Bounds on Weighted CSPs Using Constraint Propagation and Super-Reparametrizations

Tomáš Dlask[1], Tomáš Werner[1], Simon de Givry[2]

[1] Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic
[2] Université Fédérale de Toulouse, ANITI, INRAE, UR 875, Toulouse, France

CP 2021 conference paper

presented at

Combinatorial Image Analysis Workshop, TU Dresden

(Binary) Weighted Constraint Satisfaction Problem (WCSP)

- finite set of **variables** V
- finite **domain** D of each variable
- set of variable pairs (edges) E
- find assignment $x \in D^V$ (i.e., $x\colon V \to D$) maximizing

$$F(x \mid f) = \sum_{i \in V} f_i(x_i) + \sum_{\{i,j\} \in E} f_{ij}(x_i, x_j) \tag{1}$$

- **weight functions** $f_i\colon D \to \mathbb{R}$ and $f_{ij}\colon D^2 \to \mathbb{R}$

(Binary) Weighted Constraint Satisfaction Problem (WCSP)

- finite set of **variables** V
- finite **domain** D of each variable
- set of variable pairs (edges) E
- find assignment $x \in D^V$ (i.e., $x\colon V \to D$) maximizing

$$F(x \mid f) = \sum_{i \in V} f_i(x_i) + \sum_{\{i,j\} \in E} f_{ij}(x_i, x_j) \tag{1}$$

- **weight functions** $f_i\colon D \to \mathbb{R}$ and $f_{ij}\colon D^2 \to \mathbb{R}$
- WCSP instances **are identified with vectors** $f \in \mathbb{R}^T$
- set of **tuples**:

$$T = \{\, (i,k) \mid i \in V,\ k \in D \,\} \cup \{\, \{(i,k),(j,l)\} \mid \{i,j\} \in E,\ k,l \in D \,\} \tag{2}$$

- Note: $F(x \mid f)$ is linear in f

(Binary) Constraint Satisfaction Problem (CSP)

- V, D, E, T as for WCSP, recall:

$$T = \{\, (i,k) \mid i \in V,\ k \in D \,\} \cup \{\, \{(i,k),(j,l)\} \mid \{i,j\} \in E,\ k,l \in D \,\}$$

- a set $A \subseteq T$ of **allowed tuples** (tuples in $T - A$ are forbidden)

(Binary) Constraint Satisfaction Problem (CSP)

- V, D, E, T as for WCSP, recall:

$$T = \{\,(i,k) \mid i \in V,\ k \in D\,\} \cup \{\,\{(i,k),(j,l)\} \mid \{i,j\} \in E,\ k,l \in D\,\}$$

- a set $A \subseteq T$ of **allowed tuples** (tuples in $T - A$ are forbidden)
- CSP instances are **identified with subsets of T**
- assignment $x \in D^V$ is a **solution** to CSP A if

$$\forall i \in V : (i, x_i) \in A$$
$$\forall \{i,j\} \in E : \{(i, x_i), (j, x_j)\} \in A$$

Upper Bound and Active-Tuple CSP

For a WCSP $f \in \mathbb{R}^T$:

- **upper bound** on the optimal value of WCSP f:

$$B(f) = \sum_{i \in V} \max_{k \in D} f_i(k) + \sum_{\{i,j\} \in E} \max_{k,l \in D} f_{ij}(k,l) \tag{3}$$

Upper Bound and Active-Tuple CSP

For a WCSP $f \in \mathbb{R}^T$:

- **upper bound** on the optimal value of WCSP f:

$$B(f) = \sum_{i \in V} \max_{k \in D} f_i(k) + \sum_{\{i,j\} \in E} \max_{k,l \in D} f_{ij}(k,l) \geq \max_{x \in D^V} \underbrace{\left(\sum_{i \in V} f_i(x_i) + \sum_{\{i,j\} \in E} f_{ij}(x_i, x_j) \right)}_{F(x \mid f)} \tag{3}$$

Upper Bound and Active-Tuple CSP

For a WCSP $f \in \mathbb{R}^T$:

- **upper bound** on the optimal value of WCSP f:

$$B(f) = \sum_{i \in V} \max_{k \in D} f_i(k) + \sum_{\{i,j\} \in E} \max_{k,l \in D} f_{ij}(k,l) \geq \max_{x \in D^V} \underbrace{\left(\sum_{i \in V} f_i(x_i) + \sum_{\{i,j\} \in E} f_{ij}(x_i, x_j) \right)}_{F(x \mid f)} \tag{3}$$

- tuple $t = (i,k) \in T$ is **active** if $f_i(k) = \max_{k' \in D} f_i(k')$
- tuple $t = \{(i,k),(j,l)\} \in T$ is **active** if $f_{ij}(k,l) = \max_{k',l' \in D} f_{ij}(k',l')$

Upper Bound and Active-Tuple CSP

For a WCSP $f \in \mathbb{R}^T$:

- **upper bound** on the optimal value of WCSP f:

$$B(f) = \sum_{i \in V} \max_{k \in D} f_i(k) + \sum_{\{i,j\} \in E} \max_{k,l \in D} f_{ij}(k,l) \geq \max_{x \in D^V} \underbrace{\left(\sum_{i \in V} f_i(x_i) + \sum_{\{i,j\} \in E} f_{ij}(x_i, x_j) \right)}_{F(x \mid f)} \quad (3)$$

- tuple $t = (i,k) \in T$ is **active** if $f_i(k) = \max_{k' \in D} f_i(k')$
- tuple $t = \{(i,k),(j,l)\} \in T$ is **active** if $f_{ij}(k,l) = \max_{k',l' \in D} f_{ij}(k',l')$
- set of all **active tuples** for f is denoted by $A^*(f) \subseteq T$

For a WCSP $f \in \mathbb{R}^T$:

- **upper bound** on the optimal value of WCSP f:

$$B(f) = \sum_{i \in V} \max_{k \in D} f_i(k) + \sum_{\{i,j\} \in E} \max_{k,l \in D} f_{ij}(k,l) \geq \max_{x \in D^V} \underbrace{\left(\sum_{i \in V} f_i(x_i) + \sum_{\{i,j\} \in E} f_{ij}(x_i, x_j) \right)}_{F(x \mid f)} \quad (3)$$

- tuple $t = (i,k) \in T$ is **active** if $f_i(k) = \max_{k' \in D} f_i(k')$
- tuple $t = \{(i,k),(j,l)\} \in T$ is **active** if $f_{ij}(k,l) = \max_{k',l' \in D} f_{ij}(k',l')$
- set of all **active tuples** for f is denoted by $A^*(f) \subseteq T$

Theorem[1]: Upper bound is tight (i.e., $B(f) = \max_x F(x \mid f)$) iff CSP $A^*(f)$ is satisfiable.

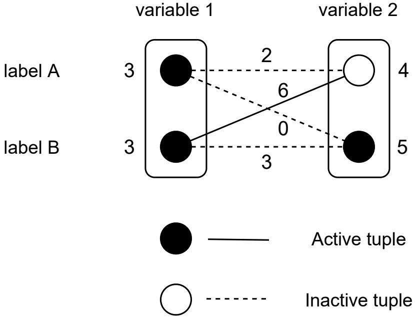
[1]Werner: A linear programming approach to max-sum problem: A review

Example

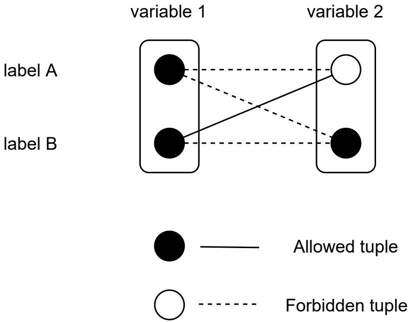
$V = \{1, 2\}$, $D = \{\text{A, B}\}$, $E = \{\{1, 2\}\}$, $F(x \mid f) = f_1(x_1) + f_2(x_2) + f_{12}(x_1, x_2)$

WCSP f

$(B(f) = 3 + 6 + 5 = 14)$



CSP $A^*(f)$



Minimizing the Upper Bound by Reparametrizations

WCSP f is a **reparametrization** of WCSP g if $F(x|f) = F(x|g)$ for all x.

Minimizing the Upper Bound by Reparametrizations

WCSP f is a **reparametrization** of WCSP g if $F(x|f) = F(x|g)$ for all x.

Given a WCSP $g \in \mathbb{R}^T$, minimize the upper bound over reparametrizations:

$$\min_{f \in \mathbb{R}^T} B(f) \qquad \text{subject to} \qquad F(x|f) = F(x|g) \;\; \forall x \in D^V \tag{4}$$

- a polynomially-sized LP formulation exists[2] – dual of the basic LP relaxation of WCSP g

---

[2] Werner: A linear programming approach to max-sum problem: A review

Minimizing the Upper Bound by Reparametrizations

WCSP f is a **reparametrization** of WCSP g if $F(x|f) = F(x|g)$ for all x.

Given a WCSP $g \in \mathbb{R}^T$, minimize the upper bound over reparametrizations:

$$\min_{f \in \mathbb{R}^T} B(f) \qquad \text{subject to} \qquad F(x|f) = F(x|g) \;\; \forall x \in D^V \tag{4}$$

- a polynomially-sized LP formulation exists[2] – dual of the basic LP relaxation of WCSP g
- many algorithms for its (approximate) optimization:
  - (a) block-coordinate descent (e.g., max-sum diffusion[2,3], TRWS[4], SPAM[5], ...)
  - (b) soft local consistencies (e.g., EDAC[6], AugDAG[7]/VAC[8], ...)

---

[2] Werner: A linear programming approach to max-sum problem: A review
[3] Kovalevsky et al.: A diffusion algorithm for decreasing energy of max-sum labeling problem
[4] Kolmogorov: Convergent tree-reweighted message passing for energy minimization
[5] Tourani et al.: Taxonomy of dual block-coordinate ascent methods for discrete energy minimization
[6] Larrosa et al.: Existential arc consistency: getting closer to full arc consistency in weighted CSPs
[7] Koval et al.: Two-dimensional programming in image analysis problems
[8] Cooper et al.: Soft arc consistency revisited

Minimizing the Upper Bound by Super-Reparametrizations

WCSP f is a **super-reparametrization** of WCSP g if $F(x|f) \geq F(x|g)$ for all x.

Minimizing the Upper Bound by Super-Reparametrizations

WCSP f is a **super-reparametrization** of WCSP g if $F(x|f) \geq F(x|g)$ for all x.

Given a WCSP $g \in \mathbb{R}^T$, minimize the upper bound over super-reparametrizations:[9]

$$\min_{f \in \mathbb{R}^T} B(f) \qquad \text{subject to} \qquad F(x|f) \geq F(x|g) \;\; \forall x \in D^V \tag{5}$$

[9] Komodakis, et al.: Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles

Minimizing the Upper Bound by Super-Reparametrizations

WCSP f is a **super-reparametrization** of WCSP g if $F(x|f) \geq F(x|g)$ for all x.

Given a WCSP $g \in \mathbb{R}^T$, minimize the upper bound over super-reparametrizations:[9]

$$\min_{f \in \mathbb{R}^T} B(f) \qquad \text{subject to} \qquad F(x|f) \geq F(x|g) \;\; \forall x \in D^V \tag{5}$$

- for any feasible f: $B(f) \geq \max_x F(x|f) \geq \max_x F(x|g)$

[9] Komodakis, et al.: Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles

Minimizing the Upper Bound by Super-Reparametrizations

WCSP f is a **super-reparametrization** of WCSP g if $F(x|f) \geq F(x|g)$ for all x.

Given a WCSP $g \in \mathbb{R}^T$, minimize the upper bound over super-reparametrizations:[9]

$$\min_{f \in \mathbb{R}^T} B(f) \qquad \text{subject to} \qquad F(x|f) \geq F(x|g) \;\; \forall x \in D^V \tag{5}$$

- for any feasible f: $B(f) \geq \max_x F(x|f) \geq \max_x F(x|g)$
- optimal value of (5) is $\max_x F(x|g)$

[9] Komodakis, et al.: Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles

Minimizing the Upper Bound by Super-Reparametrizations

WCSP f is a **super-reparametrization** of WCSP g if $F(x|f) \geq F(x|g)$ for all x.

Given a WCSP $g \in \mathbb{R}^T$, minimize the upper bound over super-reparametrizations:[9]

$$\min_{f \in \mathbb{R}^T} B(f) \qquad \text{subject to} \qquad F(x|f) \geq F(x|g) \;\; \forall x \in D^V \tag{5}$$

- for any feasible f: $B(f) \geq \max_x F(x|f) \geq \max_x F(x|g)$
- optimal value of (5) is $\max_x F(x|g)$
- feasible $f \in \mathbb{R}^T$ is optimal for (5) iff CSP $A^*(f)$ has a solution x with $F(x|f) = F(x|g)$
- satisfiability of $A^*(f)$ is a necessary (but generally insufficient) condition of optimality

[9] Komodakis, et al.: Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles

Iterative Scheme

Theorem: Let $f \in \mathbb{R}^T$. CSP $A^*(f)$ is unsatisfiable iff $\exists h \in \mathbb{R}^T$ with $B(f+h) < B(f)$ and $F(x \,|\, h) \geq 0$ for all $x \in D^V$.

Theorem: Let $f \in \mathbb{R}^T$. CSP $A^*(f)$ is unsatisfiable iff $\exists h \in \mathbb{R}^T$ with $B(f+h) < B(f)$ and $F(x \,|\, h) \geq 0$ for all $x \in D^V$. Such h is a **certificate of unsatisfiability** for f.

Iterative Scheme

Theorem: Let $f \in \mathbb{R}^T$. CSP $A^*(f)$ is unsatisfiable iff $\exists h \in \mathbb{R}^T$ with $B(f+h) < B(f)$ and $F(x \,|\, h) \geq 0$ for all $x \in D^V$. Such h is a **certificate of unsatisfiability** for f.

Iterative scheme for (approximate) optimization:
(WCSP g given as input)

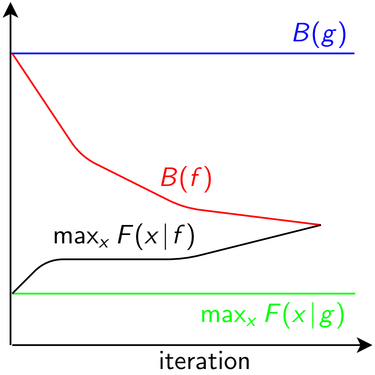
1: Initialize $f := g$.
2: If CSP $A^*(f)$ is satisfiable, stop.
3: Find certificate h.
4: Update $f := f + h$ and go to 2.

Iterative Scheme

Theorem: Let $f \in \mathbb{R}^T$. CSP $A^*(f)$ is unsatisfiable iff $\exists h \in \mathbb{R}^T$ with $B(f+h) < B(f)$ and $F(x\,|\,h) \geq 0$ for all $x \in D^V$. Such h is a **certificate of unsatisfiability** for f.

Iterative scheme for (approximate) optimization:
(WCSP g given as input)

1: Initialize $f := g$.
2: If CSP $A^*(f)$ is satisfiable, stop.
3: Find certificate h.
4: Update $f := f + h$ and go to 2.



Properties:

- $B(f)$ decreases after each iteration
- $F(x\,|\,f+h) = F(x\,|\,f) + F(x\,|\,h) \geq F(x\,|\,f)$
- $\max_x F(x\,|\,f)$ increases or stays the same after each iteration
- obtained bound is limited by the fact that $B(f) \geq \max_x F(x\,|\,f)$

Iterative Scheme with Constraint Propagation

Idea: Try to detect unsatisfiability of CSP $A^*(f)$ by constraint propagation.

Next, we will show:

- how to compute a certificate h using any constraint propagation algorithm
- how to obtain good certificates
- experiments with singleton arc consistency

Deactivating Directions

Definition: Let $A \subseteq T$ and $S \subseteq A$, $S \neq \emptyset$.
An S-**deactivating direction for CSP** A is a vector $d \in \mathbb{R}^T$ satisfying

(a) $d_t < 0$ for all $t \in S$,
(b) $d_t = 0$ for all $t \in A - S$,
(c) $F(x \,|\, d) \geq 0$ for all $x \in D^V$.

Deactivating Directions

Definition: Let $A \subseteq T$ and $S \subseteq A$, $S \neq \emptyset$.
An S-**deactivating direction for CSP** A is a vector $d \in \mathbb{R}^T$ satisfying

(a) $d_t < 0$ for all $t \in S$,
(b) $d_t = 0$ for all $t \in A - S$,
(c) $F(x \,|\, d) \geq 0$ for all $x \in D^V$.

Theorem: An S-deactivating direction $d \in \mathbb{R}^T$ for A exists iff CSPs A and $A - S$ have the same solution set.

Deactivating Directions

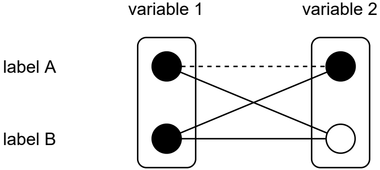
Definition: Let $A \subseteq T$ and $S \subseteq A$, $S \neq \emptyset$.
An S-**deactivating direction for CSP** A is a vector $d \in \mathbb{R}^T$ satisfying

(a) $d_t < 0$ for all $t \in S$,
(b) $d_t = 0$ for all $t \in A - S$,
(c) $F(x \,|\, d) \geq 0$ for all $x \in D^V$.

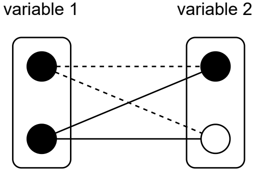
Theorem: An S-deactivating direction $d \in \mathbb{R}^T$ for A exists iff CSPs A and $A - S$ have the same solution set.

Example: $V = \{1, 2\}$, $D = \{\text{A, B}\}$, $E = \{\{1, 2\}\}$, $S = \{\{(1, \text{A}), (2, \text{B})\}\}$

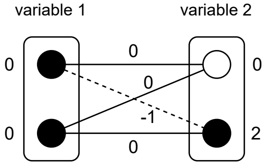
CSP A



CSP $A - S$



S-deactivating direction d for A



Deactivating Directions

$$\left.\begin{array}{l}\textbf{Domain wipeout}\text{: for some } i \in V,\ (i,k) \notin A \text{ for all } k \in D \\ \textbf{Edge wipeout}\text{: for some } \{i,j\} \in E,\ \{(i,k),(j,l)\} \notin A \text{ for all } k,l \in D\end{array}\right\} \implies A \text{ is unsatisfiable}$$

Deactivating Directions

$$\left.\begin{array}{l}\textbf{Domain wipeout}\text{: for some } i \in V,\ (i,k) \notin A \text{ for all } k \in D \\ \textbf{Edge wipeout}\text{: for some } \{i,j\} \in E,\ \{(i,k),(j,l)\} \notin A \text{ for all } k,l \in D\end{array}\right\} \implies A \text{ is unsatisfiable}$$

Theorem: Let $f \in \mathbb{R}^T$ and d be an S-deactivating direction for $A^*(f)$.

- If there is domain or edge wipeout in $A^*(f) - S$, then $\exists \alpha > 0 : B(f + \underbrace{\alpha d}_{h}) < B(f)$.
- Otherwise, $\exists \alpha > 0 : B(f + \alpha d) = B(f)$ and $A^*(f + \alpha d) = A^*(f) - S$.

Computing Deactivating Directions

One iteration of a local consistency algorithm applied to CSP $A \subseteq T$

- identify tuples $S \subseteq A$ such that CSPs A and $A - S$ have the same solution set
- forbid these tuples: update CSP to $A - S$

Computing Deactivating Directions

One iteration of a local consistency algorithm applied to CSP $A \subseteq T$

- identify tuples $S \subseteq A$ such that CSPs A and $A - S$ have the same solution set
- forbid these tuples: update CSP to $A - S$

Refinement: **proof set** $P \subseteq T - A$

- for any CSP $A' \subseteq T - P$, CSPs A' and $A' - S$ have the same solution set
- e.g., $P = T - A$

Computing Deactivating Directions

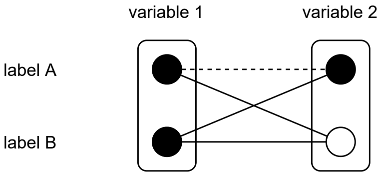
One iteration of a local consistency algorithm applied to CSP $A \subseteq T$

- identify tuples $S \subseteq A$ such that CSPs A and $A - S$ have the same solution set
- forbid these tuples: update CSP to $A - S$

Refinement: **proof set** $P \subseteq T - A$

- for any CSP $A' \subseteq T - P$, CSPs A' and $A' - S$ have the same solution set
- e.g., $P = T - A$

Example: $S = \{\{(1, \mathrm{A}), (2, \mathrm{B})\}\}$, $P = \{(2, \mathrm{B})\}$, CSP A:



Computing Deactivating Directions

One iteration of a local consistency algorithm applied to CSP $A \subseteq T$

- identify tuples $S \subseteq A$ such that CSPs A and $A - S$ have the same solution set
- forbid these tuples: update CSP to $A - S$

Refinement: **proof set $P \subseteq T - A$**

- for any CSP $A' \subseteq T - P$, CSPs A' and $A' - S$ have the same solution set
- e.g., $P = T - A$

Example (Singleton Arc Consistency): Let $i \in V$, $k \in D$, and $A \subseteq T$.

If $A|_{x_i=k}$ has empty AC closure:

- There is no solution of CSP A with $x_i = k$ and we can forbid (i, k).
- $S = \{(i, k)\}$
- P is, e.g., the set of forbidden tuples needed to infer empty AC closure of $A|_{x_i=k}$.

Computing Deactivating Directions

One iteration of a local consistency algorithm applied to CSP $A \subseteq T$

- identify tuples $S \subseteq A$ such that CSPs A and $A - S$ have the same solution set
- forbid these tuples: update CSP to $A - S$

Refinement: **proof set $P \subseteq T - A$**

- for any CSP $A' \subseteq T - P$, CSPs A' and $A' - S$ have the same solution set
- e.g., $P = T - A$

Theorem: S-deactivating direction for A:

$$d_t = \begin{cases} -1 & \text{if } t \in S \\ n & \text{if } t \in P \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where n is the number of weight functions (unary or binary) with at least one tuple in S.

Note: P is preferred to be small (so that values $F(x \mid d)$ are small)

Composing Deactivating Directions

Let $f \in \mathbb{R}^T$.

Given CSP $A_0 = A^*(f)$, apply constraint propagation to forbid some tuples:

- Forbid tuples $S_0 \subseteq A_0$, let $A_1 = A_0 - S_0$, store S_0-deactivating direction d^0 for A_0.

Composing Deactivating Directions

Let $f \in \mathbb{R}^T$.

Given CSP $A_0 = A^*(f)$, apply constraint propagation to forbid some tuples:

- Forbid tuples $S_0 \subseteq A_0$, let $A_1 = A_0 - S_0$, store S_0-deactivating direction d^0 for A_0.
- Forbid tuples $S_1 \subseteq A_1$, let $A_2 = A_1 - S_1$, store S_1-deactivating direction d^1 for A_1.

  $\vdots$

Composing Deactivating Directions

Let $f \in \mathbb{R}^T$.

Given CSP $A_0 = A^*(f)$, apply constraint propagation to forbid some tuples:

- Forbid tuples $S_0 \subseteq A_0$, let $A_1 = A_0 - S_0$, store S_0-deactivating direction d^0 for A_0.
- Forbid tuples $S_1 \subseteq A_1$, let $A_2 = A_1 - S_1$, store S_1-deactivating direction d^1 for A_1.

  $\vdots$
- Forbid tuples $S_q \subseteq A_q$, let $A_{q+1} = A_q - S_q$, store S_q-deactivating direction d^q for A_q.

Composing Deactivating Directions

Let $f \in \mathbb{R}^T$.

Given CSP $A_0 = A^*(f)$, apply constraint propagation to forbid some tuples:

- Forbid tuples $S_0 \subseteq A_0$, let $A_1 = A_0 - S_0$, store S_0-deactivating direction d^0 for A_0.
- Forbid tuples $S_1 \subseteq A_1$, let $A_2 = A_1 - S_1$, store S_1-deactivating direction d^1 for A_1.

  $\vdots$
- Forbid tuples $S_q \subseteq A_q$, let $A_{q+1} = A_q - S_q$, store S_q-deactivating direction d^q for A_q.

Theorem: The sequence $d^0, d^1, ..., d^q$ can be composed into a single S-deactivating direction d for $A^*(f)$ where $S = S_0 \cup S_1 \cup ... \cup S_q$.

Note: $A_{q+1} = A_0 - S$

Corollary: If there is domain or edge wipeout in $A_{q+1} = A^*(f) - S$, d can be used to improve the bound as $B(f + \alpha d) < B(f)$.

Overview

Iterative scheme for computing an upper bound $B(f)$ on $\max_x F(x \mid g)$:

1: Initialize $f := g$.

2: Apply constraint propagation on $A^*(f)$ while storing deactivating directions $d^0, \dots, d^q$

Overview

Iterative scheme for computing an upper bound $B(f)$ on $\max_x F(x|g)$:

1: Initialize $f := g$.

2: Apply constraint propagation on $A^*(f)$ while storing deactivating directions $d^0, ..., d^q$

3: If there is domain or edge wipeout:

  3.1: Compose (possibly a subset of) the sequence $d^0, ..., d^q$ into a single vector d.

  3.2: Compute step size α.

  3.3: Update $f := f + \alpha d$, go to 2.

4: Return $B(f)$.

Experiments

Data: Cost Function Library benchmark[10]

Compared methods:

- Virtual singleton arc consistency via super-reparametrizations (VSAC-SR)
- Virtual cycle consistency via super-reparametrizations (VCC-SR) ([11])
- EDAC[12], VAC[13], pseudo-triangles[14], triangle-based consistencies: PIC, EDPIC, maxRPC, EDmaxRPC[15]

Only the upper bound is computed.

[10] https://forgemia.inra.fr/thomas.schiex/cost-function-library
[11] Komodakis, et al.: Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles
[12] Larrosa et al.: Existential arc consistency: getting closer to full arc consistency in weighted CSPs
[13] Cooper et al.: Soft arc consistency revisited
[14] Option $-t = 8000$ in toulbar2; https://miat.inrae.fr/toulbar2
[15] Nguyen, et al.: Triangle-based consistencies for cost function networks

Instance Group	Instances	EDAC	VAC	VSAC-SR	VCC-SR	Pseudo-tr.	PIC	EDPIC	maxRPC	EDmaxRPC
/biqmaclib/	157	0.02	0.11	0.90	0.22	**0.92**	0.83	0.81	0.79	0.81
/crafted/academics/	8	0.88	0.88	0.97	0.95	0.88	0.88	0.88	0.88	**1.00**
/crafted/auction/paths/	420	0.00	0.09	0.91	0.35	**0.99**	0.45	0.68	0.64	0.57
/crafted/auction/regions/	411	0.00	0.05	**0.99**	0.10	0.98	0.08	0.18	0.23	0.13
/crafted/auction/scheduling/	419	0.00	0.02	**1.00**	0.09	0.80	0.41	0.38	0.41	0.24
/crafted/coloring/	33	0.94	0.94	0.99	0.97	0.98	**1.00**	**1.00**	**1.00**	0.99
/crafted/feedback/	6	0.00	0.00	0.54	0.58	0.71	0.49	0.53	0.51	**0.72**
/crafted/kbtree/	1800	0.25	0.29	0.60	0.67	0.80	0.73	0.81	0.76	**0.89**
/crafted/maxclique/dimacs_maxclique/	49	0.06	0.24	**0.98**	0.39	0.87	0.39	0.50	0.51	0.55
/crafted/maxcut/spinglass_maxcut/unweighted/	5	0.00	0.00	**1.00**	0.42	0.15	0.15	0.15	0.15	0.15
/crafted/maxcut/spinglass_maxcut/weighted/	5	0.00	0.00	**1.00**	0.38	0.17	0.17	0.17	0.17	0.17
/crafted/modularity/	6	0.17	0.19	0.38	0.25	**0.99**	0.96	0.94	0.96	0.97
/crafted/planning/	65	0.00	0.54	**0.94**	0.72	0.32	0.07	0.09	0.07	0.17
/crafted/sumcoloring/	43	0.04	0.15	0.47	0.50	**0.81**	0.53	0.63	0.64	0.61
/crafted/warehouses/	49	0.35	0.99	**1.00**	0.99	0.35	0.42	0.42	0.42	0.42
/qaplib/	5	0.40	0.40	0.40	0.41	**0.99**	0.97	0.97	0.98	0.97
/qplib/	23	0.00	0.10	**0.96**	0.38	0.27	0.25	0.25	0.24	0.25
/random/maxcsp/completeloose/	50	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
/random/maxcsp/completetight/	50	0.00	0.12	0.57	0.72	0.88	0.94	**0.99**	0.69	0.76
/random/maxcsp/denseloose/	50	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
/random/maxcsp/densetight/	50	0.02	0.14	0.52	**1.00**	0.68	0.48	0.49	0.52	0.60
/random/maxcsp/sparseloose/	90	0.96	0.96	**1.00**	0.96	0.96	0.96	0.96	0.96	0.96
/random/maxcsp/sparsetight/	50	0.01	0.12	0.54	**1.00**	0.64	0.40	0.40	0.43	0.51
/random/maxcut/random_maxcut/	400	0.00	0.00	0.77	0.13	0.95	0.98	0.98	0.97	**0.99**
/random/mincut/	500	0.09	**1.00**	**1.00**	**1.00**	0.10	0.10	0.10	0.10	0.10
/random/randomksat/	493	0.01	0.02	0.75	0.22	**0.95**	0.91	0.89	0.86	0.87
/random/wqueens/	6	0.00	0.52	**0.96**	0.94	0.48	0.12	0.29	0.13	0.72
/real/celar/	23	0.00	0.05	0.08	0.16	**0.97**	0.66	0.66	0.78	0.95
/real/maxclique/protein_maxclique/	1	0.00	0.00	**1.00**	0.03	0.93	0.04	0.04	0.08	0.04
/real/spot5/	1	0.00	0.08	**1.00**	0.49	**1.00**	0.74	0.66	0.41	0.74
/real/tagsnp/tagsnp_r0.5/	23	0.04	0.86	**0.95**	0.86	0.31	0.31	0.33	0.29	0.46
/real/tagsnp/tagsnp_r0.8/	80	0.13	0.66	**0.91**	0.68	0.29	0.39	0.38	0.33	0.47
Average over all groups	5371	0.20	0.36	**0.82**	0.58	0.72	0.56	0.58	0.56	0.62
Average over groups with $\geq$ 5 instances	5369	0.21	0.38	**0.80**	0.60	0.71	0.57	0.59	0.58	0.63

Instance Group	Instances	EDAC	VAC	VSAC-SR	VCC-SR	Pseudo-tr.	PIC	EDPIC	maxRPC	EDmaxRPC
/biqmaclib/	157	**0.11**	0.12	180.07	34.60	83.25	1240.00	1241.29	1242.16	1271.86
/crafted/academics/	8	**0.11**	**0.11**	28.61	1.04	29.08	121.44	120.86	108.08	104.47
/crafted/auction/paths/	420	**0.04**	**0.04**	1.96	0.83	1.92	0.19	0.23	0.48	0.64
/crafted/auction/regions/	411	**0.20**	0.32	32.14	9.45	673.42	49.85	51.37	102.61	110.48
/crafted/auction/scheduling/	419	**0.10**	0.12	16.22	2.03	49.85	26.90	26.89	32.06	32.30
/crafted/coloring/	33	**0.09**	0.10	4.99	1.40	0.20	545.50	545.50	545.51	545.50
/crafted/feedback/	6	**0.70**	**0.70**	3588.39	3600.11	11.64	1860.89	1874.08	1875.93	1873.07
/crafted/kbtree/	1800	**0.02**	**0.02**	3.13	11.25	0.10	0.04	0.05	0.06	0.07
/crafted/maxclique/dimacs_maxclique/	49	**0.71**	1.32	279.08	126.90	955.60	1345.67	1342.14	1429.73	1428.12
/crafted/maxcut/spinglass_maxcut/unweighted/	5	0.02	0.02	0.82	0.44	0.02	**0.01**	**0.01**	**0.01**	**0.01**
/crafted/maxcut/spinglass_maxcut/weighted/	5	0.02	0.02	1.09	0.53	0.02	**0.01**	**0.01**	**0.01**	**0.01**
/crafted/modularity/	6	**0.19**	0.29	1023.48	127.39	66.25	706.30	783.02	741.91	1442.57
/crafted/planning/	65	**0.16**	0.29	638.85	60.62	7.41	0.93	0.96	2.33	4.73
/crafted/sumcoloring/	43	**1.29**	1.94	727.49	963.61	255.72	1508.37	1508.36	1509.34	1512.68
/crafted/warehouses/	49	4.10	9.48	735.80	735.83	**4.09**	29.48	29.54	28.80	29.82
/qaplib/	5	**0.08**	0.09	119.05	278.53	7.38	1448.63	1444.95	1450.09	1449.22
/qplib/	23	**0.13**	0.14	255.85	43.11	195.32	626.25	626.24	626.27	626.36
/random/maxcsp/completeloose/	50	**0.06**	**0.06**	1.31	0.16	0.48	0.09	0.10	0.19	0.18
/random/maxcsp/completetight/	50	**0.02**	0.03	6.35	12.68	0.47	0.21	0.25	0.31	0.33
/random/maxcsp/denseloose/	50	**0.02**	**0.02**	166.78	0.06	0.11	0.03	0.03	0.03	0.03
/random/maxcsp/densetight/	50	**0.02**	**0.02**	4.20	17.38	0.10	0.06	0.07	0.07	0.08
/random/maxcsp/sparseloose/	90	**0.03**	**0.03**	611.38	0.05	0.06	0.04	0.04	0.04	0.04
/random/maxcsp/sparsetight/	50	**0.02**	**0.02**	11.00	9.74	0.06	0.04	0.05	0.05	0.05
/random/maxcut/random_maxcut/	400	**0.01**	**0.01**	0.73	0.15	0.04	0.03	0.03	0.05	0.07
/random/mincut/	500	1.09	2.43	14.40	86.22	1.12	0.88	**0.87**	**0.87**	**0.87**
/random/randomksat/	493	**0.02**	**0.02**	3.42	0.17	0.13	0.07	0.10	0.16	0.31
/random/wqueens/	6	**1.33**	1.49	992.85	502.42	644.87	1800.15	1800.20	1800.18	1800.60
/real/celar/	23	**0.27**	0.28	1798.51	2972.69	66.56	300.76	219.91	495.26	1066.87
/real/maxclique/protein_maxclique/	1	**0.26**	0.44	25.24	6.77	1196.62	114.62	114.99	215.30	220.81
/real/spot5/	1	**0.01**	**0.01**	0.62	0.08	0.11	0.03	0.03	0.04	0.04
/real/tagsnp/tagsnp_r0.5/	23	**4.83**	378.77	3338.53	2897.83	239.38	3155.96	3148.66	3172.58	3295.19
/real/tagsnp/tagsnp_r0.8/	80	**1.52**	22.82	1239.73	858.83	90.05	195.12	206.76	359.55	409.88
Average over all groups	5371	**0.55**	13.17	495.38	417.59	143.17	471.21	471.49	491.88	538.35
Average over groups with $\geq$ 5 instances	5369	**0.58**	14.04	527.54	445.20	112.82	498.80	499.08	517.49	566.88