A DICHOTOMY THEOREM FOR THE GENERAL MINIMUM COST HOMOMORPHISM PROBLEM

RUSTEM TAKHANOV

1 Department of Computer and Information Science, Linköping University
E-mail address: g-rusta@ida.liu.se
E-mail address: takhanov@mail.ru

ABSTRACT. In the constraint satisfaction problem (CSP), the aim is to find an assignment of values to a set of variables subject to specified constraints. In the minimum cost homomorphism problem (MinHom), one is additionally given weights $c_{va}$ for every variable $v$ and value $a$, and the aim is to find an assignment $f$ to the variables that minimizes $\sum_v c_{vf(v)}$. Let $\text{MinHom}(\Gamma)$ denote the $\text{MinHom}$ problem parameterized by a set $\Gamma$ of predicates allowed for constraints. $\text{MinHom}(\Gamma)$ is related to many well-studied combinatorial optimization problems, and concrete applications can be found in, for instance, defence logistics and machine learning. We show that $\text{MinHom}(\Gamma)$ can be studied by using algebraic methods similar to those used for CSPs. With the aid of algebraic techniques, we classify the computational complexity of $\text{MinHom}(\Gamma)$ for all choices of $\Gamma$. Our result settles a general dichotomy conjecture previously resolved only for certain classes of directed graphs, [Gutin, Hell, Rafiey, Yeo, European J. of Combinatorics, 2008].

1. Introduction

Constraint satisfaction problems (CSP) are a natural way of formalizing a large number of computational problems arising in combinatorial optimization, artificial intelligence, and database theory. This problem has the following two equivalent formulations: (1) to find an assignment of values to a given set of variables, subject to constraints on the values that can be assigned simultaneously to specified subsets of variables, and (2) to find a homomorphism between two finite relational structures. Applications of CSPs arise in propositional logic, database and graph theory, scheduling and many other areas. During the past 30 years, CSP and its subproblems have been intensively studied by computer scientists and mathematicians. Considerable attention has been paid to the case where the constraints are restricted to a given finite set of relations $\Gamma$, called a constraint language [5, 11, 20, 28]. For example, when $\Gamma$ is a constraint language over the boolean set $\{0,1\}$ with four ternary predicates $x \lor y \lor z$, $\overline{x} \lor y \lor z$, $\overline{x} \lor \overline{y} \lor \overline{z}$, $\overline{x} \lor \overline{y} \lor \overline{z}$ we obtain 3-SAT. This direction of research has been mainly concerned with the computational complexity of $\text{CSP}(\Gamma)$ as a function of $\Gamma$. It has been shown that the complexity of $\text{CSP}(\Gamma)$ is strongly connected with relational clones of universal algebra [20]. For every constraint language $\Gamma$, it has been conjectured that $\text{CSP}(\Gamma)$ is either in P or NP-complete [11].

1998 ACM Subject Classification: F.4.1, G.2.2, I.2.6.

Key words and phrases: minimum cost homomorphisms problem, relational clones, constraint satisfaction problem, perfect graphs, supervised learning.
In the minimum cost homomorphism problem ($\text{MinHom}$), we are given variables subject to constraints and, additionally, costs on variable/value pairs. Now, the task is not just to find any satisfying assignment to the variables, but one that minimizes the total cost.

**Definition 1.1.** Suppose we are given a finite domain set $A$ and a finite constraint language $\Gamma \subseteq \bigcup_{k=1}^{\infty} 2^{A^k}$. Denote by $\text{MinHom}(\Gamma)$ the following minimization task:

**Instance:** A first-order formula $\Phi(x_1, \ldots, x_n) = \bigwedge_{i=1}^{N} \rho_i(y_{i1}, \ldots, y_{im_i})$, $\rho_i \in \Gamma$, $y_{ij} \in \{x_1, \ldots, x_n\}$, and weights $w_{ia} \in \mathbb{N}$, $1 \leq i \leq n, a \in A$.

**Solution:** Assignment $f : \{x_1, \ldots, x_n\} \rightarrow A$, that satisfies the formula $\Phi$. If there is no such assignment, then indicate it.

**Measure:** $\sum_{i=1}^{n} w_{if(x_i)}$.

**Remark 1.2.** Note that when we require weights to be positive we do not lose generality, since $\text{MinHom}(\Gamma)$ with arbitrary weights can be polynomial-time reduced to $\text{MinHom}(\Gamma)$ with positive weights by the following trick: we can add $s$ to all weights, where $s$ is some integer. This trick only adds $ns$ to the value of the optimized measure. Hence, we can make all weights negative, and $\text{MinHom}(\Gamma)$ modified this way is equivalent to maximization but with positive weights only. This remark explains why both names $\text{MinHom}$ and $\text{MaxHom}$ can be allowed, though we prefer $\text{MinHom}$ due to historical reasons.

$\text{MinHom}$ was introduced in [18] where it was motivated by a real-world problem in defence logistics. The question for which directed graphs $H$ the problem $\text{MinHom}(\{H\})$ is polynomial-time solvable was considered in [15, 16, 17, 18, 19]. In this paper, we approach the problem in its most general form by algebraic methods and give a complete algebraic characterization of tractable constraint languages. From this characterization, we obtain a dichotomy for $\text{MinHom}$, i.e., if $\text{MinHom}(\Gamma)$ is not polynomial-time solvable, then it is NP-hard. Thus, our dichotomy implies the dichotomy for directed graphs.

In Section 2, we present some preliminaries together with results connecting the complexity of $\text{MinHom}$ with conservative algebras. The main dichotomy theorem is stated in Section 3 and its proof is divided into several parts which can be found in Sections 4-8. The NP-hardness results are collected in Section 4 followed by the building blocks for the tractability result: existence of majority polymorphisms (Section 5) and connections with optimization in perfect graphs (Section 6). Section 7 introduces the concept of arithmetical deadlocks which lay the foundation for the final proof in Section 8. In Section 9 we reformulate our main result in terms of relational clones. Finally, in Section 10 we explain the relation of our results to previous research and present directions for future research.

2. Algebraic structure of tractable constraint languages

Recall that an optimization problem $A$ is called NP-hard if some NP-complete language can be recognized in polynomial time with the aid of an oracle for $A$. We assume that $P \neq NP$.

**Definition 2.1.** Suppose we are given a finite set $A$ and a constraint language $\Gamma \subseteq \bigcup_{k=1}^{\infty} 2^{A^k}$. The language $\Gamma$ is said to be *tractable* if, for every finite subset $\Gamma' \subseteq \Gamma$, $\text{MinHom}(\Gamma')$ is
Definition 2.2. Let \( \rho \subseteq A^m \) and \( f : A^n \to A \). We say that the function (operation) \( f \) preserves the predicate \( \rho \) if, for every \((x_1^i, \ldots, x_m^i) \in \rho, 1 \leq i \leq n, \) we have that \( (f(x_1^1, \ldots, x_m^1), \ldots, f(x_1^n, \ldots, x_m^n)) \in \rho. \)

For a constraint language \( \Gamma \), let \( \text{Pol}(\Gamma) \) denote the set of operations preserving all predicates in \( \Gamma. \) Throughout the paper, we let \( A \) denote a finite domain and \( \Gamma \) a constraint language over \( A. \) We assume the domain \( A \) to be finite.

Definition 2.3. A constraint language \( \Gamma \) is called a relational clone if it contains every predicate expressible by a first-order formula involving only

- predicates from \( \Gamma \cup \{=^A\}; \)
- conjunction; and
- existential quantification.

First-order formulas involving only conjunction and existential quantification are often called primitive positive (pp) formulas. For a given constraint language \( \Gamma, \) the set of all predicates that can be described by pp-formulas over \( \Gamma \cup \{=^A\} \) is called the closure of \( \Gamma \) and is denoted by \( \langle \Gamma \rangle. \) It is easy to see that \( \Gamma \) is a relational clone if and only if \( \Gamma = \langle \Gamma \rangle. \)

For a set of operations \( F \) on \( A, \) let \( \text{Inv}(F) \) denote the set of predicates preserved under the operations of \( F. \) Obviously, \( \text{Inv}(F) \) is a relational clone. The next result is well-known [3, 12].

Theorem 2.4. For a constraint language \( \Gamma \) over a finite set \( A, \) \( \langle \Gamma \rangle = \text{Inv}(\text{Pol}(\Gamma)). \)

Theorem 2.4 gives another description of the closure of a constraint language. The role of this description will be clear after we prove the following theorem.

Theorem 2.5. For any finite constraint language \( \Gamma \) and any finite \( \Gamma' \subseteq \langle \Gamma \rangle, \) there is a polynomial time reduction from \( \text{MinHom}(\Gamma') \) to \( \text{MinHom}(\Gamma). \)

Proof. Since any predicate from \( \Gamma' \) can be viewed as a pp-formula with predicates in \( \Gamma, \) an input formula to \( \text{MinHom}(\Gamma') \) can be represented in the form \( \Phi(x_1, \ldots, x_n) = \bigwedge_{i=1}^N \exists z_{i1}, \ldots, z_{in_m} \Phi_i(y_{i1}, \ldots, y_{in_m}, z_{i1}, \ldots, z_{in_m}), \) where \( y_{ij} \in \{x_1, \ldots, x_n\} \) and \( \Phi_i \) is a first-order formula involving only predicates in \( \Gamma, \) equality predicate, and conjunction.

Obviously, this formula is equivalent to \( \exists z_{i1}, \ldots, z_{Nm_N} \bigwedge_{i=1}^N \Phi_i(y_{i1}, \ldots, y_{in_m}, z_{i1}, \ldots, z_{in_m}). \)

\( \bigwedge_{i=1}^N \Phi_i(y_{i1}, \ldots, y_{in_m}, z_{i1}, \ldots, z_{in_m}) \) can be considered as an instance of \( \text{MinHom}(\Gamma \cup \{=^A\}) \) with variables \( x_1, \ldots, x_n, z_{11}, \ldots, z_{Nm_N} \) where weights \( w_{ij} \) will remain the same and for additional variables \( z_{kl} \) we define \( w_{z_{kl}} = 0. \) By solving \( \text{MinHom}(\Gamma \cup \{=^A\}) \) with the described input, we can find a solution of the initial \( \text{MinHom}(\Gamma') \) problem. It is easy to see that the number of added variables is bounded by a polynomial in \( n. \) So this reduction can be carried out in polynomial time. Finally, \( \text{MinHom}(\Gamma \cup \{=^A\}) \) can be reduced polynomially to \( \text{MinHom}(\Gamma) \) because an equality constraint for a pair of variables is equivalent to identification of these variables and updating all weights by formula:

\[
\begin{align*}
  w'_{v_0} &= \sum_{v' \text{ identified with } v} w_{v_0}.
\end{align*}
\]
The previous theorem tells us that the complexity of \( \text{MinHom} (\Gamma) \) is determined by \( \text{Inv} (\text{Pol} (\Gamma)) \), i.e., by \( \text{Pol} (\Gamma) \). That is why we will be concerned with the classification of sets of operations \( F \) for which \( \text{Inv} (F) \) is a tractable constraint language.

**Definition 2.6.** An algebra is an ordered pair \( A = (A, F) \) such that \( A \) is a nonempty set (called a universe) and \( F \) is a family of finitary operations on \( A \). An algebra with a finite universe is referred to as a finite algebra.

**Definition 2.7.** An algebra \( A = (A, F) \) is called tractable if \( \text{Inv} (F) \) is a tractable constraint language and \( A \) is called NP-hard if \( \text{Inv} (F) \) is an NP-hard constraint language.

In the following theorem, we show that we only need to consider a very special type of algebras, so called conservative algebras.

**Definition 2.8.** An algebra \( A = (A, F) \) is called conservative if for every operation \( f \in F \) we have that \( f(x_1, \ldots, x_n) \in \{x_1, \ldots, x_n\} \).

**Theorem 2.9.** For any finite constraint language \( \Gamma \) over \( A \) and \( C \subseteq A \), there is a polynomial time Turing reduction from \( \text{MinHom} (\Gamma \cup \{C\}) \) to \( \text{MinHom} (\Gamma) \).

**Proof.** Let the primitive positive formula \( \Phi (x_1, \ldots, x_n) = M \bigwedge_{i=1}^M C (y_i) \bigwedge_{i=1}^N \rho_i (z_{i1}, \ldots, z_{in_i}) \), where \( \rho_i \in \Gamma, y_i, z_{ij} \in \{x_1, \ldots, x_n\} \), and weights \( w_{ia}, 1 \leq i \leq n, a \in A \) be an instance of \( \text{MinHom} (\Gamma \cup \{C\}) \). We assume without loss of generality that \( y_i \neq y_j \), when \( i \neq j \). Let \( W = \sum_{i=1}^n \sum_{a \in A} w_{ia} + 1 \) and define a new formula and weights

\[
\Phi'(x_1, \ldots, x_n) = \bigwedge_{i=1}^N \rho_i (z_{i1}, \ldots, z_{in_i})
\]

\[
w_{ia}' = \begin{cases} w_{ia} + W, & \text{if } a \notin C, \exists j \ x_i = y_j \\ w_{ia}, & \text{otherwise} \end{cases}
\]

Then, using an oracle for \( \text{MinHom} (\Gamma) \), we can solve

\[
\min_{f \text{ satisfies } \Phi'} \sum_{j} w_{j}'f(x_j).
\]

Suppose that \( \Phi (x_1, \ldots, x_n) \) is satisfiable and \( f \) is a satisfying assignment. It is easy to see that the part of the measure \( \sum_j w_{j}'f(x_j) \) that corresponds to the added values \( W \) is equal to 0 and the measure cannot be greater than \( W - 1 \). If \( g \) is any assignment that does not satisfy \( \bigwedge_{i=1}^M C (y_i) \), then we see that this part of measure cannot be 0, and hence, is greater than or equal to \( W \). This means that the minimum in the task is achieved on satisfying assignments of \( \Phi (x_1, \ldots, x_n) \) and any such assignment minimizes the part of the measure that corresponds to the initial weights, i.e., \( \sum_i w_{i}f(x_i) \).

If \( \Phi (x_1, \ldots, x_n) \) is not satisfiable, then either \( \Phi' \) is not satisfiable or \( \min_{f \text{ satisfies } \Phi'} \sum_{j} w_{j}'f(x_j) \geq W \). Using the oracle for \( \text{MinHom} (\Gamma) \), we can easily check this.

Consequently, \( \text{MinHom} (\Gamma \cup \{C\}) \) is polynomial-time reducible to \( \text{MinHom} (\Gamma) \). \( \blacksquare \)
A constraint language is called conservative if it contains all subsets of a domain set. This definition is natural due to the following simple fact.

**Theorem 2.10.** If \( \Gamma \) is a conservative constraint language over \( A \), then \( \mathbb{A} = (A, \text{Pol}(\Gamma)) \) is conservative.

**Proof.** Let \( C = \{x_1, \ldots, x_n\} \subseteq A \). If a function \( f : A^n \to A \) preserves the predicate \( C \), then \( f(x_1, \ldots, x_n) \in \{x_1, \ldots, x_n\} \).

\[ \text{Theorem 3.1.} \quad \text{The Boolean functional clone} \quad H \quad \text{is tractable if either} \quad \{x \land y, x \lor y\} \subseteq H \quad \text{or} \quad \{(x \land y) \lor (y \land z) \lor (x \land z)\} \subseteq H, \quad \text{where} \quad \land, \lor \quad \text{denote conjunction and disjunction. Otherwise,} \quad H \quad \text{is NP-hard.} \]

Let us identify all tractable conservative relational clones in the boolean case using Post’s classification [26]. From Theorem 2.4 we see that any such clone correspond to some conservative functional clone. In the case \( A = \{0, 1\} \), there is a countable number of conservative clones: we list them below according to the table on page 76 [25]. The predicates on the right are given by equalities and equalities separated by comma define different predicates. For every row, the closure of the predicates given is equal to the set of all predicates preserved under the functional clone on the left. At the same time the functional clone is defined as the set of functions preserving the predicates.

| \( T_{01} \) | \( x = 0, x = 1 \) |
| \( M_{01} \) | \( x = 0, x = 1, x_1 \leq x_2 \) |
| \( S_{01} \) | \( x = 0, x_1 \neq x_2 \) |
| \( SM \) | \( x_1 \neq x_2, x_1 \leq x_2 \) |
| \( L_{01} \) | \( x = 0, x_1 \oplus x_2 \oplus x_3 = 0 \) |
| \( U_{01} \) | \( x = 0, x_1 = x_2 \lor x_1 = x_3 \) |
| \( K_{01} \) | \( x = 0, x_1 = x_2 \lor x_3 \) |
| \( D_{01} \) | \( x = 0, x_1 \neq x_2 \lor x_3 \) |
| \( I_{1}^{m} \) | \( x_1 x_2 \ldots x_i = 0, i = \overline{1,m} \) |
| \( M_{1}^{m} \) | \( x = 1, x_1 \leq x_2, x_1 x_2 \ldots x_i = 0, i = \overline{1,m} \) |
| \( Q_{1}^{m} \) | \( x = 0, x_1 \lor x_2 \lor \cdots \lor x_i = 1, i = \overline{1,m} \) |
| \( M_{0}^{m} \) | \( x = 0, x_1 \leq x_2, x_1 \lor x_2 \lor \cdots \lor x_i = 1, i = \overline{1,m} \) |

where \( x \oplus y = x + y \mod 2 \) and \( m \) can be equal to \( \infty \).

**Lemma 3.2.** The relational clones \( \text{Inv}(T_{01}) \), \( \text{Inv}(M_{01}) \) and \( \text{Inv}(S_{01}) \) are tractable. All other conservative Boolean relational clones are NP-hard.
Proof. The class \( \text{Inv}(T_{01}) \) consists of predicates that can be pp-defined over two simple unary predicates \( \{0\} \) and \( \{1\} \). Therefore, by Theorem 2.5, \( \text{Inv}(T_{01}) \) is tractable. As we will see later, this class is included in other tractable constraint languages.

Let us prove that \( \text{Inv}(M_{01}) \) is tractable. By Theorem 2.5, it is equivalent to polynomial solvability of \( \text{MinHom}\left(\{0\},\{1\},\{(x_1,x_2) | x_1 \leq x_2\}\right) \), because the class \( \text{Inv}(M_{01}) \) is the closure of this set of predicates. Let us prove that this constraint language is tractable. Another proof of this statement can be found in [22].

Obviously, \( \text{MinHom}\left(\{0\},\{1\},\{(x_1,x_2) | x_1 \leq x_2\}\right) \) is equivalent to the following boolean linear programming task, sets \( Q_0,Q_1 \subseteq \{1,\ldots,n\} \), \( Q \subseteq \{1,\ldots,n\} \) and integer weights \( w_1,\ldots,w_n \) given as an input:

\[
\min \sum_i w_ix_i \\
\begin{cases} 
  x_i = 0, & i \in Q_0 \\
  x_i = 1, & i \in Q_1 \\
  x_i \leq x_j, & (i,j) \in Q \\
  x_i \in \{0,1\} 
\end{cases}
\]

Let us prove that the polyhedron which is given by the same equalities and inequalities as above, but with \( x_i \in \{0,1\} \) replaced by \( 0 \leq x_i \leq 1 \), is integer. Suppose it is not integer and \( v = ||v_1,v_2,\ldots,v_n||^2 \) is its extreme point where \( v_r \) is not equal to 0 or 1. Let us define \( \epsilon \) as the minimum of three values \( \min |v_i - v_j|, \min |v_i|, \min |1 - v_i| \) and two vectors \( v^+ \) and \( v^- \): \( v^+_i = v^-_i = v_i \) if \( v_i \neq v_r \) and \( v^+_i = v_i + \epsilon, v^-_i = v_i - \epsilon \), otherwise. It is easy to see that points \( v^+ \) and \( v^- \) are also in the polyhedron, and \( v = \frac{v^+ + v^-}{2} \). This contradicts the extremeness of \( v \).

Since the polyhedron is integer we can solve \( \text{MinHom}\left(\{0\},\{1\},\{(x_1,x_2) | x_1 \leq x_2\}\right) \) in polynomial time by standard linear programming algorithms. Consequently, \( \text{Inv}(M_{01}) \) is tractable.

Now let us prove that \( \text{Inv}(S_{01}) \) is tractable, i.e. \( \text{MinHom}\left(\{0\},\{(x_1,x_2) | x_1 \neq x_2\}\right) \) is polynomial-time solvable.

Let an instance of this problem be the sets \( Q_0 \subseteq \{1,\ldots,n\} \), \( Q \subseteq \{1,\ldots,n\} \) and integer weights \( w_{10},\ldots,w_{n0},w_{11},\ldots,w_{n1} \). For \( A \subseteq \{1,\ldots,n\} \), by \( \Phi(A) \) we denote the set of assignments to the variables \( x_v,v \in A \) that satisfy the input formula, i.e. such that \( x_i = 0, i \in Q_0 \cap A \) and \( x_k \neq x_l, (k,l) \in Q \cap A^2 \).

The graph \( (\{1,\ldots,n\},Q') \) where \( Q' = \{(x,y)|(x,y) \in Q \lor (y,x) \in Q\} \) can be decomposed into connected components \( (\{1,\ldots,n\},Q') = K_1 \cup \cdots \cup K_t \), where \( K_i = (V_i,E_i) \). Such a decomposition can be made in \( O(n^2) \) steps. If among these components there is a graph with an odd cycle, then, obviously, \( \Phi(\{1,\ldots,n\}) = \emptyset \). Otherwise, the optimization task can be reduced to subtasks for every component. I.e., if for some component \( \Phi(V_i) = \emptyset \), then \( \Phi(\{1,\ldots,n\}) = \emptyset \), otherwise:

\[
\min_{\pi \in \Phi(\{1,\ldots,n\})} \sum_{i=1}^n w_ix_i = \sum_{i=1}^t \min_{\pi \in \Phi(V_i)} \sum_{j \in V_i} w_{ij}.
\]

But \( |\Phi(V_i)| \leq 2 \), and a straightforward algorithm solves every subtask. So, \( \text{Inv}(S_{01}) \) is tractable.

Denote by \( \rho_\sqcup, \rho_\sqcap \) predicates \( \{(x_1,x_2) | x_1 \lor x_2\}, \{(x_1,x_2) | x_1 \land x_2\} \) respectively. Now show that the classes in the table, except \( \text{Inv}(M_{01}), \text{Inv}(S_{01}) \) and \( \text{Inv}(T_{01}) \), are NP-hard.
Since,
\[
\begin{align*}
    x_1 \lor x_2 &= \exists x_3 [x_1 \neq x_3] \land [x_3 \leq x_2] \\
    x_1 \lor x_2 &= \exists x_3 [x_3 = 1] \land [x_3 = x_1 \lor x_3 = x_2] \\
    \neg x_1 \lor \neg x_2 &= \exists x_3 [x_3 = 0] \land [x_3 = x_1 x_2] \\
    x_1 \lor x_2 &= \exists x_3 [x_3 = 1] \land [x_3 = x_1 \lor x_2] \\
    \neg x_1 \lor \neg x_2 &= \exists x_3 \ldots x_m [x_1 x_2 \ldots x_m = 0] \land [x_2 = x_3] \land \cdots \land [x_{m-1} = x_m] \\
    x_1 \lor x_2 &= \exists x_3 \ldots x_m [x_1 \lor x_2 \lor \cdots \lor x_m = 1] \land [x_2 = x_3] \land \cdots \land [x_{m-1} = x_m]
\end{align*}
\]
we see that \( \rho_\lor \in \text{Inv}(SM), \text{Inv}(U_{01}), \text{Inv}(D_{01}), \text{Inv}(O_{0}^{m}), \text{Inv}(MO_{0}^{m}) \) and \( \rho_\land \in \text{Inv}(K_{01}), \text{Inv}(I_{1}^{m}), \text{Inv}(MI_{1}^{m}) \).

We first prove that \( \text{MinHom}(\{\rho_\lor\}) \) is NP-hard. Suppose an instance of this problem consists of an undirected graph \( G = (V, E) \) where each vertex is considered as a variable. For each pair of variables \( (u, v) \in E \), we require their assignments to satisfy \( (f(u), f(v)) \in \rho_\lor \). It is easy to see that for any such assignment \( f \), the set \( \{x | f(x) = 0\} \) is independent in the graph \( G \). Furthermore, for any independent set \( S \) in the graph \( G \), \( g(x) = [x \notin S] \) is a satisfying assignment. If we define \( w_{i0} = 0, w_{i1} = 1 \) for \( i \in V \), then \( \text{MinHom} \) is equivalent to finding a maximum independent set. This implies that \( \text{MinHom}(\{\rho_\lor\}) \) is NP-hard, since finding independent sets of maximal size is an NP-hard problem. The case \( \text{MinHom}(\{\rho_\land\}) \) is analogous.

Therefore, \( \text{Inv}(SM), \text{Inv}(U_{01}), \text{Inv}(D_{01}), \text{Inv}(O_{0}^{m}), \text{Inv}(MO_{0}^{m}), \text{Inv}(K_{01}), \text{Inv}(I_{1}^{m}), \text{Inv}(MI_{1}^{m}) \) are NP-hard, too.

It remains to prove NP-hardness of \( \text{Inv}(L_{01}) \). We show that using an algorithm for \( \text{MinHom}(\{(x_1, x_2, x_3) | x_1 \oplus x_2 \oplus x_3 = 0\}) \) as an oracle, we can solve Max-CUT in polynomial time.

Let \( G = (V, E) \) be a graph and introduce variables \( x_{ij}, y_i, y_j, i, j \in V \). A system of equations \( x_{ij} \oplus y_i \oplus y_j = 0, i, j \in V \) can be viewed as an instance of \( \text{MinHom}(\{(x_1, x_2, x_3) | x_1 \oplus x_2 \oplus x_3 = 0\}) \). It is easy to see that an arbitrary boolean vector \( \vec{y} = (y_1, \ldots, y_{|V|}) \) defines a single solution \( x_{ij} = y_i \oplus y_j, i, j \in V \) of the system. Vector \( \vec{y} \) can be considered as the cut \( \{i | y_i = 1\} \subseteq V \) and the value \( \sum_{(i,j) \in E} x_{ij} \) is equal to the doubled cost of the cut. If we define all weights to be equal to 0, except \( \forall (i, j) \in E \ w_{x_{ij}} = 1 \), then minimizing a resulting functional will be equivalent to solving Max-CUT. It is easy to see that this reduction of Max-CUT to \( \text{MinHom}(\text{Inv}(L_{01})) \) can be done in a number of steps that is polynomial of a size of an input graph. Lemma proved.

\[
\textbf{Lemma 3.3.} \quad \text{A Boolean conservative relational clone } S \text{ is NP-hard if it contains any of predicates } \rho_\lor, \rho_\land, \{(x_1, x_2, x_3) | x_1 \oplus x_2 \oplus x_3 = 0\}. \text{ Otherwise, it is tractable.}
\]

\textbf{Proof.} By Theorem 2.4, \( S \text{ is equal to } \text{Inv}(\text{Pol}(S)) \) where \( \text{Pol}(S) \) is a conservative functional clone. It is easy to see from the proof of Lemma 3.2 that all NP-hard conservative functional clones contain at least one of the following polymorphisms \( \rho_\lor, \rho_\land, \{(x_1, x_2, x_3) | x_1 \oplus x_2 \oplus x_3 = 0\} \). At the same time \( \text{Inv}(T_{01}), \text{Inv}(M_{01}) \) and \( \text{Inv}(S_{01}) \) do not contain any of them and that proves the statement.

\[
\textbf{Lemma 3.4.} \quad \text{If a Boolean constraint language } S \text{ is contained neither in } \text{Inv}(M_{01}) \text{ nor in } \text{Inv}(S_{01}) \text{, then it is NP-hard.}
\]
Proof. Suppose a constraint language \( S \), and therefore \( S \cup \{ \{ 0 \}, \{ 1 \} \} \), is not contained in \( \text{Inv}(M_{01}) \) and \( \text{Inv}(S_{01}) \). Then, \( \langle S \cup \{ \{ 0 \}, \{ 1 \} \} \rangle \) is not contained in \( \text{Inv}(M_{01}) \) and \( \text{Inv}(S_{01}) \), either. Since \( \langle S \cup \{ \{ 0 \}, \{ 1 \} \} \rangle \) is a boolean conservative relational clone which is not equal to \( \text{Inv}(T_{01}) \), \( \text{Inv}(M_{01}) \) and \( \text{Inv}(S_{01}) \), then, by Lemma 3.2, it is NP-hard. By Theorems 2.5 and 2.9, we conclude that \( S \) is NP-hard.

\[ \text{Proof of Theorem 3.1.} \text{ The bases in the clones } M_{01}, S_{01} \text{ are } \langle \land, \lor \rangle \text{ and } \{(x \land \overline{y}) \lor (\overline{y} \land z) \lor (x \land z)\} \text{ [25] and the theorem follows from Lemmas 3.2 and 3.4.} \]

Every 2-element subalgebra of a tractable algebra must be tractable, which motivates the following definition.

**Definition 3.5.** Let \( F \) be a conservative functional clone. We say that \( F \) satisfies the necessary local conditions if and only if for every 2-element subset \( B \subseteq A \), either

- there exists \( f^\land, f^\lor \in F \) s.t. \( f^\land|_B \) and \( f^\lor|_B \) are different binary commutative functions; or
- there exists \( f \in F \) s.t. \( f|_B (x, y) = f|_B (y, x, x) = f|_B (y, x, y) = y \) for any \( x, y \in B \).

**Theorem 3.6.** Suppose \( F \) is a conservative functional clone. If \( F \) is tractable, then it satisfies the necessary local conditions. If \( F \) does not satisfy the necessary local conditions, then it is NP-hard.

**Proof.** Suppose \( F \) is tractable. For every two-element subset \( B \subseteq A \): \( \text{Inv}(F|_B) \subseteq \text{Inv} (F) \). Therefore, \( F|_B \) is tractable. Assume without loss of generality that \( B = \{0, 1\} \). From Theorem 3.1, we get that \( \langle \land, \lor \rangle \subseteq F|_B \) or \( \{a(x, y, z) = (x \land \overline{y}) \lor (\overline{y} \land z) \lor (x \land z)\} \subseteq F|_B \). \( \land, \lor \) is a pair of different commutative conservative functions and \( a(x, y, z) = a(y, x, x) = a(y, x, y) = y \):

\[ a(x, y, z) = (x \land \overline{y}) \lor (\overline{y} \land z) \lor (x \land z) = 0 \lor ((\overline{y} \lor x) \land y) = y \]

By symmetry, \( a(y, x, x) = y \). The third equality holds also:

\[ a(y, x, y) = (y \land \overline{y}) \lor (\overline{y} \land z) \lor (y \land z) = (y \land \overline{y}) \lor y = y \]

If \( F \) does not satisfy the necessary local conditions, then, by Theorem 3.1, \( F|_B \) is NP-hard, and therefore, \( F \) is NP-hard.

In general, the necessary local conditions are not sufficient for tractability of a conservative clone. Let \( M = \{B|B \subseteq A, |B| = 2, F|_B \text{ contains different binary commutative functions}\} \) and \( \overline{M} = \{B|B \subseteq A, |B| = 2 \} \setminus M \).

Suppose \( f \in F \). By \( a \Downarrow \) \( b \) we mean \( a \neq b \) and \( f(a, b) = f(b, a) = b \). For example, \( a \Downarrow \overline{a} \Downarrow \overline{b} \Downarrow \overline{c} \Downarrow \overline{d} \) means that \( f|_{\{1, 2, 3\}} (x, y) = \max (x, y) \).

Introduce an undirected graph without loops \( T_F = (M^o, P) \) where \( M^o = \{(a, b) \mid (a, b) \in M\} \) and \( P = \{(a, b), (c, d) \mid (a, b), (c, d) \in M^o, \text{ there is no } f \in F : a \Downarrow \overline{a} \Downarrow \overline{b} \Downarrow \overline{c} \Downarrow \overline{d} \} \).

The core result of the paper is the following.

**Theorem 3.7.** Suppose \( F \) satisfies the necessary local conditions. If the graph \( T_F = (M^o, P) \) is bipartite, then \( F \) is tractable. Otherwise, \( F \) is NP-hard.
The proof of this theorem will be given in two steps. Firstly, in the following section, we will prove NP-hardness of $F$ when $T_F = (M^0, P)$ is not bipartite. The final sections will be dedicated to the polynomial-time solvable cases.

4. NP-hard case

In this section, we will prove that if a set of functions $F$ satisfies the necessary local conditions and $T_F = (M^0, P)$ (as defined in the previous section) is not bipartite, then $F$ is NP-hard. Let $a \leftrightarrow_d b$ and $a \leftrightarrow_d b$ denote the predicates $\{(a, b) \times \{c, d\} \setminus \{(b, d)\}$ and $\{(a, d), (b, c)\}$, where $a \neq b, c \neq d$. We need the following lemmas.

**Lemma 4.1.** A constraint language that contains $a \leftrightarrow_d b$ and $b \leftrightarrow_d c$ is NP-hard.

Before proving Lemma 4.1, we need to introduce some concepts from graph theory. All graphs are assumed to be undirected and without loops. We will be interested in the complexity of finding independent sets of maximal size in classes of graphs. Let a finite number of graphs $G_1, \ldots, G_k$ be given and let $\text{Free}(G_1, \ldots, G_k)$ denote the set of graphs that has no induced subgraphs isomorphic to one of $G_1, \ldots, G_k$.

The following theorem has been proved by Alekseev[1].

**Theorem 4.2.** If there is no graph among $G_1, \ldots, G_k$ whose every connected component is a tree with at most 3 leaves, then the maximum independent set problem is NP-hard when restricted to graphs in $\text{Free}(G_1, \ldots, G_k)$.

**Definition 4.3.** The graph $G = (V, E)$ is said to be **homomorphic** to the graph $H = (W, S)$ if there is a mapping $f : V \rightarrow W$ such that $\forall (x, y) \in E \ (f(x), f(y)) \in S$. The mapping $f$ is called an $H$-homomorphism.

Let $C_d$ be a cycle of length $d$.

**Theorem 4.4.** If $d \geq 3$ is odd, then the problem of finding a maximum independent set in an undirected graph homomorphic to $C_d$ is NP-hard even if a $C_d$-homomorphism is given.

**Proof.** Following [16] we first prove the NP-hardness of finding maximum independent sets in a graph homomorphic to $C_3$, i.e. a three-partite graph. An instance consists of a graph and a partitioning into three independent sets.

Let $X$ be a class of graphs with degrees not greater than 3. This class can be characterized by forbidden induced subgraphs — it is sufficient to forbid all graphs with 5 vertices that have a vertex connected with 4 others. Obviously, every such graph is connected and if it is a tree it has 4 leaves. By Theorem 4.2 we conclude that finding maximum independent sets is NP-hard in the class $X$.

From Brooks’s theorem [4], we have that every graph in $X$, apart from the complete graph on 4 vertices, is three-partite. The required partition can be constructed in polynomial time by an algorithm of Lovasz [24]. Therefore, the problem of finding maximum independent sets in a three-partite graph is NP-hard even if a partition is given.

The case when $d = 3$ can be reduced to every odd case $d > 3$. Let a three-partite graph be given. We will represent it in the following form: $G = (V_1, V_2, V_3, E_{12}, E_{23}, E_{31})$, where $E_{12} \subseteq V_1 \times V_2, E_{23} \subseteq V_2 \times V_3, E_{31} \subseteq V_3 \times V_1$. Transform $G$ as follows: for each edge $(u, v) \in E_{12}$, add vertices $x_{uv1}, x_{uv2}, \ldots, x_{uv(d-3)}$ to the graph, delete the edge $(u, v)$,
and add edges \((u, x_{uv1}), (x_{uv1}, x_{uv2})\ldots, (x_{uv(d-3)}, v)\). The obtained graph \(G^d\) is, obviously, homomorphic to \(C_d\).

Let \(n, N\) denote the independence numbers (i.e. sizes of maximum independent sets) of \(G\) and \(G^d\) respectively. It is easy to see that \(N \geq n + \frac{d-3}{2}|E_{12}|\). We prove that we actually have an equality. Note that intersection of any maximum independent set of \(G^d\) and \(\{u, x_{uv1}, x_{uv2}\ldots, x_{uv(d-3)}, v\}\) contains no less than \(\frac{d-3}{2}\), and no more than \(\frac{d-1}{2}\) elements. In the first case \((\frac{d-3}{2})\), we can delete all elements \(u, x_{uv1}, x_{uv2}\ldots, x_{uv(d-3)}, v\) from the independent set and replace them by \(x_{uv1}, x_{uv3}, x_{uv5},\ldots, x_{uv(d-4)}\), while not destroying independency. In the second case \((\frac{d-1}{2})\), either \(u\) or \(v\) are in the independent set. Again, we delete \(u, x_{uv1}, x_{uv2}\ldots, x_{uv(d-3)}, v\) from it. If \(u\) is originally in the independent set, we replace the deleted elements by \(\{u, x_{uv2}, x_{uv4},\ldots, x_{uv(d-3)}\}\) and otherwise replace them by \(\{x_{uv3}, x_{uv4}\ldots, x_{uv(d-4)}, v\}\). As a result, we obtain an independent set of \(G^d\) with the same cardinality as initially. This operation can be done with all pairs \(uv \in E_{12}\). It is easy to see that the intersection of the obtained set with \(V_1 \cup V_2 \cup V_3\) is an independent set in \(G\) and it has cardinality \(N - \frac{d-3}{2}|E_{12}|\). Consequently, \(N = n + \frac{d-3}{2}|E_{12}|\) and the intersection is a maximum independent set in \(G\). The graph \(G^d\) can be constructed in polynomial time. Thus, by finding a maximum independent set in \(G^d\), we can easily reconstruct that of \(G\). This means that the maximum independent set problem in a three-partite graph is polynomial-time reducible to the maximum independent set problem in a graph homomorphic to \(C_d\) (with given homomorphism).

**Proof of Lemma 4.1.** We show that finding a maximum independent set in a graph homomorphic to \(C_{2k+1}\) can be reduced to \(\text{MinHom}(\Gamma)\) where \(\Gamma = \{a_0 \overline{a_1}, b_0 \overline{b_1}, a_1 \overline{a_2}, b_1 \overline{b_2}, \ldots, a_{2k-1} \overline{a_{2k}}, b_{2k-1} \overline{b_{2k}}, a_{2k} \overline{a_0}, b_{2k} \overline{b_0}\}\).

Suppose the task is to find a maximum independent set in a graph homomorphic to \(C_{2k+1}\), which, for convenience, will be given in the following form: \(G = (V_i, E_i, i \oplus 1 \subseteq V_i \times V_{i\oplus1}\) where \(i \oplus j\) denotes \(i + j\) (mod \(2k + 1\)). We consider every vertex \(v \in \bigcup_{i=0}^{2k} V_i\) as a variable and require values of variables \((u, v) \in V_i \times V_{i\oplus1}\) to satisfy the constraint \(a_i \overline{a_{i+1}} b_{i\oplus1} \overline{b_i}\). The set of satisfying assignments is denoted by \(\Phi\). It is easy to see that

\[
\Phi = \left\{ f \big| \forall v \in V_i, f(v) \in \{a_i, b_i\}, \bigcup_i \{x \in V_i, f(x) = b_i\} \text{ is an independent set in } G \right\}.
\]

\([x]\) denotes a function of a statement \(x\) such that \([x] = 1\) if \(x = \text{true}\), and 0 otherwise. The task

\[
\min_{f \in \Phi} \sum_i \sum_{x \in V_i} [f(x) \neq b_i]
\]

is equivalent to finding a maximum independent set in the graph \(G\). I.e., it is equivalent to the \(\text{MinHom}(\Gamma)\) problem with an instance consisting of the defined constraints on the variables \(\bigcup_{i=0}^{2k} V_i\) and weights \(w_{xa_i} = 1, w_{xb_i} = 0\). Consequently, by Theorem 4.4, \(\text{MinHom}(\Gamma)\) is NP-hard.
Lemma 4.5. If \( \langle (a, b), (c, d) \rangle \in P \), then either \( \overrightarrow{a \times c} \in \text{Inv} (F) \), or \( \overrightarrow{b \times d} \in \text{Inv} (F) \).

Proof. We begin by constructing functions \( \phi_1, \phi_2 \in F \) such that \( \overrightarrow{a \times c} \phi_1, \overrightarrow{b \times d} \phi_2 \). A symbol \( \overrightarrow{\lambda} \) means that either \( \overrightarrow{\alpha} \lambda \) or \( \lambda|_{(\alpha, \beta)} \) is a projection, i.e. \( \lambda|_{(\alpha, \beta)} (x_1, x_2) = x_i, \ i = 1, 2 \).

Since \( \{a, b\}, \{c, d\} \in M \), we have \( \lambda_1, \lambda_2, \lambda_3, \lambda_4 \in F : \overrightarrow{a \times c} \lambda_1, \overrightarrow{b \times d} \lambda_2 \). Moreover, by the definition of \( P \), we have \( \overrightarrow{c \times d} \lambda_3, \overrightarrow{d \times c} \lambda_4 \). By defining \( \phi_1 (x, y) = \lambda_1(x, y), \lambda_2(x, y), \lambda_4(x, y) \), we see that \( \overrightarrow{a \times d} \phi_1, \overrightarrow{b \times c} \phi_2 \).

Suppose \( \overrightarrow{a \times c} \notin \text{Inv} (F) \). We prove that in this case \( \overrightarrow{b \times d} \in \text{Inv} (F) \). Suppose \( \overrightarrow{a \times c} \) is not preserved by an \( n \)-ary function \( g \in F \). The predicate \( \overrightarrow{a \times c} \) consists of three pairs, which means that by identifying some variables of \( g \) we can obtain a function of arity two or three that does not preserve \( \overrightarrow{b \times d} \) either. Let us consider these two cases:

I. A function \( \phi \in F \) of arity two does not preserve \( \overrightarrow{a \times c} \) if (for some appropriate permutation of variables):

\[
\begin{align*}
\phi (a, b) &= b \\
\phi (d, c) &= d.
\end{align*}
\]

Then \( \overrightarrow{a \times d} \phi (x, y), \overrightarrow{b \times c} \phi (x, y) \) which contradicts that \( \langle (a, b), (c, d) \rangle \in P \).

II. A function \( \phi \in F \) of arity three does not preserve \( \overrightarrow{a \times c} \) if (for some appropriate permutation of variables):

\[
\begin{align*}
\phi (a, a, b) &= b \\
\phi (d, c, c) &= d.
\end{align*}
\]

Then, \( \langle (b, a), (d, c) \rangle \in P \), since, otherwise, we can find \( \phi_3 \in F : \overrightarrow{a \times c} \phi_3 \) and construct the following term \( \overrightarrow{b \times d} \phi_3 \). This contradicts that \( \langle (a, b), (c, d) \rangle \in P \).

Suppose instead that \( \overrightarrow{b \times c} \notin \text{Inv} (F) \), i.e., there is a function \( f \in F \) of arity two that does not preserve \( \overrightarrow{a \times c} \). If \( f \) does not preserve \( \overrightarrow{b \times c} \), then it does not preserve either \( \overrightarrow{a \times d} \), or \( \overrightarrow{b \times d} \). Since \( \langle (a, b), (c, d) \rangle \in P \), we get a contradiction in both cases via the same argument as in case I.

Proof of NP-hard case of Theorem 3.7. For binary predicates \( \alpha, \beta \), let \( \alpha \circ \beta = \{(x, y) | \exists z : \alpha(x, z) \wedge \beta(z, y)\} \). Obviously, if \( \alpha, \beta \in \text{Inv} (F) \), then \( \alpha \circ \beta \in \text{Inv} (F) \), too.

Since \( T_F = (M^o, P) \) is not bipartite, we can find a shortest odd cycle in it, i.e. a sequence \( (a_0, b_0), (a_1, b_1), \ldots, (a_k, b_k) \in M^o, k \geq 1 \), such that \( \langle (a_i, b_i), (a_{i+1}, b_{i+1}) \rangle \in P \). Here, \( i \oplus j \) denotes \( i + j (\text{mod} \ 2k + 1) \).

By Lemma 4.5, there is a cyclic sequence \( \rho_0, \rho_1, \rho_2, \ldots, \rho_{2k, 0} \in \text{Inv} (F) \) such that \( \rho_{i, i+1} \) is either equal to \( \overrightarrow{a \times a} \), or equal to \( \overrightarrow{b \times b} \). Note that all predicates cannot be of the second type: otherwise, we have \( \rho_{0, 1} \circ \rho_{1, 2} \circ \cdots \circ \rho_{2k, 0} = \overrightarrow{a \times a} \) which contradicts that \( \{a_0, b_0\} \in M \).
If the sequence contains a fragment \( \rho_{i,1} = a_i \oplus b_i, \rho_{i,2} = a_i \oplus b_i \), then these predicates can be replaced by:

\[
\rho_{i,1} = \rho_{i,1} \circ \rho_{i,1,2} \circ \rho_{i,2,3} = a_i \oplus b_i \oplus b_i,
\]

Let us replace \( \rho_{i,1}, \rho_{i,1,2}, \rho_{i,2,3} \) by \( \rho_{i,3} \) in the sequence \( \rho_{0,1}, \rho_{1,2}, \ldots, \rho_{2k,0} \). We have \((a_i, b_i), (a_i, b_i) \) \( \in P \), since otherwise the predicate \( \rho_{i,3} \) is not preserved. Hence, we can delete two vertices in the cycle \((a_0, b_0), (a_1, b_1), \ldots, (a_{2k}, b_{2k}) \) in \( M^o \). This contradicts that this sequence is the shortest among odd sequences. Therefore, such a fragment does not exist.

If the sequence contains a fragment \( \rho_{i,1} = a_i \oplus b_i, \rho_{i,2} = a_i \oplus b_i \), then these predicates can be replaced by:

\[
\rho_{i,1} = \rho_{i,1} \circ \rho_{i,1,2} \circ \rho_{i,2,3} = a_i \oplus b_i \oplus b_i,
\]

As in the previous case, we obtain a contradiction. Consequently, we have an odd sequence \( (a_0, b_0), (a_1, b_1) \) \( \in Inv (F) \). By Lemma 4.1, this class of predicates is NP-hard.

5. Existence of the majority operation

In the previous section we proved that an algebra \( F \) that satisfies the necessary local conditions is NP-hard if the graph \( T_F = (M^o, P) \) is not bipartite. In this section we will clarify the structure of an algebra if \( T_F = (M^o, P) \) is bipartite.

The necessary local conditions tell us that every two-element subalgebra of a tractable algebra contains certain operations. Let us consider two simplest algebras over a domain \( A \) that satisfy these conditions. The first case is \( F_1 = \{ \phi, \psi \} \) where \( \phi, \psi \) are conservative commutative operations such that \( \phi(a, b) \neq \phi(a, b) \) for every \( a \neq b \in A \). The algebra \( F_1 \) first appeared in the research concerning tractable subproblems of VCSP and in this context \( \phi, \psi \) are called a symmetric tournament pair [9]. And the second algebra is \( F_2 = \{ m \} \) where \( m \) is a conservative operation such that \( \forall x, y \in A \ m (x, y) = m (y, x) = m (y, x, y) = y. \) Functions that satisfy this equality are often called arithmetical or 2/3-minority operations [6].

This leads us to the following definitions.

**Definition 5.1.** Suppose a set of operations \( H \) over \( A \) is conservative and \( B \subseteq \{ \{ x, y \} | x, y \in A, x \neq y \} \). A pair of binary operations \( \phi, \psi \in H \) is called a tournament pair on \( B \), if \( \forall \{ x, y \} \in B \ \phi (x, y) = \phi (y, x), \psi (x, y) = \psi (y, x), \phi (x, y) \neq \psi (x, y) \) and for arbitrary \( \{ x, y \} \in B \), \( \phi (x, y) = x, \psi (x, y) = y. \) An operation \( m \in H \) is called arithmetical on \( B \), if \( \forall \{ x, y \} \in B \ m (x, y) = m (y, x) = m (y, x) = y. \)

**Definition 5.2.** An operation \( \mu : A^3 \rightarrow A \), satisfying the equality

\[
\mu (x, y, y) = \mu (y, x, y) = \mu (y, y, x) = y
\]

is called a majority operation.

**Theorem 5.3.** If \( F \) satisfies the necessary local conditions and \( T_F = (M^o, P) \) is bipartite, then \( F \) contains a tournament pair on \( M \).
Proof. Note first that for every \( B \in \overline{M} \), \( F|_B \) cannot contain any commutative binary function. To see this, assume that \( B = \{0, 1\} \) and suppose that \( F|_B \) contains \( S_{01} \) and either conjunction or disjunction. From Post’s results [26], we see that \( F|_B \) contains all boolean functions preserving 0 and 1, i.e., contains both conjunction and disjunction. This contradicts that \( B \notin M \). Therefore, every binary function in \( F|_B \) is a projection.

Let \( M_1, M_2 \) denote a bipartition of the bipartite graph \( T_F = (M^a, P) \). Then, for every \((a, b), (c, d) \in M_1 \), there is a function \( \phi \in F : \begin{array}{c} \phi \end{array} \). Let us prove by induction that for every \((a, b), (c, d) \in M_1 \), there is a \( \phi : \begin{array}{c} a_1 \cdots a_n \end{array} \). The base of induction \( n = 2 \) is obvious. Let \((a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n) \in M_1 \) be given. By the induction hypothesis, there are \( \phi_1, \phi_2, \phi_3 \in F : \begin{array}{c} \phi_1 \cdots \phi_3 \end{array} \). Then, it is easy to see that \begin{array}{c} \phi_1 \cdots \phi_3 \end{array} \) which completes the induction.

The analogous statement can be proved for \( M_2 \). Moreover, \( M_2 = \{((x, y), (y, x)) \in M_1 \} \). So it follows from the proof that there are binary operations \( \phi', \psi' \in F \), such that \( \forall (x, y) \in M_1 : \begin{array}{c} \phi' \end{array} \) and \( \forall (x, y) \in M_2 : \begin{array}{c} \psi' \end{array} \). Thus, the operations \( \phi(x, y) = \phi'(x, \phi'(y, x)) \) and \( \psi(x, y) = \psi'(x, \psi'(y, x)) \) satisfy the conditions of the theorem.

The proof of the following theorem uses ideas from [5].

**Theorem 5.4.** If \( F \) satisfies the necessary local conditions and \( \overline{M} \neq \emptyset \), then \( F \) contains an arithmetical operation on \( M \).

**Proof.** For \( B \in \overline{M} \), let \( m^B \) be an arithmetical function on \( B \); existence of this function follows from the necessary local conditions. Assume now that \( \overline{M} = \{(x_1, y_1), \ldots, (x_s, y_s)\} \). We prove by induction that for every \( r \leq s \), \( F \) contains a function \( m_r : A^3 \rightarrow A \) that is arithmetical on \( \{(x_1, y_1) \} \). When \( r = 1 \), \( m_1(x, y, z) = m^{(x_1, y_1)}(x, y, z) \) and the statement is obviously true. Suppose it is true for \( r \leq k < s \) and that we have the function \( m_k : A^3 \rightarrow A \). Let us prove the statement for \( r = k + 1 \). If \( m_k \) is arithmetical on \( \{(x_{k+1}, y_{k+1})\} \), then we set \( m_{k+1} = m_k \) and the statement is proved. Otherwise, one of the following three statements is true

\[
\exists x, y \in \{x_{k+1}, y_{k+1} \} [m_k(x, y, z) \neq y],
\exists x, y \in \{x_{k+1}, y_{k+1} \} [m_k(y, x, y) \neq y],
\exists x, y \in \{x_{k+1}, y_{k+1} \} [m_k(y, y, x) \neq y].
\]

Suppose the first case holds (the proof for other cases is analogous). As we see from the note in the proof of Theorem 5.3, \( m_k^{(x_{k+1}, y_{k+1})}(x, y, z) \) is a binary function and hence a projection. It is necessarily the x-projection since it is not the y-projection. It is easy to see that the function \( m_{k+1}(x, y, z) = m_k^{(x_{k+1}, y_{k+1})}(x, y, z) \) is arithmetical on \( \{(x_{k+1}, y_{k+1}) \} \). The induction is complete and it is clear that \( m_s(x, y, z) \) satisfies the condition of the theorem. □
Theorem 6.3. Let $U_n$ be a finite set of variables and $E$ be a finite set of constraints.

Definition 6.2. The microstructure graph of an instance with constraints pair $Un = \langle \rho_i \rangle_{1 \leq i \leq n}$, $Bin = \langle \rho_{kl} \rangle_{1 \leq k \neq l \leq n}$ is the graph $M_{Un, Bin} = (V, E)$, where $V = \{i, a\} | 1 \leq i \leq n, a \in \rho_i\}$ and $E = \{(i, a), (j, b) | i \neq j, (a, b) \in \rho_{ij}\}$.

Theorem 5.5. If $F$ satisfies the necessary local conditions and $T_F = (M^o, P)$ is bipartite, then $F$ contains a majority operation $\mu$.

Proof. If $M \neq \emptyset$, then by Theorem 5.4, $F$ contains a function $m : A^2 \rightarrow A$ that is arithmetical on $M$. Then the function $\mu^1(x, y, z) = m(x, m(x, y, z), z)$ satisfies the conditions $\forall \{x, y\} \in M \mu^1(x, y, y) = \mu^1(y, x, y) = \mu^1(y, y, x) = y$. It is clear that, in the case where $M = \emptyset$, we can take $\mu^1$ as majority $\mu$.

If $M \neq \emptyset$, then by Theorem 5.3, there is a tournament pair $\phi, \psi : A^2 \rightarrow A$ on $M$. Then, the function $\mu^2(x, y, z) = \phi(\psi(x, y), \psi(y, z)), \psi(x, z)$ satisfies conditions $\forall \{x, y\} \in M \mu^2(x, y, y) = \mu^2(y, x, y) = \mu^2(y, y, x) = y$, and $\forall \{x, y, z\} \in M \mu^2(x, y, z) = x$. If $M \neq \emptyset$, then we can take $\mu^2$ as the majority $\mu$.

Finally, if $M, M \neq \emptyset$, then $\mu(x, y, z) = \mu^1(\mu^2(x, y, z), \mu^2(y, z, x), \mu^2(z, x, y))$. 

6. Consistency and microstructure graphs

Every predicate in $Inv(F)$, when $F$ contains a majority operation, is equal to the join of its binary projections [2]. To prove Theorem 3.7, it is consequently sufficient to prove polynomial-time solvability of $MinHom(\Gamma)$ where $\Gamma = \{\rho|\rho \subseteq A^2, \rho \in Inv(F)\}$, i.e. the $MinHom$ problem restricted to binary constraint languages.

Definition 6.1. Suppose we are given a constraint language $\Gamma$ over $A$. Denote by $2 - MinHom(\Gamma)$ the following minimization problem:

Instance: A finite set of variables $X = \{x_1, \ldots, x_n\}$, a constraints pair $(Un, Bin)$ where $Un = \langle \rho_i \rangle_{1 \leq i \leq n}$, $Bin = \langle \rho_{kl} \rangle_{1 \leq k \neq l \leq n}$, $\rho_i \subseteq A, \rho_{kl} \subseteq A^2, \rho_i, \rho_{kl} \in \Gamma$, and weights $w_{ia}, 1 \leq i \leq n, a \in A$.

Solution: Assignment $f : \{x_1, \ldots, x_n\} \rightarrow A$, such that $\forall i f(x_i) \in \rho_i$ and $\forall k \neq l (f(x_k), f(x_l)) \in \rho_{kl}$.

Measure: $\sum_{i=1}^{n} w_{if(x_i)}$.

We suppose everywhere that $\rho_{kl} = \rho_{lk}^l$ (where $\rho^l = \{(a, b) | (a, b) \in \rho\}$). If $\rho_{kl} \neq \rho_{lk}^l$, then we can always define $\forall k \neq l \rho_{kl}^l := \rho_{kl} \cap \rho_{lk}^l$, which does not change the set $\{(a, b) | (a, b) \in \rho_{kl}, (a, b) \in \rho_{lk}\}$. For a binary predicate $\rho$, define projections $Pr_1 \rho = \{a|(a, b) \in \rho\}$ and $Pr_2 \rho = \{b|(a, b) \in \rho\}$.

Definition 6.2. The microstructure graph [21] of an instance of $2 - MinHom(\Gamma)$ with constraints pair $Un = \langle \rho_i \rangle_{1 \leq i \leq n}$, $Bin = \langle \rho_{kl} \rangle_{1 \leq k \neq l \leq n}$ is the graph $M_{Un, Bin} = (V, E)$, where $V = \{i, a\} | 1 \leq i \leq n, a \in \rho_i\}$ and $E = \{(i, a), (j, b) | a \neq b, (a, b) \in \rho_{ij}\}$.

Theorem 6.3. Let $I = (X, Un, Bin, w)$ be a satisfiable instance of $2 - MinHom(\Gamma)$. Then there is a one-to-one correspondence between cliques of maximal size in $M_{Un, Bin}$ and satisfying assignments of $I$.

Proof. The microstructure graph of an instance with constraints pair $Un = \langle \rho_i \rangle_{1 \leq i \leq n}$, $Bin = \langle \rho_{kl} \rangle_{1 \leq k \neq l \leq n}$ is, obviously, $n$-partite, since $V = \bigcup_{i=1}^{n} \{i\} \times \rho_i$ and pairs $(i, a), (j, b), a \neq b$ are not connected. Therefore, the cardinality of a maximal clique of $M_{Un, Bin} = (V, E)$ is not greater than $n$.

If the cardinality of a maximal clique $S \subseteq V$ is $n$, then, for every $i$, $|S \cap \{i\} \times \rho_i| = 1$. Then, denoting the only element of $S \cap \{i\} \times \rho_i$ by $v_i$, we see that the assignment
\( f(x_i) = v_i \) satisfies all constraints. The opposite is also true, i.e., if the constraints \( \langle \rho_i \rangle_{1 \leq i \leq n}, \langle \rho_k \rangle_{1 \leq k \neq l \leq n} \) can be satisfied by some assignment \( f \), then \( \{(i, f(x_i)) | 1 \leq i \leq n \} \) is a clique of cardinality \( n \).

Hence, \( 2 - \text{MinHom}(\Gamma) \) can be reduced to finding a maximal-size clique \( S \subseteq V \) of a microstructure graph that minimizes the following value:

\[
\sum_{(i,a) \in S} w_{ia}.
\]

**Definition 6.4.** Let \( \text{MMClique} \) (Minimal weight among maximal-size cliques) denote the following minimization problem:

**Instance:** A graph \( G = (V, E) \) and weights \( w_i \in \mathbb{N}, i \in V \).

**Solution:** A maximal-size clique \( K \subseteq V \) of \( G \).

**Measure:** \( \sum_{v \in K} w_v. \)

**Definition 6.5.** An instance of \( 2 - \text{MinHom}(\Gamma) \) with constraints pair \( \text{Un} = \langle \rho_i \rangle_{1 \leq i \leq n}, \text{Bin} = \langle \rho_k \rangle_{1 \leq k \neq l \leq n} \) is called arc-consistent if \( \forall i \neq j : P_1 \rho_{ij} = \rho_i, P_2 \rho_{ij} = \rho_j \) and is called path-consistent if for each different \( i, j, k : \rho_{ik} \subseteq \rho_{ij} \cap \rho_{jk} \).

Obviously, by applying operations \( \rho_i := \rho_i \cap P_1 \rho_{ij}, \rho_j := \rho_j \cap P_2 \rho_{ij}, \rho_{ij} := \rho_{ij} \cap (\rho_i \times A), \rho_{ij} := \rho_{ij} \cap (A \times \rho_j), \rho_{ik} := \rho_{ik} \cap (\rho_{ij} \circ \rho_{jk}) \), we can always make an instance arc-consistent and path-consistent in polynomial time. Standard algorithm \( AC - 4 \) that do the same but with complexity \( O(n^2) \) is described in [10]. It is clear that under these transformations the set of feasible solutions does not change.

The following theorem connects perfect microstructure graphs and the complexity of \( \text{MinHom} \).

**Theorem 6.6.** Suppose we are given a class of conservative functions \( F \) containing a majority operation. If the microstructure graph is perfect for an arbitrary arc-consistent and path-consistent instance of \( 2 - \text{MinHom}(\text{Inv}(F)) \), then \( F \) is tractable.

**Proof.** Recall that a graph \( G = (V, E) \) is called perfect if for every induced subgraph the chromatic number is equal to the clique number.

For a graph \( G = (V, E) \), the following polytope is called the fractional stable set polytope:

\[
\left\{ \begin{array}{l}
\sum_{v \in K} x_v \leq 1, \text{ where } K \text{ is a clique in } G \\
\quad x_v \geq 0, v \in V
\end{array} \right.
\]

By a well-known theorem of Lovasz [13], a graph \( G = (V, E) \) is perfect if and only if its fractional stable set polytope equals the convex hull of the characteristic vectors of independent sets in \( G \). By the vertex packing problem we mean the weighted version of maximum independent set. It is easy to see that vertex packing in perfect graphs is equivalent to optimizing a linear function over the fractional stable set polytope. There is a polynomial algorithm for solving the vertex packing in perfect graphs [14]. Using well-known results [13, 23] about polynomial equivalence between the separation and optimization of linear functions on polytopes we obtain that there is a polynomial algorithm that takes a perfect graph \( G = (V, E) \), a rational vector \( a_v, v \in V \) as input, and checks whether the vector is in the fractional stable set polytope of \( G \) or not. If not, it finds a hyperplane (given by rational vectors) that separates \( a_v, v \in V \) from the polytope.

Therefore, there exists a polynomial separation algorithm for the fractional stable set polytope of a perfect graph with addition of the following equality: \( \sum_{v \in V} x_v = \alpha(G) \) where
 α(G) is the independence number of the given graph G. Again, using polynomial equivalence between separation and optimization, we have a polynomial algorithm for the following task:

$$\begin{align*}
\sum_{v \in K} x_v & \leq 1, \text{ where } K \text{ is a clique in } G \\
x_v & \geq 0, \forall v \in V \\
\sum_{v \in V} x_v & = \alpha(G) \\
\sum_{v \in V} w_v x_v & \to \min
\end{align*}$$

Since a half-space defined by an inequality $\sum_{v \in V} x_v \leq \alpha(G)$ contains the fractional stable set polytope, an affine set defined by $\sum_{v \in V} x_v = \alpha(G)$ is tangent to it. Therefore, if we add this inequality to the polytope all extreme points of the new polytope are the characteristic vectors of independent sets in G with cardinality $\alpha(G)$. Therefore, this task coincides with MMClique for the complement of G. Since the complement of a perfect graph is perfect, MMClique for perfect graphs is polynomial-time solvable, too.

**Definition 6.7.** A cycle $C_{2k+1}$, $k \geq 2$, is called an odd hole and its complement graph an odd antihole.

In Section 8 we will use the following conjecture of Berge, which was proved in [7].

**Theorem 6.8.** A graph is perfect if and only if it does not contain an induced subgraph isomorphic to an odd hole or antihole.

We say that a graph is of type $S_{2k+1}$, $k \geq 2$ if it is isomorphic to the graph with vertex set $\{0, 1, \ldots, 2k\}$, where vertices $i \pmod{2k+1}$, $i + 1 \pmod{2k+1}$ are not connected and vertices $i \pmod{2k+1}$, $i + 2 \pmod{2k+1}$ are connected. Other pairs can be connected arbitrarily. Obviously, every odd hole or antihole is of one of types $S_{2k+1}$, $k \geq 2$.

7. Arithmetical deadlocks

The key idea of the proof of the polynomial time case of Theorem 3.7 is to show that path- and arc-consistent instances of $2 - \text{MinHom}(\text{Inv}(F))$ have a perfect microstructure graph. We will prove this by showing that the microstructure graph forbids certain types of subgraphs. The exact formulation of the result can be found below in Theorem 8.1. This theorem uses the nonexistence of structures called arithmetical deadlocks which are introduced in this section.

**Definition 7.1.** Suppose $H$ is a conservative set of functions over $A$, $m \in H$ is an arithmetical operation on $B \subseteq \{\{x, y\} | x, y \in A, x \neq y\}$ and a pair $\phi, \psi \in H$ is a tournament pair on $B$. An instance of $2 - \text{MinHom}(\text{Inv}(H))$ with constraints pair $Un = (\rho_i)_{1 \leq i \leq n}, \text{Bin} = (\rho_{ii'})_{1 \leq i \neq i' \leq n}$ is called an odd arithmetical deadlock if there is a subset $\{i_0, \ldots, i_{k-1}\} \subseteq \{1, \ldots, n\}, k \geq 3$ of odd cardinality and $\{x_0, y_0\}, \ldots, \{x_{k-1}, y_{k-1}\} \in B$, such that for $0 \leq s \leq k - 1$: $\rho_{i_s,i_{s+1}} \cap (\{x_s, y_s\} \times \{x_{s+1}, y_{s+1}\}) = x_s \times y_{s+1}$, where $i + j$ denotes $i + j \pmod{K}$, the subset $\{i_0, \ldots, i_{k-1}\}$ is called a deadlock subset.

**Theorem 7.2.** Suppose $H$ is a conservative set of functions over $A$, $m \in H$ is an arithmetical operation on $B \subseteq \{\{x, y\} | x, y \in A, x \neq y\}$ and a pair $\phi, \psi \in H$ is a tournament pair on $B$. If an instance of $2 - \text{MinHom}(\text{Inv}(H))$ is arc- and path-consistent, then it cannot be an odd arithmetical deadlock.
We will begin by introducing some technical concepts from the theory of CSP which we will need in the proof of Theorem 7.2. An algebra \( A \) is said to be of type \( \mathfrak{F} \) if its operations are indexed by elements of the set \( \mathfrak{F} \), called terms. For every \( f \in \mathfrak{F} \), the corresponding operation is denoted by \( f^A \). The universe of an algebra \( A_i \) is denoted by \( A_i \). Recall that \( \rho^i = \{(y,x) \mid (x,y) \in \rho\} \).

**Definition 7.3.** Let a finite set of indexes \( I \) be given and every index \( i \in I \) corresponds to some algebra \( A_i \) of type \( \mathfrak{F} \). A set of indexed multi-domain predicates over \( \{A_i\}_{i \in I} \) is a pair \( \langle \rho_i \rangle_{i \in I}, \langle \rho_{kl} \rangle_{k \neq l \in I} \), where for each \( i \) and \( k \neq l \), \( \rho_i \) is a subalgebra of \( A_i \) and \( \rho_{kl} \) is a subalgebra of \( A_k \times A_l \). We assume that \( \rho_{kl} = \rho_{lk} \).

**Definition 7.4.** A set of indexed multi-domain predicates \( \langle \rho_i \rangle_{i \in I}, \langle \rho_{kl} \rangle_{k \neq l \in I} \) over \( \{A_i\}_{i \in I} \) is called *arc-consistent* if for distinct \( i, j \in I : Pr_1 \rho_{ij} = \rho_i, Pr_2 \rho_{ij} = \rho_j \).

**Definition 7.5.** A set of indexed multi-domain predicates \( \langle \rho_i \rangle_{i \in I}, \langle \rho_{kl} \rangle_{k \neq l \in I} \) over \( \{A_i\}_{i \in I} \) is called *path-consistent* if for any distinct \( i, j, k \in I : \rho_{ik} \subseteq \rho_{ij} \circ \rho_{jk} \).

We introduce the notation \( P_i = \{(x,y) \mid x, y \in A_i, x \neq y\} \).

**Definition 7.6.** Assume that algebras \( \{A_i\}_{i \in I} \) are of type \( \mathfrak{F} \), that they are conservative, and \( B_i \subseteq P_i, i \in I \). A term \( m \in \mathfrak{F} \) is called *arithmetical on \( B_i \}_{i \in I} \) if for any \( i \in I \) \( m^{A_i} \) is arithmetical on \( B_i \). A pair of terms \( \phi, \psi \in \mathfrak{F} \) is called a *tournament pair on \( B_i \}_{i \in I} \), if for any \( i \in I \) the pair \( \phi^{A_i}, \psi^{A_i} \) is a tournament pair on \( B_i \).

We now generalize the notion of an *odd arithmetical deadlock* to multi-domain constraints.

**Definition 7.7.** Assume that algebras \( \{A_i\}_{i \in I} \) are of type \( \mathfrak{F} \), that they are conservative, and \( B_i \subseteq P_i, i \in I \). Furthermore, assume \( m \in \mathfrak{F} \) is an arithmetical term on \( \{B_i\}_{i \in I} \) and a pair \( \phi, \psi \in \mathfrak{F} \) is a tournament pair on \( \{P_i \setminus B_i\}_{i \in I} \). Then, the set of indexed multi-domain predicates \( \langle \rho_i \rangle_{i \in I}, \langle \rho_{vw} \rangle_{v \neq w \in I} \) over \( \{A_i\}_{i \in I} \) is called an *odd arithmetical deadlock* if there is a subset \( \{i_0, \ldots, i_{k-1}\} \subseteq I, k \geq 3 \) of odd cardinality and \( \{x_s, y_s\} \subseteq B_{i_0}, \ldots, \{x_{k-1}, y_{k-1}\} \subseteq B_{i_{k-1}} \), such that for \( 0 \leq s \leq k - 1 \): \( \rho_{i_s, i_{s+1}} \cap (\{x_s, y_s\} \times \{x_{s+1}, y_{s+1}\}) = \mathfrak{F}^{x_s \times y_s \setminus x_{s+1} \times y_{s+1}} \), where \( i + j \) denotes \( i + j (\text{mod} \ k) \). The subset \( \{i_0, \ldots, i_{k-1}\} \) is called a *deadlock subset*.

We will now prove the following theorem, which is a generalization of Theorem 7.2.

**Theorem 7.8.** Suppose \( m \in \mathfrak{F} \) is an arithmetical term on \( \{B_i\}_{i \in I} \), and a pair \( \phi, \psi \in \mathfrak{F} \) is a tournament pair on \( \{P_i \setminus B_i\}_{i \in I} \). If a set of indexed multi-domain predicates \( \langle \rho_i \rangle_{i \in I}, \langle \rho_{kl} \rangle_{k \neq l \in I} \) over \( \{A_i\}_{i \in I} \) is arc- and path-consistent, then it cannot be an odd arithmetical deadlock.

Any instance of \( 2 - \text{MinHom} (\text{Inv} (H)) \) can be considered as a set of indexed multi-domain predicates over \( \{A_i\}_{i \in I} \) where \( I \) is a set of variables and \( A_i = A \). By defining \( B_i = B \) we see that Theorem 7.2 is a special case of Theorem 7.8. Before proving Theorem 7.8, we need to prove some preliminary lemmas.

Recall that a congruence of an algebra \( A \) is an equivalence relation on \( A \) that is a subalgebra of \( A^2 \). If \( \theta \) is a congruence of \( A \) and \( a \in A \), then equivalence class of \( \theta \) containing \( a \) is denoted by \( a^\theta \). If for each \( s \in I, \theta_s \) is a congruence of \( A_s \), then \( \rho_{i/\theta_i} = \{x^\theta \mid x \in \rho_i\} \) and \( \rho_{kl}/(\theta_k \times \theta_l) = \{(x^\theta, y^\theta) \mid (x, y) \in \rho_{kl}\} \), which we view as subalgebras of \( A_i/\theta_i \) and \( (A_k/\theta_k) \times (A_l/\theta_l) \).
Lemma 7.9. Let \( \theta_i \) be a congruence of \( A_i \) for each \( i \in I \) and assume that a set of indexed multi-domain predicates \( \{\rho_{i/\theta_i}\}_{i \in I} \), \( \{\rho_{i/\theta_i} \cap \theta_i\}_{i \in I} \) over \( \{A_i/\theta_i\}_{i \in I} \) is arc- and path-consistent. Then the set of indexed multi-domain predicates \( \{\rho_{i/\theta_i}\}_{i \in I} \), \( \{\rho_{i/\theta_i} \cap \theta_i\}_{i \in I} \) over \( \{A_i/\theta_i\}_{i \in I} \) is arc- and path-consistent, too.

Proof. Let \( n_i : A_i \to A_i/\theta_i \) be natural homomorphisms, i.e., \( n_i(x) = x^\theta_i \). Obviously, \( n_i(x) = n_i(x) = \{n_i(x) | x \in \rho_i\} \cap \theta_i = \{(n_i(x), n_i(y)) | (x, y) \in \rho_i\} \cap \theta_i \) and \( \Pr_1[\rho_i/(\theta_i \times \theta_i)] = \{n_k(x) | x \in \Pr_1 \rho_i\} = \Pr_1 \rho_i/\theta_i \). Analogously, we can prove that \( \Pr_2[\rho_i/(\theta_i \times \theta_i)] = \Pr_2 \rho_i/\theta_i \).

From arc-consistency it follows that \( \Pr_1 \rho_i = \rho_k, \Pr_2 \rho_i = \rho_l \) and we have \( \Pr_1[\rho_i/(\theta_i \times \theta_i)] = \rho_l/\theta_i \), \( \Pr_2[\rho_i/(\theta_i \times \theta_i)] = \rho_k/\theta_i \). This is equivalent to arc-consistency of the set \( \{\rho_{i/\theta_i}\}_{i \in I} \), \( \{\rho_i/\rho_i \cap \theta_i\}_{i \in I} \).

The path-consistency condition \( \rho_k \subseteq \rho_l \circ \rho_j \) gives us:

\[
\rho_{ij}/(\theta_i \times \theta_j) \circ \rho_{jk}/(\theta_j \times \theta_k) = \\
\{(n_i(x), n_j(y)) | (x, y) \in \rho_{ij}\} \circ \{(n_j(z), n_k(t)) | (z, t) \in \rho_{jk}\} = \\
\{(n_i(x), n_k(t)) | (x, y) \in \rho_{ij}, (y, t) \in \rho_{jk}\} \\
\supseteq \\
\{(n_i(x), n_k(t)) | (x, t) \in \rho_{ik}\} = \rho_{ik}/(\theta_i \times \theta_k)
\]

This is equivalent to path-consistency of \( \{\rho_{i/\theta_i}\}_{i \in I} \) and \( \{\rho_{i/\theta_i} \cap \theta_i\}_{i \in I} \).

For \( \rho \subseteq A_1 \times A_2 \), let \( \rho(x, \cdot) = \{y | \rho(x, y)\} \) and \( \rho(\cdot, x) = \{y | \rho(y, x)\} \).

Lemma 7.10. Suppose algebras \( \{A_i\}_{i=1}^{\infty} \) are conservative and \( B_i \subseteq P_i, i = 1, 2 \). Furthermore, assume that \( m \in \mathfrak{F} \) is an arithmetical term on \( B_i, i = 1, 2 \), and a pair \( \phi, \psi \in \mathfrak{F} \) is a tournament pair on \( P_i \setminus B_i, i = 1, 2 \). If \( \rho \) is a subalgebra of \( A_1 \times A_2 \) and there are \( \{x_i, y_i\} \subseteq B_i, i = 1, 2 \), such that \( \rho \cap (\{x_i, y_i\} \times \{x_2, y_2\}) = \emptyset \), then \( \rho(x_1, \cdot) \cap \rho(y_1, \cdot) = \emptyset \) and \( \rho(\cdot, x_2) \cap \rho(\cdot, y_2) = \emptyset \).

Proof. Suppose, for example, that \( t \in \rho(x_1, \cdot) \cap \rho(y_1, \cdot) \). Then, if \( \{x_2, t\} \subseteq B_2 \), we have:

\[
\begin{pmatrix} x_1 \\ t \end{pmatrix}, \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \begin{pmatrix} y_1 \\ x_2 \end{pmatrix} \in \rho \Rightarrow \\
\begin{pmatrix} m^{\alpha_1}(x_1, y_1, y_2) \\ m^{\beta_2}(t, t, x_2) \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \subseteq \rho
\]

If \( \{x_2, t\} \subseteq B_2 \), then there is a \( \lambda \in \mathfrak{F} : \lambda^{\alpha_1} \) such that \( \lambda \equiv \phi \) or \( \lambda \equiv \psi \) and we have:

\[
\begin{pmatrix} x_1 \\ t \end{pmatrix}, \begin{pmatrix} y_1 \\ x_2 \end{pmatrix} \in \rho \Rightarrow \\
\begin{pmatrix} \lambda^{\alpha_1}(x_1, y_1) \\ \lambda^{\beta_2}(t, x_2) \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \subseteq \rho
\]

By contradiction, we see that \( \rho(x_1, \cdot) \cap \rho(y_1, \cdot) = \emptyset \) (analogously \( \rho(\cdot, x_2) \cap \rho(\cdot, y_2) = \emptyset \)).

For \( \rho \subseteq A_1 \times A_2 \), let \( \theta_1^\rho \) and \( \theta_2^\rho \) denote the transitive closures of \( \rho \circ \rho \) and \( \rho \circ \rho \) respectively.

Lemma 7.11. Suppose algebras \( \{A_i\}_{i=1}^{\infty} \) are conservative and \( B_i \subseteq P_i, i = 1, 2 \). Suppose also that \( m \in \mathfrak{F} \) is an arithmetical term on \( B_i, i = 1, 2 \), and a pair \( \phi, \psi \in \mathfrak{F} \) is a tournament pair on \( P_i \setminus B_i, i = 1, 2 \). If \( \rho \) is a subalgebra of \( A_1 \times A_2 \) and there are \( \{x_i, y_i\} \subseteq B_i, i = 1, 2 \), such that \( \rho \cap (\{x_i, y_i\} \times \{x_2, y_2\}) = \emptyset \), then \( \theta_i^\rho \neq \rho \) if \( i = 1, 2 \).

Proof. Note that for \( x \in A_1 \), the equivalence class \( x^\theta_i \) can be obtained by the following procedure: \( U_1 = \{x\}, U_2 = \{t | y \in U_1 \rho(y, t)\}, U_3 = \{t | y \in U_2 \rho(t, y)\}, U_4 = \{t | y \in U_3 \rho(y, t)\} \) and so on. The resulting equivalence class is \( U_1 \cup U_2 \cup U_3 \cup \ldots \). Consider this process for elements \( x_1, y_1 \) and denote the corresponding sets by \( U_1^x, U_2^x, \ldots \) and
We prove by induction that $U^x_1 \cap U^y_1 = \emptyset$ and $\delta_k \triangleq (U^x_1)^2 \cup (U^y_1)^2$ is a congruence of $A_1 \mid U^x_1 \cup U^y_1$, if $k$ is odd, or of $A_2 \mid U^x_1 \cup U^y_1$, if $k$ is even.

Base of induction. Obviously, $U^x_1 \cap U^y_1 = \emptyset$. Since $\rho' = \rho \cap \{\{x_1, y_1\} \times \{x_2, y_2\}\} = \frac{x_1}{y_1} \times \frac{x_2}{y_2}$ is a subalgebra of $A_1 \mid \{x_1, y_1\}$, we see that $(U^x_1)^2 \cup (U^y_1)^2 = \theta^1_{\rho'}$ is a congruence of $A_1 \mid \{x_1, y_1\}$.

Suppose the assertion is true for $s \leq k$. Consider the case when $k$ is even (the odd case is analogous). Let $\sigma = \rho \cap (A_1 \times (U^x_1 \cup U^y_1))$. Clearly, $\rho' / (=A_1 \times \delta_k)$ is a subalgebra of $A_1 \times \left(A_2 \mid U^x_1 \cup U^y_1 / \delta_k\right)$ and from $y_2 \in U^x_1, x_2 \in U^y_1$ we have

\[
\begin{align*}
U^x_{k+1} &= \rho' / (=A_1 \times \delta_k) \left(\{x_1, y_1\} \times \left\{x_2 \delta_k, y_2 \delta_k\right\}\right) = \frac{x_1}{y_1} \times \frac{x_2 \delta_k}{y_2 \delta_k}.
\end{align*}
\]

From Lemma 7.10 we see that

\[
\rho' / (=A_1 \times \delta_k) \left(\{x_1, y_1\} \times \left\{x_2 \delta_k, y_2 \delta_k\right\}\right) \cap \rho' / (=A_2 \times \delta_k) \left(\{x_2 \delta_k, y_2 \delta_k\}\right) = \emptyset
\]

which is equivalent to $U^x_{k+1} \cap U^y_{k+1} = \emptyset$.

From the emptiness of this intersection, we conclude that the predicate $\sigma = \theta^1_{\rho'}$ is a congruence and equals to $(U^x_1)^2 \cup (U^y_{k+1})^2$, and the induction is completed.

**Lemma 7.12.** Suppose $A$ is a three-element algebra containing an operation $h : A^3 \to A$ that is arithmetical on $\{a, b\} | a, b \in A, a \neq b$. Then, there cannot be two different nontrivial (i.e. not equal to $A^3$ or $A^3$) congruences of this algebra.

**Proof.** We give a proof by contradiction. Without loss of generality we can assume that $A = \{0, 1, 2\}$ and $\sim^1 = \{(0, 0), (1, 1), (2, 2), (0, 1), (1, 0)\}$, $\sim^2 = \{(0, 0), (1, 1), (2, 2), (1, 2), (2, 1)\}$. Since $h$ preserve $\sim^1$, we have:

\[
\begin{align*}
h(1, 1, 2) &= 2 \\
h(0, 1, 2) &=?
\end{align*}
\]

Preservation of $\sim^2$ leads to contradiction:

\[
\begin{align*}
h(0, 1, 1) &= 0 \\
h(0, 1, 2) &=?
\end{align*}
\]

**Proof of Theorem 7.8.** Suppose to the contrary that there exists a set of indexed multidomain predicates that is an odd arithmetical deadlock. We can assume that $I = \{0, \ldots, 2d\}$ and $\{x_0, y_0\} \subset B_0, \ldots, \{x_{2d}, y_{2d}\} \subset B_{2d}$, such that $\rho_{k, k \oplus 1} \cap \{\{x_k, y_k\} \times \{x_{k \oplus 1}, y_{k \oplus 1}\}\} = \frac{x_k}{y_k} \times \frac{x_{k \oplus 1}}{y_{k \oplus 1}}$, where $i \oplus j$ denotes $i + j$ (mod $2d + 1$).

Consider the predicates $\rho_{k \oplus 1, k}$ and $\rho_{k, k \oplus 1}$ ($\ominus$ denotes a subtraction modulo $2d + 1$). Let $\theta -$ and $\theta +$ denote congruences $\theta^1_{\rho_{k \oplus 1, k}}, \theta^1_{\rho_{k, k \oplus 1}}$ respectively. By Lemma 7.11, $x_k^{\theta +} \neq y_k^{\theta +}$. 


Obviously, $\rho_{k,k+1}(x_k y_k) \subseteq y^\rho_k$ and $\rho_{k,k+1}(y_k x_k) \subseteq x^\rho_k$. Therefore, we conclude that

$$
\rho_{k,k+1} / ((\theta+) \times (=\Lambda_{k+1})) \cap \left( \{ x^\theta_k, y^\theta_k \} \times \{ x_k y_k \} \right) = \frac{x^\rho_k}{y^\rho_k} \times \frac{x^\rho_k}{y^\rho_k}.
$$

Let us show that $\rho_{k+1,k}(x_k y_k) \subseteq y^\rho_k$ and $\rho_{k+1,k}(y_k x_k) \subseteq x^\rho_k$. Suppose on the contrary that the first one is false (the other case is absolutely analogous), i.e. there exists $t \in \rho_{k+1,k}(x_k y_k)$ and $\rho_{k+1,k}(y_k x_k) \subseteq y^\rho_k$. From $\rho_{k+1,k}(x_k y_k)$, we see that $(t, y_k) \in \theta_-$. But, from $t \in x^\rho_k$, we conclude that $(t, x_k) \in \theta_+$. Consider the three-element algebra $A_k(x_k y_k)$. The congruences $\theta_+, \theta_-$ restricted to that algebra are equal to $\{ \{ x_k t \}, \{ y_k \} \}$ and $\{ \{ y_k, t \}, \{ x_k \} \}$, since, by Lemma 7.11, $x^\rho_k \not\subseteq y^\rho_k$ and $x^\rho_k \not\subseteq y^\rho_k$. It is easy to see that the three-element conservative algebra $A_k(x_k y_k)$ with $\{ x_k, y_k \} \in B_k$ has such congruences only if $m$ is arithmetical on $\{ \{ x_k, y_k \}, \{ y_k, t \}, \{ x_k, t \} \}$. This contradicts Lemma 7.12.

From $\rho_{k+1,k}(x_k y_k) \subseteq y^\rho_k$ and $\rho_{k+1,k}(y_k x_k) \subseteq x^\rho_k$, we conclude that

$$
\rho_{k+1,k}/ ((\theta+) \times (=\Lambda_{k+1})) \cap \left( \{ x_k y_k \} \times \{ x^\theta_k, y^\theta_k \} \right) = \frac{x_k}{y_k} \times \frac{x^\rho_k}{y^\rho_k}.
$$

Therefore, changing the system of algebras $\{ A_i \}_{i \in I}$ to $\{ A_i / \lambda_i \}_{i \in I}$ where

$$
\lambda_i = \begin{cases} 
\theta^\rho_{k,k+1} & \text{if } i = k \\
\Lambda_k & \text{otherwise}
\end{cases}
$$

we obtain, by Lemma 7.9, an arc- and path-consistent set of indexed predicates $\{ \rho_i / \lambda_i \}_{i \in I}$, $\{ \rho_{kl} / (\lambda_k \times \lambda_l) \}_{k,l \in I}$. The resulting set of predicates will be an odd arithmetical deadlock, too.

Analogously, we can prove that changing a system of one-type algebras $\{ A_i \}_{i \in I}$ to $\{ A_i / \lambda_i \}_{i \in I}$, where

$$
\lambda_i = \begin{cases} 
\rho_{2k+1,k} & \text{if } i = k \\
\Lambda_k & \text{otherwise}
\end{cases}
$$

results in an arc- and path-consistent set of indexed predicates $\{ \rho_i / \lambda_i \}_{i \in I}$, $\{ \rho_{kl} / (\lambda_k \times \lambda_l) \}_{k,l \in I}$, which will be an odd arithmetical deadlock.

By these transformations for different $k$ successively, we eventually obtain an arc- and path-consistent $\{ \rho_i' \}_{i \in I}$, $\{ \rho_{kl}' \}_{k,l \in I}$, such that $\forall k \rho_{k,k+1}' \cap \left( \{ x'_k, y'_k \} \times \{ x'_k, y'_k \} \right) =$ $x'_k \times y'_k$, and $\forall k \rho_{k,k+1}(x'_k, y'_k) = \{ y'_k \}$, $\rho_{k,k+1}(y'_k, x'_k) = \{ y'_k \}$, which contradicts that there is no such set.

From path-consistency we conclude that for any $0 \leq k < l \leq 2d$: $\rho_{kl}' \subseteq \rho_{k,k+1}' \circ \rho_{k+1,k+2}' \circ \cdots \circ \rho_{l-1,l}'$. Hence,

$$
\rho_{k,k+1}' \circ \rho_{k+1,k+2}' \circ \cdots \circ \rho_{l-1,l}'(x'_k, \cdot) = \begin{cases} 
\{ x'_l \} & \text{if } l - k \text{ even} \\
\{ y'_l \} & \text{if } l - k \text{ odd}
\end{cases}
$$

Since $\rho_{kl}'(x'_k, \cdot)$ is not empty, we see that

$$
\rho_{kl}'(x'_k, \cdot) = \begin{cases} 
\{ x'_l \} & \text{if } l - k \text{ even} \\
\{ y'_l \} & \text{if } l - k \text{ odd}
\end{cases}
$$

However, we have $\rho_{0,2d}' \cap \{ x'_0, y'_0 \} \times \{ x'_2d, y'_2d \} = \frac{x'_0}{y'_0} \times \frac{x'_2d}{y'_2d}$ which contradicts that $\rho_{0,2d}'(x'_0, \cdot) = \{ x'_2d \}$. 

\[ \blacksquare \]
8. Final step in the proof of tractability

**Theorem 8.1.** Suppose that $F$ satisfies the necessary local conditions and that the graph $T_F = (M^o,P)$ is bipartite. Then for every path- and arc-consistent instance of $2 - MinHom (Inv (F))$, its microstructure graph forbids subgraphs of type $S_{2p+1}, p \geq 2$.

**Proof.** Suppose to the contrary that we have an arc- and path-consistent instance $I = (X,U,B,w)$ of $2 - MinHom (Inv (F))$ with constraints pair $U = \langle \rho_i \rangle_{1 \leq i \leq n}$, $B = \langle \rho_k \rangle_{1 \leq k \neq l \leq n}$ and its microstructure graph has a subgraph of type $S_{2p+1}, p \geq 2$. For convenience, let us introduce $\rho_{ii} = \{(a,a) | a \in P\}$. Then, there is a set of pairs $\{(i_0,b_0),(i_1,b_1),\ldots,(i_{2p},b_{2p})\}$, such that for $0 \leq l \leq 2p$: $(b_l,b_{l+1}) \notin \rho_{ii} \circ \rho_{ii}$ and $(b_l,b_{l+1}) \in \rho_{ii} \circ \rho_{ii}$, where $i + j \equiv i \pmod{2p+1}$.

From $(b_l,b_{l+1}) \in \rho_{ii} \circ \rho_{ii}$ and the path-consistency condition $\rho_{ii} \circ \rho_{ii} \subseteq \rho_{ii} \circ \rho_{ii}$, we see that there is $a_{l+1}$ such that $(b_l,a_{l+1}) \in \rho_{ii} \circ \rho_{ii}$ and $(a_{l+1},b_{l+1}) \in \rho_{ii} \circ \rho_{ii}$.

Consider the predicate $\rho'_{l+1} = \rho_{ii} \circ \rho_{ii} \setminus \{a_{l+1} \times a_{l+1}, a_{l+1} \times b_{l+1}\} \in Inv (F)$. Obviously, $\rho'_{l+1}$ equals either $\{a_{l+1} \times b_{l+1}, a_{l+1} \times b_{l+1}\}$ or $\{a_{l+1} \times a_{l+1}, a_{l+1} \times b_{l+1}\}$.

Let us show that if $\{a_l,b_l\} \in \overline{M}$, then $\{a_{l+1},b_{l+1}\} \in \overline{M}$, too. Assume to the contrary that $\{a_{l+1},b_{l+1}\} \in M$. Then, by Theorem 5.3, there is a $\phi \in F : b_l \phi a_l$, where $\phi_0$ is a projection on the first coordinate. In this case, $\phi$ preserves neither $a_{l+1} \times b_{l+1}$ nor $a_{l+1} \times b_{l+1}$.

Hence, we need to consider two cases only: 1) $\forall l \{a_l,b_l\} \in M$ and 2) $\forall l \{a_l,b_l\} \in \overline{M}$. In the first case, we have $\{(a_l,b_l),\{a_{l+1},b_{l+1}\}\} \in P$, i.e., there is an odd cycle in $T_F$ which contradicts that $T_F$ is bipartite.

Now, consider the case $\forall l \{a_l,b_l\} \in \overline{M}$. By Theorem 5.4, there is a function $m \in F$, arithmetical on $\overline{M}$. If $\rho'_{l+1} = \{a_{l+1} \times b_{l+1}, a_{l+1} \times b_{l+1}\}$, then we have that

$$(b_l,b_{l+1}) = (m(a_l,a_l),m(b_l,b_{l+1},a_{l+1},a_{l+1})) \in \rho'_{l+1}$$

and $\rho'_{l+1} = \{a_{l+1} \times b_{l+1}, a_{l+1} \times b_{l+1}\}$.

Consider the set $\{i_0,i_1,\ldots,i_{2p}\}$. Suppose first that all $i_0,i_1,\ldots,i_{2p}$ are distinct. Then, Theorems 5.3 and 5.4 show us that we have an arithmetical operation $m \in F$ on $\overline{M}$ and a tournament pair $\phi,\psi \in F$ on $M$. It is easy to see that an instance of $2 - MinHom (Inv (F))$ with constraints pair $U = \rho_{ii} \circ \rho_{ii}$ and $B = \rho_k$ is odd and $\rho_k$ is an odd arithmetical deadlock where $\{i_0,i_1,\ldots,i_{2p}\}$ is a deadlock set. This contradicts that $I$ is arc- and path-consistent.

The case when the elements $i_0,i_1,\ldots,i_{2p}$ are not distinct cannot be reduced to the previous case by the following trick: introduce a new set of variables $X' = \{(i_0,0),(i_1,1),\ldots,(i_{2p},2p)\}$ and $\rho_{ii} = \rho_{ii},$ where $0 \leq s \leq 2p$. If $i_m \neq i_n$, then $\rho_{ii} = \rho_{ii}'$, else $\rho_{ii} = \rho_{ii}'$. It is easy to see that an instance with constraints pair $U = \rho_{ii}' \in X'$, $B = \rho_k$, satisfy the conditions of Theorem 7.2 and is an odd arithmetical deadlock, where the set $\{(i_0,0),(i_1,1),\ldots,(i_{2p},2p)\}$ is a deadlock set. Therefore, we have a contradiction. \[\blacksquare\]
Proof of polynomial case of Theorem 3.7. The conditions of Theorem 3.7 coincides with the conditions of Theorem 8.1 so the microstructure graph of an arc- and path-consistent instance forbids subgraphs of type $S_{2p+1}, p \geq 2$. By Theorem 6.8, it is perfect and, by Theorem 6.6, we see that the class $F$ is tractable.

Theorems 3.6 and 3.7 give the required dichotomy for conservative algebras, which implies the dichotomy for conservative constraint languages. By Theorem 2.9, we have the following general dichotomy.

**Theorem 8.2.** If $MinHom(\Gamma)$ is not tractable then it is NP-hard.

9. Tractable constraint languages

It is possible to reformulate our results in terms of constraint languages. Let $\text{lin}_{a_0,a_1}$ denote the predicate $\{(a_x, a_y, a_z)|x, y, z \in \{0, 1\}, x \oplus y \oplus z = 0\}$ where $\oplus$ denotes an addition modulo 2. For example, $\text{lin}_{0,1} = \{(x, y, z)|x, y, z \in \{0, 1\}, x \oplus y \oplus z = 0\}$.

**Theorem 9.1.** Suppose $\Gamma$ is a constraint language over $A$ which is a conservative relational clone, then either

- $\exists a \neq b \in A$ such that $\overline{a \overline{\oplus}}_b \in \Gamma$, or
- $\exists a \neq b \in A$ such that $\text{lin}_{a,b} \in \Gamma$, or
- $\exists a_0 \neq b_0, \ldots, a_{2k} \neq b_{2k} \in A$ such that $a_0 \overline{\oplus}_b b_0, b_{2k-1} \overline{\oplus}_b a_{2k}, b_{2k} \overline{\oplus}_b a_{0} \in \Gamma$, or
- $\Gamma$ is tractable.

**Proof.** Consider a functional clone $Pol(\Gamma)$ and the algebra $(A, Pol(\Gamma))$. Recall that the necessary local conditions are equivalent to requiring a conservative algebra to have only tractable 2-element subalgebras. It is obvious from Lemma 3.3 that a conservative algebra $F$ with domain set $\{a,b\}$ is NP-hard if and only if $\overline{b \overline{\oplus}}_a \in \text{Inv}(F)$ or $\overline{a \overline{\oplus}}_b \in \text{Inv}(F)$ or $\text{lin}_{a,b} \in \text{Inv}(F)$. Otherwise, it is tractable. Therefore, the necessary local conditions for $Pol(\Gamma)$ are equivalent to $\forall a \neq b \in A, \overline{b \overline{\oplus}}_a \notin \Gamma$ and $\text{lin}_{a,b} \notin \Gamma$.

Suppose $\Gamma$ does not satisfy the first two cases of the theorem, i.e., $Pol(\Gamma)$ satisfies the necessary local conditions. As is easily seen from the proof of the NP-hard case of Theorem 3.7, $\Gamma$ is NP-hard only if it contains predicates $b_0 \overline{\oplus}_b b_1, \ldots, b_{2k-1} \overline{\oplus}_b a_{2k}, b_{2k} \overline{\oplus}_b a_{0}$ for some $b_0, a_0, \ldots, b_{2k}, a_{2k}$. If we assume that for any $a_0 \neq b_0, \ldots, a_{2k} \neq b_{2k} \in A$ this system of predicates is not contained in $\Gamma$, then $\Gamma$ is tractable.

10. Related work and open problems

$MinHom$ can be viewed as a problem that fits the VCSP (Valued CSP) framework by [8]. By a valued predicate of arity $m$ over a domain $D$, we mean a function $p : D^m \rightarrow \mathbb{N} \cup \{\infty\}$. Informally, if $\Gamma$ is a finite set of valued predicates over a finite domain $D$, then an instance of $VCSP(\Gamma)$ is a set of variables together with specified subsets of variables restricted by valued predicates from $\Gamma$. Any assignment to variables can be considered a solution and the measure of this solution is the sum of the values that the valued predicates take under the assignments of the specified subsets of variables. The problem is to minimize this measure. It is widely believed that a dichotomy conjecture holds for $VCSP(\Gamma)$, too.

Our dichotomy result for $MinHom$ encourages us to consider generalizations that belong to this framework.
1. Suppose we are given a constraint language $\Gamma$ and a finite set of unary functions $F \subseteq \{ f : D \to \mathbb{N} \}$. Let $MinHom_F(\Gamma)$ denote a minimization problem which is defined completely analogously to $MinHom(\Gamma)$ except that we are restricted to minimizing functionals of the following form: $\sum_{i=1}^{n} \sum_{f \in F} w_{if}(x_i)$. A complete classification of the complexity of this problem is an open question.

2. Suppose we have a finite valued constraint language $\Gamma$, i.e. a set of valued predicates over some finite domain set. If $\Gamma$ contains all unary valued predicates, we call $VCSP(\Gamma)$ a conservative $VCSP$. This name is motivated by the fact that in this case the multmorphisms (which is a generalization of polymorphisms for valued constraint languages [8]) of $\Gamma$ must consist of conservative functions. Since there is a well-known dichotomy for conservative CSPs [5], we suspect that there is a dichotomy for conservative $VCSPs$.

3. $MinHom$ has (just as CSP) a homomorphism formulation. If we restrict ourselves to relational structures given by digraphs, we arrive to the following problem which we call digraph $MinHom$: given digraphs $S, H$ and weights $w_{ij}, i \in S, j \in H$, find a homomorphism $h : S \to H$ that minimizes the sum $\sum_{s \in S} w_{sh(s)}$. Suppose we have sets of digraphs $G_1, G_2$.

Then, $MinHom(G_1, G_2)$ denotes the digraph $MinHom$ problem when the first digraph is from $G_1$ and the second is from $G_2$. In this case, $MinHom(\{H\}, All)$ is always polynomially tractable and $MinHom(All, \{H\})$ coincides with $MinHom(\{H\})$ which is characterized in this paper.

Recently, Hell and Rafiey [27] announced a proof of dichotomy for $MinHom(\{H\})$ using another ideas. Their approach is based on the so called duality characterization of tractable digraphs, i.e. a statement of the form: a problem $MinHom(\{H\})$ is NP-hard if $H$ has certain induced substructures, otherwise $MinHom(\{H\})$ is polynomial-time solvable. In addition, they obtained a duality characterization of digraphs that have a Min–Max ordering, i.e. that preserve a transitive tournament pair. It is easy to show from their result that for a tractable digraph $H$ an algebra $Pol(H)$ has a very special structure not spanning all the possible tractable algebras. Thus, analysis of tractability for a constraint language is principally different from a case of one digraph.

We believe that ”duality approach” could be fruitful for characterizing the complexity of $MinHom(G_1, G_2)$ with other restrictions on $G_1, G_2$. For example, the closest candidate is $MinHom(G, G)$. Is there a dichotomy for this problem in general?

Acknowledgement

The author wishes to acknowledge fruitful discussions with Peter Jonsson and Andrei Bulatov.

References


