Available online at www.sciencedirect.com
ScienceDirect
Information Processing Letters 103 (2007) 177–182
Information Processing Letters
www.elsevier.com/locate/ipl

Relaxations of semiring constraint satisfaction problems

Louise Leenen [a,*], Thomas Meyer [b], Aditya Ghose [a]

[a] *Decision Systems Laboratory, School of Computer Science and Software Engineering, University of Wollongong, Australia*
[b] *Meraka Institute, CSIR, Pretoria, South Africa*

Received 2 November 2005; received in revised form 9 November 2006; accepted 20 March 2007

Available online 5 April 2007

Communicated by A. Moffat

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. [S. Bistarelli, U. Montanari, F. Rossi, Semiring-based constraint solving and optimization, Journal of the ACM 44 (2) (1997) 201–236] defines a maximal solution to a SCSP as the best set of solution tuples for the variables in the problem. Sometimes this maximal 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 measure of the difference between the original SCSP and the relaxed SCSP. We introduce a relaxation scheme but do not address the computational aspects.

Keywords: Semiring constraint satisfaction problems; Over-constrained problems; Preferences and soft constraints; Combinatorial problems

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. This is because many real-life problems are inherently over-constrained and also because it is difficult for human users to peruse a given set of constraints that have been obtained for a given problem to determine if it is solvable. In general 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 focused 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 [6]). Subsequent efforts considered more fine-grained notions of relaxation, where entire constraints did not have to be removed (the HCLP framework [8], Fuzzy CSPs [4], Probabilistic CSPs [5]).

Bistarelli et al. [3] proposed an abstract semiring CSP scheme (SCSP) that generalized most earlier attempts, while making it possible to define several useful new instances of the scheme. The SCSP scheme as-

* Corresponding author.
E-mail address: leenenl@iinet.net.au (L. Leenen).

0020-0190/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2007.03.011

sumes 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 comparison. While a classical constraint defines which combinations of value assignments to the variables in its signature are allowed, a SCSP constraint assigns a preference value to all possible value assignments to the variables in its signature. These preferences implicitly define an approach to alter constraints ("try to satisfy the constraint using the most preferred tuples, else try the next most preferred tuples" and so on).

Our aim in this paper is to define how a SCSP might be relaxed. This may appear counter-intuitive, since a SCSP is intended to define how soft constraints are relaxed. We 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). Consider a situation where the optimal solution obtained fails to meet this threshold (e.g., the optimal profit figure is too low). We are interested in seeking a new (relaxed) set of constraints C' that is minimally different from the original set C 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.

We attempt such an exercise in the context of SCSPs. A SCSP does not have an explicit objective function. Objectives are implicitly articulated 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 a SCSP. Consider a 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 use as a running example a problem involving a hotel that is unable to attain a five-star rating and wishes to determine the minimal changes required to its infrastructure in order to achieve such a rating. The star rating of the hotel is modeled via semiring preference values. We propose a relaxation scheme for SCSPs but further research is required to develop efficient algorithms to compute these relaxations.

2. The SCSP framework

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:[1]
  - 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;
- $\times$ is a commutative, associative, and binary operation such that $\mathbf{1}$ is its unit element and $\mathbf{0}$ is its absorbing element;
- 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 c-value of the projection of a tuple of values over a set of variables onto a subset of the variables. We derive a partial ordering $\leqslant_S$ over the set A: $\alpha \leqslant_S \beta$ iff $\alpha + \beta = \beta$.[2] The minimum element in the ordering is $\mathbf{0}$, while $\mathbf{1}$ 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.

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 $def_c^p : D^k \to A_p$ (where k is the cardinality of con_c) is called the value of the constraint.

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 $def_{c_1}^p = def_{c_2}^p$.

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

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

Table 1
Constraint definitions

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

Example 1. A hotel chain acquires a star rating that is an accumulation of the different branches. Currently it has a four star rating and it aims for a five star rating. Various renovations can be done at branches to increase the rating of the hotel: lay new carpets, upgrade a swimming pool, or paint the building. The manager has to choose which (minimal) renovations to do at which branches under certain restrictions (such as the budget). We express the problem as a SCSP where the semiring structure allows the manager to express his preferences for particular tuples of domain values of the constraints. X, Y and Z denote the three branches. At most one job at a time can be done at a particular branch, and, in total, as few jobs as possible should be done.

Let $CS = \langle S_p, D, V\rangle$ and $P = \langle C, con\rangle$, 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 the value 0 represent no job, the value 1 represent re-carpeting, the value 2 represent pool renovation, and the value 3 represent painting. A renovation job with a higher value will contribute more towards a higher star rating. Assume 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 of these constraints together with their preference values (i.e. associated semiring values) are given in Table 1. The manager's choice of semiring value represents the desirability of that particular tuple. Consider $def_{c_1}^p(\langle 0, 2\rangle) = 0.75$: tuple $\langle 0, 2\rangle$ of constraint c_1 represents the case where nothing is to be done at branch X while branch Y's swimming pool is to be upgraded. Its high preference value indicates that it is preferred, for instance, to the tuple $\langle 1, 1\rangle$ with a value of 0.25. A tuple with an associated value of 0 is highly undesirable.

Definition 5. Given a constraint system $CS = \langle S_p, D, V\rangle$ with V 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 $i \leqslant j$ and $w'_i \preceq w'_j$ if $i \leqslant j$. 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 $t'_i = t_j$ if $w'_i = w_j$.

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 $c_1 = \langle def_{c_1}^p, con_{c_1}\rangle$ and $c_2 = \langle def_{c_2}^p, con_{c_2}\rangle$ 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 $def_c^p(t) = def_{c_1}^p(t\downarrow^{con_c}_{con_{c_1}}) \times_p def_{c_2}^p(t\downarrow^{con_c}_{con_{c_2}})$. Let $(\bigotimes C)$ denote $c_1 \otimes c_2 \otimes \cdots \otimes c_n$ with $C = \{c_1, \ldots, c_n\}$.

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 $def_{c'}^p(t') = \sum_{\{t \mid t\downarrow^{con_c}_{I\cap con_c} = t'\}} def_c^p(t)$.

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. A *maximal solution* consists of the set of k-tuples of D whose associated c-semiring values are maximal with respect to $\leqslant_{S_p}$.

Definition 9. Given a SCSP problem $P = \langle C, con\rangle$ with $Sol(P) = \langle def_c^p, con\rangle$, the maximal 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. First combine c_1 and c_2 to get the resulting constraint $c'_1 = c_1 \otimes c_2$. For example, $def_{c_1}^p(\langle 0, 1\rangle) \times_p def_{c_2}^p(\langle 1, 3\rangle) = \min(0.25, 0.5) = 0.25$, so tuple $\langle 0, 1, 3\rangle$ of constraint c'_1 has a preference value 0.25. Now combine c'_1 and c_3 to get $c'_2 = c'_1 \otimes c_3$. The following tuples of constraint $c'_2 = c'_1 \otimes c_3$ all have a preference value of 0.5 and form the maximal solution with $ASolV(P) = \{0.5\}$: $\langle 0,2,2\rangle$; $\langle 0,3,2\rangle$; $\langle 1,2,2\rangle$; $\langle 1,3,2\rangle$; $\langle 2,0,2\rangle$; $\langle 2,1,2\rangle$; $\langle 2,2,0\rangle$; $\langle 2,2,1\rangle$; $\langle 2,2,2\rangle$; $\langle 3,0,2\rangle$; $\langle 3,1,2\rangle$; $\langle 3,2,0\rangle$; $\langle 3,2,1\rangle$ and $\langle 3,2,2\rangle$. All other tuples have preference values of either 0.25 or 0.

3. A relaxation of a SCSP

We are interested in a SCSP for which the maximal solution is not considered to be good enough. For example, our hotel manager may require 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.

Definition 10. A nonempty subset F of a partially ordered set $(A, \leqslant_A)$ is called a filter, if the following conditions hold:

- For every $x, y \in F$, there is some element $z \in F$, such that $z \leqslant_A x$ and $z \leqslant_A y$.
- For every $x \in F$ and $y \in A$, $x \leqslant_A y$ implies that $y \in F$.

The smallest filter that contains a given element α is a principal filter and α is called its principal element. The principal filter for α is given by the set $\uparrow\alpha = \{x \in A \mid \alpha \leqslant_A x\}$.

We identify a partially ordered set of "lower bound" preference values that are regarded as being good enough.

Definition 11. Let a good enough (maximal) solution for a SCSP P be such that some element in $ASolV(P)$ is in the set $LB = \{\bigcup(\uparrow\beta) \mid \beta \in A\}$ of sufficient preference values.

If $ASolV(P) \cap LP \neq \emptyset$ we have found a good enough solution for P. Otherwise, we want to find a relaxation P' of P, such that $ASolV(P') \cap LB \neq \emptyset$. There should not exist any relaxation of P that is closer to P than P'.

Definition 12. 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: $con_i = con_j$, and for all tuples t, $def_i^p(t) \leqslant_S def_j^p(t)$.

We represent the closeness of a c-weakened constraint to the constraint c by associating a c-semiring value with the c-weakened constraint.

Definition 13. 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. $W_c = \{c_j \mid c_j \text{ is a } c\text{-weakened constraint}\}$. Let $S_d = \langle A_d, +_d, \times_d, \mathbf{0}, \mathbf{1}\rangle$ be a c-semiring and $wdef_c^d : W_c \to A_d$ be any function such that

- A_d is a well-founded set (it contains no infinite descending chains);
- $wdef_c^d(c_j) = \mathbf{0}$ 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 $wdef_c^d(c_i) <_{S_d} wdef_c^d(c_j)$.

We use $+_d$ for comparing and $\times_d$ for combining c-semiring values (see Definition 17). Observe that the set A_d is restricted to sets that do not contain infinite chains of weaker values such as the set of reals.

The function $wdef_c^d$ assigns c-semiring (difference) values from the set of the c-semiring S_d to each c-weakened constraint. If the preference values of all the tuples of a c-weakened constraint c_j are at least as good as their preference values in another c-weakened constraint c_i, $wdef_c^d$ assigns a difference value for c_j that is at least as good as the difference value it assigns to c_i. If there is at least one tuple that has a better preference value in c_j than in c_i (and all other tuples have preference values in c_j that are at least as good as those in c_i), then $wdef_c^d$ will assign a better difference value to c_j than to c_i. This framework is deliberately broad so as to accommodate any reasonable application.

Definition 14.

- 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 with respect to closeness to c iff $wdef_c^d(c_i) \nleqslant_{S_d} wdef_c^d(c_j)$ and $wdef_c^d(c_j) \nleqslant_{S_d} wdef_c^d(c_i)$.

Table 2
Definitions of the c_2-weakened constraints

t	c_2	c_{2_1}	c_{2_2}	c_{2_3}	...	c_{2_8}	c_{2_9}	$c_{2_{10}}$	$c_{2_{11}}$	...
$\langle 0,1\rangle$	0.25	0.25	0.25	0.25	...	0.75	0.25	0.25	0.25	...
$\langle 0,2\rangle$	0.75	0.75	0.75	0.75	...	0.75	0.75	0.75	0.75	...
$\langle 0,3\rangle$	1	1	1	1	...	1	1	1	1	...
$\langle 1,0\rangle$	0.25	0.25	0.25	0.25	...	0.25	0.75	0.25	0.25	...
$\langle 1,1\rangle$	0.25	0.25	0.25	0.25	...	0.25	0.25	0.75	0.25	...
$\langle 1,2\rangle$	0.5	0.75	0.5	0.5	...	0.5	0.5	0.5	0.5	...
$\langle 1,3\rangle$	0.5	0.5	0.75	0.5	...	0.5	0.5	0.5	0.5	...
$\langle 2,0\rangle$	0.5	0.5	0.5	0.75	...	0.5	0.5	0.5	0.5	...
$\langle 2,1\rangle$	0.5	0.5	0.5	0.5	...	0.5	0.5	0.5	0.5	...
$\langle 2,2\rangle$	0.5	0.5	0.5	0.5	...	0.5	0.5	0.5	0.5	...
$\langle 2,3\rangle$	0.25	0.25	0.25	0.25	...	0.25	0.25	0.25	0.75	...
$\langle 3,0\rangle$	1	1	1	1	...	1	1	1	1	...
$\langle 3,1\rangle$	0.5	0.5	0.5	0.5	...	0.5	0.5	0.5	0.5	...
$\langle 3,2\rangle$	0.5	0.5	0.5	0.5	...	0.5	0.5	0.5	0.5	...
all other tuples	0	0	0	0	0	0	0	0	0	...

Definition 15. 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 $f : C \to C'$ and $\forall c \in C$, $f(c)$ is a c-weakened constraint.

For a $f(c) \in C'$ 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 maximal solution of P' is still good enough (with respect to the set LB).

Theorem 3.1. *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

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

Proof. The multiplicative operator of a c-semiring is monotone on a partial order on the set of a c-semiring (see [3]). Thus $\times_d$ is monotone on $<_{S_d}$. □

Definition 16. Let $R(P) = \{P' \mid P'$ is a d-relaxation of $P\}$, $R_{LB}(P) = \{P' \in R(P) \mid ASolV(P') \cap LB \neq \emptyset\}$, and $ASolR_{LB}(P) = \{\langle t, v\rangle \mid \langle t, v\rangle \in ASol(P')$ and $P' \in R_{LB}(P)\}$.

$R_{LB}(P)$ contains all those SCSPs that are weakened versions of P whose best tuples intersect with LB and $ASolR_{LB}(P)$ contains those best tuples. We define a measure of difference 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 $P' \in R_{LB}(P)$, let $d(P') = \times_{d c\in C}(wdef_c^d(f(c)))$ be the difference between P and P'. [3]

We have to find every $P' \in R_{LB}(P)$ with a minimal difference between P' and P. Let $MR_{LB}(P) = \{P' \in R_{LB}(P) \mid \nexists P'' \in R_{LB}(P)$ such that $d(P'') <_S d(P')\}$.

Example 3. The manager needs a solution that gives a semiring value of at least 0.75. In our attempt to find a d-relaxation to this problem with a sufficient solution we only consider some relaxations of the second constraint as shown in Table 2. In each of the constraints c_{2_4} up to c_{2_7} we only changed a single tuple's value from 0.5 to 0.75. With $S_d = \langle\{0, 1, 2, 3, 4, 5\}, \min, \max, 5, 0\rangle$ we associate the following c-semiring values with each of the relaxed constraints: $wdef_{c_2}^d(c_2) = 0$, $wdef_{c_2}^d(c_{2_1})$ up to $wdef_{c_2}^d(c_{2_7}) = 1$, and $wdef_{c_2}^d(c_{2_8})$ up to $wdef_{c_2}^d(c_{2_{11}}) = 2$. The function $wdef$ should reflect the manager's preferred way to relax constraints: $wdef$ assigns a smaller value for relaxations where the least number of changes to tuples' preference values and the smallest adjustments have been made, i.e., a change from 0.5 to 0.75 is regarded as being smaller than a change from 0.25 to 0.75. There are a number of other possible relaxations of which the most relaxed one will

[3] We use $\times_d$ in prefix notation when it is applied to more than two arguments.

be a constraint where all tuples have a preference value of 1. Our d-relaxation must be as close as possible to the original problem. We initially consider any one of the c_2-weakened constraints with a c-semiring value of 1. If we select the relaxation c_{2_3} we will be able to raise one of the maximal solution tuples above the others. One d-relaxation of the problem P is $P'_1 = \langle C'_1, con\rangle$ with $C'_1 = \{c_1, c_{2_3}, c_3\}$. The combination of the constraints, $pc_1 = c_1 \otimes c_{2_3} \otimes c_3$ gives $def^p_{pc_1}(\langle 2, 0, 2\rangle) = 0.75$. The manager can raise the star rating of the hotel chain by selecting this solution tuple.

4. Conclusion, related work, and future work

We have proposed an extension to the SCSP framework for solving CSPs where a relaxation of a SCSP is constructed if the solution for the original SCSP is not good enough. We define a suitable relaxation of the SCSP by adjusting the preferences associated with the tuples of some of the constraints of the original SCSP. Difference 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 difference from the original problem.

Our future work will focus on computational aspects of this process. When a solution to a SCSP P is not good enough, we use a set of cut-off values, LB, to define a threshold that should be reached. We need only consider relaxations to constraints of P that have the potential to form a good enough solution. Such relaxed constraints can be found by looking for at least one tuple in the original constraint with a preference value that is not in the set LB and then raise it so that it has a preference value in LB. We plan to develop efficient algorithms to find suitable subsets of relaxations, and to develop techniques to calculate the best relaxation for a SCSP efficiently.

Bistarelli et al. [1] use the semiring-based framework to model partial CSPs: they show how to use a semiring to represent a notion of distance between a solution and a problem. It has been shown that tradeoffs between user preferences (if all requirements cannot be met) can be modeled as additional constraints. Bistarelli et al. [2] presents a framework where "tradeoffs" between preferences are modeled in the semiring framework. Our work can be seen as a form of tradeoff where the added and removed constraints involve the same variables. Ghose and Harvey [7] extended the SCSP framework by specifying a metric for each constraint in addition to the preference values associated with the tuples of values. The metric provides real valued differences between the preference values which are used to measure the deviation of a solution to a SCSP from some desired solution that is good enough.

References

[1] S. Bistarelli, E. Freuder, B. O'Sullivan, Encoding partial constraint satisfaction in the semiring-based framework for soft constraints, in: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, 2004.

[2] S. Bistarelli, J. Kelleher, B. O'Sullivan, Tradeoff generation using soft constraints, in: Proceedings of CSCLP, 2003.

[3] S. Bistarelli, U. Montanari, F. Rossi, Semiring-based constraint solving and optimization, Journal of the ACM 44 (2) (1997) 201–236.

[4] D. Dubois, H. Fargier, H. Prade, The calculus of fuzzy restrictions as a basis for flexible constraint satisfaction, in: Proceedings of the IEEE Conference on Fuzzy Systems, 1993.

[5] H. Fargier, J. Lang, Uncertainty in constraint satisfaction problems: a probabilistic approach, in: Proceedings of the ECSQARU, 1993.

[6] E.C. Freuder, J.W. Wallace, Partial constraint satisfaction, Artificial Intelligence 58 (1992) 21–70.

[7] A. Ghose, P. Harvey, Partial constraint satisfaction via semiring CSPs augmented with metrics, in: Proceedings of the Australian Joint Conference on AI, in: Lecture Notes in Computer Science, vol. 2557, Springer, 2002.

[8] M. Wilson, A. Borning, Hierarchical constraint logic programming, Journal of Logic Programming 16 (1993) 277–318.