Soft constraints: models

T. Schiex

INRA - Toulouse, France

Outline ?

- Day 1: soft constraints models
- Day 2: branch and bound algorithms
- Day 3: inference
- Day 4: local inference and B&B
- Day 5: polynomial classes, applications

Why soft constraints ?

Constraint satisfaction problems usually allow to represent many decision problems:

- identify decision variables with their domains
- list all desirable properties (constraints)
- find a solution (satisfies all constraints)

Eg: job shop scheduling.

Job Shop scheduling

- a set of tasks $T = \{t_1, \ldots, t_n\}$, task t_i has duration d_i
- task t_i may use some ressource R_i (machine. . .)
- ressources cannot be shared
- some tasks needs to be done before others ($t_i \longrightarrow t_j$)
- must have the raw materials delivered
- must deliver the finished product in time

CSP model

Variables: a starting time s_i for task t_i
Constraints:

- precedence $t_i \rightarrow t_j$: $s_j \geq s_i + d_i$
- raw materials: some $s_i \geq raw_i$
- delivery time: some $s_i \leq del$
- resources:

$$R_i = R_j \Rightarrow (s_j \geq s_i + d_i) \vee (s_i \geq s_j + d_j)$$

Finding a feasible schedule is a NP-hard problem.

What if no feasible schedule exists ?

For most real problems, constraints may represent:

- physical laws: time, space, capacity. . .
- desired properties: preferences. . .
- uncertain laws: not sure it will apply in practice

Precedence, raw materials, resources: hard constraints

Delivery time: preference.

Machine failure: uncertain.

Time tabling

- number of rooms, courses
- size of audience/size of room
- available time slots
- one teacher can only give one course at a time
- Soft: precedences between courses
- Soft: different days for different lectures
- Soft: teacher's preferences over days/times

Why soft constraints are needed

- stating everything as a constraint may lead to unfeasible (inconsistent) problems
- stating only physical laws ignoring preferences, uncertainties may lead either to:
  - poor decisions
  - likely inapplicable decisions

Needs to distinguish between these in the modeling step.

Soft constraints: a natural way to locally express a complex criteria.

Notations

- A k-tuple: a sequence of k objects $(v_1, \ldots, v_k)$
- The i^{th} component of a tuple t is denoted as $t[i]$.
- The cartesian product of sets A_1,...,A_k ($A_1 \times \cdots \times A_k$ or $\prod_{i=1}^{k} A_i$) is the set of all the k-tuples $(v_1, \ldots, v_k)$ such that $v_1 \in A_1$,...,$v_k \in A_k$.

Notations

A *v*ariable represents an unknown element of its *d*omain, a finite set of *v*alues.

- given a sequence of variables $S = (x_1, \ldots, x_k)$ and their domains $D_1, \ldots, D_k$, a relation R on S is a subset of $D_1 \times \cdots \times D_k$ (scope S, arity $|S|$).
- Scope emphasis: $t_S \in R_S =$ assignment of S.
- $S' \subseteq S, t_S[S'] =$ projection of t_S on S'.

Classical CSP

A constraint network (X, D, C):

- a set of variables $X = \{x_1, \ldots, x_n\}$
- a set of domains $D = \{D_1, \ldots, D_n\}$
- a set of e constraints C.

A constraint $c \in C$ is a relation on a sequence of variables S, denoted c_S. $|S|$ is the arity of c_S.

$c_S \subset \prod_{x_j \in S} D_j$ specifies the *allowed* assignments for the variables of S.

Fuzzy CN

Relies on the notion of fuzzy sets.

Given a set E, a fuzzy set f on E is defined by a membership degree function μ_f:

$$\mu_f : E \longrightarrow [0, 1]$$

- $\mu_f(x) = 1$ means x belongs to f
- $\mu_f(x) = 0$ means x does not belong to f

Intermediate values allow for intermediate degrees of membership.

Classical sets: only 0 and 1 are used.

from fuzzy sets to fuzzy relations

Fuzzy relation R on S: a fuzzy set of tuples on S. $\mu_R(t)$ is the membership degree of tuple t to R.

Given 2 fuzzy sets f and g, the fuzzy set $f \cap g$ has a membership degree function defined by:

$$\mu_{f \cap g}(x) = \min(\mu_f(x), \mu_g(x))$$

NB: conjunctive interpretation. Other exists ($\min \rightarrow$ mean. . .).

Join of 2 fuzzy relations R_S, $R'_{S'}$: a fuzzy rel. on $S \cup S'$. . .

$$\mu_{R_S \bowtie R'_{S'}}(t) = \min(\mu_{R_S}(t[S]), \mu_{R'_{S'}}(t[S']))$$

Fuzzy CSP

A fuzzy CN is a triple (X, D, C):

- X is the usual set of variables
- D is the usual set of domains (may be fuzzy sets).
- C is a set e of fuzzy constraints.

A fuzzy constraint $c_S \in C$ is a fuzzy relation on S. It assigns a degree of membership to each tuple on S (degree of satisfaction of the constraint).

Semantics of a fuzzy network: $\bowtie_{c \in C} c$. Is a fuzzy set of solutions.

Fuzzy dinner: drink and meal

- fish or meat: f 0.8, m 0.3
- water, Barolo or Greco di Tufo w 0.7, b 1.0, g 0.9

	w	b	g
f	0.6	0.7	1.0
m	0.6	1.0	0.5

Fuzzy set Sol of solutions: $\mu_{Sol}(t) = \min_{cs \in C}(\mu_{cs}(t[S]))$.

Fuzzy dinner, continued

$\mu_S((m,w)) = \min(0.3, 0.7, 0.6) = 0.3$
$\mu_S((m,b)) = \min(0.3, 1.0, 1.0) = 0.3$
$\mu_S((f,b)) = \min(0.8, 1.0, 0.7) = 0.7$
$\mu_S((f,g)) = \min(0.8, 1.0, 1.0) = 0.8$

What if no fish ? The infamous drowning effect.

Typical problem: find a complete assignment t that maximizes $\mu_{Sol}(t)$, i.e.

$$\max_t(\min_{c_S \in C}(\mu_{c_S}(t[S])))$$

Max-min problem. Shift from satisfaction to optimization.

Possibilistic CSP

We start from a classical CSP.
Each constraint is assigned a priority between 0 and 1.

We want to minimize the priority of the most violated constraint.

Two differences with fuzzy CSP:

- Weights are associated with constraint, not tuples
- Min-max-optimization problem (dual to the max-min)

Ex: translate the previous fuzzy pb. to possibilistic CSP.

Implied constraints

Given a classical CSP (X, D, C), an implied constraint is a constraint wich is satisfied by all solutions of the problem.

It can be added to the CSP without changing the solution set.

Similar notion in fuzzy/possibilistic CSP.

Eg.: all solutions with w have a membership degree < 0.6. A unary constraint that lowers the membership degree of w to 0.6 can be added. Fuzzy set of solutions unchanged.

Improving discrimination

Lexicographic CSPs:

- we keep fuzzy CSP
- evaluation of a complete assignment: the sorted vector of membership degrees of all assigned constraints.
- goal: to find a complete assignment with maximum evaluation (lexicographic ordering).

Ex: compare solutions of the fuzzy dinner.
Ex: the optimum lex solutions are optimum fuzzy solutions.
Ex: implied constraints ?

Modeling additive costs

Weighted CSPs:

- each constraint/tuple has a violation cost (default 1: MaxCSP)
- evaluation of a complete assignment: the sum of all costs of all violated constraints.
- goal: to find a complete assignment with minimum cost.

Ex: consider the fuzzy dinner as a weighted CSP
Ex: transform a lex. CSP in a weighted CSP and vice-versa
Ex: Implied constraints ?

Modelling uncertainty

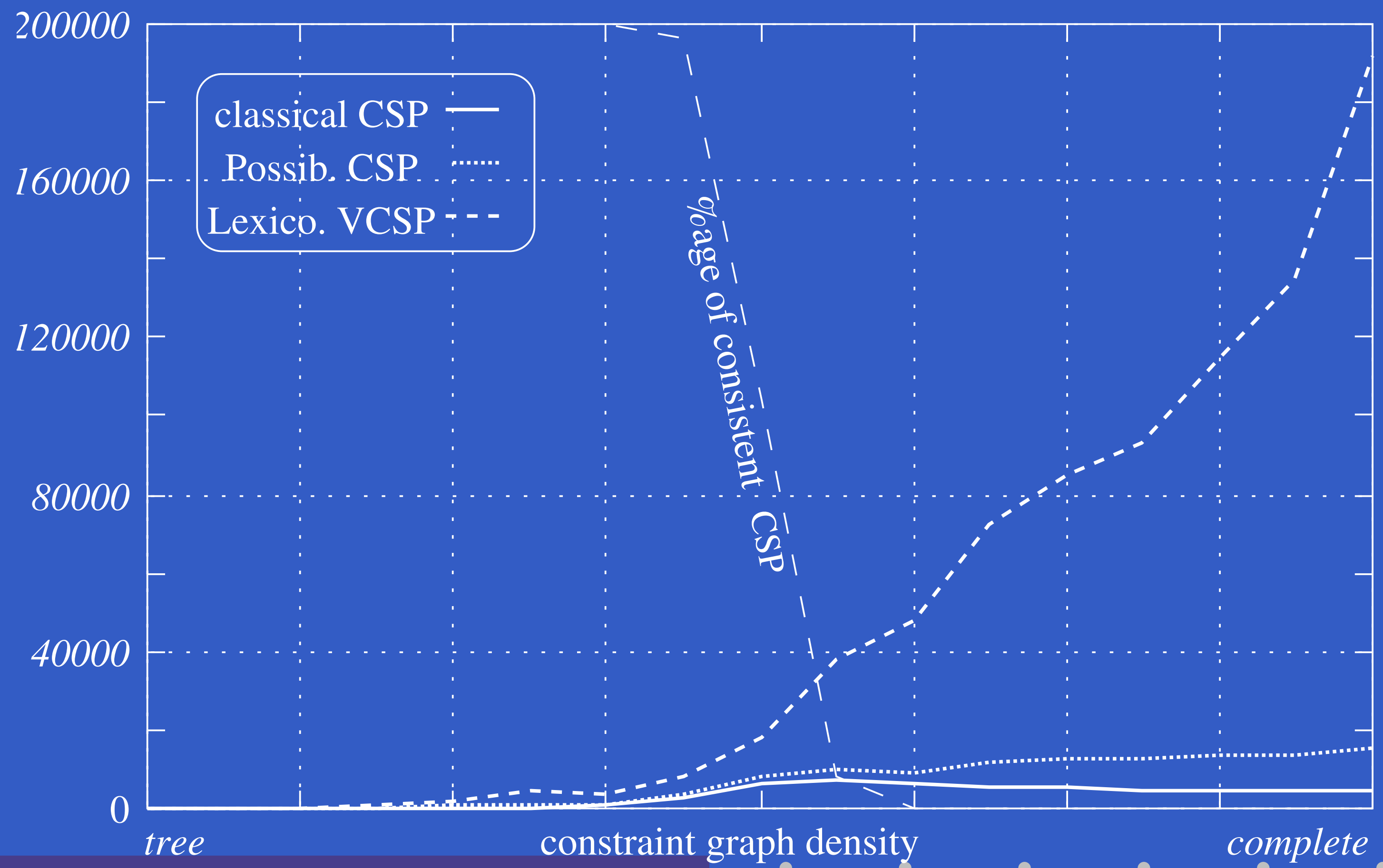
Probabilistic CSPs:

- each constraint c has a certain (independent) probability $p(c)$ to be part of the real problem (eg. failure probability).
- evaluation of a complete solution: probability that it will be a solution of the real problem
- goal: find a maximum probability assignment.

Ex: what is the evaluation ?
Ex: transform to weighted CSP
Ex: Implied constraints ?

A puzzling situation (1994)

- many frameworks, similar dedicated algorithms.
- some frameworks much harder to solve than others



Generic soft CSP models

Motivations: design a generic model that covers all existing proposals:

- to avoid repeated algorithmic work
- to understand what makes some problem harder than others

Generality: maximizes the number of frameworks covered

Specificity: stronger properties means more theorems, properties, algorithms.

Expected results

- to have a unique representation framework with efficient algorithms;
- to take into account in the same framework both hard constraints and soft constraints;
- to represent in the same settings consistent problems, with preferences on the acceptable solutions and inconsistent problems, with preferences on the way they should be relaxed.

Soft constraints vs. optimization: emphasis on a combination of local criteria. Overconstrained (unfeasible) problems.

HCSP (Borning et al. 1989)

- a strength level for each constraint (ordered: required, strong, medium, weak. . .)
- an error function for each constraint.
- find a solution that satisfies all required and other as much as possible, successively in each level.
- some pre-defined comparators on complete assignments (locally-better, weighted-sum better. . .)

Move the endpoint of an horizontal line with a mouse.
required: horizontal line, inside the window
strong: the endpoint follows the mouse position.

Partial CSP (Freuder 1989)

- when a problem is overconstrained, this means a solution satisfies only some of them (a relaxed problem).
- define a metric between problems and solve the problem which is
  - closest to the original
  - consistent
- 3 queens: ignore diagonal attack (constraint relaxation), use a 4x3 chessboard (domain relaxation). . .

Very general (metrics properties). Not completely specified. Most results on Weighted CSP (Freuder Wallace 1992).

Valued CSP (1995)

- with each constraint/tuple: a valuation that reflects the violation cost : preference, weight, priority, probability of being violated. . .
- the valuation of an assignment is the combination of the valuations expressed by each constraint using a binary operator (extra axioms).
- assignments can be compared using a total order on valuations.
- the problem is to produce an assignment of minimum valuation.

Commutative totally ordered semigroup with a monotonic operator.

More precisely

$S = \langle E, \oplus, \preccurlyeq_v, \bot, \top \rangle$.

- E = set of valuations, made of numbers or symbols, used to assess local assignments;
- $\bot$ = minimum element of E, corresponds to completely consistent assignments;
- $\top$ = maximum element of E, used to annotate hard constraints, corresponds to totally inconsistent assignments;
- $\preccurlyeq_v$ = total order on E, used to compare two valuations;
- $\oplus$ = operator used to combine two valuations;

Valued CSP

A tuple $\langle X, D, C, S \rangle$

- $X = \{x_1, \ldots, x_n\}$ is a set of n variables.
- $D = \{D_1, \ldots, D_n\}$ is the collection of the domains of the variables in X.
- C is a set of constraints. A constraint c (c_S) is a function defined on a set of variables $S \subseteq X$ that maps tuples to valuations $c : \prod_{x_i \in S} Di \rightarrow E$.
- $S = \langle E, \oplus, \preccurlyeq_v, \bot, \top \rangle$ is a valuation structure.

The valuation of a complete assignment

$$val(t) = \bigoplus_{c_S \in C} c_S(t[S])$$

Required properties

- $\forall \alpha, \beta \in E, (\alpha \oplus \beta) = (\beta \oplus \alpha)$. (commutativity)
- $\forall \alpha, \beta, \gamma \in E, (\alpha \oplus (\beta \oplus \gamma)) = ((\alpha \oplus \beta) \oplus \gamma)$. (associativity)
- $\forall \alpha, \beta, \gamma \in E, (\alpha \preccurlyeq_v \beta) \Rightarrow ((\alpha \oplus \gamma) \preccurlyeq_v (\beta \oplus \gamma))$. (monotonicity)
- $\forall \alpha \in E, (\alpha \oplus \bot) = \alpha$. (neutral element)
- $\forall \alpha \in E, (\alpha \oplus \top) = \top$. (annihilator)

Ex: justify these axioms.

Semiring CSP (1995)

Both significant and insignificant differences with VCSP.

- Insignificant: express satisfaction degrees, not violation degrees. Historically used the structural variant 1.
- Significant: can consider partially ordered structures

c-semiring

- A set E of satisfaction degrees.
- An operator $+_s$ defines a partial order $\preccurlyeq_s$ on the set E: $a \preccurlyeq_s b$ iff $a +_s b = b$ (ACI).
- a maximum element $\mathbf{1}$ and a minimum element $\mathbf{0}$. Implies that $\mathbf{1}$ is an annihilator for $+_s$, $\mathbf{0}$ a neutral element.
- an AC operator $\times_s$ combines sat. degrees. $\mathbf{0}$ is an annihilator for $\times_s$.
- $(a \times_s c) +_s (b \times_s c) = (a +_s b) \times_s c$ (distributivity).

Abelian semiring + idempotency of $+_s$.

Semiring CSP

A semiring constraint network is a tuple $\langle X, D, C, S \rangle$ where :

- $X = \{x_1, \ldots, x_n\}$ is a set of variables.
- $D = \{D_1, \ldots, D_n\}$ is the collection of the associated domains.
- C is a set of constraints. A constraint $c \in C$ (c_S) is a function defined on a set of variables $S \subseteq X$ that maps tuples to semiring values $c_S : \prod_{x_i \in S} D_i \rightarrow E$.
- $S = \langle E, +_s, \times_s, \mathbf{0}, \mathbf{1} \rangle$ is a c-semiring.

Problem: finding one/all non dominated solutions.

Totally ordered SCSP and VCSP

From totally ordered c-semiring $S = \langle E, +_s, \times_s, \mathbf{0}, \mathbf{1} \rangle$ to valuation structure $S' = \langle E, \preccurlyeq_v, \times_s, \bot, \top \rangle$ where:

$$(b \preccurlyeq_v a) \Leftrightarrow (a +_s b = b), \top = \mathbf{0}, \bot = \mathbf{1}$$

From valuation structure $S = \langle E, \preccurlyeq_v, \oplus, \bot, \top \rangle$ to c-semiring $S = \langle E, +_s, \oplus, \mathbf{0}, \mathbf{1})$:

$$(a +_s b = b) \Leftrightarrow (b \preccurlyeq_v a), \mathbf{0} = \top, \mathbf{1} = \bot$$

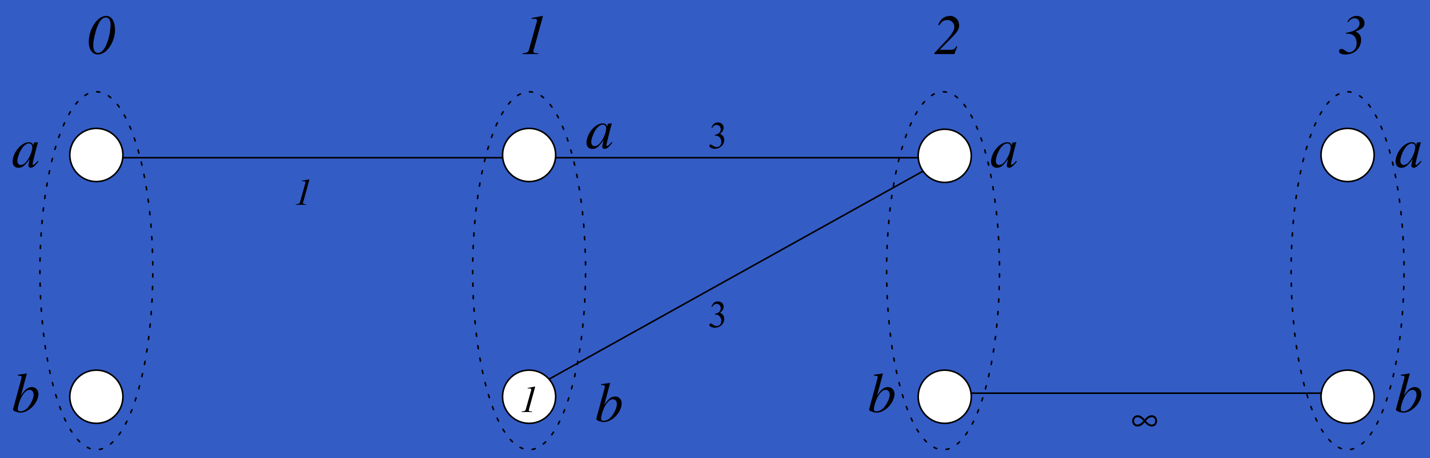
Ex: check axioms. . .

Describing a binary {V,S}CSP

By a variant of the so-called "microstructural" graph (multipartite graph):

- each value $a \in D_i$ is represented by a vertex (i, a).
- for $a \in D_i, b \in D_j$ s.t. $c_{ij} \in C$, an edge connects the vertex (i, a) and (j, b) with weight $c_{ij}(a, b)$ (if not equal to $\perp$). Weights equal to $\top$ omitted.
- unary constraints (if any) are represented as vertex labels.

Example - weighted MaxCSP



Valuation of $\langle a, a, b, b \rangle = 0 \oplus 1 \oplus 0 \oplus 3 \oplus \top = \top$.
Valuation of $\langle a, a, a, a \rangle = 0 \oplus 1 \oplus 0 \oplus 3 \oplus 0 = 4$.
Valuation of $\langle b, a, b, a \rangle = 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 = 0$.

Incompatible with classical CSP microstructure (edge = allowed). .

Instances

CSP	E	$\preccurlyeq_v$	$\top$	$\bot$	$\oplus$
*c*lassical	{t,f}	t $\preccurlyeq_v$ f	f	t	$\wedge$
*a*dditive	$\overline{N}$	$\leq$	$+\infty$	0	$+$
*f*uzzy	$[0,1]$	$\geq$	0	1	$\min$
*p*ossibilistic	$[0,1]$	$\leq$	1	0	$\max$
*l*exicographic	$[0,1]^*$	$\leq^*$	$\top$	$\emptyset$	$\cup$
*p*robabilistic	$[0,1]$	$\leq$	1	0	$1-(1-a)(1-b)$

Partially ordered SCSP

- Set-based SCSP.
  - semiring values are sets
  - $\times_s = \cup$, $+_s = \cap$ (order = set inclusion)
  - c-semiring $\langle \mathcal{P}(A), \cup, \cap, \varnothing, A \rangle$. Distributive lattice.
- Multicriteria SCSP.
  - one $S_i = \langle E_i, +_{si}, \times_{si}, \mathbf{0}_i, \mathbf{1}_i \rangle$ per criteria
  - $\langle E_1, \ldots, E_k \rangle, +_s, \times_s, \langle \mathbf{0}_1, \ldots, \mathbf{0}_n \rangle, \langle \mathbf{1}_1, \ldots, \mathbf{1}_n \rangle \rangle$
  - $\times_s, +_s$: pointwise application of each $\times_{si}, +_{si}$.

Structural variants

Depending on how we want to define cost functions, valuations can be associated with:

- 1: tuples: most general.
- 2: classical constraints: $c_i(t) = \alpha_i$ if $t \notin R_i$ ($\perp$ otherwise)
- 3: values: only unary soft constraints.
- 4: variables: cost of leaving unassigned (MUP).

Historically VCSP used 2.
Ex: Model 1 with 2, 4 with 3.

Desirable properties of $\oplus$

- avoiding the drowning effect: strict monotonicity.

$$\forall a, b, c \in E, a \succ b \wedge c \neq \top \Rightarrow (a \oplus c) \succ (b \oplus c)$$

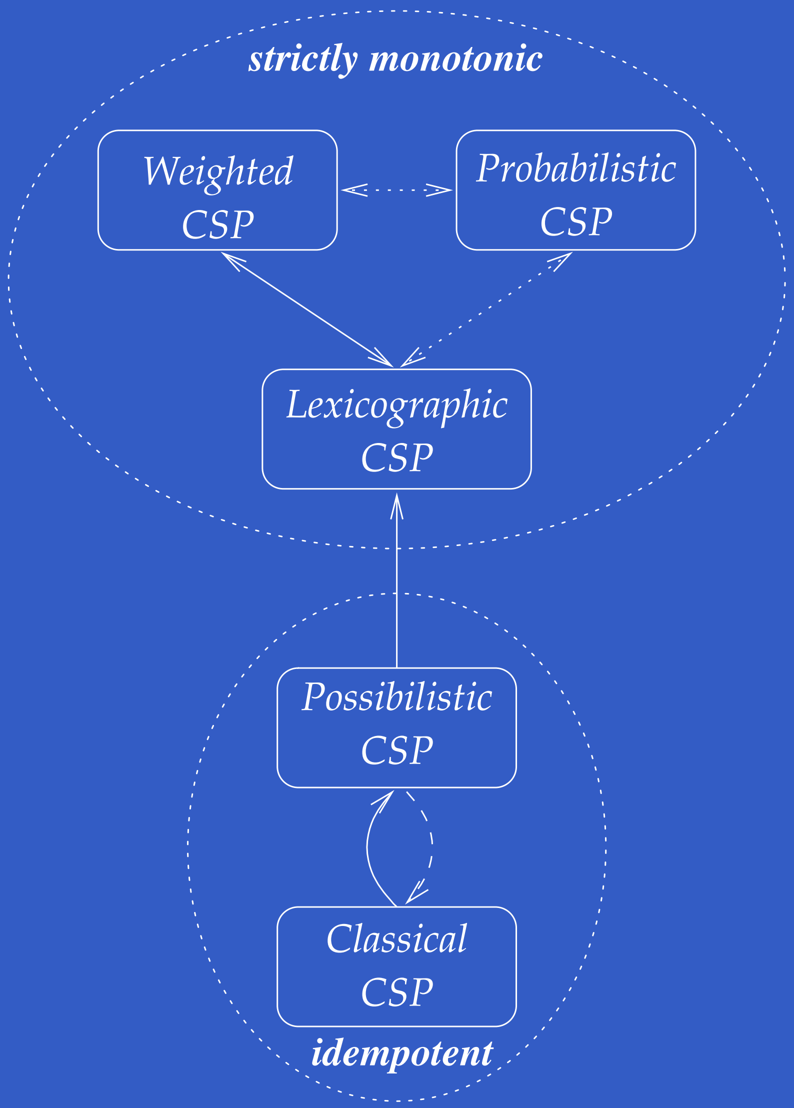
  Ex: show that if one valuation strictly improves, the assignment valuation improves strictly.

- can add implied constraints: idempotency.

$$\forall a \in E, a \oplus a = a$$

VCSP axioms + $|E| > 2$ make these 2 incompatibles.

Links with expressive power/complexity



Influence of arity on expressive prower

- Classical CSP: binary CSP can express all problems (dual problem).
- Soft CSP: binary hard constraints + soft unary are enough.

Ex: show how this is possible by transforming a VCSP into its dual (define this).