The Maximum Solution Problem on Graphs

Peter Jonsson*, Gustav Nordh**, and Johan Thapper***

Department of Computer and Information Science
Linköpings universitet
S-581 83 Linköping, Sweden
{petej,gusno,johth}@ida.liu.se

Abstract. We study the complexity of the problem MAX SOL which is a natural optimisation version of the graph homomorphism problem. Given a fixed target graph H with $V(H) \subseteq \mathbb{N}$, and a weight function $w : V(G) \to \mathbb{Q}^+$, an instance of the problem is a graph G and the goal is to find a homomorphism $f : G \to H$ which maximises $\sum_{v \in G} f(v) \cdot w(v)$. MAX SOL can be seen as a restriction of the MIN HOM-problem [Gutin et al., Disc. App. Math., 154 (2006), pp. 881-889] and as a natural generalisation of MAX ONES to larger domains. We present new tools with which we classify the complexity of MAX SOL for irreflexive graphs with degree less than or equal to 2 as well as for small graphs ($|V(H)| \leq 4$). We also study an extension of MAX SOL where value lists and arbitrary weights are allowed; somewhat surprisingly, this problem is polynomial-time equivalent to MIN HOM.

Keywords: constraint satisfaction, homomorphisms, computational complexity, optimisation

1 Introduction

Throughout this paper, by a graph we mean an undirected graph without multiple edges but possibly with loops. A *homomorphism* from a graph G to a graph H is a mapping f from $V(G)$ to $V(H)$ such that $(f(v), f(v'))$ is an edge of H whenever (v, v') is an edge of G. The homomorphism problem with a fixed target graph H takes a graph G as input and asks whether there is a homomorphism from G to H. Hence, by fixing the graph H we obtain a class of problems, one for each graph H. For example, the graph homomorphism problem with fixed target graph $H = \{(v_0, v_1), (v_1, v_0)\}$, denoted by HOM($H$), is exactly the problem of determining whether the input graph G is bipartite (i.e., the 2-COLORING problem). Similarly, if $H = \{(v_0, v_1), (v_1, v_0), (v_1, v_2), (v_2, v_1), (v_0, v_2), (v_2, v_0)\}$, then HOM($H$) is exactly the 3-COLORING problem. More generally, if H is the clique on k-vertices, then HOM(H) is the k-COLORING problem.

* Partially supported by the *Center for Industrial Information Technology* (CENIIT) under grant 04.01, and by the *Swedish Research Council* (VR) under grant 2006-4532.
** Supported by the *National Graduate School in Computer Science* (CUGS), Sweden.
*** Supported by the *Programme for Interdisciplinary Mathematics*, Department of Mathematics, Linköpings universitet.

Hence, the HOM(H) class of problems contains several well studied problems, some of which are in **P** (e.g, 2-COLORING) and others which are **NP**-complete (e.g., k-COLORING for $k \geq 3$). A celebrated result, due to Hell and Nesetril [9], states that HOM(H) is in **P** if H is bipartite or contains a looped vertex, and that it is **NP**-complete for all other graphs H. For more information on graph homomorphism problems in general and their complexity in particular, we refer the reader to the excellent monograph by Hell and Nesetril [10].

In this paper we study a natural optimisation variant of the HOM(H) problem, i.e., we are not only interested in the existence of a homomorphism but want to find the "best homomorphism". We let the vertices of H be a subset of the natural numbers, $w : V(G) \to \mathbb{Q}^+$ be a weight function and look for a homomorphism h from G to H that maximise the sum $\sum_{v \in G} w(v) \cdot h(v)$. We call this problem the maximum solution problem (with a fixed target graph H) and denote it by MAX SOL(H).

Just as the HOM(H) problem captures several interesting combinatorial *decision* problems, it is clear that the MAX SOL(H) captures several interesting combinatorial *optimisation* problems. The MAX SOL(H) problem where $H = \{(0,0),(0,1),(1,0)\}$ is exactly the **NP**-hard optimisation problem WEIGHTED MAXIMUM INDEPENDENT SET. MAX SOL can also be seen as a natural generalisation of MAX ONES or, alternatively, as a variation of the integer linear programming problem.

Gutin et al. [7] introduced MIN HOM, another homomorphism optimisation problem motivated by a real-world application in defence logistics. This problem was studied in [5, 6] and among other things, a dichotomy was established for (undirected) graphs. In particular, MIN HOM(H) was shown to be tractable whenever H is a *proper interval graph* or a *proper interval bigraph*. When formulated as a minimization problem, MAX SOL is easily seen to be a restriction of MIN HOM.

In [13], MAX SOL was studied as an optimisation variant of the constraint satisfaction problem over arbitrary constraint languages. There, languages defined using a many-valued logic were characterised as being either polynomial time solvable or **APX**-hard. This was accomplished by adopting algebraic techniques from the study of constraint satisfaction problems. In this paper, we continue the study of MAX SOL. We look at languages given by undirected graphs. In particular, we give a complete classification of the tractability of languages given by irreflexive graphs which have degree less than or equal to 2. We also classify the cases when $|V(H)| \leq 3$ and when $V(H) = \{0,1,2,3\}$. An interesting observation in these cases is that for some graphs, the complexity of the problem depends very subtly on the values of the vertices. In particular, applying an order preserving map on the values may change the complexity.

Furthermore, we consider two natural extensions of the MAX SOL-framework. One is to relax the restriction of the weights and allow arbitrary (possibly negative) rational weights on the variables. The other is to attribute a list, $L(v)$, of allowed values to each vertex v in the input instance. The list is a subset of $V(H)$ and any solution must assign v to one of the vertices in $L(v)$. In this paper we focus, apart from the ordinary MAX SOL, on the most general extreme, where we allow both lists and arbitrary weights. This problem, which we call LIST MAX AW SOL, can be seen both as an optimisation version of L-HOM(H), the *list homomorphism problem* (see [3, 4]) while it is still a restriction of MIN HOM. We show that for each undirected graph H, LIST MAX AW SOL(H) and MIN HOM(H) are in fact (polynomial time) equivalent.

The paper is organised as follows. In Section 2 we give a formal definition of CSP and the problems MAX SOL and LIST MAX AW SOL. In Section 3 we formalise the algebraic framework for studying MAX SOL. We also give a number of basic results which are used throughout the paper. These results are interesting in their own right, as many of them apply to general constraint languages. The results for MAX SOL are given in Section 4. In Section 5 we show the equivalence of LIST MAX AW SOL and Min Hom for undirected graphs, before concluding in Section 6.

2 Preliminaries

We formally define constraint satisfaction as follows: Let $D \subset \mathbb{N}$ (*the domain*) be a finite set. The set of all n-tuples of elements from D is denoted by D^n. Any subset of D^n is called an n-ary relation on D. The set of all finitary relations over D is denoted by R_D. A constraint language over a finite set, D, is a finite set $\Gamma \subseteq R_D$. Constraint languages are the way in which we specify restrictions on our problems. The constraint satisfaction problem over the constraint language Γ, denoted CSP(Γ), is defined to be the decision problem with instance (V, D, C), where V is a set of variables, D is a finite set of values (the domain), and C is a set of constraints $\{C_1, \ldots, C_q\}$, in which each constraint C_i is a pair (s_i, ϱ_i) with s_i a list of variables of length m_i, called the constraint scope, and ϱ_i an m_i-ary relation over the set D, belonging to Γ, called the constraint relation. The question is whether there exists a solution to (V, D, C) or not, that is, a function from V to D such that, for each constraint in C, the image of the constraint scope is a member of the constraint relation.

List Maximum Solution with Arbitrary Weights over a constraint language Γ, denoted LIST MAX AW SOL(Γ), is the maximization problem with

Instance: Tuple (V, D, C, L, w), where D is a finite subset of $\mathbb{N}$, (V, D, C) is a CSP instance over Γ, $L : V \to 2^D$ is a function from V to subsets of D, and $w : V \to \mathbb{Q}$ is a weight function.

Solution: An assignment $f : V \to D$ to the variables such that all constraints are satisfied and such that $f(v) \in L(v)$ for all $v \in V$.

Measure: $\sum_{v \in V} f(v) \cdot w(v)$

Weighted Maximum Solution over Γ, MAX SOL(Γ), is then defined by restricting w to non-negative rational numbers and letting $L(v) = D$ for all $v \in V$.

Let G be a graph. For a fixed graph H, the *Minimum Cost Homomorphism Problem* [7], MIN HOM(H), is the problem of finding a graph homomorphism f from G to H which minimises $\sum_{v \in V(G)} c_{f(v)}(v)$, where $c_i(v) \in \mathbb{Q}^+$ are costs, for $v \in V(G), i \in V(H)$.

Let G be a graph and H be a subgraph of G. H is a *retract* of G if there exists a graph homomorphism $f : G \to H$ such that $f(v) = v$ for all $v \in V(H)$. The *Retraction Problem*, RET(H), is to determine whether or not H is a retract of G.

Let $F = \{I_1, \ldots, I_k\}$ be a family of intervals on the real line. A graph G with $V(G) = F$ and $(I_i, I_j) \in E(G)$ if and only if $I_i \cap I_j \neq \emptyset$ is called an **interval graph**. If the intervals are chosen to be inclusion-free, G is called a **proper interval graph**.

Let $F_1 = \{I_1, \ldots, I_k\}$ and $F_2 = \{J_1, \ldots, J_l\}$ be two families of intervals on the real line. A graph G with $V(G) = F_1 \cup F_2$ and $(I_i, J_j) \in E(G)$ if and only if $I_i \cap J_j \neq \emptyset$ is called an **interval bigraph**. If the intervals in each family are chosen to be inclusion-free, G is called a **proper interval bigraph**.

Interval graphs are reflexive, while interval bigraphs are irreflexive and bipartite.

3 Methods

3.1 Algebraic framework

An operation on D is an arbitrary function $f : D^k \to D$. Any operation on D can be extended in a standard way to an operation on tuples over D, as follows: Let f be a k-ary operation on D and let R be an n-ary relation over D. For any collection of k tuples, $\boldsymbol{t_1}, \ldots, \boldsymbol{t_k} \in R$, define $f(\boldsymbol{t_1}, \ldots, \boldsymbol{t_k}) = (f(\boldsymbol{t_1}[1], \ldots, \boldsymbol{t_k}[1]), \ldots, f(\boldsymbol{t_1}[n], \ldots, \boldsymbol{t_k}[n]))$ where $\boldsymbol{t_j}[i]$ is the i-th component in tuple $\boldsymbol{t_j}$. If f is an operation such that for all $\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_k} \in R$, we have $f(\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_k}) \in R$, then R is said to be invariant under f. If all constraint relations in Γ are invariant under f, then Γ is invariant under f. An operation f such that Γ is invariant under f is called a *polymorphism* of Γ. The set of all polymorphisms of Γ is denoted $Pol(\Gamma)$. Sets of operations of the form Pol(Γ) are known as *clones*, and they are well-studied objects in algebra (cf. [15]).

A first-order formula ϕ over a constraint language Γ is said to be *primitive positive* (we say ϕ is a pp-formula for short) if it is of the form $\exists \mathbf{x} : (P_1(\mathbf{x}_1) \wedge \ldots \wedge P_k(\mathbf{x}_k))$ where $P_1, \ldots, P_k \in \Gamma$ and $\mathbf{x}_1, \ldots, \mathbf{x}_k$ are vectors of variables such that $|P_i| = |\mathbf{x}_i|$ for all i. Note that a pp-formula ϕ with m free variables defines an m-ary relation $R \subseteq D^m$; the relation R is the set of all tuples satisfying the formula ϕ. It follows, for example from the proof of [13, Lemma 4], that there is a polynomial time reduction from MAX SOL$(\Gamma \cup \{R\})$ to MAX SOL(Γ).

For any relation R and a unary operation f, let $f(R)$ denote the relation $f(R) = \{f(r) \mid r \in R\}$. Accordingly, let $f(\Gamma)$ denote the constraint language $\{f(R) \mid R \in \Gamma\}$.

Definition 1. *A constraint language Γ is a max-core if and only if there is no non-injective unary operation f in $Pol(\Gamma)$ such that $f(d) \geq d$ for all $d \in D$. A constraint language Γ' is a max-core of Γ if and only if Γ' is a max-core and $\Gamma' = f(\Gamma)$ for some unary operation $f \in Pol(\Gamma)$ such that $f(d) \geq d$ for all $d \in D$.*

We refer to [13] for the proof of the following lemma.

Lemma 1. *If Γ' is a max-core of Γ, then* MAX SOL(Γ) *and* MAX SOL(Γ') *are polynomial time equivalent.*

3.2 Basic lemmas

We now present a series of lemmas which will prove useful in the coming section. Several of the lemmas are, however, interesting in their own right and can be applied to a wider class of problems. Let Γ denote a constraint language over $D = \{d_1, \ldots, d_k\}$.

Lemma 2. *Arbitrarily choose $D' \subseteq D$, assume without loss of generality that $D' = \{d_1, \ldots, d_l\}$ with $d_1 < d_2 < \ldots < d_l$ and let $F = \{f \in Pol_1(\Gamma) \mid f|_{D'} = \mathbf{id}_{D'}\}$. Assume that there exists constants $a_1, \ldots, a_l > 0$ such that for every $f \in Pol_1(\Gamma) \setminus F$ it holds that $\sum_{i=1}^{l} a_i \cdot d_i > \sum_{i=1}^{l} a_i \cdot f(d_i)$. Then,* MAX SOL$(\Gamma \cup \{\{(d_1)\}, \ldots, \{(d_l)\}\})$ *and* MAX SOL(Γ) *are polynomial time equivalent problems.*

Proof. The non-trivial reduction follows from a construction which uses the concept of an *indicator problem* [11].

Corollary 1. *Let Γ be an arbitrary constraint language and let U be a unary relation on D. If* MAX SOL(Γ) *and* MAX SOL$(\Gamma \cup \{U\})$ *are polynomial time equivalent problems, then so are* MAX SOL(Γ) *and* MAX SOL$(\Gamma \cup \{\{(\max U)\}\})$. *In particular,* MAX SOL(Γ) *and* MAX SOL$(\Gamma \cup \{(d_k)\})$ *are polynomial time equivalent.*

Proof. Use Lemma 2 on $\Gamma \cup \{U\}$, $k = 1, d_1 = m$ and $a_1 > 0$.

Lemma 3. *Let Γ be a constraint language on a finite domain D. Assume that there is a set $F \subseteq Pol_1(\Gamma)$, such that for each $f \in F$,* MAX SOL$(f(\Gamma))$ *is in* **PO** *and such that for each choice of $a_1, \ldots, a_d \in \mathbb{Q}^+$ there is a $f \in F$ for which $\sum_{i=1}^{d} a_i \cdot d_i \leq \sum_{i=1}^{d} a_i \cdot f(d_i)$. Then,* MAX SOL$(\Gamma)$ *is in* **PO**.

Let $\mathcal{H} = \{H_1, \ldots, H_n\}$ be a set of connected graphs and let H be the disjoint union of these graphs. We are interested in the complexity of MAX SOL(H), given the complexities of the individual problems. Let $\mathcal{H}_i = \mathcal{H} \setminus \{H_i\}$. We say that H_i *extends* the set $\mathcal{H}_i$ if there exists an instance $I = (V, D, C, w)$ of MAX SOL(H_i) for which OPT$(I) >$ OPT(I_j) where $I_j = (V, D_j, \{xH_jy \mid xH_iy \in C\}, w)$, for all j such that $1 \leq j \neq i \leq n$. We call I a *witness* to the extension.

Assume that for some $1 \leq i \leq n$, it holds that H_i does not extend $\mathcal{H}_i$. It is clear that for any connected instance $I = (V, D, C, w)$ of MAX SOL(H), we have OPT$(I) =$ OPT(I_j) for some j, where $I_j = (V, D_j, \{xH_jy \mid xHy \in C\}, w)$. Furthermore, since H_i does not extend $\mathcal{H}_i$, we know that we can choose this $j \neq i$. Let H' be the disjoint union of the graphs in $\mathcal{H}_i$. Then, OPT$(I) =$ OPT(I'), where $I' = (V, D, \{xH'y \mid xHy \in C\}, w)$ is an instance of MAX SOL(H'). For this reason, we may assume that every $H_i \in \mathcal{H}$ extends every graph in $\mathcal{H}_i$.

Lemma 4. *Let $H_1, \ldots, H_n$ be connected graphs and H their disjoint union. If the problems* MAX SOL(H_i), $1 \leq i \leq n$ *are all tractable, then* MAX SOL(H) *is tractable. If* MAX SOL(H_i) *is* **NP**-*hard and H_i extends the set $\{H_1, \ldots, H_{i-1}, H_{i+1}, \ldots, H_n\}$ for some i, then* MAX SOL(H) *is* **NP**-*hard.*

The next lemma can be shown by a reduction from MAXIMUM INDEPENDENT SET.

Lemma 5. *If $a < b$ and $R = \{(a, a), (a, b), (b, a)\}$, then* MAX SOL$(R)$ *is* **NP**-*hard.*

4 Results for MAX SOL

Throughout this section, we will assume that all graphs defining constraint languages are max-cores and connected. Due to Lemma 1 and Lemma 4, we can do this without loss of generality. There is a straightforward reduction from MAX SOL to MIN HOM, so polynomiality results for MIN HOM translates directly to MAX SOL. Additionally, the following reduction can sometimes be used to show hardness.

Lemma 6. *Let H be a graph for which the retraction problem is* **NP***-complete. Then* MAX SOL$(H \cup \{\{(d_1)\}, \ldots, \{(d_k)\}\})$ *is* **NP***-hard.*

4.1 Irreflexive graphs with $\deg(v) \leq 2$

There are two types of irreflexive graphs H with $\deg(v) \leq 2$ for all $v \in V(H)$, paths and cycles. Since irreflexive paths are proper interval bigraphs, a reduction to MIN HOM, and [5, Corollary 2.6] shows that:

Proposition 1. *Let H be an irreflexive path. Then* MAX SOL(H) *is in* **PO***.*

When H is an odd cycle, we have that CSP(H) is **NP**-complete and therefore MAX SOL(H) is **NP**-hard. It remains to investigate even cycles. Since we do not allow multiple edges, C_2 is a single edge, for which MAX SOL is trivially in **PO**. When $H \cong C_4 \cong K_{2,2}$, there is always an increasing endomorphism from H to one of its edges. Thus no max-core is isomorphic to C_4. For even cycles of length greater or equal to 6, it has been shown in [4] that the retraction problem is **NP**-complete. We will use this with Lemma 6 to prove the **NP**-hard cases. The tractable cases are proven by Lemma 3. We will assume a bipartition $V(H) = \{d_1, \ldots, d_k\} \cup \{d'_1, \ldots, d'_k\}$ of H with $d_1 < d_2 < \cdots < d_k$ and $d'_1 < d'_2 < \cdots < d'_k$ and without loss of generality that $d_k > d'_k$.

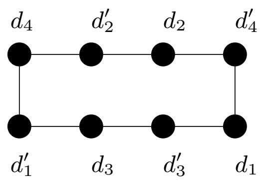

Fig. 1. The graph H in Proposition 3.

Proposition 2. *Let H be isomorphic to C_6 and a max-core. Then,* MAX SOL(H) *is* **NP***-hard.*

Proposition 3. *Let H be isomorphic to C_8 and a max-core. If H is isomorphic to the graph in Figure 1 and $(d_4 - d_3)(d'_4 - d'_3) \geq (d_3 - d_2)(d'_3 - d'_2)$, then* MAX SOL$(H)$ *is in* **PO***. Otherwise it is* **NP***-hard.*

In general, for even cycles, the following holds:

Proposition 4. *Let H be a max-core isomorphic to C_{2k}, $k \geq 3$. Then* MAX SOL(H) *and* MAX SOL$(H \cup \{\{(d_k)\}, \{(d'_k)\}\})$ *are polynomial time equivalent problems.*

Assume that there exists non-negative constants $a_1, \ldots, a_{k-1}, a'_1, \ldots, a'_{k-1}$ such that for each $f \in Pol_1(H) \setminus F$, where $F = \{f \in Pol_1(\Gamma) \mid \exists j \neq k : f(d_j) \neq d_j \vee f(d'_j) \neq d'_j\}$, it is true that

$$\sum_{i=1}^{k-1} \big(a_i \cdot d_i + a'_i \cdot d'_i\big) > \sum_{i=1}^{k-1} \big(a_i \cdot f(d_i) + a'_i \cdot f(d'_i)\big). \tag{1}$$

Then MAX SOL(H) *is* **NP***-hard, otherwise it is in* **PO***.*

4.2 Small Graphs

In this section we determine the complexity of MAX SOL(H) for all graphs $H = (V, E)$ on at most 4 vertices, i.e., $|V(H)| \leq 4$. For $|V(H)| = 4$ we only consider the case where $V = \{0, 1, 2, 3\}$, but for $|V(H)| \leq 3$ we classify the complexity for all $V(H) \subset \mathbb{N}$. In the process we discover a new tractable class for the MAX SOL(H) problem which is closely related to the critical independent set problem [1, 17].

We know from Lemma 1 that it is sufficient to consider graphs H that are max-cores. The case $|V(H)| = 1$ is trivial since there are only two such graphs and both are tractable. For $|V(H)| = 2$ there are two graphs that are max-cores, namely, the irreflexive path on two vertices (in **PO** by Proposition 1) and the graph on $V = \{d_1, d_2\}$, $d_1 < d_2$ where d_1 is adjacent to d_2 and d_1 is looped (which is **NP**-hard by Lemma 5).

When $|V(H)| = 3$ we have the following classification.

Theorem 1. *There are six (types of) max-cores over $\{d_1, d_2, d_3\}$ where $d_1 < d_2 < d_3$, denoted $H_1, \ldots, H_6$ and shown in Figure 2.* MAX SOL(H) *is* **NP***-hard for all of these except H_5.* MAX SOL(H_5) *is in* **PO** *if $d_3 + d_1 \leq 2d_2$ and* **NP***-hard otherwise.*

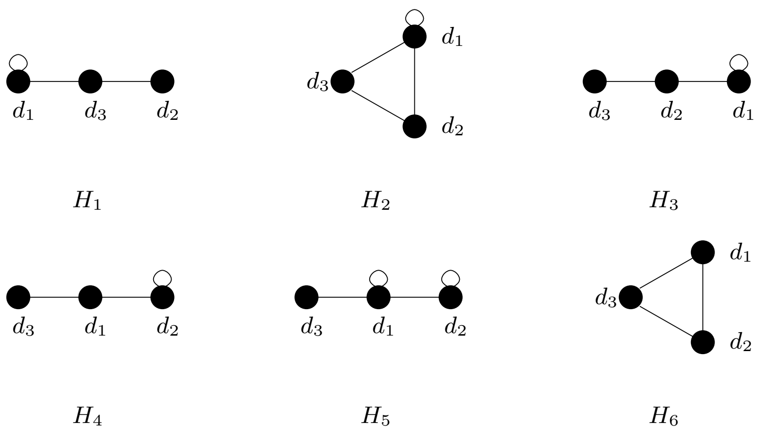

Fig. 2. The graphs H_i.

The MAX SOL(H_5) problem is related to the critical independent set problem [1, 17] in the following way. An independent set $I_C \subseteq V(G)$ is called critical if

$$|I_C| - |N(I_C)| = \max\{|I| - |N(I)| \mid I \text{ is an independent set in } G\},$$

where $N(I)$ denote the neighborhood of I, i.e., the set of vertices in G that are adjacent to at least one vertex in I. Zhang [17] proved that critical independent sets can be found in polynomial time.

We extend the notion of a critical independent sets to (k, m)-critical independent sets. A (k, m)-critical independent set is an independent set $I_C \subseteq V(G)$ such that

$$k \cdot |I_C| - m \cdot |N(I_C)| = \max\{k \cdot |I| - m \cdot |N(I)| \mid I \text{ is an independent set in } G\}.$$

Note that the maximum independent set problem is exactly the problem of finding a $(1, 0)$-critical independent set. The following proposition shows that that MAX SOL(H_5) is polynomial-time equivalent to the $(d_3 - d_2, d_2 - d_1)$-critical independent set problem.

Proposition 5. *I_C is a $(d_3 - d_2, d_2 - d_1)$-critical independent set in G if and only if the homomorphism h from G to H_5, defined by $h^{-1}(d_3) = I_C$ and $h^{-1}(d_1) = Nbd(I_C)$ is an optimal solution for* MAX SOL*(H_5).*

Proof. Assume that I_C is a $(d_3 - d_2, d_2 - d_1)$-critical independent set in G but h is not an optimal solution to MAX SOL(H_5), i.e., there exists a homomorphism g from G to H_5 such that $m(g) > m(h)$. That is,

$$w(g^{-1}(d_3)) \cdot d_3 + w(g^{-1}(d_1)) \cdot d_1 + w(g^{-1}(d_2)) \cdot d_2 >$$

$$w(h^{-1}(d_3)) \cdot d_3 + w(h^{-1}(d_1)) \cdot d_1 + w(h^{-1}(d_2)) \cdot d_2.$$

Subtracting $w(V(G)) \cdot d_2$ from both sides, we get

$$w(g^{-1}(d_3)) \cdot (d_3 - d_2) - w(g^{-1}(d_1)) \cdot (d_2 - d_1) >$$

$$w(h^{-1}(d_3)) \cdot (d_3 - d_2) - w(h^{-1}(d_1)) \cdot (d_2 - d_1).$$

This contradicts the fact that I_C is a $(d_3 - d_2, d_2 - d_1)$-critical independent set. The proof in the other direction is similar. □

Building upon the results in [1], we are able to completely classify the complexity of the (k, m)-critical independent set problem and, hence, also the complexity of MAX SOL(H_5). More specifically, we prove that the (k, m)-critical independent set problem is in **PO** if $k \le m$ and that it is **NP**-hard if $k > m$.

Finally, we present the complexity classification of MAX SOL for all graphs $H = (V, E)$ where $V = \{0, 1, 2, 3\}$. Just as in the case where $|V(H)| \le 3$ we make heavy use of the fact that only graphs that are max-cores need to be classified. Our second tool is the following lemma, stating that we can assume that we have access to all constants.

Lemma 7. *Let H be a max-core over $\{0, 1, 2, 3\}$. Then* MAX SOL(H) *is in* **PO** *(***NP***-hard) if and only if* MAX SOL$(H \cup \{\{(0)\}, \{(1)\}, \{(2)\}, \{(3)\}\})$ *is in* **PO** *(***NP***-hard).*

As an immediate corollary, we get that MAX SOL(H) is **NP**-hard for all max-cores H on $D = \{0, 1, 2, 3\}$ when the retraction problem (RET(H)) is **NP**-complete. Note that the complexity of the retraction problem for all graphs on at most 4 vertices have been classified in [16]. The classification is completed by considering the remaining max-cores (for which RET(H) is in **P**) one by one. Our result is the following.

Theorem 2. *Let H be a max-core on $D = \{0, 1, 2, 3\}$. Then,* MAX SOL(H) *is in* **PO** *if H is an irreflexive path, and otherwise,* MAX SOL(H) *is* **NP**-*hard.*

5 Results for LIST MAX AW SOL

The main theorem of this section is stated as follows.

Theorem 3. *Let H be an undirected graph with loops allowed. Then* LIST MAX AW SOL(H) *is solvable in polynomial time if all components of H are proper interval graphs or proper interval bigraphs. Otherwise,* LIST MAX AW SOL(H) *is* **NP**-*hard.*

Corollary 2. *Let H be an undirected graph with loops allowed. Then,* LIST MAX AW SOL(H) *is polynomial time equivalent to* MIN HOM(H).

The reduction from LIST MAX AW SOL(H) is easy. The lists are replaced by weights of ∞ for the appropriate variable-value pairs. Remaining weights are negated and translated so that the smallest weight becomes 0 for MIN HOM. The rest of this section is devoted to proving the other direction. We assume that the input instance is connected. If it is not, then we can solve each component separately and add the solutions.

Lemma 8. *Let H be an undirected graph. Then,* LIST MAX AW SOL(H) *is* **NP**-*hard if there exists a connected component H' of H such that* LIST MAX AW SOL(H') *is* **NP**-*hard. Otherwise, if for each connected component H' of H we have that* LIST MAX AW SOL(H') *is in* **PO**, *then* LIST MAX AW SOL(H) *is in* **PO**.

Lemma 9. *Let H be an undirected graph in which there exists both loop-free vertices and vertices with loops. Then,* LIST MAX AW SOL(H) *is* **NP**-*hard.*

Proof. This is proved by reduction from MAXIMUM INDEPENDENT SET.

Proposition 6. *If H is a connected graph which is a proper interval graph or a proper interval bigraph, then* LIST MAX AW SOL(H) *is polynomial time solvable.*

Proof. This follows from [5, Corollary 2.6] which states that the corresponding MIN HOM(H)-problem is polynomial time solvable. □

Theorem 4 (P. Hell, J. Huang [8]). *A bipartite graph H is a proper interval bigraph if and only if it does not contain an induced cycle of length at least six, or a bipartite claw, or a bipartite net, or a bipartite tent.*

Figure 3 displays the bipartite net and bipartite tent graphs.

Lemma 10. *Let H be a cycle of length at least six. Then* LIST MAX AW SOL(H) *is* **NP**-*hard.*

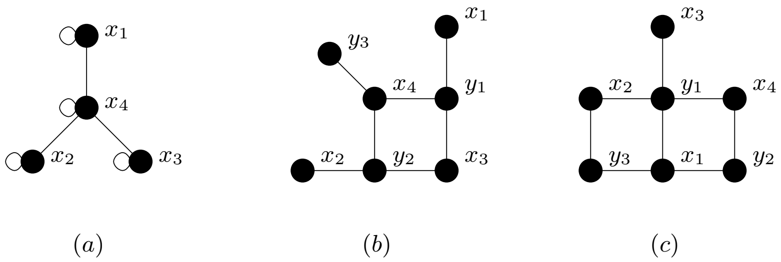

Fig. 3. (a) reflexive claw, (b) bipartite net, (c) bipartite tent.

Proof. The proof is by a simple reduction from the retraction problem on H. This problem is shown to be **NP**-complete in [4]. □

Lemma 11. *Let H be one of the graphs shown in Figure 3. Then* LIST MAX AW SOL(H) *is* **NP**-*hard.*

Proof. The proof follows the same ideas as those in [5]. That is, one reduces from the problem of finding a maximum independent set in a 3-partite graph. The apparent lack of expressive power of the LIST MAX AW SOL-framework, and the dependence on the labels of the target graph, are resolved by a precise choice of weights in the constructed instances. We carry out the case in Figure 3(a) in detail.

Let G be a 3-partite graph with partite sets X, Y and Z. We create an instance $I = (V, D, C, L, w)$ of LIST MAX AW SOL(H) as follows. Let $V = V(G)$, $D = V(H) = \{x_1, x_2, x_3, x_4\}$ and create a constraint uHv in C for each $(u, v) \in E(G)$. Now, define the lists and weights as follows.

$$L(u) = \begin{cases} \{x_4, x_1\} & \text{when } u \in X \\ \{x_4, x_2\} & \text{when } u \in Y \\ \{x_4, x_3\} & \text{when } u \in Z. \end{cases}$$

$$w(u) = \begin{cases} 1/(x_1 - x_4) & \text{when } u \in X \\ 1/(x_2 - x_4) & \text{when } u \in Y \\ 1/(x_3 - x_4) & \text{when } u \in Z. \end{cases}$$

Now, if s is a solution to I, let $X_1 = s^{-1}(x_1)$, $X_0 = X \setminus X_1$ and define similarly Y_0, Y_1 and Z_0, Z_1. Note that s defines an independent set $X_1 \cup Y_1 \cup Z_1$ of G. Conversely, it is also clear that any independent set of G yields a solution to I by assigning each variable to x_4 precisely when it is not a part of the independent set. The value of s can be written as

$$\sum_{u \in V} s(u) \cdot w(u) = \sum_{x \in X} s(x) \cdot w(x) + \sum_{y \in Y} s(y) \cdot w(y) + \sum_{z \in Z} s(z) \cdot w(z) =$$

$$\frac{|X_0| \cdot x_4 + |X_1| \cdot x_1}{x_1 - x_4} + \frac{|Y_0| \cdot x_4 + |Y_1| \cdot x_2}{x_2 - x_4} + \frac{|Z_0| \cdot x_4 + |Z_1| \cdot x_3}{x_3 - x_4} =$$

$$\frac{|X| \cdot x_4}{x_1 - x_4} + |X_1| + \frac{|Y| \cdot x_4}{x_2 - x_4} + |Y_1| + \frac{|Z| \cdot x_4}{x_3 - x_4} + |Z_1| = M + |X_1| + |Y_1| + |Z_1|,$$

where M is independent of s and can be calculated in polynomial time from I. Thus, an optimal solution to I gives a maximal independent set in G. □

We now have all the tools necessary to complete the proof of Theorem 3.

Proof of Theorem 3. According to Lemma 8 we can assume that H is connected. Furthermore, due to Lemma 9 we can assume that H is either loop-free or reflexive, or LIST MAX AW SOL(H) is **NP**-hard. Proposition 6 gives the polynomial cases.

If H is loop-free and non-bipartite, we can reduce from HOM(H), which is **NP**-complete for non-bipartite graphs. So assume that H is bipartite. If H is not a proper interval bigraph, then, due to Theorem 4, H has either an induced cycle of length at least 6, an induced bipartite claw, an induced bipartite net or an induced bipartite tent. We can use the lists L to induce each of these graphs, so **NP**-hardness follows from Lemma 10 and 11. Note that hardness for the reflexive claw implies hardness for the bipartite claw.

Finally, if H is reflexive, then it is either not an interval graph, or a non-proper interval graph. If H is not an interval graph, then we can reduce from the list homomorphism problem L-HOMH which is shown to be **NP**-complete for reflexive, non-interval graphs in [3]. In the second case, it has been shown by Roberts [14] that H must contain an induced claw. Lemma 11 shows that this problem is **NP**-hard, which finishes the proof. □

6 Discussion and future work

In this paper we have initiated a study of the complexity of the maximum solution problem on graphs. Our results indicate that giving a complete complexity classification of MAX SOL(H) for every fixed graph H is probably harder than first anticipated. In particular, the new tractable class for the MAX SOL problem identified in Section 4.2 depends very subtly on the values of the domain elements and we have not yet been able to characterize this tractable class in terms of polymorphisms. Hence, this tractable class seems to be of a very different flavour compared to the previously identified tractable classes for the MAX SOL problem [13].

On the other hand, we are able to give a complete classification for the complexity of the arbitrary weighted list version of the problem, LIST MAX AW SOL(H). Interestingly, the borderline between tractability and **NP**-hardness for LIST MAX AW SOL(H) coincide exactly with Gutin et al.'s [5] recent complexity classification of MIN HOM(H). This is surprising, since the MIN HOM(H) problem is much more expressive than the LIST MAX AW SOL(H) problem, and hence, we were expecting graphs H such that MIN HOM(H) were **NP**-hard and LIST MAX AW SOL(H) were in **PO**. The obvious question raised by this result is how far can we extend the agreement in complexity between LIST MAX AW SOL(Γ) and MIN HOM(Γ)? To this end, we state the MIN HOM problem for general constraint languages.

Minimum Cost Homomorphism over constraint language Γ, denoted MIN HOM(Γ), is the minimization problem with

Instance: Tuple $(V, D, C, c_i(v))$, where D is a finite subset of $\mathbb{N}$, (V, D, C) is a CSP instance over Γ, $c_i : V \to Q^+$ are costs for $i \in V(H)$.
Solution: An assignment $f : V \to D$ to the variables such that all constraints are satisfied.
Measure: $\sum_{v \in V} c_{f(v)}(v)$

Problem 1. Is it the case that the complexity of LIST MAX AW SOL(Γ) and MIN HOM(Γ) are equal for all constraint languages Γ?

It can be shown, using results from [2, 12], that LIST MAX AW SOL(Γ) and MIN HOM(Γ) are polynomial time equivalent when Γ is a boolean constraint language.

References

1. A. Ageev. On finding critical independent and vertex sets. *SIAM J. Discrete Math.*, 7(2):293–295, 1994.
2. D. Cohen, M. Cooper, P. Jeavons, and A. Krokhin. The complexity of soft constraint satisfaction. *Artif. Intell.*, 170(11):983–1016, 2006.
3. T. Feder and P. Hell. List homomorphisms to reflexive graphs. *J. Combin. Th. (B)*, 72:236–250, 1998.
4. T. Feder, P. Hell, and J. Huang. List Homomorphisms and Circular Arc Graphs. *Combinatorica*, 19:487–505, 1999.
5. G. Gutin, P. Hell, A. Rafiey, and A. Yeo. A dichotomy for minimum cost graph homomorphisms . *European J. Combin.*, To appear.
6. G. Gutin, A. Rafiey, and A. Yeo. Minimum cost and list homomorphisms to semicomplete digraphs. *Discrete Applied Mathematics*, 154(6):890–897, 2006.
7. G. Gutin, A. Rafiey, A. Yeo, and M. Tso. Level of repair analysis and minimum cost homomorphisms of graphs. *Discrete Applied Mathematics*, 154(6):881–889, 2006.
8. P. Hell and J. Huang. Interval bigraphs and circular arc graphs. *J. Graph Theory*, 46:313–327, 2004.
9. P. Hell and J. Nešetřil. The complexity of H-coloring. *J. Combinatorial Theory B*, 48:92–110, 1990.
10. P. Hell and J. Nešetřil. *Graphs and Homomorphisms*. Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2004.
11. P. Jeavons, D. Cohen, and M. Gyssens. A test for tractability. In *Proceedings of the 2nd International Conference on Principles and Practice of Constraint Programming (CP-96)*, pages 267–281, 1996.
12. P. Jonsson. Boolean constraint satisfaction: complexity results for optimization problems with arbitrary weights. *Theoretical Computer Science*, 244(1–2):189–203, 2000.
13. P. Jonsson and G. Nordh. Generalised integer programming based on logically defined relations. In *Proceedings of the 31st International Symposium on Mathematical Foundations of Computer Science (MFCS-2006)*, pages 549–560, 2006.
14. F. Roberts. Indifference graphs. In F. Harary, editor, *Proof Techniques in Graph Theory*, pages 139–146. Academic Press, 1969.
15. Á. Szendrei. *Clones in Universal Algebra*, volume 99 of *Seminaires de Mathématiques Supérieures*. University of Montreal, 1986.
16. N. Vikas. A complete and equal computational complexity classification of compaction and retraction to all graphs with at most four vertices and some general results. *J. Comput. Syst. Sci.*, 71(4):406–439, 2005.
17. C.-Q. Zhang. Finding critical independent sets and critical vertex subsets are polynomial problems. *SIAM J. Discrete Math.*, 3(3):431–438, 1990.