Complexity and Approximability for Parameterized Constraint Satisfaction Problems

Holger Dell, Eunjung Kim, Michael Lampis, Valia Mitsou, Tobias Mömke

Institute for Computer Science and Control,
Hungarian Academy of Sciences

GROW 2015, Aussois (France)

Definition

(Boolean) Constraint Satisfaction Problem (CSP):

- A set of variables X over $\{0,1\}$.
- A set of constraints ϕ involving positive and negative appearances of the variables (literals).

Definition

(Boolean) Constraint Satisfaction Problem (CSP):

- A set of variables X over $\{0, 1\}$.
- A set of constraints ϕ involving positive and negative appearances of the variables (literals).

Decide whether there exists an assignment on X that satisfies:

- (AllCSP) all the constraints in ϕ;
- (MaxCSP) at least k constraints in ϕ (given some value k).

Definition

(Boolean) Constraint Satisfaction Problem (CSP):

- A set of variables X over $\{0, 1\}$.
- A set of constraints ϕ involving positive and negative appearances of the variables (literals).

Decide whether there exists an assignment on X that satisfies:

- (AllCSP) all the constraints in ϕ;
- (MaxCSP) at least k constraints in ϕ (given some value k).

Example: CNFSat is a CSP.

$\phi = (\neg x \vee z) \wedge (x \vee y \vee \neg w) \wedge (\neg z \vee w)$

$X = \{x, y, z, w\}$.

CSP is hard!

- Almost all interesting (Max)CSP are NP-hard [Schaefer 1978] and APX-hard [Creignou 1995],[Khanna et al. 2001].

CSP is hard!

- Almost all interesting (Max)CSP are NP-hard [Schaefer 1978] and APX-hard [Creignou 1995],[Khanna et al. 2001].
- Study special cases (parameterized complexity!)

CSP is hard!

- Almost all interesting (Max)CSP are NP-hard [Schaefer 1978] and APX-hard [Creignou 1995],[Khanna et al. 2001].
- Study special cases (parameterized complexity!)
- [Samer, Szeider 2010]: Classification of an extensive list of parameters including # vars, # cons, max constraint size (arity), max # of var occurence . . .

CSP is hard!

- Almost all interesting (Max)CSP are NP-hard [Schaefer 1978] and APX-hard [Creignou 1995],[Khanna et al. 2001].
- Study special cases (parameterized complexity!)
- [Samer, Szeider 2010]: Classification of an extensive list of parameters including # vars, # cons, max constraint size (arity), max # of var occurence ...
- ...Structural parameterizations

Structural CSP

- Define some graph structure of ϕ.
- Study CSPs for special graph classes.

Structural CSP

- Define some graph structure of ϕ.

Structural CSP

Incidence graph representation of a CSP

- (Unsigned) variables and constraints are represented by vertices;
- a constraint vertex is connected to a variable vertex iff the corresponding constraint involves the corresponding variable.

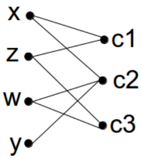


Figure: The incidence graph representation of the previous formula $(\neg x \vee z) \wedge (x \vee y \vee \neg w) \wedge (\neg z \vee w)$.

Structural CSP

- Study CSPs for special graph classes.

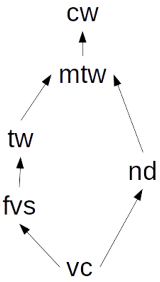
Example classes (CNFSAT)

- Low degree: Bounding degree of incidence graph doesn't help (3CNFSAT where every variable appears at most 3 times is NP-complete).

Example classes (CNFSAT)

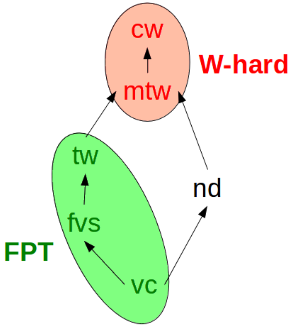
- Low degree: Bounding degree of incidence graph doesn't help (3CNFSAT where every variable appears at most 3 times is NP-complete).
- Acyclicity: Start from the leaves and work your way up (poly-time).

Structural Parameterizations



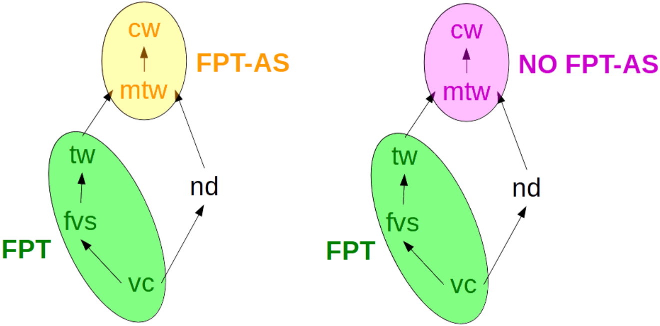
Parameter map: $q \leftarrow p$ (which reads 'q *dominates* p') between two parameters means that q is bounded when p is bounded.

Structural Parameterizations

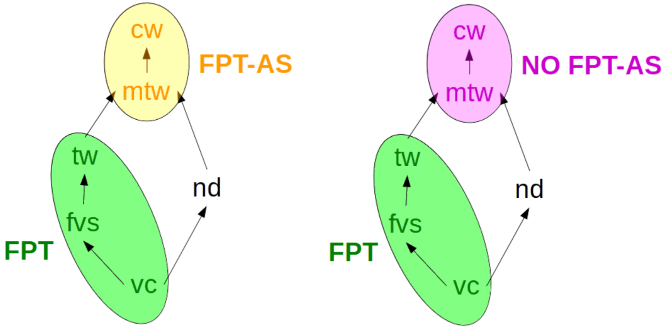


Goal: design algorithms for most dominant parameter (hold downward) and hardness for least dominant (hold upward).

Structural Parameterizations



New approach: study FPT approximations to evade hardness. In this talk we examine the existence of FPT Approximation Schemes.



Definition

FPT Approximation Scheme (FPT-AS): $\forall \epsilon > 0$ there is an $(1-\epsilon)$-approximation algorithm running in time $O(f(\epsilon, k) \cdot \text{poly}(n))$.

Constraints Studied

We explore four natural, well-studied boolean functions which exhibit wildly different behaviors:

1. *or* constraints;

An assignment satisfies such a constraint if at least one literal is made true.

Constraints Studied

We explore four natural, well-studied boolean functions which exhibit wildly different behaviors:

1. *or* constraints;
2. *and* constraints;

An assignment satisfies such a constraint if all literals are made true.

Constraints Studied

We explore four natural, well-studied boolean functions which exhibit wildly different behaviors:

1. *or* constraints;
2. *and* constraints;
3. *parity* constraints;

An assignment satisfies such a constraint if the sum of the literals is odd - or even.

Constraints Studied

We explore four natural, well-studied boolean functions which exhibit wildly different behaviors:

1. *or* constraints;
2. *and* constraints;
3. *parity* constraints;
4. *majority* constraints.

An assignment satisfies such a constraint if at least half of the literals are made true.

Overview of Results

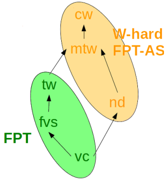


Figure: Diagram for CNFSAT and MAXCNFSAT.

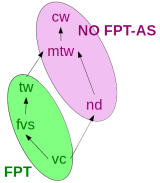


Figure: Diagram for MAXDNFSAT.

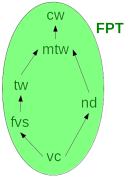


Figure: Diagram for MAXPARITY.

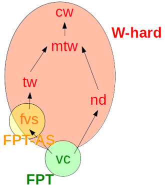
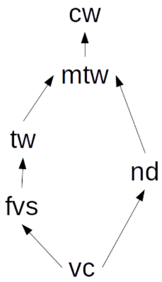
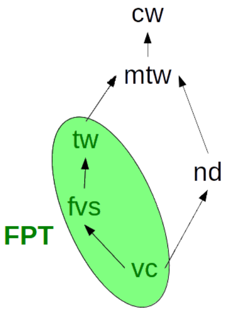


Figure: Diagram for MAJORITY and MAXMAJORITY.


cw
mtw
tw
nd
fvs
vc

Theorem [Szeider 2004]

MaxCNFSat parameterized by incidence treewidth (tw^*) is FPT.

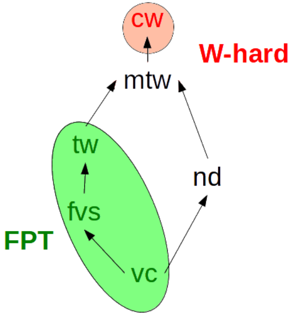


Theorem [Szeider 2004]

MAXCNFSAT parameterized by incidence treewidth (tw^*) is FPT.

Theorem [Ordyniak, Paulusma, Szeider 2013]

CNFSAT parameterized by cw^* is W[1]-hard.



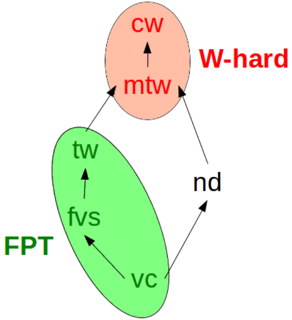
Theorem [Szeider 2004]

MAXCNFSAT parameterized by incidence treewidth (tw^*) is FPT.

Theorem [Ordyniak, Paulusma, Szeider 2013]

CNFSAT parameterized by cw^* is W[1]-hard.

Hardness even holds for a more restricted parameter *modular treewidth* [Paulusma, Slivovsky, Szeider 2013].



CNFSAT and MAXCNFSAT

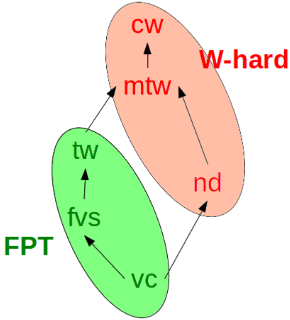
Theorem [Szeider 2004]

MAXCNFSAT parameterized by incidence treewidth (tw^*) is FPT.

Theorem [Ordyniak, Paulusma, Szeider 2013]

CNFSAT parameterized by cw^* is W[1]-hard.

Hardness even holds for a more restricted parameter *modular treewidth* [Paulusma, Slivovsky, Szeider 2013].
$\rightarrow$ **We extend W[1]-hardness to incidence neighborhood diversity (nd^*).**



Theorem [Szeider 2004]

MaxCNFSat parameterized by incidence treewidth (tw^*) is FPT.

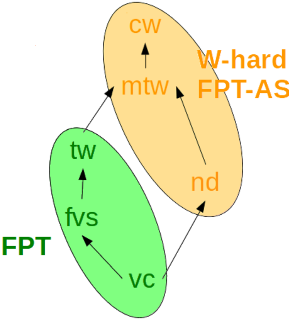
Theorem [Ordyniak, Paulusma, Szeider 2013]

CNFSat parameterized by cw^* is W[1]-hard.

Hardness even holds for a more restricted parameter *modular treewidth* [Paulusma, Slivovsky, Szeider 2013].

$\rightarrow$ **We extend W[1]-hardness to incidence neighborhood diversity (nd^*).**

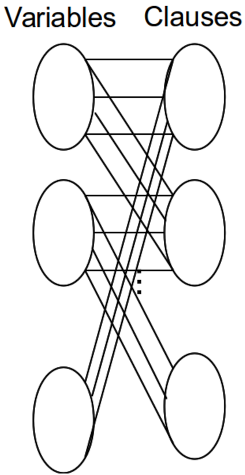
$\rightarrow$ **We also present an FPT-AS for cw^*.**



Neighborhood diversity

Definition [Lampis 2010]

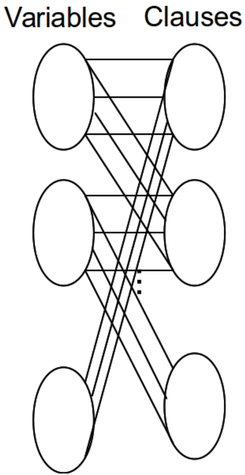
A graph has neighborhood diversity k if its vertices can be separated into k independent sets or cliques where any two vertices in a set have common neighborhood.



Formula ϕ has $\mathrm{nd}^*(G_\phi) \leq k$

There are at most k blocks of variables and clauses, where:

- variables in the same block belong in the same clauses;
- clauses in the same block involve the same variables.

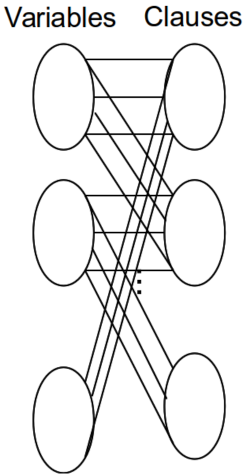


Neighborhood diversity

Formula ϕ has $\mathrm{nd}^*(G_\phi) \leq k$

Positive and negative appearances of variables can:

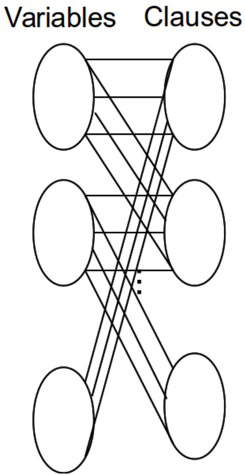
- produce exponentially many clauses;
- encode exponentially many items in binary.



Neighborhood diversity

Theorem

CNFSat *parameterized by the incidence neighborhood diversity* $nd^*(G_\phi)$ *is W[1]-hard.*



On the positive side...

Theorem

MaxCNFSat *parameterized by* cw^* *admits an FPT-AS.*

On the positive side...

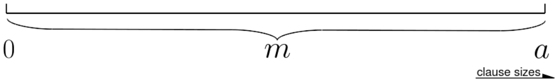
Theorem

MAXCNFSAT *parameterized by* cw^* *admits an FPT-AS.*

Reminder

FPT-AS (FPT Approximation Scheme) for a maximization problem parameterized by k: $\forall \epsilon > 0$ there exists an $(1 - \epsilon)$-approximation algorithm running in $O(f(\epsilon, k) \cdot \text{poly}(n))$.

An FPT-AS for MAXCNFSAT parameterized by cw*

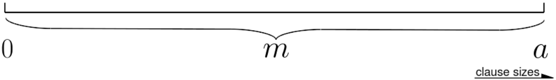


Arrange the clauses in increasing order of arity (0 to a).

An FPT-AS for MaxCNFSat parameterized by cw*

B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

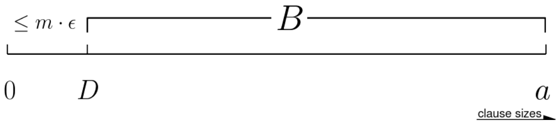
$D = d \cdot \epsilon^4$



Arrange the clauses in increasing order of arity (0 to a).
Split them into big (arity at least $g(\epsilon)$), small (arity at most $g'(\epsilon)$, and medium.
Consider the following cases:

B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

$D = d \cdot \epsilon^4$

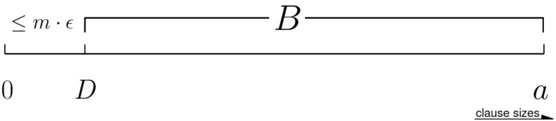


(Almost) all clauses are big:

An FPT-AS for MaxCNFSat parameterized by cw*

B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

$D = d \cdot \epsilon^4$

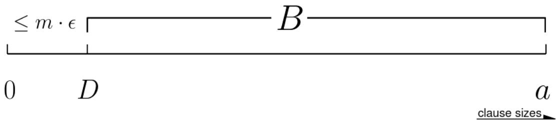


(Almost) all clauses are big:

- ignore small clauses;

B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

$D = d \cdot \epsilon^4$

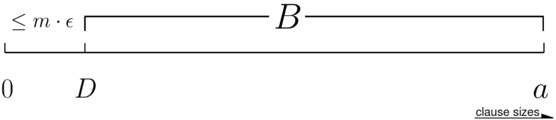


(Almost) all clauses are big:

- ignore small clauses;
- a random assignment satisfies $\geq (1 - \epsilon)(1 - 2^{-g(\epsilon)}) \cdot m$ clauses (with high probability).

B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

$D = d \cdot \epsilon^4$

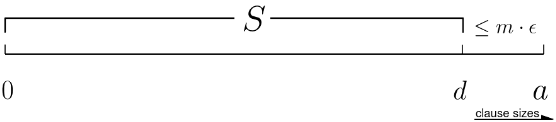


(Almost) all clauses are big:

- ignore small clauses;
- a random assignment satisfies $\geq (1-\epsilon)(1-2^{-g(\epsilon)}) \cdot m$ clauses (with high probability).
- Since $m \geq OPT$, $SOL \geq (1-\epsilon')OPT$, for some ϵ' depending on ϵ.

B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

$D = d \cdot \epsilon^4$

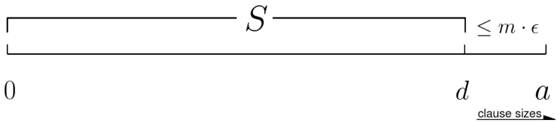


(Almost) all clauses are small:

An FPT-AS for MaxCNFSat parameterized by cw*

B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

$D = d \cdot \epsilon^4$

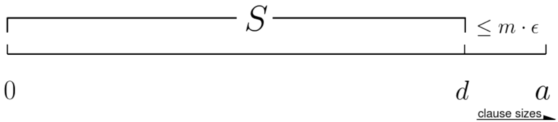


(Almost) all clauses are small:

- ignore large clauses;

B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

$D = d \cdot \epsilon^4$

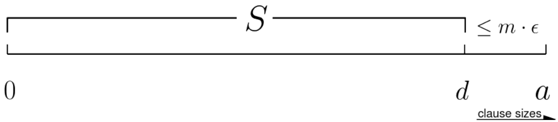


(Almost) all clauses are small:

- ignore large clauses;
- degree on one side of the incidence graph is bounded
  $\rightarrow$ no large biclique subgraphs;

B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

$D = d \cdot \epsilon^4$

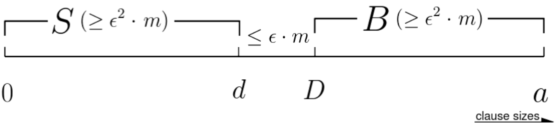


(Almost) all clauses are small:

- ignore large clauses;
- degree on one side of the incidence graph is bounded $\rightarrow$ no large biclique subgraphs;
- By [Gurski, Wanke 2000], the incidence graph has bounded treewidth $\rightarrow$ solve optimally the remaining small clauses;

B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

$D = d \cdot \epsilon^4$

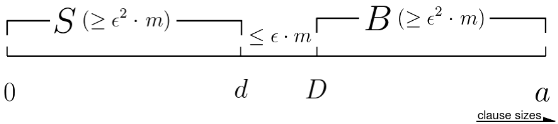


(Almost) no medium-size clauses and B, S are balanced:

An FPT-AS for MaxCNFSat parameterized by cw*

B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

$D = d \cdot \epsilon^4$



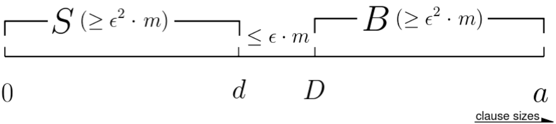
(Almost) no medium-size clauses and B, S are balanced:

- variable occurences$(B) \geq |B| \cdot D = \frac{m \cdot d}{\epsilon^2}$;
- variable occurences$(S) \leq |S| \cdot d \leq m \cdot d$.

An FPT-AS for MaxCNFSat parameterized by cw*

B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

$D = d \cdot \epsilon^4$



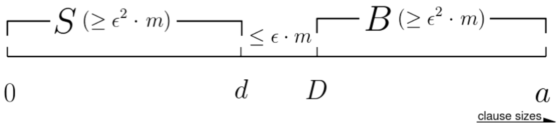
(Almost) no medium-size clauses and B, S are balanced:

- variable occurences$(B) \geq |B| \cdot D = \frac{m \cdot d}{\epsilon^2}$;
- variable occurences$(S) \leq |S| \cdot d \leq m \cdot d$.

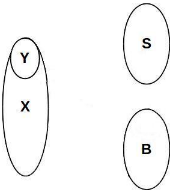
$\rightarrow \exists y \in V$ that appears $1/\epsilon^2$ more times in B than in S.

B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

$D = d \cdot \epsilon^4$

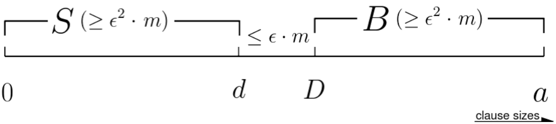


(Almost) no medium-size clauses and B, S are balanced:
From the previous observation, we iteratively create a set of variables Y with the following properties:



B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

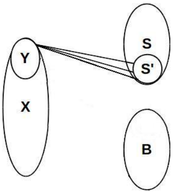
$D = d \cdot \epsilon^4$



(Almost) no medium-size clauses and B, S are balanced:

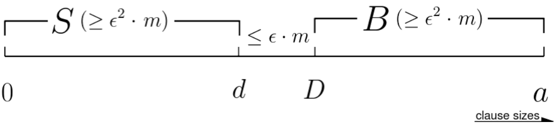
From the previous observation, we iteratively create a set of variables Y with the following properties:

- Y *hits* few clauses of S (call this set S');



B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

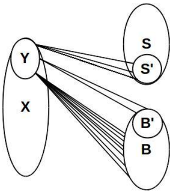
$D = d \cdot \epsilon^4$



(Almost) no medium-size clauses and B, S are balanced:

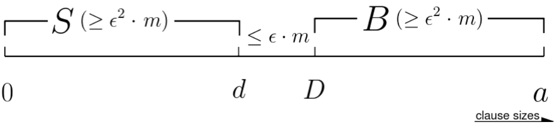
From the previous observation, we iteratively create a set of variables Y with the following properties:

- Y *hits* few clauses of S (call this set S');
- at most ϵ^2 clauses of B have $\leq {}^1/_\epsilon$ neighbors in Y (call this set B').



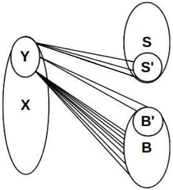
B: Clauses of arity $\geq D = g(\epsilon)$.
S: Clauses of arity $\leq d = g'(\epsilon)$.

$D = d \cdot \epsilon^4$



(Almost) no medium-size clauses and B, S are balanced:
From the previous observation, we iteratively create a set of variables Y with the following properties:

- Y *hits* few clauses of S (call this set S');
- at most ϵ^2 clauses of B have $\leq 1/\epsilon$ neighbors in Y (call this set B').



Randomly assigning Y should satisfy whp $\geq (1-\epsilon^2)\cdot(1-2^{-1/\epsilon})$ of $B \setminus B'$, while $S \setminus S'$ can be solved optimally.

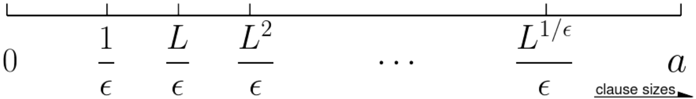
Lemma

We can always find a small set M ($|M| \leq \epsilon \cdot m$) of medium-size clauses (arities $d \sim D$).

An FPT-AS for MaxCNFSat parameterized by cw*

Lemma

We can always find a small set M ($|M| \leq \epsilon \cdot m$) of medium-size clauses (arities $d \sim D$).

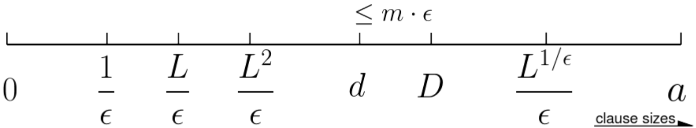


Define $1/\epsilon + 1$ independent intervals of medium-arity clauses (right-left bounds are an $L(= \epsilon^{-4})$-factor apart).

An FPT-AS for MaxCNFSat parameterized by cw*

Lemma

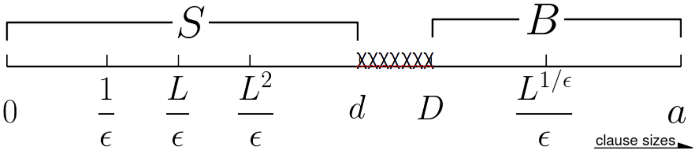
We can always find a small set M ($|M| \leq \epsilon \cdot m$) of medium-size clauses (arities $d \sim D$).



There should be at least one interval $[d, D]$ ($D = L \cdot d$) containing $\leq \epsilon \cdot m$ clauses.

Lemma

We can always find a small set M ($|M| \leq \epsilon \cdot m$) of medium-size clauses (arities $d \sim D$).



Removing them divides the clauses into small (S) and big (B).

The algorithm

- Find interval [d,D] of at most $\epsilon \cdot m$ clauses of medium arities as in the previous Lemma and ignore them.

The algorithm

- Find interval [d,D] of at most $\epsilon \cdot m$ clauses of medium arities as in the previous Lemma and ignore them.
- Split remaining clauses into S (arity $< d$) and B (arity $> D$).

The algorithm

- Find interval [d,D] of at most $\epsilon \cdot m$ clauses of medium arities as in the previous Lemma and ignore them.
- Split remaining clauses into S (arity $< d$) and B (arity $> D$).
- If $|S| \leq \epsilon^2 \cdot m$

The algorithm

- Find interval [d,D] of at most $\epsilon \cdot m$ clauses of medium arities as in the previous Lemma and ignore them.
- Split remaining clauses into S (arity $< d$) and B (arity $> D$).
- If $|S| \leq \epsilon^2 \cdot m$
  - Ignore S;

The algorithm

- Find interval [d,D] of at most $\epsilon \cdot m$ clauses of medium arities as in the previous Lemma and ignore them.
- Split remaining clauses into S (arity $< d$) and B (arity $> D$).
- If $|S| \leq \epsilon^2 \cdot m$
  - Ignore S;
  - Randomly assign variables to satisfy most of B.

The algorithm

- Find interval [d,D] of at most $\epsilon \cdot m$ clauses of medium arities as in the previous Lemma and ignore them.
- Split remaining clauses into S (arity $< d$) and B (arity $> D$).
- If $|S| \leq \epsilon^2 \cdot m$
  - Ignore S;
  - Randomly assign variables to satisfy most of B.
- If at most $|B| \leq \epsilon^2 \cdot m$

The algorithm

- Find interval [d,D] of at most $\epsilon \cdot m$ clauses of medium arities as in the previous Lemma and ignore them.
- Split remaining clauses into S (arity $< d$) and B (arity $> D$).
- If $|S| \leq \epsilon^2 \cdot m$
  - Ignore S;
  - Randomly assign variables to satisfy most of B.
- If at most $|B| \leq \epsilon^2 \cdot m$
  - Ignore B;

The algorithm

- Find interval [d,D] of at most $\epsilon \cdot m$ clauses of medium arities as in the previous Lemma and ignore them.
- Split remaining clauses into S (arity $< d$) and B (arity $> D$).
- If $|S| \leq \epsilon^2 \cdot m$
  - Ignore S;
  - Randomly assign variables to satisfy most of B.
- If at most $|B| \leq \epsilon^2 \cdot m$
  - Ignore B;
  - G_S has bounded treewidth $\rightarrow$ solve optimally.

The algorithm

- Find interval [d,D] of at most $\epsilon \cdot m$ clauses of medium arities as in the previous Lemma and ignore them.
- Split remaining clauses into S (arity $< d$) and B (arity $> D$).
- If $|S| \leq \epsilon^2 \cdot m$
  - Ignore S;
  - Randomly assign variables to satisfy most of B.
- If at most $|B| \leq \epsilon^2 \cdot m$
  - Ignore B;
  - G_S has bounded treewidth $\rightarrow$ solve optimally.
- Otherwise

The algorithm

- Find interval [d,D] of at most $\epsilon \cdot m$ clauses of medium arities as in the previous Lemma and ignore them.
- Split remaining clauses into S (arity $< d$) and B (arity $> D$).
- If $|S| \leq \epsilon^2 \cdot m$
  - Ignore S;
  - Randomly assign variables to satisfy most of B.
- If at most $|B| \leq \epsilon^2 \cdot m$
  - Ignore B;
  - G_S has bounded treewidth $\rightarrow$ solve optimally.
- Otherwise
  - Find set of variables Y as in the last case and set it randomly to satisfy most of B.

The algorithm

- Find interval [d,D] of at most $\epsilon \cdot m$ clauses of medium arities as in the previous Lemma and ignore them.
- Split remaining clauses into S (arity $< d$) and B (arity $> D$).
- If $|S| \leq \epsilon^2 \cdot m$
  - Ignore S;
  - Randomly assign variables to satisfy most of B.
- If at most $|B| \leq \epsilon^2 \cdot m$
  - Ignore B;
  - G_S has bounded treewidth $\rightarrow$ solve optimally.
- Otherwise
  - Find set of variables Y as in the last case and set it randomly to satisfy most of B.
  - Ignore part of S that contains variables from Y and solve the rest optimally.

And and Parity Constraints

Similarities

- DNFSAT and PARITY are both in P.

And and Parity Constraints

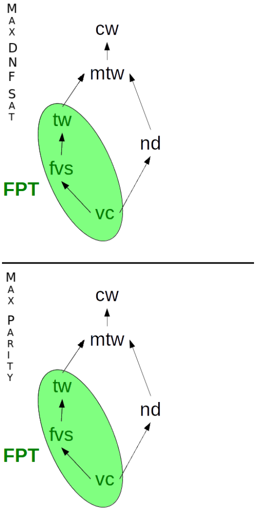
Similarities

- DNFSat and Parity are both in P.
- MaxDNFSat and MaxParity are both APX-hard.

And and Parity Constraints

Similarities

- DNFSat and Parity are both in P.
- MaxDNFSat and MaxParity are both APX-hard.
- MaxDNFSat and MaxParity are both FPT parameterized by tw^*.



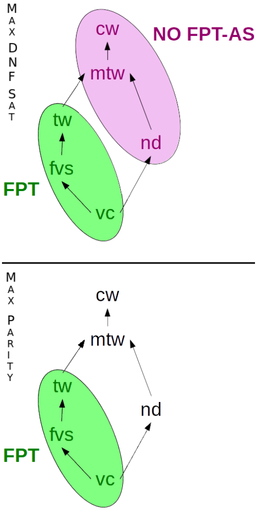
And and Parity Constraints

Similarities

- DNFSat and Parity are both in P.
- MaxDNFSat and MaxParity are both APX-hard.
- MaxDNFSat and MaxParity are both FPT parameterized by tw^*.

Different behavior for dense structural parameters

- MaxDNFSat parameterized by nd^* does not admit FPT-AS (unless FPT=W[1]);



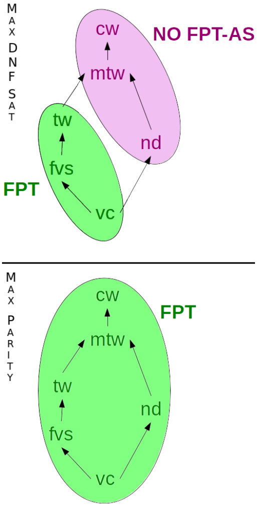
And and Parity Constraints

Similarities

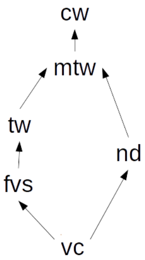
- DNFSat and Parity are both in P.
- MaxDNFSat and MaxParity are both APX-hard.
- MaxDNFSat and MaxParity are both FPT parameterized by tw^*.

Different behavior for dense structural parameters

- MaxDNFSat parameterized by nd^* does not admit FPT-AS (unless FPT=W[1]);
- MaxParity parameterized by cw^* is FPT.



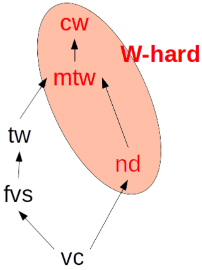
A more difficult constraint - Majority



A more difficult constraint - Majority

Corollary (tweak of W-hardness for CNFSAT)

MAJORITY parameterized by nd^* is W[1]-hard.

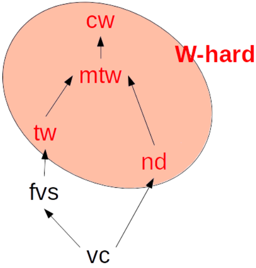


A more difficult constraint - Majority

Corollary (tweak of W-hardness for CNFSat)

MAJORITY *parameterized by* nd^* *is W[1]-hard.*

$\rightarrow$ MAJORITY parameterized by tw^* is W[1]-hard.



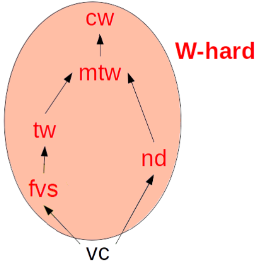
Corollary (tweak of W-hardness for CNFSAT)

MAJORITY *parameterized by* nd^* *is W[1]-hard.*

$\rightarrow$MAJORITY parameterized by tw^* is W[1]-hard.
In fact,

Theorem

MAJORITY *parameterized by* fvs^* *is W[1]-hard.*



A more difficult constraint - Majority

Corollary (tweak of W-hardness for CNFSat)

MAJORITY *parameterized by* nd^* *is W[1]-hard.*

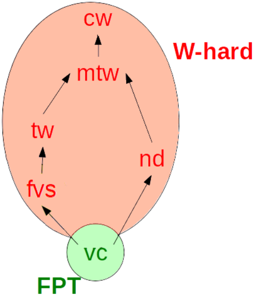
$\rightarrow$MAJORITY parameterized by tw^* is W[1]-hard.
In fact,

Theorem

MAJORITY *parameterized by* fvs^* *is W[1]-hard.*

Theorem

MAXMAJORITY *parameterized by* vc^* *is FPT.*



A more difficult constraint - Majority

Corollary (tweak of W-hardness for CNFSAT)

MAJORITY *parameterized by nd^* is W[1]-hard.*

→MAJORITY parameterized by tw^* is W[1]-hard.
In fact,

Theorem

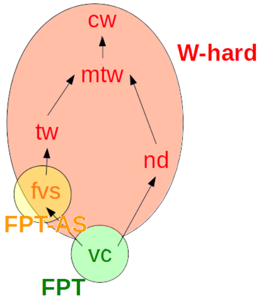
MAJORITY *parameterized by fvs^* is W[1]-hard.*

Theorem

MAXMAJORITY *parameterized by vc^* is FPT.*

Theorem

MAXMAJORITY *parameterized by fvs^* admits an FPT-AS.*



MaxMajority parameterized by vc^* is FPT

Remark 1

vc^* dominates both $n = \#vars$ and $m = \#cons$.

MaxMajority parameterized by vc^* is FPT

Remark 1

vc^* dominates both $n = \#vars$ and $m = \#cons$.

Remark 2

We can reduce MaxMajority parameterized by vc^* to MaxMajority parameterized by m.

MaxMajority parameterized by vc^* is FPT

Remark 1

vc^* dominates both $n = \#vars$ and $m = \#cons$.

Remark 2

We can reduce MaxMajority parameterized by vc^* to MaxMajority parameterized by m.

Remark 3

We can reduce MaxMajority to Majority.

MaxMajority parameterized by vc^* is FPT

Remark 1

vc^* dominates both $n = \#vars$ and $m = \#cons$.

Remark 2

We can reduce MaxMajority parameterized by vc^* to MaxMajority parameterized by m.

Remark 3

We can reduce MaxMajority to Majority.

Remark 4

We can reduce Majority to an ILP with 3^m variables $\rightarrow$ FPT [Lenstra 1983].

(Again, we can assume that fvs^* contains only constraint vertices.)
We consider two cases:

MaxMajority parameterized by fvs^* admits an FPT-AS

(Again, we can assume that fvs^* contains only constraint vertices.)
We consider two cases:

- If $m < {}^{2 \cdot \mathrm{fvs}^*}/_{\epsilon}$, the graph reduces to the bounded m case.

(Again, we can assume that fvs^* contains only constraint vertices.)
We consider two cases:

- If $m < {}^{2 \cdot \text{fvs}^*}/_{\epsilon}$, the graph reduces to the bounded m case.
- If $m \geq {}^{2 \cdot \text{fvs}^*}/_{\epsilon}$, then $\text{fvs}^* \leq {}^{m \cdot \epsilon}/_{2}$:

MaxMajority parameterized by fvs^* admits an FPT-AS

(Again, we can assume that fvs^* contains only constraint vertices.)
We consider two cases:

- If $m < {}^{2 \cdot \mathrm{fvs}^*}/_{\epsilon}$, the graph reduces to the bounded m case.
- If $m \geq {}^{2 \cdot \mathrm{fvs}^*}/_{\epsilon}$, then $\mathrm{fvs}^* \leq {}^{m \cdot \epsilon}/_{2}$:
    - ignore fvs^*;
    - solve optimally the acyclic graph.

(Again, we can assume that fvs^* contains only constraint vertices.)
We consider two cases:

- If $m < {}^{2 \cdot \mathrm{fvs}^*}/_{\epsilon}$, the graph reduces to the bounded m case.
- If $m \geq {}^{2 \cdot \mathrm{fvs}^*}/_{\epsilon}$, then $\mathrm{fvs}^* \leq {}^{m \cdot \epsilon}/_{2}$:
  - ignore fvs^*;
  - solve optimally the acyclic graph.

In order to obtain the desired outcome in the second case, we need linear dependence of OPT and m:

MaxMajority parameterized by fvs^* admits an FPT-AS

(Again, we can assume that fvs^* contains only constraint vertices.)
We consider two cases:

- If $m < {}^{2 \cdot \text{fvs}^*}/_{\epsilon}$, the graph reduces to the bounded m case.
- If $m \geq {}^{2 \cdot \text{fvs}^*}/_{\epsilon}$, then $\text{fvs}^* \leq {}^{m \cdot \epsilon}/_{2}$:
  - ignore fvs^*;
  - solve optimally the acyclic graph.

In order to obtain the desired outcome in the second case, we need linear dependence of OPT and m:

- $OPT \geq \frac{m}{2}$ (if an assignment doesn't satisfy at least $\frac{m}{2}$ constraints, it's negation does).

Conclusions

- We studied four natural boolean CSPs (or, and, parity, majority constraints).
- Structural parameterizations (incidence graph).
- Provided some of the first **parameterized approximation** results for CSP.
- Complexity-wise, studied CSPs exhibit wildly different behaviors (FPT, W-hard admitting FPT-AS, no FPT-AS unless FPT=W[1]).
- Complete classification?

Conclusions

- We studied four natural boolean CSPs (or, and, parity, majority constraints).
- Structural parameterizations (incidence graph).
- Provided some of the first **parameterized approximation** results for CSP.
- Complexity-wise, studied CSPs exhibit wildly different behaviors (FPT, W-hard admitting FPT-AS, no FPT-AS unless FPT=W[1]).
- Complete classification?

Conclusions

- We studied four natural boolean CSPs (or, and, parity, majority constraints).
- Structural parameterizations (incidence graph).
- Provided some of the first **parameterized approximation** results for CSP.
- Complexity-wise, studied CSPs exhibit wildly different behaviors (FPT, W-hard admitting FPT-AS, no FPT-AS unless FPT=W[1]).
- Complete classification?

Conclusions

- We studied four natural boolean CSPs (or, and, parity, majority constraints).
- Structural parameterizations (incidence graph).
- Provided some of the first **parameterized approximation** results for CSP.
- Complexity-wise, studied CSPs exhibit wildly different behaviors (FPT, W-hard admitting FPT-AS, no FPT-AS unless FPT=W[1]).
- Complete classification?

Summary of Results

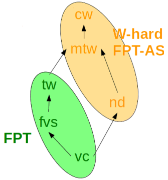


Figure: Diagram for CNFSat and MaxCNFSat.

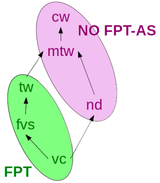


Figure: Diagram for MaxDNFSat.

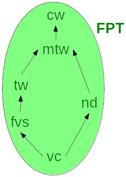


Figure: Diagram for MaxParity.

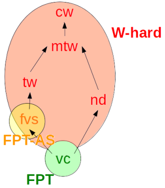


Figure: Diagram for Majority and MaxMajority.

Conclusions

- We studied four natural boolean CSPs (or, and, parity, majority constraints).
- Structural parameterizations (incidence graph).
- Provided some of the first **parameterized approximation** results for CSP.
- Complexity-wise, studied CSPs exhibit wildly different behaviors (FPT, W-hard admitting FPT-AS, no FPT-AS unless FPT=W[1]).
- Complete classification?

Thank you!
Questions?