Discrete Applied Mathematics 161 (2013) 670–676

Contents lists available at SciVerse ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

# On the approximation of minimum cost homomorphism to bipartite graphs

Monaldo Mastrolilli, Arash Rafiey *

*IDSIA, Switzerland*

ARTICLE INFO

*Article history:*
Received 3 December 2010
Received in revised form 6 April 2011
Accepted 13 May 2011
Available online 22 June 2011

*Keywords:*
Minimum cost homomorphism
Approximation algorithm
Min–max ordering

ABSTRACT

For a fixed target graph $H$, the minimum cost homomorphism problem, MinHOM($H$), asks, for a given graph $G$ with integer costs $c_i(u)$, $u \in V(G)$, $i \in V(H)$, and an integer $k$, whether or not there exists a homomorphism of $G$ to $H$ of cost not exceeding $k$. When the target graph $H$ is a bipartite graph a dichotomy classification is known: MinHOM($H$) is solvable in polynomial time if and only if $H$ does not contain bipartite claws, nets, tents and any induced cycles $C_{2k}$ for $k \geq 3$ as an induced subgraph.

In this paper, we start studying the approximability of MinHOM($H$) when $H$ is bipartite. First we note that if $H$ has as an induced subgraph $C_{2k}$ for $k \geq 3$, then there is no approximation algorithm. Then we suggest an integer linear program formulation for MinHOM($H$) and show that the integrality gap can be made arbitrarily large if $H$ is a bipartite claw. Finally, we obtain a 2-approximation algorithm when $H$ is a subclass of doubly convex bipartite graphs that has as special case bipartite nets and tents.



## 1. Introduction

For graphs $G$ and $H$, a mapping $f : V(G) \to V(H)$ is a *homomorphism of G to H* if $f(u)f(v)$ is an edge of $H$ whenever $uv$ is an edge of $G$. Let $H$ be a fixed graph. The *homomorphism problem* for $H$, denoted HOM($H$), asks whether or not an input graph $G$ admits a homomorphism to $H$. The *list homomorphism problem* for $H$, denoted ListHOM($H$), asks whether or not an input graph $G$, with lists $L_u \subseteq V(H)$, $u \in V(G)$, admits a homomorphism $f$ to $H$ in which $f(u) \in L_u$, for all $u \in V(G)$. The *minimum cost homomorphism problem* for $H$, denoted MinHOM($H$), asks whether or not an input graph $G$, with integer costs $c_i(u)$, $u \in V(G)$, $i \in V(H)$, and an integer $k$, admits a homomorphism to $H$ of total cost $\sum_{u \in V(G)} c_{f(u)}(u)$ not exceeding $k$. It is easy to see that MinHOM($H$) generalizes ListHOM($H$) which generalizes HOM($H$). Since a homomorphism must take a connected graph to a connected graph, it suffices to consider the problems MinHOM($H$) for connected graphs $H$.

For an undirected graph $H$, the complexity of the problem HOM($H$) has been classified in [9]. If $H$ is a bipartite graph or $H$ has a loop then HOM($H$) is polynomial time solvable and NP-complete otherwise. The problem ListHOM($H$) is polynomial time solvable when $H$ is a bi-arc graph and NP-complete otherwise [3]. In the case of bipartite graphs if the complement of bipartite graph $H$ is a circular arc graph with clique cover two then ListHOM($H$) is polynomial time solvable and NP-complete otherwise. The complement of a bipartite graph $H$ is a circular arc graph if there is a family of circular arcs $A_v$ for $v \in V(H)$ such that $v$ and $v'$ are adjacent if the corresponding arcs $A_v$ and $A_{v'}$ do not intersect (see Fig. 1).

A typical example of a bipartite graph $H$ whose complement is not a circular arc graph is an induced cycle $C_{2k}$, $k \geq 3$. For simplicity in the rest of this paper when we say $C_{2k}$, we mean an induced cycle $C_{2k}$.

The minimum cost homomorphism problems MinHOM($H$) were introduced, in the context of undirected graphs, in [7]; they were motivated by a repair analysis problem in defense logistics. In general, the problem seems to offer a natural

* Corresponding author. Fax: +41 1 778 782 3045.
*E-mail addresses:* monaldo@idsia.ch (M. Mastrolilli), arash@idsia.ch, arashr@sfu.ca (A. Rafiey).

0166-218X/$ – see front matter Crown Copyright © 2011 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2011.05.002

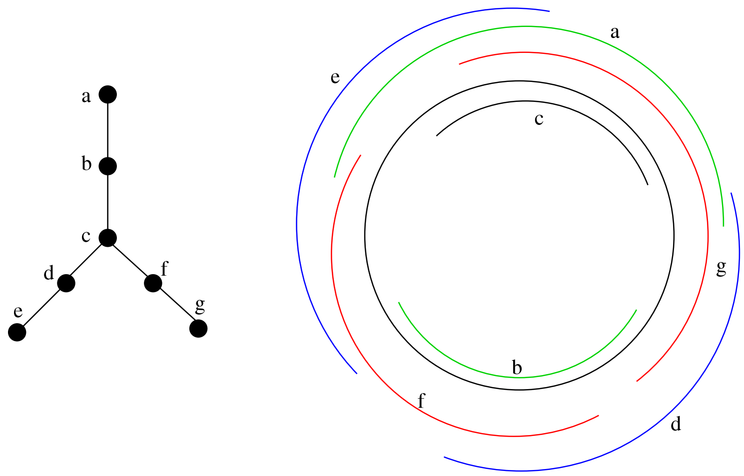


**Fig. 1.** Circular arc representation.

and practical way to model many optimization problems. Special cases include, in addition to the homomorphism and list homomorphism problems, also the optimum cost chromatic partition problem [8,11,12], which itself has a number of well-studied special cases and applications [13,14]. A slightly different version of minimum cost homomorphism was introduced in [1] that was motivated by the application of channel assignment in wireless networks.

If a bipartite graph $H$ is a proper interval bigraph then MinHOM$(H)$ is polynomial time solvable and NP-complete otherwise [6]. A bipartite graph $H = (V, U)$ is called *interval* bigraph if the vertices in $V, U$ are represented by intervals on the real line and $uv$; $u \in U$, $v \in V$, is an edge of $H$ if their corresponding intervals intersect. If the intervals correspond to $U$ are inclusion free and the intervals correspond to $V$ are inclusion free then $H$ is called a *proper interval* bigraph. We say a bipartite graph $H = (U, V)$ has a *min–max ordering* if there are ordering $u_1, u_2, \ldots, u_p$ of $U$ and ordering $v_1, v_2, \ldots, v_q$ of $V$ such that if $u_i v_j$, $u_r v_s$ are edges of $H$ then $u_{min\{i,r\}} v_{min\{j,s\}}$ and $u_{max\{i,r\}} v_{max\{j,s\}}$ are edges of $H$. We say the ordering $u_1, u_2, \ldots, u_p, v_1, v_2, \ldots, v_q$ is a min–max ordering of $H$. The following theorems were proved in [6].

**Theorem 1.1.** *A bipartite graph $H$ is a proper interval bigraph if and only if it admits a min–max ordering.*

**Theorem 1.2.** *The bipartite graph $H$ has min–max ordering if and only if $H$ does not contain bipartite claw, bipartite net, bipartite tent and any cycle $C_{2k}$, $k \geq 3$ as an induced subgraph.*

**Theorem 1.3.** *Let $H$ be a bipartite graph. If $H$ admits a min–max ordering then* MinHOM$(H)$ *is polynomial time solvable. Otherwise,* MinHOM$(H)$ *is NP-complete.*

The class of bipartite graphs $H$, where MinHOM$(H)$ is solvable in polynomial time is a subset of whole bipartite graphs. This serves as an indication that one should relax the requirements to face these problems.

A natural way is to require only an approximate solution—one that is not optimal, but is within a small factor $C > 1$ of optimal. More specifically, a $C$-approximation algorithm is a polynomial time algorithm that produces a solution with an objective value at most $C$ times the optimal value. Sometimes $C$ is called the (worst-case) performance guarantee of the algorithm. We formulate this relaxation in the following problem.

**Problem 1.4.** For a fixed bipartite graph $H$ and an input bipartite graph $G$ together with the costs, is there a $C$-approximation ($C$ is a constant number) algorithm for MinHOM$(H)$?

In this paper, we study Problem 1.4 for bipartite graphs. In Section 2, we consider bipartite graphs $H$ that there is no approximation algorithm for MinHOM$(H)$, in particular bipartite graphs $H$ that contain $C_{2k}$, $k \geq 3$ as an induced subgraph. In Section 3, we suggest an integer linear program formulation ILP for the minimum cost homomorphism problem. Moreover, in Section 3 we deal with bipartite graphs $H$ that contain bipartite claw as an induced subgraph and we show that the integrality gap between the optimal solution and the solution of the suggested linear program can be made arbitrarily large. In Section 4, we obtain a 2-approximation algorithm for a class of bipartite graphs that includes bipartite net and bipartite tent as special cases, by rounding the linear program relaxation LP of ILP. This class is a subclass of the doubly convex bipartite graphs.

## 2. MinHOM$(H)$ when $H$ contains $C_{2k}$ as an induced subgraph

We observe the following. If ListHOM$(H)$ is NP-complete then we show that there is no approximation algorithm for MinHOM$(H)$. Indeed, from an instance of the ListHOM$(H)$ we obtain an instance of MinHOM$(H)$ as follows. For a vertex $u$

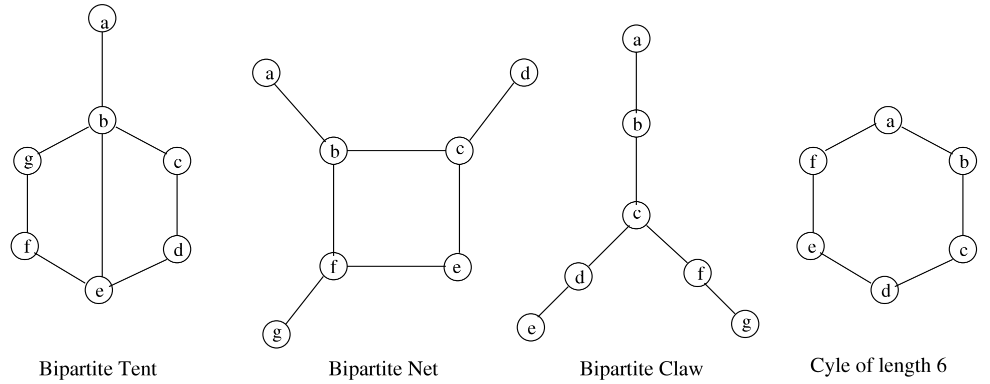


**Fig. 2.** Obstruction to min–max ordering.

of input graph $G$, if $i \in V(H)$ is in $L_u$, list of $u$, then we set the cost of mapping $u$ to $i$ to zero otherwise the cost of mapping $u$ to $i$ is $|V(G)|$. This way we obtain an instance of MinHOM($H$). Now we have one of the following:

- If there is a list homomorphism from $G$ to $H$ then there is a homomorphism from $G$ to $H$ of cost zero.
- If there is no list homomorphism from $G$ to $H$ then the cost of any homomorphism from $G$ to $H$ is at least $|V(G)|$.

This implies that either the minimum cost homomorphism from $G$ to $H$ has value zero or it has value at least $|V(G)|$ and it is hard to distinguish which case happens. We conclude that the class of bipartite graphs $H$ for which there is a constant approximation algorithm for MinHOM($H$), is a subset of the class of bipartite graph whose complement is a circular arc graph with clique cover two.

Since the complement of any induced cycle $C_{2k}$, $k \geq 3$ is not a circular arc graph with clique cover two (see [4]), we have the following proposition.

**Proposition 2.1.** *If the target bipartite graph H contains $C_{2k}$, $k \geq 3$ as an induced subgraph then there is no approximation algorithm for* MinHOM($H$).

Now it remains to deal with the other three obstructions of min–max ordering depicted in Fig. 2. For bipartite tent and bipartite net we show that they are special cases of a class of bipartite graphs $H$ where there is a 2-approximation algorithm for MinHOM($H$). For the bipartite claw we present a large integrality gap that might be considered as a hint that no constant approximation for MinHOM($H$) when $H$ contains a bipartite claw as an induced subgraph. We leave this as an open question.

## 3. An integer linear program formulation for MinHOM(*H*)

Consider digraph $D$ with a source vertex $s$ and sink vertex $t$. Each arc $ij$ of $D$ has a weight denoted by $w_{ij}$. The minimum cut problem is partitioning the vertices $V(D)$ into two sets $S$ and $T = V(D) - S$ with $s \in S$ and $t \in T$, such that the sum of the weights of the arcs from $S$ to $T$ is minimized. The weight of the cut $(S, T)$ is the sum of the weights of the arcs from $S$ to $T$. There is an equivalent linear program formulation of the problem as follows.

For every vertex $a$ of digraph $D$ we define variable $0 \leq X_a \leq 1$. If there is an arc from $a$ to $b$ in $D$ set $Z_{a,b} \geq X_a - X_b$.

We want to minimize

$$\sum_{Z_{a,b}>0} Z_{a,b} w_{a,b}$$

with respect to $X_s = 1$ and $X_t = 0$.

It is known that the constraint matrix of the above linear program is totally unimodular and hence the LP provides an integral solution. Now we explain how to relate the minimum cost homomorphism to a minimum cut in a network by starting when $H$ has a *min–max ordering* and then we generalize it to arbitrary bipartite graphs.

Let $H = (A, B)$ be a bipartite graphs with vertices $a_1, a_2, \ldots, a_p \in A$ and vertices $b_1, b_2, \ldots, b_q \in B$ such that $a_1, a_2, \ldots, a_p, b_1, b_2, \ldots, b_q$ is a min–max ordering. Let $\ell(i)$ be the first index such that $a_i b_{\ell(i)}$ is an edge of $H$. Let $r(i)$ be the first index such that $b_i a_{r(i)}$ is an edge of $H$.

We assume that $H$ is connected and hence every vertex in $H$ has at least one neighbor.

Let $G = (U, V)$ be an input graph with the costs. Note that $G$ should be bipartite, and we may assume that the vertex set $U$ is mapped to $A$ and the vertex set $V$ is mapped to $B$, and we want to find a minimum cost homomorphism from $G$ to $H$ of this type.

We construct network $D$ as follows : the vertices of $D$ are pairs $(u, a_i)$ for $u \in U$ and $a_i \in A$, $1 \leq i \leq p$ and $(v, b_i)$ for $v \in V$ and $b_i \in B$, $1 \leq i \leq q$. There are two extra vertices $s$ and $t$. We add the following arcs to $D$:

- for every vertex $u \in U$, an arc $(u, a_i)$ to $(u, a_{i+1})$ with weight $c(u, a_i)$, $1 \leq i \leq p - 1$, and an arc from $(u, a_{i+1})$ to $(u, a_i)$ with weight $\infty$,
- an arc from $s$ to $(u, a_1)$ of weight $\infty$,

- an arc from $(u, a_p)$ to $t$ of weight $c(u, a_p)$,
- for every vertex $v \in V$, an arc $(v, b_i)$ to $(v, b_{i+1})$ with weight $c(v, b_i)$, $1 \le i \le q-1$, and an arc from $(v, b_{i+1})$ to $(v, b_i)$ with weight $\infty$,
- an arc from $s$ to $(v, b_1)$ of weight $\infty$,
- an arc from $(v, b_q)$ to $t$ of weight $c(v, b_q)$,
- for every edge $uv$ of $G$, an arc of weight $\infty$ from $(u, a_i)$ to $(v, b_{\ell(i)})$ and,
- for every edge $uv$ of $G$, an arc of weight $\infty$ from $(v, b_i)$ to $(u, a_{r(i)})$.

By a similar argument as in [6] one can show that the minimum cut in $D$ corresponds to a minimum cost homomorphism from $G$ to $H$. Consider a cut $(S, T)$ of $D$ with weight less than $\infty$, such that $s \in S$ and $t \in T$. Indeed, if an arc $(u, a_i)(u, a_{i+1})$ belongs to the cut then we map $u$ to $a_i$ and if $(v, b_j)(v, b_{j+1})$ is an arc of the cut we map $v$ to $b_j$. Observe that if $(u, a_i) \in S$ and $(u, a_{i+1}) \in T$ then for every $j > i$, $(u, a_j)$ is in $T$ as otherwise there would be an arc of weight infinity from $S$ to $T$ and hence the weight of the cut would be $\infty$. This implies that if the weight of any cut in $D$ is less than $\infty$ then we cut only one of the arcs $(u, a_i)(u, a_{i+1})$, $1 \le i \le p$ and only one of the arcs $(v, b_i)(v, b_{i+1})$, $1 \le i \le q$. On the other hand, if homomorphism $f : V(G) \to V(H)$ assigns vertex $u \in U$ to $a_i$ of $H$ then we put all the vertices $(u, a_1), (u, a_2), \ldots, (u, a_i)$ into $S$ and all the vertices $(u, a_{i+1}), (u, a_{i+2}), \ldots, (u, a_p)$ to $T$. If $f(v) = b_j$ for $v \in V$ then we add all the nodes $(v, b_j)$, $j \le i$ to $S$. Finally, we add $s$ to $S$ and $t$ to $T$.

If $H$ has a min–max ordering then the LP program would give an integral solution, the optimal solution corresponds to a minimum cut and we obtain an optimal solution. If $H$ does not admit a min–max ordering then MinHOM($H$) is NP-complete [6].

If $H$ has no min–max ordering then we add new edges to $H$ in order to obtain a min–max ordering. Now we construct network $D'$ with $G$ and (new) $H$.

If $a_ib_j$ is a new edge in $H$ then for every edge $uv$ of $G$ we add

$$X_{u,a_i} - X_{u,a_{i+1}} + X_{v,b_j} - X_{v,b_{j+1}} \le 1. \tag{1}$$

The objective function remains the same. Observe that in the LP solution the weight of the cut is less than $\infty$. Now since there is an arc with weight $\infty$ from $(u, a_{i+1})$ to $(u, a_i)$, $1 \le i \le p$, we have $X_{u,a_i} \ge X_{u,a_{i+1}}$. Also there is an arc with weight $\infty$ from $(v, b_{i+1})$ to $(v, b_i)$, $1 \le i \le q$ and hence $X_{v,b_i} \ge X_{v,b_{i+1}}$.

If the LP program provides an integral solution, corresponds to minimum cut, then we define a homomorphism $f$ from $G$ to new $H$ in a same way as explained before. According to the constraints in the LP program $a_ib_j$ is an old edge in $H$ as otherwise $X_{u,a_i} - X_{u,a_{i+1}} + X_{v,b_j} - X_{v,b_{j+1}} > 1$; violating a constraint of the LP. Therefore, $f$ is a homomorphism from $G$ to $H$ with the old edges. If the LP program does not provide an integral solution then we explain in the next section how to round the values provided by the LP and obtain a homomorphism from $G$ to $H$, by losing the optimality of the solution.

### *3.1. Integrality gap of the LP program relaxation*

The following result shows that even for a bipartite claw the integrality gap of the suggested LP can be made arbitrarily large.

**Lemma 3.1.** *If the target bipartite graph H has the bipartite claw as an induced subgraph then the integrality gap of the LP described in Section 2 can be arbitrarily large.*

**Proof.** In Fig. 3, the input graph $G$ is a path $v_1, u_1, v_2, u_2, \ldots, v_{n-1}, u_{n-1}$ and the target graph $H = \{12, 23, 34, 45, 36, 76\}$ is a bipartite claw. Note that we need to add new edge 56 to $H$ in order to obtain a min–max ordering. We have the following cost function. For $1 \le i \le n-1$, $c_1(u_i) = c_2(v_i) = M$ and $c_3(u_i) = 2nM$. We have $c_4(v_1) = c_7(u_{n-1}) = 2nM$. In any other case, the cost is 1. According to the LP solution shown in Fig. 3, for every edge $v_iu_i$, $1 \le i \le n-1$, we have $X_{v_i,2} = 1$, $X_{v_i,6} = 1 - i/n$ and $X_{u_i,1} = 1$, $X_{u_i,5} = 1 - 1/n$, $X_{u_i,7} = 1 - (i+1)/n$. Therefore, for edge $u_iv_i$, $1 \le i \le n-2$, $X_{v_i,6} + X_{u_i,5} - X_{u_i,7} = 1$ and for edge $u_iv_{i+1}$ we have $X_{v_{i+1},6} + X_{u_i,5} - X_{u_i,7} = 1 - (i+1)/n + 1 - 1/n - (1 - (i+1)/n) = 1 - 1/n < 1$. Also we have $X_{v_{n-1},6} + X_{u_{n-1},5} - X_{u_{n-1},7} = 1 - (n-1)/n + 1 - 1/n = 1$. Thus the constraints in Eq. (1) are satisfied. There is a homomorphism $f : V(G) \to V(H)$ that assigns $u_i$ to 1 and $v_i$ to 2 for $1 \le i \le n-1$. Since $c_1(u_i) = c_2(v_i) = M$, $1 \le i \le n-1$, the cost of $f$ is $2(n-1)M$. We claim that any other homomorphism $g : V(G) \to V(H)$ has cost at least $2(n-1)M$. If $g$ maps $v_1$ to 6 then it must map $u_1$ to 7 as otherwise the cost of $g$ would be at least $2nM$. Now $g$ must map $v_2$ to 6 and again $g$ maps $u_2$ to 7 and if we continue along the path, at the end $g$ maps $u_{n-1}$ to 7 and hence the cost of $g$ is at least $2nM$. If $g$ maps $v_1$ to 4 then the cost of $g$ would be at least $2nM$. If $g$ maps $v_1$ to 2 then it must map $u_1$ to 1 as otherwise the cost of $g$ would be at least $2nM$, and now $g$ must map $v_2$ to 2 and hence $g$ must map every edge of $G$ to edge 12 of $H$ as otherwise the cost of $g$ would be at least $2nM$. Therefore, the cost of $g$ is at least $2(n-1)M$. For every $1 \le i \le n-2$ the value contributed by vertex $u_i$ to the objective function of the LP is $(1 - X_{u_i,3})M + X_{u_i,5} - X_{u_i,7} + X_{u_i,7} = M/n + 1 - 1/n$. The value contributed by vertex $u_{n-1}$ to the objective function of the LP is $M/n + 1 - 1/n$.

For every $2 \le i \le n-1$ the value contributed by vertex $v_i$ to the objective function of the LP is $(1 - X_{v_i,4})M + X_{v_i,4} - X_{v_i,6} + X_{v_i,6} = M/n + 1 - 1/n$. The value contributed by vertex $v_1$ to the objective function of the LP is $M/n + 1 - 1/n$. Therefore, the value of the solution provided by LP is $2(n-1)M/n + 2(n-1)(1 - 1/n) \le 2M + 2(n-1)$. By setting $M = n-1$, we have $OPT/LP > (n-1)/2$. □

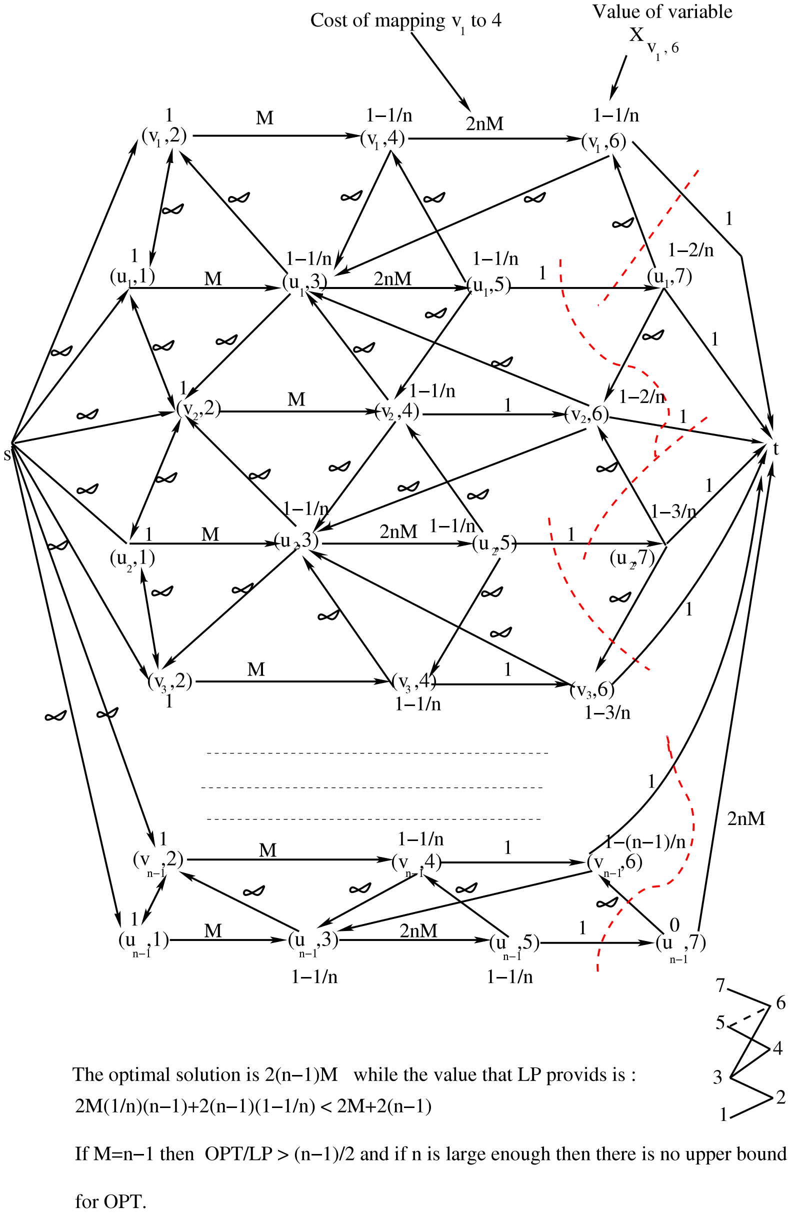


**Fig. 3.** Integrality gap for bi-claw.

Note that in the next section we show that there is a 2-approximation algorithm for minimum cost homomorphism to bipartite tent and bipartite net (see Fig. 2).

## 4. A 2-approximation algorithm

In this section, we consider a class of bipartite graphs $H$ that includes as special cases the bipartite tent and bipartite net (see Fig. 2). The latter implies that MinHOM($H$) is NP-complete [6] and we provide a 2-approximation algorithm for MinHOM($H$).

We say a bipartite graph $H = (A, B)$ has a *min ordering* if there are ordering $a_1, a_2, \ldots, a_p$ of $A$ and ordering $b_1, b_2, \ldots, b_q$ of $B$ such that if $a_i b_j$, $a_r b_s$ are edges of $H$ then $a_{min\{i,r\}} b_{min\{j,s\}}$ is an edge of $H$. We say the ordering $a_1, a_2, \ldots, a_p, b_1, b_2, \ldots, b_q$ is a min ordering of $H$ (see [5]).

Bipartite graph $H$ is called *double convex* bipartite if there is an ordering of the vertices in $A$ and there is an ordering of the vertices in $B$ such that every vertex in $A$ is adjacent to consecutive vertices in $B$ and every vertex in $B$ is also adjacent to consecutive vertices in $A$; the neighborhood of each vertex is an interval [15].

Observe that if $H$ admits a min–max ordering then $H$ admits a min ordering and also $H$ is a doubly convex bipartite graph. There are bipartite graphs that do not admit min–max ordering but they admit a min ordering and are doubly convex bipartite graphs. For example, bipartite tent and bipartite net (see Fig. 2) admit min ordering and are doubly convex bipartite graphs, while bipartite claw admits a min ordering but it is not a doubly convex bipartite graph. For every $k \geq 3$, $C_{2k}$ does not admit a min ordering.

**Theorem 4.1.** *If bipartite graph $H = (A, B)$ admits a min ordering such that the neighborhood of each vertex is an interval then there is a 2-approximation algorithm for* MinHOM($H$).

**Proof.** Let $\pi = a_1, a_2, \ldots, a_p, b_1, b_2, \ldots, b_q$ be a min ordering such that the neighborhood of each vertex is an interval. If $\pi$ is a min–max ordering then we construct the network $D$ as explained in Section 3 and we obtain an optimal solution in polynomial time. Otherwise, there are vertices $a_i, a_j, b_r, b_s$ with $i < j, r < s$ such that $a_ib_s$ and $a_jb_r$ are edges and $a_jb_s$ is not an edge. Note that $a_ib_r$ is an edge since $\pi$ is min ordering. Since $\pi$ is a min ordering, for every $1 \leq i \leq p$ the first neighbor of $a_i$; is not before the first neighbor of $a_{i-1}$, and for every $1 \leq j \leq q$ the first neighbor of $b_j$ is not before the first neighbor of $b_{j-1}$. We need to add a set of new edges to $H$ in order to obtain a min–max ordering.

Observe that $\pi$ is min ordering. Without loss of generality, we may assume that $H$ is connected. If $\pi$ is not min–max ordering then there are some $a_k$ and $a_{k'}$, $k < k'$ such that the last neighbor of $a_k$, say $b_s$ is after the last neighbor of $a_{k'}$, and hence we need to add edges from $a_{k'}$ to all the neighbors of $a_k$ that are after the last neighbor of $a_{k'}$. Now vertex $b_{s+1}$ does not have any neighbor $a_r$ with $k' \leq r$. As otherwise since $H$ is connected, $a_r$ should have a neighbor to some vertex before $b_{s+1}$ and hence $a_r$ would be adjacent to $b_s$ (by interval property) and consequently $b_s$ and $b_{k'}$ are adjacent, a contradiction. Therefore, we should also add an edge from $a_{k'}$ to $b_{s+1}$. By continuing this argument, we need to add an edge from $a_{k'}$ to any vertex after $b_s$. This allows us to obtain a way of adding new edges to $H$ such that at the end $\pi$ is min–max ordering. At each step we add a set of new edges to $H$.

1. Let $1 \leq k' \leq p$ be the smallest index (from left to right) that there is some $k > k'$ such that the last neighbor of $a_k$ (according to the ordering) is before the last neighbor of $a_{k'}$, and $k$ is minimum. Now let $b_s$ be the last neighbor of $a_k$.
2. Let $1 \leq t' \leq q$ be the smallest index (from left to right) that there is some $t > t'$ such that the last neighbor of $b_t$ is before the last neighbor of $b_{t'}$, and $t$ is minimum. Now let $a_r$ be the last neighbor of $b_t$.

If there are $k', k$ or $t, t'$ in the current step, we add all the new edges $a_ib_j$ that $i \geq k, j \geq s+1$, and all the new edges $a_ib_j$ that $i \geq t, j \geq r+1$ to $H$.

Observe that $\pi$ is a min–max ordering for new $H$. We construct network $D'$ with input graph $G = (U, V)$ and new $H$. Now we write the LP program for network $D'$ and we add extra constraints. The set of constraints added here is slightly different from the LP in Section 3. We show that under these new constraints we have an equivalent formulation of the MinHOM($H$).

At step $\ell$ of obtaining new $H$, for every edge $uv$ of $G$ if there are $k, s$; according to 1, or there are $t, r$; according to 2, then we add the following extra constraints respectively.

$$X_{u,a_k} + X_{v,b_{s+1}} \leq 1, \qquad X_{v,b_t} + X_{u,a_{r+1}} \leq 1$$

The objective function remains the same. Since there are arcs with weight infinity from $(u, a_{i+1})$ to $(u, a_i)$ and from $(v, b_{i+1})$ to $(v, b_i)$, we have the following proposition. □

**Claim 4.2.** *In any optimal fractional solution found by the above LP program, $X_{u,a_i} \geq X_{u,a_{i+1}}$ and $X_{v,b_i} \geq X_{v,b_{i+1}}$*

**Claim 4.3.** *If there is an integer solution for the above LP, then there is homomorphism from G to H that does not map any edge of G to a new edge of H.*

**Proof of the Claim.** We define homomorphism $f : V(G) \to V(H)$ in same way as we defined in Section 3. Indeed, if edge $uv$ of $G$ is mapped to a new edge $a_ib_j$ of $H$ such that $i \geq k, j \geq s+1$ then as $X_{u,a_k} \geq X_{u,a_i}$ and $X_{v,b_{s+1}} \geq X_{v,b_j}$ we have $X_{u,a_k} = X_{v,b_{s+1}} = 1$ and hence the constraint $X_{u,a_k} + X_{v,b_{s+1}} \leq 1$ is violated. Similarly, we get a contradiction when $i \geq t$, $j \geq r+1$. □

Each constraint in the LP has 2 variables and therefore they satisfy the conditions in [10] (Section 3). The results in [10] imply a 2-approximation algorithm for the addressed problem. Alternatively following [2] the simple arguments below show that the integrality gap is upper bounded by 2.

Let *OPTLP* be the optimal solution obtained by an LP. We obtain an integral solution as follows. We choose a variable $X$ uniformly at random between $[\frac{1}{2}, 1]$ and we do the following: for every $u \in V(G)$ and $1 \leq i \leq p$ if $X \leq X_{u,a_i}$ then we round $X_{u,a_i}$ to 1 otherwise $X_{u,a_i}$ is set to zero. For every $v \in V(G)$ and $1 \leq j \leq q$ if $X \leq X_{v,b_j}$ then we round $X_{v,b_j}$ to 1 otherwise $X_{v,b_j}$ is set to zero. This guarantees that no edge $uv$ of $G$ is mapped a new edge of $H$.

Let $E[Z_{u,i}] \geq X_{u,a_i} - X_{u,a_{i+1}}$, and $E[Z_{v,j}] \geq X_{v,b_j} - X_{v,b_{j+1}}$. Now the expected value that an edge is being cut is as follows: $E[Z_{u,i}] = Pr[X_{u,a_i} = 1 \wedge X_{u,a_{i+1}} = 0]$ and $E[Z_{v,j}] = Pr[X_{v,b_j} = 1 \wedge X_{v,b_{j+1}} = 0]$.

When $X_{u,a_{i+1}} < \frac{1}{2}$, $Pr[X_{u,a_i} = 1] = \frac{X_{u,a_i} - 1/2}{2} = 2X_{u,a_i} - 1$. Hence $Pr[X_{u,a_i} = 1 \wedge X_{u,a_{i+1}} = 0] = 2X_{u,a_i} - 1 \leq 2(X_{u,a_i} - X_{u,a_{i+1}})$. If $X_{u,a_{i+1}} \geq \frac{1}{2}$ then $Pr[X_{u,a_i} = 1 \wedge X_{u,a_{i+1}} = 0] = 2(X_{u,a_i} - X_{u,a_{i+1}})$. Therefore, we have

$$E[Z_{u,i}] = Pr[X_{u,a_i} = 1 \wedge X_{u,a_{i+1}} = 0] \leq 2(X_{u,a_i} - X_{u,a_{i+1}}),$$

Similarly, we have

$$E[Z_{v,j}] = Pr[X_{v,b_j} = 1 \wedge X_{v,b_{j+1}} = 0] \leq 2(X_{v,b_j} - X_{v,b_{j+1}})$$

Therefore, there is a way of rounding the variables to obtain a solution that is at most twice the value of the OPTLP. Since the $OPTLP \leq OPT$, we obtain a 2-approximation ratio. □

## 5. Future work

It would be interesting to settle the dichotomy for the approximation of MinHOM($H$).

**Open Problem 5.1.** *Characterize bipartite graphs $H$ that there is a constant approximation algorithm for* MinHOM($H$)*?*

## Acknowledgment

The authors were supported by the Swiss National Science Foundation project 200020-122110/1, "Approximation Algorithms for Machine Scheduling III".

## References

[1] G. Aggarwal, T. Feder, R. Motwani, A. Zhu, Channel assignment in wireless networks and classification of minimum graph homomorphism, Electronic Colloquium on Computational Complexity (ECCC) TR06-040 (2006).
[2] D. Bertsimas, C.P Teo, Rakesh Vohra, On dependent randomized rounding algorithms, Oper. Res. Lett. 24 (3) (1999) 105–114.
[3] T. Feder, P. Hell, J. Huang, Bi-arc graphs and the complexity of list homomorphisms, J. Graph Theory. 42 (2003) 61–80.
[4] T. Feder, P. Hell, J. Huang, List homomorphisms and circular arc graphs, Combinatorica. 19 (1999) 487–505.
[5] T. Feder, P. Hell, J. Huang, A. Rafiey, Interval graphs, adjusted interval digraphs, and reflexive list homomorphisms, Discrete Appl. Math., in press (doi:10.1016/j.dam.2011.04.016).
[6] G. Gutin, P. Hell, A. Rafiey, A. Yeo, A dichotomy for minimum cost graph homomorphisms, European J. Combin. 29 (2008) 900–911.
[7] G. Gutin, A. Rafiey, A. Yeo, M. Tso, Level of repair analysis and minimum cost homomorphisms of graphs, Discrete Appl. Math. 154 (2006) 881–889.
[8] M.M. Halldorsson, G. Kortsarz, H. Shachnai, Minimizing average completion of dedicated tasks and interval graphs, in: Approximation, Randomization, and Combinatorial Optimization Berkeley, Calif, 2001, in: Lecture Notes in Computer Science, vol. 2129, Springer, Berlin, 2001, pp. 114–126.
[9] P. Hell, J. Nešetřil, On the complexity of $H$-colouring, J. Combin. Theory B. 48 (1990) 92–110.
[10] D.S. Hochbaum, Instant recognition of half integrality and 2-approximation, in: Procdeedings of APPROX98. Aalborg Denmark, July 1998.
[11] K. Jansen, Approximation results for the optimum cost chromatic partition problem, J. Algorithms. 34 (2000) 54–89.
[12] T. Jiang, D.B. West, Coloring of trees with minimum sum of colors, J. Graph Theory. 32 (1999) 354–358.
[13] L.G. Kroon, A. Sen, H. Deng, A. Roy, The optimal cost chromatic partition problem for trees and interval graphs, in: Graph-Theoretic Concepts in Computer Science Cadenabbia, 1996, in: Lecture Notes in Computer Science, vol. 1197, Springer, Berlin, 1997, pp. 279–292.
[14] K. Supowit, Finding a maximum planar subset of a set of nets in a channel, IEEE Trans. Computer-Aided Des. 6 (1987) 93–94.
[15] D. West, Introduction to Graph Theory, Prentice Hall, Upper Saddle River, 1996.