Excluding Polynomial-time Approximation Schemes for Max CSP

Fredrik Kuivinen
freku@ida.liu.se
Joint work with Peter Jonsson and Andrei Krokhin

Department of Computer and Information Science
Linköpings Universitet, Sweden

Nashville 2007

Outline

MAX CSP

Let D be a finite set, the domain.

- An n-ary relation is a subset of D^n.
- A constraint language L is a set of relations over D.
- "$R_i(x_1, \ldots, x_n)$" where $R_i \in L$, is an L-constraint.

MAX CSP

Let D be a finite set, the domain.

- An n-ary relation is a subset of D^n.
- A constraint language L is a set of relations over D.
- "$R_i(x_1, \ldots, x_n)$" where $R_i \in L$, is an L-constraint.

The problem MAX CSP(L) is:

Instance: A collection of L-constraints C.

Solution: An assignment to the variables.

Measure: The number of satisfied constraints.

MAX CSP

Let D be a finite set, the domain.

- An n-ary relation is a subset of D^n.
- A constraint language L is a set of relations over D.
- "$R_i(x_1, \ldots, x_n)$" where $R_i \in L$, is an L-constraint.

The problem MAX CSP(L) is:

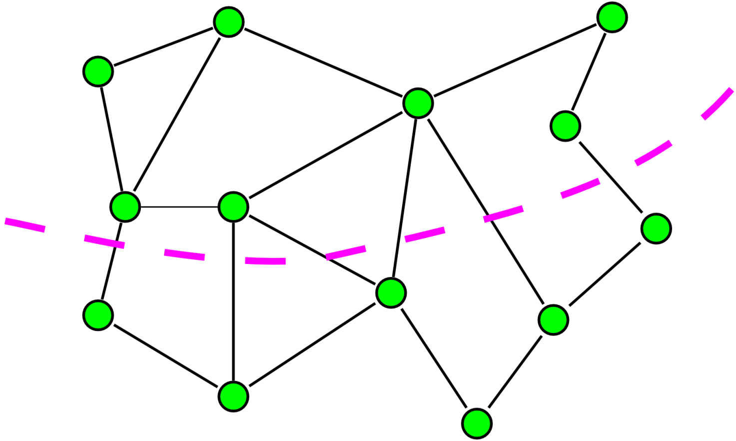
Instance: A collection of L-constraints C.

Solution: An assignment to the variables.

Measure: The number of satisfied constraints.

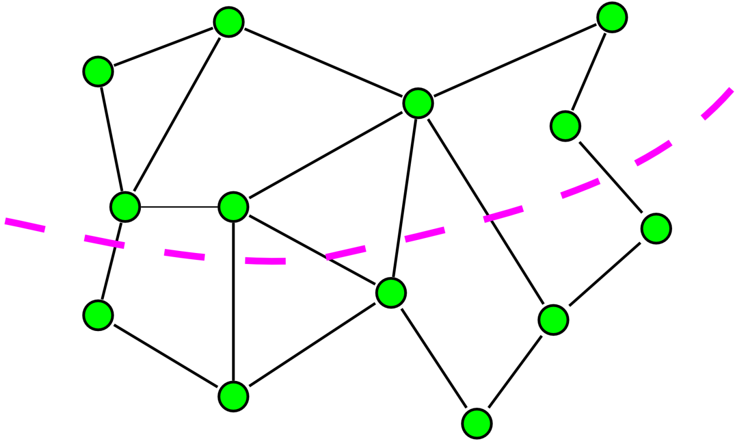
Natural optimisation analogue of CSP(L).

Example: MAX CUT



Find a partition A, B of the set of vertices such that the number of edges between A and B are maximised.

Example: MAX CUT



Find a partition A, B of the set of vertices such that the number of edges between A and B are maximised.

Equivalent to MAX CSP($\{R\}$) where $R = \{(0, 1), (1, 0)\}$.

Open Problem

Open problem: Characterise the constraint languages L such that MAX CSP(L) is tractable.

Hard Constraint Languages

Theorem (Bulatov, Jeavons, Krokhin)

Let L be a core constraint language. If $\mathcal{A}_L^c$ contains a non-trivial factor which only have projections as term operations, then CSP(L) *is **NP**-complete.*

We say that a constraint language which have this property is hard.

Hard Constraint Languages

Theorem (Bulatov, Jeavons, Krokhin)

Let L be a core constraint language. If $\mathcal{A}_L^c$ contains a non-trivial factor which only have projections as term operations, then CSP(L) *is **NP**-complete.*

We say that a constraint language which have this property is hard.

What happens for MAX CSP in this case?

Hard Constraint Languages

Theorem (Bulatov, Jeavons, Krokhin)

Let L be a core constraint language. If $\mathcal{A}_L^c$ contains a non-trivial factor which only have projections as term operations, then CSP(L) *is **NP**-complete.*

We say that a constraint language which have this property is hard.

What happens for MAX CSP in this case?

As CSP(L) is **NP**-hard we cannot hope to find optimal solutions to MAX CSP(L) in polynomial time.

Can we do anything at all?

Approximation

An approximation algorithm is a polynomial-time algorithm such that:

$$R \geq \frac{\text{Optimal value}}{\text{Found value}}$$

Worst case over the instances. The value R is called the performance ratio of the algorithm.

Known Results

For any L there is an R and an approximation algorithm for MAX CSP(L) which produces solutions of performance ratio R.

Known Results

For any L there is an R and an approximation algorithm for MAX CSP(L) which produces solutions of performance ratio R.

Algorithm: Pick a random value, uniformly and independently, for each variable. In expectation we will satisfy a constant fraction of the constraints.

Known Results

For any L there is an R and an approximation algorithm for MAX CSP(L) which produces solutions of performance ratio R.

Algorithm: Pick a random value, uniformly and independently, for each variable. In expectation we will satisfy a constant fraction of the constraints.

So, the answer to the question “can we do anything at all?” is Yes.

Known Results

For any L there is an R and an approximation algorithm for MAX CSP(L) which produces solutions of performance ratio R.

Algorithm: Pick a random value, uniformly and independently, for each variable. In expectation we will satisfy a constant fraction of the constraints.

So, the answer to the question "can we do anything at all?" is Yes.

Can we do (much) better?

Approximation, PTAS

A polynomial-time approximation scheme (PTAS) is an algorithm such that for any $R > 1$ we have

$$R \geq \frac{\text{Optimal value}}{\text{Found value}}$$

in time polynomial in the size of the instance.

Approximation, PTAS

A polynomial-time approximation scheme (PTAS) is an algorithm such that for any $R > 1$ we have

$$R \geq \frac{\text{Optimal value}}{\text{Found value}}$$

in time polynomial in the size of the instance.

Note: The time may depend arbitrarily on R! In particular

$$n^{1/(R-1)}$$

is OK. (Here n denotes the size of the instance.)

PTAS's for MAX CSP(L)?

Question: Is there a PTAS for MAX CSP(L) for some hard constraint language L?

PTAS's for MAX CSP(L)?

Question: Is there a PTAS for MAX CSP(L) for some hard constraint language L? **No!** (unless **P** = **NP**)

Theorem (Jonsson, Krokhin, Kuivinen)

*For any hard constraint languages L, there is a $\alpha > 1$ such that it is **NP**-hard to approximate* MAX CSP(L) *better than α.*

Hardness at gap location 1

A Max CSP problem is hard at gap location 1 if it is **NP**-hard to distinguish instances in which all constraints are simultaneously satisfiable from instances where only an α-fraction of the constraints are simultaneously satisfiable.

Hardness at gap location 1

A MAX CSP problem is hard at gap location 1 if it is **NP**-hard to distinguish instances in which all constraints are simultaneously satisfiable from instances where only an α-fraction of the constraints are simultaneously satisfiable.

Example (Håstad)

It is **NP**-hard to distinguish instances of MAX 3SAT in which all constraints are simultaneously satisfiable from instances where only $7/8 + \epsilon$ of the constraints are satisfiable.

Hardness at gap location 1

A MAX CSP problem is hard at gap location 1 if it is **NP**-hard to distinguish instances in which all constraints are simultaneously satisfiable from instances where only an α-fraction of the constraints are simultaneously satisfiable.

Example (Håstad)

It is **NP**-hard to distinguish instances of MAX 3SAT in which all constraints are simultaneously satisfiable from instances where only $7/8 + \epsilon$ of the constraints are satisfiable.
Therefore, MAX 3SAT is hard at gap location 1.

Hardness at gap location 1

A Max CSP problem is hard at gap location 1 if it is **NP**-hard to distinguish instances in which all constraints are simultaneously satisfiable from instances where only an α-fraction of the constraints are simultaneously satisfiable.

Example (Håstad)

It is **NP**-hard to distinguish instances of Max 3Sat in which all constraints are simultaneously satisfiable from instances where only $7/8 + \epsilon$ of the constraints are satisfiable.
Therefore, Max 3Sat is hard at gap location 1.

Example

Max 2Sat is not hard at gap location 1. Why?

Hardness at gap location 1

A Max CSP problem is hard at gap location 1 if it is **NP**-hard to distinguish instances in which all constraints are simultaneously satisfiable from instances where only an α-fraction of the constraints are simultaneously satisfiable.

Example (Håstad)

It is **NP**-hard to distinguish instances of Max 3Sat in which all constraints are simultaneously satisfiable from instances where only $7/8 + \epsilon$ of the constraints are satisfiable.
Therefore, Max 3Sat is hard at gap location 1.

Example

Max 2Sat is not hard at gap location 1. Why?
It is easy (in **P**) to decide if all constraints are simultaneously satisfiable.

A Stronger Result

Theorem (Jonsson, Krokhin, Kuivinen)

For hard constraint languages L, MAX CSP(L)-B *is hard at gap location 1.*

A Stronger Result

Theorem (Jonsson, Krokhin, Kuivinen)

For hard constraint languages L, MAX CSP(L)-B *is hard at gap location 1.*

Corollary

For hard constraint languages L, CSP(L)-B *is **NP**-complete.*

Proof Ingredients

- Prove hardness at gap location 1.
- Use bounded occurrence instances.
- Use an alternative characterisation of hard constraint languages.
- Use the fact that the problem MAX NOT-ALL-EQUAL-3SAT is hard at gap location 1.

Why hardness at gap location 1 and bounded occurrence?

For the CSP problem we have

Lemma

Let L be a constraint language and let R be a relation which can be expressed by a primitive-positive formula using L. If $\mathrm{CSP}(L \cup \{R\})$ *is **NP**-hard, then* $\mathrm{CSP}(L)$ *is **NP**-hard.*

We want something similar for MAX CSP.

Why hardness at gap location 1 and bounded occurrence? cont.

Lemma

Let L be a constraint language and let R be a relation which can be expressed by a primitive-positive formula using L. If MAX CSP$(L \cup \{R\})$-k *has a hard gap at location 1, then there is an integer k' such that* MAX CSP(L)-k' *has a hard gap at location 1.*

Preserves non-approximability!

Proof Idea

Use the primitive-positive formula to replace constraints using *R* by constraints which only use *L*.

Due to proving hardness at gap location 1 we get two cases:

- When all constraints are satisfiable in the original instance, it is not hard to see that all constraints in the resulting instance are satisfiable.
- Otherwise, less than an α (some constant $\alpha < 1$) fraction of the constraints are satisfied in the original instance. Prove that there is a constant $\alpha' < 1$ such that at most an α' fraction of the constraints are satisfied in the resulting instance.

The bounded occurrence property is used in the second case.

Alternative Characterisation

Let NAE be the not-all-equal relation, that is,
$NAE = \{(0,0,1),(0,1,0),(1,0,0),(1,1,0),(1,0,1),(0,1,1)\}$.

Theorem (Bulatov, Jeavons, Krokhin)

Let L be a core constraint language. The following are equivalent:

- *The algebra $\mathcal{A}_L^c$ has a non-trivial factor whose term operations are only projections.*
- *There exists a subset B of D and a surjective mapping $\phi : B \to \{0,1\}$ such that the relational clone $\langle L \cup C_D \rangle$ contains the relation $\phi^{-1}(NAE)$ which is the full preimage (under ϕ) of NAE.*

C_D is the set of all singleton unary relations (the constants).

The Constants Cause Problems

For CSP we have:

Theorem (Bulatov, Jeavons, Krokhin)

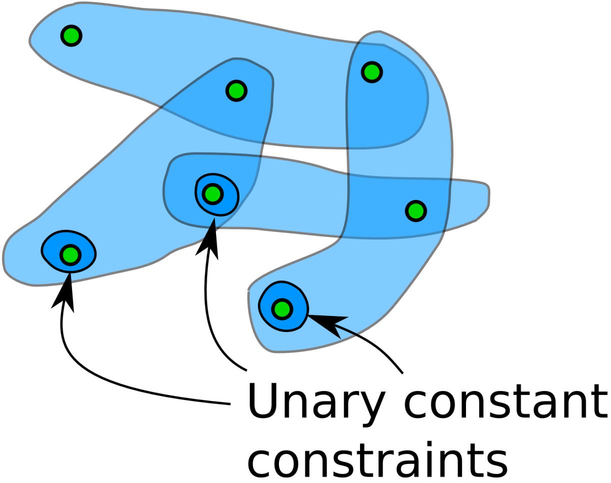
If L is a core, then $\mathrm{CSP}(L \cup C_D)$ *is tractable if and only if* $\mathrm{CSP}(L)$ *is tractable.*

The construction introduces one variable per domain element.

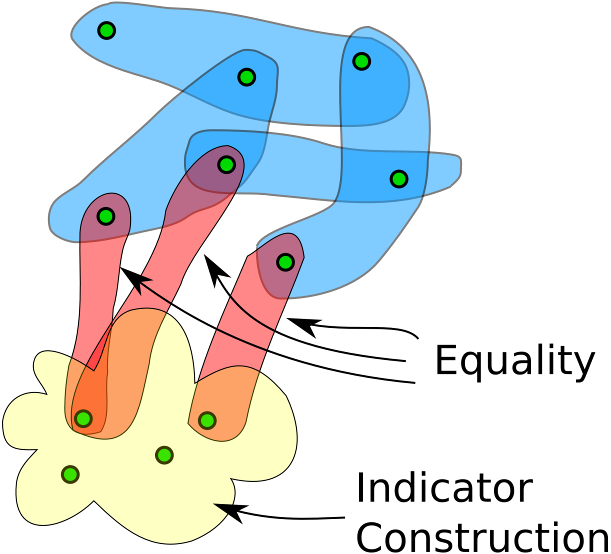
For MAX CSP there are problems with this construction:

- Equality constraints are introduced.
- The resulting instance is not of bounded occurrence.

The CSP Construction



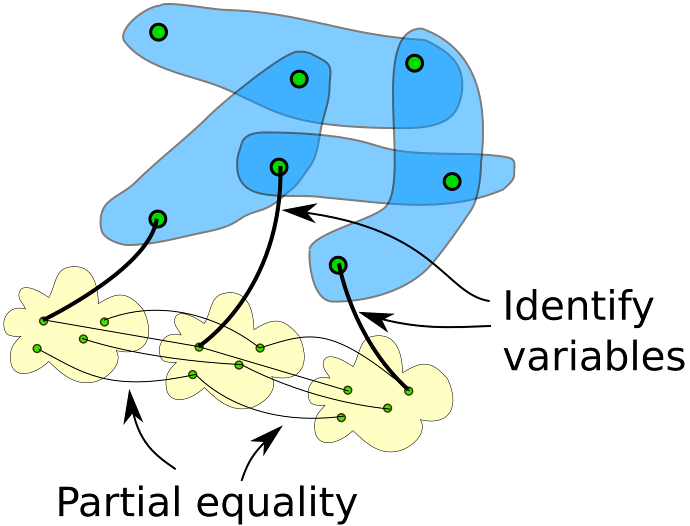
The CSP Construction



A Workaround

- Prove that for any orbit Ω of the automorphism group of L, we can pp-express the equality relation restricted to Ω.
- Use several indicator constructions, instead of one, and impose partial equality constraints on the relevant variables.
- Use expander graphs to bound the number of variable occurrences.

The MAX CSP Construction



Only One Relation, Problem Statement

We want to characterise the complexity of MAX CSP($\{R\}$).
That is, MAX CSP in which only one constraint type is allowed.

Includes MAX CUT, MAX DICUT and MAX H-COLOURING among others.

Known Results

Say that a relation R is valid if there is a d such that $(d, d, \ldots, d) \in R$.

Theorem

If R is valid, then $\textsc{Max}$ $\text{CSP}(\{R\})$ *is tractable.*

Known Results

Say that a relation R is valid if there is a d such that $(d, d, \ldots, d) \in R$.

Theorem

If R is valid, then $\textsc{Max}$ $\mathrm{CSP}(\{R\})$ *is tractable.*

Proof: Assign all variables the value d. All constraints will be satisfied.

Known Results

Say that a relation R is valid if there is a d such that $(d, d, \ldots, d) \in R$.

Theorem

If R is valid, then $\text{MAX CSP}(\{R\})$ *is tractable.*

Proof: Assign all variables the value d. All constraints will be satisfied.

Theorem (Jonsson, Krokhin)

If R is not valid, then $\text{MAX CSP}(\{R\})$ *is* ***NP****-hard.*

Validity is the only way to make $\text{MAX CSP}(\{R\})$ tractable! (unless **P** = **NP**)

What about PTAS's?

Is there a non-valid relation R such that MAX CSP(R) have a PTAS?

What about PTAS's?

Is there a non-valid relation R such that $\text{MAX CSP}(R)$ have a PTAS?

Theorem

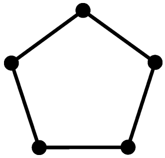
If R is not valid, then $\text{MAX CSP}(\{R\})$ do not admit a PTAS, unless ***P*** *=* ***NP****.*

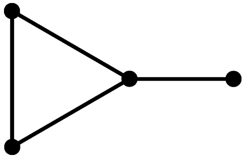
Proof Ingredients

- Reduce the problem to one binary relation.
- Study the automorphism group of the binary relation. Only vertex-transitive digraphs remain after this step.
- Adapt a result of MacGillivray, which characterises the complexity of CSP(G) for vertex-transitive digraphs G, to the algebraic framework.
- Use the knowledge of when CSP is hard for vertex-transitive digraphs to get hardness at gap location 1 for MAX CSP.

Constraints Satisfaction and Vertex-transitive Digraphs

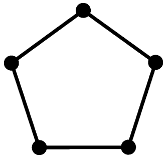
A digraph $G = (V, E)$ is vertex-transitive if for any $x, y \in V$ there is an automorphism ρ such that $\rho(x) = y$.

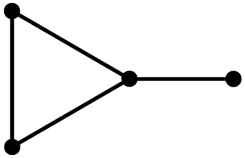




Constraints Satisfaction and Vertex-transitive Digraphs

A digraph $G = (V, E)$ is vertex-transitive if for any $x, y \in V$ there is an automorphism ρ such that $\rho(x) = y$.





Theorem (MacGillivray)

*Let $G = (V, E)$ be a vertex-transitive digraph which is a core. If G is a directed cycle, then $\mathrm{CSP}(\{E\})$ is tractable. Otherwise, $\mathrm{CSP}(\{E\})$ is **NP**-hard.*

Conclusions and Open Problems

- Our result holds for bounded occurrence instances, but we have not bothered to state any explicit bounds. What is the fewest number of occurrences we need to rule out PTAS's?
- Similarly, we have not calculated any explicit non-approximability bounds. What are the best bounds we can get?
- Characterise the complexity of MAX CSP(L) for all L.