/frontdoor/index/index/docId/2684>.
- [21] R. M. Amadio and P.-L. Curien, *Domains and Lambda-Calculi*, ser. Cambridge Tracts in Theoretical Computer Science 46. Cambridge University Press, 1998.
- [22] H. Fargier, E. Rollon, and N. Wilson, “Enabling Local Computation for Partially Ordered Preferences,” *Constraints*, vol. 15, no. 4, pp. 516–539, 2010.
- [23] S. Bistarelli, F. Gadducci, J. Larrosa, and E. Rollon, “A Soft Approach to Multi-objective Optimization,” in *Proc. 24th Int. Conf. Logic Programming (ICLP’08)*, ser. Lect. Notes Comp. Sci. 5366, M. G. de la Banda and E. Pontelli, Eds. Springer, 2008, pp. 764–768.
- [24] S. Bistarelli, P. Codognet, H. K. C. Hui, and J. H.-M. Lee, “Solving Finite Domain Constraint Hierarchies by Local Consistency and Tree Search,” *J. Exp. Theor. Artif. Intell.*, vol. 21, no. 4, pp. 233–257, 2009.
- [25] I. P. Gent, E. MacIntyre, P. Presser, B. M. Smith, and T. Walsh, “An Empirical Study of Dynamic Variable Ordering Heuristics for the Constraint Satisfaction Problem,” in *Proc. 2nd Int. Conf. Principles and Practice of Constraint Programming (CP’96)*, ser. Lect. Notes Comp. Sci. 1118, E. C. Freuder, Ed. Springer, 1996, pp. 179–193.
- [26] N. Levasseur, P. Boizumault, and S. Loudni, “A Value Ordering Heuristic for Weighted CSP,” in *Proc. 19th IEEE Int. Conf. Tools with Artificial Intelligence (ICTAI’07)*, Vol. 1. IEEE, 2007, pp. 259–262.
- [27] M. C. Cooper, S. de Givry, M. Sanchez, T. Schiex, M. Zymnicki, and T. Werner, “Soft Arc Consistency Revisited,” *Artif. Intell.*, vol. 174, no. 7–8, pp. 449–478, 2010.
- [28] H. Hosobe, “Constraint Hierarchies as Semiring-based CSPs,” in *Proc. 21st IEEE Int. Conf. Tools with Artificial Intelligence (ICTAI’09)*. IEEE, 2009, pp. 176–183.