Relaxations of Semiring Constraint Satisfaction Problems

Louise Leenen[1], Thomas Meyer[2], and Aditya Ghose[1]

[1] Decision Systems Laboratory
School of IT and Computer Science
University of Wollongong, Australia
{ll916,aditya}@uow.edu.au
[2] National ICT Australia
School of Computer Science and Engineering
University of New South Wales, Sydney, Australia
thomas.meyer@nicta.com.au | tmeyer@cse.unsw.edu.au

Abstract. The Semiring Constraint Satisfaction Problem (SCSP) framework is a popular approach for the representation of partial constraint satisfaction problems. In this framework preferences can be associated with tuples of values of the variable domains. Bistarelli et al. [1] define an abstract solution to a SCSP which consists of the best set of solution tuples for the variables in the problem. Sometimes this abstract solution may not be good enough, and in this case we want to change the constraints so that we solve a problem that is slightly different from the original problem but has an acceptable solution. We propose a relaxation of a SCSP, and use a semiring to give a distance measure between the original SCSP and the relaxed SCSP.

1 Introduction

There has been considerable interest over the past decade in *over-constrained problems*, *partial constraint satisfaction problems* and *soft constraints*. This has been motivated by the observation that with most real-life problems, it is difficult to offer *a priori* guarantees that the input set of constraints to a constraint solver is solvable. In part, this is because many real-life problems are inherently over-constrained. In part, this is also because it is difficult for human users to peruse a given set of constraints that might have been obtained for a given problem to determine if it is solvable. In the general case, constraint solvers must be able to deal with problems that are potentially over-constrained. The key challenge in dealing with an over-constrained problem is identifying appropriate *relaxations* of the original problem that are solvable. Early approaches to such relaxations largely focussed on finding maximal subsets (with respect to set cardinality) of the original set of constraints that are solvable (such as Freuder and Wallace's work on the MaxCSP problem [2]). Subsequent efforts considered more fine-grained notions of relaxation, where entire constraints did not have to be removed from consideration. Examples of such efforts include the HCLP framework [3], Fuzzy CSPs [4] and Probabilistic CSPs [5].

Bistarelli et al. [1] proposed an abstract semiring CSP scheme (henceforth referred to as the SCSP framework) that generalized most of these earlier attempts, while making possible to define several useful new instance of the scheme. The SCSP scheme assumes the existence of a semiring of abstract preference values, such that the associated multiplicative operator is used for combining preference values, while the associated additive operator is used for comparing preference values. While a classical constraint defines which combinations of value assignments to the variables in its signature are allowed, an SCSP constraint assigns a preference value to all possible value assignments to the variables in its signature. These preferences implicitly define a relaxation strategy ("try to satisfy the constraint using the most preferred tuples, else try the next most preferred tuples" and so on). Note that the actual mechanism is somewhat more involved than this informal expository description, because the semiring preference values are partially ordered in the general case.

Our aim in this paper is to define how an SCSP might be relaxed. At first blush, this might appear counter-intuitive, since an SCSP is intended to define how soft constraints are relaxed. We will explain our motivations by describing it in terms of a generic optimization problem (C, O), defined by a set of constraints C and an objective function O. Assume that we have been given a lower bound on the value of the optimal solution (e.g., a minimal threshold on profit by a business unit set by management). Consider a situation where the optimal solution obtained fails to meet this threshold (e.g., the optimal profit figure falls short of the profit target). We are interested in seeking a new (and potentially relaxed) set of constraints C' that is minimally different from the original set C (under some notion of minimal difference that we will leave undefined for the time being), such that the revised optimization problem (C', O) admits an optimal solution that satisfies the threshold. The revised (or relaxed) set of constraints C' is potentially very useful, because it can point to minimal changes in the physical reality being modeled by the constraints, which, if effected, would permit us to meet the threshold on the value of the objective function.

In this paper, we attempt such an exercise in the context of SCSPs. A SCSP does not have an explicit objective function. Objectives are implicitly articulated (in a distributed fashion) via the preferences over tuples in each SCSP constraint. Instead of an optimal solution, we are able to articulate the preference values of the (potentially many) "best" solutions to an SCSP. The version of the problem that we address in this paper is as follows. Consider an SCSP P and a threshold β on the preference value of the "best" solution(s) to P. Assume that the "best" solutions to P fall short of this threshold. We define a mechanism by which we may "minimally" alter (i.e. relax) P to obtain a P' such that it admits a "best" solution that meets this threshold. We will use as a running example a problem involving a hotel that is currently unable to attain a five-star rating and that is interested in determining the minimal changes required to its infrastructure in order to achieve such a rating. In this example, the star rating of the hotel is modeled via semiring preference values.

The rest of this paper is organized as follows. In Section 2 we describe the SCSP framework. In Section 3 we describe our proposals by defining what a good enough solution is, and how to find a suitable relaxation for a SCSP. In Section 4 we compare our proposal with the Metric SCSPs of [6]. Section 5 contains the conclusion and a discussion of our future research.

2 The SCSP Framework

When we deal with constraints, the type of semirings that are used are called c-semirings. Bistarelli et al. [1] define a c-semiring, a constraint system, a constraint and a constraint problem w.r.t. c-semirings. They also define combination and projection operations in order to define a solution to a SCSP. These definitions follow below.

Definition 1. *A c-semiring is a tuple* $S = \langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$ *such that*

- A *is a set with* $\mathbf{0}, \mathbf{1} \in A$;
- $+$ *is defined over (possibly infinite) sets of elements of* A *as follows* [3]:
  - *for all* $a \in A$, $\sum(\{a\}) = a$;
  - $\sum(\emptyset) = \mathbf{0}$ *and* $\sum(A) = \mathbf{1}$;
  - $\sum(\bigcup A_i, i \in I) = \sum(\{\sum(A_i), i \in I\})$ *for all sets of indices* I *(flattening property);*
- $\times$ *is a commutative, associative, and binary operation such that* $\mathbf{1}$ *is its unit element and* $\mathbf{0}$ *is its absorbing element;*
- $\times$ *distributes over* $+$ *(i.e., for any* $a \in A$ *and* $B \subseteq A$, $a \times \sum(B) = \sum(\{a \times b, b \in B\})$*).*

The elements of the set A are the preference values to be assigned to tuples of values of the domains of constraints. The operator $\times$ is used to combine constraints in order to find a solution (i.e. a single constraint) to a SCSP, and the operator $+$ is used to define the projection of a tuple of values for a set of variables onto a tuple of values for the variables in a constraint. It is now possible to derive a partial ordering $\leqslant_S$ over the set A: $\alpha \leqslant_S \beta$ iff $\alpha + \beta = \beta$.[4] This partial ordering will be used to to distinguish the maximal solution(s) in our constraint problems. The element $\mathbf{0}$ is the minimum element in the ordering, while the element $\mathbf{1}$ is the maximum element.

Definition 2. *A constraint system is a 3-tuple* $CS = \langle S_p, D, V\rangle$, *where* $S_p = \langle A_p, +_p, \times_p, \mathbf{0}, \mathbf{1}\rangle$ *is a c-semiring,* V *is an ordered finite set of variables, and* D *is a finite set containing the allowed values for the variables in* V.

For each tuple of values (of D) for the involved variables of a constraint, a corresponding element of A_p is assigned.

[3] When $+$ is applied to sets of elements, we will use the symbol $\sum$ in prefix notation.

[4] Singleton subsets of the set A are represented without braces.

Definition 3. *Given a constraint system* $CS = \langle S_p, D, V \rangle$*, where* $S_p = \langle A_p, +_p, \times_p, \mathbf{0}, \mathbf{1} \rangle$*, a constraint over CS is a pair* $c = \langle def_c^p, con_c \rangle$ *where* $con_c \subseteq V$ *is called the type of the constraint, and* $def_c^p : D^k \rightarrow A_p$ *(where k is the cardinality of* con_c*) is called the value of the constraint.*

We now have the building blocks required to define a SCSP.

Definition 4. *Given a constraint system* $CS = \langle S_p, D, V \rangle$*, a Semiring Constraint Satisfaction Problem (SCSP) over CS is a pair* $P = \langle C, con \rangle$ *where C is a finite set of constraints over CS and* $con = \bigcup_{c \in C} con_c$ *We also assume that* $\langle def_{c_1}^p, con_c \rangle \in C$ *and* $\langle def_{c_2}^p, con_c \rangle \in C$ *implies* $def_{c_1}^p = def_{c_2}^p$*.*

Consider the following example that is used throughout this paper.

Example 1. A hotel chain acquires a star rating that is an accumulative rating of the different branches. Currently it has a four star rating and it aims for a five star rating. There are various renovations that can be done at branches to increase the rating of the hotel: 1) Lay new carpets, 2) Upgrade a swimming pool, or 3) Paint the building.

The manager of the hotel chain has to choose which (minimal) renovations to do at which branches under certain restrictions (such as the budget, renovations needed at each branch, and the constraints of the renovating teams). This problem can be expressed as a CSP. We can then add a semiring structure to allow the manager to express his preferences for particular tuples of domain values of the constraints. The hotel chain consist of three branches which are denoted by X, Y and Z. To avoid unnecessary disruptions, the manager wants at most one renovation job at a time to be performed at a particular branch, and as few renovation jobs in total as possible.

This problem can be expressed as a SCSP: a constraint system $CS = \langle S_p, D, V \rangle$ and a SCSP $P = \langle C, con \rangle$, where $V = con = \{X, Y, Z\}$, $D = \{0, 1, 2, 3\}$, $C = \{c_1, c_2, c_3\}$, and $S_p = \langle \{0, 0.25, 0.5, 0.75, 1\}, max, min, 0, 1 \rangle$.

The value of a decision variable indicates which job is to be done at a particular branch: let re-carpeting be represented by the value 1, pool renovation by the value 2, and painting by the value 3. The value 0 represents no job being done at a particular branch. A renovation job with a higher value will contribute more towards a higher star rating. Assume there are three binary constraints, $c_1 = \langle def_{c_1}^p, \{X, Y\} \rangle$, $c_2 = \langle def_{c_2}^p, \{Y, Z\} \rangle$, and $c_3 = \langle def_{c_3}^p, \{X, Z\} \rangle$. The tuples in the domains of these constraints together with their preference values (i.e. associated c-semiring values) are given in Table 1.

Note that the manager can assign any value in the set of the c-semiring to a tuple. His choice of value represents the desirability of that particular tuple. Consider the entry $def_{c_1}^p(\langle 0, 2 \rangle) = 0.75$. The tuple $\langle 0, 2 \rangle$ is a tuple of values for constraint c_1 that represents the case where no renovation is to be done at branch X while branch Y is to be painted. The assigned preference value of 0.75 is high and this indicates that it is an option that is preferred, for instance, to the one represented by the tuple $\langle 1, 1 \rangle$ with its value of 0.5. This tuple ($\langle 1, 1 \rangle$) represents the case where both branches X and Y are to be re-carpeted. Also consider the

Table 1. Constraint Definitions

t	$def^p_{c_1}(t)$	$def^p_{c_2}(t)$	$def^p_{c_3}(t)$
$\langle 0,0\rangle$	0.25	0	0
$\langle 0,1\rangle$	0.5	0	0
$\langle 0,2\rangle$	0.75	0	0.75
$\langle 0,3\rangle$	1	0.75	0
$\langle 1,0\rangle$	0.5	0	0
$\langle 1,1\rangle$	0.5	0	0.5
$\langle 1,2\rangle$	0.75	0.25	0
$\langle 1,3\rangle$	0	0.5	0
$\langle 2,0\rangle$	0.75	0	0.75
$\langle 2,1\rangle$	0.75	0.25	0
$\langle 2,2\rangle$	0	0.5	0
$\langle 2,3\rangle$	0	0.5	0
$\langle 3,0\rangle$	1	0.75	0
$\langle 3,1\rangle$	0	0.5	0
$\langle 3,2\rangle$	0	0.5	0
$\langle 3,3\rangle$	0	0.5	0

assigned preference values for constraint c_3 (the values in the last column): the manager prefers either one of the tuples $\langle 0,2\rangle$ or $\langle 2,0\rangle$ over any other tuples. These tuples represent the cases where the swimming pool at either branch X or branch Z is to be upgraded. Laying new carpets at both branches X and Z is the only other acceptable choice for constraint c_3. A tuple with an associated value of 0 is highly undesirable.

The values specified for the tuples of each constraint are used to compute values for the tuples of the variables in the set *con* according to the semiring operations; multiplication and addition. The multiplicative operation is used to combine the c-semiring values of the tuples of each constraint to get the c-semiring value of a tuple for all the variables, and the additive operation is used to obtain the value of the tuples of the variables in the type of the problem.

Definition 5. *Given a constraint system* $CS = \langle S_p,\, D,\, V\rangle$ *where* V *is totally ordered via* $\preceq$*, consider any* k*-tuple* $t = \langle t_1,\, t_2,\, \ldots,\, t_k\rangle$ *of values of* D *and two sets* $W = \{w_1,\, \ldots,\, w_k\}$ *and* $W' = \{w'_1,\, \ldots,\, w'_m\}$ *such that* $W' \subseteq W \subseteq V$ *and* $w_i \preceq w_j$ *if* $i \leq j$ *and* $w'_i \preceq w'_j$ *if* $i \leq j$*. Then the projection of* t *from* W *to* W'*, written* $t \downarrow^W_{W'}$*, is defined as the tuple* $t' = \langle t'_1,\, \ldots,\, t'_m\rangle$ *with* $t'_i = t_j$ *iff* $w'_i = w_j$*.*

The following definition defines the operation of combining two constraints to form a single constraint. We will use this operation to combine all the constraints in a problem into a single constraint.

Definition 6. *Given a constraint system* $CS = \langle S_p,\, D,\, V\rangle$ *where* $S_p = \langle A_p,$ $+_p,\, \times_p,\, \mathbf{0},\, \mathbf{1}\rangle$ *and two constraints* $c_1 = \langle def^p_{c_1},\, con_{c_1}\rangle$ *and* $c_2 = \langle def^p_{c_2},\, con_{c_2}\rangle$

over CS, their combination, written $c_1 \otimes c_2$*, is the constraint* $c = \langle def_c^p,\ con_c\rangle$ *with* $con_c = con_{c_1} \cup con_{c_2}$ *and* $def_c^p(t) = def_{c_1}^p(t \downarrow_{con_{c_1}}^{con_c}) \times_p def_{c_2}^p(t \downarrow_{con_{c_2}}^{con_c})$*.*

The operation $\otimes$ is commutative and associative because $\times$ is. We can extend the operation $\otimes$ to more than two arguments, say $C = \{c_1, ..., c_n\}$, by performing $c_1 \otimes c_2 \otimes ... \otimes c_n$, which we will denote by $(\bigotimes C)$.

Definition 7. *Given a constraint system* $CS = \langle S_p,\ D,\ V\rangle$*, where* $S_p = \langle A_p,\ +_p,\ \times_p,\ \mathbf{0},\ \mathbf{1}\rangle$*, a constraint* $c = \langle def_c^p,\ con_c\rangle$ *over CS, and a set I of variables* $(I \subseteq V)$*, the projection of c over I, written* $c \Downarrow I$*, is the constraint* $c' = \langle def_{c'}^p,\ con_{c'}\rangle$ *over CS with* $con_{c'} = I \cap con_c$ *and* $def_{c'}^p(t') = \sum_{\{t|t\downarrow_{I\cap con_c}^{con_c}=t'\}} def_c^p(t)$*.*

A solution to a SCSP can now be defined.

Definition 8. *Given a SCSP* $P = \langle C,\ con\rangle$ *over a constraint system CS, the solution of P is a constraint defined as* $Sol(P) = (\bigotimes C)$*.*

A solution to a SCSP is a single constraint formed by the combination of all the original constraints of the problem. Such a constraint provides, for each tuple of values of D for the variables in *con*, a corresponding c-semiring value. We now consider the definition of an *abstract solution* that consists of the set of k-tuples of D whose associated c-semiring values are maximal w.r.t. $\leqslant_{S_p}$.

Definition 9. *Given a SCSP problem* $P = \langle C,\ con\rangle$*, consider* $Sol(P) = \langle def_c^p,\ con\rangle$*. Then the abstract solution of P is the set*
$ASol(P) = \{\langle t,\ v\rangle \mid def_c^p(t) = v$ *and there is no* t' *such that* $v <_{S_p} def_c^p(t')\}$*.*
Let $ASolV(P) = \{v \mid \langle t, v\rangle \in ASol(P)\}$*.*

Example 2. We now compute an abstract solution for our hotel chain example. The first step is to combine the first and second constraints, c_1 and c_2. Table 2 shows the c-semiring values associated with each tuple in the constraint $c_1' = c_1 \otimes c_2$. Then we combine the constraint c_1' and the constraint c_3: $c_2' = c_1' \otimes c_3$. See Table 3. We now have an abstract solution, $ASol(P) = \{\langle\langle 0, 2, 2\rangle,\ 0.5\rangle,\ \langle\langle 0, 3, 2\rangle,\ 0.5\rangle\}$, with $ASolV(P) = \{0.5\}$. Thus the best solution tuples provide a preference value of 0.5.

3 A Relaxation of a SCSP

We are interested in the case of a SCSP for which the abstract solution is not considered to be good enough. For example, the manager in our hotel chain example may require a better solution. For instance, a solution tuple with a preference value of at least 0.75. The constraints of a problem model requirements that may be relaxed. We attempt to find a satisfactory solution to a relaxed version of the original problem. In this section we define when a solution is regarded to be good enough, and how to find suitable relaxations of the constraints of a SCSP.

Table 2. Definition of Constraint c'_1

t	$def^p_{c'_1}(t)$
$\langle 0,0,3 \rangle$	0.25
$\langle 0,1,2 \rangle$	0.25
$\langle 0,1,3 \rangle$	0.5
$\langle 0,2,1 \rangle$	0.25
$\langle 0,2,2 \rangle$	0.5
$\langle 0,2,3 \rangle$	0.5
$\langle 0,3,0 \rangle$	0.75
$\langle 0,3,1 \rangle$	0.5
$\langle 0,3,2 \rangle$	0.5
$\langle 0,3,3 \rangle$	0.5
$\langle 1,0,3 \rangle$	0.5
$\langle 1,1,2 \rangle$	0.25
$\langle 1,1,3 \rangle$	0.5
$\langle 1,2,1 \rangle$	0.25
$\langle 1,2,2 \rangle$	0.5
$\langle 1,2,3 \rangle$	0.5
$\langle 2,0,3 \rangle$	0.75
$\langle 2,1,2 \rangle$	0.25
$\langle 2,1,3 \rangle$	0.5
$\langle 3,0,3 \rangle$	0.75
all other tuples	0

Table 3. Definition of Constraint c'_2

t	$def^p_{c'_2}(t)$
$\langle 0,1,2 \rangle$	0.25
$\langle 0,2,2 \rangle$	0.5
$\langle 0,3,2 \rangle$	0.5
$\langle 1,2,1 \rangle$	0.25
all other tuples	0

Definition 10. *[6] Let a good enough (abstract) solution for a SCSP P be such that some element in ASolV(P) is in the region* $\hat{\beta}$ *where* $\hat{\beta} = \{\gamma \epsilon A : \beta \leqslant_{S_p} \gamma\}$.

If $ASolV(P) \cap \hat{\beta} \neq \emptyset$ then we have found a good enough solution for a problem P. If this is not the case, we want to find a relaxation P' of P, such that $ASolV(P') \cap \hat{\beta} \neq \emptyset$. P' should be as close to the original P as possible, that is, P' should be such that there does not exist any other relaxation of P that is closer to P than P'.

We first define a relaxation of a single constraint.

Definition 11. *A constraint* $c_j = \langle def_j^p,\ con_j \rangle$ *is called a* c_i*-weakened constraint of the constraint* $c_i = \langle def_i^p,\ con_i \rangle$ *iff the following hold:*

- $con_i = con_j$;
- *for all tuples t,* $def_i^p(t) \leqslant_S def_j^p(t)$;
- *for every two tuples* t_1 *and* t_2*, if* $def_i^p(t_1) \leqslant_{S_p} def_i^p(t_2)$*, then* $def_j^p(t_1) \leqslant_{S_p} def_j^p(t_2)$.

Note that a constraint c is itself a c-weakened constraint.

We want to represent the closeness of a c-weakened constraint to the constraint c by associating a c-semiring value with the c-weakened constraint. Every c-weakened constraint of a constraint c (including the constraint c) will be assigned such a distance value.

Definition 12. *Given a constraint system* $CS = \langle S_p,\ V,\ D \rangle$ *and a SCSP* $P = \langle C,\ con \rangle$*, for each* $c \in C$*, let* W_c *be the set containing all c-weakened constraints, i.e.* $W_c = \{c_j \mid c_j \text{ is a c-weakened constraint}\}$. *Let* $S_d = \langle A_d,\ +_d,\ \times_d,\ \mathbf{0},\ \mathbf{1} \rangle$ *be a c-semiring and* $wdef_c^d : W_c \rightarrow A_d$ *be any function such that*

- $wdef_c^d(c_j) = \mathbf{0}$ *iff* $c_j = c$;
- $\forall c_i, c_j \in W_c$*, if for all tuples t* $def_i^p(t) \leqslant_{S_p} def_j^p(t)$ *then* $wdef_c^d(c_i) \leqslant_{S_d} wdef_c^d(c_j)$;
- *if there exists one tuple t such that* $def_i^p(t) <_{S_p} def_j^p(t)$ *and for all tuples s we have* $def_i^p(s) \leqslant_{S_p} def_j^p(s)$*, then* $wdef_c^d(c_i) <_{S_d} wdef_c d(c_j)$.

Definition 12 describes a function $wdef_c^d$ that assigns c-semiring values (or distance values) from the set of the c-semiring S_d to each c-weakened constraint. This function is restricted by the preference values associated with the tuples of the c-weakened constraints. If the assigned preference values of all the tuples of a c-weakened constraint c_j are at least as good as their assigned preference values in another c-weakened constraint c_i, then the function $wdef_c^d$ will assign a distance value for c_j that is at least as good as the distance value it assigns to c_i. If there is at least one tuple that has a better associated preference value in c_j than in c_i (and all other tuples have associated preference values in c_j that are at least as good as those in c_i), then $wdef_c^d$ will assign a better distance value to c_j than to c_i. (We compare c-semiring values in terms of the partial ordering on them.) This framework is deliberately broad so as to accommodate any reasonable application.

We now define the concept of *closeness* w.r.t. a constraint c and a c-weakened constraint.

Definition 13. – *The c-weakened constraint c_i is closer to c than the c-weakened constraint c_j, iff $wdef_c^d(c_i) <_{S_d} wdef_c^d(c_j)$.*

– *The c-weakened constraint c_i is no closer to c than the c-weakened constraint c_j, iff $wdef_c^d(c_j) \leqslant_{S_d} wdef_c^d(c_i)$.*

– *The c-weakened constraints c_i and c_j are incomparable w.r.t. closeness to c iff $wdef_c^d(c_i) \nleq_{S_d} wdef_c^d(c_j)$ and $wdef_c^d(c_j) \nleq_{S_d} wdef_c^d(c_i)$.*

Below we define a relaxation of a SCSP, and then we describe a way to formalise "closeness" of relaxations.

Definition 14. *A SCSP $P' = \langle C', con\rangle$ is a d-relaxation of the SCSP $P = \langle C, con\rangle$ where $S_d = \langle A_d, +_d, \times_d, \mathbf{0}, \mathbf{1}\rangle$, iff there is a bijection $f : C \rightarrow C'$ and $\forall c \in C, f(c)$ is a c-weakened constraint.*

For every $f(c) \in C'$ and $c \in C$, $wdef_c^d(f(c))$ is an indication of the closeness of $f(c)$ to c. For every $c \in C$, C' contains one c-weakened constraint, i.e. every c can be regarded as being replaced by a c-weakened constraint $f(c)$. We want to find a d-relaxation $P' = \langle C', con\rangle$ of $P = \langle C, con\rangle$ such that every c-weakened constraint $c' \in C'$ is the closest possible to the constraint $c \in C$ while the abstract solution of P' is still good enough (w.r.t. $\hat{\beta}$). It is necessary to place some restrictions on the multiplicative operator $\times_d$ so that the distance of a d-relaxation will indeed reflect the closeness of the relaxed problem to the original problem.

Definition 15. *Let c_{ik} be a c_i-weakened constraint, and c_{jm} and c_{jn} be c_j-weakened constraints. If $wdef_{c_j}^d(c_{jm}) <_{S_d} wdef_{c_j}^d(c_{jn})$, then*

$$wdef_{c_i}^d(c_{ik}) \times_d wdef_{c_j}^d(c_{jm}) <_{S_d} wdef_{c_i}^d(c_{ik}) \times_d wdef_{c_j}^d(c_{jn}).$$

Definition 16. *Let $R(P) = \{P' \mid P'$ is a d-relaxation of $P\}$,*
$R_{\hat{\beta}}(P) = \{P' \in R(P) \mid ASolV(P') \cap \hat{\beta} \neq \emptyset\}$, and

$$ASolR_{\hat{\beta}}(P) = \{\langle t, v\rangle \mid \langle t, v\rangle \in ASol(P') \;\&\; P' \in R_{\hat{\beta}}(P)\}.$$

$R_{\hat{\beta}}(P)$ contains all those SCSPs that are weakened versions of P whose best tuples intersect with $\hat{\beta}$. $ASolR_{\hat{\beta}}(P)$ actually contains those best tuples. Note that every tuple in $ASol(P')$ is a tuple with a maximal c-semiring value.

The next step is to define a distance measure between a problem P and a d-relaxation P'.

Definition 17. *Given a d-relaxation $P' = \langle C', con\rangle$ of a SCSP $P = \langle C, con\rangle$ such that $P' \in R_{\hat{\beta}}(P)$, let $d(P') = \times_{d\ c\in C}\ (wdef_c^d(f(c)))$ be the distance between P and P'.*[5]

Now we have to find every $P' \in R_{\hat{\beta}}(P)$ for which the distance between P' and P is minimal. Thus, let
$MR_{\hat{\beta}}(P) = \{P' \in R_{\hat{\beta}}(P) \mid \nexists\ P'' \in R_{\hat{\beta}}(P)$ such that $d(P'') <_S d(P')\}$.

Example 3. In order to raise the hotel chain's four star rating to a five star rating, the manager has calculated that he needs an abstract solution that provides a c-semiring value of at least 0.75. Our abstract solution to the hotel chain problem is not good enough. We will now find a d-relaxation to this problem with a better solution. We only consider relaxations of the second constraint. Some of the possible c_2-weakened constraints are shown as constraints $c_{21}, \ldots, c_{28}$ in Table 4.

Table 4. Definitions of the c_2-weakened Constraints

t	c_2	c_{21}	c_{22}	c_{23}	c_{24}	c_{25}	c_{26}	c_{27}	c_{28}
$\langle 0,3\rangle$	0.75	1	0.75	0.75	1	1	0.75	1	1
$\langle 1,2\rangle$	0.25	0.25	0.5	0.25	0.5	0.25	0.5	0.5	1
$\langle 1,3\rangle$	0.5	0.5	0.5	0.75	0.5	0.75	0.75	0.75	1
$\langle 2,1\rangle$	0.25	0.25	0.5	0.25	0.5	0.25	0.5	0.5	1
$\langle 2,2\rangle$	0.5	0.5	0.5	0.75	0.5	0.75	0.75	0.75	1
$\langle 2,3\rangle$	0.5	0.5	0.5	0.75	0.5	0.75	0.75	0.75	1
$\langle 3,0\rangle$	0.75	1	0.75	0.75	1	1	0.75	1	1
$\langle 3,1\rangle$	0.5	0.5	0.5	0.75	0.5	0.75	0.75	0.75	1
$\langle 3,2\rangle$	0.5	0.5	0.5	0.75	0.5	0.75	0.75	0.75	1
$\langle 3,3\rangle$	0.5	0.5	0.5	0.75	0.5	0.75	0.75	0.75	1
all other tuples	0	0	0	0	0	0	0	0	0

Let $S_d = \langle\{1,2,3,4,5\},\ min,\ max,\ \infty,\ -\infty\rangle$. Then we can associate the c-semiring values shown in Table 5 with each of the weakened constraints.

We aim to keep our d-relaxation as close as possible to the original problem. Any one of the c_2-weakened constraints with a c-semiring value of 1 would be a good initial choice. Thus, one possible d-relaxation of the problem P is
$P'_1 = \langle C'_1, con\rangle$ with $C'_1 = \{c_1,\ c_{23},\ c_3\}$. The combination of the constraints, $pc_1 = c_1 \otimes c_{23} \otimes c_3$ is shown in Table 6.

Now the abstract solution is $ASol(P'_1) = \{\langle\langle 0,2,2\rangle,\ 0.75\rangle,\ \langle\langle 0,3,2\rangle,\ 0.75\rangle\}$, with $ASolV(P'_1) = \{0.75\}$ and $d(P'_1) = 0 \times_d 1 \times_d 0 = 1$. This means that our abstract solution is good enough, and the manager can raise the star rating of

[5] We use the symbol $\times_d$ in prefix notation when this binary operator is applied to more than two arguments

Table 5. Distance values for the c_2-weakened Constraints

$wdef^d_{c_2}(c_2)$	0
$wdef^d_{c_2}(c_{21})$	1
$wdef^d_{c_2}(c_{22})$	1
$wdef^d_{c_2}(c_{23})$	1
$wdef^d_{c_2}(c_{24})$	2
$wdef^d_{c_2}(c_{25})$	3
$wdef^d_{c_2}(c_{26})$	3
$wdef^d_{c_2}(c_{27})$	4
$wdef^d_{c_2}(c_{28})$	5

Table 6. Definition of Constraint pc_1.

t	$def^p_{pc_1}(t)$
$\langle 0, 1, 2\rangle$	0.25
$\langle 0, 2, 2\rangle$	0.75
$\langle 0, 3, 2\rangle$	0.75
$\langle 1, 2, 1\rangle$	0.25
all other tuples	0

the hotel chain by selecting either one of the two tuples in the set $ASol(P'_1)$ as a solution.

4 Related Work: Metric SCSPs

Ghose & Harvey [6] extended the SCSP framework by specifying a metric for each constraint in addition to the preference values that are associated with the tuples of values for that constraint. The metric provides real valued distances between the preference values. Metric SCSPs are similar to our proposal in the sense that both frameworks allow us to establish whether a solution is regarded as being good enough. Both approaches obtain a measure of the deviation required from a problem P to a relaxation of P that has a good enough solution.

For Metric SCSPs, the definition of a constraint (Definition 3) is modified by including a metric $d_c : A \times A \rightarrow \mathbb{R}^+$ expressing the perceived difference between c-semiring values. Each constraint is a triple $c = \langle def^p_c, con_c, d_c\rangle$ where con_c are the variables to be operated on, def^p_c is a function matching tuples to values in the set of a c-semiring, and a metric d_c. The formal properties of the metric are given in [6].

If a c-semiring $S_p = \langle A, +_p, \times_p, \mathbf{0}, \mathbf{1}\rangle$ is used to assign preference values to the tuples of values of constraints, the distance of a preference (or c-semiring) value α to a region $\hat{\beta}$ (see Definition 10) is defined as $d(\alpha, \hat{\beta}) = inf\{d(\alpha, \gamma) : \gamma \in \hat{\beta}\}$.

Note that given two c-semiring (preference) values, α and γ, with $\gamma \leq_{S_p} \alpha$, we have $d(\alpha, \hat{\beta}) \leq d(\gamma, \hat{\beta})$.

In the definition of a Metric SCSP which follows below, an additional function f is added. This function will be used to combine distance values provided by the metric functions of the constraints.

Definition 18. *[6] Given a constraint system $CS = \langle S_p, D, V \rangle$, a Metric SCSP is a triple $P = \langle C, con, f \rangle$ where con is a set of variables, $C = \{c_1, c_2, \ldots, c_m\}$ is a finite set of constraints, and $f : (\mathbb{R}^+)^m \rightarrow \mathbb{R}^+$ is used for combining the results of the functions d_{c_i} for all $i = 1, \ldots, m$.*

The following two properties are imposed on the function f in Definition 18: if $f(x_1, \ldots, x_m) = 0 \Leftrightarrow \forall i, x_i = 0$, and f is monotonic increasing in each argument. The aim is to find solution(s) such that minimal deviation is required from the SCSP while ensuring they are assigned a c-semiring value in a specified region $\hat{\beta}$. The value for a solution of a Metric SCSP, as defined for SCSPs, is $t = def^p(t) = (def^p_{c_1}(t \downarrow^{con}_{con_{c_1}}) \otimes \ldots \otimes (def^p_{c_m}(t \downarrow^{con}_{con_{c_m}})$. To ensure that the value $def(t)^p$ is in $\hat{\beta}$ we need only ensure that all $def^p_{c_i}$ are also within $\hat{\beta}$.

Let $f_\beta(t) = f(d_1(def^p_{c_1}(t \downarrow^{con}_{con_{c_1}}), \hat{\beta}), ..., d_m(def^p_{c_m}(t \downarrow^{con}_{con_{c_m}}), \hat{\beta}))$. The function f_β determines the deviation from P required to move $def^p(t)$ into the region $\hat{\beta}$. Let $m^*_\beta = min\{f_\beta : u \in ASol(P)\}$ represent the minimum deviation from the problem P required to find a complete tuple with a semiring value in $\hat{\beta}$.

To summarise, the function f_β provides us with a measurement of how much a problem P should be relaxed in order to provide a good enough solution. This measurement is calculated by combining the distance between the maximal tuple for each constraint and $\hat{\beta}$.

In our work, we describe how to construct a relaxation that has a good enough solution by relaxing constraints. We decide which tuple is a maximal choice for each constraint by ensuring that the preference value of the combination of all the relaxed (or weakened) constraints will lie in the region $\hat{\beta}$ with the least possible deviation from the original constraints.

5 Conclusion and Future Work

We have proposed an extension to the SCSP framework for solving Constraint Satisfaction Problems where a relaxation of a SCSP is defined and solved in case an acceptable solution for the original SCSP can not be found.

If the preference value associated with the solution of a SCSP is not regarded as good enough, we showed how to find a suitable relaxation of the SCSP that has a good enough solution. A relaxation to a SCSP is found by adjusting the preferences associated with the tuples of some of the constraints of the original SCSP. In other words, the constraints of the original problem are relaxed until the resulting problem has a satisfactory solution. Distance values (i.e. c-semiring values) are associated with each relaxed constraint so that different relaxations of a problem can be compared in terms of their distance to the original problem.

Metric SCSPs are related to our work. A metric function calculates a real valued distance between preference values. These distance values are used to measure the deviation of a solution to a SCSP from some desired solution that is good enough.

In this paper we have described how to construct acceptable relaxations for a SCSP with an unsatisfactory solution. Our future work will focus on computational aspects of this process. We aim to develop techniques to calculate the best relaxation for a SCSP efficiently. We want to impose structure on the definitions that respectively assign preference values to tuples of values for constraints and distance values to relaxed constraints, so that existing CSP algorithms can be applied to find the best d-relaxation for a SCSP.

References

1. Bistarelli, S., Montanari, U., Rossi, F.: Semiring-based constraint satisfaction and optimization. Journal of the ACM **44** (1997) 201–236
2. Freuder, E.C., Wallace, R.J.: Partial constraint staisfaction. Artificial Intelligence **58** (1992) 21–70
3. Wilson, M., Borning, A.: Hierarchical constraint logic programming. Journal of Logic Programming **16** (1993) 277–318
4. Dubois, D., Fargier, H., Prade, H.: The calculus of fuzzy restrictions as a basis for flexible constraint satisfaction. In: Proc. of IEEE Conference on Fuzzy Systems. (1993)
5. Fargier, H., Lang, J.: Uncertainty in constraint satisfaction problems: a probabilistic approach. In: Proc. ECSQARU. (1993)
6. Ghose, A., Harvey, P.: Partial constraint satisfaction via semiring CSPs augmented with metrics. In: Proceedings of the 2002 Australian Joint Conference on Artificial Intelligence. Volume 2557 of Lecture Notes in Computer Science., Springer (2002)