Marginal Consistency: Unifying Constraint Propagation on Commutative Semirings

Tomáš Werner

Center for Machine Perception
Czech Technical University
Prague, Czech Republic

- Max-sum diffusion [Koval-Kovalevsky-1976] is a simple algorithm to decrease upper bound on Weighted CSP due to [Schlesinger-1976].
- Originally formulated only for binary problems.
- Yields very good (sometimes exact) uppers bounds of WCSP instances.

We generalise max-sum diffusion in two ways:

- from binary networks to networks of any arity [Werner-2008]
- from Weighted CSP to Semiring CSP [Werner-2007]

This offers a unified view on crisp and soft constraint propagation.

Notation

V	(finite) set of variables, totally ordered
$v \in V$	a single variable
X_v	(finite) domain of variable $v \in V$
$x_v \in X_v$	state of variable $v \in V$

$A \subseteq V$	a subset of variables
$X_A = \prod_{v \in A} X_v$	joint domain of variables $A \subseteq V$ (ordered by the order on V)
$x_A \in X_A$	joint state of variables $A \subseteq V$

Convention: "Implicit restriction"

For $B \subset A$, if symbols x_A and x_B appear in the same logical expression, x_B denotes the restriction of joint state x_A onto variables B.

S	set of weights
$f_A \colon X_A \to S$	constraint with scope $A \subseteq V$

2^V	the set of all subsets of V
$\binom{V}{k}$	the set of all k-element subsets of V

Semiring-based CSP

Definition (Constraint network)

Let $E \subseteq 2^V$ be a hypergraph. Let each hyperedge $A \in E$ be assigned a constraint $f_A \colon X_A \to S$. This collection of constraints is called a constraint network.

Denoting $T(E) = \{ (A, x_A) \mid A \in E,\, x_A \in X_A \}$, a constraint network is a mapping

$$\begin{aligned} f : \quad T(E) &\to S \\ (A, x_A) &\mapsto f_A(x_A) \end{aligned}$$

Definition (Semiring-based CSP)

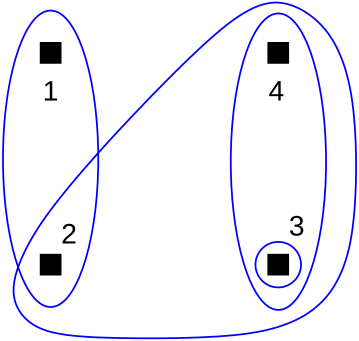
Given a commutative semiring $(S, \oplus, \odot)$ and a constraint network f, calculate expression

$$\bigoplus_{x_V \in X_V} \bigodot_{A \in E} f_A(x_A)$$

Example: A ternary problem

Let $V = (1, 2, 3, 4)$ and $E = \{(2, 3, 4), (1, 2), (3, 4), (3)\}$. Then

$$\bigoplus_{x_V} \bigodot_{A \in E} f_A(x_A) = \bigoplus_{x_1, x_2, x_3, x_4} [f_{234}(x_2, x_3, x_4) \odot f_{12}(x_1, x_2) \odot f_{34}(x_3, x_4) \odot f_3(x_3)]$$



Example: A binary Weighted CSP

Let $E = \binom{V}{1} \cup E'$ where $E' \subseteq \binom{V}{2}$. Let $(S, \oplus, \odot) = (\mathbb{R} \cup \{-\infty\}, \max, +)$.

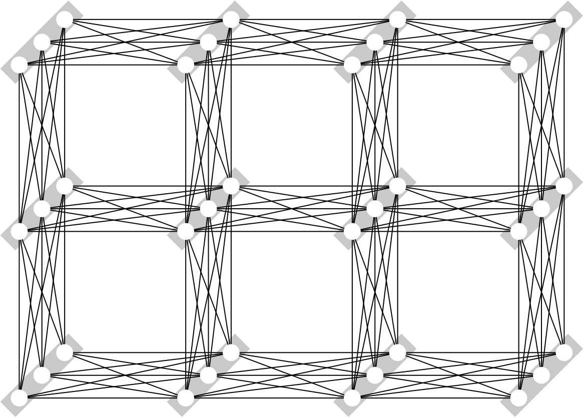
$$\bigoplus_{x_V} \bigodot_{A \in E} f_A(x_A) = \max_{x_V} \sum_{A \in E} f_A(x_A) = \max_{x_V} \left[\sum_{v \in V} f_v(x_v) + \sum_{vv' \in E'} f_{vv'}(x_v, x_{v'}) \right]$$

Example: A binary Weighted CSP

Let $E = \binom{V}{1} \cup E'$ where $E' \subseteq \binom{V}{2}$. Let $(S, \oplus, \odot) = (\mathbb{R} \cup \{-\infty\}, \max, +)$.

$$\bigoplus_{x_V} \bigodot_{A \in E} f_A(x_A) = \max_{x_V} \sum_{A \in E} f_A(x_A) = \max_{x_V} \left[\sum_{v \in V} f_v(x_v) + \sum_{vv' \in E'} f_{vv'}(x_v, x_{v'}) \right]$$

Microstructure for E a grid graph and $X_v = \{1, 2, 3\}$:

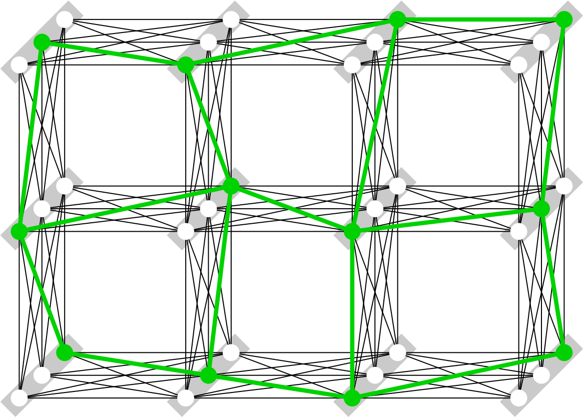


Example: A binary Weighted CSP

Let $E = \binom{V}{1} \cup E'$ where $E' \subseteq \binom{V}{2}$. Let $(S, \oplus, \odot) = (\mathbb{R} \cup \{-\infty\}, \max, +)$.

$$\bigoplus_{x_V} \bigodot_{A \in E} f_A(x_A) = \max_{x_V} \sum_{A \in E} f_A(x_A) = \max_{x_V} \left[\sum_{v \in V} f_v(x_v) + \sum_{vv' \in E'} f_{vv'}(x_v, x_{v'}) \right]$$

Microstructure for E a grid graph and $X_v = \{1, 2, 3\}$:



Equivalent transformations

Definition

Constraint networks f and f' are equivalent iff they have the same variables V, domains X_v and structure E, and

$$\forall x_V \in X_V : \quad \bigodot_{A \in E} f_A(x_A) = \bigodot_{A \in E} f'_A(x_A)$$

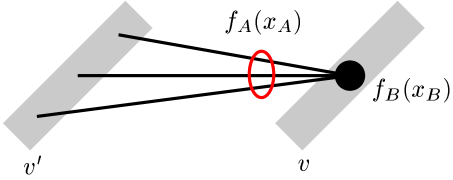
A change of f to an equivalent network is an equivalent transformation.

Definition

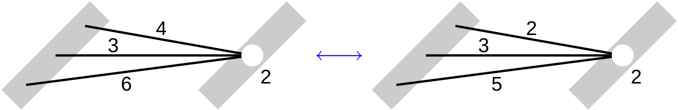
An equivalent transformation is local iff it changes not more than two constraints, f_A and f_B, and it does it such that $f_A(x_A) \odot f_B(x_B)$ is preserved for all $x_{A \cup B}$.

Note: Equivalent transformations depend only on semigroup $(S, \odot)$ and not on $\oplus$.

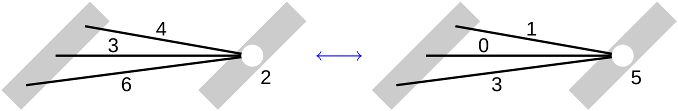
Let $A = (v, v')$ and $B = (v)$:



$(S, \odot) = (\mathbb{R}, \min)$:



$(S, \odot) = (\mathbb{R}, +)$:

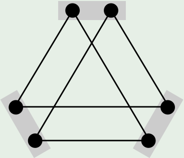


Covering equivalent transformations by local ones

An equivalent transformation may or may not be possible to compose of a sequence of local equivalent transformations.

Example

Let $(S, \otimes) = (\{0, 1\}, \min)$. Let f represent an unsatisfiable crisp CSP.



Then f is equivalent to the zero network $f \equiv 0$ but the two networks cannot be transformed to each other by local equivalent transformations.

Example

Let $(S, \odot)$ be a group, i.e., we have division. Then every equivalent transformation can be composed of local ones.

Marginals

Definition

Given a function $f_A\colon X_A \to S$ and a set $B \subseteq A$, we define function $f_A|_B\colon X_B \to S$ by

$$f_A|_B(x_B) = \bigoplus_{x_{A \setminus B}} f_A(x_A)$$

We call $f_A|_B(x_B)$ the marginal of f_A associated with joint state x_B of variables B.

Example

Let $A = (1, 2, 3, 4)$ and $B = (1, 3)$. The marginal of a function f_A associated with joint state x_B of variables B is given by

$$f_{1234}|_{13}(x_1, x_3) = \bigoplus_{x_2, x_4} f_{1234}(x_1, x_2, x_3, x_4)$$

Marginal consistency

Definition

A pair of constraints (f_A, f_B) is marginal consistent iff $f_A|_{A \cap B} \equiv f_B|_{A \cap B}$.

Definition

A constraint network f is marginal consistent iff for every $A \in E$ and $B \in E$, constraint pair (f_A, f_B) is marginal consistent.

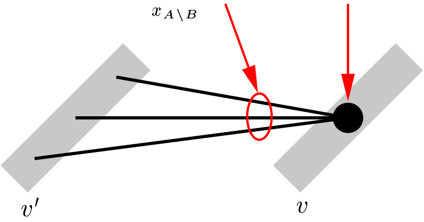
Example

A network f with structure $E = \{(1), (1,2), (2,3)\}$ is marginal consistent iff $f_1 \equiv f_{12}|_1$, $f_1|_\emptyset \equiv f_{23}|_\emptyset$, and $f_{12}|_2 \equiv f_{23}|_2$.

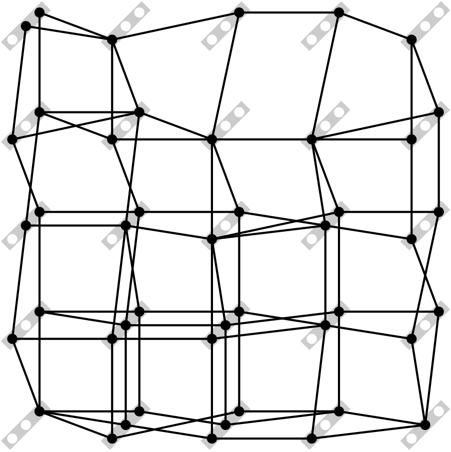
Note: Marginal consistency depends only on $(S, \oplus)$ and not on $\odot$.

Let $A = (v, v')$ and $B = (v)$:

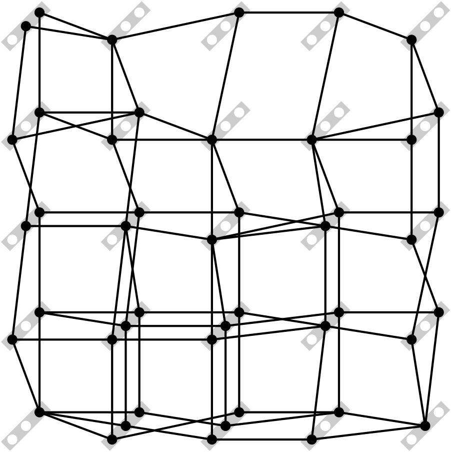
$$\bigoplus_{x_{A \setminus B}} f_A(x_A) = f_B(x_B)$$



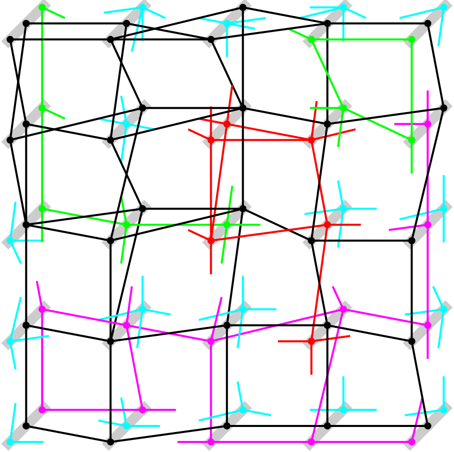
$(S, \oplus) = (\{0, 1\}, \max)$



$(S, \oplus) = (\{0, 1\}, \max)$



$(S, \oplus) = (\mathbb{R}, \max)$



Definition

Enforcing marginal consistency of a constraint pair (f_A, f_B) is a local equivalent transformation of the pair that makes the pair marginal consistent.

That means, replace the pair (f_A, f_B) with a new pair (f'_A, f'_B) satisfying the system

$$\begin{aligned} f'_A(x_A) \odot f'_B(x_B) &= f_A(x_A) \odot f_B(x_B) & \forall x_{A\cup B} \\ f'_A|_{A\cap B}(x_{A\cap B}) &= f'_B|_{A\cap B}(x_{A\cap B}) & \forall x_{A\cap B} \end{aligned}$$

The system is...

- uniquely solvable in semirings $(\mathbb{R}_+, +, \times)$, $(\mathbb{R} \cup \{-\infty\}, \max, +)$, $(\mathbb{R}_+, \min, +)$, a distributive lattice $(S, \vee, \wedge)$ (e.g., $(\{0,1\}, \max, \min)$ and $([0,1], \max, \min)$)
- solvable but not uniquely in semiring $([-1,0], \max, \lfloor + \rfloor)$ where $\lfloor + \rfloor$ is the truncated addition defined by $a \lfloor + \rfloor b = \max\{-1, a+b\}$
- not solvable in semirings $(\mathbb{N}, \max, +)$, $(\mathbb{R}, +, \times)$, $(\mathbb{Q}_+, +, \times)$

Enforcing marginal consistency of a network

Observation

Let the semiring $(S, \oplus, \odot)$ be such that enforcing marginal consistency of a constraint pair is possible and unique. Enforcing marginal consistency repetitively for different constraint pairs converges to a state when the whole network is marginal consistent. The pairs can be visited in any order such that each has a non-zero probability to be visited.

Currently, we have neither a proof of the observation nor a counter-example.

Marginal consistency algorithm

repeat
  for $(A, B) \in E \times E$ **do**
    Enforce marginal consistency of constraint pair (f_A, f_B).
  end for
until convergence

Fundamental property of soft constraint networks

By locally changing constraints, any constraint network can be transformed to an equivalent form in which corresponding marginals of each constraint pair coincide.

Upper bound

Definition (Green's preorder)

Let relation $\leq$ be defined on semigroup $(S, \oplus)$ by

$$a \leq b \iff (a = b) \text{ or } (\exists c \in S\colon\ a \oplus c = b)$$

Relation $\leq$ is reflexive and transitive, hence a preorder. Often, it is also antisymmetric, hence a (partial or total) order.

Theorem (Upper bounds on Semiring CSP)

For a constraint network f, we have

$$\bigoplus_{x_V} \bigodot_{A \in E} f_A(x_A) \leq \bigodot_{A \in E} \bigoplus_{x_A} f_A(x_A) \leq \left[\left[\bigoplus_{A \in E} \bigoplus_{x_A} f_A(x_A)\right]^{\oplus 1/|E|}\right]^{\odot |E|}$$

If f is marginal consistent then the middle and right-hand expressions equal.

For proving this, we need semiring $(S, \oplus, \odot)$ to satisfy the arithmetic-geometric mean inequality

$$\bigodot_{i=1}^{n} a_i \leq \left[\left(\bigoplus_{i=1}^{n} a_i\right)^{\oplus 1/n}\right]^{\odot n}$$

Enforcing marginal consistency does not worsen the bound

Theorem

Enforcing marginal consistency of any constraint pair does not increase the upper bound.

In fact, marginal consistency is neither sufficient not necessary for minimum of the upper bound in the equivalence class.

Theorem

If a network is marginal consistent then the upper bound cannot be improved by any single local equivalent transformation.

Upper bound for Valued CSP

Definition (Valued CSP [Schiex-1995])

If $\leq$ is a total order and $\oplus$ is idempotent, then, $\oplus$ is necessarily the maximum with respect to $\leq$. In that case, Semiring CSP on $(S, \oplus, \odot)$ is called Valued CSP.

Definition

Joint state x_A of hyperedge $A \in E$ is called active if $f_A(x_A) = \max_{y_A} f_A(y_A)$.

Theorem

The upper bound is tight iff the (crisp) constraint satisfaction problem (CSP) formed by the active joint states is satisfiable.

Upper bound for Valued CSP

Definition (Valued CSP [Schiex-1995])

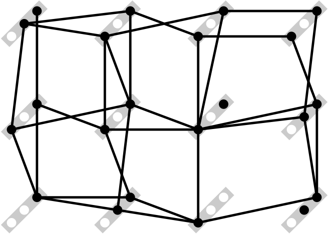
If $\leq$ is a total order and $\oplus$ is idempotent, then, $\oplus$ is necessarily the maximum with respect to $\leq$. In that case, Semiring CSP on $(S, \oplus, \odot)$ is called Valued CSP.

Definition

Joint state x_A of hyperedge $A \in E$ is called active if $f_A(x_A) = \max_{y_A} f_A(y_A)$.

Theorem

The upper bound is tight iff the (crisp) constraint satisfaction problem (CSP) formed by the active joint states is satisfiable.



Definition (Valued CSP [Schiex-1995])

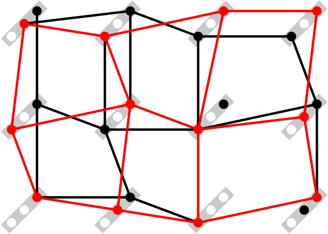
If $\leq$ is a total order and $\oplus$ is idempotent, then, $\oplus$ is necessarily the maximum with respect to $\leq$. In that case, Semiring CSP on $(S, \oplus, \odot)$ is called Valued CSP.

Definition

Joint state x_A of hyperedge $A \in E$ is called active if $f_A(x_A) = \max_{y_A} f_A(y_A)$.

Theorem

The upper bound is tight iff the (crisp) constraint satisfaction problem (CSP) formed by the active joint states is satisfiable.



Enforcing marginal consistency for important semirings

Next, we will discuss the marginal consistency algorithm for generic semirings:

- Distributive lattice, including $([0,1], \max, \min)$ and $(\{0,1\}, \max, \min)$
- Max-sum semirings $(\mathbb{R}, \max, +)$ and $(\mathbb{R} \cup \{-\infty\}, \max, +)$
- Sum-product semiring $(\mathbb{R}_{++}, +, \times)$ and $(\mathbb{R}_{+}, +, \times)$

We will address the following questions:

- Does the marginal consistency algorithm converge in a finite or infinite number of iterations?
- Is the marginal consistency closure unique?
- Does the marginal consistency algorithm finds the global minimum of the upper bound in the class of equivalent networks?
- Is the upper bound evaluated at marginal consistency practically useful?

Distributive lattice $(S, \vee, \wedge)$

Distributive lattice $(S, \vee, \wedge)$ is given by:

- $(S, \leq)$ is a partially ordered set
- $\vee$ is the supremum (join) induced by $\leq$
- $\wedge$ is the infimum (meet) induced by $\leq$
- $\wedge$ distributes over $\vee$

Properties:

- Marginal consistency algorithm converges in a finite number of iterations.
- The marginal consistency closure is unique.
- The closure does not minimise the upper bound in the equivalence class. In fact, minimising the upper bound is not tractable.
- Closures provide useful upper bounds.

Special cases:

- Fuzzy CSP semiring $([0, 1], \max, \min)$
- Crisp CSP semiring $(\{0, 1\}, \max, \min)$

Crisp CSP semiring $(\{0,1\}, \max, \min)$

For crisp CSP, marginal consistency is one of well-known local consistencies:

Crisp CSP semiring $(\{0, 1\}, \max, \min)$

For crisp CSP, marginal consistency is one of well-known local consistencies:

- For general structure $E \subseteq 2^V$, making the network marginal consistent yields pairwise consistency.

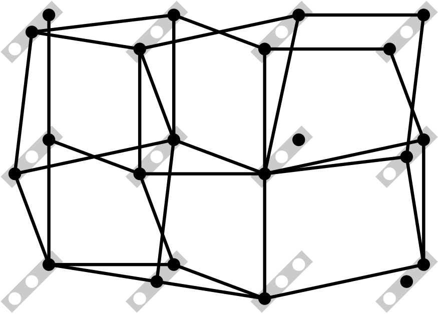
Crisp CSP semiring $(\{0, 1\}, \max, \min)$

For crisp CSP, marginal consistency is one of well-known local consistencies:

- For general structure $E \subseteq 2^V$, making the network marginal consistent yields pairwise consistency.
- For a binary structure $E = \binom{V}{1} \cup E'$ with $E' \subseteq \binom{V}{2}$, making the network marginal consistent yields arc consistency.

For crisp CSP, marginal consistency is one of well-known local consistencies:

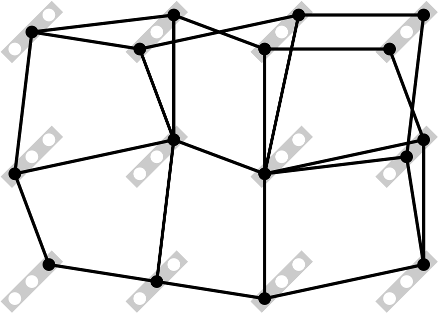
- For general structure $E \subseteq 2^V$, making the network marginal consistent yields pairwise consistency.
- For a binary structure $E = \binom{V}{1} \cup E'$ with $E' \subseteq \binom{V}{2}$, making the network marginal consistent yields arc consistency.



Crisp CSP semiring $(\{0,1\}, \max, \min)$

For crisp CSP, marginal consistency is one of well-known local consistencies:

- For general structure $E \subseteq 2^V$, making the network marginal consistent yields pairwise consistency.
- For a binary structure $E = \binom{V}{1} \cup E'$ with $E' \subseteq \binom{V}{2}$, making the network marginal consistent yields arc consistency.



Crisp CSP semiring $(\{0,1\}, \max, \min)$

For crisp CSP, marginal consistency is one of well-known local consistencies:

- For general structure $E \subseteq 2^V$, making the network marginal consistent yields pairwise consistency.
- For a binary structure $E = \binom{V}{1} \cup E'$ with $E' \subseteq \binom{V}{2}$, making the network marginal consistent yields arc consistency.
- For $E \subseteq 2^V$ with $\binom{V}{1} \subseteq E$, making pairs $\{\, (A, (v)) \mid A \in E,\ v \in A \,\}$ marginal consistent yields generalised arc consistency.

Max-sum semirings $(\mathbb{R}, \max, +)$ and $(\mathbb{R} \cup \{-\infty\}, \max, +)$

- The algorithm converges in an infinite number of iterations. (But convergence in parameter only conjectured.)
- Marginal consistency closure is not unique.
- In semiring $(\mathbb{R}, \max, +)$, minimising the upper bound in the equivalence class is polynomial: it leads to an unconstrained minimisation of a convex piecewise-linear function, i.e., a linear program.

  In general, the marginal consistency algorithm does not find the global minimum of this linear program. It can converge to a point where the bound can be improved by changing no single variable separately but only several variables simultaneously.
- In semiring $(\mathbb{R} \cup \{-\infty\}, \max, +)$, minimising the upper bound is no longer polynomial. But up to the crisp component, it leads to the same linear program.
- Marginal consistency closures provide very good upper bounds, often tight for non-trivial instances (especially sparse).
- If constraints f_A are supermodular for all $A \in E$, the closure solves the problem exactly.

Sum-product semirings $(\mathbb{R}_{++}, +, \times)$ and $(\mathbb{R}_{+}, +, \times)$

- The algorithm converges in an infinite number of iterations.
- Marginal consistency closure is unique.
- In semiring $(\mathbb{R}_{++}, +, \times)$, minimising the upper bound in the equivalence class leads to an unconstrained minimisation of a smooth convex function, in fact a geometric program. This function attains its global minimum at the marginal consistency closure.
- In semiring $(\mathbb{R}_{+}, +, \times)$, minimising the upper bound has an additional crisp component.
- The least upper bound is loose, yielding poor approximations of the partition function.

Let $f\colon T(E) \to \{0,1\}$ represent a crisp CSP.

- Expression $\sum_{x_V} \prod_{A \in E} f_A(x_A)$ counts the number of solutions of CSP f.
- The sum-product upper bound is an upper bound on this number.
- If this upper bound is less than 1 then CSP f is not satisfiable.

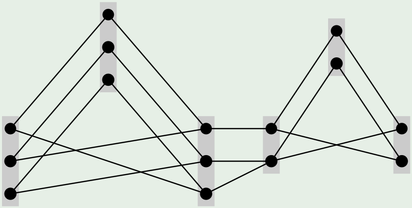
Compare two necessary conditions for satisfiability of CSP f:

1. $(\max, \times)$ marginal consistency closure of f must be non-zero.
2. $(+, \times)$ marginal consistency closure of f must be non-zero.

Condition 2 is strictly stronger than condition 1!

Example

This CSP has non-zero $(\max, \times)$ closure but zero $(+, \times)$ closure:



Adding neutral constraints

Let $\mathbf{1}$ denote the identity element of semiring $(S, \oplus, \odot)$, i.e., $a \odot \mathbf{1} = a$ for $a \in S$.

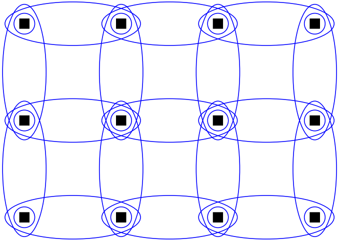
Definition (Neutral constraint)

A constraint f_A is neutral iff $f_A(x_A) = \mathbf{1}$ for all $x_A \in X_A$. In short, we write $f_A \equiv \mathbf{1}$.

Let us add a neutral constraint to a network. Then:

- Objective function $\bigodot_{A \in E} f_A(x_A)$ is preserved.
- If $\oplus$ is idempotent, then also upper bound $\bigodot_{A \in E} \bigoplus_{x_A} f_A(x_A)$ is preserved.
- Some previously impossible local equivalent transformations are allowed. Thus, the new upper bound may be possible to decrease even if the old upper bound was not.

- Randomly draw instances of a binary problem from an instance type.
- Count instances solved exactly by enforcing max-sum marginal consistency.
- Two relaxations tested:
  - Plain relaxation without zero constraints

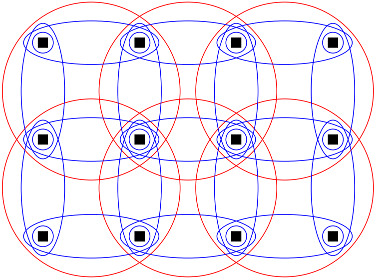


without zero constraints

type	image side	$\lvert X_v \rvert$	r_{plain}
random	15	5	0.01
random	25	3	0.00
random	100	3	0.00
Potts	15	5	0.79
Potts	25	5	0.48
Potts	100	5	0.00
lines	10	4	0.72
lines	25	4	0.00
curve	10	9	0.17
curve	15	9	0.00
curve	25	9	0.00
Pi	15	5	0.00

Example: Adding zero 4-cycle constraints to binary problems

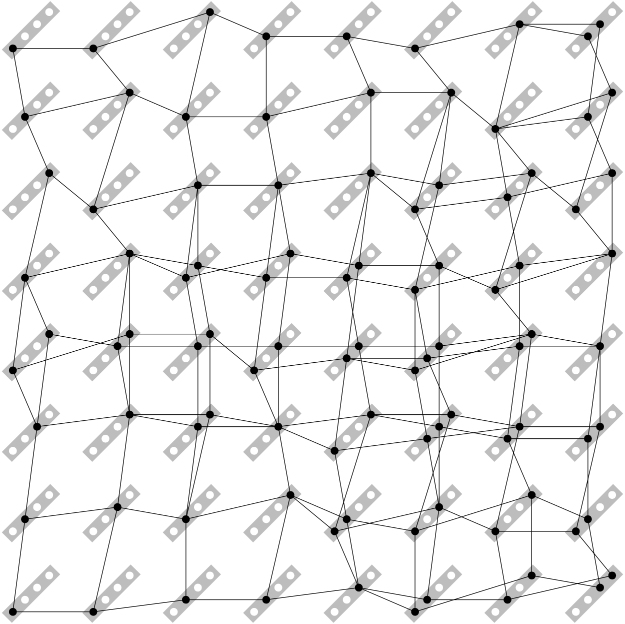
- Randomly draw instances of a binary problem from an instance type.
- Count instances solved exactly by enforcing max-sum marginal consistency.
- Two relaxations tested:
  - Plain relaxation without zero constraints
  - Relaxation augmented by 4-ary zero constraints on neighboring variables.



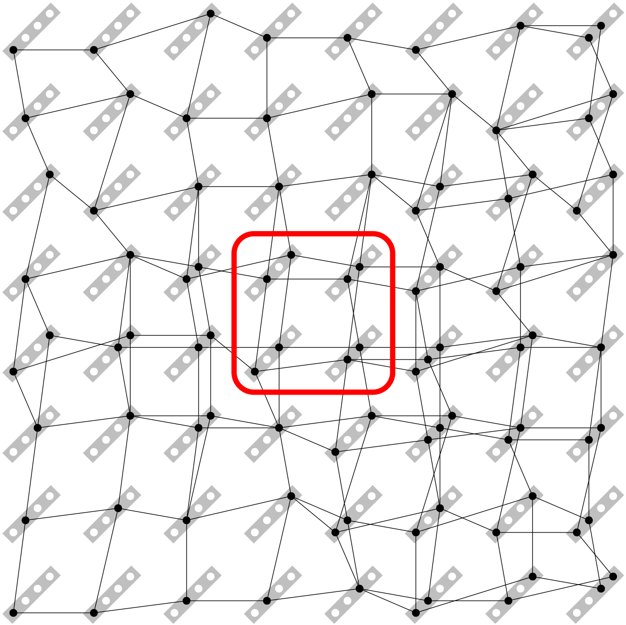
with zero constraints

type	image side	$\|X_v\|$	r_{plain}	r_{4cycle}
random	15	5	0.01	1.00
random	25	3	0.00	0.98
random	100	3	0.00	0.72
Potts	15	5	0.79	0.99
Potts	25	5	0.48	0.98
Potts	100	5	0.00	0.81
lines	10	4	0.72	0.88
lines	25	4	0.00	0.00
curve	10	9	0.17	0.65
curve	15	9	0.00	0.24
curve	25	9	0.00	0.00
Pi	15	5	0.00	0.82

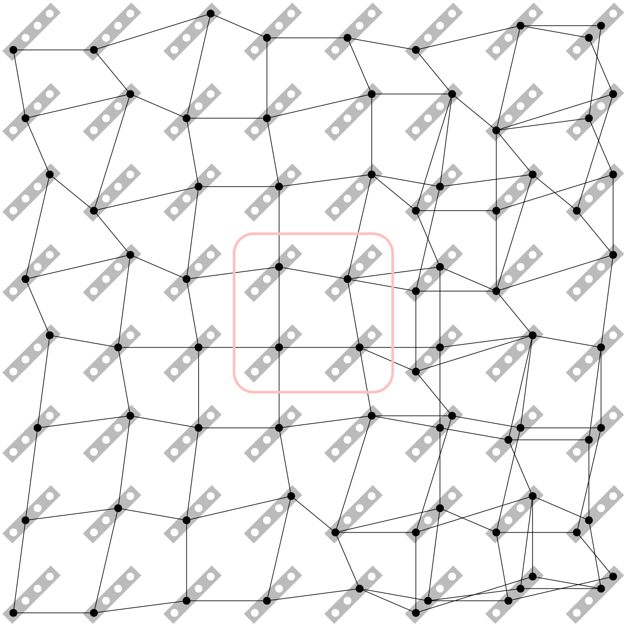
Adding neutral constraints dynamically



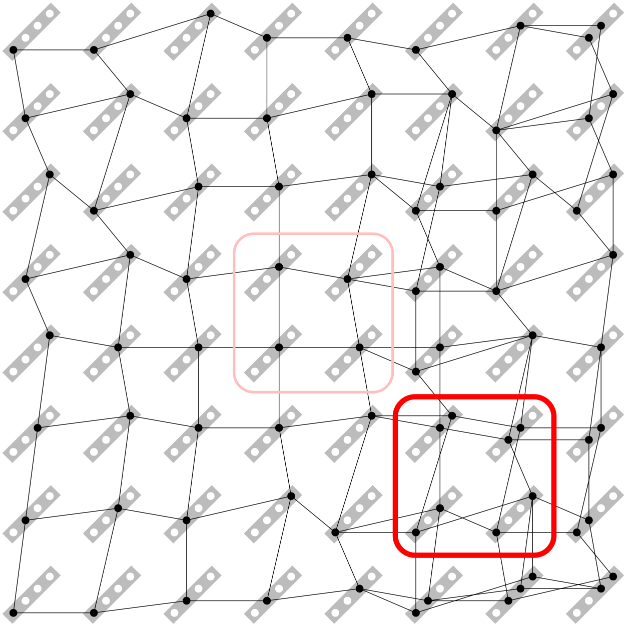
Adding neutral constraints dynamically



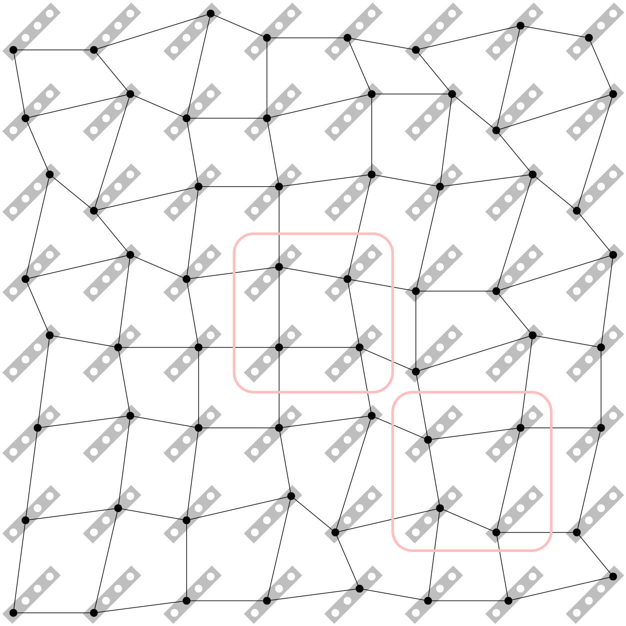
Adding neutral constraints dynamically



Adding neutral constraints dynamically



Adding neutral constraints dynamically



Suppose a constraint $f_A \equiv \mathbf{1}$ with $A \notin E$ has been added to a network and then marginal consistency has been enforced.

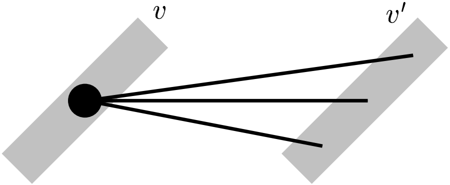
- In semiring $([0,1], \max, \min)$, we can now remove f_A from the network because this leaves the network equivalent with the original network. Thus, f_A can be in fact added only temporarily, which results in changing some constraints but does not increase the number of constraints in the network.
- In semiring $(\mathbb{R}, \max, +)$, we cannot remove f_A because this would yield an non-equivalent network. Thus, adding a neutral constraint and enforcing marginal consistency does increase the number of constraints in the network.

Example

Let us have a binary network, $E = \binom{V}{1} \cup E'$ where $E' \subseteq \binom{V}{2}$. Let $|X_v| = 2$. Add neutral constraints so that $\binom{V}{2} \cup \binom{V}{3} \subseteq E$. Enforce marginal consistency.

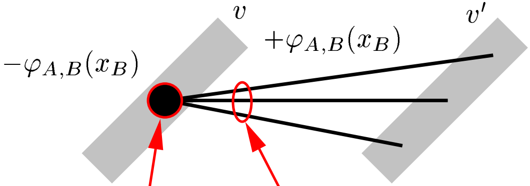
- For semiring $(\{0,1\}, \max, \min)$, we can remove ternary constraints, obtaining a path consistent binary CSP. Since the domains are Boolean, it is tractable.
- For semiring $(\mathbb{R}, \max, +)$, we cannot remove ternary constraints. The active joint states form a ternary 3-consistent CSP, which in general is not tractable.

Let $A = (v, v')$ and $B = (v)$:



Max-sum marginal consistency algorithm

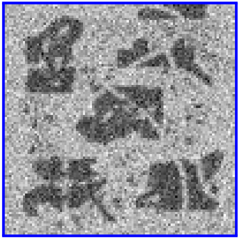
Let $A = (v, v')$ and $B = (v)$:

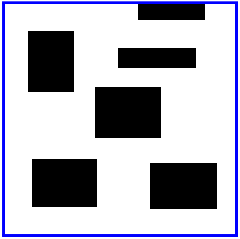


$$f_B(x_B) = \max_{x_{A \setminus B}} f_A(x_A)$$

Example WCSP: Syntactic image analysis

Find the image containing non-overlapping rectangles, nearest to input image!

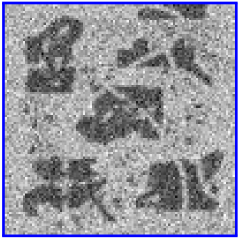

input image


output image

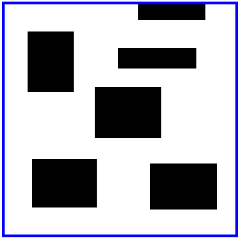
Example WCSP: Syntactic image analysis

Find the image containing non-overlapping rectangles, nearest to input image!

- Variables V are pixels, hypergraph E is the image grid.



input image

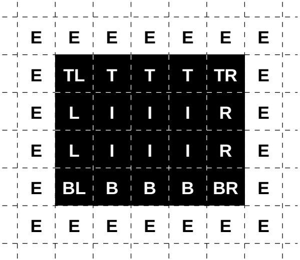


output image

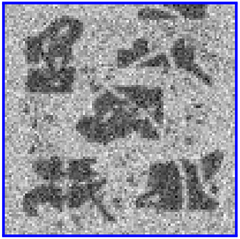
Example WCSP: Syntactic image analysis

Find the image containing non-overlapping rectangles, nearest to input image!

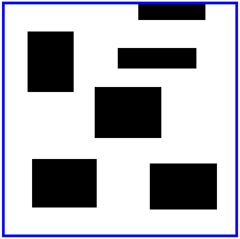
- Variables V are pixels, hypergraph E is the image grid.
- Variable domains $X_v = \{\,\mathrm{E, I, L, R, T, B, TL, TR, BL, BR}\,\}$ are syntactic parts of a rectangle.



hidden states = syntactic parts



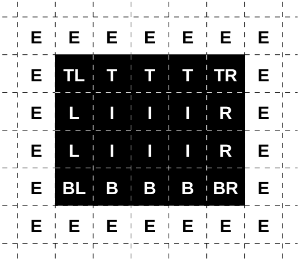
input image



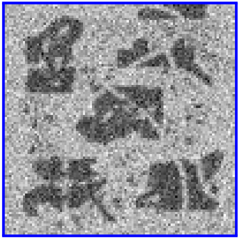
output image

Find the image containing non-overlapping rectangles, nearest to input image!

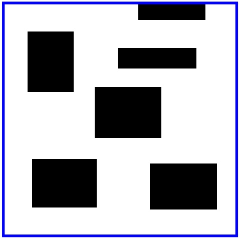
- Variables V are pixels, hypergraph E is the image grid.
- Variable domains $X_v = \{\mathrm{E, I, L, R, T, B, TL, TR, BL, BR}\}$ are syntactic parts of a rectangle.
- Unary constraint $f_v(x_v)$ quantifies agreement of intensity of state x_v and intensity of input pixel v.



hidden states = syntactic parts
observed states = {black,white}



input image



output image

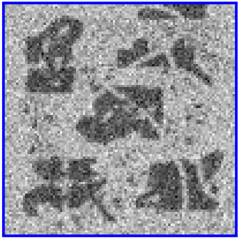
Example WCSP: Syntactic image analysis

Find the image containing non-overlapping rectangles, nearest to input image!

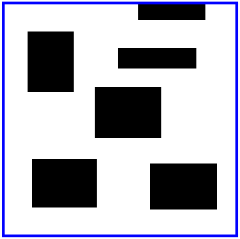
- Variables V are pixels, hypergraph E is the image grid.
- Variable domains $X_v = \{\,\mathsf{E, I, L, R, T, B, TL, TR, BL, BR}\,\}$ are syntactic parts of a rectangle.
- Unary constraint $f_v(x_v)$ quantifies agreement of intensity of state x_v and intensity of input pixel v.
- Binary constraint $f_{vv'}(x_v, x_{v'})$ equals 0 if syntactic parts x_v and $x_{v'}$ are allowed to neighbor and $-\infty$ otherwise.

E	E	E	E	E	E	E
E	TL	T	T	T	TR	E
E	L	I	I	I	R	E
E	L	I	I	I	R	E
E	BL	B	B	B	BR	E
E	E	E	E	E	E	E

hidden states = syntactic parts
observed states = {black,white}



input image



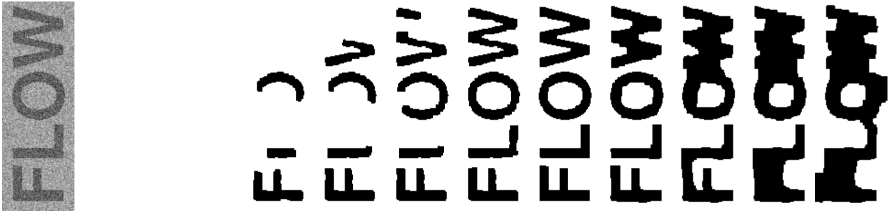
output image

Example: Binary WCSP with a global constraint [Werner-2008]

- $E = \binom{V}{1} \cup E' \cup \{V\}$ where $E' \subseteq \binom{V}{2}$, $X_v = \{\text{white}, \text{black}\}$.
- $(S, \oplus, \odot) = (\mathbb{R} \cup \{-\infty\}, \max, +)$
- Unary constraint f_v quantifies agreement with intensity of input pixel v.
- Binary constraints $f_{vv'}$ penalise transition between black and white pixels.
- Global constraint $f_V(x_V)$ is: 0 if x_V contains n black pixels and $-\infty$ otherwise.

Interpretation: Minimum *st*-cut in a graph such that the number of pixels in the first partition equals n (NP-hard).

- Max-sum diffusion enforced generalised arc consistency of active joint states.
- Marginal equalisation between f_v and f_V seen as a soft global propagator.



input									
	n required:	2000	3000	4000	5000	5368	6000	7000	8000	9000
	n achieved:	2008	3004	4011	5006	5368	6004	7024	7982	9032

Supermodular problems

Let each domain X_v be totally ordered. A function f_A is supermodular if

$$f_A(x_A \wedge y_A) + f_A(x_A \vee y_A) \geq f_A(x_A) + f_A(y_A)$$

for any $x_A, y_A \in X_A$, where $\wedge$ ($\vee$) denotes the elementwise minimum (maximum).

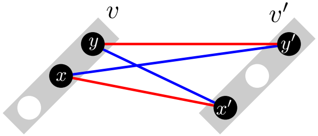
Theorem ([Schlesinger-Flach-00] for binary case, [Werner-2008] for non-binary case)

Let f_A be supermodular for each $A \in E$. Finding an equivalent network whose active joint states are generalised arc consistent solves the WCSP f exactly.

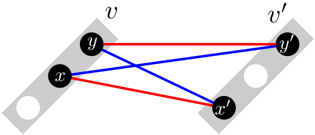
Proof

- Equivalent transformations preserve supermodularity.
- The maximisers of a supermodular function on a distributive lattice form a sublattice of this lattice [Topkis78]. Hence, the active joint states of each constraint form a lattice. Hence, all active joint states form a well-known tractable CSP [Jeavons-Cooper-95] (lattice CSP).
- Generalised arc consistency suffices for a lattice CSP to be satisfiable.

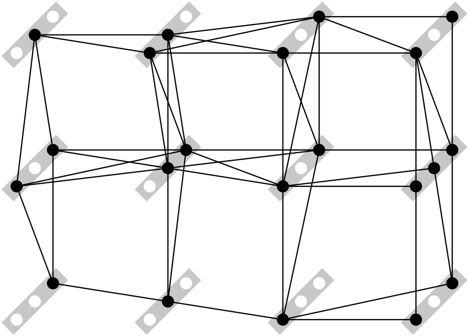
Before, [Cooper-2008] showed that the LP relaxation is tight for non-binary supermodular WCSPs. Our statement is stronger and the proof is simpler.

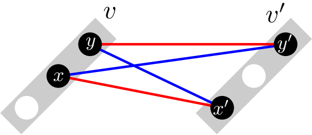


$$x \le x', y \le y' \quad \Longrightarrow \quad f_{vv'}(x,x') + f_{vv'}(y,y') \ge f_{vv'}(x,y') + f_{vv'}(y,x')$$

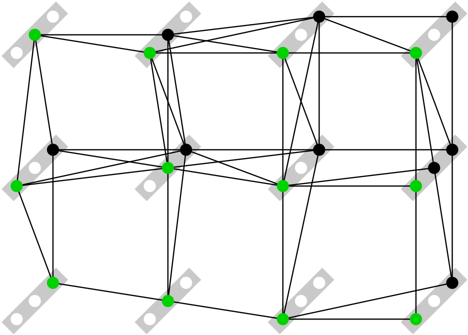


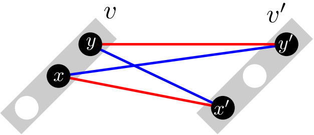
$$x \leq x', y \leq y' \quad \Longrightarrow \quad f_{vv'}(x, x') + f_{vv'}(y, y') \geq f_{vv'}(x, y') + f_{vv'}(y, x')$$



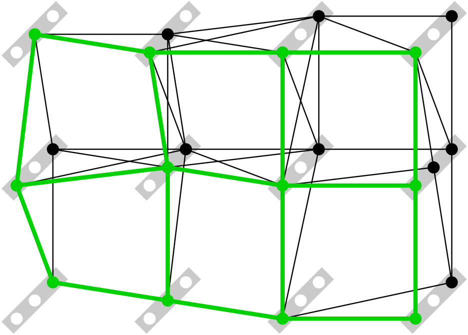


$$x \leq x', y \leq y' \implies f_{vv'}(x, x') + f_{vv'}(y, y') \geq f_{vv'}(x, y') + f_{vv'}(y, x')$$





$$x \leq x', y \leq y' \quad \Longrightarrow \quad f_{vv'}(x, x') + f_{vv'}(y, y') \geq f_{vv'}(x, y') + f_{vv'}(y, x')$$



Summary

- A deep property of constraint networks on commutative semirings revealed:

By changing constraints locally, any constraint network can be transformed to an equivalent form in which corresponding marginals of each constraint pair coincide.

- Marginal consistency can be enforced in a wide class of semirings, including the max-min, max-sum, and sum-product semirings.
- It corresponds to finding a local minimum of an upper bound on the true objective function.
- Enforcing marginal consistency is a simple concept, hinging only on local equivalent transformations and marginal consistency.