Soft Constraints

Models, Algorithms, Applications

Pedro Meseguer

IIIA-CSIC, Spain

Francesca Rossi

University of Padova, Italy

Thomas Schiex

INRA, France

Outline of the tutorial

- Part 1 – Soft constraints: models
  Francesca Rossi
- Part 2 – Soft constraints: algorithms
  Pedro Meseguer
  - Systematic search
  - Local search
  - Approximation methods
- Part 3 – Soft constraints: applications
  Thomas Schiex
  - RNA secondary structure prediction
  - satellite scheduling
  - radio link frequency assignment

PART 1 – Soft Constraints: Models

- Motivation
- Examples of soft problems in real-life
- Specific soft CSP models: fuzzy, lexicographic, weighted, probabilistic CSPs
- Generic soft CSP models: hierarchical, partial, valued, semiring-based CSPs, instances
- Soft temporal CSPs
- Soft constraint propagation
- Global and local preferences

Motivations for soft constraints

- Hard constraint problems (CSPs):
  - variables over finite domains
  - constraints: tuples of domain values are either allowed or not
- Most real-life situations need fuzziness, possibilities, preferences, probabilities, costs, . . . :
  - Over-constrained problems
  - Problems with both preferences and hard statements, and/or uncertainties
  - Optimization problems (also multi-criteria)
- Soft constraints: preferences rather than strict requirements (a tuple or constraint has a level of preference)

Time-tabling problems

- Hard constraints:
  - number of rooms and courses
  - estimated audience for each course
  - size of each room
  - number of lectures every week
  - a professor cannot teach two lectures at the same time
- Soft constraints (preferences):
  - different days for different lectures
  - teachers' preferences over days and times
  - order of the lectures of different courses

A fuzzy problem

CP'01

- To decide what to eat for dinner at a restaurant
- Preferences (for example integers) over combinations of drinks and dishes:
  - water and meat: 0.4, red wine and meat: 0.7
- Preferences also over the type of dish (and drink):
  - fish: 0.8, meat: 0.3
  - water: 0.7, red wine: 0.8, white wine: 1
- Goal: to find a combination which maximizes the overall preference (min, conjunctive fuzzy problem)
- meat and red wine: 0.3 = min(0.3,0.7,0.8)
  meat and water: 0.3 = min(0.3,0.4,0.7)

A hierarchical problem

- Place some pieces of furniture in an office
- Some most important constraints:
  - chair close to the table
- Medium-importance constraints:
  - computer not in front of the window
- Not-so-important constraints:
  - window visible from the chair
- Goal: find a solution which satisfies the highest number of constraints, with precedence to the more important ones

Temporal preferences

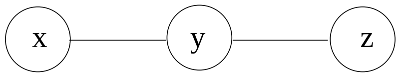
- Many events to be scheduled over the time line
- Constraints give ranges for their duration and distance
- Each element in a range has a level of preference:
  - to minimize the delay, a decreasing preference function over the distance range
- Goal: find a most preferred scheduling of the events

Specific soft CSP models

- Fuzzy CSPs
- Lexicographic CSPs
- Weighted CSPs
- Probabilistic CSPs

Fuzzy CSPs

- A preference level to each tuple of values, between 0 (worst value) and 1 (best value)
- Value associated with a complete instantiation: the minimum of the values of all its subtuples
- Best solution = complete instantiation with maximum value



x y	y z
a a ... 0	a a ... 0.3
a b ... 1	a b ... 0.1
b a ... 0.5	b a ... 1
b b ... 0.7	b b ... 1

Solutions:

a a a ... min(0,0.3) = 0
a a b ... min(0,0.1) = 0
a b a ... min(1,1) = 1
a b b ... min(1,1) = 1
b a a ... min(0.5,0.3) = 0.3
b a b ... min(0.5,0.1) = 0.1
b b a ... min(0.7,1) = 0.7
b b b ... min(0.7,1) = 0.7

Best solutions:

a b a ... 1
a b b ... 1

Dubois, Fargier, Prade, IEEE Fuzzy Systems 1993; Ruttkay, Fuzzy Systems 1994; Schiex, UAI 1992

Weighted CSPs

- Each tuple of values, or constraint, has a cost
- Cost of a complete assignment: sum of costs of all tuples
- Goal: to minimize the overall cost
- Max-CSPs:
  - weighted CSPs where each constraint has a weight 0 if satisfied, and 1 if violated
  - weigth of a complete assignment: number of violated constraints
  - goal: to minimize the number of violated constraints

Lexicographic CSPs

- Combination of weighted and fuzzy CSPs
- Value of a solution:
  - not just the min
  - it depends also on the number of violated constraints at each preference level
- Multiset of preferences in [0,1], combined via multiset union
- Lexicographic order to compare solutions
- Example: meat and water: $\langle 0.3, 0.4, 0.7 \rangle$
  meat and red wine: $\langle 0.3, 0.7, 0.8 \rangle \Rightarrow$ better!

Fargier, Lang, Schiex, EUFIT 1993

Probabilistic CSPs

- To reason about problems which are only partially known
- Each constraint c has a certain independent probability $p(c)$ to be part of the given real problem
- Value of a complete instantiation t: probability that it is a solution of the real problem $\Rightarrow$ product of all $1 - p(c)$ for all c violated by t, 1 otherwise
- We want the instantiation with the maximum probability

H. Fargier, J. Lang, ECSQARU 1993

Generic soft CSP models

- Originally for over-constrained CSPs:
  - hierarchical CSPs: hierarchy of importance for constraints
  - partial CSPs: only some constraints are satisfied
- For soft CSPs:
  - Valued CSPs
    - preference: impact for a constraint violation
    - best solutions: minimum global preference
  - Semiring-based CSPs
    - preference: likeness for a tuple (a way to satisfy a constraint)
    - best solutions: best preference

Hierarchical CSPs

- A strength level for each constraint (ordered: required, strong, weak, ...)
- Find the solutions which satisfy all required constraints and the other constraints as much as possible
- Pre-defined comparators on solutions
- Example: we move with the mouse one endpoint of a horizontal line in a window
  - `required`: the line must remain horizontal and must not exit the window
  - `strong`: the endpoint must follow the mouse

Borning et al., ICLP 1989 (HCLP)

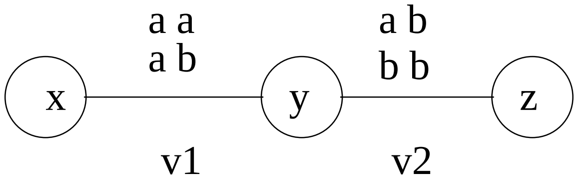
Partial CSPs

- When not all constraints can be satisfied, a solution satisfies only some of them
- Metric to choose among solutions:
  - example: count the difference in the number of constraints
- Example: 3-queen problem (unsolvable)
  - diagonal attack $\Rightarrow$ constraint enlarging
  - expand to a 4x3 grid $\Rightarrow$ domain enlarging
- Goal: to solve a problem which is closest to the original one, according to the metric

E. Freuder, R. Wallace, AI Journal, 1992.

Valued CSPs

- Valuations belong to a totally ordered set (commutative monoid):
  - minimum (best) element $\perp$
  - operation $*$ to combine valuations
- Global valuation: combines the valuations of all the constraints violated by it
- Goal: assignment with a minimum valuation



Solutions:

a a b ... $\perp$
a b b ... $\perp$
a b a ... v2
b a b ... v1
b b a ... v1 $*$ v2

T. Schiex, IJCAI 1995

Semiring-based CSPs

CP'01

- A set of preferences A to be associated to tuples of values in each constraint
- an operation $\times$ to combine the preferences
- an operation $+$ to compare the preferences $\Rightarrow$ partial order: $a \leq b$ iff a+b = b (b is better than a)
- $\langle A, +, \times, 0, 1 \rangle$ is a semiring
- C-semiring: semiring plus
  - $+$ idempotent (to get a partial order over A)
  - $\times$ commutative
  - $a + 1 = 1$

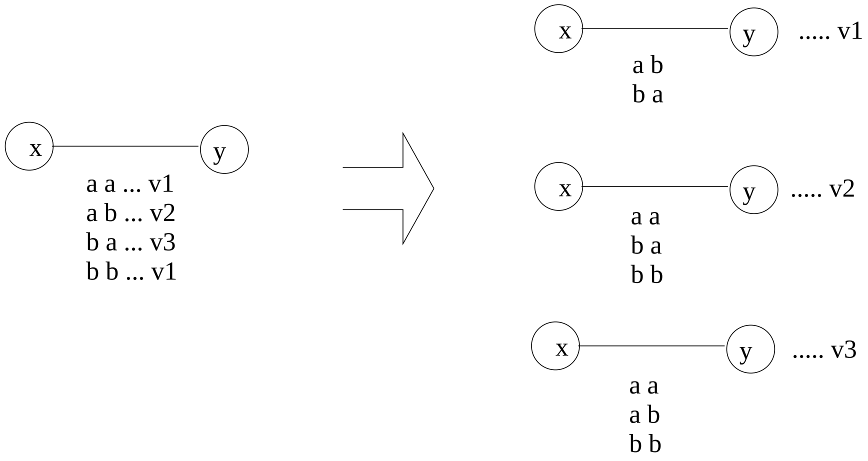
Bistarelli, Montanari, Rossi, IJCAI 1995, JACM 1997

Differences between VCSPs and SCSPs

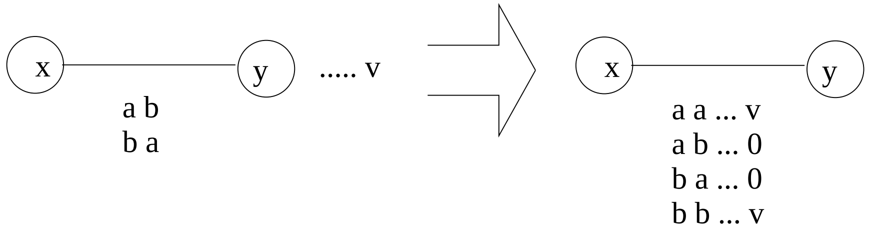
- Preferences to
  - tuples (semiring-based CSPs)
  - constraints (valued CSPs)
- Order of the preferences:
  - total (valued CSPs)
  - partial (semiring-based CSPs)

Preference to constraints or to tuples

- From tuples to constraints:



- From constraints to tuples:



Bistarelli, Fargier et al., LNCS 1106, 1996

When a partial order is useful

- Set-based CSPs:
  - preferences are sets of elements
  - combined via intersection, compared via union
  - c-semiring $S_{set} = \langle \wp(A), \bigcup, \bigcap, \emptyset, A \rangle$
    $\Rightarrow$ order = set inclusion
- Multi-criteria CSPs:
  - one $S_i = \langle A_i, +_i, \times_i, \mathbf{0_i}, \mathbf{1_i} \rangle$ for each criteria
  - $\langle\langle A_1, \ldots, A_n \rangle, +, \times, \langle \mathbf{0_1} i, \ldots, \mathbf{0_n} \rangle, \langle \mathbf{1_1} \ldots \mathbf{1_n} \rangle\rangle$
  - $+$ and $\times$ obtained by pointwise application of $+_i$ and $\times_i$ on each S_i
  - Partial order even if all criteria totally ordered

Instances of SCSPs and VCSPs

- CSPs: semiring $\langle \{false, true\}, \vee, \wedge, false, true \rangle$
- Fuzzy CSPs: $\langle [0,1], max, min, 0, 1 \rangle$:
- Probabilistic CSPs: $\langle [0,1], max, \times, 0, 1 \rangle$
- Weighted CSPs: $\langle \mathcal{R}^+, min, +, +\infty, 0 \rangle$

Soft temporal CSPs

- Temporal constraints as intervals: $a \leq X - Y \leq b$
- A preference for each element in the interval
- Choice of a specific semiring: combination and comparison of preferences via $\times$ and $+$
- Hard temporal CSPs are tractable if one interval per constraint (Dechter)
- Soft temporal CSPs are tractable if
  - one interval per constraint
  - preferences with at most one local maximum
  - (idempotent $\times$ if we use path-consistency)

Khatib et al. IJCAI 2001 (soft TCSPs solver)

Combination and Projection

- Constraint $c = \langle def, con \rangle$
  - def: association tuples-preferences
  - con: set of variables
- Projection: $c \Downarrow_I = \langle def', I \cap con \rangle$, where

$$def'(t') = \Sigma_{\{t \mid t \downarrow^{con}_{I \cap con} = t\}} def(t)$$

- Combination: $c_1 \otimes c_2 = \langle def, con_1 \cup con_2 \rangle$, where

$$def(t) = def_1(t \downarrow^{con}_{con_1}) \times def_2(t \downarrow^{con}_{con_2})$$

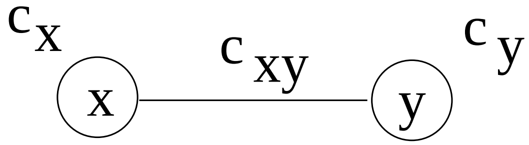
- Examples:
  - CSPs: logical or, logical and
  - fuzzy CSPs: max, min

Soft constraint propagation

- For hard CSPs: to eliminate inconsistencies prior or during the search
- For soft CSPs: to get more "realistic" preferences $\Rightarrow$ tighter bounds during the search for an optimal solution
- C is *k-consistent* if, for all subsets of k vars W and any other var x: $\otimes\{c_i \mid c_i \in C \wedge con_i \subseteq W\} = (\otimes\{c_i \mid c_i \in C \wedge con_i \subseteq (W \cup \{x\})\}) \Downarrow_W$
- Considering only the constraints in W is the same as considering all those in W, plus those connecting x to W, and then projecting over W

Soft arc consistency

- $k = 2$, $W = \{y\}$: $c_x = (\bigotimes\{c_y, c_{xy}, c_x\}) \Downarrow_x$



- CSPs: any value in the domain of x can be extended to a value in the domain of y such that c_{xy} is satisfied
- Fuzzy CSPs: the preference given to a value for x by c_x is the same as that given by $(\bigotimes\{c_y, c_{xy}, c_x\}) \Downarrow_x$
- To achieve SAC: for each x and y, change the definition of c_x to make it coincide with $(\bigotimes\{c_y, c_{xy}, c_x\}) \Downarrow_x$, and iterate fairly until stability

Properties of soft constraint propagation

- $\times$ idempotent $\Rightarrow$
  - equivalence
  - termination
  - order-independence
- Some results also for non-idempotent operators (but we need another operation to compensate for the additional work)

S. Bistarelli, R. Gennari, F. Rossi, CP 2000;

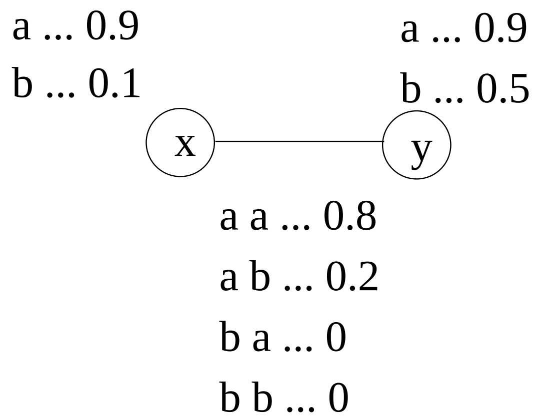
T.Schiex, CP 2000

Examples

- CSPs: $\langle\{0,1\},\vee,\wedge,0,1\rangle$:
  $\wedge$ idempotent $\Rightarrow$ equivalence, order-independence and termination
- Fuzzy CSPs: $\langle[0,1],max,min,0,1\rangle$:
  min idempotent
- Probabilistic CSPs: $\langle[0,1],max,\times,0,1\rangle$:
  $\times$ not idempotent
- Weighted CSPs: $\langle\mathcal{R}^+,min,+,+\infty,0\rangle$:
  $+$ not idempotent

SAC on fuzzy CSPs

- Fuzzy CSPs: $\langle [0,1], max, min, 0, 1 \rangle$:



- Combination via min. Example: $\langle a, a \rangle$ gets 0.8
- This fuzzy CSP is not SAC:
  - c_x gives 0.9 to $x = a$
  - $(\otimes\{c_y, c_{xy}, c_x\}) \Downarrow_x$ gives it 0.8:
    - combination: $val(\langle a, a \rangle) = min(0.9, 0.8, 0.9) = 0.8$ and $val(\langle a, b \rangle) = min(0.9, 0.2, 0.5) = 0.2$
    - projection: $max(0.8, 0.2) = 0.8$

Global and local preferences

- Local preferences: over constraints or tuples
- Global preferences: over complete assignments
- Usually knowledge involves both local and global preferences
- Easy to give, or to check, some solution ratings, but difficult to assign values to all tuples
- Soft constraint systems can usually handle only local preferences
- Learning techniques to induce (or refine) local preferences from global ones
- For example: learning algorithm based on gradient descent

Biso, Rossi, Sperduti, JETAI 1998 and KR 2000 (learning system)

Hard CSPs plus objective function?

- Same expressive power:
  - given a soft CSP, one can get an equivalent hard CSP with a suitable objective function, and viceversa
- But: soft constraint propagation on soft CSPs may generate tighter bounds to be used during the search