Interval-valued Soft Constraint Problems

M. Gelain[1], M. S. Pini[1], F. Rossi[1], K. B. Venable[1], and N. Wilson[2]

[1] Dipartimento di Matematica Pura ed Applicata, University of Padova, Italy
E-mail: {mgelain,mpini,frossi,kvenable}@math.unipd.it
[2] Cork Constraint Computation Centre, University College Cork, Ireland, Email: n.wilson@4c.ucc.ie

Abstract. Constraints and quantitative preferences, or costs, are very useful for modelling many real-life problems. However, in many settings, it is difficult to specify precise preference values, and it is much more reasonable to allow for preference intervals. We define several notions of optimal solutions for such problems, providing algorithms to find optimal solutions and also to test whether a solution is optimal. Most of the time these algorithms just require the solution of soft constraint problems, which suggests that it may be possible to handle this form of uncertainty in soft constraints without significantly increasing the computational effort needed to reason with such problems. This is supported also by experimental results. We also identify classes of problems where the same results hold if users are allowed to use multiple disjoint intervals rather than a single one.

1 Introduction

Constraints [11] are useful to model real-life problems when it is clear what should be accepted and what should be forbidden. Soft constraints [9] extend the constraint notion by allowing several levels of acceptance. This allows to express preferences and/or costs rather than just strict requirements.

In soft constraints, each instantiation of the variables of a constraint must be associated to a precise preference or cost value. Sometimes it is not possible for a user to know exactly all these values. For example, a user may have a vague idea of the preference value, or may not be willing to reveal his preference, for example for privacy reasons.

In this paper we consider these forms of imprecision, and we handle them by extending soft constraints to allow users to state an interval of preference values for each instantiation of the variables of a constraint. This interval can contain a single element (in this case we have usual soft constraints), or the whole range of preference values (when there is complete ignorance about the preference value), or it may contain more than one element but a strict subset of the set of preference values. We call such problems *interval-valued* soft CSPs (or also IVSCSPs).

In an elicitation procedure there will typically be some degree of imprecision, so attributing an interval rather than a precise preference degree can be a more reliable model of the information elicited. Also, linguistic descriptions of degrees of preference (such as ”quite high” or ”low” or ”undesirable”) may be more naturally mapped to preference intervals, especially if the preferences are being elicited from different experts, as they may mean somewhat different things by these terms.

Two examples of real world application domains where preference intervals can be useful or necessary are energy trading and network traffic analysis [15], where the data information is usually incomplete or erroneous. In energy trading, costs may be imprecise because they may evolve due to market changes; in network traffic analysis, the overwhelming amount of information and measurement difficulties force the use of partial or imprecise information. Many other application domains that are usually modelled via hard or soft constraints could benefit by increased expressed power of preference intervals. To give a concrete example in this paper we consider the meeting scheduling problem, that is a typical benchmark for CSPs, and we allow the specification of preference intervals. This benchmark will be used both to clarify notions related to IVCSPs and to run experimental tests.

Given an IVSCSP, we consider several notions of optimal solutions. We first start with general notions of optimality, which apply whenever we have several scenarios to consider. For example, as done in [7], we consider *necessarily optimal* solutions, which are optimal in all scenarios, or *possibly optimal* solutions, which are optimal in at least one scenario. We then pass to *interval-based optimality notions*, that define optimality in terms of the upper and lower bounds of the intervals associated to the solution by the constraints.

Since IVSCSPs generalize soft constraint problems, the problem of finding an optimal solution in an IVSCP (according to any of the considered optimality notions) is at least as difficult as finding an optimal solution in a soft constraint problem ans thus it is NP-hard.

We provide algorithms to find solutions according to all the notions defined, and also to test whether a given solution is optimal. In most of the cases, finding or testing an optimal solution amounts to solving a soft constraint problem. Thus, even if our formalism significantly extends soft constraints, and gives users much more power in modelling their knowledge of the real world, in the end the work needed to find an optimal solution (or to test if it is optimal) is not more than that needed to find an optimal solution in a soft constraint problem. This claim is supported by the experimental results we present, obtained by extensive tests over instances of the meeting scheduling problem.

We also show that for some classes of IVSCSPs the optimality notions considered in this paper would not produce different results if users were allowed to use *multiple disjoint intervals* rather than a single one. This means that a level of precision greater than a single interval does not add any useful information when looking for an optimal solution.

Previous approaches to uncertainty in soft constraint problems assumed either a complete knowledge of the preference value, or a complete ignorance. In other words, a preference value in a domain or a constraint was either present or not [4, 6, 8, 14]. Then, the solver was trying to find optimal solutions with the information given by the user or via some form of elicitation of additional preference values. Here instead we consider a more general setting where the user may specify preference intervals. Also, we assume that the user has given us all the information he has about the problem, so we do not resort to preference elicitation (or the elicitation phase is over with the user being unable or unwilling to give us more precise information). Moreover, previous work looks only for necessarily optimal solutions, and uses preference elicitation, if needed, to find them. Here instead we consider many other notions of optimal solutions, with the aim of returning interesting solutions without resorting to preference elicitation.

Another work that analyzes the impact of the uncertainty in soft constraint problems is shown in [10]. However, while we assume to have only preference intervals, in [10] it is assumed that all the preferences are given and some of them are tagged as possibly unstable and are provided with a range, of possible variations, around their value.

Other papers consider preference intervals, such as the work in [3]. However, these lines of work focus on specific preference aggregation mechanisms (such as the Choquet integral) and of modelling issues without addressing the algorithmic questions related to finding optimal solutions according to different risk attitudes. We are instead interested in providing efficient algorithms to find optimal solutions according to different risk attitudes (called pessimistic and optimistic in the paper), besides the modelling concerns.

The paper is structured as follows. In Section 2 we recall the main definitions for soft constraints and in Section 3 we introduce interval-valued soft constraint problems. In Section 4 we give general notions of optimal solutions, which apply whenever we have several scenarios to consider, while in Section 5 we introduce interval-based optimality notions. In Sections 6 and 7 we present algorithms to find solutions according to optimality notions defined. Then, in Section 8 we introduce notions of dominance between solutions, we show how they are related to the notions of optimality, and we describe how to test dominance. In Section 9 we analyze the impact of having multiple preference intervals. In Section 10 we present an experimental study of the algorithms to find optimal solutions. Finally, in Section 11 we give some final considerations and we propose some hints for future work.

2 Background: soft constraints

In the literature there are many formalizations of the concept of soft constraints [5, 12]. Here we refer to the one described in [1, 5], which however can be shown to generalize and express many others [2].

A soft constraint [1] is just a classical constraint where each instantiation of its variables has an associated value from a (totally or partially ordered) set, which is called a c-semiring. More precisely, a c-semiring is a tuple $\langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ such that: A is a set, called the carrier of the c-semiring, and $\mathbf{0}, \mathbf{1} \in A$; $+$ is commutative, associative, idempotent, $\mathbf{0}$ is its unit element, and $\mathbf{1}$ is its absorbing element; $\times$ is associative, commutative, distributes over $+$, $\mathbf{1}$ is its unit element and $\mathbf{0}$ is its absorbing element. Consider the relation $\leq_S$ over A such that $a \leq_S b$ iff $a + b = b$. $\leq_S$ is a partial order; $+$ and $\times$ are monotone on $\leq_S$; $\mathbf{0}$ is its minimum and $\mathbf{1}$ its maximum; $\langle A, \leq_S \rangle$ is a lattice and, for all $a, b \in A$, $a + b = lub(a, b)$. Moreover, if $\times$ is idempotent, then $\langle A, \leq_S \rangle$ is a distributive lattice and $\times$ is its glb. The relation $\leq_S$ gives us a way to compare preference values: when $a \leq_S b$, we say that *b is better than a*. Element $\mathbf{0}$ is the worst value and $\mathbf{1}$ is the best one.

A c-semiring $\langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ is said to be *idempotent* when the combination operator $\times$ is idempotent, while it is said to be *strictly monotonic* when the combination operator $\times$ is strictly monotonic. If a c-semiring is totally ordered, i.e., if $\leq_S$ is a total order, then the $+$ operation is just max with respect to $\leq_S$. If the c-semiring is also idempotent, then $\times$ is equal to min, and the c-semiring is of the kind used for fuzzy constraints (see below). Notice that there are also c-semirings that are neither idempotent nor strictly monotonic.

Given a c-semiring $S = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$, a finite set D (the domain of the variables), and an ordered set of variables V, a soft constraint is a pair $\langle def, con \rangle$ where $con \subseteq V$ and $def : D^{|con|} \rightarrow A$. Therefore, a soft constraint specifies a set of variables (the ones in con), and assigns to each tuple of values of D of these variables an element of the c-semiring set A, which will be seen as its *preference*. A soft constraint satisfaction problem (SCSP) is just a set of soft constraints over a set of variables.

A classical CSP is just an SCSP where the chosen c-semiring is $S_{CSP} = \langle \{false, true\}, \vee, \wedge, false, true \rangle$. Fuzzy CSPs are instead modeled by choosing the idempotent c-semiring $S_{FCSP} = \langle [0, 1], max, min, 0, 1 \rangle$: we want to maximize the minimum preference. For weighted CSPs, the strictly monotonic c-semiring is $S_{WCSP} = \langle \Re^+, min, +, +\infty, 0 \rangle$: preferences are interpreted as costs from 0 to $+\infty$, and we want to minimize the sum of costs.

Given an assignment s to all the variables of an SCSP Q, that is, a solution of Q, its preference, written $\mathrm{pref}(Q, s)$, is obtained by combining the preferences associated by each constraint to the subtuples of s referring to the variables of the constraint: $\mathrm{pref}(Q, s) = \Pi_{\langle idef,con \rangle \in C}\ def(s_{\downarrow con})$, where Π refers to the $\times$ operation of the c-semiring and $s_{\downarrow con}$ is the projection of tuple s on the variables in con. For example, in fuzzy CSPs, the preference of a complete assignment is the minimum preference given by the constraints. In weighted constraints, it is instead the sum of the costs given by the constraints. An optimal solution of an SCSP Q is then a complete assignment s such that there is no other complete assignment s'' with $\mathrm{pref}(Q, s) <_S \mathrm{pref}(Q, s'')$. We denote with $Opt(Q)$ the set of all optimal solutions of an SCSP Q and with $Sol(Q)$ the set of all the solutions of an SCSP Q.

Given an SCSP Q defined over an idempotent c-semiring, and a preference α, we will denote as $cut_\alpha(Q)$ (resp., $scut_\alpha(Q)$) the CSP obtained from Q allowing only tuples with preference greater than or equal to α (resp., strictly greater than α). It is known that the set of solutions of Q with preference greater than or equal to α(resp., strictly greater than α) coincides with the set of solutions of $cut_\alpha(Q)$ (resp., $scut_\alpha(Q)$).

3 Interval-valued soft constraints

Soft constraint problems require users to specify a preference value for each tuple in each constraint. Sometimes this is not reasonable, because a user may have a vague idea of what preferences to associate to some tuples. In [6] a first generalization allowed users to specify either a fixed preference (as in usual soft constraints) or the complete $[\mathbf{0}, \mathbf{1}]$ interval. Thus an assumption of complete ignorance was made when the user was not able to specify a fixed preference. Here we generalize further by allowing users to state any interval over the preference set.

Definition 1 (interval-valued soft constraint). *Given a set of variables V with finite domain D and a* totally-ordered *c-semiring $S = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$, an interval-valued* *soft constraint is a pair* $\langle int, con\rangle$ *where* $con \subseteq V$ *is the scope of the constraint and int:* $D^{|con|} \longrightarrow A \times A$ *specifies an interval over* A *by giving its lower and upper bound. If* $int(x) = (a, b)$*, it must be* $a \leq_S b$*.*

In the following we will denote with $l(int(x))$ (resp., $u(int(x))$) the first (resp., second) component of $int(x)$, representing the lower and the upper bound of the preference interval.

Definition 2 (IVSCSP). *An interval-valued soft constraint problem (IVSCSP) is a 4-tuple* $\langle V, D, C, S\rangle$*, where* C *is a set of interval-valued soft constraints over* S *defined on the variables in* V *with domain* D*.*

Figure 1 shows an IVSCSP P defined over the fuzzy c-semiring $\langle [0, 1], max, min, 0, 1\rangle$, that contains three variables X_1, X_2, and X_3, with domain $\{a, b\}$, and five constraints: a unary constraint on each variable, and two binary constraints on (x_1, x_2) and (x_2, x_3).

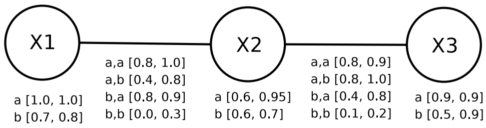


Fig. 1. An IVSCSP over fuzzy semiring.

In an IVSCSP, a complete assignment of values to all the variables can be associated to an interval as well. The lower bound (resp., the upper bound) of such an interval is obtained by combining all the lower bounds (resp., the upper bounds) of the preference intervals of the appropriate subtuples of this assignment in the various constraints.

Definition 3 (preference interval). *Given an IVSCSP* $P = \langle V, D, C, S\rangle$ *and an assignment* s *to all its variables over* D*, the preference interval of* s *in* P *is* $[L(s), U(s)]$*, where* $L(s) = \Pi_{<int,con>\in C} l(int(\ s_{\downarrow con}))$ *and* $U(s) = \Pi_{<int,con>\in C} u(int(s_{\downarrow con}))$*, and* Π *is the combination operator of the c-semiring* S*.*

Figure 2 shows all the complete assignments of the IVSCSP in Figure 1, together with their preference interval and the computation details for s_1.

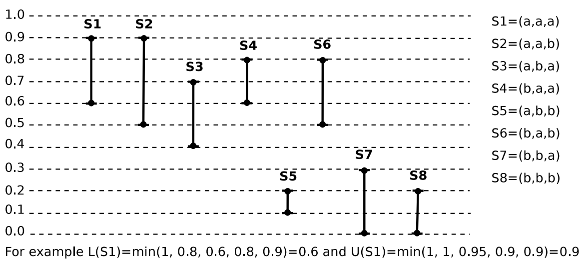


Fig. 2. Solutions of the IVSCSP shown in Figure 1.

Once we have an IVSCSP, it is useful to consider specific scenarios arising from choosing a preference value from each interval.

Definition 4 (scenario). *Given an IVSCSP P, a scenario of P is an SCSP P' obtained from P as follows: given any constraint $c = \langle int, con \rangle$ of P, we insert in P' the constraint $c' = \langle def, con \rangle$, where $def(t) \in [l(int(t)), u(int(t))]$ for every tuple $t \in D^{|con|}$.*

We will denote with $Sc(P)$ the set of all possible scenarios of P.

Definition 5 (best and worst scenario). *Given an IVSCSP P, the best scenario ($bs(P)$) (resp., the worst scenario ($ws(P)$)) of P is the scenario obtained by replacing every interval with its upper (resp., lower) bound.*

We will denote with l_{opt} and u_{opt} the optimal preferences of the worst and best scenario.

The preference interval of a complete assignment is a good way of representing the quality of the solution in all scenarios, as stated by the following theorem.

Theorem 1. *Consider an IVSCSP P over a c-semiring S and a complete assignment s of its variables. Then, for all $Q \in Sc(P)$, $\mathrm{pref}(Q, s) \in [L(s), U(s)]$. Also, for $p \in \{L(s), U(s)\}$, there exists a $Q \in Sc(P)$ such that $p = \mathrm{pref}(Q, s)$. If the c-semiring S is idempotent, then for all $p \in [L(s), U(s)]$, there exists a $Q \in Sc(P)$ such that $p = \mathrm{pref}(Q, s)$.*

Proof: $\mathrm{pref}(Q, s) \in [L(s), U(s)]$ follows by monotonicity. If $p = L(s)$ (resp., $p = U(s)$), it is possible to build a scenario where $p = pref(Q, s)$, by fixing all the tuples of s to their lower bound (resp., to their upper bound). If the c-semiring is idempotent, since we are considering totally ordered c-semirings, the operator $\times$ is minimum (with respect to the total order), so there exists some interval-valued constraint $\langle int, con \rangle$ in P such that $l(int(s_{\downarrow con})) = L(s)$. We must also have $u(int(s_{\downarrow con})) \geq U(s)$. Let p be an element of $[L(s), U(s)]$. Define a scenario Q by replacing this interval-valued constraint with any soft constraint which assigns the tuple $s_{\downarrow con}$ the preference value p, and replacing any of the other elements of P with soft constraints which assign preference value $U(s)$ to the appropriate projection of s. We then have $p = \mathrm{pref}(Q, s)$. □

This means that, in general, the upper and lower bounds of the solution preference interval always model preferences of solutions in some scenarios. In the idempotent case we have more: the whole interval, and not just the bounds, represents all and only the preferences coming from the scenarios. Intuitively, if $\times$ is idempotent (let us consider min for simplicity): given an assignment s, for every element x in $[L(s), U(s)]$, we can construct a scenario where s has preference x by fixing preference x on at least one tuple (that has x in its interval) and by fixing all other preferences of tuples in s to their upper bound.

4 Necessary and possible optimality

We will now consider general notions of optimality, that are applicable to any setting where the lack of precision gives rise to several possible scenarios. First we define optimal solutions that guarantee optimality in some or all scenarios (i.e., the possibly and the necessarily optimal solutions [6]), and then we will define solutions that guarantee a certain level of preference in some or all scenarios.

Definition 6 (necessarily optimal). *Given an IVSCSP $P = \langle V, D, C, S\rangle$ and an assignment s to the variables in V, s is necessarily optimal iff it is optimal in all scenarios.*

Given an IVSCSP P, the set of its necessarily optimal solutions will be denoted by $NO(P)$. Necessarily optimal solutions are very attractive because they are very robust: they are optimal independently of the uncertainty of the problem. Unfortunately, $NO(P)$ may be empty, as in the IVSCSP P of Figure 1.

Definition 7 (possibly optimal). *Given an IVSCSP $P = \langle V, D, C, S\rangle$ and an assignment s to the variables in V, s is possibly optimal iff it is optimal in some scenario.*

Given an IVSCSP P, the set of possibly optimal solutions of P will be denoted by $PO(P)$. In the IVSCSP P of Figure 1 we have $PO(P) = \{s_1, s_2, s_3, s_4, s_6\}$. $PO(P)$ is never empty. However, the possibly optimal solutions are less attractive than the necessarily optimal ones, in fact they guarantee optimality only for a specific completion of the uncertainty.

We assume now to want to guarantee a certain level of preference in some or all the scenarios.

Definition 8 (necessarily of at least preference α). *Given an IVSCSP $P = \langle V, D, C, S\rangle$ and an assignment s to the variables in V, s is necessarily of at least preference α iff, for all scenarios, its preference is at least α.*

The set of all solutions of an IVSCSP P with this feature will be denoted by $Nec(P, \alpha)$. In our running example, if we consider $\alpha = 0.5$, we have $Nec(P, 0.5) = \{s_1, s_2, s_4, s_6\}$. If α is a satisfactory preference level, elements in $Nec(P, \alpha)$ are ideal, because they guarantee such a preference level independently of the uncertainty of the problem.

Definition 9 (possibly of at least preference α). *Given an IVSCSP $P = \langle V, D, C, S\rangle$ and an assignment s to the variables in V, s is possibly of at least preference α iff, for some scenario, its preference is at least α.*

The set of all solutions of an IVSCSP P with this feature will be denoted by $Pos(P, \alpha)$. In the IVSCSP P of Figure 1, if we take $\alpha = 0.3$, we have $Pos(P, 0.3) = \{s_1, s_2, s_3, s_4, s_6, s_7\}$.

5 Interval-based optimality notions

In an IVSCSP, uncertainty is specified via the preference intervals. Depending on how one decides to deal with this form of uncertainty, different notions of optimality can be given. Here we will consider interval-based optimality notions, and we will relate them to the necessarily and possibly optimal solutions.

5.1 Interval-dominant assignments

In the attempt to characterize the necessarily optimal solutions, we can consider the following notion.

Definition 10 (interval-dominant). *Given an IVSCSP $P = \langle V, D, C, S \rangle$ and an assignment s to the variables in V, s is interval-dominant iff, for every other complete assignment s', $L(s) \geq U(s')$.*

Interval-dominant assignments are better than or equal to all others in all scenarios, and thus are very robust w.r.t. uncertainty. We denote with $ID(P)$ the set of the interval dominant assignments of P. The IVSCSP P of Figure 1 has $ID(P) = \emptyset$.

Proposition 1. *If $ID(P) \neq \emptyset$, either $ID(P)$ contains a single solution, or all the solutions in $ID(P)$ have their lower bound equal to their upper bound, and all these bounds are equal to the same value. Given an IVSCSP P, $ID(P)$ may be empty.*

Proof: $ID(P)$ may be empty as in the IVSCSP P of Figure 1.

We now show, by contradiction, that if $ID(P) \neq \emptyset$, either $ID(P)$ contains a single solution, or several solutions all with the lower bound equal to the upper bound, and all equal to the same value. If $ID(P)$ contains two solutions, say s_1 and s_2, with different values of lower and upper bounds, then $L(s_1) < U(s_1)$ and $L(s_2) < U(s_2)$. Since $s_1 \in ID(P)$, then for any other solution s', $L(s_1) \geq U(s')$ and thus also $L(s_1) \geq U(s_2)$. Similarly, since $s_2 \in ID(P)$, then for any other solution s', $L(s_2) \geq U(s')$ and thus $L(s_2) \geq U(s_1)$. Therefore, $L(s_1) \geq U(s_2) > L(s_2) \geq U(s_1)$ and so $L(s_1) > U(s_1)$, that is a contradiction. □

It is possible to show that the interval-dominant optimality notion is stronger than the necessary optimality notion. More precisely:

Proposition 2. *Given an IVSCSP P, we have that $ID(P) \subseteq NO(P)$. Also, if $ID(P) \neq \emptyset$, then $ID(P) = NO(P)$.*

Proof: We first show that $ID(P) \subseteq NO(P)$. If a solution is in $ID(P)$, its preference is always greater than or equal to the upper bounds of all the other solutions, hence it is optimal in all the scenarios.

We now prove that, if $ID(P) \neq \emptyset$, then $ID(P) = NO(P)$. We have already shown that $ID(P) \subseteq NO(P)$. It remains to prove that $NO(P) \subseteq ID(P)$. Let us denote with s^* a solution of $ID(P)$. If a solution s of P is not in $ID(P)$ and $ID(P) \neq \emptyset$, then s is not in $NO(P)$. In fact, if $L(s^*) \neq U(s^*)$, then $U(s^*) > L(s^*) \geq U(s)$, and so s is not optimal in the best scenario. If $L(s^*) = U(s^*)$, since $s \notin ID(P)$, $L(s) < L(s^*)$ and so s is not optimal in the worst scenario. □

5.2 Weakly-interval-dominant assignments

A more relaxed interval-based optimality notion is the following one.

Definition 11 (weakly-interval-dominant). *Given an IVSCSP $P = \langle V, D, C, S\rangle$ and an assignment s to the variables in V, s is weakly-interval-dominant iff, for every other complete assignment s', $L(s) \geq L(s')$ and $U(s) \geq U(s')$.*

Weakly-interval-dominant assignments are better than or equal to all others in both the worst and the best scenario. We denote with $WID(P)$ the set of the weakly interval dominant assignments of P. The IVSCSP P of Figure 1 has $WID(P) = \{s_1\}$.

Proposition 3. *Given an IVSCSP P, $WID(P)$ may be empty. Moreover, $ID(P) \subseteq WID(P)$.*

Proof: $WID(P)$ may be empty. For example, one can construct an IVSCSP over fuzzy c-semiring with only three solutions, say s_1, s_2, and s_3, with the following lower and upper bounds: $L(s_1) = 0.2, U(s_1) = 0.6, L(s_2) = 0.3, U(s_2) = 0.8, L(s_3) = 0.4$, and $U(s_3) = 0.7$.

We now show that $ID(P) \subseteq WID(P)$. If $s \in ID(P)$, then $L(s) \geq U(s')$ for every other s'. Hence, since $U(s) \geq L(s)$ and $U(s') \geq L(s')$ for every other s', we have $U(s) \geq L(s) \geq U(s') \geq L(s')$ for every other s', that is, $U(s) \geq U(s')$ and $L(s) \geq L(s')$ for every other s', hence $s \in WID(P)$. □

The weakly-interval-dominant optimality notion is weaker than the necessary optimality notion. In fact, $NO(P) \subseteq WID(P)$ and for some IVSCSP P (for example, the IVSCSP of Figure 1) this inclusion is strict. More precisely:

Proposition 4. *Given an IVSCSP P, we have that $ID(P) \subseteq NO(P) \subseteq WID(P)$.*

Proof: By Proposition 2, we know that $ID(P) \subseteq NO(P)$.

We now show that $NO(P) \subseteq WID(P)$. If $s \in NO(P)$, then s must be optimal in every scenario and so also in the best and in the worst scenario. Given that s is optimal in the worst scenario, then $L(s) \geq L(s')$ for every other solution s'. Moreover, as s is optimal in the best scenario, then $U(s) \geq U(s')$ for every other solution s'. Therefore, $L(s) \geq L(s')$ and $U(s) \geq U(s')$ for every other solution s'. This allows us to conclude that $s \in WID(P)$. □

Since $ID(P) \subseteq NO(P) \subseteq WID(P)$, $ID(P)$ and $WID(P)$ can be seen as lower and upper approximations of $NO(P)$.

5.3 Lower and upper optimal assignments

Until now we have considered how to characterize, via interval-based optimality notions, the necessarily optimal solutions. In particular, we have found lower and upper approximations of these optimal solutions. We now move to consider possibly optimal solutions via new interval-based optimality notions.

Definition 12 (lower and upper optimal). *Given an IVSCSP $P = \langle V, D, C, S\rangle$ and an assignment s to the variables in V, s is lower-optimal (resp., upper-optimal) iff, for every other complete assignment s', $L(s) \geq L(s')$ (resp., $U(s) \geq U(s')$).*

A lower-optimal (resp., an upper-optimal) assignment is better than or equal to all other complete assignments in the worst scenario (resp., in the best scenario). Lower-optimal (resp., upper-optimal) assignments are useful in pessimistic (resp., optimistic) approaches to uncertainty, because they outperform the other assignments in the worst (resp., in the best) case. We denote with $LO(P)$ (resp., $UO(P)$) the set of the lower (resp., upper) optimal assignments of P. The IVSCSP P of Figure 1 has $LO(P) = \{s_1, s_4\}$ and $UO(P) = \{s_1, s_2\}$.

Lower and upper optimal solutions are never empty. Moreover, they are related to weakly-interval-dominant and interval-dominant solutions as follows.

Proposition 5. *Given an IVSCSP P, and the optimal preference l_{opt} (resp., u_{opt}) of $ws(P)$ (resp., $bs(P)$),*

- *$LO(P)$ and $UO(P)$ are never empty;*
- *$UO(P) \cap LO(P) = WID(P)$;*
- *if $l_{opt} = u_{opt}$, then $ID(P) = LO(P)$;*
- *if $l_{opt} < u_{opt}$, and $|UO(P)| \geq 2$, then $ID(P) = \emptyset$;*
- *if $|UO(P)| = 1$, let us call s this single solution. If $L(s) \neq l_{opt}$ then $ID(P) = \emptyset$.*

Proof: $LO(P)$ is never empty because it is always possible to find the solutions with the lower bound greater than or equal to all the other solutions. A similar argument shows that $UO(P)$ is never empty.

We now show that $UO(P) \cap LO(P) = WID(P)$. We first show that $UO(P) \cap LO(P) \subseteq WID(P)$. If $s \in UO(P) \cap LO(P)$, then, by definition of $UO(P)$, $U(s) \geq U(s')$ for every other s' and, by definition of $LO(P)$, $L(s) \geq L(s')$ for every other s', therefore $s \in WID(P)$. We now show that $WID(P) \subseteq UO(P) \cap LO(P)$. If $s \in WID(P)$, by definition of $WID(P)$, $U(s) \geq U(s')$ and $L(s) \geq L(s')$ for every other s', hence both $s \in LO(P)$ and $s \in UO(P)$, therefore $s \in LO(P) \cap UO(P)$.

To show that, if $l_{opt} = u_{opt}$, then $ID(P) = LO(P)$, it is sufficient to show that $l_{opt} = u_{opt}$ implies $LO(P) \subseteq ID(P)$, as $ID(P) \subseteq LO(P)$ follows from Theorem 2. In fact, if $s \in ID(P)$, then $s \in Opt(ws(P))$ and thus, by Theorem 2, $s \in LO(P)$. If $s \in LO(P)$ then $L(s) = l_{opt}$. Moreover, since $l_{opt} = u_{opt}$, $L(s) = u_{opt}$, and so $L(s) \geq U(s')$, for every other solution s', that is $s \in ID(P)$.

We now prove, by contradiction, that, if $l_{opt} < u_{opt}$ and $|UO(P)| \geq 2$, then $ID(P) = \emptyset$. Suppose $ID(P) \neq \emptyset$. Let us denote with s one of the solutions of $ID(P)$. Then, by definition of $ID(P)$, $L(s) \geq U(s')$, for every other solution s'. Since $|UO(P)| \geq 2$, we are sure that there is a solution $s'' \neq s$ such that $U(s'') = u_{opt}$. Hence, $L(s) \geq U(s'') = u_{opt} > l_{opt}$, and so $L(s) > l_{opt}$, that is a contradiction, because, by the definition of l_{opt}, l_{opt} is greater than or equal to the lower bound of every solution.

Assume that $|UO(P)| = 1$ and let us call s this single solution. We now show, by contradiction, that, if $L(s) \neq l_{opt}$, then $ID(P) = \emptyset$. Let us denote with s_1 one of the solutions with $L(s_1) = l_{opt}$. Suppose that $ID(P) \neq \emptyset$, and let s' be an element of $ID(P)$. If $s' \neq s$ then $U(s') \geq L(s') \geq U(s)$, which implies that $s' \in UO(P)$, a contradiction. Hence $s' = s$. But then $s' \neq s_1$, so $L(s') \geq U(s_1) \geq L(s_1) = l_{opt}$, which contradicts $L(s) \neq l_{opt}$. □

As every lower (resp., upper) optimal solution is optimal in the worst (resp. best) scenario, then $LO(P) \subseteq PO(P)$, $UO(P) \subseteq PO(P)$, and these inclusions may be strict, because there may be solutions that are optimal only in scenarios that are different from the best and the worst scenario.

Proposition 6. *Given an IVSCSP P, we have that $LO(P) \cup UO(P) \subseteq PO(P)$.*

Proof: Let s be a complete assignment to the variables of P.

$LO(P) \subseteq PO(P)$. In fact, if $s \in LO(P)$, then s is optimal in the worst scenario and so $s \in PO(P)$.

$UO(P) \subseteq PO(P)$. In fact, if $s \in UO(P)$, then s is optimal in the best scenario and so $s \in PO(P)$.

Therefore, $LO(P) \cup UO(P) \subseteq PO(P)$. □

Therefore, the lower and upper optimality notions are stronger than the possible optimality notion.

The lower and upper optimal assignments are also related to the necessarily and possibly of at least preference α assignments as follows.

Proposition 7. *Given an IVSCSP P and the optimal preference l_{opt} of $ws(P)$,*

- *$Nec(P, \alpha) \neq \emptyset$ iff $\alpha \leq l_{opt}$;*
- *if $\alpha \leq l_{opt}$, $LO(P) \subseteq Nec(P, \alpha)$;*
- *let α_* be the maximum α such that there exists a solution in $Nec(P, \alpha)$, then $\alpha_* = l_{opt}$ and $Nec(P, \alpha_*) = LO(P)$, and so $Nec(P, \alpha_*) \subseteq PO(P)$.*

Proof: Let us show the first item of the theorem. To show that $Nec(P, \alpha) \neq \emptyset$ iff $\alpha \leq l_{opt}$, we first prove that, if $Nec(P, \alpha) \neq \emptyset$, then $\alpha \leq l_{opt}$. If $Nec(P, \alpha) \neq \emptyset$, then there is a solution, say s, such that $\text{pref}(Q_i, s) \geq \alpha$ for every scenario Q_i of P and so also for the worst scenario. Hence, $l_{opt} \geq \text{pref}(ws(P), s) \geq \alpha$. Therefore, $l_{opt} \geq \alpha$. We now show that, if $\alpha \leq l_{opt}$, then $Nec(P, \alpha) \neq \emptyset$. If $Nec(P, \alpha) = \emptyset$, then for every solution s we have that $\text{pref}(Q_i, s) < \alpha$ for some scenario Q_i. This holds also for any solution, say s^*, such that $\text{pref}(ws(P), s^*) = l_{opt}$, and so $l_{opt} = \text{pref}(ws(P), s^*) < \alpha$.

We now show the second item of the theorem: given $\alpha \leq l_{opt}$, $LO(P) \subseteq Nec(P, \alpha)$. If $LO(P) \not\subseteq Nec(P, \alpha)$, then there is a solution, say s, such that $s \in LO(P) \setminus Nec(P, \alpha)$. Since $s \in LO(P)$, $\text{pref}(ws(P), s) = l_{opt}$. Since $s \notin Nec(P, \alpha)$, then $\text{pref}(Q_i, s) < \alpha$ for some scenario Q_i, and so, as $ws(P)$ is the worst scenario, $l_{opt} = \text{pref}(ws(P), s) \leq \text{pref}(Q_i, s) < \alpha$. Therefore, $l_{opt} < \alpha$.

We now show, by contradiction, that $\alpha_* = l_{opt}$. If $\alpha_* > l_{opt}$, then, by the previous part of the proof, $Nec(P, \alpha_*) = \emptyset$, that is a contradiction because α_* is the maximum α such that $Nec(P, \alpha) \neq \emptyset$. If $\alpha_* < l_{opt}$, then α_* is not the maximum α such that $Nec(P, \alpha) \neq \emptyset$, since such a value is l_{opt}, and so we have a contradiction.

We now prove that, if $\alpha_* = l_{opt}$, then $Nec(P, \alpha_*) = LO(P)$. Let s be a complete assignment to the variables of P. If $s \in Nec(P, l_{opt})$, then for every scenario Q, $\text{pref}(Q, s) \geq l_{opt}$ and so also for the worst scenario. Therefore, as l_{opt} is the optimal preference of the worst scenario, $s \in LO(P)$. If $s \in LO(P)$, then $\text{pref}(ws(P), s) = l_{opt}$. Since for every scenario Q, $\text{pref}(Q, s) \geq \text{pref}(ws(P), s) = l_{opt}$, then $s \in Nec(P, l_{opt})$.

Since $Nec(P, \alpha_*) = LO(P)$ and since, by Proposition 6, $LO(P) \subseteq PO(P)$, then $Nec(P, \alpha^*) \subseteq PO(P)$. □

Thus, in general, $Nec(P, \alpha)$ is not empty only if α is at most the optimal preference of the worst scenario, and in such a case every lower-optimal solution is in $Nec(P, \alpha)$. Moreover, if we consider a particular value of α, also the converse holds. Therefore, in this case the necessarily of at least preference α solutions are lower-optimal solutions and thus they are possibly optimal solutions.

Moreover, a solution is in $Pos(P, \alpha)$ only if α is at most the optimal preference of the best scenario, and in such a case, for a particular value of α, the possibly of at least preference α solutions coincide with the upper optimal solutions, and thus they are possibly optimal solutions.

Proposition 8. *Given an IVSCSP P and an assignment s to the variables of P,*

- *s is in $Pos(P, \alpha)$ if and only if $\alpha \leq U(s)$;*
- *let α^* be the maximum α such that $Pos(P, \alpha)$ is not empty, then $Pos(P, \alpha^*) = UO(P)$, and so $Pos(P, \alpha^*) \subseteq PO(P)$.*

Proof: We first show that s is in $Pos(P, \alpha)$ if and only if $\alpha \leq U(s)$. If $s \in Pos(P, \alpha)$, then there is a scenario where $\text{pref}(Q, s) \geq \alpha$. By Theorem 1, we know that $U(s)$ is the highest preference associated to s in any scenario, then $U(s) \geq \text{pref}(Q, s)$ and so $U(s) \geq \alpha$. If $\alpha \leq U(s)$, then, by Theorem 1, there is a scenario Q, where $\text{pref}(Q, s) = U(s)$. Since $U(s) \geq \alpha$, then $s \in Pos(P, \alpha)$.

We now show that $Pos(P, \alpha^*) = UO(P)$. If $s \in Pos(P, \alpha^*)$, then there is a scenario Q where $\text{pref}(Q, s) \geq \alpha^*$. Since α^* is the maximum α such that $Pos(P, \alpha) \neq \emptyset$, then, $\alpha^* = u_{opt}$, where u_{opt} is the optimal preference in the best scenario. Hence, $s \in UO(P)$. If $s \in UO(P)$, then $\text{pref}(Q, s) = u_{opt}$, hence in the best scenario $\text{pref}(bs(P), s) = u_{opt}$ and thus $s \in Pos(P, \alpha^*)$, where $\alpha^* = u_{opt}$.

Since by Proposition 6, $UO(P) \subseteq PO(P)$, then $Pos(P, \alpha^*) \subseteq PO(P)$. □

5.4 Lower and upper lexicographically-optimal assignments

We now introduce two optimality notions that refine the lower and upper optimal notions.

Definition 13 (Lower and upper lexicographically-optimal). *Given an IVSCSP $P = \langle V, D, C, S\rangle$ and an assignment s to the variables in V, s is lower (resp., upper) lexicographically-optimal iff, for every other complete assignment s', either $L(s) > L(s')$ (resp., $U(s) > U(s')$), or $L(s) = L(s')$ and $U(s) \geq U(s')$ (resp., $U(s) = U(s')$ and $L(s) \geq L(s')$).*

Lower (resp., upper) lexicographically-optimal assignments are those optimal assignments of the worst scenario (resp., best scenario) that are the best ones in the best scenario (resp., in the worst scenario). We denote with $LLO(P)$ (resp., $ULO(P)$) the set of the lower (resp., upper) lexicographically-optimal assignments of P. The IVSCSP P of Figure 1 has $LLO(P) = ULO(P) = \{s_1\}$.

Proposition 9. *Given an IVSCSP P,*

– $LLO(P) \subseteq LO(P)$ *and so* $LLO(P)$ *is never empty;*
– $ULO(P) \subseteq UO(P)$ *and so* $ULO(P)$ *is never empty;*
– $ID(P) \subseteq (LLO(P) \cap ULO(P)) = WID(P)$.

Proof: We show that $LLO(P) \subseteq LO(P)$. The relation $ULO(P) \subseteq UO(P)$ can be shown similarly. If $s \in LLO(P)$, then, by definition of $LLO(P)$, $L(s) > L(s')$ or $(L(s) = L(s')$ and $U(s) \geq U(s'))$ for every other s', hence $L(s) \geq L(s')$ for every other s' and so $s \in LO(P)$.

Since $LLO(P)$ is contained in $LO(P)$ and, by Proposition 5, $LO(P)$ is never empty, then $LLO(P)$ is never empty. Similarly, it is possible to show that $ULO(P)$ is never empty.

We now prove that $(LLO(P) \cap ULO(P)) = WID(P)$. We first show that $(LLO(P) \cap ULO(P)) \subseteq WID(P)$. If $s \in (LLO(P) \cap ULO(P))$, then, by definition of $LLO(P)$, $L(s) \geq L(s')$ for every other s' and, by definition of $ULO(P)$, $U(s) \geq U(s')$ for every other s', hence $s \in WID(P)$. We now show that $WID(P) \subseteq (LLO(P) \cap ULO(P))$. If $s \in WID(P)$, then, by definition of $WID(P)$, $L(s) \geq L(s')$ and $U(s) \geq U(s')$ for every other s'. It could happen that $(L(s) > L(s')$ and $U(s) > U(s'))$ or $(L(s) > L(s')$ and $U(s) = U(s'))$ or $(L(s) = L(s')$ and $U(s) > U(s'))$ or $(L(s) = L(s')$ and $U(s) = U(s'))$ for every other s'. If $L(s) > L(s')$ and $U(s) > U(s')$ for every other s', then $s \in LLO(P) \cap ULO(P)$ by the first part of the definitions of $LLO(P)$ and $ULO(P)$. If $L(s) > L(s')$ and $U(s) = U(s')$ for every other s' , then $s \in LLO(P) \cap ULO(P)$ by the first part of the definition of $LLO(P)$ and by the second part of the definition of $ULO(P)$. If $L(s) = L(s')$ and $U(s) > U(s')$ for every other s', then $s \in LLO(P) \cap ULO(P)$ by the second part of the definition of $LLO(P)$ and by the first part of the definition of $ULO(P)$. If $L(s) = L(s')$ and $U(s) = U(s')$ for every other s', then $s \in LLO(P) \cap ULO(P)$ by the second part of the definitions of $LLO(P)$ and $ULO(P)$. □

Since lower and upper lexicographically-optimal solutions are refinements of lower and upper optimal solutions, they are possibly optimal solutions as well. However, the converse does not hold in general.

Proposition 10. *Given an IVSCSP P, $(LLO(P) \cup ULO(P)) \subseteq PO(P)$.*

Proof: We know, by Proposition 9, that $LLO(P) \subseteq LO(P)$ and $ULO(P) \subseteq UO(P)$. Since, by Proposition 6, $LO(P)$ and $UO(P)$ are contained $PO(P)$, then also $LLO(P)$ and $ULO(P)$ are contained in $PO(P)$. □

5.5 Interval-optimal assignments

Until now we have considered optimality notions that are stronger than the possibly optimal notion. In the attempt to fully characterize possibly optimal solutions, we now consider an interval-based optimality notion that is weaker than the lower and upper optimality notions.

Definition 14 (interval-optimal). *Given an IVSCSP $P = \langle V, D, C, S\rangle$ and an assignment s to the variables in V, s is defined to be interval-optimal iff, for every other complete assignment s', $L(s) \geq L(s')$ or $U(s) \geq U(s')$.*

An interval-optimal assignment is a complete assignment with either a higher or equal lower bound, or a higher or equal upper bound, w.r.t. all other assignments. This means that, for every other complete assignment, it must be better than, or equal to it in either the worst or the best scenario. We denote with $IO(P)$ the set of the interval optimal assignments of P. The IVSCSP P of Figure 1 has $IO(P) = \{s_1, s_2, s_4\}$.

Proposition 11. *Given an IVSCSP P, $(UO(P) \cup LO(P)) \subseteq IO(P)$ and so $IO(P)$ is never empty.*

Proof: Let s be a complete assignment to the variables of P. Suppose that $s \in UO(P) \cup LO(P)$. There are two cases, (i) $s \in UO(P)$, and (ii) $s \in LO(P)$. Suppose (i) that $s \in UO(P)$. Then $U(s) \geq U(s')$ for every other complete assignment s' and so $s \in IO(P)$. Similarly, (ii) if $s \in LO(P)$ then $L(s) \geq L(s')$ for every other s', hence $s \in IO(P)$.

Since $(UO(P) \cup LO(P)) \subseteq IO(P)$ and, by Proposition 5, $LO(P)$ and $UO(P)$ are never empty, then $IO(P)$ is never empty. □

The interval-optimal solutions are possibly optimal solutions, but the converse does not hold in general, as shown in the following proposition. Therefore, also the interval-optimality notion is stronger than the possible optimality notion.

Proposition 12. *Given an IVSCSP P, if the c-semiring is strictly monotonic or idempotent, then $IO(P) \subseteq PO(P)$. Moreover, $PO(P) \not\subseteq IO(P)$.*

Proof: Let s be a complete assignment to the variables of P.

Let us consider a strictly monotonic c-semiring. We know, by Theorem 10, that $s \in PO(P)$ iff $s \in Opt(Q^s)$, where Q^s is the scenario where all the preferences of tuples in s are set to their upper bound and all other tuples are associated to the lower bound of their preferences. We now show that, if $s \in IO$, then $s \in Opt(Q^s)$ and so, by Theorem 10, $s \in PO(P)$. Assume that $s \notin Opt(Q^s)$, we will show that $s \notin IO(P)$. If $s \notin Opt(Q^s)$, then there is a solution s' such that $\mathrm{pref}(Q^s, s') > \mathrm{pref}(Q^s, s)$.

- If s has no tuples in common with s', then, by construction of Q^s, $\mathrm{pref}(Q^s, s') = L(s')$ and $\mathrm{pref}(Q^s, s) = U(s)$. Since $\mathrm{pref}(Q^s, s') > \mathrm{pref}(Q^s, s)$, and for every solution its lower bound is lower than or equal to its upper bound, then $U(s') \geq L(s') > U(s) \geq L(s)$ and so $U(s') > U(s)$ and $L(s') > L(s)$, that implies that $s \notin IO(P)$.
- If s has some tuple in common with s', then, $\mathrm{pref}(Q^s, s') = \lambda \times u$, and $\mathrm{pref}(Q^s, s) = \mu \times u$, where λ (resp., μ) is the combination of the preferences of the tuples that are in s' but not in s (resp., in s but not in s'), and u is the combination of the preferences of the tuples that are both in s and in s'. By hypothesis, $\mathrm{pref}(Q^s, s') > \mathrm{pref}(Q^s, s)$, i.e., $\lambda \times u > \mu \times u$. By construction of Q^s, $U(s') \geq \lambda \times u > \mu \times u = U(s)$, and so $U(s') > U(s)$. Moreover, since the combination operator is monotonic, if $\lambda \times u > \mu \times u$, then $\lambda > \mu$. In fact, if $\lambda \leq \mu$, by monotonicity, $\lambda \times u \leq \mu \times u$. Let us denote with u' (resp., μ') the combination of

the lower bounds of the preferences of the tuples that are both in s and in s' (resp., in s but not in s'). Then, by strict monotonicity and by construction of Q^s, $L(s') = \lambda \times u' > \mu \times u' \geq \mu' \times u' = L(s)$, and so $L(s') > L(s)$. Therefore, if s has some tuple in common with s', then $U(s') > U(s)$ and $L(s') > L(s)$, i.e., $s \notin IO(P)$.

Let us now consider an idempotent c-semiring. We want to show that if $s \in IO(P)$, then $s \in PO(P)$. We will show that, if $s \in IO(P)$, then $s \in Opt(Q^*)$, where Q^* is the scenario such that all the preferences of the tuples of s are set to $U(s)$, if $U(s)$ is contained in their preference interval, and to their upper bound, if $U(s)$ is not contained in their preference interval, and all other tuples are associated to the lower bound of their preferences. First, we show that $\text{pref}(Q^*, s) = U(s)$. Then, we show that $\text{pref}(Q^*, s) \geq \text{pref}(Q^*, s')$, for every other solution s' that has no tuples in common with s and for every solution s' that has some tuple in common with s.

- $\text{pref}(Q^*, s) = U(s)$, by construction of Q^*, by Theorem 1 and by idempotency. In fact, by Theorem 1, $\text{pref}(Q^*, s) \leq U(s)$. Moreover, $\text{pref}(Q^*, s) \not< U(s)$. In fact, we now show that $\text{pref}(Q^*, s)$ is given by the combination of the preferences that are all greater than or equal to $U(s)$. By construction of Q^* we have two results. (1) Every tuple of s in Q^* with preference interval that contains $U(s)$ is assigned to $U(s)$ and, by definition of $U(s)$ and by idempotency, there must be at least one of these preferences. (2) Every tuple with preference interval that does not contain $U(s)$ is assigned to its upper bound that must be a value greater than $U(s)$, since, by definition of $U(s)$, the upper bound of every tuple of s must be greater than or equal to $U(s)$, otherwise the upper bound of s is not $U(s)$ but a value lower than $U(s)$, that is a contradiction. Therefore, $\text{pref}(Q^*, s) \not< U(s)$ and so $\text{pref}(Q^*, s) = U(s)$.
- If s has no tuples in common with s', then, by construction of Q^*, $\text{pref}(Q^*, s') = L(s')$ and $\text{pref}(Q^*, s) = U(s)$. Since $s \in IO(P)$, then $L(s) \geq L(s')$ or $U(s) \geq U(s')$. If $L(s) \geq L(s')$, then $\text{pref}(Q^*, s) = U(s) \geq L(s) \geq L(s') = \text{pref}(Q^*, s')$. If $U(s) \geq U(s')$, then $\text{pref}(Q^*, s) = U(s) \geq U(s') \geq L(s') = \text{pref}(Q^*, s')$.
- If s has some tuple in common with s', then, by construction of Q^* $\text{pref}(Q^*, s') \leq U(s) = \text{pref}(Q^*, s)$.

Therefore, for every solution s', $\text{pref}(Q^*, s') \leq U(s) = \text{pref}(Q^*, s)$. Hence, s is optimal in Q^* and so $s \in PO(P)$.

$PO(P) \not\subseteq IO(P)$. In fact, assume to have an IVSCSP over a fuzzy c-semiring, where there is only one variable x with three values in its domain, say x_1, x_2, and x_3, with preference intervals respectively $[0.4, 0.6]$, $[0.5, 0.7]$, and $[0.5, 0.8]$. Then, $x_1 \notin IO(P)$, because $L(x_1) < L(x_2)$ and $U(x_1) < U(x_2)$. However, $x_1 \in PO(P)$, because x_1 is optimal in the scenario where we associate to x_1 the value 0.6 and to x_2 and x_3 the value 0.5.

□

5.6 Summary of the various notions of optimality and of their relations

The various notions of optimality defined above are summarized in Table 1. For each notion, we refer to a solution s and we describe compactly when s belongs to each of the optimality sets.

Table 1. Optimality notions.

Optimality notions	*Definition*
$NO(P)$	$s \in Opt(Q), \forall Q \in Sc(P)$
$PO(P)$	$s \in Opt(Q), \exists Q \in Sc(P)$
$Nec(P, \alpha)$	$pref(Q, s) \geq \alpha, \forall Q \in Sc(P)$
$Pos(P, \alpha)$	$pref(Q, s) \geq \alpha, \exists Q \in Sc(P)$
$ID(P)$	$L(s) \geq U(s'), \forall s' \in Sol(P)$
$WID(P)$	$L(s) \geq L(s')$ and $U(s) \geq U(s'), \forall s' \in Sol(P)$
$LO(P)$	$L(s) \geq L(s'), \forall s' \in Sol(P)$
$UO(P)$	$U(s) \geq U(s'), \forall s' \in Sol(P)$
$LLO(P)$	$L(s) > L(s')$ or ($L(s) = L(s')$ and $U(s) \geq U(s')$), $\forall s' \in Sol(P)$
$ULO(P)$	$U(s) > U(s')$ or ($U(s) = U(s')$ and $L(s) \geq L(s')$), $\forall s' \in Sol(P)$
$IO(P)$	$L(s) \geq L(s')$ or $U(s) \geq U(s'), \forall s' \in Sol(P)$

The set-based relations between the various optimality notions are described in Figure 3.

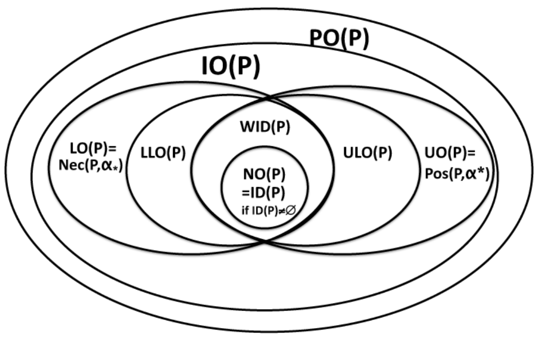


Fig. 3. Relation among optimality sets.

5.7 An example: meeting scheduling problems

To better explain how to use the various optimality notions introduced in the previous sections, we consider an example of a class of problems, related to meeting scheduling.

The meeting scheduling problem is a benchmark for CSPs [13], and we have adapted it to allow also for preference intervals.

A meeting scheduling problem (MSP) is informally the problem of scheduling some meetings by allowing the participants to attend all the meetings they are involved in. More formally, a MSP can be described by

- a set of agents;
- a set of meetings, each with a location and a duration;
- a set of time slots where meetings can take place;
- for each meeting, a subset of agents that are supposed to attend such a meeting;
- for each pair of locations, the time to go from one location to the other one.

Typical simplifying assumptions concern having the same duration for all meetings (one time slot), and the same number of meeting for each agent. To solve a MSP, we need to allocate each meeting in a time slot in a way that each agent can participate in his meetings. The only way that an agent cannot participate has to do with the time needed to go from the location of a meeting to the location of his next meeting.

The MSP can be easily seen as a CSP: variables represent meetings and variable domains represent all time slots. Each constraint between two meetings model the fact that one or more agents must participate in both meetings, and it is satisfied by all pairs of time slots that allow the participation to both meetings according to the time needed to pass between the corresponding locations. For this reason, it is often used as a typical benchmark for CSPs.

For our purposes, we consider a generalization of the MSP, called IVMSP, where there is a chair, who is in charge of the meeting scheduling, and who declares his preferences over the variable domains and over the compatible pairs of time slots in the binary constraints. The preferences over the variable domains can model the fact that the chair prefers some time slots to others for a certain meeting. On the other hand, the preferences in the binary constraints can model a preference for certain feasible pairs of time slots, over others, for the two meetings involved in the constraint.

Such preferences can be exact values when the chair works with complete information. However, at the time the meeting scheduling has to be done, it may be that some information, useful for deciding the preferences, is still missing. For example, the chair could have invited agents to meetings, but he does not yet know who will accept his invitations. As other examples, weather considerations or the presence of other events in the same time slots may affect the preferences. Because of this uncertainty, some preferences may be expressed by using an interval of values, which includes all preference values that are associated to all possible outcomes of the uncertain events.

Since MSPs can be expressed as CSPs, it is thus clear that IVMSPs can be expressed as IVSCSPs. The problem of solving an IVMSP concerns finding time slots for the meetings such that all agents can participate and, among all possible solutions, to choose an optimal one according to some optimality criteria. We will now consider several of the optimality notions defined above and describe their use in this class of problems.

In this context, given an IVMSP P, necessarily optimal solutions (i.e., solutions in $NO(P)$) are meeting schedulings that are optimal no matter how the uncertainty is resolved. Thus, if there is at least one of such solutions, this is certainly preferred to any other. By working with the optimality notions defined over intervals, to find a solution in $NO(P)$, we may try to find a solution in $ID(P)$, given that solutions in $ID(P)$, if any, coincide with solutions in $NO(P)$. Otherwise, if $ID(P)$ is empty, and given that $NO(P)$ is included in $WID(P)$, we may look for a solution in $WID(P)$. We recall that solutions in $ID(P)$ are meeting schedulings where the preference interval of the optimal solution is above the preference intervals of all other solutions, while solutions in $WID(P)$ have the upper bound of their preference interval above the upper bounds of the preference intervals of all other solutions, and the same for the lower bound.

Solutions in $Nec(P, \alpha_*)$ are also attractive, because they guarantee a preference level of α_* in all scenarios. Since $LO = Nec(P, \alpha_*)$, we may find a solution in $LO(P)$, that is, a solution which is optimal in the worst scenario. This solution will guarantee the chair against the uncertainty of the problem by assuring a certain level of overall preference. This notion can be useful if the chair is pessimistic, because such solutions provide a preference guarantee over all scenarios. However, such a guaranteed preference level may be very low.

If instead the chair is optimistic, he may ask for a solution in $Pos(P, \alpha^*)$, that is, a solution with the highest preference level in some scenario. Since $UO(P) = Pos(P, \alpha_*)$, we may find a solution in $UO(P)$, that is, a solution which is optimal in the best scenario.

When looking for solutions in $LO(P)$ and $UO(P)$, we may want to be as close as possible to solutions in $NO(P)$, as $NO(P)$ is included in $LO(P)$ and $UO(P)$. To do this, we can try to find solutions in $LLO(P)$ or $ULO(P)$, respectively. For example, solutions in $LLO(P)$ are solutions in $LO(P)$ that have the highest upper bound of their preference interval. This means that, depending on how the uncertainty is resolved, they give more hope of achieving a higher level of preference.

6 Finding and testing interval-based optimal assignments

In this section we analyze how to determine if a complete assignment is one of the different kinds of optimal assignments previously defined in Section 5, and how to find such optimal assignments. These results will be useful to find and test possibly and necessarily optimal solutions.

6.1 Lower and upper optimal assignments

It is easy to show that, by following directly the definitions of lower and upper optimal assignments, the lower (resp., upper) optimal solutions coincide with the optimal elements of the worst (resp., best) scenario.

Theorem 2. *Given an IVSCSP P, $LO(P) = Opt(ws(P))$ and $UO(P) = Opt(bs(P))$.*

Proof: We show that $LO(P) = Opt(ws(P))$. Let s be a solution of P. If $s \in LO(P)$, then $L(s) \geq L(s')$ for every other solution s', hence if we consider $ws(P)$, i.e., the worst scenario of P, that is the scenario where we fix all the preference intervals to their lower bound, then $\text{pref}(ws(P), s) = L(s)$ and so $\text{pref}(ws(P), s) \geq \text{pref}(ws(P), s')$ for every other solution s', hence $s \in Opt(ws(P))$. If $s \in Opt(ws(P))$, then $\text{pref}(ws(P),$

$s) \geq \mathrm{pref}(ws(P), s')$ for every other solution s' of P, that is, by definition of worst scenario, $L(s) \geq L(s')$ for every s' and so $s \in LO(P)$. Similarly, it is possible to show that $UO(P) = Opt(bs(P))$. □

A lower-optimal solution is a complete assignment whose lower bound is greater than or equal to the lower bound of every other complete assignment. Thus, it is a complete assignment that is better than or equal to all other assignments in the scenario obtained by replacing every interval with its lower bound, i.e., the worst scenario.

Thus, finding a lower-optimal (resp. upper-optimal) solution is as complex as solving an SCSP. This holds also for testing if an assignment s is in $LO(P)$ (resp. in $UO(P)$), since it is enough to solve the SCSP representing the worst or the best scenario and to check if the preference of the optimal solution coincides with $L(s)$ (resp. $U(s)$).

6.2 Interval optimal assignments

To find an interval optimal assignment, it is sufficient to find a lower-optimal solution or an upper-optimal solution, because $(UO(P) \cup LO(P)) \subseteq IO(P)$, and neither $UO(P)$ nor $LO(P)$ can be empty. Thus, finding assignments of $IO(P)$ can be achieved by solving an SCSP.

To test if a solution is interval optimal, if the c-semiring is idempotent, we can exploit the preference levels of the best and worst scenarios, as stated by the following theorem.

Theorem 3. *Given an IVSCSP P defined over an idempotent c-semiring, and an assignment s, we have $s \in IO(P)$ iff the CSP obtained by joining*[1] *$scut_{L(s)}(ws(P))$ and $scut_{U(s)}(bs(P))$ has no solution.*

Proof: Let us denote with Q the CSP defined in the theorem. We first show that, if Q has no solution, then $s \in IO(P)$. Suppose that $s \notin IO(P)$. Then there exists some complete assignment s' with $L(s') > L(s)$ and $U(s') > U(s)$. Then $pref(ws(P), s') = L(s') > L(s)$ and $pref(bs(P), s') = U(s') > U(s)$, so s' is a solution of Q. We now show that, if $s \in IO(P)$, then Q has no solution. If Q has a solution, say s^*, then, by definition of Q, $L(s^*) > L(s)$ and $U(s^*) > U(s)$, and so $s \notin IO(P)$. □

In fact, all and only the solutions of such a CSP strictly dominate s with respect to both the lower and the upper bound. Thus, testing membership in $IO(P)$ when the semiring is idempotent amounts to solving a CSP.

More generally (that is, even if the combination operator is not idempotent), we can test interval optimality by checking if a suitably defined SCSP has solutions with preference above certain threshold.

Theorem 4. *Given an IVSCSP P and an assignment s, let l_{opt} and u_{opt} be the optimal preferences of the worst and best scenario. Then, $s \in IO(P)$ iff at least one*

[1] The join of two CSPs P_1 and P_2 is the CSP whose set of variables (resp., constraints) is given by the union of the sets of variables (resp., constraints) of P_1 and P_2.

of the following conditions holds: (1) $L(s) = l_{opt}$; (2) $U(s) = u_{opt}$; (3) the SCSP Q with the same variables, domains, and constraint topology as P, defined on the c-semiring $\langle(A \times A), (+, +), (\times, \times), (\mathbf{0}, \mathbf{0}), (\mathbf{1}, \mathbf{1})\rangle$, where the preference of each tuple in each constraint is set to the pair containing the lower and upper bound of its interval in P, has no solution s' with preference pair $(L(s'), U(s'))$ pointwise greater than $(L(s), U(s))$, i.e., such that $L(s') > L(s)$ and $U(s') > U(s)$.

Proof: We first show that if $L(s) = l_{opt}$, $U(s) = u_{opt}$, or Q has no solution with preference greater than $(L(s), U(s))$, then $s \in IO(P)$. If $L(s) = l_{opt}$ (resp., $U(s) = u_{opt}$), then $L(s) \geq L(s')$ (resp., $U(s) > U(s')$) for every other solution s', hence $s \in LO(P)$ (resp., $s \in UO(P)$) and so, since $LO(P) \cup UO(P) \subseteq IO(P)$, $s \in IO(P)$. If Q has no solution with preference greater than $(L(s), U(s))$, then $s \in IO(P)$. In fact, if $s \notin IO(P)$, then there is a solution, say s^*, such that $L(s^*) > L(s)$ and $U(s^*) > U(s)$, and so Q has a solution with preference greater than $(L(s), U(s))$.

We now show, that if $s \in IO(P)$, then $L(s) = l_{opt}$, $U(s) = u_{opt}$, or Q has no solution with preference greater than $(L(s), U(s))$. If $L(s) \neq l_{opt}$, $U(s) \neq u_{opt}$ and Q has a solution s^* with preference greater than $(L(s), U(s))$, then, by definition of Q, the preference of $(L(s^*), U(s^*))$ is greater than the preference of $(L(s), U(s))$, hence $L(s^*) > L(s)$ and $U(s^*) > U(s)$ and so $s \notin IO(P)$. □

The first two conditions simply check if s is either lower or upper optimal. The second condition is satisfied when there is no solution better than s on both bounds. Notice that this can be checked for example by running branch and bound on Q with a strict bound equal to $(L(s), U(s))$. Therefore, testing membership in $IO(P)$ with any c-semiring can be achieved by solving at most three SCSPs.

6.3 Lower and upper lexicographically optimal assignments

To find the lower-lexicographically optimal solutions of an IVSCSP P we consider the optimal solutions of a suitable SCSP, as described by the following theorem.

Theorem 5. *Given an IVSCSP P over a strictly monotonic c-semiring S, let us consider the SCSP Q with the same variables, domains, and constraint topology as P, and defined over the c-semiring $\langle A \times A, \max_{lex}, (\times, \times), (\mathbf{0}, \mathbf{0}), (\mathbf{1}, \mathbf{1})\rangle$. The binary operation $\max_{lex}$ is defined to be the maximum with respect to the ordering $\succeq_{lex}$ defined as follows: for each $(a, a'), (b, b') \in (A \times A)$, $(a, a') \succeq_{lex} (b, b')$ iff $a >_S b$ or $a = b$ and $a' \geq_S b'$. For each tuple in each constraint of Q, its preference is set to the pair containing the lower and upper bound of its interval in P. Then, $LLO(P) = Opt(Q)$.*

Proof: We first show that $LLO(P) \subseteq Opt(Q)$. If $s \in LLO(P)$, then $s \in Opt(Q)$. In fact, if $s \notin Opt(Q)$, then, there is a solution, say s', of Q such that $\text{pref}(Q, s') > \text{pref}(Q, s)$, that is, by definition of preference given in the theorem, $(L(s'), U(s')) \succ_{lex} (L(s), U(s))$, that is, by definition of $\succ_{lex}$, either $L(s') > L(s)$ or ($L(s') = L(s)$ and $U(s') > U(s)$), and so $s \notin LLO(P)$.

We now show that $Opt(Q) \subseteq LLO(P)$. If $s \in Opt(Q)$, then $\text{pref}(Q, s') \geq \text{pref}(Q, s)$, for every s', that is, $(L(s'), U(s')) \succeq_{lex} (L(s), U(s))$, for every other s', that is, for every other s', either $L(s') > L(s)$ or ($L(s') = L(s)$ and $U(s') \geq U(s)$), and so $s \in LLO(P)$.

Note that the assumption of strict monotonicity of S guarantees that the structure defined in the theorem $\langle A \times A, \max_{lex}, (\times, \times), (\mathbf{0}, \mathbf{0}), (\mathbf{1}, \mathbf{1}) \rangle$ is a c-semiring. If we don't make this assumption, then distributivity property does not hold and so the structure above is not a c-semiring. □

In words, the first component of the pairs in the semiring of Theorem 5 is the most important, and the second one is used to break ties. To find the upper-lexicographically optimal solutions, it is sufficient to consider the same SCSP as defined above except for the ordering which considers the second component as the most important. Thus, finding assignments in $LLO(P)$ and $ULO(P)$ can be achieved by solving one SCSP.

To test if a solution s is in $LLO(P)$, it is enough to find the preference pair, say $(p1, p2)$, of an optimal solution of the SCSP defined above and to check if $(L(s), U(s)) = (p1, p2)$. Similarly to test if a solution is in $ULO(P)$.

6.4 Weakly interval dominant assignments

We know that $WID(P) = LO(P) \cap UO(P)$. Thus a straightforward, but costly, way to find a solution in $WID(P)$ is to compute all the optimal solutions of the best and the worst scenario and to check if there is a solution in the intersection of the two sets. However, if the c-semiring is idempotent, this is not necessary, as shown by the following theorem.

Theorem 6. *Given an IVSCSP P defined over an idempotent c-semiring, and l_{opt} and u_{opt} as defined above, an assignment s is in $WID(P)$ iff it is a solution of the CSP obtained by joining $cut_{l_{opt}}(ws(P))$ and $cut_{u_{opt}}(bs(P))$.*

Proof: Let us denote with Q the CSP described in the theorem. We first show that, if s is a solution of Q, then $s \in WID(P)$. If s is a solution of Q, then, by definition of Q, s is a solution of the CSP $cut_{l_{opt}}(ws(P))$ obtained from the worst scenario by allowing only the tuples with preference greater than or equal to l_{opt}, hence, by definition of l_{opt}, $L(s) \geq L(s')$ for every other solution s'. Moreover, by definition of Q, s is also a solution of the CSP $cut_{u_{opt}}(bs(P))$ obtained from the best scenario by allowing only the tuples with preferences greater than or equal to u_{opt}. Hence, by the definition of u_{opt}, $U(s) \geq U(s')$, for every other s'. Therefore, if s is a solution of Q, then $L(s) \geq L(s')$ and $U(s) \geq U(s')$ for every other s', and so $s \in WID(P)$.

We now show that, if $s \in WID(P)$, then s is a solution of Q. If s is not a solution of Q, then $L(s) < l_{opt}$ or $U(s) < u_{opt}$. If $L(s) < l_{opt}$ (resp., $U(s) < u_{opt}$), then $L(s) < L(s')$ (resp., $U(s) < U(s')$) for any solution s' such that $\mathrm{pref}(ws(P), s') = l_{opt}$ (resp., $\mathrm{pref}(bs(P), s') = u_{opt}$). Therefore, $s \notin WID(P)$. □

In words, any solution of the join CSP is optimal both in the worst and in the best scenario and this implies that it is undominated on both bounds. Thus, if the c-semiring is idempotent, finding a weakly interval dominant solution amounts to solving two SCSPs and one CSP. Moreover, to test whether a solution s is in $WID(P)$, it is sufficient to check if $L(s) = l_{opt}$ and $U(s) = u_{opt}$, which amounts to solving two SCSPs.

6.5 Interval dominant assignments

To find an assignment in $ID(P)$, we can use Proposition 5. Thus, if $l_{opt} = u_{opt}$, then it is sufficient to find a lower-optimal solution. If instead $l_{opt} < u_{opt}$ then, if $|UO(P)| \geq 2$, then we know that $ID(P) = \emptyset$. Moreover, if $|UO(P)| = 1$ (let us call s this single solution), if $L(s) \neq l_{opt}$ then we know that $ID(P) = \emptyset$.

If the c-semiring is idempotent, cuts can be exploited in the same style as above, to build a suitably defined CSP, leading to a sound and complete procedure to find an assignment, if any, in $ID(P)$.

Theorem 7. *Given an IVSCSP P over an idempotent c-semiring, and l_{opt} as defined above, if $scut_{l_{opt}}(bs(P))$ has no solution, then $ID(P) = LO(P)$. If $scut_{l_{opt}}(bs(P))$ has one solution, say s, and $L(s) = l_{opt}$, then this solution is the only one in $ID(P)$. Otherwise, $ID(P) = \emptyset$.*

Proof: Let us denote with Q the CSP $scut_{l_{opt}}(bs(P))$. We first show that if Q has no solution, then $ID(P) = LO(P)$. If Q has no solution, then, since Q is the CSP obtained by the best scenario by allowing only tuples with preference greater than l_{opt}, there is no solution with upper bound greater than l_{opt}, that is, for all the solutions s' of P, $l_{opt} \geq U(s')$. To show that $ID(P) = LO(P)$ it is sufficient to show that $LO(P) \subseteq ID(P)$, since Theorem 2 implies that $ID(P) \subseteq LO(P)$. Let s be a solution of P. If $s \in LO(P)$, then $L(s) = l_{opt}$ and thus, by the reasoning above, $L(s) \geq U(s')$ for every other s', hence $s \in ID(P)$.

If Q has a solution, say s, then $U(s) > l_{opt} \geq L(s')$ for all solutions s', and so $ID(P)$ is either empty or equal to $\{s\}$. Therefore if Q has more than one solution then $ID(P)$ is empty. Suppose that Q has exactly one solution, s. If $L(s) < l_{opt}$ then $L(s) < L(s')$ for any solution s' with $L(s') = l_{opt}$, and so $L(s) < U(s')$, which implies that $s \notin ID(P)$ and so $ID(P) = \emptyset$. If $L(s) = l_{opt}$ then for any other solution s' we have $U(s') \leq l_{opt}$ (since Q has only one solution), and so $L(s) \geq U(s')$ which implies that $s \in ID(P)$ and so $ID(P) = \{s\}$. □

Performing a strict cut of the best scenario at the optimal level of the worst scenario means isolating solutions that have an upper bound higher than l_{opt}. If there is no such solution, then the upper bound of the lower-optimal solutions must coincide with their lower bound (l_{opt}). Thus, lower-optimal solutions coincide with interval dominant solutions. If, instead, such a CSP has only one solution, all other solutions must have an upper bound which is at most l_{opt}. This means that, if this solution is also lower-optimal, then it is the only interval dominant solution. Finally, if there is more than one solution with an upper bound above l_{opt}, then there cannot be any solution whose lower bound dominates the upper bound of all others and, thus, $ID(P)$ is empty.

Summarizing, when the c-semiring is idempotent, to find a solution in $ID(P)$ we need to solve an SCSP and then one CSP. Proposition 5 and Theorem 7 can also be used to test if a solution is interval dominant.

7 Finding and testing necessarily optimal and possibly optimal assignments

We will now show how to test if an assignment is possibly or necessarily optimal (or of at least preference α) and how to find these kinds of assignments. To do that, we will exploit the relation between possibly and necessarily optimal assignments and the various kinds of interval-based optimal assignments, shown in Section 5.

7.1 Necessarily optimal solutions

To find a necessarily optimal solution, we exploit the results shown in Propositions 2 and 4 (i.e., if $ID(P) \neq \emptyset$ then $NO(P) = ID(P)$, and $ID(P) \subseteq NO(P) \subseteq WID(P)$), and thus we perform the following steps:

1. If $ID(P) \neq \emptyset$, then return $s \in ID(P)$;
2. If $WID(P) = \emptyset$, then $NO(P) = \emptyset$;
3. Otherwise, return the first solution in $WID(P)$ that is necessarily optimal. If none, $NO(P) = \emptyset$

Testing if a solution is necessarily optimal when $ID(P) \neq \emptyset$ coincides with testing if it is in $ID(P)$. Otherwise, we need to test if it is an optimal solution of some suitably defined SCSPs, as shown by the following theorem.

Theorem 8. *Consider an IVSCSP P and an assignments s. Let Q_s (resp., Q^s) be the scenario where every preference associated to a tuple of s is set to its lower bound (resp., upper bound) and the preferences of all other tuples are set to their upper bound (resp., lower bound). The following results hold:*

– *If $s \in NO(P)$, then $s \in Opt(Q_s)$. Moreover, if the c-semiring is strictly monotonic, the converse holds as well: $s \in NO(P) \iff s \in Opt(Q_s)$.*
– *If $s \in NO(P)$ then, for every s', $s \in Opt(Q^{s'})$. If the c-semiring is idempotent, the converse holds as well: $s \in NO(P) \iff$ for every s', $s \in Opt(Q^{s'})$.*

Proof: We first show that, if $s \in NO(P)$, then $s \in Opt(Q_s)$. If $s \in NO(P)$, then it is optimal in all scenarios and so also in Q_s.

We now show that, if the c-semiring is strictly monotonic and if $s \in Opt(Q_s)$, then $s \in NO(P)$. If $s \in Opt(Q_s)$, then $\text{pref}(Q_s, s) \geq \text{pref}(Q_s, s')$ for every other solution s'. For every other s', let λ (resp., μ) be the combination of the preference values of tuples associated to s but not to s' (resp., associated to s' but not to s) in Q_s, and let u be the combination of the preference values of tuples associated to both s and s' in Q_s. Since, for every s', $\text{pref}(Q_s, s) \geq \text{pref}(Q_s, s')$, then for every s', $\lambda \times u \geq \mu \times u$ that implies that $\lambda \geq \mu$. In fact, if $\lambda < \mu$, then, by strict monotonicity of $\times$, then $\lambda \times u < \mu \times u$. For every scenario Q_i, for every s', let λ_i (resp., μ_i) be the combination of the preference values of tuples associated to s' but not to s (resp., associated to s' but not to s) in Q_i and let u_i be the combination of the preference values of tuples associated to both s and s' in Q_i. Since Q_s is the least favorable scenario for s, then for every scenario Q_i, $\lambda_i \times u \geq \lambda \times u$ that implies $\lambda_i \geq \lambda$. In fact, if $\lambda_i < \lambda$, then, by strict monotonicity, $\lambda_i \times u < \lambda \times u$. Since Q_s is the most favorable scenario for the tuples in s' but not in s, then $\mu \geq \mu_i$ for every scenario Q_i. Therefore, for every scenario Q_i, for every s', we have that $\lambda \geq \mu$, $\lambda_i \geq \lambda$ and $\mu \geq \mu_i$, hence, by monotonicity, $\mathrm{pref}(Q_i, s) = \lambda_i \times u_i \geq \lambda \times u_i \geq \mu \times u_i \geq \mu_i \times u_i = \mathrm{pref}(Q_i, s')$, hence s is optimal in every scenario and so $s \in NO(P)$.

If $s \in NO(P)$, then s is optimal in all the scenarios and so, for every s', s is optimal in $Q^{s'}$. If the c-semiring is idempotent and, for every s', $s \in Opt(Q^{s'})$, then $s \in NO(P)$. In fact, assume that $s \notin NO(P)$, then there is a scenario Q, where s is not optimal, i.e., there is s' such that $\mathrm{pref}(Q, s) < \mathrm{pref}(Q, s')$. We want to show that this holds also in the scenario $Q^{s'}$. If we consider the scenario Q_1 obtained from Q by putting the preference value of any tuple that is in s but not in s' to its lower bound, then, the preference of s decreases or remains the same, by monotonicity, and the preference of s' does not change. Hence, $\mathrm{pref}(Q_1, s) \leq \mathrm{pref}(Q, s) < \mathrm{pref}(Q, s') = \mathrm{pref}(Q_1, s')$, and so $\mathrm{pref}(Q_1, s) < \mathrm{pref}(Q_1, s')$. If we consider the scenario Q_2 obtained from Q_1 by setting the preference value of any tuple that is in s' but not in s to its upper bound, then the preference of s' increases or remains the same, by monotonicity, and the preference of s does not change. Hence, $\mathrm{pref}(Q_2, s) = \mathrm{pref}(Q_1, s) < \mathrm{pref}(Q_1, s') \leq \mathrm{pref}(Q_2, s')$ and so $\mathrm{pref}(Q_2, s) < \mathrm{pref}(Q_2, s')$. If we consider the scenario obtained from Q_2 by setting the preference value of the tuples that are in s and s' to their upper bound, then we have the scenario $Q^{s'}$. The preferences of the tuples that are in s and s' does not modify $\mathrm{pref}(Q_2, s)$ and $\mathrm{pref}(Q_2, s')$. In fact, since the c-semiring is idempotent, then $\mathrm{pref}(Q_2, s)$ (resp., $\mathrm{pref}(Q_2, s')$) is given by the tuple with the worst preference of s (resp., s'), and, since $\mathrm{pref}(Q_2, s) < \mathrm{pref}(Q_2, s')$, $\mathrm{pref}(Q_2, s)$ and $\mathrm{pref}(Q_2, s')$ must be given by different tuples, otherwise $\mathrm{pref}(Q_2, s) = \mathrm{pref}(Q_2, s')$. Hence, $\mathrm{pref}(Q^{s'}, s) = \mathrm{pref}(Q_2, s) < \mathrm{pref}(Q_2, s') = \mathrm{pref}(Q^{s'}, s)$. Therefore, there is a solution s' such $s' \notin Opt(Q^{s'})$. □

The intuition behind this theorem is that, in order for a solution to be necessarily optimal, it must be optimal also in its least favorable scenario, when the c-semiring is strictly monotonic, and it must be optimal in the most favorable scenario of every other solution, when the c-semiring is idempotent.

7.2 Necessarily of at least preference α solutions

By Proposition 7, we know that $s \in Nec(P, \alpha)$ if and only if $\alpha \leq L(s)$. Thus, testing whether a solution s is in $Nec(P, \alpha)$ amounts at checking this condition that takes linear time.

To find a solution in $Nec(P, \alpha)$, we know, by Proposition 7, that $Nec(P, \alpha)$ is not empty only if α is at most the optimal preference of the worst scenario, and in such a case any lower-optimal solution is in $Nec(P, \alpha)$. This amounts to solving one SCSP. However, if the c-semiring is idempotent, it is sufficient to solve one CSP, as shown by the following theorem.

Theorem 9. *Given an IVSCSP P, if the c-semiring is idempotent, then $Nec(P, \alpha)$ coincides with the set of solutions of $cut_\alpha(ws(P))$.*

Proof: Let us denote with SL the set of the solutions of $cut_\alpha(ws(P))$. We first show that $Nec(P,\alpha) \supseteq SL$ and then we show that $Nec(P,\alpha) \subseteq SL$. Let be s a solution of P. If $s \in SL$, then, since $cut_\alpha(ws(P))$ is the CSP obtained from the worst scenario of P by allowing only tuples with preference greater than or equal to α, $\mathrm{pref}(ws(P), s) \geq \alpha$, by idempotence. Since $ws(P)$ is the worst scenario of P, then $\mathrm{pref}(Q_i, s) \geq \mathrm{pref}(ws(P), s) \geq \alpha$ for every scenario Q_i and so $s \in Nec(P,\alpha)$. Therefore, $Nec(P,\alpha) \supseteq SL$. If $s \in Nec(P,\alpha)$, then $\mathrm{pref}(Q_i, s) \geq \alpha$ for every scenario Q_i and so also for the worst scenario. Hence, $\mathrm{pref}(ws(P), s) \geq \alpha$ and so, by definition of $cut_\alpha(ws(P))$, $s \in SL$. Therefore, $Nec(P,\alpha) \subseteq SL$. □

By Proposition 7, we know that $Nec(P,\alpha_*) = LO(P)$. Therefore, to find a solution in $Nec(P,\alpha_*)$, it is sufficient to find a solution of the worst scenario, and thus to solve one SCSP.

7.3 Possibly optimal solutions

To find a solution in $PO(P)$, we can observe that $LO(P)$, $UO(P)$, $LLO(P)$, and $ULO(P)$ are all contained in $PO(P)$ (Propositions 6 and 10) and they are never empty (Propositions 5 and 9).

To test if a solution is in $PO(P)$, it is sufficient to test if s is optimal in one of the two scenarios defined in the following theorem. This amounts to solving an SCSP.

Theorem 10. *Given an IVSCSP P and an assignment s to the variables of P, let Q^s be the scenario where all the preferences of tuples in s are set to their upper bound and all other tuples are associated to the lower bound of their preferences, and let Q^* be the scenario where all the preferences of the tuples of s are set to $U(s)$, if $U(s)$ is contained in their preference interval, and to their upper bound otherwise, and all other tuples are associated to the lower bound of their preferences. Then,*

- *if the c-semiring is strictly monotonic, $s \in PO(P) \iff s \in Opt(Q^s)$;*
- *if the c-semiring is idempotent, $s \in PO(P) \iff s \in Opt(Q^*)$.*

Proof: We first show that, if $s \in Opt(Q^s)$, then $s \in PO(P)$. If $s \in Opt(Q^s)$, then s is optimal in the scenario Q^s, and so $s \in PO(P)$. We now show that, if $s \in PO(P)$ then $s \in Opt(Q^s)$. If $s \in PO(P)$, then there is a scenario, say Q_i, where s is optimal, that is, $\mathrm{pref}(Q_i, s) \geq \mathrm{pref}(Q_i, s')$, for every other solution s'. Assume to use the same notations used in the proof of Theorem 8. Using these notations, since $\mathrm{pref}(Q_i, s) \geq \mathrm{pref}(Q_i, s')$, for every other solution s', then, for every other s', $\lambda_i \times u_i \geq \mu_i \times u_i$ in the scenario Q_i. This implies that, for every other s', $\lambda_i \geq \mu_i$. In fact, if $\lambda_i < \mu_i$, then, by strict monotonicity, $\lambda_i \times u_i < \mu_i \times u_i$. Since Q^s is the most favorable scenario for s, then for every scenario and so also for the scenario Q_i, by monotonicity, $\lambda \times u \geq \lambda \times u_i \geq \lambda_i \times u_i$, that implies $\lambda \geq \lambda_i$. In fact, if $\lambda < \lambda_i$, then, by strict monotonicity, $\lambda \times u_i < \lambda_i \times u$. Since Q_s is the least favorable scenario for the tuples in s' but not in s, then $\mu_i \geq \mu$ for every scenario and so also for Q_i. Hence, since for every s', $\lambda \geq \lambda_i$, $\lambda_i \geq \mu_i$, and $\mu_i \geq \mu$, then, by monotonicity, for every s', $\mathrm{pref}(Q^s, s) = \lambda \times u \geq \lambda_i \times u \geq \mu_i \times u \geq \mu \times u = \mathrm{pref}(Q^s, s')$, hence s is optimal in the scenario Q^s.

If $s \in Opt(Q^*)$, then $s \in PO(P)$. We now show that, if $s \in PO(P)$, then $s \in Opt(Q^*)$. If $s \notin Opt(Q^*)$, then there is a solution s' such that $\mathrm{pref}(Q^*, s') > \mathrm{pref}(Q^*, s)$. By construction of Q^*, by Theorem 1 and by idempotency, we have that $\mathrm{pref}(Q^*, s) = U(s)$. In fact, by Theorem 1, $\mathrm{pref}(Q^*, s) \leq U(s)$. Moreover, $\mathrm{pref}(Q^*, s) \not< U(s)$. In fact, we now show that $\mathrm{pref}(Q^*, s)$ is given by the combination of the preferences that are all greater than or equal to $U(s)$. By construction of Q^* we have two results. (1) Every tuple of s in Q^* with preference interval that contains $U(s)$ is assigned to $U(s)$ and, by definition of $U(s)$ and by idempotency, there must be at least one of these preferences. (2) Every tuple with preference interval that does not contain $U(s)$ is assigned to its upper bound that must be a value greater than $U(s)$, since, by definition of $U(s)$, the upper bound of every tuple of s must be greater than or equal to $U(s)$, otherwise the upper bound of s is not $U(s)$ but a value lower than $U(s)$, that is a contradiction. Therefore, $\mathrm{pref}(Q^*, s) \not< U(s)$ and so $\mathrm{pref}(Q^*, s) = U(s)$. If s and s' have tuples in common, by construction of Q^*, $\mathrm{pref}(Q^*, s') \leq U(s)$. In such a case, since we have shown above that $\mathrm{pref}(Q^*, s) = U(s)$, and since we are assuming that there is a solution s' such that $\mathrm{pref}(Q^*, s') > \mathrm{pref}(Q^*, s)$, then $U(s) \geq \mathrm{pref}(Q^*, s') > \mathrm{pref}(Q^*, s) = U(s)$, and so we have a contradiction. If s and s' have no tuples in common, then, for every scenario Q, $\mathrm{pref}(Q, s') \geq L(s') = \mathrm{pref}(Q^*, s') > \mathrm{pref}(Q^*, s) = U(s) \geq \mathrm{pref}(Q, s)$, and so $s \notin PO(P)$. □

In Theorem 10 we have characterized possibly optimal solutions for IVSCSPs with idempotent c-semiring and for IVCSPs with strictly monotonic c-semiring. The characterization of possibly optimal solutions for IVSCSPs with a c-semiring that is neither idempotent nor strictly monotonic is an open question.

7.4 Possibly of at least preference α solutions

We know, by Proposition 8, that, given an IVSCSP P and an assignment s, s is in $Pos(P, \alpha)$ if and only if $\alpha \leq U(s)$. Thus, to test whether a solution is in $Pos(P, \alpha)$, it is enough to check this condition, that takes linear time.

If the c-semiring is idempotent, to find a solution in $Pos(P, \alpha)$ it is sufficient to solve one CSP, as shown in the following theorem.

Theorem 11. *Given an IVSCSP P over an idempotent c-semiring and an assignment s, $s \in Pos(P, \alpha)$ iff it is a solution of $cut_\alpha(bs(P))$.*

Proof: We first show that, if s is a solution of $cut_\alpha(bs(P))$, then $s \in Pos(P, \alpha)$. If s is a solution of $cut_\alpha(bs(P))$, then, since $cut_\alpha(bs(P))$ is the CSP obtained from the best scenario by allowing only tuples with preference greater than or equal to α, $\mathrm{pref}(bs(P), s) \geq \alpha$. Hence, in the best scenario s has preference greater than or equal to α, hence $s \in Pos(P, \alpha)$.

To conclude the proof, we show that if $s \in Pos(P, \alpha)$, then s is a solution of $cut_\alpha(bs(P))$. If $s \in Pos(P, \alpha)$, then there is a scenario, say Q_i, where $\mathrm{pref}(Q_i, s) \geq \alpha$. Hence, since the preference of a solution in a scenario is always lower than or equal to its preference in the best scenario, then $\mathrm{pref}(bs(P), s) \geq \mathrm{pref}(Q_i, s) \geq \alpha$, and so s is a solution of $cut_\alpha(bs(P))$. □

By Proposition 8, we know that $Pos(P, \alpha^*) = UO(P)$. Therefore, to find a solution in $Pos(P, \alpha^*)$, it is sufficient to find an optimal solution of the best scenario of P, i.e., a solution in $UO(P)$, and thus to solve one SCSP.

7.5 Finding and testing optimality notions: summary of the results

We have provided algorithms to find solutions according to the various optimality notions and also to test whether a given solution is optimal. In most of the cases, these algorithms amounts to solving a soft constraint problem as shown in Table 2.

Table 2. Finding and testing optimal solutions.

Optimality notion	*c-semiring*	*Finding*	*Testing*
$LO(P)$	generic	1 SCSP	1SCSP
$UO(P)$	generic	1 SCSP	1SCSP
$IO(P)$	generic	1 SCSP	3 SCSPs
	idempotent	1SCSP	1CSP
$LLO(P)$	strictly monotonic	1 SCSP	1 SCSP
$WID(P)$	idempotent	2 SCSPs + 1 CSP	2SCSPs
$ID(P)$	generic	2 SCSPs	2 SCSPs
	idempotent	1 SCSP + 1 CSP	1 SCSP + 1 CSP
$NO(P)$	idempotent	2 SCSPs + 2 CSPs	2 SCSPs + 1 CSP
	strictly monotonic	1 SCSP	1 SCSP
$Nec(P, \alpha)$	generic	1 SCSP	linear time
	idempotent	1 CSP	linear time
$Nec(P, \alpha_*)$	generic	1 SCSP	linear time
$PO(P)$	idempotent	1 SCSP	1 SCSP
	strictly monotonic	1 SCSP	1 SCSP
$Pos(P, \alpha)$	idempotent	1 CSP	linear time
$Pos(P, \alpha^*)$	generic	1 SCSP	linear time

8 Necessary and possible dominance

Besides finding or testing for optimality, it may sometimes be useful to know if a solution dominates another one. We will consider four notions of dominance, which are related to the general optimality notions defined above.

Definition 15 ((strictly) dominance). *Given a scenario Q, a solution s strictly dominates (resp., dominates) a solution s' if and only if $\mathrm{pref}(Q, s) > \mathrm{pref}(Q, s')$ (resp.,$\mathrm{pref}(Q, s) \geq \mathrm{pref}(Q, s')$) in the ordering of the considered c-semiring.*

Definition 16 (necessarily (strictly) dominance). *Given an IVSCSP P and two solutions s and s' of P, s necessarily strictly dominates (resp., necessarily dominates) s' if* *and only if, in all scenarios, s strictly dominates (resp., dominates) s'. We will denote with $NDTOP(P)$ (resp., $NSDTOP(P)$) the undominated elements in the binary relation given by the necessarily dominance (resp., strictly necessarily dominance).*

Definition 17 (possibly (strictly) dominance). *Given an IVSCSP P and two solutions s and s' of P, s possibly strictly dominates (resp., possibly dominates) s' if and only if there is at least one scenario where s strictly dominates (resp., dominates) s'. We will denote with with $PDTOP(P)$ (resp., $PSDTOP(P)$) the undominated elements of the binary relation given by the possibly dominance (resp., strictly possible dominance).*

In the IVSCSP P of Figure 1, s_1 necessarily strictly dominates s_8. In the best scenario, s_2 strictly dominates s_4, while in the worst scenario s_4 strictly dominates s_2. Thus s_2 possibly strictly dominates s_4, and viceversa.

Theorem 12. *Consider an IVSCSP P. The following results hold:*

- $NO(P) \subseteq NDTOP(P) \subseteq NSDTOP(P)$.
- $NSDTOP(P) \supseteq PO(P)$.
- *If the c-semiring is strictly monotonic or idempotent, then $NDTOP(P) \subseteq PO(P)$.*
- *If the c-semiring is strictly monotonic, $NSDTOP(P) = PO(P)$.*
- *The sets $PSDTOP(P)$ and $PDTOP(P)$ may be empty.*
- *If $PDTOP(P) \neq \emptyset$, then $|PDTOP(P)| = 1$.*
- $PDTOP(P) \subseteq PSDTOP(P) = NO(P)$.

Proof: Let s be a solution of P.

We first show that $NO(P) \subseteq NDTOP(P)$. If $s \notin NDTOP(P)$, then there a solution s' that necessarily dominates s, and so there is a scenario Q where s' strictly dominates s, that is, $\text{pref}(Q, s') > \text{pref}(Q, s)$. Hence, s is not optimal in that scenario and so $s \notin NO(P)$.

We now show that $NDTOP(P) \subseteq NSDTOP(P)$. If $s \notin NSDTOP(P)$, then there is a solution s' that necessarily strictly dominates s and so s' necessarily dominates s and thus $s \notin NDTOP(P)$.

We now show that $PO \subseteq NSDTOP(P)$. If $s \notin NSDTOP(P)$, then there is a solution s' that necessarily strictly dominates s, hence, for every scenario Q, s' strictly dominates s, that is, for every scenario Q, $pref(Q, s') > pref(Q, s)$, hence for every scenario Q, s is not optimal, hence $s \notin PO(P)$.

To prove that $NDTOP(P) \subseteq PO(P)$ when P is idempotent, we will show that if $s \in NDTOP(P)$ then s is optimal in the scenario Q^s, where every tuple in s is set to its maximum preference value and all other tuples are set to their minimum preference value. This then implies that s is possibly optimal, and hence in $PO(P)$, as required.

Suppose, that $s \in NDTOP(P)$ is not optimal in the scenario Q^s, so there exists some solution s' with $pref(Q^s, s') > pref(Q^s, s)$. Since $s \in NDTOP(P)$ there exists a scenario Q with $pref(Q, s) > pref(Q, s')$ or else s' would necessarily dominate s. We have $pref(Q^s, s') > pref(Q, s')$. Since the combination is minimum, this means that the preference value of the worst tuple of s' (i.e., of the worst constraint) is worse in Q than it is in Q^s. The definition of Q^s means that this tuple is also in s' (i.e., s and s' agree on the scope of the worst constraint). This implies that

$pref(Q, s) \leq pref(Q, s')$, which contradicts $pref(Q, s) > pref(Q, s')$, completing the proof that $NDTOP(P) \subseteq PO(P)$ when P is idempotent.

If the c-semiring is strictly monotonic, $NSDTOP(P) = PO(P)$. We have already shown that $NSDTOP(P) \supseteq PO(P)$. We now show that $NSDTOP(P) \subseteq PO(P)$. If $s \in NSDTOP(P)$, then there is no solution s' such that for every scenario Q_i, $\text{pref}(Q_i, s') > \text{pref}(Q_i, s)$. Hence, for every s', there is a scenario Q_i where $\text{pref}(Q_i, s') \leq \text{pref}(Q_i, s)$. By following the same reasoning done above, it is possible to show that, $\forall s'$, $\text{pref}(Q^s, s') \leq \text{pref}(Q^s, s)$. Therefore, s is optimal in Q^s and so $s \in PO(P)$.

Furthermore, if the c-semiring is strictly monotonic, then we have $NDTOP(P) \subseteq PO(P)$ since $NDTOP(P) \subseteq NSDTOP(P) = PO(P)$.

$PSDTOP(P)$ and $PDTOP(P)$ may be empty, because there can be cycles in the *possibly dominates* and *possibly strictly dominates* relations. Let us consider the solutions s_2 and s_4 in the running example. s_2 has preference interval $[0.5, 0.9]$ and s_4 has preference interval $[0.6, 0.8]$. Then, s_2 possibly strictly dominates (and so possibly dominates) s_4, since s_2 strictly dominates s_4 in the best scenario, and s_4 possibly strictly dominates (and so possibly dominates) s_2, since s_4 strictly dominates s_2 in the worst scenario.

If $PDTOP(P) \neq \emptyset$, then $|PDTOP(P)| = 1$. In fact, assume that $PDTOP(P)$ contains two complete assignments s_1 and s_2. If s_1 and s_2 are in $PDTOP(P)$, then s_1 does not possibly dominate s_2 and s_2 does not possibly dominate s_1. Since s_1 does not possibly dominate s_2, then for every scenario Q of P, $\text{pref}(Q, s_1) < \text{pref}(Q, s_2)$, and, since s_2 does not possibly dominate s_1, then for every scenario Q of P, $\text{pref}(Q, s_2) < \text{pref}(Q, s_1)$, that is a contradiction.

$PSDTOP(P) = NO(P)$. In fact, $s \in PSDTOP(P)$ iff there is no solution s' such that s' possibly strictly dominates s, iff there is no solution s' that strictly dominates s, iff there is no solution s' such that $\text{pref}(Q, s') > \text{pref}(Q, s)$ for some scenario Q, iff for every solution s', $\text{pref}(Q, s) \geq \text{pref}(Q, s')$ for every scenario Q, iff $s \in NO(P)$.

$PDTOP(P) \subseteq PSDTOP(P)$. In fact, if $s \notin PSDTOP(P)$, then there is a solution s' that possibly strictly dominates s and thus s' possibly dominates s and so $s \notin PDTOP(P)$. □

Summarizing, given an IVSCSP P with an idempotent or a strictly monotonic c-semiring, we have the following inclusions, that are shown in Figure 4: $PDTOP(P) \subseteq PSDTOP(P) = NO(P) \subseteq NDTOP(P) \subseteq PO(P) \subseteq NSDTOP(P)$. Moreover, when the c-semiring is strictly monotonic, we have also $NSDTOP(P) = PO(P)$. Therefore, the set of the necessarily optimal solutions of P coincides with the set of the undominated elements of the binary relation given by the possibly strictly dominance over P, both if the c-semiring is strictly monotonic and if it idempotent. Moreover, the set of the possibly optimal solutions of P coincides with the set of the undominated elements of the binary relation given by the necessarily strictly dominance over P, if the c-semiring is strictly monotonic.

To test if s possibly strictly dominates (resp., possibly dominates) s' we can set each interval associated with s but not with s' to its upper bound; let λ be the combination

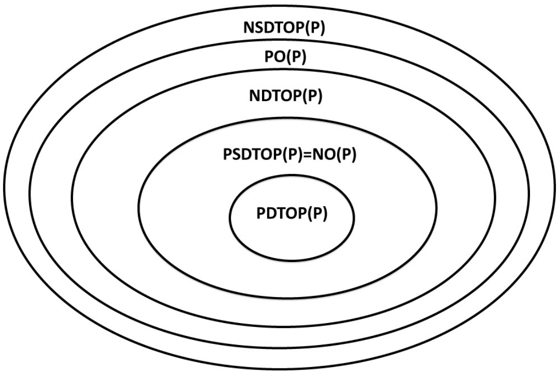


Fig. 4. Relation between undominated elements of the binary relation given by the (strictly) necessarily dominance and the undominated elements of the binary relation given by the (strictly) possibly dominance for an IVSCSP P defined over an idempotent or a strictly monotonic c-semiring.

of these values. Then we set each interval associated with s' but not with s to its lower bound; let μ be the combination of these values. Finally, we compare the preference values of s and s', by testing if the condition $\lambda \times u_1 \times \cdots \times u_k > \mu \times u_1 \times \cdots \times u_k$ (resp., $\lambda \times u_1 \times \cdots \times u_k \geq \mu \times u_1 \times \cdots \times u_k$) holds for any selections of values $u_1, \ldots, u_k$ in the intervals of both s and s'. If we have strict monotonicity, testing this condition amounts to testing if $\lambda > \mu$ (resp., $\lambda \geq \mu$). If we have idempotence, we can replace each u_i with its upper bound, and then test the condition.

To test if s necessarily dominates s', we first check if s possibly strictly dominates s'. Then:

– If s possibly strictly dominates s', then there is a scenario where s strictly dominates s' and so s' does not necessarily dominate s. Then, we check if s' possibly strictly dominates s. If so, then there is a scenario where s' strictly dominates s, hence s does not necessarily dominate s'. Therefore, s and s' are incomparable w.r.t. the necessarily dominance relation and so we conclude negatively. Otherwise, if s' does not possibly strictly dominates s, then, for every scenario, s dominates s' and, since, by hypothesis, there is a scenario where s strictly dominates s', then s necessarily dominates s' and so we conclude positively.
– If s does not possibly strictly dominate s', then, for every scenario, s' dominates s, i.e., for every scenario Q, $\mathrm{pref}(Q, s') \geq \mathrm{pref}(Q, s)$. Then, we check if s' possibly strictly dominates s. If so, then s' necessarily dominates s and so we conclude negatively. Otherwise, if s' does not possibly strictly dominates s, then, for every scenario, s dominates s', i.e., for every scenario Q, $\mathrm{pref}(Q, s) \geq \mathrm{pref}(Q, s')$, and so, since by the hypothesis above $\mathrm{pref}(Q, s') \geq \mathrm{pref}(Q, s)$, we have that, for every scenario Q, $\mathrm{pref}(Q, s) = \mathrm{pref}(Q, s')$, hence s does not necessarily dominates s' and so we conclude negatively.

To test if s necessarily strictly dominates s', we follow a reasoning similar to the one presented above, but we consider the possibly dominance relation instead of the possibly strictly dominance relation. Moreover, when s does not possibly dominate s' (i.e., the second item above), we can conclude immediately negatively, since in this case s' necessarily strictly dominates s.

9 Multiple intervals

One may wonder if IVSCSPs would be more expressive if we allowed not just a single preference interval for each assignment, but a set of such intervals. For example, instead of giving us the interval [0.1, 0.8], a user could be more precise and give us [0.1,0.5] and [0.7,0.8]. This would reduce the uncertainty of the problem. We will now show that all the interval-based optimality notions and all the scenario-based optimality notions that guarantee a certain level of preference would give the same set of optimals in this more general setting. Moreover, when the c-semiring is strictly monotonic, also the possibly and necessary optimality notions give the same set of optimals. Also, when the c-semiring is idempotent, the necessary optimality notions give the same set of optimals. In the other cases, we are however able to find approximations of the possibly and necessarily optimal solutions. More precisely, we have the following results, that are also summarized in Table 3.

Theorem 13. *Consider an IVSCSP P. Take now a new problem P' with the same variables, domains, and constraint topology as P, where, for each interval $[l, u]$ in P, there is a set of intervals $[l, u_1], [l_2, u_2], \ldots, [l_n, u]$ such that $u_i < l_{i+1}$ for $i = 1, \ldots, n-1$. Then:*

- $X(P) = X(P')$ *for* $X \in \{LO, UO, IO, LLO, ULO, WID, ID\}$.
- $Nec(P, \alpha) = Nec(P', \alpha)$ *for all* α.
- $Pos(P, \alpha) = Pos(P', \alpha)$ *for all* α.
- $NO(P') \supseteq NO(P)$.
- $PO(P') \subseteq PO(P)$.
- *If the c-semiring is strictly monotonic,* $NO(P) = NO(P')$ *and* $PO(P) = PO(P')$.
- *If the c-semiring is idempotent,* $NO(P) = NO(P')$.

Proof: To show that $X(P) = X(P')$ for $X \in \{LO, UO, IO, LLO, ULO, WID, ID\}$, it is sufficient to recall that all solutions in $\{LO, UO, IO, LLO, ULO, WID, ID\}$ are computed by considering for every tuple associated with interval $[l, u]$ only the lower bound l and the upper bound u that, by construction of P', are the same in P and P'.

Let s be a complete assignment of P. Let us consider a generic α. To show that $Nec(P, \alpha) = Nec(P', \alpha)$, we first show that $Nec(P, \alpha) \subseteq Nec(P', \alpha)$. If $s \in Nec(P, \alpha)$, then, for every scenario Q of P, $\text{pref}(Q, s) \geq \alpha$. Since the set of the scenarios of P is a superset of the scenarios of P', this holds also for every scenarios of P'. Therefore, $s \in Nec(P', \alpha)$. We now show that $Nec(P', \alpha) \subseteq Nec(P, \alpha)$. If $s \notin Nec(P, \alpha)$, then $\text{pref}(Q, s) < \alpha$ for some scenario Q of P and this holds also for the worst scenario, since $\text{pref}(ws(P), s) \leq \text{pref}(Q, s) < \alpha$. Since the worst scenario is one of the scenario of P', then $s \notin Nec(P', \alpha)$.

To show that $Pos(P, \alpha) = Pos(P', \alpha)$, we first show that $Pos(P', \alpha) \subseteq Pos(P, \alpha)$. If $s \in Pos(P', \alpha)$, then for some scenario Q of P', $\text{pref}(Q, s) \geq \alpha$. Since every scenario of P' is also a scenario of P, then $s \in Pos(P, \alpha)$. We now show that

$Pos(P,\alpha) \subseteq Pos(P',\alpha)$. If $s \in Pos(P,\alpha)$, then $\mathrm{pref}(Q,s) \geq \alpha$ for some scenario Q of P, and this holds also for the best scenario, since $\mathrm{pref}(bs(P),s) \geq \mathrm{pref}(Q,s) \geq \alpha$. Since the best scenario is one of the scenarios of P', then $s \in Pos(P',\alpha)$.

Since the set of the scenarios of P is a superset of the scenarios of P', then $NO(P) \subseteq NO(P')$. In fact, if $s \in NO(P)$, then it is optimal for every scenario of P and also for every scenario of P'.

Moreover, $PO(P') \subseteq PO(P)$. In fact, if $s \in PO(P')$, then there is a scenario of P' where s is optimal and, as every scenario of P' is also a scenario of P, then $s \in PO(P)$.

If the c-semiring is strictly monotonic, then $NO(P) = NO(P')$. By Theorem 8, we know that, if the c-semiring is strictly monotonic, then $s \in NO(P)$ iff $s \in Opt(Q_s)$, where Q_s is the scenario where every preference associated to a tuple of s is set to its lower bound and the preferences of all other tuples are set to their upper bound. Since Q_s is one of the scenarios of P', it is possible to show that $s \in NO(P')$ iff $s \in Opt(Q_s)$, by following the same proof of Theorem 8. Hence, $NO(P') = NO(P)$.

Similarly, if the c-semiring is strictly monotonic, then $PO(P) = PO(P')$. By Theorem 10, we know that, if the c-semiring is strictly monotonic, then $s \in PO(P)$ iff $s \in Opt(Q^s)$, where Q^s is the scenario where all the preferences of tuples in s are set to their upper bound and all other tuples are associated to their lower bound. Since Q^s is one of the scenarios of P', it is possible to show, by following the same proof of Theorem 10, that $s \in PO(P)$ iff $s \in Opt(Q^s)$.

If the c-semiring is idempotent, $NO(P) = NO(P')$. In fact, by Theorem 8, we know that $s \in NO(P)$ iff for every s', $s \in Opt(Q^{s'})$, where $Q^{s'}$ is the scenario where we put every tuple of s' to its upper bound and every other tuple to its lower bound. Since, for every s', $Q^{s'}$ is a scenario of P', then by following the same proof of Theorem 8, we can show that $s \in NO(P')$ iff for every s', $s \in Opt(Q^{s'})$, Hence, $NO(P') = NO(P)$. □

Table 3. Comparison of the optimality sets of problems P (with single intervals) and P' (with multiple intervals), as defined in Theorem 13.

Optimality notion	*c-semiring*	*Comparison*
LO	generic	$LO(P) = LO(P')$
UO	generic	$UO(P) = UO(P')$
IO	generic	$IO(P) = IO(P')$
LLO	generic	$LLO(P) = LLO(P')$
ULO	generic	$ULO(P) = ULO(P')$
$Nec(\alpha)$	generic	$Nec(P,\alpha) = Nec(P',\alpha)$
$Pos(\alpha)$	generic	$Pos(P,\alpha) = Pos(P',\alpha)$
NO	generic	$NO(P) \subseteq NO(P')$
	idempotent	$NO(P) = NO(P')$
	strictly monotonic	$NO(P) = NO(P')$
PO	generic	$PO(P) \supseteq PO(P')$
	strictly monotonic	$PO(P) = PO(P')$

10 Experimental results

10.1 Instance generator

We randomly generated fuzzy IVMSPs (as defined in Section 5.7) according to the following parameters:

- m: number of meetings (default 12);
- n: number of agents (default 5);
- k: number of meetings per agent (default 3);
- l: number of time slots (default 10);
- min and max: minimal (default 1) and maximal (default 2) distance (in time slots) between two locations;
- i: percentage of preference intervals (default 30%).

Given such parameters, we generate an IVSCSP with m variables, representing the meetings, each with domain of size l. The domain values $1, \ldots, l$ represent the time slots, that are assumed to all have the same length equal to one time unit, and to be adjacent to each other. Thus, for example, time slot i ends when time slot $i + 1$ starts. Given two time slots i and $j > i$, they can be used for two meetings only if the distance between their locations (see later) is at most $j - i - 1$.

For each of the n agents, we generate randomly k integers between 1 and m, representing the meetings he needs to participate in. Also, for each pair of time slots, we randomly generate a integer between min and max that represents the time needed to go from one location to the other one. This will be called the distance table.

Given two meetings, if there is at least one agent who needs to participate in both, we generate a binary constraint between the corresponding variables. Such a constraint is satisfied by all pairs of time slots that are compatible according to the distance table.

We then generate the preferences over the domain values and the compatible pairs in the binary constraints, by randomly generating a number in $(0, 1]$ or an interval over $(0, 1]$, according to the parameter i.

As an example, assume to have $m = 5$, $n = 3$, $k = 2$, $l = 5$, $min = 1$, $max = 2$, and $i = 30$. According to these parameters, we generate a IVMSP with the following features:

- 5 meetings: m_1, m_2, m_3, m_4, and m_5;
- 3 agents: a_1, a_2, and a_3;
- 5 time slots: $t_1, \ldots, t_5$;
- agents' participation to meetings: we randomly generate 2 meetings for each agent, for example
  - a_1 must participate in meetings m_1 and m_2;
  - a_2 must participate in meetings m_4 and m_5;
  - a_3 must participate in meetings m_2 and m_3;
- distance table: we randomly generate its values, for example as in Table 4;
- we randomly generate the preferences associated to domain values and compatible pairs in the constraints, in a way that 30% of the preferences are preference intervals contained in $(0, 1]$ and 70% of the preferences are single values in $(0, 1]$.

Table 4. Distance between meeting locations.

	1	2	3	4	5
1	0	1	2	1	2
2	1	0	2	1	2
3	2	2	0	1	1
4	1	1	1	0	2
5	2	2	1	2	0

In this example, a feasible meeting scheduling is obtained by assigning the following time slots to meetings: $(m_1, t_3), (m_2, t_1), (m_3, t_5), (m_4, t_2), (m_5, t_5)$. The preference interval for such a scheduling will depend on the preference values in the domains and constraints. More precisely, as we use preference values between 0 and 1 and we adopt the fuzzy criteria, the preference interval will be $[l, u]$, where l (resp., u) is the minimum among all the lower (resp., upper) bounds of the preference intervals selected by this assignment in the constraints.

10.2 Experimental tests

We implemented our algorithms using a Java (version 1.6.0_07) c-semiring based framework and the Choco constraint programming toolkit (version 1.2.06). Experiments were run on AMD Opteron 2.3GHz machines with 2GB of RAM.

We used 4 different test sets, each one generated varying in turn n, m, k, and i, while fixing the others to their default values. Moreover, α, i.e., the minimum level of preference used in $Pos(P, \alpha)$ and $Nec(P, \alpha)$, is always 0.5.[2] The sample size is 50 for each data point.

Figure 5(a) shows the execution time (measured in milliseconds) of the algorithms to find a solution, belonging to each type of the interval-based optimality notions, as a function of the number of agents. We can notice that there is a peak when the number of agents is 8, which represents problems with a small number of solutions. With more agents, the problems have no solution, while with a smaller number of agents there are many solutions. In both such cases, it is easy to find a feasible meeting scheduling.

For the more general optimality notions, Figure 5(b) shows that the behavior is the same except for POS(0.5) and NEC(0.5) because, in these algorithms, we need to solve a CSP, while in the other algorithms we solve at least one SCSP. In fact, POS(0.5) and NEC(0.5) takes approximately the same time no matter the number of agents in the problem.

Figures 5(c) and 5(d) show the performance of the algorithms for all optimality notions, as a function of the number of meetings per agent. Since $LO(P) = Nec(P, \alpha_*)$ and $UO(P) = Pos(P, \alpha^*)$, these curves in the two graphs coincide. The lines corresponding to the WID algorithm in Figure 5(c) and to the NO algorithm in Figure 5(d) are similar, and are above the others in both figures, because the WID algorithm needs to find the lower and upper optimal preference, to perform two cuts, and to solve the

[2] In the following figures, we will omit writing P in the names of the algorithms.

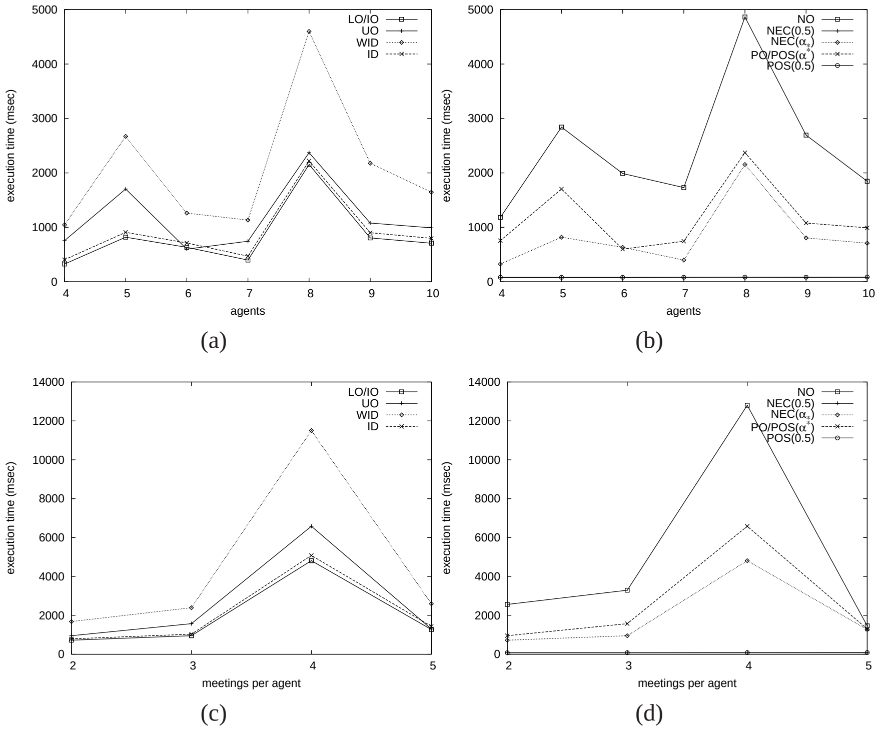


Fig. 5. Execution time (msec.) as a function of number of agents and meetings per agent.

CSP obtained combining the cuts, while the other algorithms (expect NO) only need to solve an SCSP. Moreover, the WID algorithm is a sub-routine of the NO algorithm.

Notice that finding solutions in NO, $Nec(P, \alpha_*)$, or $POS(P, \alpha^*)$ is more expensive than finding solutions in $Nec(P, 0.5)$, or $POS(P, 0.5)$, as expected since α_* and α^* are the best preference levels that one can reach.

The peak at 4 meetings per agents, shown in Figures 5(c) and 5(d), corresponds to problems which are more difficult to solve because they have very few solutions. This is analogous to what we have noticed in Figures 5(a) and 5(b) with the peak at 8 agents.

Figures 6(a) and 6(b) show that the execution time increases exponentially when the number of meetings (i.e., the number of variables in the problem) arises. In this case, the execution time is mainly influenced by the size of the problems, no matter which algorithm is used.

Figures 6(c) and 6(d) show that the execution time is not influenced by the amount of intervals in the problem. As in all the other graphs, finding a WID or an NO solution is more expensive than finding other kinds of solutions. The two peaks at 20% and 60% of intervals are due to two very hard problems inside the test set.

Figure 7(a) and Figure 7(b) consider those optimality sets that can be empty (that is, WID, ID, and NO) and show the percentage of times a solution of a certain kind exists. Clearly, when there is no solution, WID, ID and NO contain all assignments

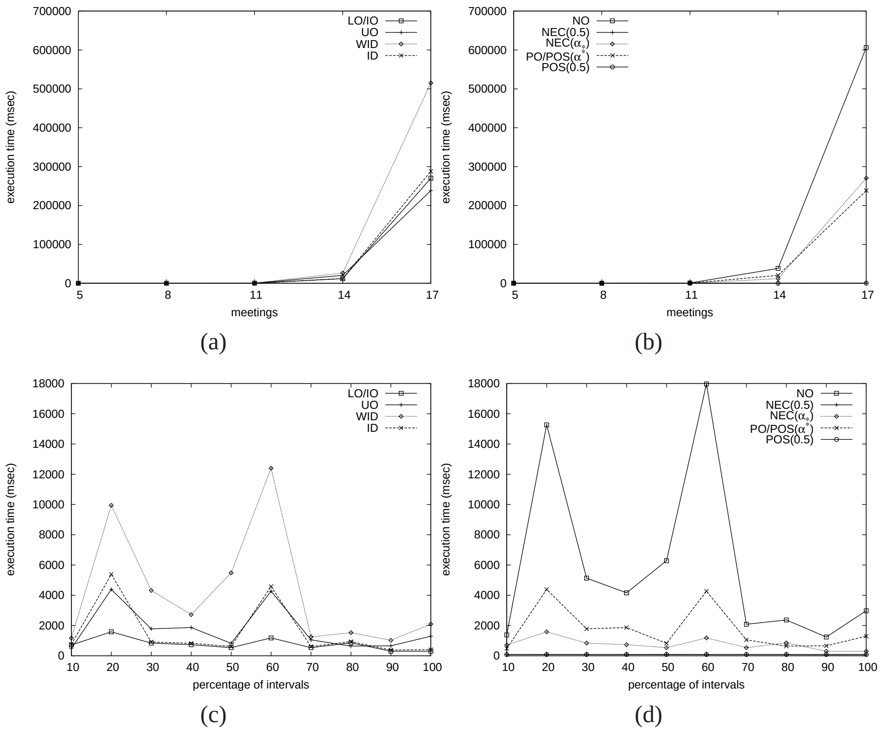


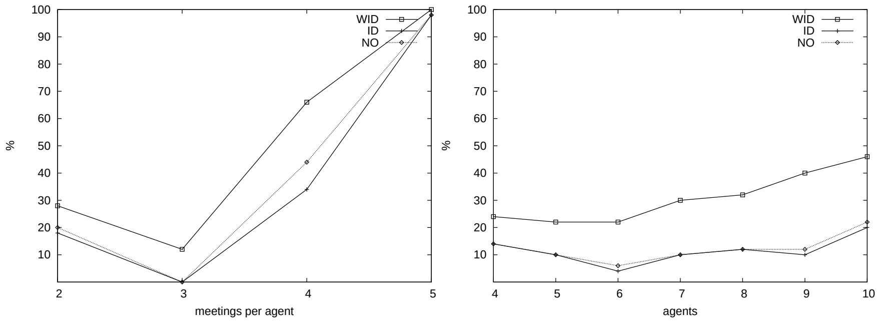
Fig. 6. Execution time (msec.) as a function of the number of meetings and the percentage of intervals.

and coincide. This is the case when the number of meetings per agents is larger (more than 3 meetings per agent in our settings). When we consider less constrained problems with 2-3 meetings per agent, as expected, we have more WID solutions than ID and NO solutions. Notice that the size of WID, ID and NO varies very little when the number of agents is between 4 and 8 (Figure 7(b)). However, when such a number is between 8 to 10, the size of the solution sets is larger because there are more instances with no solution. If we vary the number of meetings, we can see in Figure 8(a) that the number of such a kind of solutions tends to decrease slightly as the number of variables (i.e. meetings) arises. In fact, a larger number of variables may imply a larger number of constraints, which may imply a smaller number of WID, ID, and NO solutions.

In figure 8(b) we consider instances where we vary the percentage of intervals from 10 to 100%. When incompleteness is higher than 40%, most of the instances don't have WID, ID, and NO solutions. This is predictable, because a larger number of intervals makes it less probable the existence of solutions that are optimal in all scenarios, since the number of scenarios is larger.

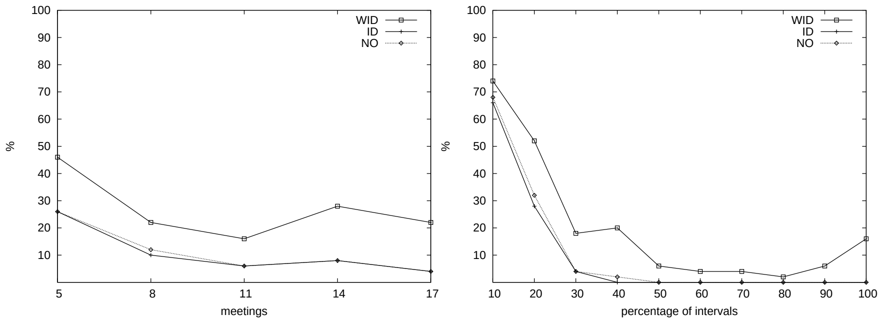
11 Final considerations and future work

Summarizing, given an IVSCSP P, the solutions in $NO(P)$ are certainly the most attractive, as they are the best ones in every scenario. However, if there is none, we



(a) (b)

Fig. 7. Existence of WID, ID, and NO solution, varying agents and meetings per agent.



(a) (b)

Fig. 8. Existence of WID, ID, and NO solution, varying meetings and intervals.

can look for solutions in $Nec(P, \alpha_*)$ (which coincide with solutions in $LO(P)$), which guarantee a preference level α_* in all scenarios. If α_* is too low, we can consider other notions of optimality; for example, if we feel optimistic, we can consider the solutions in $Pos(P, \alpha^*)$(which coincide with solutions in $UO(P)$): they guarantee that it is possible to reach a higher level of preference, although not in all scenarios.

If we allow users to associate to each partial assignment in the constraints not just a single interval, but a set of multiple intervals, this would reduce the uncertainty of the problem. However, when the c-semiring is strictly monotonic (resp., idempotent), this added generality does not change the set of the optimal solutions in any of the considered notions (resp., in any of the considered notions with the exception of the possibly optimal notions). This means that a level of precision greater than a single interval does not add useful information when looking for an optimal solution.

This paper considers only totally ordered preferences. IVSCSPs can be defined also for a partially ordered setting. We plan to extend the analysis of the optimality notions also to this more general case. We also intend to define dedicated solving or propagating schemes to tackle IVSCSPs rather than relying on existing solvers for SCSPs. It is interesting also to consider the addition of probability distributions over preference intervals and to interleave search with elicitation as in [6, 7].

12 Acknowledgements

The work of Nic Wilson has been supported by the Science Foundation Ireland under Grant No. 05/IN/I886 and Grant No. 08/PI/I1912.
The work of Mirco Gelain, Maria Silvia Pini, Francesca Rossi and Kristen Brent Venable has been partially supported by the Italian MIUR PRIN project 20089M932N: ”Innovative and multi-disciplinary approaches for constraint and preference reasoning”.

References

1. S. Bistarelli, U. Montanari, and F. Rossi. Semiring-based constraint solving and optimization. *JACM*, 44(2):201–236, March 1997.
2. S. Bistarelli, U. Montanari, F. Rossi, T. Schiex, G. Verfaillie, and H. Fargier. Semiring-based CSPs and valued CSPs: Frameworks, properties, and comparison. *Constraints*, 4(3), 1999.
3. M. Ceberio and F. Modave. An interval-valued, 2-additive choquet integral for multi-criteria decision making. In *IPMU'04*, 2004.
4. B. Faltings and S. Macho-Gonzalez. Open constraint programming. *AI Journal*, 161(1-2):181–208, 2005.
5. H. Fargier, T. Schiex, and G. Verfaille. Valued Constraint Satisfaction Problems: Hard and Easy Problems. In *IJCAI-95*, pages 631–637. Morgan Kaufmann, 1995.
6. M. Gelain, M. S. Pini, F. Rossi, and K. B. Venable. Dealing with incomplete preferences in soft constraint problems. In *Proc. CP'07*, volume 4741 of *LNCS*, pages 286–300. Springer, 2007.
7. M. Gelain, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Dealing with incomplete preferences in soft constraint problems. In *Proc. CP'08*, volume 5202 of *LNCS*, pages 402–417. Springer, 2008.
8. S. Macho González, C. Ansótegui, and P. Meseguer. On the relation among open, interactive and dynamic CSP. In *The Fifth Workshop on Modelling and Solving Problems with Constraints (IJCAI'05)*, 2005.
9. P. Meseguer, F. Rossi, and T. Schiex. Soft constraints. In F. Rossi, P Van Beek, and T. Walsh, editors, *Handbook of Constraint Programming*. Elsevier, 2006.
10. M. S. Pini, F. Rossi, K. B. Venable, and R. Dechter. Robust solutions in unstable optimization problems. In *Proc. Recent Advances in Constraints*, LNAI. Springer, 2009.
11. F. Rossi, P. Van Beek, and T. Walsh, editors. *Handbook of Constraint Programming*. Elsevier, 2006.
12. Z. Ruttkay. Fuzzy constraint satisfaction. In *Proceedings 1st IEEE Conference on Evolutionary Computing*, pages 542–547, Orlando, 1994.
13. R. Zivan U. Shapen and A. Meisels. Meeting scheduling problem (MSP). Available at *http://www.cs.st-andrews.ac.uk/∼ianm/CSPLib/prob/prob046/index.html* , 2010.
14. N. Wilson, D. Grimes, and E. C. Freuder. A cost-based model and algorithms for interleaving solving and elicitation of csps. In *Proc. CP'07*, volume 4741 of *LNCS*, pages 666–680. Springer, 2007.
15. N. Yorke-Smith and C. Gervet. Certainty closure: A framework for reliable constraint reasoning with uncertainty. In *Proc. CP'03*, volume 2833 of *LNCS*, pages 769–783. Springer, 2003.