Toward a Dichotomy for Approximation of H-Coloring

Akbar Rafiey
Department of Computing Science, Simon Fraser University, Burnaby, Canada
arafiey@sfu.ca

Arash Rafiey
Indiana State University, Terre Haute, IN, USA
Simon Fraser University, Burnaby, Canada
arash.rafiey@indstate.edu

Thiago Santos
Indiana State University, Terre Haute, IN, USA
tsantos2@sycamores.indstate.edu

Abstract

Given two (di)graphs G, H and a cost function $c : V(G) \times V(H) \to \mathbb{Q}_{\geq 0} \cup \{+\infty\}$, in the minimum cost homomorphism problem, MinHOM(H), we are interested in finding a homomorphism $f : V(G) \to V(H)$ (a.k.a H-coloring) that minimizes $\sum_{v \in V(G)} c(v, f(v))$. The complexity of *exact minimization* of this problem is well understood [35], and the class of digraphs H, for which the MinHOM(H) is polynomial time solvable is a small subset of all digraphs.

In this paper, we consider the approximation of MinHOM within a constant factor. In terms of digraphs, MinHOM(H) is not approximable if H contains a *digraph asteroidal triple (DAT)*. We take a major step toward a dichotomy classification of approximable cases. We give a dichotomy classification for approximating the MinHOM(H) when H is a graph (i.e. symmetric digraph). For digraphs, we provide constant factor approximation algorithms for two important classes of digraphs, namely bi-arc digraphs (digraphs with a *conservative semi-lattice polymorphism* or *min-ordering*), and k-arc digraphs (digraphs with an *extended min-ordering*). Specifically, we show that:

- **Dichotomy for Graphs:** MinHOM(H) has a $2|V(H)|$-approximation algorithm if graph H admits a *conservative majority polymorphims* (i.e. H is a *bi-arc graph*), otherwise, it is inapproximable;
- MinHOM(H) has a $|V(H)|^2$-approximation algorithm if H is a bi-arc digraph;
- MinHOM(H) has a $|V(H)|^2$-approximation algorithm if H is a k-arc digraph.

In conclusion, we show the importance of these results and provide insights for achieving a dichotomy classification of approximable cases. Our constant factors depend on the size of H. However, the implementation of our algorithms provides a much better approximation ratio. It leaves open to investigate a classification of digraphs H, where MinHOM(H) admits a constant factor approximation algorithm that is independent of $|V(H)|$.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases Approximation algorithms, minimum cost homomorphism, randomized rounding

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.91

Category Track A: Algorithms, Complexity and Games

Related Version Full version hosted on `https://arxiv.org/abs/1902.02201`.

Funding *Arash Rafiey*: Supported by NSF 1751765.

Acknowledgements We are thankful to Andrei Bulatov for proofreading several drafts of the work and many valuable discussions that significantly improved the paper and its presentation.

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).
Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;
Article No. 91; pp. 91:1–91:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

For a digraph D, let $V(D)$ denote the vertex set of D, and let $A(D)$ denote the arcs of D. We denote the number of vertices of D by $|D|$. Instead of $(u, v) \in A(D)$, we use the shorthand $uv \in A(D)$ or simply $uv \in D$. A graph G is a symmetric digraph, that is, $xy \in A(G)$ if and only if (iff) $yx \in A(G)$. An edge is just a symmetric arc.

A *homomorphism* of a digraph D to a digraph H (a.k.a H-COLORING) is a mapping $f : V(D) \to V(H)$ such that for each arc xy of D, $f(x)f(y)$ is an arc of H. We say mapping f does not satisfy arc xy, if $f(x)f(y)$ is not an arc of H. The homomorphism problem for a fixed target digraph H, HOM(H), takes a digraph D as input and asks whether there is a homomorphism from D to H. Therefore, by fixing the digraph H we obtain a class of problems, one problem for each digraph D. For example, HOM(H), when H is an edge, is exactly the problem of determining whether the input graph G is bipartite (i.e., the 2-COLORING problem). Similarly, if $V(H) = \{u, v, x\}$, $A(H) = \{uv, vu, vx, xv, ux, xu\}$, then HOM($H$) is exactly the classical 3-COLORING problem. More generally, if H is a clique on k vertices, then HOM(H) is the K-COLORING problem. The H-COLORING problem can be considered within a more general framework, the constraint satisfaction problem (CSP). In the CSP associated with a finite relational structure $\mathbb{H}$, CSP($\mathbb{H}$), the question is whether there exists a homomorphism of a given finite relational structure to $\mathbb{H}$. Thus, the H-COLORING problem is a particular case of the CSP in which the involved relational structures are digraphs. A celebrated result due to Hell and Nesetril [31], states that, for graph H, 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. See [9] for an algebraic proof of the same result, and [12, 55] for a dichotomy for CSP($\mathbb{H}$).

There are several natural optimization versions of the HOM(H) problem. One is to find a mapping $f : V(D) \to V(H)$ that maximizes (minimizes) number of satisfied (unsatisfied) arcs in D. This problem is known under the name of MAX 2-CSP (MIN 2-CSP). For example, the most basic BOOLEAN MAX 2-CSP problem is MAX CUT where the target graph H is an edge. This line of research has received a lot of attention in the literature and there are very strong results concerning various aspects of approximability MAX 2-CSP and MIN 2-CSP [2, 22, 28, 41, 45]. See [47] for a recent survey on this and approximation of MAX K-CSP and MIN K-CSP. We consider another natural optimization version 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". The *minimum cost homomorphism problem* to H, denoted by MinHOM(H), for a given input digraph D, and a cost function $c(x, i), x \in V(D), i \in V(H)$, seeks a homomorphism f of D to H that minimizes the total cost $\sum_{x \in V(D)} c(x, f(x))$. The cost function c can take non-negative rational values and positive infinity, that is $c : V(D) \times V(H) \to \mathbb{Q}_{\geq 0} \cup \{+\infty\}$. The MinHOM was introduced in [25], where it was motivated by a real-world problem in defence logistics. The MinHOM problem offers a natural and practical way to model and generalizes many optimization problems.

▶ **Example 1** ((WEIGHTED) MINIMUM VERTEX COVER)**.** This problem can be seen as MinHOM(H) where $V(H) = \{0, 1\}, E(H) = \{11, 01\}$ and $c(u, 0) = 0$, $c(u, 1) > 0$ for every $u \in V(G)$. Note that G and H are graphs.

▶ **Example 2** (LIST HOMOMORPHISM (LHOM))**.** LHOM(H), seeks, for a given input digraph D and lists $L(x) \subseteq V(H), x \in V(D)$, a homomorphism f from D to H such that $f(x) \in L(x)$ for all $x \in V(D)$. This is equivalent to MinHOM(H) with $c(u, i) = 0$ if $i \in L(u)$, otherwise $c(u, i) = +\infty$. This problem is also known as LIST H-COLORING and its complexity is fully understood due to series of results [5, 8, 10, 11, 18, 33].

The MinHOM problem generalizes many other problems such as (WEIGHTED) MIN ONES [1, 15, 40], MIN SOL [39, 53], a large class of bounded integer linear programs, retraction problems [19], MINIMUM SUM COLORING [4, 21, 44], and various optimum cost chromatic partition problems [27, 37, 38, 43].

A special case of MinHOM problem is where the cost function c is choosen from a fixed set Δ. This problem is denoted by MinHOM(H, Δ) [14, 53, 54]. The VALUED CONSTRAINED SATISFACTION PROBLEMS (VCSPs) is a generalization of this special case of the MinHOM problem. An instance of the VCSP is given by a collection of variables that must be assigned labels from a given domain with the goal to minimize the objective function that is given by the sum of cost functions, each depending on some subset of the variables [13]. Interestingly, a recent work by Cohen *et al.* [14] proved that VCSPs over a fixed valued constraint language are polynomial-time equivalent to MinHOM(H, Δ) over a fixed digraph and a proper choice of Δ.

Exact Minimization. The complexity of *exact minimization* of MinHOM(H) was studied in a series of papers, and complete complexity classifications were given in [23] for undirected graphs, in [35] for digraphs, and in [51] for more general structures. Certain minimum cost homomorphism problems have polynomial time algorithms [23, 24, 25, 35], but most are NP-hard. We remark that, the complexity of *exact minimization* of VCSPs is well understood [42, 52].

Approximation. For a minimization problem, an α*-approximation* algorithm is a (randomized) polynomial-time algorithm that finds an approximate solution of cost at most α times the minimum cost. A constant ratio approximation algorithm is an α-approximation algorithm for some constant α. We say a problem is not approximable if there is no polynomial time approximation algorithm with a multiplicative guarantee unless $\mathrm{P} = \mathrm{NP}$. The approximability of MinHOM is fairly understood when we restrict the cost function to a fixed set Δ, and further, we restrict it to take only finite values (not ∞). This setting is a special case of finite VCSPs, and there are strong approximation results on finite VCSPs. For finite VCSPs, Raghavendra [50] showed how to use the basic SDP relaxation to obtain a constant approximation. Moreover, he proved that the approximation ratio cannot be improved under UNIQUE GAME CONJECTURE (UGC). This constant is not explicit, but there is an algorithm that can compute it with any given accuracy in doubly exponential time. In another line of research, the power of so-called *basic linear program (BLP)* concerning constant factor approximation of finite VCSPs has been recently studied in [16, 17]. However, the approximability of VCSPs for constraint languages that are not finite-valued remains poorly understood, and [30, 39] are the only results on approximation of VCSP for languages that have cost functions that can take infinite values.

Hell *et al.,* [30] proved a dichotomy for approximating MinHOM(H) when H is a bipartite graph by transforming the MinHOM(H) to a linear program, and rounding the fractional values to get a homomorphism to H.

▶ **Theorem 3** (Dichotomy for bipartite graphs [30]). *For a fixed bipartite graph H, MinHOM(H) admits a constant factor approximation algorithm if H admits a* min-ordering *(complement of H is a* circular arc *graph), otherwise MinHOM(H) is not approximable unless $P = NP$.*

Beyond this, there is no result concerning the approximation of MinHOM(H). We go beyond bipartite case and present a constant factor approximation algorithm for *bi-arc* graphs (graphs with a *conservative majority* polymorphism). Designing an approximation algorithm for MinHOM(H) when H is a digraph is much more complex than when H is a graph. We improve state-of-the-art by providing constant factor approximation algorithms for MinHOM(H) where H belongs to these two important cases of digraphs, namely *bi-arc* digraphs (digraphs with a *conservative semi-lattice* polymorphism a.k.a min-ordering), and *k-arc* digraphs (digraphs with a k-min-ordering). To do so, we introduce new LPs that reflect the structural properties of the target (di)graph H as well as new methods to round the fractional solutions and obtain homomorphisms to H. We will show our randomized rounding procedure can be de-randomized, and hence, we get a deterministic polynomial algorithm. Furthermore, we argue that our techniques can be used towards finding a dichotomy for the approximation of MinHOM(H).

1.1 Our Contributions

Most of the minimum cost homomorphism problems are NP-hard, therefore we investigate the approximation of MinHoM(H).

> Approximating Minimum Cost Homomorphism to Digraph H.
> *Input:* A digraph D and a vertex-mapping costs $c(x,u), x \in V(D), u \in V(H)$,
> *Output:* A homomorphism f of D to H with the total cost of $\sum_{x \in V(D)} c(x, f(x)) \leq \alpha \cdot OPT$, where α is a constant.

Here, OPT denotes the cost of a minimum cost homomorphism of D to H. Moreover, we assume size of H is constant. Recall that we approximate the cost over real homomorphisms, rather than approximating the maximum weight of satisfied constraints, as in, say, Max Csp. One can show that if LHOM(H) is not polynomial time solvable then there is no approximation algorithm for MinHOM(H) [30, 48].

▶ **Observation 4.** *If LHOM(H) is not polynomial time solvable, then there is no approximation algorithm for MinHOM(H).*

The complexity of the LHOM problems for graphs, digraphs, and relational structures (with arity two and higher) have been classified in [18, 33, 10] respectively. LHOM(H) is polynomial time solvable if the digraph H does not contain a *digraph asteroidal triple (DAT)*[1] as an induced sub-digraph, and NP-complete when H contains a DAT [33].

MinHOM(H) is polynomial time solvable when digraph H admits a k-min-max-ordering, a subclass of DAT-free digraphs, and otherwise, NP-complete [35, 34]. Here, in this paper, we take an important step towards closing the gap between DAT-free digraphs and the one that admit a k-min-max-ordering. First, we consider digraphs that admit a min-ordering. Digraphs that admit a min-ordering have been studied under the name of *bi-arc* digraphs [36] and *signed interval* digraphs [29]. Deciding if digraph H has a min-ordering and finding a min-ordering of H is in P [36]. We provide a constant factor approximation algorithm for MinHOM(H) where H admits a min-ordering.

▶ **Theorem 5** (Digraphs with a min-ordering). *If digraph H admits a min-ordering, then MinHOM(H) has a constant factor approximation algorithm.*

Sections 4,5 are dedicated to the proof of Theorem 5. In section 6, we turn our attention to digraphs with k-min-orderings, for integer $k > 1$. They are also called digraphs with *extended X-underbar* [3, 26, 46]. It was shown in [26] that if H has the X-underbar property,

[1] The definition of DAT is rather technical and it is not necessary to fully understand it in this paper.

then the HOM(H) problem is polynomial time solvable. In Lemma 21, we show that if H admits a k-min-ordering, then H is a DAT-free digraph, and provide a simple proof that LHOM(H) is polynomial time solvable. Finally, we have the following theorem.

▶ **Theorem 6** (Digraphs with a k-min-ordering)**.** *If digraph H admits a k-min-ordering for some integer $k > 1$, then MinHOM(H) has a constant factor approximation algorithm.*

Considering graphs, Feder *et al.,* [18] proved that LHOM(H) is polynomial time solvable if H is a *bi-arc* graph, and is NP-complete otherwise. In the same paper, they showed graph H is a bi-arc graph iff it admits a conservative majority polymorphism. In Section 7, we show that the same dichotomy classification holds in terms of approximation.

▶ **Theorem 7** (Dichotomy for graphs)**.** *There exists a constant factor approximation algorithm for MinHOM(H) if H is a bi-arc graph, otherwise MinHOM(H) is inapproximable.*

In section 8, we give a concrete plan of how to solve the general case. By combining the approach for obtaining the dichotomy in the graph case, together with the idea of getting an approximation algorithm for digraphs admitting a min-ordering, we might be able to achieve a constant factor approximation algorithm for MinHOM(H) when H is DAT-free.

Our constant factors depend on the size of H. However, the implementation of the LP and the ILP would yield a small integrality gap (details in the full version [49]). This indicates perhaps a better analysis of the performance of our algorithm is possible.

▶ **Open Problem 8.** *For which digraphs MinHOM(H) is approximable within a constant factor independent of size of H?*

2 Preliminaries and Definitions

Complexity and approximation of the minimum cost homomorphism problems, and in general the constraint satisfaction problems, are often studied under the existence of *polymorphisms* [6]. A polymorphism of H of arity k is a mapping f from the set of k-tuples over $V(H)$ to $V(H)$ such that if $x_iy_i \in A(H)$ for $i = 1, 2, \ldots, k$, then $f(x_1, x_2, \ldots, x_k)f(y_1, y_2, \ldots, y_k) \in A(H)$. If f is a polymorphism of H we also say that H admits f. A polymorphism f is *idempotent* if it satisfies $f(x, x, \ldots, x) = x$ for all $x \in V(H)$, and is *conservative* if $f(x_1, x_2, \ldots, x_k) \in \{x_1, x_2, \ldots, x_k\}$. A conservative *semi-lattice* polymorphism is a conservative binary polymorphism that is associative, idempotent, commutative. A conservative *majority* polymorphism μ of H is a conservative ternary polymorphism such that $\mu(x, x, y) = \mu(x, y, x) = \mu(y, x, x) = x$ for all $x, y \in V(H)$.

A conservative semi-lattice polymorphism of H naturally defines a binary relation $x \le y$ on the vertices of H by $x \le y$ iff $f(x, y) = x$; by associative, the relation $\le$ is a linear order on $V(H)$, which we call a *min-ordering* of H.

▶ **Definition 9.** *The ordering $v_1 < v_2 < \cdots < v_n$ of $V(H)$ is a*

- min-ordering *iff $uv \in A(H), u'v' \in A(H)$ and $u < u', v' < v$ implies that $uv' \in A(H)$;*
- max-ordering *iff $uv \in A(H), u'v' \in A(H)$ and $u < u', v' < v$ implies that $u'v \in A(H)$;*
- min-max-ordering *iff $uv \in A(H), u'v' \in A(H)$ and $u < u', v' < v$ implies that $uv', u'v \in A(H)$.*

For bipartite graph $H = (B, W)$ let $\overrightarrow{H}$ be the digraph obtained by orienting all the edges of H from B to W. If $\overrightarrow{H}$ admits a min-ordering then we say H admits a min-ordering. It is worth mentioning that, a bipartite graph H admits a conservative majority, iff it admits a min-ordering [30]. Moreover, the complement of H is a circular arc graphs with clique cover two [18].

▶ **Definition 10.** *Let $H = (V, E)$ be a digraph that admits a homomorphism $f : V(H) \to \overrightarrow{C_k}$ (here $\overrightarrow{C_k}$ is the induced directed cycle on $\{0, 1, 2, \ldots, k-1\}$(i.e., arc set $\{(01, 12, 23, ..., (k-2)(k-1), (k-1)0\}$). Let $V_i = f^{-1}(i)$, $0 \le i \le k-1$.*

- *A* k-min-ordering *of H is a linear ordering $<$ of the vertices of H, so that $<$ is a min-ordering on the subgraph induced by any two circularly consecutive V_i, V_{i+1} (subscript addition modulo k).*
- *A* k-min-max-ordering *of H is a linear ordering $<$ of the vertices of H, so that $<$ is a min-max-ordering on the subgraph induced by any two circularly consecutive V_i, V_{i+1} (subscript addition modulo k).*

3 LP for Digraphs with a min-max-ordering

Before presenting the LP, we give a procedure to modify lists associated to the vertices of D. To each vertex $x \in D$, associate a list $L(x)$ that initially contains $V(H)$. Think of $L(x)$ as the set of possible images for x in a homomorphism from D to H. Apply the *arc consistency* procedure as follows. Take an arbitrary arc $xy \in A(D)$ ($yx \in A(D)$) and let $a \in L(x)$. If there is no out-neighbor (in-neighbor) of a in $L(y)$ then remove a from $L(x)$. Repeat this until a list becomes empty or no more changes can be made. Note that if we end up with an empty list after arc consistency then there is no homomorphism of D to H.

Let $a_1, a_2, a_3, \ldots, a_p$ be a min-max-ordering $<$ of the target digraph H. Define $\ell^+(i)$ to be the smallest subscript j such that a_j is an out-neighbor of a_i (and $\ell^-(i)$ to be the smallest subscript j such that a_j is an in-neighbor of a_i).

Consider the following linear program. For every vertex v of D and every vertex a_i of H define variable v_i. Moreover, define variable v_{p+1} for every $v \in D$ whose value is set to zero.

$$
\begin{array}{rlrr}
\min & \sum_{v,i} c(v, a_i)(v_i - v_{i+1}) & & \\
\text{subject to:} & v_i \ge 0 & & \text{(C1)} \\
& v_1 = 1 & & \text{(C2)} \\
& v_{p+1} = 0 & & \text{(C3)} \\
& v_{i+1} \le v_i & & \text{(C4)} \\
& v_{i+1} = v_i & \text{if } \ a_i \notin L(v) & \text{(C5)} \\
& u_i \le v_{l^+(i)} & \forall uv \in A(D) & \text{(C6)} \\
& v_i \le u_{l^-(i)} & \forall uv \in A(D) & \text{(C7)}
\end{array}
$$

Let $\mathcal{S}$ denote the set of constraints of the above LP, then:

▶ **Theorem 11.** *If digraph H admits a min-max-ordering, then there is a one-to-one correspondence between homomorphisms of D to H and integer solutions of $\mathcal{S}$.*

Proof. For homomorphism $f : D \to H$, if $f(v) = a_t$ we set $v_i = 1$ for all $i \le t$, otherwise we set $v_i = 0$. We set $v_1 = 1$ and $v_{p+1} = 0$ for all $v \in V(D)$. Now all the variables are nonnegative and we have $v_{i+1} \le v_i$. Note that if $a_i \notin L(v)$ then $f(v) \ne a_i$ and we have $v_i - v_{i+1} = 0$. It remains to show that $u_i \le v_{l^+(i)}$ for every uv arc in D. Suppose for contradiction that $u_i = 1$ and $v_{l^+(i)} = 0$ and let $f(u) = a_r$ and $f(v) = a_s$. This implies that $u_r = 1$, whence $i \le r$; and $v_s = 1$, whence $s < l^+(i)$. Since $a_i a_{l^+(i)}$ and $a_r a_s$ both are arcs of H with $i \le r$ and $s < l^+(i)$, the fact that H has a min-ordering implies that $a_i a_s$ must also be an arc of H, contradicting the definition of $l^+(i)$. The proof for $v_i \le u_{l^-(i)}$ is analogous.

Conversely, if there is an integer solution for $\mathcal{S}$, we define a homomorphism f as follows: we let $f(v) = a_i$ when i is the largest subscript with $v_i = 1$. We prove that this is indeed a homomorphism by showing that every arc of D is mapped to an arc of H. Let uv be an arc of D and assume $f(u) = a_r$, $f(v) = a_s$. We show that $a_r a_s$ is an arc in H. Observe that $1 = u_r \leq v_{l^+(r)} \leq 1$ and $1 = v_s \leq u_{l^-(s)} \leq 1$, therefore we must have $v_{l^+(r)} = u_{l^-(s)} = 1$. Since r and s are the largest subscripts such that $u_r = v_s = 1$ then $l^+(r) \leq s$ and $l^-(s) \leq r$. Since $a_r a_{l^+(r)}$ and $a_{l^-(s)} a_s$ are arcs of H, we must have the arc $a_r a_s$, as H admits a max-ordering. Furthermore, $f(v) = a_i$ iff $v_i = 1$ and $v_{i+1} = 0$, so, $c(v, a_i)$ contributes to the sum iff $f(v) = a_i$. ◀

We have translated the minimum cost homomorphism problem to a linear program. In fact, this linear program corresponds to a minimum cut problem in an auxiliary network, and can be solved by network flow algorithms [23, 48]. In [30], a similar result to Theorem 11 was proved for the MinHOM(H) problem on undirected graphs when target graph H is bipartite and admits a min-max-ordering. We shall enhance the above system $\mathcal{S}$ to obtain an approximation algorithm for the case where H is only assumed to admit a min-ordering.

4 LP for Digraphs with a min-ordering

In the rest of the paper assume lists are not empty. Moreover, non-empty lists guarantee a homomorphism when H admits a min-ordering.

▶ **Lemma 12.** *[32] Let H be a digraph that admits a min-ordering. If all the lists are non-empty after arc consistency, then there exists a homomorphism from D to H.*

Suppose $a_1, a_2, \ldots, a_p$ is a min-ordering of H. Let E' denote the set of all the pairs (a_i, a_j) such that $a_i a_j$ is not an arc of H, but there is an arc $a_i a_{j'}$ of H with $j' < j$ and an arc $a_{i'} a_j$ of H with $i' < i$. Let $E = A(H)$ and define H' to be the digraph with vertex set $V(H)$ and arc set $E \cup E'$. Note that E and E' are disjoint sets.

▶ **Observation 13.** *The ordering $a_1, a_2, \cdots, a_p$ is a min-max-ordering of H'.*

▶ **Observation 14.** *Let $e = a_i a_j \in E'$. Then a_i does not have any out-neighbor in H after a_j, or a_j does not have any in-neighbor in H after a_i.*

Observation 14 easily follows from the fact that H has a min-ordering. Since H' has a min-max-ordering, we can form system of linear inequalities $\mathcal{S}$, for H' as described in Section 3. Homomorphisms of D to H' are in a one-to-one correspondence with integer solutions of $\mathcal{S}$, by Theorem 11. However, we are interested in homomorphisms of D to H, not H'. Therefore, we shall add further inequalities to $\mathcal{S}$ to ensure that we only admit homomorphisms from D to H, i.e., avoid mapping arcs of D to the arcs in E'.

For every arc $e = a_i a_j \in E'$ and every arc $uv \in A(D)$, by Observation 14, two of the following set of inequalities will be added to $\mathcal{S}$ (i.e. either (C8), (C11) or (C9), (C10) or (C9), (C11)).

$$v_j \leq u_s + \sum_{t<i, a_t a_j \in E, a_t \in L(u)} (u_t - u_{t+1}) \quad \text{if } a_s \in L(u) \text{ is the first in-neighbour of } a_j \text{ after } a_i \tag{C8}$$

$$v_j \leq v_{j+1} + \sum_{t<i, a_t a_j \in E, a_t \in L(u)} (u_t - u_{t+1}) \quad \text{if } a_j \text{ has no in-neighbour after } a_i \tag{C9}$$

$$u_i \leq v_s + \sum_{t<j, a_i a_t \in E, a_t \in L(v)} (v_t - v_{t+1}) \quad \text{if } a_s \in L(v) \text{ is the first out-neighbour of } a_i \text{ after } a_j \tag{C10}$$

$$u_i \leq u_{i+1} + \sum_{t<j, a_i a_t \in E, a_t \in L(v)} (v_t - v_{t+1}) \quad \text{if } a_i \text{ has no out-neighbour after } a_j \tag{C11}$$

Additionally, for every pair $(x, y) \in V(D) \times V(D)$ consider a list $L(x, y)$ of possible pairs (a, b), $a \in L(x)$ and $b \in L(y)$. Perform *pair consistency* procedure as follows. Consider three vertices $x, y, z \in V(D)$. For $(a, b) \in L(x, y)$ if there is no $c \in L(z)$ such that $(a, c) \in L(x, z)$

and $(c,b) \in L(z,y)$ then remove (a,b) from $L(x,y)$. Repeat this until a pair list becomes empty or no more changes can be made. Here, we assume that after pair consistency procedure no pair list is empty, as otherwise there is no homomorphism of D to H. Therefore, by pair consistency, add the following constraints for every u, v in $V(D)$ and $a_i \in L(u)$:

$$u_i - u_{i+1} \leq \sum_{\substack{j: \\ (a_i,a_j)\in L(u,v)}} (v_j - v_{j+1}) \quad \text{(C12)}$$

▶ **Lemma 15.** *If H admits a min-ordering, then there is a one-to-one correspondence between homomorphisms of D to H and integer solutions of the extended system $\mathcal{S}$.*

5 Approximation for Digraphs with a min-ordering

In what follows, we describe an overview of our approximation algorithm for MinHOM(H) where the fixed digraph H has a min-ordering. We encourage the reader to see Algorithm 1 while reading this section. An overview of the proofs of the correctness and approximation bound are postponed for the later subsections (further details in the full version [49]).

Let D be the input digraph together with a cost function c. Let $a_1, \ldots, a_p$ be a min ordering of the vertices of H. The algorithm, first constructs digraph H' from H as in Section 4. By Observation 13, $a_1, \ldots, a_p$ is a min-max-ordering for H'. By Lemma 15, the integral solutions of the extended LP are in one-to-one correspondence to homomorphisms from D to H. At this point, our algorithm will minimize the cost function over extended $\mathcal{S}$ in polynomial time using a linear programming algorithm. This will generally result in a fractional solution (Even though the original system $\mathcal{S}$ is known to be totally unimodular [23, 48] and hence have integral optima, we have added inequalities, and hence lost this advantage). We will obtain an integer solution by a randomized procedure called *rounding*. Choose, uniformly at random, a random variable $X \in [0,1]$, and define the rounded values $u'_i = 1$ when $u_i \geq X$ (u_i is the returned value by the LP), and $u'_i = 0$ otherwise. It is easy to check that the rounded values satisfy the original inequalities, i.e., correspond to a homomorphism f of D to H'.

Now the algorithm will once more modify the solution f to become a homomorphism from D to H, i.e., to avoid mapping the arcs of D to the arcs in E'. This will be accomplished by another randomized procedure, which we call Shift. We choose, uniformly at random, another random variable $Y \in [0,1]$, which will guide the shifting. Let F denote the set of all arcs in E' to which some arcs of D are mapped by f. If F is empty, we need no shifting. Otherwise, let $a_i a_j$ be an arc of F. Since $F \subseteq E'$, Observation 14 implies that either a_j has no in-neighbor after a_i or a_i has no out-neighbor after a_j. Suppose the first case happens (the shifting process is similar in the other case).

Consider a vertex v in D such that $f(v) = a_j$ (i.e. $v'_j = 1$ and $v'_{j+1} = 0$) and v has an in-neighbor u in D with $f(u) = a_i$ (i.e. $u'_i = 1$ and $u'_{i+1} = 0$). For such a vertex v, let $S_v = \{a_{t_1}, a_{t_2}, \ldots, a_{t_k}\}$ be the set of all vertices a_t with $t < j$ such that $a_i a_t \in E$ and $a_t \in L(v)$. Suppose S_v consists of a_t with subscripts t ordered as $t_1 < t_2 < \cdots < t_k$.

▶ **Lemma 16.** *During procedure* Shift*, the set of indices $t_1 < \cdots < t_k$ considered in Line 6 of the Algorithm 1 is non-empty.*

By Lemma 16, S_v is not empty. The algorithm now selects one vertex from this set as follows. Let $P_{v,t} = \frac{v_t - v_{t+1}}{P_v}$, where $P_v = \sum_{\substack{t<j \\ a_i a_t \in E, a_t \in L(v)}} (v_t - v_{t+1})$.

Note that $P_v > 0$ because of constraints (C9) and (C10). Then a_{t_q} is selected if $\sum_{p=1}^{q} P_{v,t_p} < Y \leq \sum_{p=1}^{q+1} P_{v,t_p}$. Thus a concrete a_t is selected with probability $P_{v,t}$, which is proportional to the difference of the fractional values $v_t - v_{t+1}$. When the selected vertex is a_t, we shift the image of the vertex v from a_j to a_t, and set $v'_r = 1$ if $r \leq t$, else set $v'_r = 0$. Note that a_t is before a_j in the min-ordering. Now we might need to shift images of the neighbors of v. In this case, repeat the shifting procedure for neighbors of v. This processes continues in a Breadth-first search (BFS) like manner, until no more shift is required (Figure 1 gives an illustration). Note that a vertex might be visited multiple times in procedure shift while a pair $(v, a_i) \in V(D) \times V(H)$ is considered at most one time.

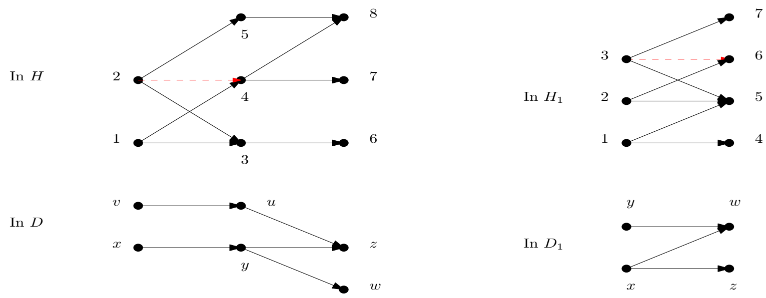

Figure 1 Two examples. In the right example, the target digraph is H_1 and input is D_1. Digraphs D_1 and H_1 both can be view as bipartite graphs and $1, 2, 3, 4, 5, 6, 7$ is a min-ordering of H. When x, y are mapped to 3 and w is mapped to 6 then the algorithm should shift the image of w from 6 to 5 and since 35 is an arc there is no need to shift the image of y. In the left example, the target digraph is H and the input is D. $1, 2, 3, 4, 5, 6, 7, 8$ is a min-ordering of H and 24 is a missing arc. Suppose x is mapped to 2, y to 4, w to 7, z to 8, u to 5 and v to 2. Then we should shift the image of y to 3 and then w to 6 and z to 6 and then u to 3 and v to one of the $1, 2$.

We remark that the images of vertices in D are always shifted towards smaller elements in their lists. Lemma 17 shows that this shifting modifies the homomorphism f, and hence, the corresponding values of the variables. Namely, $v'_{t+1}, \ldots, v'_j$ are reset to 0, keeping all other values the same. Note that these modified values still satisfy the original set of constraints $\mathcal{S}$, i.e., the modified mapping is still a homomorphism.

▶ **Lemma 17.** *Procedure* SHIFT*, in polynomial time, returns a homomorphism of D to H'.*

We repeat the same process for the next v with these properties, until no edge of D is mapped to an edge in E'. Each iteration involves at most $|V(H)| \cdot |V(D)|$ shifts. After at most $|E'|$ iterations, no edge of D is mapped to an edge in F and we no longer need to shift. Next theorem follows from Lemma 16 and 17.

▶ **Theorem 18.** *Our algorithm, in polynomial time, returns a homomorphism of D to H.*

5.1 Analyzing the Approximation Ratio

We now claim that the cost of this homomorphism is at most $|V(H)|^2$ times the minimum cost of a homomorphism. Let w denote the value of the objective function with the fractional optimum u_i, v_j, and w' denote the value of the objective function with the final values u'_i, v'_j, after the rounding and all the shifting. Also, let w^* be the minimum cost of a homomorphism of D to H. Obviously, $w \leq w^* \leq w'$.

We now show that the expected value of w' is at most a constant times w. Let us focus on the contribution of one summand, say $v'_t - v'_{t+1}$, to the calculation of the cost. In any integer solution, $v'_t - v'_{t+1}$ is either 0 or 1. The probability that $v'_t - v'_{t+1}$ contributes to w' is the probability of the event that $v'_t = 1$ and $v'_{t+1} = 0$. This can happen in the following:

Algorithm 1 Approximation MinHOM(H).

procedure APPROX–MINHOM(D, H)
  Construct H' from H (as in Section 3)
  Let u_is be the (fractional) values returned by the extended LP
  Choose a random variable $X \in [0, 1]$, and $\forall u_i$s : if $X \leq u_i$ let $u'_i = 1$, else let $u'_i = 0$
  Let $f(u) = a_i$ where i is the largest subscript with $u'_i = 1$ ▷ f is a homomorphism from D to H'
  Choose a random variable $Y \in [0, 1]$
  while $\exists uv \in A(D)$ such that $f(u)f(v) \in A(H') \setminus A(H)$ **do**
    if $f(v)$ does not have an in-neighbor after $f(u)$ **then** SHIFT(f, v)
    else if $f(u)$ does not have an out-neighbor after $f(v)$ **then** SHIFT(f, u)
  return f ▷ f is a homomorphism from D to H

Algorithm 2 The Shifting Procedure.

procedure SHIFT(f, x)
  Let Q be a Queue, $Q.enqueue(x)$
  while Q is not empty **do**
    $v \leftarrow Q.dequeue()$
    for $uv \in A(D)$ with $f(u)f(v) \notin A(H)$ **or** $vu \in A(D)$ with $f(v)f(u) \notin A(H)$ **do**
      ▷ Here we assume the first condition holds, the other case is similar
      ▷ Further, we assume $f(v)$ does not have an in-neighbor after $f(u)$
      Let $t_1 < \cdots < t_k$ be indices so that $a_{t_j} < f(v), a_{t_j} \in L(v), f(u)a_{t_j} \in A(H)$
      Let $P_v \leftarrow \sum_{j=1}^{j=k} (v_{t_j} - v_{t_j+1})$ and $P_{v,t} \leftarrow (v_t - v_{t+1}) \ / \ P_v$
      if $\sum_{p=1}^{q} P_{v,t_p} < Y \leq \sum_{p=1}^{q+1} P_{v,t_p}$ **then**
        $f(v) \leftarrow a_{t_q}$, set $v'_i = 1$ for $1 \leq i \leq t_q$, and set $v'_i = 0$ for $t_p < i$
      for $vz \in A(D)$ $(zv \in A(D))$ with $f(v)f(z) \notin A(H)$ $(f(z)f(v) \notin A(H))$ **do**
        $Q.enqueue(z)$
  return f ▷ f is a homomorphism from D to H'

1. v is mapped to a_t by rounding, and is not shifted away. In other words, we have $v'_t = 1$ and $v'_{t+1} = 0$ after rounding, and these values don't change by procedure SHIFT.

2. v is first mapped to some $a_j, j > t$, by rounding, and then re-mapped to a_t by procedure SHIFT.

▶ **Lemma 19.** *The expected contribution of one summand, say $v'_t - v'_{t+1}$, to the expected cost of w' is at most $|V(H)|^2 c(v, a_t)(v_t - v_{t+1})$.*

▶ **Theorem 20.** *Algorithm 1 returns a homomorphism with expected cost $|V(H)|^2 \cdot OPT$. The algorithm can be de-randomized to obtain a deterministic $|V(H)|^2$-approximation algorithm.*

6 Approximation for Digraphs with a k-min-ordering

Digraphs admitting k-min-ordering ($k > 1$) do not admit a min-ordering or a conservative majority polymorphism. However, this does not rule out the possibility of a constant factor approximation algorithm. We show that they are in fact DAT-free digraphs, and they admit a nice geometric representation (see the full version [49]).

▶ **Lemma 21.** *Let H be a digraph that admits a k-min-ordering. Then H is DAT-free, and LHOM(H) is polynomial time solvable.*

Let H be a digraph with a k-min-ordering ($k > 1$) and partition $V_0, V_1, \ldots, V_{k-1}$ of its vertices, and let $<$ be a k-min-ordering of $V(H)$. It is easy to argue that the input digraph D must be homomorphic to $\overrightarrow{C_k}$ otherwise there is no homomorphism from D to H. Therefore, we assume (for some $0 \le \ell \le k-1$), $L(u) \subseteq V_i$ for every $u \in U_{i+\ell}$, $0 \le i \le k-1$. Now the LP is designed according to the lists L. Since $<$ is a min-ordering of $V_i \cup V_{i+1}$, the constraints are very similar to the ones in Section 3. The conclusion of this section is the following:

▶ **Theorem 22.** *There is a (deterministic) $|V(H)|^2$-approximation algorithm for MinHOM(H) when the target digraph H admits a k-min-ordering, $k > 1$.*

7 A Dichotomy for Graphs

Feder and Vardi [20] proved that if a graph H admits a conservative majority polymorphism, then LHOM(H) is polynomial time solvable. Later, Feder *et al.,* [18] showed that LHOM(H) is polynomial time solvable iff H is a *bi-arc* graph. Hence, by Observation 4, the problem is inapproximable beyond bi-arc graphs. A bi-arc graph is represented by a pair of families of arcs on a circle with specific conditions (exact definition is given in the full version [49]). Note that in a bi-arc graph a vertex may have a self-loop.

▶ **Theorem 23** ([7, 18]). *A graph admits a conservative majority polymorphism iff it is a bi-arc graph.*

▶ **Definition 24** (G^*). *Let $G = (V, E)$ be a graph. Let G^* be a bipartite graph with partite sets V, V' where V' is a copy of V. Two vertices $u \in V$, and $v' \in V'$ of G^* are adjacent in G^* iff uv is an edge of G.*

A *circular arc* graph is a graph that is the intersection graph of a family of arcs on a circle. A bipartite graph whose complement is a circular arc graph, is called a *co-circular arc* graph. Note that co-circular arc graphs are irreflexive, meaning no vertex has a loop.

▶ **Lemma 25.** *Let H^* be the bipartite graph constructed from a bi-arc graph H. Then H^* is a co-circular arc graph and H^* admits a min-ordering.*

Let H be a bi-arc graph, with vertex set I, and let $H^* = (I, I')$ be the bipartite graph constructed from H. Let $a_1, a_2, \ldots, a_p$ be an ordering of the vertices in I and $b_1, b_2, \ldots, b_p$ be an ordering of the vertices of I'. Note that each a_i has a copy $b_{\pi(i)}$ in $\{b_1, b_2, \ldots, b_n\}$ where π is a permutation on $\{1, 2, 3, \ldots, p\}$. By Lemma 25, let us assume $a_1, a_2, \ldots, a_p, b_1, b_2, \ldots, b_p$ is a min-ordering for H^*.

Let G be the input graph with vertex set V and a cost function c. Construct G^* from G with vertex set $V \cup V'$ as in Definition 24. Now construct an instance of the MinHOM(H^*) for the input graph G^* and set $c(v', b_{\pi(i)}) = c(v, a_i)$ for $v \in V$ and $v' \in V'$. Further, make H^* a digraph by orienting all its edges from I to I', and similarly make G^* a digraph by orienting all its edges from V to V'. Note that, by construction of H^* and G^*, there exists a homomorphism $f : G \to H$ with cost $\mathfrak{C}$ iff there exists homomorphism $f^* : G^* \to H^*$ with cost $2\mathfrak{C}$ such that, if $f^*(v) = a_i$ then $f^*(v') = b_j$ with $j = \pi(i)$.

We first perform the arc consistency and pair consistency procedures for the vertices in G^*. Note that if $L(u)$ contains element a_i then $L(u')$ contains $b_{\pi(i)}$ and when $L(u')$ contains some b_j then $L(u)$ contains $a_{\pi^{-1}(j)}$. Next, we define the system of linear equations $\widehat{S^*}$ with the same construction as in Sections 3, 4. Further, we add the following constraint to $\widehat{S^*}$. The full set of constraints in $\widehat{S^*}$ is presented in the extended version.

$$u_i - u_{i+1} = u'_{\pi(i)} - u'_{\pi(i)+1} \qquad \forall u, u' \in G^*, \forall a_i, b_{\pi(i)} \in H^*$$

▶ **Lemma 26.** *If H is a bi-arc graph, then there is a one-to-one correspondence between homomorphisms from G to H and integer solutions of $\widehat{S^*}$.*

Once again we round an optimal fractional solution of $\widehat{S^*}$, using random variable $X \in [0, 1]$. Let $\mathcal{F}$ be a mapping form $V(G^*)$ to $V(H^*)$ obtained after rounding using X. We give an algorithm that modifies $\mathcal{F}$ and achieves a homomorphism $f : G \to H$ (i.e. an integral solution that satisfies $\widehat{S^*}$). The algorithm deploys a shifting procedure that first uses a random variable Y to shift the images of some of the vertices of $V(G^*)$ to obtain a homomorphism f from G^* to H^*. Second, it applies a breadth-first search function to make f consistent on V and V'; meaning that $f(u) = a_i$, $u \in V$ iff $f(u') = b_{\pi(i)}$, $u' \in V'$. The proof of the following theorems and de-randomization of the algorithm appear in the full version [49].

▶ **Theorem 27.** *There exists a randomized algorithm that modifies $\mathcal{F}$ and obtain a homomorphism $f : G \to H$. Moreover, the expected cost of the homomorphism returned by this algorithm is at most $2|V(H)| \cdot OPT$.*

▶ **Theorem 28.** *If H admits a conservative majority polymorphism, then MinHOM(H) has a (deterministic) $2|V(H)|$-approximation algorithm, otherwise it is inapproximable.*

8 Beyond majority and min-ordering (DAT-free cases)

This section offers a view of moving forward to get a dichotomy classification for constant approximability of MinHOM(H). We believe the class of DAT-free digraphs is the right boundary between the approximable cases, and the ones that do not admit any approximation.

▶ **Conjecture 29.** *MinHOM(H) admits a constant approximation polynomial time algorithm when H is a DAT-free digraph, otherwise, MinHOM(H) is not approximable.*

For digraph $D = (V, A)$, let D^* be a bipartite digraph with partite sets V, V' where V' is a copy of V. There is an arc in D^* from $u \in V$ to $v' \in V'$ iff uv is an arc of D. In what follows, we give a road map for solving the conjecture. Let us start off by making a connection between homomorphisms from D to a DAT-free target digraph H, and the homomorphisms from D^* to H^*.

▶ **Proposition 30.** *Let D, H be two digraphs and let D, H, L (here L are the lists) be an instance of the LHOM(H). Suppose H is DAT-free. Then H^* admits a min-ordering, and LHOM(H^*) is polynomial time solvable for instance D^*, H^* where $L^*(v') = \{a' | a \in L(v)\}$ and $L^*(v) = L(v)$ for every $v \in V(D)$.*

Similar to Lemmas 15 and 26, we can obtain set of constraints $\widehat{S^*}$ such that there is a one-to-one correspondence between homomorphisms from D to H and integer solutions of $\widehat{S^*}$ (details in the full version [49]). Our primary challenge would be finding a rounding procedure to obtain a homomorphism from D to H. We believe there is a need to deploy the shift procedure in min-ordering case (Section 3), as well as, the shifting procedure in the majority case (Section 7). This essentially means obtaining a new way of solving a list homomorphism from D to H when H is a DAT-free, using the bi-partition method. The calculation should work out; yielding a constant bound between the fractional value of the LP and the integral value obtained by rounding. Notice that in the majority case the symmetry of the arcs is heavily used in our argument, whereas in the digraph case we no longer have this property in hand.

References

1 Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The Hardness of Approximate Optima in Lattices, Codes, and Systems of Linear Equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997. doi:10.1006/jcss.1997.1472.

2 Per Austrin. Towards Sharp Inapproximability For Any 2-CSP. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 307–317, 2007. doi:10.1109/FOCS.2007.73.

3 Guillaume Bagan, Arnaud Durand, Emmanuel Filiot, and Olivier Gauwin. Efficient Enumeration for Conjunctive Queries over X-underbar Structures. In *Computer Science Logic, 24th International Workshop, CSL 2010, 19th Annual Conference of the EACSL, Brno, Czech Republic, August 23-27, 2010. Proceedings*, pages 80–94, 2010. doi:10.1007/978-3-642-15205-4_10.

4 Amotz Bar-Noy, Mihir Bellare, Magnús M. Halldórsson, Hadas Shachnai, and Tami Tamir. On Chromatic Sums and Distributed Resource Allocation. *Inf. Comput.*, 140(2):183–202, 1998. doi:10.1006/inco.1997.2677.

5 Libor Barto. The Dichotomy for Conservative Constraint Satisfaction Problems Revisited. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21-24, 2011, Toronto, Ontario, Canada*, pages 301–310, 2011. doi:10.1109/LICS.2011.25.

6 Libor Barto, Andrei A. Krokhin, and Ross Willard. Polymorphisms, and How to Use Them. In Andrei A. Krokhin and Stanislav Zivny, editors, *The Constraint Satisfaction Problem: Complexity and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 1–44. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/DFU.Vol7.15301.1.

7 Richard C. Brewster, Tomás Feder, Pavol Hell, Jing Huang, and Gary MacGillivray. Near-Unanimity Functions and Varieties of Reflexive Graphs. *SIAM J. Discrete Math.*, 22(3):938–960, 2008. doi:10.1137/S0895480103436748.

8 Andrei A. Bulatov. Tractable conservative Constraint Satisfaction Problems. In *18th IEEE Symposium on Logic in Computer Science (LICS 2003), 22-25 June 2003, Ottawa, Canada, Proceedings*, page 321, 2003. doi:10.1109/LICS.2003.1210072.

9 Andrei A. Bulatov. H-Coloring dichotomy revisited. *Theor. Comput. Sci.*, 349(1):31–39, 2005. doi:10.1016/j.tcs.2005.09.028.

10 Andrei A. Bulatov. Complexity of conservative constraint satisfaction problems. *ACM Trans. Comput. Log.*, 12(4):24:1–24:66, 2011. doi:10.1145/1970398.1970400.

11 Andrei A. Bulatov. Conservative constraint satisfaction re-revisited. *J. Comput. Syst. Sci.*, 82(2):347–356, 2016. doi:10.1016/j.jcss.2015.07.004.

12 Andrei A. Bulatov. A Dichotomy Theorem for Nonuniform CSPs. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 319–330, 2017. doi:10.1109/FOCS.2017.37.

13 David A. Cohen, Martin C. Cooper, Peter Jeavons, and Andrei A. Krokhin. The complexity of soft constraint satisfaction. *Artif. Intell.*, 170(11):983–1016, 2006. doi:10.1016/j.artint.2006.04.002.

14 David A. Cohen, Martin C. Cooper, Peter G. Jeavons, Andrei A. Krokhin, Robert Powell, and Stanislav Zivny. Binarisation for Valued Constraint Satisfaction Problems. *SIAM J. Discrete Math.*, 31(4):2279–2300, 2017. doi:10.1137/16M1088107.

15 Nadia Creignou, Sanjeev Khanna, and Madhu Sudan. *Complexity classifications of boolean constraint satisfaction problems*, volume 7. SIAM, 2001.

16 Víctor Dalmau, Andrei A. Krokhin, and Rajsekar Manokaran. Towards a characterization of constant-factor approximable finite-valued CSPs. *J. Comput. Syst. Sci.*, 97:14–27, 2018. doi:10.1016/j.jcss.2018.03.003.

17 Alina Ene, Jan Vondrák, and Yi Wu. Local Distribution and the Symmetry Gap: Approximability of Multiway Partitioning Problems. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 306–325, 2013. doi:10.1137/1.9781611973105.23.

18 Tomás Feder, Pavol Hell, and Jing Huang. Bi-arc graphs and the complexity of list homomorphisms. *Journal of Graph Theory*, 42(1):61–80, 2003. doi:10.1002/jgt.10073.

19 Tomás Feder, Pavol Hell, Peter Jonsson, Andrei A. Krokhin, and Gustav Nordh. Retractions to Pseudoforests. *SIAM J. Discrete Math.*, 24(1):101–112, 2010. doi:10.1137/080738866.

20 Tomás Feder and Moshe Y. Vardi. The Computational Structure of Monotone Monadic SNP and Constraint Satisfaction: A Study through Datalog and Group Theory. *SIAM J. Comput.*, 28(1):57–104, 1998. doi:10.1137/S0097539794266766.

21 Krzysztof Giaro, Robert Janczewski, Marek Kubale, and Michal Malafiejski. A 27/26-Approximation Algorithm for the Chromatic Sum Coloring of Bipartite Graphs. In *Approximation Algorithms for Combinatorial Optimization, 5th International Workshop, APPROX 2002, Rome, Italy, September 17-21, 2002, Proceedings*, pages 135–145, 2002. doi:10.1007/3-540-45753-4_13.

22 Michel X. Goemans and David P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. ACM*, 42(6):1115–1145, 1995. doi:10.1145/227683.227684.

23 Gregory Z. Gutin, Pavol Hell, Arash Rafiey, and Anders Yeo. A dichotomy for minimum cost graph homomorphisms. *Eur. J. Comb.*, 29(4):900–911, 2008. doi:10.1016/j.ejc.2007.11.012.

24 Gregory Z. Gutin, Arash Rafiey, and Anders Yeo. Minimum Cost Homomorphisms to Semicomplete Bipartite Digraphs. *SIAM J. Discrete Math.*, 22(4):1624–1639, 2008. doi:10.1137/060668316.

25 Gregory Z. Gutin, Arash Rafiey, Anders Yeo, and Michael Tso. Level of repair analysis and minimum cost homomorphisms of graphs. *Discrete Applied Mathematics*, 154(6):881–889, 2006. doi:10.1016/j.dam.2005.06.012.

26 Wolfgang Gutjahr, Emo Welzl, and Gerhard J. Woeginger. Polynomial graph-colorings. *Discrete Applied Mathematics*, 35(1):29–45, 1992. doi:10.1016/0166-218X(92)90294-K.

27 Magnús M. Halldórsson, Guy Kortsarz, and Hadas Shachnai. Minimizing Average Completion of Dedicated Tasks and Interval Graphs. In *Approximation, Randomization and Combinatorial Optimization: Algorithms and Techniques, 4th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2001 and 5th International Workshop on Randomization and Approximation Techniques in Computer Science, RANDOM 2001 Berkeley, CA, USA, August 18-20, 2001, Proceedings*, pages 114–126, 2001. doi:10.1007/3-540-44666-4_15.

28 Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. doi:10.1145/502090.502098.

29 Pavol Hell, Jing Huang, Ross M. McConnell, and Arash Rafiey. Interval-Like Graphs and Digraphs. In *43rd International Symposium on Mathematical Foundations of Computer Science, MFCS 2018, August 27-31, 2018, Liverpool, UK*, pages 68:1–68:13, 2018. doi:10.4230/LIPIcs.MFCS.2018.68.

30 Pavol Hell, Monaldo Mastrolilli, Mayssam Mohammadi Nevisi, and Arash Rafiey. Approximation of Minimum Cost Homomorphisms. In *Algorithms - ESA 2012 - 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings*, pages 587–598, 2012. `doi:10.1007/978-3-642-33090-2_51`.

31 Pavol Hell and Jaroslav Nesetril. On the complexity of H-coloring. *J. Comb. Theory, Ser. B*, 48(1):92–110, 1990. `doi:10.1016/0095-8956(90)90132-J`.

32 Pavol Hell and Jaroslav Nesetril. *Graphs and homomorphisms*. Oxford University Press, 2004.

33 Pavol Hell and Arash Rafiey. The Dichotomy of List Homomorphisms for Digraphs. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1703–1713, 2011. `doi:10.1137/1.9781611973082.131`.

34 Pavol Hell and Arash Rafiey. Monotone Proper Interval Digraphs and Min-Max Orderings. *SIAM J. Discrete Math.*, 26(4):1576–1596, 2012. `doi:10.1137/100783844`.

35 Pavol Hell and Arash Rafiey. The Dichotomy of Minimum Cost Homomorphism Problems for Digraphs. *SIAM J. Discrete Math.*, 26(4):1597–1608, 2012. `doi:10.1137/100783856`.

36 Pavol Hell and Arash Rafiey. Bi-Arc Digraphs and Conservative Polymorphisms. *CoRR*, abs/1608.03368, 2016. `arXiv:1608.03368`.

37 Klaus Jansen. Approximation Results for the Optimum Cost Chromatic Partition Problem. *J. Algorithms*, 34(1):54–89, 2000. `doi:10.1006/jagm.1999.1022`.

38 Tao Jiang and Douglas B West. Coloring of trees with minimum sum of colors. *Journal of Graph Theory*, 32(4):354–358, 1999.

39 Peter Jonsson and Gustav Nordh. Introduction to the Maximum SolutionProblem. In *Complexity of Constraints - An Overview of Current Research Themes [Result of a Dagstuhl Seminar].*, pages 255–282, 2008. `doi:10.1007/978-3-540-92800-3_10`.

40 Sanjeev Khanna, Madhu Sudan, Luca Trevisan, and David P. Williamson. The Approximability of Constraint Satisfaction Problems. *SIAM J. Comput.*, 30(6):1863–1920, 2000. `doi:10.1137/S0097539799349948`.

41 Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal Inapproximability Results for Max-Cut and Other 2-Variable CSPs? In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 146–154, 2004. `doi:10.1109/FOCS.2004.49`.

42 Vladimir Kolmogorov, Andrei A. Krokhin, and Michal Rolinek. The Complexity of General-Valued CSPs. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1246–1258, 2015. `doi:10.1109/FOCS.2015.80`.

43 Leo G. Kroon, Arunabha Sen, Haiyong Deng, and Asim Roy. The Optimal Cost Chromatic Partition Problem for Trees and Interval Graphs. In *Graph-Theoretic Concepts in Computer Science, 22nd International Workshop, WG '96, Cadenabbia (Como), Italy, June 12-14, 1996, Proceedings*, pages 279–292, 1996. `doi:10.1007/3-540-62559-3_23`.

44 Ewa Kubicka and Allen J. Schwenk. An Introduction to Chromatic Sums. In *Computer Trends in the 1990s - Proceedings of the 1989 ACM 17th Annual Computer Science Conference, Louisville, Kentucky, USA, February 21-23, 1989*, pages 39–45, 1989. `doi:10.1145/75427.75430`.

45 Michael Lewin, Dror Livnat, and Uri Zwick. Improved Rounding Techniques for the MAX 2-SAT and MAX DI-CUT Problems. In *Integer Programming and Combinatorial Optimization, 9th International IPCO Conference, Cambridge, MA, USA, May 27-29, 2002, Proceedings*, pages 67–82, 2002. `doi:10.1007/3-540-47867-1_6`.

46 Gary MacGillivray and Jacobus Swarts. The $\mathrm{C_k}$-extended graft construction. *Discrete Applied Mathematics*, 159(12):1293–1301, 2011. `doi:10.1016/j.dam.2011.04.006`.

47 Konstantin Makarychev and Yury Makarychev. Approximation Algorithms for CSPs. In Andrei A. Krokhin and Stanislav Zivny, editors, *The Constraint Satisfaction Problem: Complexity* *and Approximability*, volume 7 of *Dagstuhl Follow-Ups*, pages 287–325. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/DFU.Vol7.15301.11.

48 Monaldo Mastrolilli and Arash Rafiey. On the approximation of minimum cost homomorphism to bipartite graphs. *Discrete Applied Mathematics*, 161(4-5):670–676, 2013. doi:10.1016/j.dam.2011.05.002.

49 Akbar Rafiey, Arash Rafiey, and Thiago Santos. Toward a Dichotomy for Approximation of H-coloring. *CoRR*, abs/1902.02201, 2019. arXiv:1902.02201.

50 Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 245–254, 2008. doi:10.1145/1374376.1374414.

51 Rustem Takhanov. A Dichotomy Theorem for the General Minimum Cost Homomorphism Problem. In *27th International Symposium on Theoretical Aspects of Computer Science, STACS 2010, March 4-6, 2010, Nancy, France*, pages 657–668, 2010. doi:10.4230/LIPIcs.STACS.2010.2493.

52 Johan Thapper and Stanislav Zivny. The complexity of finite-valued CSPs. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 695–704, 2013. doi:10.1145/2488608.2488697.

53 Hannes Uppman. The Complexity of Three-Element Min-Sol and Conservative Min-Cost-Hom. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, pages 804–815, 2013. doi:10.1007/978-3-642-39206-1_68.

54 Hannes Uppman. Computational Complexity of the Extended Minimum Cost Homomorphism Problem on Three-Element Domains. In *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014), STACS 2014, March 5-8, 2014, Lyon, France*, pages 651–662, 2014. doi:10.4230/LIPIcs.STACS.2014.651.

55 Dmitriy Zhuk. A Proof of CSP Dichotomy Conjecture. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 331–342, 2017. doi:10.1109/FOCS.2017.38.