Beyond fractional hypertree width

Dániel Marx

Tel Aviv University, Israel

October 26, 2009

Dagstuhl Seminar 09441

Constraint Satisfaction Problems (CSP)

A CSP instance is given by describing the

- variables,
- domain of the variables,
- constraints on the variables.

Task: Find an assignment that satisfies every constraint.

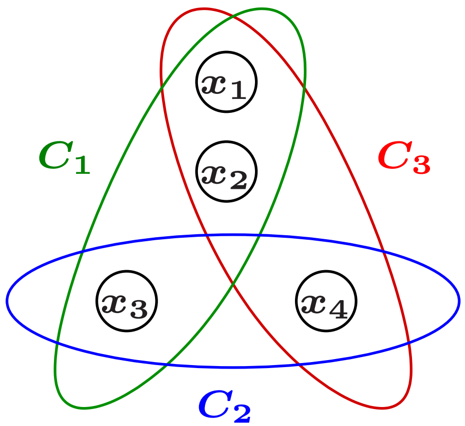
$$I = C_1(x_1, x_2, x_3) \wedge C_2(x_2, x_4) \wedge C_3(x_1, x_3, x_4)$$

In this talk: constraints are represented by listing all the tuples.

Hypergraphs and CSP

Hypergraph: vertices are the variables, constraints are the hyperedges.

$$I = C_1(x_2, x_1, x_3) \wedge C_2(x_4, x_3) \wedge C_3(x_1, x_4, x_2)$$



CSP($\mathcal{H}$): The CSP problem restricted to instances where the hypergraph belongs to the class $\mathcal{H}$.

CSP($\mathcal{H}$) is **polynomial-time solvable** if there is a $O(\|I\|^c)$ time algorithm.

CSP($\mathcal{H}$) is **fixed-parameter tractable (FPT)** if there is a $f(H) \cdot \|I\|^c$ time algorithm.

Main result

Main result: Let $\mathcal{H}$ be a recursively enumerable set of hypergraphs. Assuming ETH,

$$\text{CSP}(\mathcal{H}) \text{ is FPT} \iff \mathcal{H} \text{ has bounded submodular width.}$$

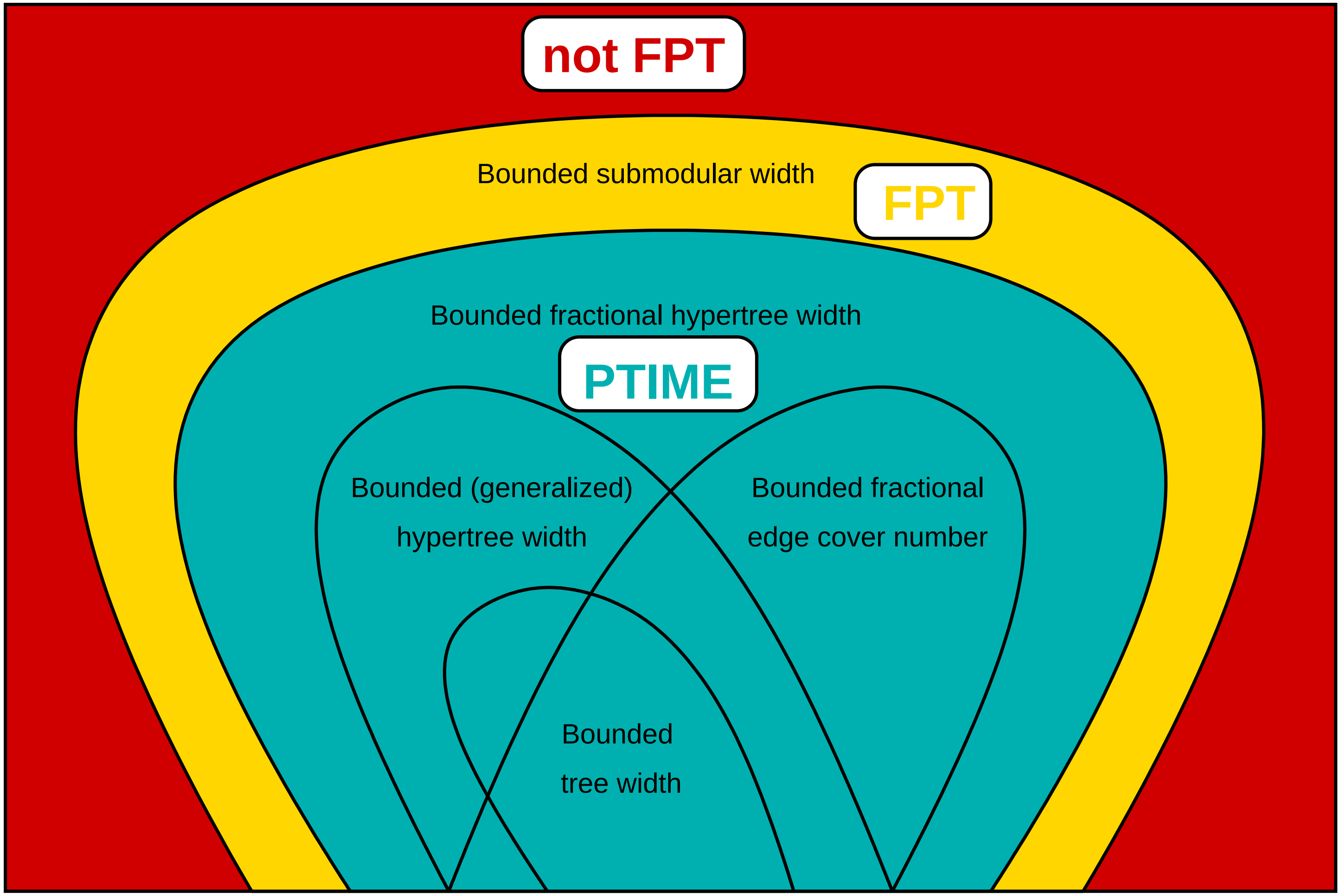
Exponential Time Hypothesis (ETH):

There is no $2^{o(n)}$ time algorithm for n-variable 3SAT.

Known to be equivalent to:

There is no $2^{o(m)}$ time algorithm for m-clause 3SAT.

Tractable classes



Tree decomposition of hypergraphs

Tree decomposition: Bags of vertices are arranged in a tree structure satisfying the following properties:

1. For every hyperedge e, there is a bag containing the vertices of e.
2. For every vertex v, the bags containing v form a connected subtree.

Standard definitions:

Width of the decomposition: size of the largest bag minus 1.

Tree width: width of the best decomposition.

Tree decomposition of hypergraphs

Tree decomposition: Bags of vertices are arranged in a tree structure satisfying the following properties:

1. For every hyperedge e, there is a bag containing the vertices of e.
2. For every vertex v, the bags containing v form a connected subtree.

Standard definitions:

Width of the decomposition: size of the largest bag minus 1.

Tree width: width of the best decomposition.

Let us introduce a more general framework that includes treewidth and many of its generalizations.

Width measures for decompositions

Definition: Let $f : 2^{V(H)} \longrightarrow \mathbb{R}^+$ be a function assigning values to the vertex subsets of H.

- The f-width($\mathcal{T}$) of a tree decomposition $\mathcal{T}$ is the maximum of $f(B)$ over all bags B.
- The f-width(H) of hypergraph H is the minimum of f-width($\mathcal{T}$) over all tree decompositions $\mathcal{T}$ of H.

Width measures for decompositions

Definition: Let $f : 2^{V(H)} \to \mathbb{R}^+$ be a function assigning values to the vertex subsets of H.

- The f-width($\mathcal{T}$) of a tree decomposition $\mathcal{T}$ is the maximum of $f(B)$ over all bags B.
- The f-width(H) of hypergraph H is the minimum of f-width($\mathcal{T}$) over all tree decompositions $\mathcal{T}$ of H.

Example: If $s(B) = |B| - 1$, then s-width(H) is treewidth.

Example: If $\varrho_H(B)$ is the edge cover number of B, then ϱ_H-width(H) is generalized hypertree width.

Example: If $\varrho^*_H(B)$ is the fractional edge cover number of B, then ϱ^*_H-width(H) is fractional hypertree width.

Note: $\varrho^*_H(B) \leq \varrho_H(B) \leq s(B) + 1$

Useful width measures

Definition: sol(B): number of solutions in the instance projected to B.

We say that f-width is **useful** if in every instance I and for every subset B, sol(B) is at most $\|I\|^{O(f(B))}$.

Note: treewidth, hypertree width, fractional hypertree width are all useful.

Recall:

Fact: If we are given a tree decomposition of the primal graph of instance I such that sol$(B) \leq C$ for any bag B, then I can be solved in time polynomial in $\|I\|$ and C.

Thus if f-width is useful, then bounded f-width implies polynomial-time solvability if a decomposition is available.

$\Rightarrow$ This immediately implies fixed-parameter tractability.

Going beyond fractional hypertree width

To go beyond fractional hypertree width it is sufficient to identify a function $f(B) \leq \varrho^*_H(B)$ such that f-width is useful.

Going beyond fractional hypertree width

To go beyond fractional hypertree width it is sufficient to identify a function $f(B) \le \varrho^*_H(B)$ such that f-width is useful.

Unfortunately, there is no such function f:

Fact: There are arbitrarily large instances I with hypergraph H where the projection to B has $\|I\|^{\Omega(\varrho^*_H(B))}$ solutions.

Thus if f-width is useful, it cannot be less than fractional hypertree width.

$\mathcal{F}$-width

Definition: Let $\mathcal{F}$ be a set of functions from $2^{V(H)}$ to $\mathbb{R}^+$. The $\mathcal{F}$-width of H is the maximum of f-width(H) over every $f \in \mathcal{F}$.

$\mathcal{F}$-width$(H) \leq w \iff$ **for every** $f \in \mathcal{F}$ **exists** a tree decomposition $\mathcal{T}$ of $\mathcal{F}$ such that **for every** bag B of $\mathcal{T}$, $f(B) \leq w$.

Note: the tree decomposition $\mathcal{T}$ can be different for different functions $f \in \mathcal{F}$.

Submodular width

Definition: The **submodular width** of H is $\mathcal{F}$-width(H), where $\mathcal{F}$ is the set of all monotone, edge-dominated, submodular functions on the vertices of H.

Monotone: $b(X) \le b(Y)$ for every $X \subseteq Y$.

Edge-dominated: $b(e) \le 1$ for every hyperedge e of H.

Submodular: For arbitrary sets X, Y

$$b(X) + b(Y) \ge b(X \cap Y) + b(X \cup Y).$$

Main result

Main result: Let $\mathcal{H}$ be a recursively enumerable set of hypergraphs. Assuming ETH,

$$\text{CSP}(\mathcal{H}) \text{ is FPT} \iff \mathcal{H} \text{ has bounded submodular width.}$$

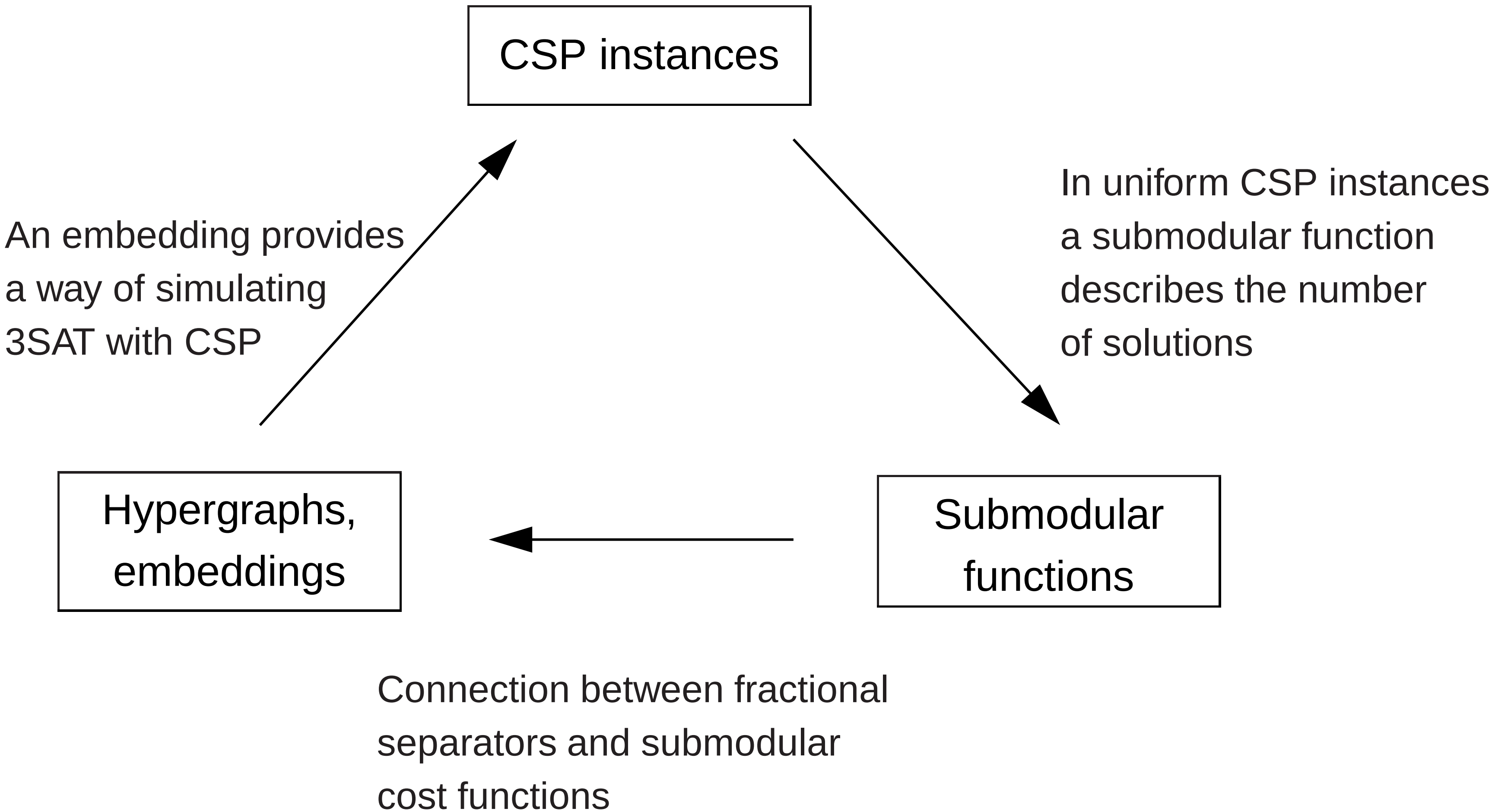
- **Algorithmic side:** If $\mathcal{H}$ has bounded submodular width, then $\text{CSP}(\mathcal{H})$ is FPT.

  How does it help if we know that every submodular function has a good tree decomposition?

- **Hardness:** If $\mathcal{H}$ has bounded submodular width, then $\text{CSP}(\mathcal{H})$ is not FPT.

  To simulate 3SAT by $\text{CSP}(\mathcal{H})$, we need an efficient embedding of a graph into a hypergraph. We know that certain submodular functions do not have good tree decompositions. How does that help in finding good embeddings?

Three battlefields



A crazy idea

Let I be a CSP instance with hypergraph H. Let $N := \|I\|$ and suppose that the submodular width of H is at most w.

Let $b(B) := \log_N \text{sol}(B)$, which is edge-dominated.

Crazy assumption: b is monotone and submodular.

Then by the definition of submodular width, there is a tree decomposition where sol$(B) \leq N^w$ for every bag
$\Rightarrow$ FPT algorithm!

A crazy idea

Let I be a CSP instance with hypergraph H. Let $N := \|I\|$ and suppose that the submodular width of H is at most w.

Let $b(B) := \log_N \text{sol}(B)$, which is edge-dominated.

Crazy assumption: b is monotone and submodular.

Then by the definition of submodular width, there is a tree decomposition where $\text{sol}(B) \le N^w$ for every bag
$\Rightarrow$ FPT algorithm!

Problems:

- b is not necessarily monotone.
- b is not necessarily submodular.
- we don't even know the function b.

Small sets

Let X be M**-small** if sol$(Y) \leq M$ for every $Y \subseteq X$.

Fact: In time $f(H) \cdot (\|I\| \cdot M)^{O(1)}$, we can identify all M-small sets and compute sol(X) for every such set X.

We will care about the value of b only on the N^w-small sets, every other set will be "too large."

By introducing further constraints, we can ensure that b is monotone on N^w-small sets: if an assignment $Y \subset X$ is not extendible to X, then we forbid it.

Now we know b and it is monotone on the sets we care about. But what about submodularity?

Uniformity

Definition: Instance I is c-uniform, if for every $B \subseteq A$, every satisfying assignment of B has at most $c \cdot \text{sol}(A)/\text{sol}(B)$ extensions to a satisfying assignment of A.

Fact: If I is 1-uniform, then b is submodular.

$$b(X) + b(Y) \geq b(X \cap Y) + b(X \cup Y).$$

Uniformity

Definition: Instance I is c-uniform, if for every $B \subseteq A$, every satisfying assignment of B has at most $c \cdot \text{sol}(A)/\text{sol}(B)$ extensions to a satisfying assignment of A.

Fact: If I is 1-uniform, then b is submodular.

$$b(X) + b(Y) \geq b(X \cap Y) + b(X \cup Y).$$

If I is N^{ϵ}-uniform on N^{w}-small sets, then with some tweaking (adding low order terms) we can make b submodular.

But why would be the instance N^{ϵ}-uniform?

Decomposition into uniform instances

Suppose that two N^w-small sets $B \subseteq A$ violate N^ϵ-uniformity: there are assignments on B having more than $N^\epsilon \cdot \text{sol}(A)/\text{sol}(B)$ extensions.

By adding a new constraint, we split the instance in two cases:

- in I_{small} every assignment on B has at most $\sqrt{N^\epsilon} \cdot \text{sol}(A)/\text{sol}(B)$ extensions,
- in I_{large} every assignment has more than that many extensions.

Repeat if necessary.

We can show that the number of instances created by the procedure can be bounded by a function of the number of variables (independent of the size of the domain and the relations!).

The algorithm

Algorithm for hypergraphs with submodular with at most w:

- Locate the N^w-small sets.
- Decompose the instance into a bounded number of N^ϵ-uniform instances
  $\Rightarrow b = \log_N \mathrm{sol}(B)$ is submodular (after some tweaking).
- For each new instance, try every tree decomposition — there has to be one where $b(B) \leq w$ and hence $\mathrm{sol}(B) \leq N^w$ for every bag b.
- Solve the new instance using this tree decomposition.

This completes the algorithmic part of the main result.

The algorithm

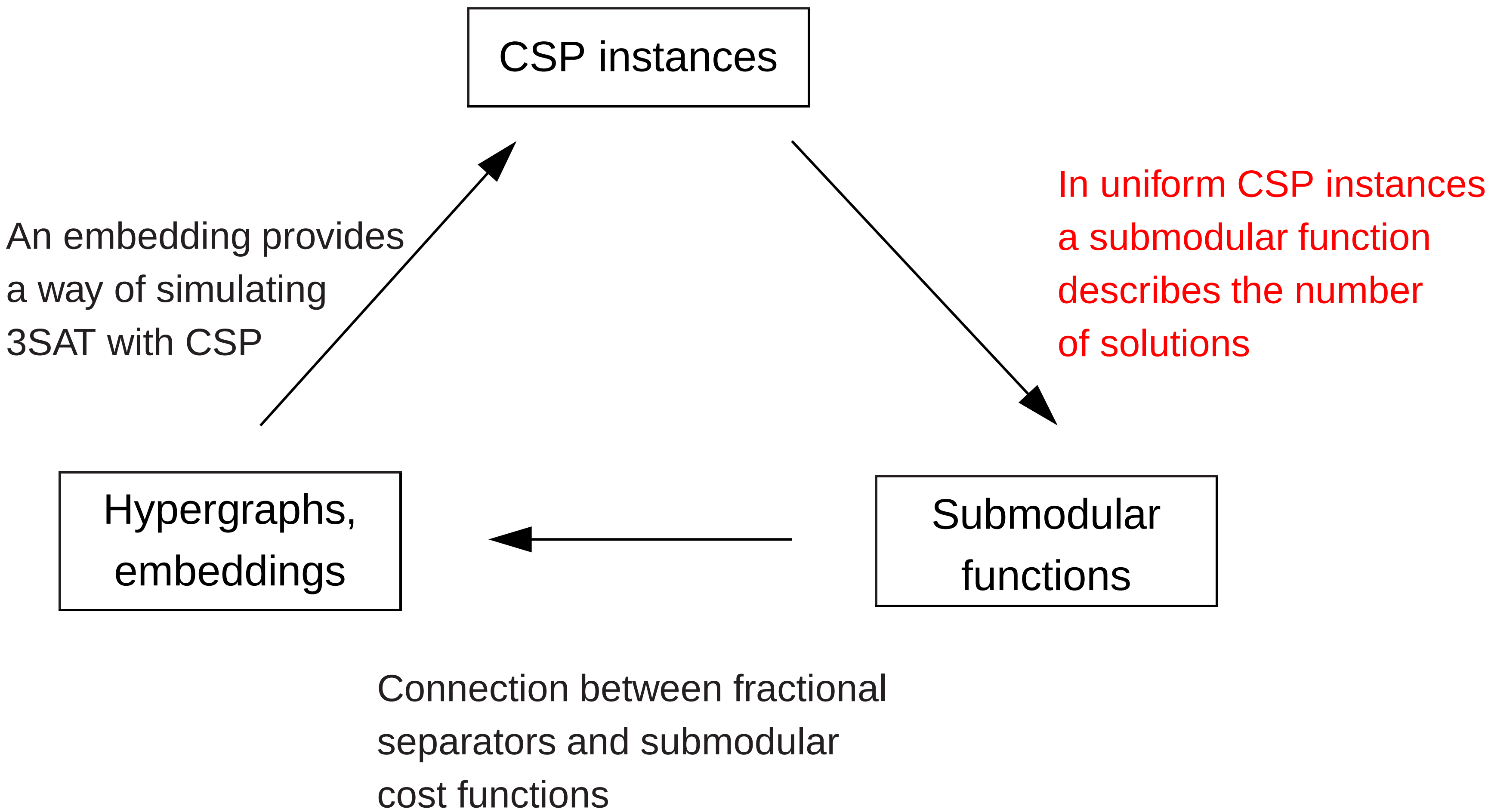
Algorithm for hypergraphs with submodular with at most w:

- Locate the N^w-small sets.
- Decompose the instance into a bounded number of N^ϵ-uniform instances $\Rightarrow b = \log_N \text{sol}(B)$ is submodular (after some tweaking).
- For each new instance, try every tree decomposition — there has to be one where $b(B) \leq w$ and hence sol$(B) \leq N^w$ for every bag b.
- Solve the new instance using this tree decomposition.

This completes the algorithmic part of the main result.

- **Idea #1:** The decomposition depends not only on the hypergraph of the instance, but on the actual constraint relations.
- **Idea #2:** We branch on adding further restrictions, and apply different tree decompositions to each resulting instance.

Three battlefields



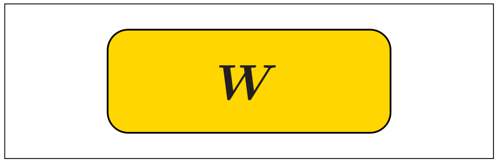
Highly connected sets

For the hardness part, we want to characterize submodular width analogously to other width measures: if submodular width is large, then there is a "large highly connected set" in the hypergraph.

Highly connected sets

For the hardness part, we want to characterize submodular width analogously to other width measures: if submodular width is large, then there is a “large highly connected set” in the hypergraph.

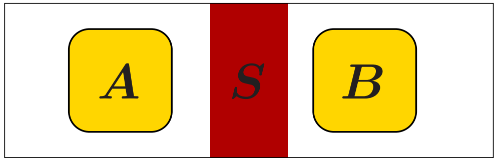
Separator-based approach:



Highly connected sets

For the hardness part, we want to characterize submodular width analogously to other width measures: if submodular width is large, then there is a "large highly connected set" in the hypergraph.

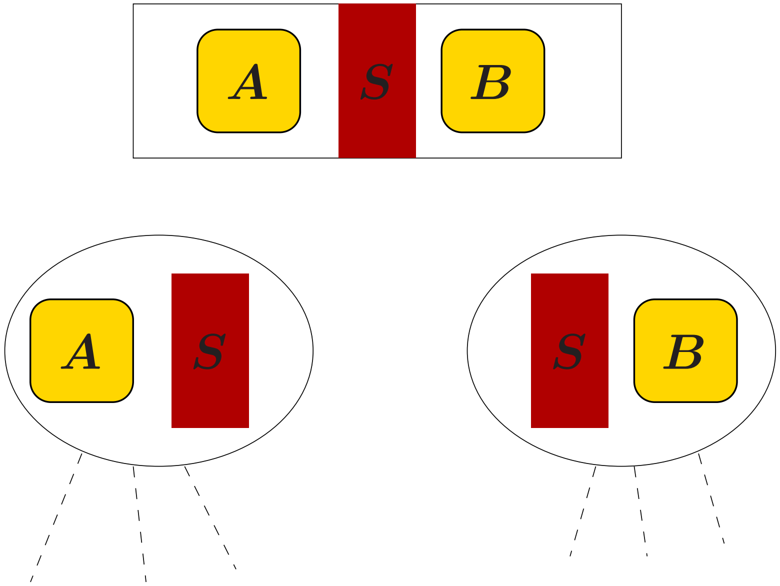
Separator-based approach:



Highly connected sets

For the hardness part, we want to characterize submodular width analogously to other width measures: if submodular width is large, then there is a “large highly connected set” in the hypergraph.

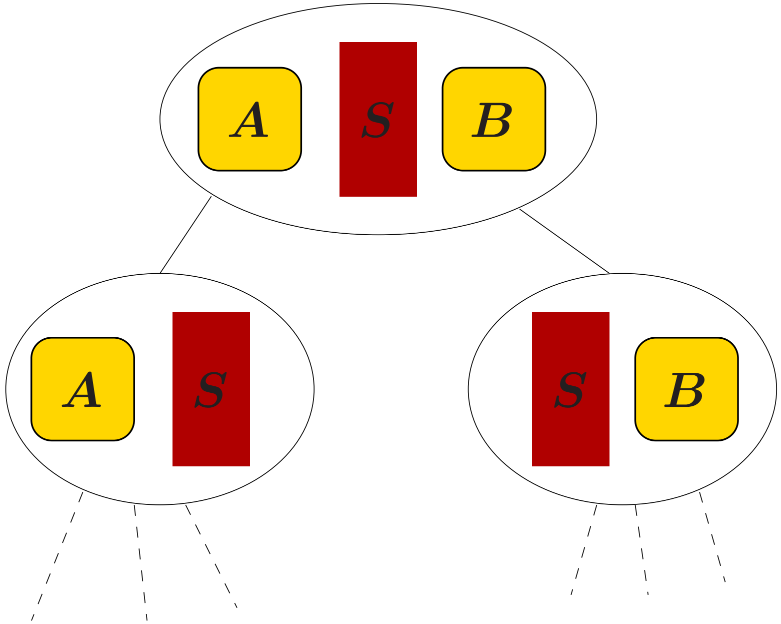
Separator-based approach:



Highly connected sets

For the hardness part, we want to characterize submodular width analogously to other width measures: if submodular width is large, then there is a "large highly connected set" in the hypergraph.

Separator-based approach:



Highly connected set

Definition: A set W is b-connected, if for every disjoint $X, Y \subseteq W$, there is no (X, Y)-separator S with $b(S) < \min\{b(X), b(Y)\}$.

If b-width is at least w for some submodular function b, the separator-based approach of finding tree decompositions **almost** gives us (there is one major technical difficulty) a set b-connected set W with $b(W) = \Omega(w)$.

But we want a notion of highly connected set that is determined only by the hypergraph H and is not related to any submodular function.

Highly connected set

Definition: A fractional independent set of H is an assignment $\mu : V(H) \to \{0,1\}$ such that $\mu(e) \le 1$ for every hyperedge e (we define $\mu(X) = \sum_{v \in X} \mu(v)$).

Definition: A fractional (X,Y)-separator is an assignment $E(H) \to \{0,1\}$ such that every $X - Y$ path is covered by total weight at least 1.

Definition: Let $\lambda > 0$ be a constant (say, 0.01) and let μ be a fractional independent set. A set W is (μ,λ)**-connected** if for every disjoint $X, Y \subseteq W$, there is no fractional (X,Y)-separator of weight less than $\lambda \cdot \min\{\mu(X), \mu(Y)\}$.

We need to connect somehow the notions of “fractional (X,Y)-separator having small weight” and “(X,Y)-separator S with $b(S)$ small.”

Result on separation

What does it mean that there is a fractional (X, Y)-separator of small weight?

Fact: If there is a fractional (X, Y)-separator of weight w, then for every edge-dominated monotone submodular function b, there is a (X, Y)-separator S with $b(S) = O(w)$.

Result on separation

What does it mean that there is a fractional (X, Y)-separator of small weight?

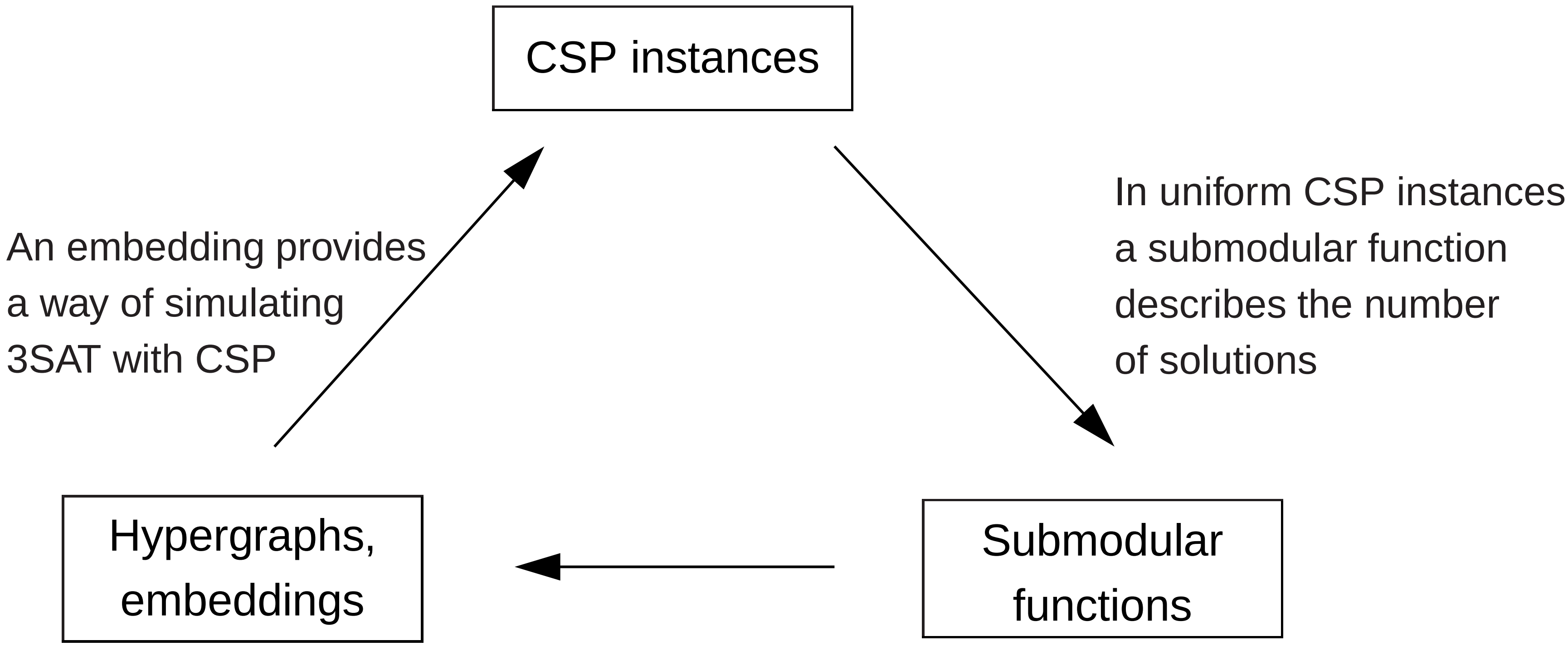
Fact: If there is a fractional (X, Y)-separator of weight w, then for every edge-dominated monotone submodular function b, there is a (X, Y)-separator S with $b(S) = O(w)$.

Definition: (repeated) A set W is b-connected, if for every disjoint $X, Y \subseteq W$, there is no (X, Y)-separator S with $b(S) < \min\{b(X), b(Y)\}$.

If there is no (X, Y)-separator S with $b(S) < \min\{b(X), b(Y)\}$, then there is no fractional separator of weight $\lambda \cdot \min\{b(X), b(Y)\}$ for some $\lambda > 0$.

So we have obtained a set W that is "highly connected" in the sense that certain fractional separators do not exist, and this takes us into the domain of purely hypergraph properties, separators, flows, etc.

Three battlefields



Connection between fractional
separators and submodular
cost functions

Embeddings

Definition: A q**-embedding** of graph F in hypergraph H maps a subset of $V(H)$ to each vertex of H such that

- For every $v \in V(F)$, $\phi(v)$ is connected.
- If $u, v \in V(F)$ are adjacent in F, then $\phi(u)$ and $\phi(v)$ touch: there is a hyperedge intersecting both of them
- Every hyperedge e of H intersects the images of at most q vertices of F.

Fact: For **graphs** F and G, if $m = |E(F)|$ is sufficiently large and $k = \mathrm{tw}(G)$, then there is a q-embedding of F in G for $q = O(m \log k / k)$.

Embeddings

Definition: A q-**embedding** of graph F in hypergraph H maps a subset of $V(H)$ to each vertex of H such that

- For every $v \in V(F)$, $\phi(v)$ is connected.
- If $u, v \in V(F)$ are adjacent in F, then $\phi(u)$ and $\phi(v)$ touch: there is a hyperedge intersecting both of them
- Every hyperedge e of H intersects the images of at most q vertices of F.

Fact: For **graphs** F and G, if $m = |E(F)|$ is sufficiently large and $k = \text{tw}(G)$, then there is a q-embedding of F in G for $q = O(m \log k/k)$.

We show:

Fact: For a graph F and hypergraph H, if $m = |E(F)|$ is sufficiently large and H has submodular width w, then there is a q-embedding of F in H for $q = O(m/w^{\frac{1}{4}})$.

Combinatorial optimization techniques, linear programming, etc..

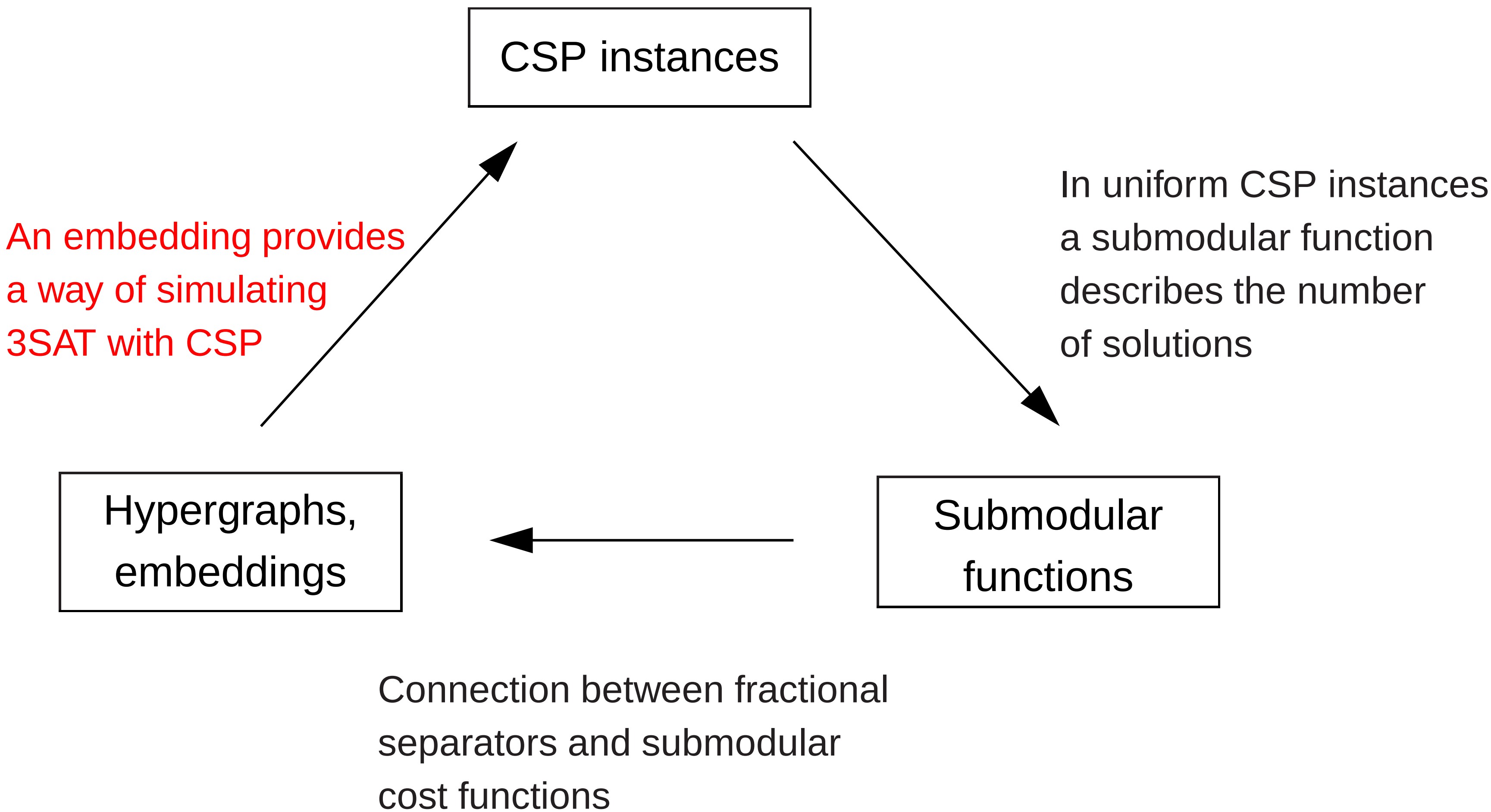
Hardness proof

Fact: If $\mathcal{H}$ is a recursively enumerable class of hypergraphs with unbounded submodular width, then CSP$(\mathcal{H})$ is not fixed-parameter tractable (assuming ETH).

Proof outline:

- Given a 3SAT instance with m clauses and n variables, we turn it into a CSP instance I_1 with $3m$ binary constraints, and domain size 3.
- We use the embedding result to find a q-embedding of the primal graph F of I_1 into some $H_k \in \mathcal{H}$ (chosen appropriately).
- We simulate I_1 by an instance I_2 whose primal graph is H_k: each edge of I_2 "sees" at most q variables of I_1, thus each constraint relation has size $\leq 3^q$.
- Now the 3SAT problem can be solved by solving I_2. Calculation of the running time shows that that an FPT algorithm for CSP$(\mathcal{H})$ would give a $2^{o(m)}$ algorithm for m-clause 3SAT, violating ETH.

Three battlefields



Conclusions

- Characterization of CSP$(\mathcal{H})$ with respect to fixed-parameter tractability.
- Main new definition: submodular width.
- Why fixed-parameter tractability?
- What happens in the "gray zone"?

Tractable classes