An Algebraic Characterisation of Complexity for Valued Constraints

David A. Cohen[1] Martin C. Cooper[2] Peter G. Jeavons[3]

[1]Department of Computer Science, Royal Holloway, University of London, UK
d.cohen@rhul.ac.uk

[2]IRIT, University of Toulouse III,
France cooper@irit.fr

[3]Computing Laboratory, University of Oxford, UK
peter.jeavons@comlab.ox.ac.uk

Dagstuhl Meeting on the Complexity of CSP, October 2006

Outline

1. Motivation
   - The Valued CSP

2. Our Results/Contribution
   - The Results
   - Basic Ideas for Proof

Outline

1. Motivation
   - The Valued CSP
2. Our Results/Contribution
   - The Results
   - Basic Ideas for Proof

Outline

1. Motivation
   - The Valued CSP
2. Our Results/Contribution
   - The Results
   - Basic Ideas for Proof

From CSP to VCSP

Definition

A CSP is a triple $\langle V, D, C \rangle$ where

- V is a finite set of variables
- D is a finite domain
- C is a finite set of constraints.
  Each constraint is a pair $\langle \sigma, \rho \rangle$ where
  - $\sigma \in V^*$ is the scope
  - $\rho \subseteq D^{|\sigma|}$ is the relation

From CSP to VCSP

Definition

A CSP is a triple $\langle V, D, C \rangle$ where

- V is a finite set of variables
- D is a finite domain
- C is a finite set of constraints.
  Each constraint is a pair $\langle \sigma, \rho \rangle$ where
  - $\sigma \in V^*$ is the scope
  - ρ is a mapping from $D^{|\sigma|}$ to $\{0, \infty\}$ is the relation

From CSP to VCSP

Definition

A VCSP is a four tuple $\langle V, D, C, \Omega \rangle$ where

- V is a finite set of variables
- D is a finite domain
- C is a finite set of constraints.
  Each constraint is a pair $\langle \sigma, \phi \rangle$ where
  - $\sigma \in V^*$ is the scope
  - ϕ is a mapping from $D^{|\sigma|}$ to Ω called the cost function

From CSP to VCSP

Definition

A VCSP is a four tuple $\langle V, D, C, \Omega \rangle$ where

- V is a finite set of variables
- D is a finite domain
- C is a finite set of constraints.
  Each constraint is a pair $\langle \sigma, \phi \rangle$ where
  - $\sigma \in V^*$ is the scope
  - ϕ is a mapping from $D^{|\sigma|}$ to Ω called the cost function

Definition

The valuation structure Ω, is ordered, with a 0 and ∞ and an AC aggregation $\oplus$, where $\alpha \geq \beta, \gamma \in \Omega$ and $\alpha \oplus \gamma \geq \beta \oplus \gamma$.

From CSP to VCSP

Definition

A VCSP is a four tuple $\langle V, D, C, \Omega \rangle$ where

- V is a finite set of variables
- D is a finite domain
- C is a finite set of constraints.
  Each constraint is a pair $\langle \sigma, \phi \rangle$ where
  - $\sigma \in V^*$ is the scope
  - ϕ is a mapping from $D^{|\sigma|}$ to Ω called the cost function

Definition

An assignment is a mapping $s : V \mapsto D$. Its cost is given by
$\text{Cost}_p(s) = \bigoplus_{\langle\langle v_1,\ldots,v_r\rangle,\phi\rangle \in C} \phi(s(v_1),\ldots,s(v_r))$.

From CSP to VCSP

Definition

A VCSP is a four tuple $\langle V, D, C, \Omega \rangle$ where

- V is a finite set of variables
- D is a finite domain
- C is a finite set of constraints.
  Each constraint is a pair $\langle \sigma, \phi \rangle$ where
  - $\sigma \in V^*$ is the scope
  - ϕ is a mapping from $D^{|\sigma|}$ to Ω called the cost function

Definition

A solution is an assignment with minimal cost.

From CSP to VCSP

Definition

A VCSP is a four tuple $\langle V, D, C, \Omega \rangle$ where

- V is a finite set of variables
- D is a finite domain
- C is a finite set of constraints.
  Each constraint is a pair $\langle \sigma, \phi \rangle$ where
  - $\sigma \in V^*$ is the scope
  - ϕ is a mapping from $D^{|\sigma|}$ to Ω called the cost function

Definition

Here we will stick to the valuation structure $\mathbb{Q} \cup \infty$.

Motivation

- The VCSP framework is a natural extension of CSP when we are interested in optimisation rather that simply feasibility.
- VCSP properly extends MAX-CSP, CSP and MAX-SAT.
- There are large well-known tractable VCSP languages.

Background

- Tractable cases known when $D = \{0, 1\}, \Omega = \{0, 1, \infty\}$ and all cost functions have finite values (Creignou).
- Extended to $|D| = 3$. There is still essentially just one class (Jonsson).
- Tractable cases identified for $|D| = 2$ with rational or infinite values. There are precisely eight cases.
- Several other maximal tractable languages discovered for larger domains.
- All known maximal tractable cases are characterised by single multimorphisms
- Intractable languages have no multimorphisms (to speak of)

Background

- Tractable cases known when $D = \{0, 1\}, \Omega = \{0, 1, \infty\}$ and all cost functions have finite values (Creignou).
- Extended to $|D| = 3$. There is still essentially just one class (Jonsson).
- Tractable cases identified for $|D| = 2$ with rational or infinite values. There are precisely eight cases.
- Several other maximal tractable languages discovered for larger domains.
- All known maximal tractable cases are characterised by single multimorphisms
- Intractable languages have no multimorphisms (to speak of)

Background

- Tractable cases known when $D = \{0, 1\}, \Omega = \{0, 1, \infty\}$ and all cost functions have finite values (Creignou).
- Extended to $|D| = 3$. There is still essentially just one class (Jonsson).
- Tractable cases identified for $|D| = 2$ with rational or infinite values. There are precisely eight cases.
- Several other maximal tractable languages discovered for larger domains.
- All known maximal tractable cases are characterised by single multimorphisms
- Intractable languages have no multimorphisms (to speak of)

Multimorphisms

Definition

A *k-ary weighted function* F on a set D is a set of the form $\{\langle w_1, f_1\rangle, \ldots, \langle w_n, f_n\rangle\}$ where each w_i is a positive integer, and each f_i is a distinct function from D^k to D.

For any r-ary cost function ϕ, we say that a k-ary weighted function F is a *k-ary fractional polymorphism* of ϕ if, for all $\langle x_1, \ldots, x_r\rangle \in D^k$,

$$k \sum_{i=1}^{n} w_i \phi(f_i(x_1), \ldots, f_i(x_r)) \leq \left(\sum_{i=1}^{n} w_i\right) \cdot \left(\sum_{i=1}^{k} \phi(x_1[i], \ldots, x_r[i])\right)$$

The set of all fractional polymorphisms of Γ is denoted fPol(Γ).

Multimorphisms

Definition

A *k-ary weighted function* F on a set D is a set of the form $\{\langle w_1, f_1\rangle, \ldots, \langle w_n, f_n\rangle\}$ where each w_i is a positive integer, and each f_i is a distinct function from D^k to D.

For any r-ary cost function ϕ, we say that a k-ary weighted function F is a *k-ary fractional polymorphism* of ϕ if, for all $\langle x_1, \ldots, x_r\rangle \in D^k$,

$$k \sum_{i=1}^{n} w_i \phi(f_i(x_1), \ldots, f_i(x_r)) \leq \left(\sum_{i=1}^{n} w_i\right) \cdot \left(\sum_{i=1}^{k} \phi(x_1[i], \ldots, x_r[i])\right)$$

The set of all fractional polymorphisms of Γ is denoted fPol(Γ).
It is a multimorphism precisely when $\sum_{i=1}^{n} w_i = k$.

Feasibility Relations

- We define Feas(Γ) . . . in the natural way.
- Every function in a fractional polymorphism of ϕ is a classical polymorphism of Feas(ϕ).

VCSP Expressiveness defined

We have three ways of generating new cost functions.

- Rational Equivalence. There exist positive integers a, b and a constant c such that

$$\forall x \in D^r, a.\phi'(x) = b.\phi(x) + c$$

- Explicit construction. We build ϕ' as the projection to a set of variables of the solutions of an instance in VCSP(Γ). Here projection is minimisation and not existential quantification.
- Feasibility. All of Feas(Γ) is allowed.

Outline

The Expressible Cost Functions

Definition

The set of expressible cost functions of Γ is the smallest set generated by explicit construction that is closed under rational equivalence and feasibility. We denote this set $\hat{\Gamma}$.

Theorem

The problem $\mathrm{VCSP}(\hat{\Gamma})$ *is NP-hard if and only if* $\mathrm{VCSP}(\Gamma)$ *is NP-hard.*
The problem $\mathrm{VCSP}(\hat{\Gamma})$ *is tractable if and only if* $\mathrm{VCSP}(\Gamma)$ *is tractable.*

The Expressible Cost Functions

Definition

The set of expressible cost functions of Γ is the smallest set generated by explicit construction that is closed under rational equivalence and feasibility. We denote this set $\hat{\Gamma}$.

Theorem

The problem VCSP($\hat{\Gamma}$) *is NP-hard if and only if* VCSP(Γ) *is NP-hard.*
The problem VCSP($\hat{\Gamma}$) *is tractable if and only if* VCSP(Γ) *is tractable.*

Using PTIME reductions. Probably LOGSPACE

Capturing Complexity.

Theorem

For any valued constraint language Γ *with costs in* $\mathbb{Q} \cup \{\infty\}$*, and any cost function* ϕ *taking values in* $\mathbb{Q} \cup \{\infty\}$*,* $\phi \in \hat{\Gamma}$ *if and only if* $\mathrm{Pol}(\Gamma) \subseteq \mathrm{Pol}(\{\phi\})$ *and* $\mathrm{fPol}(\Gamma) \subseteq \mathrm{fPol}(\{\phi\})$*.*

Corollary

The tractability or NP-hardness of a valued constraint language Γ *with costs in* $\mathbb{Q} \cup \{\infty\}$ *is determined by its feasibility polymorphisms and fractional polymorphisms.*

Finite Valued OR Classical

Finite Valued

When the cost functions in Γ take only finite rational values, the tractability or NP-hardness is determined by the fractional polymorphisms alone.

Classical CSP

When $\Gamma = \text{Feas}(\Gamma)$ the tractability or NP-hardness is determined by the feasibility polymorphisms alone.

Having no Fractional Polymorphisms is Bad

Lemma

For any D with $|D| \geq 2$ there is an NP-hard cost function whose fractional polymorphisms are precisely the fractional projections, and which has every possible feasibility polymorphism.

Theorem

If the only fractional polymorphisms of Γ are the fractional projections then Γ is NP-hard.

Outline

1 Motivation
- The Valued CSP

2 Our Results/Contribution
- The Results
- Basic Ideas for Proof

Its all about the Costs.

- We want to build an r-ary ϕ using a gadget in VCSP(Γ).
- Write down the k distinct r-tuples for which ϕ has finite value.
- Consider this as r (possibly not distinct) k-tuples.
- Name these $\phi_1, \ldots, \phi_r$

A Gadget for ϕ.

- The variables are the set D^k.
- The domain is D.
- An s-ary cost function γ matches the columns $c_1, \ldots, c_s$ if it takes a finite value on each of the k induced s-ary rows.
- If γ matches s then create a (positive integer) variable $X(\langle S, \gamma \rangle)$.
- The X gadget is obtained by applying the constraint $\langle S, \mathrm{Feas}(\gamma) \rangle$ and $X(\langle S, \gamma \rangle)$ copies of the constraint $\langle S, \gamma \rangle$ for each $\gamma \in \Gamma$ and each S which matches γ.
- Consider the projection of the solutions onto the variables. $\phi_1, \ldots, \phi_r$.

What We Prove

If there is no assignment to X for which the X gadget builds ϕ plus some constant, then we use Farkas Lemma to get a fractional polymorphism of Γ that is not a fractional polymorphism of ϕ.

Step One

It is clear (to everyone here) that the X gadget builds something finitely equivalent to ϕ (finite valued exactly when ϕ is finite valued).

Step Two

Just write down the inequalities and equations that define when the X gadget works.

The Magic of Farkas

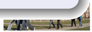
What Farkas Does

Farkas Lemma gives a certificate for a set of unsatisfiable inequalities and equalities.

The Trick

The certificate we obtain concerns weighted sums of values of the individual cost functions in Γ. It is the same weighted sum for every place where a cost function matches. These coefficients give us directly a fractional polymorphism for Γ. The n is approximately D^{D^k} where k can be as large as D^r.

The certificate also promises that the same coefficients give us an inequality on weighted sums of values of ϕ. Here it is on specific values but that is all we need. For these tuples it tells us precisely that the values of ϕ do not satisfy the fractional polymorphism inequality.



VCSP Summary

- The expressibility of a VCSP language is determined by its fractional and feasibility polymorphisms.
- The complexity of a VCSP language is determined by its fractional and feasibility polymorphisms.
- Having no (real) fractional polymorphisms makes a language NP-hard.

- Still to do...
  - What is a fractional clone?
  - Can fractional and Feasibility Polymorphisms be combined?.