DAT-free Digraphs and Approximation of H-coloring

Akbar Rafiey [*] Arash Rafiey [†] Thiago Santos [‡]

Abstract

We study the approximability of minimum cost homomorphism (MinHOM) problem within a constant factor. Given two (di)graphs G, H and a cost function $c : (V(G), V(H)) \to \mathbb{R} \cup \{+\infty\}$, in the MinHOM($H$) problem we are interested in finding a homomorphism (a.k.a *H-coloring*) $f : V(G) \to V(H)$ that minimizes $\sum_{v \in V(G)} c(v, f(v))$. This is indeed a valued constraint satisfaction problem (VCSP(Γ)) where language Γ contains *unary* and *crisp* cost functions. There are only a few results concerning the approximability of VCSP(Γ) for languages Γ containing cost functions that can take infinite values.

The only result concerning approximability of MinHOM(H) is due to Hell *et al.,* [11]. They prove that, for graphs without loops, if bipartite graph H admits a *min-ordering* then MinHOM(H) is approximable within $|V(H)|$ factor; otherwise it is known to be not approximable. Moreover, for graphs with loops on all vertices, if H is an interval graph, then MinHOM(H) admits a polynomial time constant ratio approximation algorithm, and otherwise it is not approximable.

In terms of digraphs, MinHOM(H) is not approximable if H contains a *digraph astroidal triple (DAT)*.

On the other hand, if H is DAT-free then we devise a constant approximation algorithm for MinHOM(H). Therefore, we obtain a dichotomy classification for approximability of minimum cost homomorphism to digraphs.

Our constant factor depends on the size of H, but in practice we have obtained much better bounds. This shows perhaps a better rounding or better estimation is possible. This leaves open problems such as the dichotomy classification of digraphs H that admit a c- approximation algorithm for MinHOM(H) when c is independent from size of H.

1 Introduction

We study the approximability of the minimum cost homomorphism problem, introduced below. A *c-approximation* algorithm produces a solution of cost at most c times the minimum cost. A constant ratio approximation algorithm is a c-approximation algorithm for some constant c. When we say a problem has a c-approximation algorithm, we mean a polynomial time algorithm. We say that a problem is not approximable if it there is no polynomial time approximation algorithm with a multiplicative guarantee unless $P = NP$.

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 any arc xy of D the pair $f(x)f(y)$ is an arc of H. The *minimum cost homomorphism problem* to H, denoted by MinHOM(H), seeks, for a given input digraph D and vertex-mapping costs $c(x, u), x \in V(D), u \in V(H)$, a homomorphism f of D to H that minimizes the total cost $\sum\limits_{x \in V(D)} c(x, f(x))$. This offers a natural and practical way to model many optimization problems. For instance, in [6] it was used to model a problem of minimizing the cost of a repair and maintenance schedule for large machinery. It generalizes many other problems such as list homomorphism problems, retraction problems [3], and various optimum cost chromatic partition problems [7, 12, 13, 15].

*SFU email : arafiey@sfu.ca

†Indiana State University, IN, USA arash.rafiey@indstate.edu and Simon Fraser University, BC, Canada, arashr@sfu.ca,
supported by NSF 1751765

‡Indiana State email: tsantos2@sycamores.indstate.edu

This problem is an important special case of *Valued Constrained Satisfaction Problems* (VCSPs). The complexity of MinHOM(H) was studied in a series of papers, and complete complexity classifications were given in [5] for undirected graphs, in [10] for digraphs, and in [17] for more general structures. Cohen *et al.,* [1] proved that MinHOM is polynomial-time equivalent to VCSP. Certain minimum cost homomorphism problems have polynomial time algorithms [4, 6, 5, 10], but most are NP-hard. Therefore we investigate the approximability of these problems. Note that we approximate the cost over real homomorphisms, rather than approximating the maximum weight of satisfied constraints, as in, say, MAXSAT.

Complexity and approximability of minimum cost homoomorphism problems, and in general constraint satisfaction problems, are often studied under the existence of *polymorphisms*. A *product* of digraphs D and H has the vertex set $V(D) \times V(H)$ and arc set $A(D \times H)$ consisting of all pairs $(u,x)(v,y)$ such that $uv \in A(D)$ and $xy \in A(H)$. The product of k copies of the same digraph H is denoted by H^k. A polymorphism of H of order k is a homomorphism of H^k to H. In other words, it is a mapping f from the set of k-tuples over $V(H)$ to $V(H)$ such that if $x_i y_i \in A(H)$ for $i = 1, 2, \ldots, k$, then $f(x_1, x_2, \ldots, x_k) f(y_1, y_2, \ldots, y_k) \in A(H)$.

A polymorphism f is *conservative* if each value $f(x_1, x_2, \ldots, x_k)$ is one of the arguments $x_1, x_2, \ldots, x_k$. A binary (order two) f polymorphism that is conservative and commutative ($f(x,y) = f(y,x)$ for all vertices x, y) is called a *CC polymorphism.* If f is additionally associative ($f(f(x,y),z) = f(x, f(y,z))$ for all vertices x, y, z) it will be called a *conservative semi-lattice* or an *CSL polymorphism.* A CSL polymorphism of H naturally defines a binary relation $x \leq y$ on the vertices of H by $x \leq y$ if and only if $f(x,y) = x$; by associativity, the relation $\leq$ is a linear order on $V(H)$, which we call a *min-ordering* of H. In other words, an ordering of vertices $v_1 < v_2 < \cdots < v_n$ of H is a min-ordering if and only if $uv \in A(H), u'v' \in A(H) \implies \min(u,u') \min(v,v') \in A(H)$. Yet another way to state this is as follows. The ordering $v_1 < v_2 < \cdots < v_n$ of $V(H)$ is a min-ordering if and only if $uv \in A(H), u'v' \in A(H)$ and $u < u', v' < v$ implies that $uv' \in A(H)$. It is also clear that, conversely, a min-ordering $<$ of H defines a CSL polymorphism $f : H^2 \to H$ by $f(x,y) = \min(x,y)$. Similarly, we define *max-ordering* and *min-max-ordering* as follows.

Definition 1.1. *The ordering $v_1 < v_2 < \cdots < v_n$ of $V(H)$ is a*

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

Hell *et al.,* [11] showed a dichotomy for approximating the MinHOM(H) when H is a bipartite graph. When H admits a min-ordering (complement of H is a *circular arc* graph) then MinHOM(H) admits a constant approximation algorithm and otherwise there is no approximation for MinHOM(H). Beyond this, there is no result concerning the approximability of MinHOM(H). We remark that, [11, 14] are the only two results concerning the approximability of VCSP(Γ) for languages Γ containing cost functions that can take infinite values. This paper improves state-of-the-art by providing constant factor approximation algorithms for two important cases, namely *bi-arc* digraphs (digraphs with a min-ordering), and *k-arc* digraphs (digraphs with a k-min-ordering).

We first give a formal definition of digraph astroidal triple (DAT). Let H be a digraph.

Definition 1.2. *Let $\hat{H}^2$ be a digraph with vertex set $\{(a,b) | a, b \in A(H)\}$ and arc set*

$$\{(a,b)(a',b') | aa', bb' \in A(H), ab' \notin A(H)\} \cup \{(a,b)(a',b') | a'a, b'b \in A(H), b'a \notin A(H)\}.$$

We say x, y are invertible if $(x,y), (y,x)$ both belong to the same strong component of $\hat{H}^2$ and we say (x,y) is an invertible pair. Note that if x, y are invertible then H does not admit a semmi-lattice [9].

Definition 1.3. *Let $\hat{H}^3$ be a digraph with vertex set $\{(a,b,c) | a, b, c \in A(H)\}$ and arc set*

$$\{(a,b,c)(a',b',c') | aa', bb', cc' \in A(H), ab', ac' \notin A(H)\} \cup \{(a,b,c)(a',b',c') | a'a, b'b, c'c \in A(H), b'a, c'a \notin A(H)\}.$$

Definition 1.4. *A digraph asteroidal triple of H is an induced sub-digraph of $\hat{H}^3$ on three directed paths P_1 from (a,b,c) to (α,β,β), P_2 from (b,a,c) to (α,β,β), and P_3 from (c,a,b) to (α,β,β) where (α,β) is an invertible pair.*

If H contains a DAT then all three pairs $(a,b),(b,c),(c,a)$ are invertible. Moreover, H does not admit a conservative majority polymorphism g because the value of $g(a,b,c)$ can not be any of the a,b,c [9].

2 An Approximation Algorithm (LP formulation)

It was shown that if H contains a DAT then LHOM(H) is NP-complete. Let $H_1=(B,W)$ be a bipartite digraph where all the arcs of H_1 are from B to W. If H_1 does not admits a min-ordering then LHOM(H_1) is NP-complete.

It was shown in [11] that bipartite graph H admits a min ordering iff H is a co-circular arc bigraph. The authors in [2] showed that LHOM(H) is polynomial time solvable if H is a co-circular bi-arc graph and NP-complete otherwise. Therefore if bipartite graph H does not admit a min ordering then LHOM(H) is NP-complete. Let $H_1=(B,W)$ be a bipartite digraph where all the arcs of H_1 are from B to W. Now according to the above discussion the following proposition follows easily.

Proposition 2.1. *Let $H_1=(B,W)$ be a bipartite digraph where all the arcs of H_1 are from B to W. Then LHOM(H_1) is polynomial time solvable if H_1 admits a min ordering, otherwise, LHOM(H_1) is NP-complete.*

We note that according to [9], if H contains a DAT then LHOM(H) is NP-complete.

Definition 2.2. *Let $H=(I,A)$ be a digraph. Let $H^*=(I,I',A^*)$ be a biparite digraph with vertex set $I,I'A$ where I' is a copy of I and ij', $i\in I$, $j'\in I'$ is an arc of H^* iff $ij\in A(H)$.*

Proposition 2.3. *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 the following hold*

- *If H is DAT-free then LHOM(H^*) is polynomial time solvable for instance D^*,H^*,L' where $L'(v)=L'(v')=L(v)$ for every $v\in V(G)$.*
- *H^* admits a min ordering.*

Proof. Suppose $f:V(D)\rightarrow_L V(H)$ be an L-homomorphism (list homomorphism with respect to the lists L) from D to H. Then $f':V(D^*)\rightarrow V(H^*)$ in which $f'(v)=f'(v')=f(v)$ for every $v\in V(D)$ is an L'-homomorpshism from D^* to H^*. Now by Proposition 2.1 H^* must admit a min ordering. $\square$

In the remaining part we focus on finding such a f^* with cost within a constant factor from the optimal one.

Let $a_1,a_2,\dots,a_p$ be an ordering of the vertices in I and $b_1,b_2,\dots,b_p$ be an ordering of the vertices of I' such that $a_1<a_2<\cdots<a_p<b_1<b_2<\cdots<b_p$ is a min ordering. Note that each a_i has a copy $b_{\pi(i)}$ in $\{b_1,b_2,\dots,b_p\}$ where π is a permutation on $\{1,2,3,\dots,p\}$. Observe that all the arcs go from I to I'.

Construct $D^*=(V,V',A')$ from input graph $D=(V,A)$ as in Definition **??**. Now we construct an instance of the MinHOM(H^*) for the input graph D^*. We set $c(v',b_j)=c(v,a_i)$ where $\pi(i)=j$. Here c is the cost function and $c(x,i)$ is the cost of mapping vertex $x\in V(G)$ to vertex $i\in V(H)$.

Define $\ell^+(i)$ to be the smallest subscript j such that b_j is an out-neighbour of a_i (and $\ell^-(i)$ to be the smallest subscript j such that a_j is a in-neighbour of b_i).

Introducing Lists: Before presenting the rest of the inequalities of $\mathcal{S}$, we give a procedure to modify lists associated to the vertices of D with vertex set V and arc set $A(D)$. To each vertex $x\in D$, we associate a list $L_0(x)$ that initially contains $V(H)$. Think of $L_0(x)$ 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 $a \in L_0(x)$. If there is no out-neighbor (in-neighbor) of a in $L_0(y)$ then remove a from $L_0(x)$. 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. Therefore, in the rest of the paper assume lists are not empty. Moreover, non-empty lists guarantee a homomorphism. In the following three sections we assume the target digraph H admits a min ordering.

Lemma 2.4. *[8] 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.*

We define the lists L for $D^* = (V, V', A')$. For every $v \in V$, $L(v) = L_0(v)$ and for every $v' \in V'$, $L(v') = \{b_{\pi(i)} | a_i \in L_0(v)\}$.

Now we are ready to define the system of linear equation $\widehat{S^*}$ as follows. For every vertex v of D^* and every vertex a_i, b_j, $j = \pi(i)$ of H^* define variables v_i, v'_j, the goal is to minimize the following objective function:

$$\text{(LP)} \qquad \sum_{v \in V(G), i \in I} c(v, a_i)(v_i - v_{i+1}) + c(v', b_{\pi(i)})(v'_{\pi(i)} - v'_{\pi(i)+1})$$

subject to:

$$\begin{array}{lll}
\text{(C1)} & v_i, v'_i \geq 0 & \\
\text{(C2)} & v_1 = v'_1 = 1 & \\
\text{(C3)} & v_{p+1} = v'_{p+1} = 0 & \\
\text{(C4)} & v_{i+1} \leq v_i \textit{ and } v'_{j+1} \leq v'_j & \\
\text{(C5)} & v_{i+1} = v_i \textit{ and } v'_{\pi(i)+1} = v'_{\pi(i)} & \text{if } a_i \notin L(v) \\
\text{(C6)} & u_i \leq v'_{l^+(i)} & \forall\, uv' \in A(D^*) \\
\text{(C7)} & v'_i \leq u_{l^-(i)} & \forall\, uv' \in A(D^*)
\end{array}$$

The following lemma immediately follows from the construction of H^* and G^*.

Lemma 2.5. *There exists homomorphism $f : G \to H$ with cost c if and only if there exists homomorphism $f^* : G^* \to H^*$ with cost $2c$ such that, if $f^*(v) = a_i$ then $f^*(v') = b_j$ with $j = \pi(i)$.*

Note that once we performed the arc-consistency and pair-consistency check for the vertices in G^*, 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 $a_{\pi^{-1}(j)}$.

Let E' denote the set of all pairs (a_i, b_j) such that $a_i b_j$ is not an arc of H^*, but there is an arc $a_i b_{j'}$ of H^* with $j' < j$ and an arc $a_{i'} b_j$ of H^* with $i' < i$. Let $E = A(H^*)$ and define $H^{*\prime}$ to be the digraph with vertex set $V(H^*)$ and arc set $E \cup E'$. Note that E and E' are disjoint sets.

Observation 2.6. *The ordering $a_1, a_2, \cdots, a_p, b_1, b_2, \ldots, b_p$ is a min-max-ordering of $H^{*\prime}$.*

It is easy to see that (see also in the bipartite case in [11]) if the $\mathcal{S}$ has a solution with integeral solution then there exists a homomorphism from D^* to H^*. However, we need to add further inequalities to $\mathcal{S}$ to ensure that we only admit homomorphisms of D^* to H^*, i.e., avoid mapping arcs of D^* to the arcs in E'.

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

$$v'_j \leq u_s + \sum_{\substack{t<i \\ a_t b_j \in E \\ a_t \in L(u)}} (u_t - u_{t+1}) \qquad \text{if } a_s \in L(u) \text{ is the first in-neighbour of } b_j \text{ after } a_i \tag{C8}$$

$$v'_j \leq v'_{j+1} + \sum_{\substack{t<i \\ a_t b_j \in E \\ a_t \in L(u)}} (u_t - u_{t+1}) \qquad \text{if } b_j \text{ has no in-neighbour after } a_i \tag{C9}$$

$$u_i \leq v'_s + \sum_{\substack{t<j \\ a_i b_t \in E \\ b_t \in L(v')}} (v'_t - v'_{t+1}) \qquad \text{if } b_s \in L(v') \text{ is the first out-neighbour of } a_i \text{ after } b_j \tag{C10}$$

$$u_i \leq u_{i+1} + \sum_{\substack{t<j \\ a_i b_t \in E \\ b_t \in L(v')}} (v'_t - v'_{t+1}) \qquad \text{if } a_i \text{ has no out-neighbour after } b_j \tag{C11}$$

Additionally, for every pair $(x, y) \in V(D^*) \times V(D^*)$ consider a list of possible pairs (a, b), $a \in L(x)$ and $b \in L(y)$. Perform *pair consistency* 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.

Therefore, by pair consistency, add the following constraints for every u^*, v^* in $V(D^*)$ and $c_i \in L'(u^*)$:

$$u^*_i - u^*_{i+1} \leq \sum_{\substack{j: \\ (c_i, c_j) \in L(u^*, v^*)}} (v^*_j - v^*_{j+1}) \qquad \text{here } * \text{ refer to blank or } ' \tag{C12}$$

Due to the additional condition on f^*, for every $u, u' \in D^*$ and $a_i \in H^*$, we add the following constraint into $\widehat{S^*}$:

$$u_i - u_{i+1} = u'_{\pi(i)} - u'_{\pi(i)+1} \tag{C13}$$

Lemma 2.7. *There is a one-to-one correspondence between homomorphisms from G to H and integer solutions of $\widehat{S^*}$.*

Proof. For homomorphism $f : G \to H$, if $f(v) = a_t$ we set $v_i = 1$ for all $i \leq t$, otherwise we set $v_i = 0$, we also set $v'_j = 1$ and $v'_{j+1} = 0$ where $\pi(a_i) = b_j$. We set $v_1 = 1$, $v'_1 = 1$ and $v_{p+1} = v'_{p+1} = 0$ for all $v, v' \in V(G^*)$. Now all the variables are non-negative and we have $v_{i+1} \leq v_i$ and $v'_{j+1} \leq v'_j$. We note that by this assignment constraint (C13) is satisfied. It remains to show that $u_i \leq v'_{l^+(i)}$ for every edge $uv \in H$ and every arc $uv' \in H^*$. 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 $i \leq r$; and $v'_s = 1$, whence $s < l^+(i)$. Since $a_i b_{l^+(i)}$ and $a_r b_s$ both are arcs of H^* with $i \leq r$ and $s < l^+(i)$, the fact that H^* has a min-ordering implies that $a_i b_s$ must also be an arc of H, contradicting the definition of $l^+(i)$. The proof for $v'_j \leq u_{l^-(i)}$ is analogous.

Conversely, if there is an integer solution for $\widehat{S^*}$, we define a homomorphism f from D to H. First we define a homomorphism $g : D^* \to H^*$ as follows : we let $g(u) = a_i$ when i is the largest subscript with $v_i = 1$, and $g(v') = b_j$ when j is the largest subscript with $v_j = 1$. We prove that this is indeed a homomorphism by showing that every edge 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 $a_r b_s$ is an arc in H^*. Observe that, by (C6) and (C7), $1 = u_r \leq v'_{l^+(r)} \leq 1$ 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 $a_r b_{l^+(r)}$ and $a_{l^-(s)} b_s$ are arcs of H^*, we must have the arc $a_r b_s$ in H^* as H^* admits a min-ordering. Furthermore, $g(u) = a_i$ if and only if $u_i = 1$ and $u_{i+1} = 0$, so, $c(u, a_i)$ contributes to the sum if and only if $g(u) = a_i$ and $c(v', b_j)$ contributes to the sum if and only if $g(v') = b_j$.

Now let $f(u) = a_i$ when $g(u) = a_i$. We show that if uv is an arc of H then $f(u)f(v)$ is an arc of H. Since g is a homomorphism from D^* to H^*, $g(u)g(v') \in A(H^*)$. Suppose $g(v') = b_j$. This means $u_i = v'_j = 1$, $u_{i+1} = v'_{j+1} = 0$. Now by constraint (C13), we have $v_{\pi^{-1}(j)} = 1$, and $v_{\pi^{-1}(j)+1} = 0$, and hence, we have $f(v) = a_{\pi^{-1}(j)}$. Now by definition of H^*, $a_i a_{\pi^{-1}(j)} 1$ is an arc of H because $a_i b_j$ is an arc of H^*. Furthermore, $f(u) = a_i$ if and only if $u_i = 1$ and $u_{i+1} = 0$, so, $c(u, a_i)$ contributes to the sum if and only if $f(u) = a_i$.

□

Algorithm 1 Approximation MinHOM(H)

procedure APPROX–MINHOM(H)
    Construct H' from H (as in Section **??**)
    Let u_is be the (fractional) values returned by the extended LP
    Choose a random variable $X \in [0, 1]$
    For all u_is: 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 $f(u)f(v) \in E(H') \setminus E(H)$ **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

procedure SHIFT(f, x)
    Let Q be a Queue, $Q.enqueue(x)$
    while Q is not empty **do**
        $v \leftarrow Q.dequeue()$
        for uv with $f(u)f(v) \notin E(H)$ **or** vu with $f(v)f(u) \notin E(H)$ **do**
▷ Here we assume the first condition hold, the other case is similar
            Let $t_1 < \cdots < t_k$ be indices so that $a_{t_j} < f(v), a_{t_j} \in L(v), f(u)a_{t_j} \in E(H)$
            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 every $vz \in A(D^*)$ with $f(v)f(z) \notin E(H)$ **or** $zv \in A(D)$ with $f(z)f(v) \notin E(H)$ **do**
                $Q.enqueue(z)$
    return f ▷ f is a homomorphism from D to H'

2.1 The Approximation Algorithm (LP rounding)

In what follows, we describe our approximation algorithm for MinHOM(H) where the fixed digraph H has a min-ordering. We start off with an overview of our algorithm. The proofs of the correctness and approximation bound are postponed for the later subsections.

Given a DAT-free digraph H with vertex set I and arc set $A(H)$, Algorithm 1, first construct bipartite digraph H^* with vertex set $I \cup I'$ and arc set $A(H^*)$ from H as explained in the previous subsection. 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 the vertices in I' such that $a_1 < a_2 < \cdots < a_p < b_1 < b_2 < \cdots < b_p$ is a min-ordering of H (this task is polynomial, see [**arash-esa-14**]). Note that $a_1 < a_2 < \cdots < a_p < b_1 < b_2 < \cdots < b_p$ is a min-max ordering of $H^{*\prime}$ (obtaining by adding extra arcs to H^*).

By Lemma 2.7, the integral solutions of the extended LP is 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 $\mathcal{S}$, restricted to constraints $C_1, \ldots, C_7$, is known to be totally unimodular [16, 11] 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*. The rounding procedure has three main steps.

Using Random Variable X to obtain the first partial homomorphism We choose a random variable $X \in [0,1]$, and define the rounded values $\hat{u}_i = 1$ ($u \in V$, $a_i \in I$) when $X \leq u_i$, and $u'_i = 0$ otherwise. Similarly define the rounded values $\hat{v}'_j = 1$ ($v' \in V'$, $b_j \in I'$) when $X \leq v'_j$, and $\hat{v}'_i = 0$ otherwise.

It is easy to check that the rounded values satisfy the original inequalities, i.e., correspond to a homomorphism f of D^* to $H^{*\prime}$.

First Round of shifting to get a homomrphism from D^* to H^* Now the algorithm will once more modify the solution f to become a homomorphism of D^* to H^*, i.e., to avoid mapping arcs of D^* to the arcs in E'. This will be accomplished by another randomized procedure, which we call *shifting*. We choose 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^* is mapped by f. If F is empty, we need no shifting. Otherwise, let a_ib_j be an arc of F.

Suppose for some edge uv' of D^*, $\hat{u}_i = 1$, $\hat{u}_{i+1} = 0$, $\hat{v}'_j = 1$, $\hat{v}'_{j+1} = 0$ ($a_ib_j \notin A(H^*)$). By Observation **??**, either b_j has no in-neighbor after a_i or a_i has no out-neighbor after b_j. Suppose the former is the case. The other case is similar. We use random variable Y for shifting the image of v' from b_j to some b_t which $a_ib_t \in E(H^*)$, and b_t appears before b_j in the min-ordering of H^*. Consider the set of such b_t's. This set is non-empty and suppose it consists of b_t with subscripts t ordered as $t_1 < t_2 < \ldots t_k$. Let $P_{v',t} = \frac{v'_t - v'_{t+1}}{P_{v'}}$ with $P_{v'} = \sum_{a_ib_t \in E(H^*),\ t<j} (v'_t - v'_{t+1})$. Select b_{t_q} if $\sum_{p=1}^{q} P_{v',t_p} < Y \leq \sum_{p=1}^{q+1} P_{v',t_p}$. Thus a concrete b_t is selected with probability $P_{v',t}$, which is proportional to the difference of the fractional values $v'_t - v'_{t+1}$. This means we set $\hat{v}'_t = 1$ and $\hat{v}'_{t+1} = 0$, i.e. $f(v') = b_t$. Let w be an in-neighbor v' such that $f(w) = a_\ell$ and $f(v') = b_j$ and $a_\ell b_j \in A(H^*)$. We note that since the in-neighbors of b_j are before a_i in the ordering $<$, and because $<$ is a min-ordering $a_\ell b_t$. This means there is no need to change the image of w.

We repeat the above procedure, until f becomes a homomorphism from D^* to H^*.

Second Round of shifting to make f consistent on both I and I' and obtain a homomorphism from D to H

We say a vertex u of D^* is *unstable* if $f(u) = i$ and $f(u') \neq b_{\pi(i)}$, i.e. $\hat{u}_i = 1$ and $\hat{u}'_{\pi(i)} = 0$ and $\hat{u}'_q = 1$, $\hat{u}'_{q+1} = 0$ with $q \neq \pi(i)$. The goal of this step is to modify f so that it is a homomorphsim from D^* to H^* and also make f stable on both V and V', i.e. there is no unstable vertex $u \in V$.

Now we start a Breath First Search (BFS) in $V(G^*)$ and continue as long as there exists a non-fixed vertex u in G. We start from the biggest subscripts i for which there exists an unstable u with $\hat{u}_i = 1$, $\hat{u}_{i+1} = 0$. We may assume that $\pi(i) < q$ (the procedure would be analogous when $i < \pi^{-1}(q)$). In this case we shift the image of u' from b_q to $b_{\pi(i)}$ (shift the image of u from a_i to $a_{\pi^{-1}(q)}$). Now this may force us to change the image of some neighbor of u', say v from a_ℓ to some a_t if $a_\ell b_{\pi(i)}$ is not an arc of H^*, and consequently the image of v' should be moved to $a_{\pi(t)}$. Now this change may affect some out-neighbor of v, say w'. So the next step would be shifting the image of w' to an out-neighbor of $a_{\pi^{-1}(t)}$, say b_r (only if the current image of w' is not an out-neighbor of $a_{\pi^{-1}(t)}$), and consequently the image of w should be moved to $a_{\pi^{-1}(j)}$.

This means we need to deploy a shifting procedure which is almost identical to the shift procedure in Algorithm 1. The BFS shift algorithm switches between the vertices in V and V' at each step.

2.2 Correctness and Analysis

Lemma 2.8. *Procedure* Shift *runs in polynomial time and it returns a homomorphism from D to H.*

Proof. We need to observe that H^* may not be weakly connected. However, every weakly connected component of D^* is mapped to a weakly connected component of H^* after the first rounding step of the LP values.

We need to show that the Shift does not enter a loop. In other words, once we shift the image of u' from b_q to $b_{\pi(i)}$ then either the image of u and u' stay on a_i and $b_{\pi(i)}$ (respectively) or the image of u and u' shift to some a_r, $r \neq i$ and $a_{\pi(r)}$ after polynomially many steps. Suppose there is a such a loop. This means there is a sequence $u', v_1, v_2', v_3, v_4', \ldots, v_{2k}', u$ and sequence □

//////////////////// Form the old part ////////////////////
We choose 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 of D to H, i.e., to avoid mapping arcs of D to the arcs in E'. This will be accomplished by another randomized procedure, which we call *shifting*. We choose 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 is 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 **??** 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, consider the set of all vertices a_t with $t < j$ such that $a_i a_t \in E$ and $a_t \in L(v)$.

Lemma 2.9. *During procedure* Shift, *the set of indices $t_1 < \cdots < t_k$ considered in Line 19 is non-empty.*

Proof. In procedure Shift, consider vz such that $f(v)f(z) \notin E(H')$ and $f(v) = a_t$ and $f(z) = a_l$. This means $0 < v_t - v_{t+1}$, and together with constraint (C12), it implies $0 < v_t - v_{t+1} \leq \sum_{\substack{j: \\ (a_t, a_j) \in L(v,z)}} (z_j - z_{j+1})$. Therefore, there must be an index l' such that $(a_t, a_{l'}) \in L(v, z)$. It remains to show that $a_{l'}$ appears before a_l in the min-ordering. There are two cases to consider. First is $f(v)$ is set to a_t in rounding step (Line 5). Second is image of v was shifted from a_j to a_t in procedure Shift.

For the first case, note that, since f is a homomorphism from D to H' then $a_t a_l \in E(H') \setminus E(H)$. Arc vz is mapped to $a_t a_l$ in rounding step (Line 5) according to random variable X. Note that, during procedure Shift, we do not map any arc of D to edges in $E(H') \setminus E(H)$. Therefore, we have $X \leq v_t, z_l$. Consider the situation where a_l has no in-neighbor after a_t. Let a_s be the first out-neighbor of a_t after a_l, then we have $z_s < X \leq v_t$. This together with inequality (C10) implies that $0 < \sum_{\substack{l' < l \\ a_t a_l \in E \\ a_{l'} \in L(z)}} (z_{l'} - z_{l'+1})$. Hence, there exists an index $l' < l$ as we wanted. The argument for the case where a_t has no out-neighbor after a_l is similar.

For the second case, before mapping v to a_t, there was an index a_j such that $a_t < a_j$. There are two cases regarding $a_j a_l$. Either it is in $E(H)$ or it is in $E(H') \setminus E(H)$. In both cases, $a_{l'}$ must appear before a_l as otherwise, minmax-ordering implies $a_t a_l \in E(H')$, contradicting our assumption. □

By Lemma 2.9, this set is not empty. Suppose the set consists of a_t with subscripts t ordered as $t_1 < t_2 < \cdots < t_k$. 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 neighbours of v. This continues until no more shift is required (see Figure 1 for an illustration).

Lemma 2.10. *Procedure* SHIFT *runs in polynomial time and returns a homomorphism from D to H'.*

Proof. It it easy to see that, if there exists a homomorphism from D to H, then there is a homomorphism from D to H that maps every vertex of D to the smallest vertex in its list(Lemma 2.4). We show that, a sequence of shifting, either stops at some point, or it keeps shifting to a smaller vertex in each list. In the later case, after finite (polynomially many) steps, we end up mapping every vertex of D to the smallest vertex in its list.

Consider an arc $vz \in A(D)$. Suppose $f(v) = a_t$ and $f(z) = a_l$. Assume that we have shifted the image of v from a_t to $a_{t'} \in L(v)$ where $a_{t'}$ is before a_t in the min-ordering. If $a_{t'}a_l$ is in $E(H)$ then we do not have to shift the image of z. Note that, since $a_{t'}$ is in $L(v)$ then it has to have an out-neighbor in $L(z)$. Let say $a_{l'} \in L(z)$ is an out-neighbor of $a_{t'}$. If $a_{l'}$ is after a_l in the min-ordering then it implies $a_{t'}a_{l'} \in A(H)$. Else, $a_{l'}$ is before a_l in the min-ordering and we shift the image of z to a smaller vertex in its list. □

Lemma 2.10 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.

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 2.9 and 2.10.

Theorem 2.11. *Algorithm 1, in polynomial time, returns a homomorphism of D to H.*

Consider Figure 1 as an example of the algorithm. The right digraphs (D_1, 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 is 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 H, $1, 2, 3, 4, 5, 6, 7, 8$ is a min ordering 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 $1, 2$.

2.3 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 26â29 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 situations:

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.

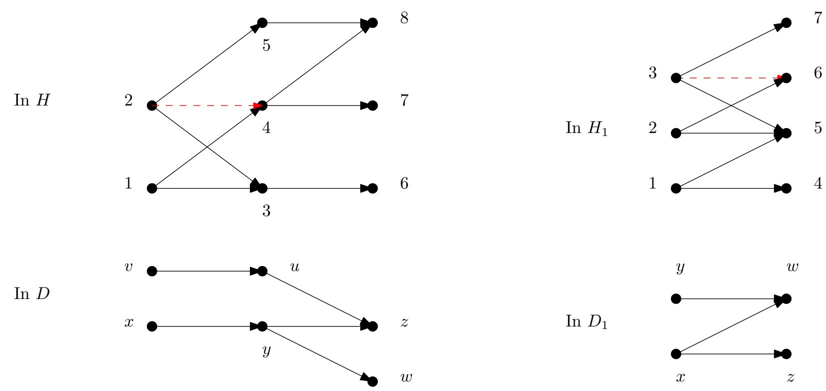

Figure 1: The shifting process in bipartite graphs and digraphs with a min-ordering.

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

Lemma 2.12. *The expected contribution of one summand, say $v'_t - v'_{t+1}$, to the expected cost of w' is at most $|V(H)|^2 c(v, a_t)(v_t - v_{t+1})$.*

Proof. Vertex v is mapped to a_t in two cases. The first case is where v is mapped to a_t by rounding Line 5, and is not shifted away. In other words, we have $v'_t = 1$ and $v'_{t+1} = 0$ after rounding, and these values do not change by procedure Shift. Hence, for this case we have:

$$\begin{aligned} Pr[f(v) = a_t] &= Pr[v_{t+1} < X \le v_t] \cdot Pr[v \text{ is not shifted in procedure Shift}] \\ &\le v_t - v_{t+1} \end{aligned}$$

Whence this situation occurs with probability at most $v_t - v_{t+1}$, and the expected contribution is at most $c(v, a_t)(v_t - v_{t+1})$.

Second case is where $f(v)$ is set to a_t during procedure Shift. The algorithm calls Shift if there exists $u^0u^1 \in A(D)$ such that $f(u^0)f(u^1) \in E(H') \setminus E(H)$ (Line 8). Let us assume it calls Shift(f, u^1). Procedure Shift modifies images of vertices $u^1, u^2, \cdots$. Consider the last time that Shift changes image of v. Note that $u^1, \cdots, u^k = v$ is an oriented walk, meaning that there is an arc between every two consecutive vertices of the sequence and the u^is are not necessarily distinct.

We first compute the contribution for a fixed j, that is the contribution of shifting v from a fixed a_j to a_t. Consider the simplest case where $k = 1$. In this case v is first mapped to $a_j, j > t$, by rounding, and then re-mapped to a_t during procedure Shift. This happens if there exist i and u such that uv is an arc of D mapped to $a_ia_j \in E'$, and then the image of v is shifted to a_t ($a_t < a_j$ in the min-ordering), where $a_ia_t \in E(H)$. In other words, we have $u'_i = v'_j = 1$ and $u'_{i+1} = v'_{j+1} = 0$ after rounding (Line 5); and then v is shifted from a_j to a_t. Therefore,

$$\begin{aligned} &Pr[u'_i = v'_j = 1, u'_{i+1} = v'_{j+1} = 0] = Pr[\max\{u_{i+1}, v_{j+1}\} < X \le \min\{u_i, v_j\}] \\ &= \min\{u_i, v_j\} - \max\{u_{i+1}, v_{j+1}\} \le v_j - v_{j+1} \\ &\le \sum_{\substack{t<j \\ a_ia_t \in E \\ a_t \in L(v)}} (v_t - v_{t+1}) = P_v \end{aligned}$$

The last inequality is because a_j has no in-neighbor after a_i and it follows from inequality (C9). Having uv mapped to a_ia_j in the rounding step, we shift v to a_t with probability $P_{v,t} = \frac{(v_t - v_{t+1})}{P_v}$. Note that the upper bound P_v is independent from the choice of u and a_i. Therefore, for a fixed a_j, the probability that v is shifted from a_j to a_t is at most $\frac{v_t - v_{t+1}}{P_v} \cdot P_v = v_t - v_{t+1}$.

For $k > 1$, consider oriented walk $u^0, \cdots, u^k = v$. Before calling $\textsc{Shift}(f, u^1)$, this walk is mapped to some vertices in H. Without loss of generality, let us assume these vertices are $a_0, a_1, \cdots, a_k$. Note that a_is may not be distinct. Once again we compute the contribution for a fixed $k = j$, that is the contribution of shifting v from a fixed $a_k = a_j$ to a_t. First, we give an upper bound on the probability of existence of such a situation after rounding step (Line 5),

$$\begin{aligned}
&Pr[u_0^{0\prime} = \cdots = u_k^{k\prime} = 1, u_1^{0\prime} = \cdots = u_{k+1}^{k\prime} = 0] \\
&= Pr[\max\{u_1^0, \cdots, u_{k+1}^k\} < X \leq \min\{u_0^0, \cdots, u_k^k\}] \\
&= \min\{u_0^0, \cdots, u_k^k\} - \max\{u_1^0, \cdots, u_{k+1}^k\} \leq (u_k^k) - u_{k+1}^k = v_j - v_{j+1} \\
&\leq \sum_{\substack{t<k-1 \\ a_{k-1}a_t \in A(H) \\ a_t \in L(u^{k-1})}} (u_t^{k-1} - u_{t+1}^{k-1}) = P_v
\end{aligned}$$

Now the algorithm calls $\textsc{Shift}(f, u^1)$ and, in procedure $\textsc{Shift}$, images of $u^1, u^2, \cdots, u^k = v$ are changed in this order. We are interested in probability of mapping v from fixed $a_k = a_j$ to a_t. Analyzing the situation for u^1 is the same as the case for $k = 2$. As induction hypothesis, assume for $u^1, \cdots, u^{k-1}$, the probability that the algorithm shifts image of u^i to some a_i is at most $u_i^i - u_{i+1}^i$, particularly for $u^{k-1} = u$. At this point $f(u) = a_i$ and $f(v) = a_k$. Note that $a_i a_k$ is not an edge in H, as otherwise no change is required for image of v. Here, the algorithm choses a_t where $a_t \in L(v), a_t < a_k$ and $a_i a_t \in E(H)$ with probability

$$\frac{v_t - v_{t+1}}{\sum_{\substack{j<k \\ a_i a_j \in A(H) \\ a_j \in L(v)}} (v_j - v_{j+1})}$$

It remains to argue that $u_i - u_{i+1} \leq \sum_{\substack{j<k \\ a_i a_j \in A(H) \\ a_j \in L(v)}} (v_j - v_{j+1})$. Having that gives us the probability of shifting v from a_j to a_t is at most $v_t - v_{t+1}$.

Observe that a_i does not have any neighbor a_s after a_k. This is because $a_{k-1}a_k, a_i a_s \in A(H')$ and the min-ordering implies $a_i a_k \in A(H)$ which contradicts our assumption. Thus, by inequality (C11), we get $u_i - u_{i+1} \leq \sum_{\substack{j<k \\ a_i a_j \in A(H) \\ a_j \in L(v)}} (v_j - v_{j+1})$. This completes the proof.

Let $L(v) = a_1^v \cdots, a_k^v$. Clearly, during procedure $\textsc{Shift}$, image of v can be shifted to a_i^v from any of vertices $a_{i+1}^v, \cdots, a_k^v$. For any fixed $a_j \in \{a_{i+1}^v, \cdots, a_k^v\}$, this shift is initiated from vertices in $V(H)$ that are incident with some edges in E', and reaches to a_j to shift image of v. Shifting of image of v happens because of missing edges from a_j that is at most $|V(H)| - d^+(a_j) - d^-(a_j) \leq |V(H)|$ ($d^+(a_j)$ and $d^-(a_j)$ are out-degree and in-degree of a_j respectively). Therefore, the contribution of v and a_i^v to the expected value of w' is at most $(1 + |V(H)|(k-i))c(v, a_i^v)(v_{a_i^v} - v_{a_{i+1}^v})$ where $(v_{a_i^v} - v_{a_{i+1}^v})$ is the upper bound on the probability provided before. Thus the expected value of w' is at most

$$\begin{aligned}
\mathbb{E}[w'] = \mathbb{E}\left[\sum_{v,i} c(v, a_i)(v_i' - v_{i+1}')\right] &= \sum_{v,i} c(v, a_i)\mathbb{E}[v_i' - v_{i+1}'] \\
&\leq |V(H)|^2 \sum_{v,i} c(v, a_i)(v_i - v_{i+1}) \leq |V(H)|^2 w \leq |V(H)|^2 w^*.
\end{aligned}$$

$\square$

Theorem 2.13. *Algorithm 1 returns a homomorphism with expected cost $|V(H)|^2$ times optimal solution. The algorithm can be derandomized to obtain a deterministic $|V(H)|^2$-approximation algorithm.*

Proof. At this point we have proved that Algorithm 1 produces a homomorphism whose expected cost is at most $|V(H)|^2$ times the minimum cost. It can be transformed to a deterministic algorithm as follows. There are only polynomially many values v_t (at most $|V(D)| \cdot |V(H)|$). When X lies anywhere between two such consecutive values, all computations will remain the same. Thus we can derandomize the first phase by trying all these values of X and choosing the best solution. Similarly, there are only polynomially many values of the partial sums $\sum_{i=1}^{p} P_{u,t_i}$ (again at most $|V(D)||V(H)|$), and when Y lies between two such consecutive values, all computations remain the same. Thus we can also derandomize the second phase by trying all possible values and choosing the best. Since the expected value is at most $|V(H)|^2$ times the minimum cost, this bound also applies to this best solution. □

2.4 Rounding the LP values : Should be moved to a different place

The rounding algorithm has three main steps.

Using Random Variable X to obtain the first partial homomorphism For every variable u_i, $u \in V(D^*)$, set $\hat{u}_i = 1$ if $X \le u_i$ else $\hat{u}_i = 0$. Similarly for every v'_j, $v' \in V(D^*)$, set $\hat{v}'_j = 1$ if $X \le v'_j$ else $\hat{v}'_i = 0$.

At this we won't be able to interpret the values of $\hat{u}_i$, $\hat{u}'_{\pi(i)}$ and obtain a homomprphism. The reason is given in the first sentence in the next paragraph.

Shifting the images to repair the partial homomorphism a get a homomorphism f from D^* to H^*.
Suppose for some edge uv' of D^*, $\hat{u}_i = 1$, $\hat{u}_{i+1} = 0$, $\hat{v}'_j = 1$, $\hat{v}'_{j+1} = 0$ where $a_i b_j \notin A(H)$. By Observation **??**, either b_j has no in-neighbor after a_i or a_i has no out-neighbor after b_j. We also note that because of the constrains A5,A6, $a_i b_j$ is one of the arcs that needed to be add into H^* in order to obtain a min-max ordering for H'^*. Suppose the former is the case.

Choose a random variable $Y \in [0,1]$, which will guide to shift the image of v' from b_j to some b_t which $a_i b_t \in E(H^*)$, and b_t appears before b_j in the min-ordering of H^*. Consider the set of such b_ts. This set is non-empty and suppose it consists of b_t with subscripts t ordered as $t_1 < t_2 < \dots t_k$. Let $P_{v',t} = \frac{v'_t - v'_{t+1}}{P_{v'}}$ with $P_{v'} = \sum_{a_i b_t \in E(H^*),\ t<j} (v'_t - v'_{t+1})$. Select b_{t_q} if $\sum_{p=1}^{q} P_{v',t_p} < Y \le \sum_{p=1}^{q+1} P_{v',t_p}$. Thus a concrete b_t is selected with probability $P_{v',t}$, which is proportional to the difference of the fractional values $v'_t - v'_{t+1}$. This means we set $\hat{v}'_t = 1$ and $\hat{v}'_{t+1} = 0$. In order to be consistent in both side we also need to shift the image of v from ℓ (assuming $\hat{v}_\ell = 1$, $\hat{v}_{\ell+1} = 0$ in the first stage) to $a_{\pi^{-1}(t)}$. This is done by probability of $v'_t - v'_{t+1} = v_{\pi^{-1}(t)} - v_{\pi^{-1}(t)+1}$. Now this change may affect some out-neighbor of v, say w'. So the next step would be shifting the image of w' to an out-neighbor of $a_{\pi^{-1}(t)}$, say b_r (only if the current image of w' is not an out-neighbor of $a_{\pi^{-1}(t)}$), and consequently the image of w should be moved to $a_{\pi^{-1}(j)}$.

This means we need to deploy a shifting procedure which is almost identical to the shift procedure in Algorithm 1. The BFS shift algorithm switches between the vertices in V and V' at each step.

Making f consistent on both sides, i.e. on $I \cup I'$ This step is necessary if there is some $u \in V(D^*)$ such that $\hat{u}_i = 1$, $\hat{u}_{i+1} = 0$, and $\hat{u}'_\ell = 1$, $\hat{u}'_{\ell+1} = 0$, and $\ell \ne \pi(i)$. At this point i is the biggest index for which there exists such u. As we explained in the Section **??** we need to run a BFS shifting algorithm to make a consistent assignment, i.e. for every $i \in \{1, 2, \dots, p\}$, and every $u \in V(D)$, if $\hat{u}_i = 1$, $\hat{u}_{i+1} = 0$ then $\hat{u}_{\pi(i)} = 1$, $\hat{u}_{\pi(i)+1} = 0$.

Size of G	Majority				Min ordering balanced					
	Size of H				Size of H					
	H7		H15		H10 - Diclaw		H12		Random Hs	
	Average Ration	Min Ratio	Average Ration	Min Ratio	Average Ration	Min Ratio	Average Ration	Min Ratio	Average Ration	Min Ratio
100	0.999995	0.999896	0.999966	0.998726	0.986136	0.916565	0.99816	0.97043	0.98839	0.884701
150	0.999418	0.98528	0.999993	0.999682	0.986932	0.902389	0.996242	0.97920	0.941237	0.845272
200	0.998706	0.981862	1	1	0.981337	0.930423	0.99525	0.95748	0.999998	0.999864
300	0.999667	0.990967	0.999868	0.995851	0.987645	0.889039	0.999068	0.99359 8	0.999945	0.994763
1000	0.998999	0.989997	N/A	N/A	0.990757	0.966996	N/A	N/A	N/A	N/A
1100	N/A	N/A	N/A	N/A	0.990656	0.962157	N/A	N/A	N/A	N/A
1200	0.999879	0.988798	N/A	N/A	0.977565	0.931021	N/A	N/A	N/A	N/A

Figure 2: Ratio results from continuous and integral solution for minimum cost homomorphism, considering majority and min ordering balanced graphs H, with size of G ranging from 100 to 1200

3 Experiments

3.1 Finding a solution using GNU GLPK

GLPK extends for GNU Linear Programming Kit, and it is an open source software package, wrote in C. It is intended for solving large-scale linear programming problems(LP). GLPK is a well designed algorithm to solve LP problems, in a reasonable time. It implements different algorithms, such as, simplex method and the Interior-point method for non-integer problems and branch-and-bound together with Gomory's mixed integer cuts for integer problems. With GLPK we are able to add each constraint of our problem as a new row of a matrix. Before calculating the minimum cost, we have to set the type of solution we are looking for, integral only or if we allow a continuous cut.

3.2 Results

For our experiments, we have used graphs from 4 different classes: *Majority*, *Min ordering balanced*, *Mix and Min ordering bipartite*. For each class, we have used a variety of graphs and sizes, ranging from 7 to 15. Then, For a particular graph in each class, we use a variety of graphs G, created randomly, with size from 100 to 1300. for each instance, we execute it twice, once for continuous and once for only integral solution. To calculate the ratio, for a single graph H of size N, we execute our solution for each graph G of size T, 100 times. We then get the ratio by calculating the average of continuous and integral solution, for every instance of different sizes of G. Results of graphs from classes Majority and Min ordering balanced were summarized in figure 4, and from classes Mix and Min ordering bipartite can be viewed in figure 5.

	Mix				Min ordering bipartite													
Size of G	Size of H				Size of H													
	H8		H14		H7_1		H7_2		H7_3		H8		H9		H10		H12	
	AVG ratio	Min Ratio	AVG ratio	Min Ratio	AVG ratio	Min Ratio	AVG ratio	Min Ratio	AVG ratio	Min Ratio	AVG ratio	Min Ratio	AVG ratio	Min Ratio	AVG ratio	Min Ratio	AVG ratio	Min Ratio
100	0.995995	0.921548	0.963223	0.87101	0.999794	0.99762	0.996023	0.956193	0.999347	0.982341	0.99745	0.969842	0.996412	0.954612	0.986324	0.937843	0.956918	0.845843
150	0.98252	0.820747	0.978074	0.882826	0.999993	0.999856	0.99732	0.978942	0.999945	0.998512	0.989797	0.931819	0.997635	0.956999	0.9824	0.931672	0.971337	0.808316
200	0.989912	0.850658	0.984629	0.927852	0.999798	0.99664	0.999095	0.985997	0.998116	0.990253	0.99105	0.914226	0.995933	0.952518	0.99403	0.944462	0.966757	0.713945
300	0.98552	0.820499	0.971294	0.881047	0.999989	0.999724	0.99747	0.978127	0.99935	0.98872	0.992	0.90679	N/A	N/A	0.985672	0.943486	N/A	N/A

Figure 3: Ratio results from continuous and integral solution for minimum cost homomorphism, considering mix and min ordering bipartite graphs H, with size of G ranging from 100 to 300

Majority					Min ordering balanced			
Size of G	Size of H				Size of H			
	H7		H15		H10 - Diclaw		H12	
	Average Ration	Min Ratio	Average Ration	Min Ratio	Average Ration	Min Ratio	Average Ration	Min Ratio
1300	0.959287	0.86508	0.916007	0.783067	0.967156	0.922342	0.969588	0.91161
1500	0.96678	0.878423	0.982143	0.944924	0.979536	0.953689	0.97775	0.89523
1700	0.955847	0.882683	0.97783	0.837977	0.981337	0.955349	0.979853	0.93053
2000	0.966982	0.884385	0.957262	0.839615	N/A	N/A	N/A	N/A
3000	0.967934	0.94397	0.965702	0.891063	N/A	N/A	N/A	N/A

Figure 4: Ratio results from continuous and integral solution for minimum cost homomorphism, considering majority and min ordering balanced graphs H, with large sizes of G, ranging from 1300 to 3000

Mix					Min ordering bipartite											
Size of G	Size of H				Size of H											
	H8		H14		H7_1		H7_2		H7_3		H8		H10		H12	
	AVG ratio	Min Ratio	AVG ratio	Min Ratio	AVG ratio	Min Ratio	AVG ratio	Min Ratio	AVG ratio	Min Ratio	AVG ratio	Min Ratio	AVG ratio	Min Ratio	AVG ratio	Min Ratio
1300	0.98752	0.927072	0.98752	0.951324	0.974936	0.898161	0.985802	0.931392	0.9892	0.948316	0.94749	0.94749	0.942213	0.942213	0.97504	0.9236
1500	0.986391	0.963837	0.994971	0.963427	0.983722	0.948113	0.990652	0.97402	0.976557	0.972266	0.992537	0.974116	0.935	0.9157	0.961337	0.96316
1700	N/A	N/A	0.995041	0.968233	0.910783	0.910783	0.96466	0.96466	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2000	N/A	N/A	0.996976	0.983957	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
3000	N/A	N/A	0.966682	0.825441	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Figure 5: Ratio results from continuous and integral solution for minimum cost homomorphism, considering mix and min ordering bipartite graphs H, with large sizes of G, ranging from 1300 to 3000