Parameterized Complexity of Equality Constraint Satisfaction

George Osipov & Magnus Wahlström

May 2, 2023

Linköping University, Sweden & Royal Holloway University of London, UK

Constraint Satisfaction (CSP)

Fix a domain of values D. A **relation** R is a subset of tuples $(d_1, \ldots, d_r) \in D^r$, e.g.

- $\{(a, a) : a \in D\}$ is the binary equality relation,
- $\{(a, b) : a, b \in D, a \neq b\}$ is the binary disequality relation.

Constraint Satisfaction (CSP)

Fix a domain of values D. A **relation** R is a subset of tuples $(d_1, \ldots, d_r) \in D^r$, e.g.

- $\{(a, a) : a \in D\}$ is the binary equality relation,
- $\{(a, b) : a, b \in D, a \neq b\}$ is the binary disequality relation.

A **constraint** $R(x_1, \ldots, x_r)$ consists of a relation R applied to a tuple of variables $(x_1, \ldots, x_r)$.

An assignment $\alpha : V(X) \to D$ **satisfies** the constraint if $(\alpha(x_1), \ldots, \alpha(x_r)) \in R$.

Constraint Satisfaction (CSP)

Fix a domain of values D. A **relation** R is a subset of tuples $(d_1, \ldots, d_r) \in D^r$, e.g.

- $\{(a, a) : a \in D\}$ is the binary equality relation,
- $\{(a, b) : a, b \in D, a \neq b\}$ is the binary disequality relation.

A **constraint** $R(x_1, \ldots, x_r)$ consists of a relation R applied to a tuple of variables $(x_1, \ldots, x_r)$.

An assignment $\alpha : V(X) \to D$ **satisfies** the constraint if $(\alpha(x_1), \ldots, \alpha(x_r)) \in R$.

A **constraint language** Γ is a subset of relations.

Constraint Satisfaction (CSP)

Constraint Satisfaction Problem for Γ aka CSP(Γ)

Instance: A set variables V and a set of constraints C of the form $R(x_1, \ldots, x_r)$, where $R \in \Gamma$ and $x_1, \ldots, x_r \in V$.
Question: Is there an assignment $\alpha : V \to D$ that satisfies all constraints in C?

Constraint Satisfaction (CSP)

CONSTRAINT SATISFACTION PROBLEM FOR Γ AKA CSP(Γ)

INSTANCE: A set variables V and a set of constraints C of the form $R(x_1, \ldots, x_r)$, where $R \in \Gamma$ and $x_1, \ldots, x_r \in V$.
QUESTION: Is there an assignment $\alpha : V \to D$ that satisfies all constraints in C?

Example: Consider an instance of CSP($=, \neq$) with domain $\mathbb{N}$:

$$x_1 = x_2, x_2 = x_3, x_3 = x_4, x_1 \neq x_4.$$

This instance is not satisfiable because $x_1 = x_2 = x_3 = x_4$ implies $x_1 = x_4$ and contradicts $x_1 \neq x_4$.

MinCSP(Γ)

Instance: An instance (V, C) of CSP(Γ), and integer k.
Question: Is there a subset $X \subseteq C$ of $\leq k$ constraints such that $(V, C \setminus X)$ is satisfiable?

MinCSP

MinCSP(Γ)

Instance: An instance (V, C) of CSP(Γ), and integer k.
Question: Is there a subset $X \subseteq C$ of $\leq k$ constraints such that $(V, C \setminus X)$ is satisfiable?

We study this problem under the natural parameter k.

Equality MinCSP

MinCSP$(=, \neq)$ with domain $\mathbb{N}$ is equivalent to

Edge Multicut

Instance: Graph G, cut requests $s_1t_1, \ldots, s_\ell t_\ell$, and integer k.
Question: Is there $X \subseteq E(G)$ of size $\leq k$ separating s_i and t_i for all i?

Edge uv becomes $u = v$. Request s_it_i becomes $s_i \neq t_i$.
Edge Multicut is in FPT ([BDT'11], [MR'11]).

Equality MinCSP

An **equality constraint relation** is any relation definable using $\wedge, \vee$ and predicates $=, \neq$, e.g.

$\{(a, b, c) \mid a, b, c \in \mathbb{N}, (a = b \wedge b = c) \vee (a \neq b \wedge b \neq c \wedge a \neq c)\}$.

This relation accepts $(1, 1, 1)$, $(1, 2, 3)$, but rejects $(1, 2, 2)$.

Equality MinCSP

An **equality constraint relation** is any relation definable using $\wedge, \vee$ and predicates $=, \neq$, e.g.

$\{(a, b, c) \mid a, b, c \in \mathbb{N}, (a = b \wedge b = c) \vee (a \neq b \wedge b \neq c \wedge a \neq c)\}$.

This relation accepts $(1, 1, 1)$, $(1, 2, 3)$, but rejects $(1, 2, 2)$.

We consider MinCSP(Γ) for all finite equality constraint languages Γ.

Why?

- It generalizes Edge Multicut, Steiner Multicut, Multiway Cut*.

Equality MinCSP

An **equality constraint relation** is any relation definable using $\wedge, \vee$ and predicates $=, \neq$, e.g.

$\{(a, b, c) \mid a, b, c \in \mathbb{N}, (a = b \wedge b = c) \vee (a \neq b \wedge b \neq c \wedge a \neq c)\}$.

This relation accepts $(1, 1, 1)$, $(1, 2, 3)$, but rejects $(1, 2, 2)$.

We consider MinCSP(Γ) for all finite equality constraint languages Γ.

Why?

- It generalizes Edge Multicut, Steiner Multicut, Multiway Cut*.
- It is a prerequisite for all infinite-domain MinCSP classifications.

Equality MinCSP

An **equality constraint relation** is any relation definable using $\wedge, \vee$ and predicates $=, \neq$, e.g.

$\{(a, b, c) \mid a, b, c \in \mathbb{N}, (a = b \wedge b = c) \vee (a \neq b \wedge b \neq c \wedge a \neq c)\}$.

This relation accepts $(1, 1, 1)$, $(1, 2, 3)$, but rejects $(1, 2, 2)$.

We consider MinCSP(Γ) for all finite equality constraint languages Γ.

Why?

- It generalizes Edge Multicut, Steiner Multicut, Multiway Cut*.
- It is a prerequisite for all infinite-domain MinCSP classifications.
- Full classifications highlight power and limits of fpt algorithms.

Equality MinCSP

An **equality constraint relation** is any relation definable using $\wedge, \vee$ and predicates $=, \neq$, e.g.

$\{(a, b, c) \mid a, b, c \in \mathbb{N}, (a = b \wedge b = c) \vee (a \neq b \wedge b \neq c \wedge a \neq c)\}$.

This relation accepts $(1, 1, 1)$, $(1, 2, 3)$, but rejects $(1, 2, 2)$.

We consider MinCSP(Γ) for all finite equality constraint languages Γ.

Why?

- It generalizes Edge Multicut, Steiner Multicut, Multiway Cut*.
- It is a prerequisite for all infinite-domain MinCSP classifications.
- Full classifications highlight power and limits of fpt algorithms.
- CSP is a nice fragment of NP for obtaining dichotomies.

Some Equality Constraint Relations

1. $\mathrm{NEQ}_3 = \{(a, b, c) : a \neq b \wedge b \neq c \wedge a \neq c\}$, accepts $(1, 2, 3)$.

MinCSP($\mathrm{NEQ}_3, =$) reduced to Multicut with Deletable Triples. FPT.

Some Equality Constraint Relations

1. $\mathrm{NEQ}_3 = \{(a, b, c) : a \neq b \wedge b \neq c \wedge a \neq c\}$, accepts $(1, 2, 3)$.

MinCSP($\mathrm{NEQ}_3, =$) reduced to Multicut with Deletable Triples. FPT.

2. $\mathrm{NAE}_3 = \{(a, b, c) : a \neq b \vee b \neq c \vee a \neq c\}$, rejects $(1, 1, 1)$.

MinCSP($\mathrm{NAE}_3, =$) $\approx$ 3-Steiner Multicut. W[1]-hard [BHMvL'16], 2-FPA.

Some Equality Constraint Relations

1. $\mathrm{NEQ}_3 = \{(a, b, c) : a \neq b \wedge b \neq c \wedge a \neq c\}$, accepts $(1, 2, 3)$.

MinCSP($\mathrm{NEQ}_3, =$) reduced to Multicut with Deletable Triples. FPT.

2. $\mathrm{NAE}_3 = \{(a, b, c) : a \neq b \vee b \neq c \vee a \neq c\}$, rejects $(1, 1, 1)$.

MinCSP($\mathrm{NAE}_3, =$) $\approx$ 3-Steiner Multicut. W[1]-hard [BHMvL'16], 2-FPA.

3. $\mathrm{R}^d_{\neq} = \{(a_1, b_1, \ldots, a_d, b_d) : a_1 \neq b_1 \vee \cdots \vee a_d \neq b_d\}$.

MinCSP($\mathrm{R}^d_{\neq}, =$) reduces to Disjunctive Multicut.
W[1]-hard for $d \geq 2$, has constant-factor FPA for all $d \in O(1)$.

Some Equality Constraint Relations

1. $\mathrm{NEQ}_3 = \{(a, b, c) : a \neq b \wedge b \neq c \wedge a \neq c\}$, accepts $(1, 2, 3)$.

MinCSP($\mathrm{NEQ}_3, =$) reduced to Multicut with Deletable Triples. FPT.

2. $\mathrm{NAE}_3 = \{(a, b, c) : a \neq b \vee b \neq c \vee a \neq c\}$, rejects $(1, 1, 1)$.

MinCSP($\mathrm{NAE}_3, =$) $\approx$ 3-Steiner Multicut. W[1]-hard [BHMvL'16], 2-FPA.

3. $\mathrm{R}^d_{\neq} = \{(a_1, b_1, \ldots, a_d, b_d) : a_1 \neq b_1 \vee \cdots \vee a_d \neq b_d\}$.

MinCSP($\mathrm{R}^d_{\neq}, =$) reduces to Disjunctive Multicut.
W[1]-hard for $d \geq 2$, has constant-factor FPA for all $d \in O(1)$.

4. $\mathrm{ODD}_3 = \{(a, b, c) : (a = b = c) \vee (a \neq b \wedge b \neq c \wedge a \neq c)\}$.

MinCSP($\mathrm{ODD}_3, =, \neq$) $\approx_{APX}$ Hitting Set. W[2]-hard, no c-FPA.

Classification Theorem

Let Γ be a finite equality constraint language such that CSP(Γ) is in P and MinCSP(Γ) is NP-hard. Then one of the following holds.

- MinCSP(Γ) is in FPT.
- MinCSP(Γ) is W[1]-hard but admits constant-factor FPA.
- MinCSP(Γ) is W[2]-hard and admits no constant-factor FPA.

Full Classification Details

MinCSP(Γ) is in FPT if it reduces to Multicut with Deletable Triples, and W[1]-hard otherwise.

MinCSP(Γ) admits constant-factor FPA if it reduces to Disjunctive Multicut, and $\approx_{APX}$ Hitting Set otherwise.

Full Classification Details

MinCSP(Γ) is in FPT if it reduces to Multicut with Deletable Triples, and W[1]-hard otherwise.

MinCSP(Γ) admits constant-factor FPA if it reduces to Disjunctive Multicut, and $\approx_{APX}$ Hitting Set otherwise.

Multicut with Deletable Triples solved using [KKPW'23].
Disjunctive Multicut constant-factor FPA algorithm – coming up.

The quest guided by equality CSP results of [BK'08] and [BCP'10].

Let G be a graph and $\mathcal{L}$ be a family of *list requests*, where $\mathcal{L} \ni L = \{s_1 t_1, \ldots, s_d t_d\}$.

A subset $X \subseteq V(G)$ satisfies L if X separates s_1, t_1 or s_2, t_2 or ...s_d, t_d.

Define $\mathsf{cost}(G, \mathcal{L}) = \min\{|X| : X \subseteq V(G), \forall L \in \mathcal{L}\ X$ satisfies $L\}$.

Disjunctive Multicut

Instance: Graph G, a family of request lists $\mathcal{L}$, and integer k.

Question: Is $\mathsf{cost}(G, \mathcal{L}) \leq k$?

Disjunctive Multicut

Instance: Graph G, a family of request lists $\mathcal{L}$, and integer k.

Question: Is $\mathrm{cost}(G, \mathcal{L}) \leq k$?

We allow **singleton** cut requests ss, which are satisfied by X if $s \in X$.

Disjunctive Multicut

Instance: Graph G, a family of request lists $\mathcal{L}$, and integer k.

Question: Is $\mathsf{cost}(G, \mathcal{L}) \leq k$?

We allow **singleton** cut requests ss, which are satisfied by X if $s \in X$.

Hitting Set $\approx_{APX}$ DJMC: set $\{a, b, c\}$ becomes request list $\{aa, bb, cc\}$.

We assume that the lists have at most $d \in O(1)$ entries.

$2d$-Hitting Set is in FPT.

Main idea: design an algorithm $(G, \mathcal{L}) \to (G', \mathcal{L}')$ such that $\mathsf{cost}(G', \mathcal{L}') \leq c \cdot \mathsf{cost}(G, \mathcal{L})$ and $(G', \mathcal{L}')$ is closer to $2d$-Hitting Set.

Progress measure μ = maximum # non-singletons in a list of $\mathcal{L}$.

Disjunctive Multicut

Main idea: design an algorithm $(G, \mathcal{L}) \to (G', \mathcal{L}')$ such that $\mathsf{cost}(G', \mathcal{L}') \leq c \cdot \mathsf{cost}(G, \mathcal{L})$ and $(G', \mathcal{L}')$ is closer to $2d$-Hitting Set.

Progress measure $\mu =$ maximum # non-singletons in a list of $\mathcal{L}$.

Main idea: design an algorithm $(G, \mathcal{L}) \to (G', \mathcal{L}')$ such that $\mathsf{cost}(G', \mathcal{L}') \leq c \cdot \mathsf{cost}(G, \mathcal{L})$ and $(G', \mathcal{L}')$ is closer to $2d$-Hitting Set.

Progress measure $\mu =$ maximum # non-singletons in a list of $\mathcal{L}$.

Example: $\mathsf{cost}(G, \mathcal{L}) = C$ and $d = 4$.

$$
\begin{array}{ccccccccc}
(G, \mathcal{L}) & \to & (G_1, \mathcal{L}_1) & \to & (G_2, \mathcal{L}_2) & \to & (G_3, \mathcal{L}_3) & \to & (G_4, \mathcal{L}_4) \\
\text{cost } C & \to & \text{cost } 3C & \to & \text{cost } 3^2C & \to & \text{cost } 3^3C & \to & \text{cost } 3^4C \\
\mu = 4 & \to & \mu = 3 & \to & \mu = 2 & \to & \mu = 1 & \to & \mu = 0
\end{array}
$$

Disjunctive Multicut

Main idea: design an algorithm $(G, \mathcal{L}) \to (G', \mathcal{L}')$ such that $\mathsf{cost}(G', \mathcal{L}') \leq c \cdot \mathsf{cost}(G, \mathcal{L})$ and $(G', \mathcal{L}')$ is closer to $2d$-Hitting Set.

Progress measure $\mu =$ maximum # non-singletons in a list of $\mathcal{L}$.

Example: $\mathsf{cost}(G, \mathcal{L}) = C$ and $d = 4$.

$$\begin{array}{ccccccccc}
(G, \mathcal{L}) & \to & (G_1, \mathcal{L}_1) & \to & (G_2, \mathcal{L}_2) & \to & (G_3, \mathcal{L}_3) & \to & (G_4, \mathcal{L}_4) \\
\text{cost } C & \to & \text{cost } 3C & \to & \text{cost } 3^2C & \to & \text{cost } 3^3C & \to & \text{cost } 3^4C \\
\mu = 4 & \to & \mu = 3 & \to & \mu = 2 & \to & \mu = 1 & \to & \mu = 0
\end{array}$$

If $\mathsf{cost}(G, \mathcal{L}) \leq k$, then $\mathsf{cost}(G_4, \mathcal{L}_4) \leq 3^4 k$.
If $\mathsf{cost}(G, \mathcal{L}) > 3^4 k$, then $\mathsf{cost}(G_4, \mathcal{L}_4) > 3^4 k$.

We obtain 3^4-approximation.

1. Iterative compression: X_{in} of size $c_d \cdot k + 1$ that satisfies $\mathcal{L}$.

Disjunctive Multicut: Outline

1. Iterative compression: X_{in} of size $c_d \cdot k + 1$ that satisfies $\mathcal{L}$.
2. By $2^{O(k)}$-branching, assume there is X_{opt} such that $X_{in} \cap X_{opt} = \emptyset$.

1. Iterative compression: X_{in} of size $c_d \cdot k + 1$ that satisfies $\mathcal{L}$.
2. By $2^{O(k)}$-branching, assume there is X_{opt} such that $X_{in} \cap X_{opt} = \emptyset$.
3. At cost $\leq |X_{opt}|$, assume X_{in} has ≤ 1 vertex in every connected component of $G - X_{opt}$.

1. Iterative compression: X_{in} of size $c_d \cdot k + 1$ that satisfies $\mathcal{L}$.
2. By $2^{O(k)}$-branching, assume there is X_{opt} such that $X_{in} \cap X_{opt} = \emptyset$.
3. At cost $\leq |X_{opt}|$, assume X_{in} has ≤ 1 vertex in every connected component of $G - X_{opt}$.
4. Let $\mathcal{T}_{in}$ be the set of cut requests satisfied by X_{in}.

Disjunctive Multicut: Outline

1. Iterative compression: X_{in} of size $c_d \cdot k + 1$ that satisfies $\mathcal{L}$.
2. By $2^{O(k)}$-branching, assume there is X_{opt} such that $X_{in} \cap X_{opt} = \emptyset$.
3. At cost $\leq |X_{opt}|$, assume X_{in} has ≤ 1 vertex in every connected component of $G - X_{opt}$.
4. Let $\mathcal{T}_{in}$ be the set of cut requests satisfied by X_{in}.
5. Let $\mathcal{T}_{opt}$ be the set of cut requests satisfied by X_{opt}.

1. Iterative compression: X_{in} of size $c_d \cdot k + 1$ that satisfies $\mathcal{L}$.
2. By $2^{O(k)}$-branching, assume there is X_{opt} such that $X_{in} \cap X_{opt} = \emptyset$.
3. At cost $\leq |X_{opt}|$, assume X_{in} has ≤ 1 vertex in every connected component of $G - X_{opt}$.
4. Let $\mathcal{T}_{in}$ be the set of cut requests satisfied by X_{in}.
5. Let $\mathcal{T}_{opt}$ be the set of cut requests satisfied by X_{opt}.
6. Let $\mathcal{F}$ be the set of all st-walks for $st \in \mathcal{T}_{in} \cap \mathcal{T}_{out}$.

1. Iterative compression: X_{in} of size $c_d \cdot k + 1$ that satisfies $\mathcal{L}$.
2. By $2^{O(k)}$-branching, assume there is X_{opt} such that $X_{in} \cap X_{opt} = \emptyset$.
3. At cost $\leq |X_{opt}|$, assume X_{in} has ≤ 1 vertex in every connected component of $G - X_{opt}$.
4. Let $\mathcal{T}_{in}$ be the set of cut requests satisfied by X_{in}.
5. Let $\mathcal{T}_{opt}$ be the set of cut requests satisfied by X_{opt}.
6. Let $\mathcal{F}$ be the set of all st-walks for $st \in \mathcal{T}_{in} \cap \mathcal{T}_{out}$.
7. X_{opt} contains an $\mathcal{F}$-transversal Y.

Disjunctive Multicut: Outline

1. Iterative compression: X_{in} of size $c_d \cdot k + 1$ that satisfies $\mathcal{L}$.
2. By $2^{O(k)}$-branching, assume there is X_{opt} such that $X_{in} \cap X_{opt} = \emptyset$.
3. At cost $\leq |X_{opt}|$, assume X_{in} has ≤ 1 vertex in every connected component of $G - X_{opt}$.
4. Let $\mathcal{T}_{in}$ be the set of cut requests satisfied by X_{in}.
5. Let $\mathcal{T}_{opt}$ be the set of cut requests satisfied by X_{opt}.
6. Let $\mathcal{F}$ be the set of all st-walks for $st \in \mathcal{T}_{in} \cap \mathcal{T}_{out}$.
7. X_{opt} contains an $\mathcal{F}$-transversal Y.
8. Replace Y with a union Z of important separators*.
   Note that $Z \cup X_{opt}$ satisfies $\mathcal{L}$ and $|Z \cup X_{opt}| \leq 2|X_{opt}|$.

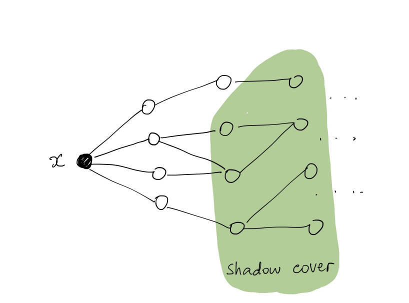
Disjunctive Multicut: Outline

1. Iterative compression: X_{in} of size $c_d \cdot k + 1$ that satisfies $\mathcal{L}$.
2. By $2^{O(k)}$-branching, assume there is X_{opt} such that $X_{in} \cap X_{opt} = \emptyset$.
3. At cost $\leq |X_{opt}|$, assume X_{in} has ≤ 1 vertex in every connected component of $G - X_{opt}$.
4. Let $\mathcal{T}_{in}$ be the set of cut requests satisfied by X_{in}.
5. Let $\mathcal{T}_{opt}$ be the set of cut requests satisfied by X_{opt}.
6. Let $\mathcal{F}$ be the set of all st-walks for $st \in \mathcal{T}_{in} \cap \mathcal{T}_{out}$.
7. X_{opt} contains an $\mathcal{F}$-transversal Y.
8. Replace Y with a union Z of important separators*.
   Note that $Z \cup X_{opt}$ satisfies $\mathcal{L}$ and $|Z \cup X_{opt}| \leq 2|X_{opt}|$.
9. Benefit: decrease μ by removing or replacing every $st \in \mathcal{T}_{in}$ with pairs of singletons (examples coming up).

Disjunctive Multicut: Outline

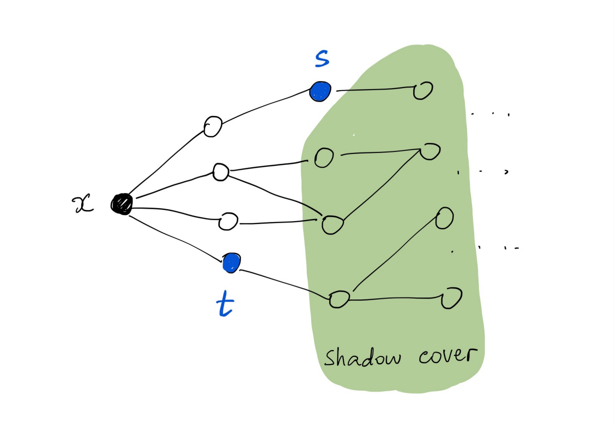
1. Iterative compression: X_{in} of size $c_d \cdot k + 1$ that satisfies $\mathcal{L}$.
2. By $2^{O(k)}$-branching, assume there is X_{opt} such that $X_{in} \cap X_{opt} = \emptyset$.
3. At cost $\leq |X_{opt}|$, assume X_{in} has ≤ 1 vertex in every connected component of $G - X_{opt}$.
4. Let $\mathcal{T}_{in}$ be the set of cut requests satisfied by X_{in}.
5. Let $\mathcal{T}_{opt}$ be the set of cut requests satisfied by X_{opt}.
6. Let $\mathcal{F}$ be the set of all st-walks for $st \in \mathcal{T}_{in} \cap \mathcal{T}_{out}$.
7. X_{opt} contains an $\mathcal{F}$-transversal Y.
8. Replace Y with a union Z of important separators*.
   Note that $Z \cup X_{opt}$ satisfies $\mathcal{L}$ and $|Z \cup X_{opt}| \leq 2|X_{opt}|$.
9. Benefit: decrease μ by removing or replacing every $st \in \mathcal{T}_{in}$ with pairs of singletons (examples coming up).
10. Cost increased by a factor ≤ 3.

DISJUNCTIVE MULTICUT: Shadow Removal

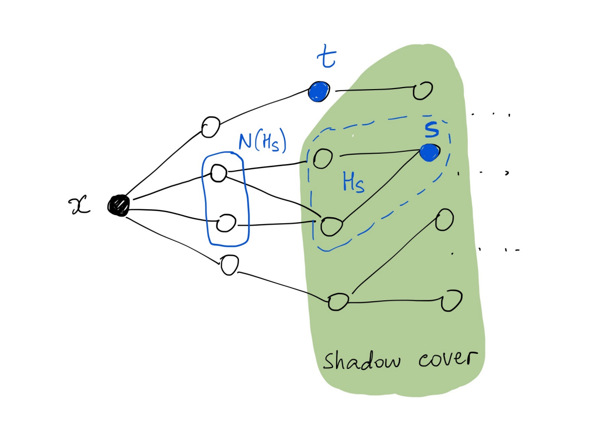


Shadow covering algorithm of [MR'11, CCHM'12] provides $W \subseteq V(G)$ s.t.

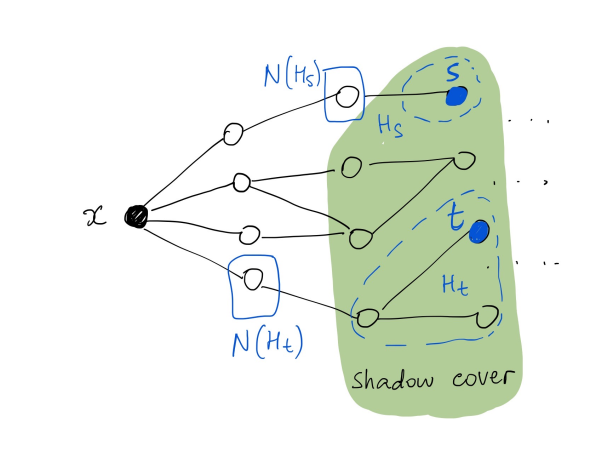
- $Z \subseteq W$, and
- if Z disconnects x from $s \in W$, then $s \in Z$.



Case 1: $L = \{st, \ldots\} \rightarrow \{ss, tt, \ldots\}$.



Case 2: $L = \{st, \ldots\} \rightarrow \{aa, tt, \ldots\}$ for all $a \in N(H_s) \cap W$.



Case 3: $L = \{st, \dots\} \to \{aa, bb, \dots\}$ for all $a \in N(H_s) \cap W$ and $b \in N(H_t) \cap W$

Wrap-up

Theorem

Disjunctive Multicut with lists of length d admits 3^d-approximation in fpt time.

Wrap-up

Theorem

DISJUNCTIVE MULTICUT with lists of length d admits 3^d-approximation in fpt time.

STEINER MULTICUT admits a much simpler $2^{O(k)}$-time 2-approximation algorithm.

Wrap-up

Theorem

Disjunctive Multicut with lists of length d admits 3^d-approximation in fpt time.

Steiner Multicut admits a much simpler $2^{O(k)}$-time 2-approximation algorithm.

We also study MinCSP(Γ^+) for equality constraint languages Γ extended with unit assignments like $x = 1$, $y = 2$, etc.

Wrap-up

Theorem

Disjunctive Multicut with lists of length d admits 3^d-approximation in fpt time.

Steiner Multicut admits a much simpler $2^{O(k)}$-time 2-approximation algorithm.

We also study MinCSP(Γ^+) for equality constraint languages Γ extended with unit assignments like $x = 1$, $y = 2$, etc.
Conclusion: MinCSP(Γ^+) is either trivial, equivalent to MinCSP(Γ), equivalent to MinCSP(B) for a Boolean language B, solved by simple branching, solved by reduction to Multiway Cut, or hard.

Questions?

Classification Theorem

Let Γ be a finite equality constraint language such that CSP(Γ) is in P and MinCSP(Γ) is NP-hard. Then one of the following holds.

- MinCSP(Γ) is in FPT.
- MinCSP(Γ) is W[1]-hard but admits constant-factor FPA.
- MinCSP(Γ) is W[2]-hard and admits no constant-factor FPA.

Questions?