Counting Matrix Partitions

Martin Dyer

School of Computing
University of Leeds

BIRS, Banff

27th November, 2014

Graph homomorphisms

- A *graph homomorphism* is a map $f : G \to H$ from an input graph $G = (V, E)$ to a target graph $H = (\mathcal{V}, \mathcal{E})$, which preserves the adjacency relation.
- i.e. we have $f : V \to \mathcal{V}$ and, for all $e = (u, v) \in E$, we must have $f(e) = \big(f(u), f(v)\big) \in \mathcal{E}$.
- Usually G is a simple graph (no loops or parallel edges), but H may have loops.
- Graph homomorphisms are also called H-*colourings* of G.
- Example: Independent sets

$G =$ and $H =$

Graph homomorphisms

- A *graph homomorphism* is a map $f : G \to H$ from an input graph $G = (V, E)$ to a target graph $H = (\mathcal{V}, \mathcal{E})$, which preserves the adjacency relation.
- i.e. we have $f : V \to \mathcal{V}$ and, for all $e = (u, v) \in E$, we must have $f(e) = \big(f(u), f(v)\big) \in \mathcal{E}$.
- Usually G is a simple graph (no loops or parallel edges), but H may have loops.
- Graph homomorphisms are also called *H-colourings* of G.
- Example: Independent sets

Graph homomorphisms

- A *graph homomorphism* is a map $f : G \to H$ from an input graph $G = (V, E)$ to a target graph $H = (\mathcal{V}, \mathcal{E})$, which preserves the adjacency relation.
- i.e. we have $f : V \to \mathcal{V}$ and, for all $e = (u, v) \in E$, we must have $f(e) = \big(f(u), f(v)\big) \in \mathcal{E}$.
- Usually G is a simple graph (no loops or parallel edges), but H may have loops.
- Graph homomorphisms are also called H*-colourings* of G.
- Example: Independent sets

$G =$ and $H =$

Graph homomorphisms

- A *graph homomorphism* is a map $f : G \to H$ from an input graph $G = (V, E)$ to a target graph $H = (\mathcal{V}, \mathcal{E})$, which preserves the adjacency relation.
- i.e. we have $f : V \to \mathcal{V}$ and, for all $e = (u, v) \in E$, we must have $f(e) = \big(f(u), f(v)\big) \in \mathcal{E}$.
- Usually G is a simple graph (no loops or parallel edges), but H may have loops.
- Graph homomorphisms are also called *H-colourings* of G.
- Example: Independent sets

$G =$ and $H =$

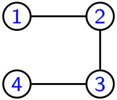
Graph homomorphisms

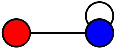
- A *graph homomorphism* is a map $f : G \to H$ from an input graph $G = (V, E)$ to a target graph $H = (\mathcal{V}, \mathcal{E})$, which preserves the adjacency relation.
- i.e. we have $f : V \to \mathcal{V}$ and, for all $e = (u, v) \in E$, we must have $f(e) = \big(f(u), f(v)\big) \in \mathcal{E}$.
- Usually G is a simple graph (no loops or parallel edges), but H may have loops.
- Graph homomorphisms are also called H-*colourings* of G.
- Example: Independent sets

$G =$ and $H =$

Graph homomorphisms

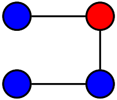
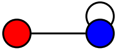
- A *graph homomorphism* is a map $f : G \to H$ from an input graph $G = (V, E)$ to a target graph $H = (\mathcal{V}, \mathcal{E})$, which preserves the adjacency relation.
- i.e. we have $f : V \to \mathcal{V}$ and, for all $e = (u, v) \in E$, we must have $f(e) = \big(f(u), f(v)\big) \in \mathcal{E}$.
- Usually G is a simple graph (no loops or parallel edges), but H may have loops.
- Graph homomorphisms are also called H-*colourings* of G.
- Example: Independent sets

$G =$ 

and $H =$ 

Graph homomorphisms

- A *graph homomorphism* is a map $f : G \to H$ from an input graph $G = (V, E)$ to a target graph $H = (\mathcal{V}, \mathcal{E})$, which preserves the adjacency relation.
- i.e. we have $f : V \to \mathcal{V}$ and, for all $e = (u, v) \in E$, we must have $f(e) = \big(f(u), f(v)\big) \in \mathcal{E}$.
- Usually G is a simple graph (no loops or parallel edges), but H may have loops.
- Graph homomorphisms are also called H-*colourings* of G.
- Example: Independent sets

$G =$  and $H =$ 

Computational complexity

We take the input size to be the number of vertices n of G, and H to be a fixed object, of constant size.

There are two questions we can ask:

- Decision: Is there any H-colouring of G?
- Counting: How many H-colourings does G have?

Hell & Nešetřil(1990) showed a *dichotomy* for the decision problem:

It is in P if H has any component which is bipartite or has a looped vertex. Otherwise it is NP-complete.

Dyer & Greenhill (2000) showed a dichotomy for the counting problem:

It is in FP if every component H is either a complete bipartite graph or a complete graph with a loop on every vertex. Otherwise it is #P-complete. The dichotomy holds for bounded-degree graphs.

Hell & Nešetřil (2004) extended this to the list version of the problem.

Computational complexity

We take the input size to be the number of vertices n of G,
and H to be a fixed object, of constant size.

There are two questions we can ask:

- Decision: Is there any H-colouring of G?
- Counting: How many H-colourings does G have?

Hell & Nešetřil(1990) showed a *dichotomy* for the decision problem:

It is in P if H has any component which is bipartite or has a looped vertex. Otherwise it is NP-complete.

Dyer & Greenhill (2000) showed a dichotomy for the counting problem:

It is in FP if every component H is either a complete bipartite graph or a complete graph with a loop on every vertex. Otherwise it is #P-complete. The dichotomy holds for bounded-degree graphs.

Hell & Nešetřil (2004) extended this to the list version of the problem.

Computational complexity

We take the input size to be the number of vertices n of G, and H to be a fixed object, of constant size.

There are two questions we can ask:

- Decision: Is there any H-colouring of G?
- Counting: How many H-colourings does G have?

Hell & Nešetřil(1990) showed a *dichotomy* for the decision problem:

It is in P if H has any component which is bipartite or has a looped vertex. Otherwise it is NP-complete.

Dyer & Greenhill (2000) showed a dichotomy for the counting problem:

It is in FP if every component H is either a complete bipartite graph or a complete graph with a loop on every vertex. Otherwise it is #P-complete. The dichotomy holds for bounded-degree graphs.

Hell & Nešetřil (2004) extended this to the list version of the problem.

Computational complexity

We take the input size to be the number of vertices n of G,
and H to be a fixed object, of constant size.

There are two questions we can ask:

- Decision: Is there any H-colouring of G?
- Counting: How many H-colourings does G have?

Hell & Nešetřil(1990) showed a *dichotomy* for the decision problem:

It is in P if H has any component which is bipartite or has a looped vertex. Otherwise it is NP-complete.

Dyer & Greenhill (2000) showed a dichotomy for the counting problem:

It is in FP if every component H is either a complete bipartite graph or a complete graph with a loop on every vertex. Otherwise it is #P-complete. The dichotomy holds for bounded-degree graphs.

Hell & Nešetřil (2004) extended this to the list version of the problem.

Computational complexity

We take the input size to be the number of vertices n of G,
and H to be a fixed object, of constant size.

There are two questions we can ask:

- Decision: Is there any H-colouring of G?
- Counting: How many H-colourings does G have?

Hell & Nešetřil(1990) showed a *dichotomy* for the decision problem:

It is in P if H has any component which is bipartite or has a looped vertex. Otherwise it is NP-complete.

Dyer & Greenhill (2000) showed a dichotomy for the counting problem:

It is in FP if every component H is either a complete bipartite graph or a complete graph with a loop on every vertex. Otherwise it is #P-complete. The dichotomy holds for bounded-degree graphs.

Hell & Nešetřil (2004) extended this to the list version of the problem.

Computational complexity

We take the input size to be the number of vertices n of G, and H to be a fixed object, of constant size.

There are two questions we can ask:

- Decision: Is there any H-colouring of G?
- Counting: How many H-colourings does G have?

Hell & Nešetřil(1990) showed a *dichotomy* for the decision problem:

It is in P if H has any component which is bipartite or has a looped vertex. Otherwise it is NP-complete.

Dyer & Greenhill (2000) showed a dichotomy for the counting problem:

It is in FP if every component H is either a complete bipartite graph or a complete graph with a loop on every vertex. Otherwise it is #P-complete. The dichotomy holds for bounded-degree graphs.

Hell & Nešetřil (2004) extended this to the list version of the problem.

Computational complexity

We take the input size to be the number of vertices n of G, and H to be a fixed object, of constant size.

There are two questions we can ask:

- Decision: Is there any H-colouring of G?
- Counting: How many H-colourings does G have?

Hell & Nešetřil(1990) showed a *dichotomy* for the decision problem:

It is in P if H has any component which is bipartite or has a looped vertex. Otherwise it is NP-complete.

Dyer & Greenhill (2000) showed a dichotomy for the counting problem:

It is in FP if every component H is either a complete bipartite graph or a complete graph with a loop on every vertex. Otherwise it is #P-complete. The dichotomy holds for bounded-degree graphs.

Hell & Nešetřil (2004) extended this to the list version of the problem.

Constraint satisfaction

CSP generalises the homomorphism notion to finite structures, and has the corresponding decision and counting problems.

Graph homomorphism problems are (essentially) the case of CSP with a single *symmetric* binary relation.

For decision CSP, a general dichotomy between P and NP-complete remains an open problem.

However, for the counting variant #CSP, Bulatov (2008) showed that there is a dichotomy between FP and #P-complete.

D & Richerby (2011) showed that the dichotomy is decidable in NP.

But no simple structural characterisation of the dichotomy is known, even for digraph homomorphisms.

Constraint satisfaction

CSP generalises the homomorphism notion to finite structures, and has the corresponding decision and counting problems.

Graph homomorphism problems are (essentially) the case of CSP with a single *symmetric* binary relation.

For decision CSP, a general dichotomy between P and NP-complete remains an open problem.

However, for the counting variant #CSP, Bulatov (2008) showed that there is a dichotomy between FP and #P-complete.

D & Richerby (2011) showed that the dichotomy is decidable in NP.

But no simple structural characterisation of the dichotomy is known, even for digraph homomorphisms.

Constraint satisfaction

CSP generalises the homomorphism notion to finite structures, and has the corresponding decision and counting problems.

Graph homomorphism problems are (essentially) the case of CSP with a single *symmetric* binary relation.

For decision CSP, a general dichotomy between P and NP-complete remains an open problem.

However, for the counting variant #CSP, Bulatov (2008) showed that there is a dichotomy between FP and #P-complete.

D & Richerby (2011) showed that the dichotomy is decidable in NP.

But no simple structural characterisation of the dichotomy is known, even for digraph homomorphisms.

Constraint satisfaction

CSP generalises the homomorphism notion to finite structures, and has the corresponding decision and counting problems.

Graph homomorphism problems are (essentially) the case of CSP with a single *symmetric* binary relation.

For decision CSP, a general dichotomy between P and NP-complete remains an open problem.

However, for the counting variant #CSP, Bulatov (2008) showed that there is a dichotomy between FP and #P-complete.

D & Richerby (2011) showed that the dichotomy is decidable in NP.

But no simple structural characterisation of the dichotomy is known, even for digraph homomorphisms.

Constraint satisfaction

CSP generalises the homomorphism notion to finite structures, and has the corresponding decision and counting problems.

Graph homomorphism problems are (essentially) the case of CSP with a single *symmetric* binary relation.

For decision CSP, a general dichotomy between P and NP-complete remains an open problem.

However, for the counting variant #CSP, Bulatov (2008) showed that there is a dichotomy between FP and #P-complete.

D & Richerby (2011) showed that the dichotomy is decidable in NP.

But no simple structural characterisation of the dichotomy is known, even for digraph homomorphisms.

Constraint satisfaction

CSP generalises the homomorphism notion to finite structures, and has the corresponding decision and counting problems.

Graph homomorphism problems are (essentially) the case of CSP with a single *symmetric* binary relation.

For decision CSP, a general dichotomy between P and NP-complete remains an open problem.

However, for the counting variant #CSP, Bulatov (2008) showed that there is a dichotomy between FP and #P-complete.

D & Richerby (2011) showed that the dichotomy is decidable in NP.

But no simple structural characterisation of the dichotomy is known, even for digraph homomorphisms.

Matrix partitions: motivation

Hereditary graph classes are those such that all induced subgraphs of any graph in the class also belongs to the class.

Structural graph theory studies properties of these classes by showing that a general graph can be decomposed in some way into “simpler” graphs.

A notable achievement of this approach has been the Strong Perfect Graph Theorem of Chudnovsky, Robertson, Seymour & Thomas (2005).

The existence of such decompositions often relates to determining whether the graph admits certain partitions of its vertex set.

Chudnovsky (2003) studied these partitions by introducing *trigraphs*, which have edges, non-edges and “don’t know” edges.

Independently, Feder, Hell, Klein & Motwani (1999) examined the complexity of computing such partitions, by representing them as *matrix partition* problems, using the same types of edges.

Matrix partitions: motivation

Hereditary graph classes are those such that all induced subgraphs of any graph in the class also belongs to the class.

Structural graph theory studies properties of these classes by showing that a general graph can be decomposed in some way into “simpler” graphs.

A notable achievement of this approach has been the Strong Perfect Graph Theorem of Chudnovsky, Robertson, Seymour & Thomas (2005).

The existence of such decompositions often relates to determining whether the graph admits certain partitions of its vertex set.

Chudnovsky (2003) studied these partitions by introducing *trigraphs*, which have edges, non-edges and “don’t know” edges.

Independently, Feder, Hell, Klein & Motwani (1999) examined the complexity of computing such partitions, by representing them as *matrix partition* problems, using the same types of edges.

Matrix partitions: motivation

Hereditary graph classes are those such that all induced subgraphs of any graph in the class also belongs to the class.

Structural graph theory studies properties of these classes by showing that a general graph can be decomposed in some way into “simpler” graphs.

A notable achievement of this approach has been the Strong Perfect Graph Theorem of Chudnovsky, Robertson, Seymour & Thomas (2005).

The existence of such decompositions often relates to determining whether the graph admits certain partitions of its vertex set.

Chudnovsky (2003) studied these partitions by introducing *trigraphs*, which have edges, non-edges and “don’t know” edges.

Independently, Feder, Hell, Klein & Motwani (1999) examined the complexity of computing such partitions, by representing them as *matrix partition* problems, using the same types of edges.

Matrix partitions: motivation

Hereditary graph classes are those such that all induced subgraphs of any graph in the class also belongs to the class.

Structural graph theory studies properties of these classes by showing that a general graph can be decomposed in some way into “simpler” graphs.

A notable achievement of this approach has been the Strong Perfect Graph Theorem of Chudnovsky, Robertson, Seymour & Thomas (2005).

The existence of such decompositions often relates to determining whether the graph admits certain partitions of its vertex set.

Chudnovsky (2003) studied these partitions by introducing *trigraphs*, which have edges, non-edges and “don't know” edges.

Independently, Feder, Hell, Klein & Motwani (1999) examined the complexity of computing such partitions, by representing them as *matrix partition* problems, using the same types of edges.

Matrix partitions: motivation

Hereditary graph classes are those such that all induced subgraphs of any graph in the class also belongs to the class.

Structural graph theory studies properties of these classes by showing that a general graph can be decomposed in some way into “simpler” graphs.

A notable achievement of this approach has been the Strong Perfect Graph Theorem of Chudnovsky, Robertson, Seymour & Thomas (2005).

The existence of such decompositions often relates to determining whether the graph admits certain partitions of its vertex set.

Chudnovsky (2003) studied these partitions by introducing *trigraphs*, which have edges, non-edges and “don't know” edges.

Independently, Feder, Hell, Klein & Motwani (1999) examined the complexity of computing such partitions, by representing them as *matrix partition* problems, using the same types of edges.

Matrix partitions: motivation

Hereditary graph classes are those such that all induced subgraphs of any graph in the class also belongs to the class.

Structural graph theory studies properties of these classes by showing that a general graph can be decomposed in some way into “simpler” graphs.

A notable achievement of this approach has been the Strong Perfect Graph Theorem of Chudnovsky, Robertson, Seymour & Thomas (2005).

The existence of such decompositions often relates to determining whether the graph admits certain partitions of its vertex set.

Chudnovsky (2003) studied these partitions by introducing *trigraphs*, which have edges, non-edges and “don’t know” edges.

Independently, Feder, Hell, Klein & Motwani (1999) examined the complexity of computing such partitions, by representing them as *matrix partition* problems, using the same types of edges.

Matrix partitions

Let M be a symmetric matrix in $\{0,1,*\}^{D\times D}$. An M-*partition* of an undirected graph $G=(V,E)$ is a partition of V into parts (some of which may be empty) labelled by the elements of D. Thus the partition is a function $\sigma\colon V\to D$ where $\sigma(v)$ labels vertex v with its part.

It satisfies the following property: For all pairs of distinct vertices u and v,

- $M_{\sigma(u),\sigma(v)}\in\{1,*\}$ if $(u,v)\in E$ and
- $M_{\sigma(u),\sigma(v)}\in\{0,*\}$ if $(u,v)\notin E$.

Thus, if $M_{i,j}=0$, no edges are permitted between vertices in parts i and j and, if $M_{i,j}=1$, then all edges must be present between the two parts. If $M_{i,j}=*$, there is no restriction on edges between parts i and j.

Note that M is a trigraph, but G is a graph, so this is not a graph homomorphism problem. Also, the symmetrical status of edges and non-edges means that it is not obviously a CSP.

Matrix partitions

Let M be a symmetric matrix in $\{0, 1, *\}^{D \times D}$. An *M-partition* of an undirected graph $G = (V, E)$ is a partition of V into parts (some of which may be empty) labelled by the elements of D. Thus the partition is a function $\sigma \colon V \to D$ where $\sigma(v)$ labels vertex v with its part.

It satisfies the following property: For all pairs of distinct vertices u and v,

- $M_{\sigma(u),\sigma(v)} \in \{1, *\}$ if $(u, v) \in E$ and
- $M_{\sigma(u),\sigma(v)} \in \{0, *\}$ if $(u, v) \notin E$.

Thus, if $M_{i,j} = 0$, no edges are permitted between vertices in parts i and j and, if $M_{i,j} = 1$, then all edges must be present between the two parts. If $M_{i,j} = *$, there is no restriction on edges between parts i and j.

Note that M is a trigraph, but G is a graph, so this is not a graph homomorphism problem. Also, the symmetrical status of edges and non-edges means that it is not obviously a CSP.

Matrix partitions

Let M be a symmetric matrix in $\{0, 1, *\}^{D \times D}$. An *M-partition* of an undirected graph $G = (V, E)$ is a partition of V into parts (some of which may be empty) labelled by the elements of D. Thus the partition is a function $\sigma \colon V \to D$ where $\sigma(v)$ labels vertex v with its part.

It satisfies the following property: For all pairs of distinct vertices u and v,

- $M_{\sigma(u),\sigma(v)} \in \{1, *\}$ if $(u, v) \in E$ and
- $M_{\sigma(u),\sigma(v)} \in \{0, *\}$ if $(u, v) \notin E$.

Thus, if $M_{i,j} = 0$, no edges are permitted between vertices in parts i and j and, if $M_{i,j} = 1$, then all edges must be present between the two parts. If $M_{i,j} = *$, there is no restriction on edges between parts i and j.

Note that M is a trigraph, but G is a graph, so this is not a graph homomorphism problem. Also, the symmetrical status of edges and non-edges means that it is not obviously a CSP.

Matrix partitions

Let M be a symmetric matrix in $\{0, 1, *\}^{D \times D}$. An *M-partition* of an undirected graph $G = (V, E)$ is a partition of V into parts (some of which may be empty) labelled by the elements of D. Thus the partition is a function $\sigma \colon V \to D$ where $\sigma(v)$ labels vertex v with its part.

It satisfies the following property: For all pairs of distinct vertices u and v,

- $M_{\sigma(u),\sigma(v)} \in \{1, *\}$ if $(u, v) \in E$ and
- $M_{\sigma(u),\sigma(v)} \in \{0, *\}$ if $(u, v) \notin E$.

Thus, if $M_{i,j} = 0$, no edges are permitted between vertices in parts i and j and, if $M_{i,j} = 1$, then all edges must be present between the two parts. If $M_{i,j} = *$, there is no restriction on edges between parts i and j.

Note that M is a trigraph, but G is a graph, so this is not a graph homomorphism problem. Also, the symmetrical status of edges and non-edges means that it is not obviously a CSP.

Example

A *skew cutset* of a connected graph $G = (V, E)$ is a pair of disjoint sets $A, B \subset V$ such that $A \cup B$ is a cutset (deleting A and B disconnects G) and G contains all possible edges between A and B. Then skew cutsets correspond to M-partitions for the matrix

$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} * & 1 & * & * \\ 1 & * & * & * \\ * & * & * & 0 \\ * & * & 0 & * \end{bmatrix} \end{array}.$$

The rows and columns correspond to parts A, B, C and D, respectively. Consider an M-partition in which every part is non-empty. $M_{A,B} = 1$ so G must contain every edge between A and B. The rest of the graph must be assigned to parts C and D, but with no edges allowed between them. So each must be a union of components of $G \setminus (A \cup B)$. Therefore, the partition corresponds exactly to a skew cutset.

Example

A *skew cutset* of a connected graph $G = (V, E)$ is a pair of disjoint sets $A, B \subset V$ such that $A \cup B$ is a cutset (deleting A and B disconnects G) and G contains all possible edges between A and B. Then skew cutsets correspond to M-partitions for the matrix

$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \left[\begin{array}{cccc} * & 1 & * & * \\ 1 & * & * & * \\ * & * & * & 0 \\ * & * & 0 & * \end{array} \right] \end{array}.$$

The rows and columns correspond to parts A, B, C and D, respectively. Consider an M-partition in which every part is non-empty. $M_{A,B} = 1$ so G must contain every edge between A and B. The rest of the graph must be assigned to parts C and D, but with no edges allowed between them. So each must be a union of components of $G \setminus (A \cup B)$. Therefore, the partition corresponds exactly to a skew cutset.

Example

A *skew cutset* of a connected graph $G = (V, E)$ is a pair of disjoint sets $A, B \subset V$ such that $A \cup B$ is a cutset (deleting A and B disconnects G) and G contains all possible edges between A and B. Then skew cutsets correspond to M-partitions for the matrix

$$M = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} * & 1 & * & * \\ 1 & * & * & * \\ * & * & * & 0 \\ * & * & 0 & * \end{bmatrix} \end{array}.$$

The rows and columns correspond to parts A, B, C and D, respectively. Consider an M-partition in which every part is non-empty. $M_{A,B} = 1$ so G must contain every edge between A and B. The rest of the graph must be assigned to parts C and D, but with no edges allowed between them. So each must be a union of components of $G \setminus (A \cup B)$. Therefore, the partition corresponds exactly to a skew cutset.

Decision

Feder, Hell, Klein & Motwani (1999) examined the decision version of the matrix partitions problems.

They were unable to prove a dichotomy, but showed that dichotomy for CSP would imply a weaker dichotomy for this problem.

The same authors (2003) studied the *list* version of the problem, where all unary relations are allowed. (This has been called the *conservative* case of CSP.)

They also considered the case where the lists are *restricted* to lie in some given subset $\mathcal{L}$ of 2^D.

They were able to give an almost complete dichotomy between P and NP-complete for $|D| \leq 4$.

For the general case, they conjectured only a weaker dichotomy, between quasipolynomial time and NP-complete.

Decision

Feder, Hell, Klein & Motwani (1999) examined the decision version of the matrix partitions problems.

They were unable to prove a dichotomy, but showed that dichotomy for CSP would imply a weaker dichotomy for this problem.

The same authors (2003) studied the *list* version of the problem, where all unary relations are allowed. (This has been called the *conservative* case of CSP.)

They also considered the case where the lists are *restricted* to lie in some given subset $\mathcal{L}$ of 2^D.

They were able to give an almost complete dichotomy between P and NP-complete for $|D| \leq 4$.

For the general case, they conjectured only a weaker dichotomy, between quasipolynomial time and NP-complete.

Decision

Feder, Hell, Klein & Motwani (1999) examined the decision version of the matrix partitions problems.

They were unable to prove a dichotomy, but showed that dichotomy for CSP would imply a weaker dichotomy for this problem.

The same authors (2003) studied the *list* version of the problem, where all unary relations are allowed. (This has been called the *conservative* case of CSP.)

They also considered the case where the lists are *restricted* to lie in some given subset $\mathcal{L}$ of 2^D.

They were able to give an almost complete dichotomy between P and NP-complete for $|D| \leq 4$.

For the general case, they conjectured only a weaker dichotomy, between quasipolynomial time and NP-complete.

Decision

Feder, Hell, Klein & Motwani (1999) examined the decision version of the matrix partitions problems.

They were unable to prove a dichotomy, but showed that dichotomy for CSP would imply a weaker dichotomy for this problem.

The same authors (2003) studied the *list* version of the problem, where all unary relations are allowed. (This has been called the *conservative* case of CSP.)

They also considered the case where the lists are *restricted* to lie in some given subset $\mathcal{L}$ of 2^D.

They were able to give an almost complete dichotomy between P and NP-complete for $|D| \leq 4$.

For the general case, they conjectured only a weaker dichotomy, between quasipolynomial time and NP-complete.

Decision

Feder, Hell, Klein & Motwani (1999) examined the decision version of the matrix partitions problems.

They were unable to prove a dichotomy, but showed that dichotomy for CSP would imply a weaker dichotomy for this problem.

The same authors (2003) studied the *list* version of the problem, where all unary relations are allowed. (This has been called the *conservative* case of CSP.)

They also considered the case where the lists are *restricted* to lie in some given subset $\mathcal{L}$ of 2^D.

They were able to give an almost complete dichotomy between P and NP-complete for $|D| \leq 4$.

For the general case, they conjectured only a weaker dichotomy, between quasipolynomial time and NP-complete.

Decision

Feder, Hell, Klein & Motwani (1999) examined the decision version of the matrix partitions problems.

They were unable to prove a dichotomy, but showed that dichotomy for CSP would imply a weaker dichotomy for this problem.

The same authors (2003) studied the *list* version of the problem, where all unary relations are allowed. (This has been called the *conservative* case of CSP.)

They also considered the case where the lists are *restricted* to lie in some given subset $\mathcal{L}$ of 2^D.

They were able to give an almost complete dichotomy between P and NP-complete for $|D| \leq 4$.

For the general case, they conjectured only a weaker dichotomy, between quasipolynomial time and NP-complete.

Counting

Hell, Hermann & Nevisi (2012) proved a dichotomy for the counting problem in the cases $|D| = 2$ and $|D| = 3$.

Unfortunately, most of the the tools developed for the #CSP dichotomy do not seem to be applicable to this problem.

So their proof was based only on the following:

- Interchanging the 0's and 1's in M gives the same problem on the complement of G, so it is not necessary to consider both,
- If there are no 1's (or no 0's) in M, then the problem is graph homomorphism with H given by the $*$ edges in M.
- Connecting a simple disjoint graph (e.g. a clique) to every vertex of G. This allows the count for M to be expressed as a sum of counts for smaller matrices. (This technique was used in D & Greenhill's proof of the H-colouring dichotomy.)

They conjectured that the dichotomy would hold for $|D| = 4$.

Counting

Hell, Hermann & Nevisi (2012) proved a dichotomy for the counting problem in the cases $|D| = 2$ and $|D| = 3$.

Unfortunately, most of the the tools developed for the #CSP dichotomy do not seem to be applicable to this problem.

So their proof was based only on the following:

- Interchanging the 0's and 1's in M gives the same problem on the complement of G, so it is not necessary to consider both,
- If there are no 1's (or no 0's) in M, then the problem is graph homomorphism with H given by the $*$ edges in M.
- Connecting a simple disjoint graph (e.g. a clique) to every vertex of G. This allows the count for M to be expressed as a sum of counts for smaller matrices. (This technique was used in D & Greenhill's proof of the H-colouring dichotomy.)

They conjectured that the dichotomy would hold for $|D| = 4$.

Counting

Hell, Hermann & Nevisi (2012) proved a dichotomy for the counting problem in the cases $|D| = 2$ and $|D| = 3$.

Unfortunately, most of the the tools developed for the #CSP dichotomy do not seem to be applicable to this problem.

So their proof was based only on the following:

- Interchanging the 0's and 1's in M gives the same problem on the complement of G, so it is not necessary to consider both,
- If there are no 1's (or no 0's) in M, then the problem is graph homomorphism with H given by the $*$ edges in M.
- Connecting a simple disjoint graph (e.g. a clique) to every vertex of G. This allows the count for M to be expressed as a sum of counts for smaller matrices. (This technique was used in D & Greenhill's proof of the H-colouring dichotomy.)

They conjectured that the dichotomy would hold for $|D| = 4$.

Counting

Hell, Hermann & Nevisi (2012) proved a dichotomy for the counting problem in the cases $|D| = 2$ and $|D| = 3$.

Unfortunately, most of the the tools developed for the #CSP dichotomy do not seem to be applicable to this problem.

So their proof was based only on the following:

- Interchanging the 0's and 1's in M gives the same problem on the complement of G, so it is not necessary to consider both,
- If there are no 1's (or no 0's) in M, then the problem is graph homomorphism with H given by the $*$ edges in M.
- Connecting a simple disjoint graph (e.g. a clique) to every vertex of G. This allows the count for M to be expressed as a sum of counts for smaller matrices. (This technique was used in D & Greenhill's proof of the H-colouring dichotomy.)

They conjectured that the dichotomy would hold for $|D| = 4$.

Counting

Hell, Hermann & Nevisi (2012) proved a dichotomy for the counting problem in the cases $|D| = 2$ and $|D| = 3$.

Unfortunately, most of the the tools developed for the #CSP dichotomy do not seem to be applicable to this problem.

So their proof was based only on the following:

- Interchanging the 0's and 1's in M gives the same problem on the complement of G, so it is not necessary to consider both,
- If there are no 1's (or no 0's) in M, then the problem is graph homomorphism with H given by the $*$ edges in M.
- Connecting a simple disjoint graph (e.g. a clique) to every vertex of G. This allows the count for M to be expressed as a sum of counts for smaller matrices. (This technique was used in D & Greenhill's proof of the H-colouring dichotomy.)

They conjectured that the dichotomy would hold for $|D| = 4$.

Counting

Hell, Hermann & Nevisi (2012) proved a dichotomy for the counting problem in the cases $|D| = 2$ and $|D| = 3$.

Unfortunately, most of the the tools developed for the #CSP dichotomy do not seem to be applicable to this problem.

So their proof was based only on the following:

- Interchanging the 0's and 1's in M gives the same problem on the complement of G, so it is not necessary to consider both,
- If there are no 1's (or no 0's) in M, then the problem is graph homomorphism with H given by the $*$ edges in M.
- Connecting a simple disjoint graph (e.g. a clique) to every vertex of G. This allows the count for M to be expressed as a sum of counts for smaller matrices. (This technique was used in D & Greenhill's proof of the H-colouring dichotomy.)

They conjectured that the dichotomy would hold for $|D| = 4$.

Counting

Hell, Hermann & Nevisi (2012) proved a dichotomy for the counting problem in the cases $|D| = 2$ and $|D| = 3$.

Unfortunately, most of the the tools developed for the #CSP dichotomy do not seem to be applicable to this problem.

So their proof was based only on the following:

- Interchanging the 0's and 1's in M gives the same problem on the complement of G, so it is not necessary to consider both,
- If there are no 1's (or no 0's) in M, then the problem is graph homomorphism with H given by the $*$ edges in M.
- Connecting a simple disjoint graph (e.g. a clique) to every vertex of G. This allows the count for M to be expressed as a sum of counts for smaller matrices. (This technique was used in D & Greenhill's proof of the H-colouring dichotomy.)

They conjectured that the dichotomy would hold for $|D| = 4$.

The case $|D| = 4$

D, Goldberg & Richerby (2014) proved the dichotomy for $|D| = 4$, by extending the methods used by Hell, Hermann & Nevisi.

There are some technical difficulties, but essentially this was done by examining all cases up to symmetries.

This was done by using a computer program to handle almost all cases, and then solving the remaining cases "by hand". In fact, there were only 6 of these exceptional cases.

The same proof method appears possible for $|D| = 5$, but has not been carried out, since the number of cases increases exponentially with $|D|$, and we cannot estimate how may exceptional cases may need to be handled individually.

We conjecture a dichotomy for all D, and we conjecture a specific (decidable) criterion for the dichotomy. based on the *list* version of counting M-partitions.

The case $|D| = 4$

D, Goldberg & Richerby (2014) proved the dichotomy for $|D| = 4$, by extending the methods used by Hell, Hermann & Nevisi.

There are some technical difficulties, but essentially this was done by examining all cases up to symmetries.

This was done by using a computer program to handle almost all cases, and then solving the remaining cases “by hand”. In fact, there were only 6 of these exceptional cases.

The same proof method appears possible for $|D| = 5$, but has not been carried out, since the number of cases increases exponentially with $|D|$, and we cannot estimate how may exceptional cases may need to be handled individually.

We conjecture a dichotomy for all D, and we conjecture a specific (decidable) criterion for the dichotomy. based on the *list* version of counting M-partitions.

The case $|D| = 4$

D, Goldberg & Richerby (2014) proved the dichotomy for $|D| = 4$, by extending the methods used by Hell, Hermann & Nevisi.

There are some technical difficulties, but essentially this was done by examining all cases up to symmetries.

This was done by using a computer program to handle almost all cases, and then solving the remaining cases "by hand". In fact, there were only 6 of these exceptional cases.

The same proof method appears possible for $|D| = 5$, but has not been carried out, since the number of cases increases exponentially with $|D|$, and we cannot estimate how may exceptional cases may need to be handled individually.

We conjecture a dichotomy for all D, and we conjecture a specific (decidable) criterion for the dichotomy. based on the *list* version of counting M-partitions.

The case $|D| = 4$

D, Goldberg & Richerby (2014) proved the dichotomy for $|D| = 4$, by extending the methods used by Hell, Hermann & Nevisi.

There are some technical difficulties, but essentially this was done by examining all cases up to symmetries.

This was done by using a computer program to handle almost all cases, and then solving the remaining cases "by hand". In fact, there were only 6 of these exceptional cases.

The same proof method appears possible for $|D| = 5$, but has not been carried out, since the number of cases increases exponentially with $|D|$, and we cannot estimate how may exceptional cases may need to be handled individually.

We conjecture a dichotomy for all D, and we conjecture a specific (decidable) criterion for the dichotomy. based on the *list* version of counting M-partitions.

The case $|D| = 4$

D, Goldberg & Richerby (2014) proved the dichotomy for $|D| = 4$, by extending the methods used by Hell, Hermann & Nevisi.

There are some technical difficulties, but essentially this was done by examining all cases up to symmetries.

This was done by using a computer program to handle almost all cases, and then solving the remaining cases "by hand". In fact, there were only 6 of these exceptional cases.

The same proof method appears possible for $|D| = 5$, but has not been carried out, since the number of cases increases exponentially with $|D|$, and we cannot estimate how may exceptional cases may need to be handled individually.

We conjecture a dichotomy for all D, and we conjecture a specific (decidable) criterion for the dichotomy. based on the *list* version of counting M-partitions.

The case $|D| = 4$

Göbel, Goldberg, McQuillan, Richerby & Yamakami (2013) proved a dichotomy for the M-partition counting problem with lists. We will suppose that all lists in 2^D are allowed, though GGMRY gave a slightly stronger result (for "subset-closed" lists).

The lists allow submatrices of M to be selected. GGMRY focussed on what they called *pure* submatrices: those containing only 0's and $*$'s, or 1's and $*$'s, (since these relate to graph homomorphisms).

They were then able to express the problem as a #CSP with a template containing all these binary relations and all unary relations.

A decidable dichotomy now follows from Bulatov, D, Richerby, but GGMRY were able to give a somewhat simpler explicit form for the dichotomy, based on composing the relations corresponding to the pure submatrices. They also showed that the resulting criterion is decidable in NP.

The case $|D| = 4$

Göbel, Goldberg, McQuillan, Richerby & Yamakami (2013) proved a dichotomy for the M-partition counting problem with lists. We will suppose that all lists in 2^D are allowed, though GGMRY gave a slightly stronger result (for "subset-closed" lists).

The lists allow submatrices of M to be selected. GGMRY focussed on what they called *pure* submatrices: those containing only 0's and $*$'s, or 1's and $*$'s, (since these relate to graph homomorphisms).

They were then able to express the problem as a #CSP with a template containing all these binary relations and all unary relations.

A decidable dichotomy now follows from Bulatov, D, Richerby, but GGMRY were able to give a somewhat simpler explicit form for the dichotomy, based on composing the relations corresponding to the pure submatrices. They also showed that the resulting criterion is decidable in NP.

The case $|D| = 4$

Göbel, Goldberg, McQuillan, Richerby & Yamakami (2013) proved a dichotomy for the M-partition counting problem with lists. We will suppose that all lists in 2^D are allowed, though GGMRY gave a slightly stronger result (for "subset-closed" lists).

The lists allow submatrices of M to be selected. GGMRY focussed on what they called *pure* submatrices: those containing only 0's and $*$'s, or 1's and $*$'s, (since these relate to graph homomorphisms).

They were then able to express the problem as a #CSP with a template containing all these binary relations and all unary relations.

A decidable dichotomy now follows from Bulatov, D, Richerby, but GGMRY were able to give a somewhat simpler explicit form for the dichotomy, based on composing the relations corresponding to the pure submatrices. They also showed that the resulting criterion is decidable in NP.

The case $|D| = 4$

Göbel, Goldberg, McQuillan, Richerby & Yamakami (2013) proved a dichotomy for the M-partition counting problem with lists. We will suppose that all lists in 2^D are allowed, though GGMRY gave a slightly stronger result (for "subset-closed" lists).

The lists allow submatrices of M to be selected. GGMRY focussed on what they called *pure* submatrices: those containing only 0's and $*$'s, or 1's and $*$'s, (since these relate to graph homomorphisms).

They were then able to express the problem as a #CSP with a template containing all these binary relations and all unary relations.

A decidable dichotomy now follows from Bulatov, D, Richerby, but GGMRY were able to give a somewhat simpler explicit form for the dichotomy, based on composing the relations corresponding to the pure submatrices. They also showed that the resulting criterion is decidable in NP.

Conjectures

In fact, the GGMRY criterion describes the dichotomy *without* lists in the cases $|D| = 2, 3, 4$, although we can currently only show this by checking all the cases for each $|D|$.

We conjecture that this criterion gives the dichotomy for all $|D|$.

Finally, observe that the M-partition idea can be generalised, in a fairly obvious way, to any CSP problem. We conjecture that this general "#CSP-partition" problem will also have a dichotomy.

Conjectures

In fact, the GGMRY criterion describes the dichotomy *without* lists in the cases $|D| = 2, 3, 4$, although we can currently only show this by checking all the cases for each $|D|$.

We conjecture that this criterion gives the dichotomy for all $|D|$.

Finally, observe that the M-partition idea can be generalised, in a fairly obvious way, to any CSP problem. We conjecture that this general "#CSP-partition" problem will also have a dichotomy.

Conjectures

In fact, the GGMRY criterion describes the dichotomy *without* lists in the cases $|D| = 2, 3, 4$, although we can currently only show this by checking all the cases for each $|D|$.

We conjecture that this criterion gives the dichotomy for all $|D|$.

Finally, observe that the M-partition idea can be generalised, in a fairly obvious way, to any CSP problem. We conjecture that this general "#CSP-partition" problem will also have a dichotomy.

Conjectures

In fact, the GGMRY criterion describes the dichotomy *without* lists in the cases $|D| = 2, 3, 4$, although we can currently only show this by checking all the cases for each $|D|$.

We conjecture that this criterion gives the dichotomy for all $|D|$.

Finally, observe that the M-partition idea can be generalised, in a fairly obvious way, to any CSP problem. We conjecture that this general "#CSP-partition" problem will also have a dichotomy.

Thank You!