Theoretical Computer Science 526 (2014) 90–96

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Approximation algorithms for node deletion problems on bipartite graphs with finite forbidden subgraph characterization

Mrinal Kumar [a], Sounaka Mishra [a], N. Safina Devi [a], Saket Saurabh [b,*]

[a] Indian Institute of Technology Madras, India
[b] Institute of Mathematical Sciences, India

ARTICLE INFO

Article history:
Received 30 April 2013
Received in revised form 24 December 2013
Accepted 15 January 2014
Communicated by V.Th. Paschos

Keywords:
Graph deletion problems
Approximation algorithms
Primal–dual

ABSTRACT

In this paper, we develop approximation algorithms for a few node deletion problems when the input is restricted to be a bipartite graph. We look at node deletion problems for non-trivial properties which can be characterized by forbidden structure which has a bounded intersection with both the bipartitions. The approximation factors obtained directly depend upon the size of the largest such intersection. Special instances of this general problem include problems such as the Minimum Chain Vertex Deletion, Minimum Dissociation Vertex Deletion, Minimum Bipartite Claw Vertex Deletion, Minimum Bi-complement Vertex Deletion and Minimum Bipartite Threshold Vertex Deletion problems. The algorithms are based upon the techniques of linear programming and iterative rounding. We also use the node deletion algorithms to marginally improve the trivial approximation factor for complementary problem of determining the size of the maximum sized vertex induced subgraph lying in the given graph class and prove the *APX*-completeness of all of these problems.

1. Introduction

Given an undirected graph G and any graph class $\mathcal{G}$, two natural optimization problems can be associated with it. What is the minimum number of vertices need to be deleted from G to obtain a graph in $\mathcal{G}$ and what is the largest induced subgraph G' of G such that $G' \in \mathcal{G}$? A seminal result by Lewis and Yannakakis [8] shows that a large number of such graph optimization problems are *NP*-complete. To state their results we need the following definitions. A graph property Π is an isomorphism-closed set of graphs. A graph property Π is non-trivial if there exists an infinite family of graphs satisfying Π and an infinite family not satisfying Π and it is hereditary if $G \in \Pi$ implies that every induced subgraph of G is also in Π. They proved that if Π is a non-trivial hereditary property, then the vertex-deletion problem for Π is *NP*-hard and if Π can be tested in polynomial time, then such vertex-deletion problem for Π is *NP*-complete. In 1994, Lund and Yannakakis [10] proved that in general these kinds of minimum vertex deletion problems are *APX*-hard. They also complemented this with an approximation algorithm for the property Π that has finite forbidden set characterization. A property Π has a finite forbidden set characterization if there exists a finite set $\mathcal{H}$ of graphs such that G has property Π if and only if no element of $\mathcal{H}$ is an induced subgraph of G. A trivial greedy algorithm for this class of vertex deletion problems gives a solution with approximation factor c, where c is the number of vertices in a largest graph in $\mathcal{H}$. Apart from this generic result presented in [10] there has also been a lot of work done when $\mathcal{G}$ is some specific graph classes

* Corresponding author.
E-mail addresses: mrinalk@cse.iitm.ac.in (M. Kumar), sounak@iitm.ac.in (S. Mishra), safina@smail.iitm.ac.in (N. Safina Devi), saket@imsc.res.in (S. Saurabh).

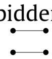
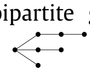
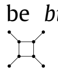
0304-3975/$ – see front matter
http://dx.doi.org/10.1016/j.tcs.2014.01.019

like independent sets, forests and bipartite graphs. For example MINIMUM VERTEX COVER – when $\mathcal{G}$ is independent sets and MINIMUM FEEDBACK VERTEX SET – when $\mathcal{G}$ is forests admit factor 2 approximating algorithm [13,2]. When $\mathcal{G}$ is bipartite graphs, we get MINIMUM ODD CYCLE TRANSVERSAL which is known to admit $O(\log n)$ factor approximation [13]. On the other hand, all the corresponding maximization problems MAXIMUM INDEPENDENT SET, MAXIMUM INDUCED FOREST and MAXIMUM INDUCED BIPARTITE SUBGRAPH are known to be hard to approximate within a factor of $n^{1-\epsilon}$, for any $\epsilon > 0$ [13,5].

In this paper, we will talk about non-trivial hereditary properties which are characterized by a finite set $\mathcal{H}$ of forbidden induced subgraphs which have the following property on bipartite graphs. The number of vertices that induces each graph structure $H \in \mathcal{H}$ is at most $2k$, for a fixed k and for any bipartite graph $G = (A \cup B, E)$, if G has an induced subgraph $G[S]$ isomorphic to a graph $H \in \mathcal{H}$, then it is such that $|A \cap S| = |B \cap S| = k$. We refer to such properties as $\Pi_{k,k}$. We also restrict ourselves to the case where our input is always a bipartite graph. We call the node deletion problem for such a property as MINIMUM $\Pi_{k,k}$ VERTEX DELETION and the complementary problem of finding the maximum sized vertex induced subgraph belonging to the class $\Pi_{k,k}$ as MAXIMUM $\Pi_{k,k}$ SUBGRAPH. A slightly more generalized case of the above case will be when the forbidden structures for the class Π do not have equal intersections with the two bipartitions but contain at most k_1 vertices from one bipartition and at most k_2 vertices from the other one, for some constants k_1, k_2. In this case, we write the class Π as Π_{k_1,k_2}. The corresponding node deletion problem will be called MINIMUM Π_{k_1,k_2} VERTEX DELETION and the maximization problem will be called MAXIMUM Π_{k_1,k_2} SUBGRAPH. The formal definitions of the two problems for this class of graphs are given as follows.

- MINIMUM Π_{k_1,k_2} VERTEX DELETION: Given a bipartite graph $G = (V, E)$ and a weight function $w : V \to \mathbb{Q}^+$, find a set $S \subseteq V$ of minimum weight $w(S) = \sum_{v \in S} w(v)$ such that $G[V \setminus S]$ is in class Π_{k_1,k_2}.
- MAXIMUM Π_{k_1,k_2} SUBGRAPH: Given a bipartite graph $G = (V, E)$ and a weight function $w : V \to \mathbb{Q}^+$, find a set $S \subseteq V$ of maximum weight $w(S) = \sum_{v \in S} w(v)$ such that $G[S]$ is in class Π_{k_1,k_2}.

Let Π be any non-trivial graph property with a finite forbidden subgraph characterization and let c be the size of a largest forbidden subgraph for this property. Then Π restricted to bipartite graphs becomes Π_{k_1,k_2}, where $0 < k_1 < c$ and $0 < k_2 < c$. This motivates us to look for algorithms with better approximation factor for these problems on bipartite graphs. For the ease of presentation, the algorithms in the paper are described considering the case where $k_1 = k_2$, but can be easily extended to the case of $k_1 \neq k_2$. Our algorithms in this general setup can be adapted to many particular graph classes on bipartite graphs. A brief enumeration is given below.

1. *Chain graph*: A bipartite graph $G = (A \cup B, E)$ is called a chain graph if there exists an ordering $\langle v_1, v_2, \ldots, v_n \rangle$ of the vertices in A such that $N(v_1) \subseteq N(v_2) \subseteq \cdots \subseteq N(v_n)$. Chain graphs were introduced in [7,14] and proved to be *NP*-complete [14]. This class of bipartite graphs has many characterizations including a finite forbidden subgraph characterization. It is known that a bipartite graph is chain if and only if G is $\{2K_2\}$-free [14], where  is $2K_2$. This is a special case of $\Pi_{2,2}$. Based on this characterization, we consider the following problems related to chain graphs which are special cases of the above mentioned optimization problems with $\mathcal{H} = \{2K_2\}$.
   - MINIMUM CHAIN VERTEX DELETION: Given a bipartite graph $G = (V, E)$ and a weight function $w : V \to \mathbb{Q}^+$, find a set $S \subseteq V$ of minimum weight $w(S) = \sum_{v \in S} w(v)$ such that $G[V \setminus S]$ is a chain graph.
   - MAXIMUM INDUCED CHAIN SUBGRAPH: Given a bipartite graph $G = (V, E)$ and a weight function $w : V \to \mathbb{Q}^+$, find a set $S \subseteq V$ of maximum weight $w(S) = \sum_{v \in S} w(v)$ such that $G[S]$ is a chain graph.
2. *Dissociation number*: Dissociation number of a graph is the minimum number of vertices to be deleted such that the remaining graph has maximum degree 1. It is known that finding the dissociation number of a graph G is *NP*-complete even when restricted to bipartite graphs [8]. It is known that a graph G has maximum degree at most 1 if and only if G is $\{C_3, P_3\}$-free, where C_3 and P_3 are simple cycle and simple path with 3 vertices, respectively. Therefore, a bipartite graph G is of maximum degree at most 1 if and only if G is $\{P_3\}$-free. Hence, this is a case of $\Pi_{1,2}$. We consider the approximability of the following optimization problems associated with dissociation number of a bipartite graph which are special cases of MINIMUM Π_{k_1,k_2} VERTEX DELETION and MAXIMUM Π_{k_1,k_2} SUBGRAPH with $\mathcal{H} = \{P_3\}$.
   - MINIMUM DISSOCIATION VERTEX DELETION: Given a bipartite graph $G = (V, E)$ and a weight function $w : V \to \mathbb{Q}^+$, find a set $S \subseteq V$ of minimum weight $w(S) = \sum_{v \in S} w(v)$ such that degree of each vertex in $G[V \setminus S]$ is at most 1.
   - MAXIMUM INDUCED DISSOCIATION SUBGRAPH: Given a bipartite graph $G = (V, E)$ and a weight function $w : V \to \mathbb{Q}^+$, find a set $S \subseteq V$ of maximum weight $w(S) = \sum_{v \in S} w(v)$ such that $G[S]$ is of maximum degree at most 1.
3. *Bipartite claw-free graph*: In literature, the complete bipartite graph $K_{1,3}$ is known as claw. A bipartite graph is claw-free if it does not contain an induced graph $K_{1,3}$. This is a $\Pi_{1,3}$ property. The optimization problems related to claw-free bipartite graphs given below are known to be *NP*-complete [8].
   - MINIMUM BIPARTITE CLAW VERTEX DELETION: Given a bipartite graph $G = (V, E)$ and a weight function $w : V \to \mathbb{Q}^+$, find a set $S \subseteq V$ of minimum weight $w(S) = \sum_{v \in S} w(v)$ such that $G[V \setminus S]$ is claw-free.
   - MAXIMUM INDUCED BIPARTITE CLAW-FREE SUBGRAPH: Given a bipartite graph $G = (V, E)$ and a weight function $w : V \to \mathbb{Q}^+$, find a set $S \subseteq V$ of maximum weight $w(S) = \sum_{v \in S} w(v)$ such that $G[S]$ is claw-free.
4. *Bi-complement reducible graph*: A bipartite graph is said to be *bi-complement reducible graph* if and only if G is $\{Star_{1,2,3}, Sun_4, P_7\}$-free [9], where  is $Star_{1,2,3}$ and  is Sun_4. Here Sun_4 gives rise to the maximum

intersection with both the sets of a bipartition and hence this is a $\Pi_{4,4}$ property. Based on this finite forbidden subgraph characterization we define the following related problems.

- Minimum Bi-complement Vertex Deletion: Given a bipartite graph $G = (V, E)$ and a weight function $w : V \to \mathbb{Q}^+$, find a set $S \subseteq V$ of minimum weight $w(S) = \sum_{v \in S} w(v)$ such that $G[V \setminus S]$ is a bi-complement reducible graph.
- Maximum Induced Bi-complement Subgraph: Given a bipartite graph $G = (V, E)$ and a weight function $w : V \to \mathbb{Q}^+$, find a set $S \subseteq V$ of maximum weight $w(S) = \sum_{v \in S} w(v)$ such that $G[S]$ is bi-complement reducible graph.

5. *Bipartite threshold graph*: A bipartite graph is said to be a *threshold* graph if and only if G is $\{2K_2, P_4, C_4\}$-free [4]. This is also a case of $\Pi_{2,2}$ like the chain graphs. For this graph class, we consider the following problems.
   - Minimum Bipartite Threshold Vertex Deletion: Given a bipartite graph $G = (V, E)$ and a weight function $w : V \to \mathbb{Q}^+$, find a set $S \subseteq V$ of minimum weight $w(S) = \sum_{v \in S} w(v)$ such that $G[V \setminus S]$ is threshold graph.
   - Maximum Induced Bipartite Threshold Subgraph: Given a bipartite graph $G = (V, E)$ and a weight function $w : V \to \mathbb{Q}^+$, find a set $S \subseteq V$ of maximum weight $w(S) = \sum_{v \in S} w(v)$ such that $G[S]$ is a threshold graph.

The *NP*-completeness of all the above mentioned problems follows from the generic result for node deletion problems on bipartite graphs due to Yannakakis [14]. For unweighted Minimum Π_{k_1,k_2} Vertex Deletion, the greedy algorithm for general graphs [10] (due to the finite forbidden subgraph characterization) gives rise to a $k_1 + k_2$ factor approximation algorithm. However, the approximability of this problem has been improved only for a few cases. Mishra et al. [11] use a primal–dual approach to give a 2-approximation to Minimum Chain Vertex Deletion and a 1.5-approximation to Maximum Induced Chain Subgraph. Also, both the problems are shown to be *APX*-complete. Fujito [6] gives a 2-approximation to Minimum Dissociation Vertex Deletion in general graphs using local ratio technique and Tu and Yang [12] give a 1.57-approximation in cubic graphs. The approximability of the other mentioned problems remain unknown to the best of our knowledge.

Our results are applicable to all of the above mentioned graph classes. We present the algorithms in general and mention the particular approximation factors for all of the above classes. The algorithm for node deletion problems uses the technique of iterative rounding applied to a linear integer programming formulation based on the finite forbidden subgraph characterization of the corresponding graph classes. We first show that a simple rounding of an optimal solution for a linear program gives a $(k+1)$ approximate solution where $2k$ is the number of vertices in the maximum sized forbidden subgraph. We further use iterative rounding and improve this algorithm to obtain an approximation factor k.

In this paper we shall use the following notations. Given a bipartite graph $G = (A \cup B, E)$, we sometime denote the vertex set of G as V $(V = A \cup B)$. We denote a vertex weighted graph as a pair (G, w), where $G = (V, E)$ is a simple graph and $w : V \to \mathbb{Q}^+$. Unless otherwise mentioned, we use $n = |V|$ and $m = |E|$. The weight of a vertex $v \in V$ is denoted by $w(v)$ and weight of a set of vertices $S \subseteq V$ by $w(S) = \sum_{v \in S} w(v)$. P_n is a simple path with n vertices, C_n is a simple cycle with n vertices and $K_{p,q}$ is a complete bipartite graph with p vertices in one bipartition and q vertices in other bipartition.

The organization of the paper is as follows. Section 2 contains the algorithm for Minimum $\Pi_{k,k}$ Vertex Deletion and a simple factor $2 - \frac{1}{k}$ approximation algorithm for the Maximum $\Pi_{k,k}$ Subgraph which is essentially based upon the algorithm for the corresponding node deletion problems. Section 3 consists of results for *APX*-completeness of the problems. We conclude with some further directions and open problems in Section 4.

2. Approximation algorithms

2.1. Minimum $\Pi_{k,k}$ vertex deletion

In this section we design a k-approximation algorithm for Minimum $\Pi_{k,k}$ Vertex Deletion for $k \geqslant 2$ based on iterative rounding. In a given bipartite graph $G = (A \cup B, E)$, let $\mathcal{K} = \{T_1, T_2, \ldots, T_l\}$ be the set of all induced subgraphs of G that are isomorphic to the forbidden graph structures. For convenience, we shall call any forbidden induced subgraph as a T-structure. It can be observed that l, the number of T-structures in G, can be at most $\binom{|A|}{k} + \binom{|B|}{k}$, which is polynomial in n and exponential in k. Each T_i can be represented as a set of $2k$ vertices $\{a_1, a_2, \ldots, a_k, b_1, b_2, \ldots, b_k\}$, where $a_i \in A$ and $b_i \in B$ for each $i = 1, \ldots, k$. This representation is consistent as any set of $2k$ vertices can induce at most one T-structure.

We assume that every vertex v is in some T_i else, we remove v from G. Based on this understanding, for a given graph $G = (V, E)$ and a weight function $w : V \to \mathbb{Q}^+$, Minimum $\Pi_{k,k}$ Vertex Deletion can be formulated as an integer linear program as follows:

$$\begin{array}{llll} ILP(G,w): & \text{Minimize} & \sum_{v\in V} w(v)x_v & \\ & \text{Subject to} & \sum_{v\in T_j} x_v \geqslant 1, & 1 \leqslant j \leqslant l, \\ & & x_v \in \{0,1\}, & \forall v \in V. \end{array}$$

The linear program relaxation $LP(G, w)$ of this problem and its dual $DLP(G, w)$ are as given below.

$$\begin{array}{llll} LP(G,w): & \text{Minimize} & \sum_{v\in V} w(v)x_v & \\ & \text{Subject to} & \sum_{v\in T_j} x_v \geqslant 1, & 1 \leqslant j \leqslant l, \\ & & x_v \geqslant 0, & \forall v \in V. \end{array}$$

$$DLP(G, w): \quad \text{Maximize} \quad \sum_{j=1}^{l} y_j$$
$$\text{Subject to} \quad \sum_{j:\ v \in T_j} y_j \leqslant w(v), \quad \forall v \in V,$$
$$y_j \geqslant 0, \quad 1 \leqslant j \leqslant l.$$

Using this linear program formulation we design an approximation algorithm for the Minimum $\Pi_{k,k}$ Vertex Deletion. To compute the approximation factor we will use the following $\alpha\beta$-relaxed complementary slackness conditions:

Primal: Let $\alpha \geqslant 1$.
For each $v \in V$, either $x_v = 0$ or $\frac{w(v)}{\alpha} \leqslant \sum_{j:\ v \in T_j} y_j \leqslant w(v)$.

Dual: Let $\beta \geqslant 1$.
For each $1 \leqslant j \leqslant l$, either $y_j = 0$ or $1 \leqslant \sum_{v \in T_j} x_v \leqslant \beta$.

Proposition 2.1. *(See [13].) If x and y are primal and dual feasible solutions for $LP(G, w)$ and $DLP(G, w)$, respectively, satisfying the $\alpha\beta$-relaxed complementary slackness conditions stated as above then $\sum_{v \in V} w(v)x_v \leqslant \alpha\beta \sum_{j=1}^{k} y_j$.*

We design a k-approximation algorithm for Minimum $\Pi_{k,k}$ Vertex Deletion for $k \geqslant 2$, based on iterative rounding procedure. In each iteration, we solve $LP(G, w)$ and include the vertices for which the associated variables have value at least $\frac{1}{k}$ and remove these vertices from the graph. In this smaller graph, we also remove the vertices which are not in any T-structure. We continue with this iterative process till the graph is empty or all the variables in the optimal solution for $LP(G, w)$ have value less than $\frac{1}{k}$. In the latter case, we include all the vertices in the solution set which have non-zero value in the vertex set B. We formally express this algorithm as follows.

Algorithm 1: k-Approximation Algorithm for Minimum $\Pi_{k,k}$ Vertex Deletion

Input: A bipartite graph $G = (A \cup B, E)$ and $w : A \cup B \to \mathbb{Q}^+$;
Output: A solution S to Minimum $\Pi_{k,k}$ Vertex Deletion for G;

$S = \emptyset$;
Compute an optimal solution x for $LP(G, w)$;
Compute the set $P_{(G,w)}(x) = \{v \in V \mid x_v \geqslant \frac{1}{k}\}$;
while *G is not empty and $P_{(G,w)}(x) \neq \emptyset$* **do**
  $S = S \cup P_{(G,w)}(x)$;
  $G = G[(A \cup B) - P_{(G,w)}(x)]$;
  Remove all the vertices which are not in any T-structure of G and call this new smaller graph as G;
  Compute an optimal solution x to $LP(G, w)$ and compute the set $P_{(G,w)}(x)$;
end
if *G is not empty* **then**
  Compute the set $P_{(G,w)}(x) = \{v \in V \mid x_v > 0 \text{ and } v \in B\}$;
end
$S = S \cup P_{(G,w)}(x)$;
Return S;

First, we show that the set S returned by Algorithm 1 is a solution to Minimum $\Pi_{k,k}$ Vertex Deletion.

Lemma 2.2. *The set S returned by Algorithm 1 is a solution to* Minimum $\Pi_{k,k}$ Vertex Deletion *for G.*

Proof. We claim that the set S returned by Algorithm 1 is a solution to Minimum $\Pi_{k,k}$ Vertex Deletion. Let the while loop in the algorithm make t loops. Let P_i be the set of vertices, given by $P_{(G,w)}(x)$, that are included into the solution set S in the ith execution of the while loop. Also in each execution of the while loop the algorithm deletes few more vertices from the graph G. Let Z_i be the set of these vertices which are deleted from the graph without being included in the solution set S during the ith execution of the while loop. Then from the algorithm, there is no T-structure in $G[\bigcup_{i=1}^{t} Z_i]$.

If the algorithm terminates with G as an empty graph at the end of while loop then $S = \bigcup_{i=1}^{t} P_i$ is a solution to Minimum $\Pi_{k,k}$ Vertex Deletion for G as $S \cap T_j \neq \emptyset$, for all $T_j \in \mathcal{K}$. Suppose G is not an empty graph at the end of while loop and $x_v < \frac{1}{k}$, for all $v \in V$. Let $T_j = \{a_1, a_2, \ldots, a_k, b_1, b_2, \ldots, b_k\}$ be a T-structure in G with $a_i \in A$ and $b_i \in B$ for $1 \leqslant i \leqslant k$. Clearly, at least one of $\{x_{b_i}\colon 1 \leqslant i \leqslant k\}$ is strictly positive, otherwise $\sum_{t \in T_j} x_t < 1$. Hence after the while loop, the set $P_{(G,w)}(x) = \{v \in V \mid x_v > 0 \text{ and } v \in B\}$ has nonempty intersection with each T-structure in G. Hence S is a solution to Minimum $\Pi_{k,k}$ Vertex Deletion. $\square$

Theorem 1. *Algorithm 1 is a k-approximation algorithm for* Minimum $\Pi_{k,k}$ Vertex Deletion.

Proof. Let (G,w) be a given weighted graph. Let S^* be a minimum weight solution to Minimum $\Pi_{k,k}$ Vertex Deletion and S be the set returned by the algorithm.

We give a proof by induction on the number of iterations (or the number of times we solve $LP(G,w)$) of the algorithm.

Base case: $w(S) \leqslant kw(S^*)$ for all weighted graphs (G,w) for which Algorithm 1 makes exactly one iteration.

We prove this by considering two cases based on whether $P_{(G,w)}(x)$ is empty or not. Suppose $P_{(G,w)}(x) \neq \emptyset$. Since $S = P_{(G,w)}(x)$, $w(S) = \sum_{v\in S} w(v) \leqslant k\sum_{v\in S} w(v)x_v \leqslant kw(S^*)$. For the other case, assume that $P_{(G,w)}(x) = \emptyset$. Here $w(S) \leqslant kw(S^*)$ because the solutions χ_S and the optimal dual solution y satisfy the relaxed complementary slackness conditions with $\alpha = 1$ and $\beta = k$. Since the input graph is a bipartite graph and S contains vertices only from B, $|S \cap T_j| \leqslant k$, for all $y_j > 0$.

Induction hypothesis: For some fixed positive integer $t \geqslant 1$, we assume that $w(S) \leqslant kw(S^*)$ for all weighted graphs (G,w) for which Algorithm 1 makes exactly t iterations.

Induction step: We show that, $w(S) \leqslant kw(S^*)$ for all weighted graphs (G,w) for which Algorithm 1 makes exactly $t+1$ iterations.

Let x be an optimal solution for $LP(G,w)$ in the first iteration of the algorithm for the weighted graph (G,w). Let $P_1 = \{v \in V \mid x_v \geqslant \frac{1}{k}\}$. P_1 is nonempty as the algorithm makes more than one iteration. Now the input for the second iteration is (G',w') where $G' = G[V-P_1]$ and $w'(v) = w(v)$, for all $v \in V - P_1$. Clearly, for the input (G',w') Algorithm 1 takes exactly t iterations. Let $S' = S \setminus P_1$ be the solution to Minimum $\Pi_{k,k}$ Vertex Deletion for (G',w') obtained from Algorithm 1 and $S^{*\prime}$ be an optimal solution to $LP(G',w')$. By induction hypothesis, we have $w(S') \leqslant kw(S^{*\prime})$.

Since S^* is a feasible integral solution for $LP(G,w)$, $w(S^*) \geqslant \sum_{v\in V} w(v)x_v = \sum_{v\in P_1} w(v)x_v + \sum_{v\in V-P_1} w(v)x_v \geqslant \frac{1}{k}w(P_1) + \sum_{v\in V-P_1} w(v)x_v$.

The vector x' defined as $x'(v) = x(v)$, for $v \in V - P_1$, is a feasible solution for $LP(G',w')$. Therefore, $w(x') = \sum_{v\in V-P_1} w(v)x_v \geqslant w(S^{*\prime}) \geqslant \frac{1}{k}w(S')$. Now $w(S^*) \geqslant \frac{1}{k}w(P_1) + \frac{1}{k}w(S') = \frac{1}{k}w(S)$. Hence $w(S) \leqslant kw(S^*)$. □

For $k_1 \neq k_2$, the computation of $P_{(G,w)}(x)$ in each iteration inside the while loop can be modified as $P_{(G,w)}(x) = \{v \in V \mid x_v \geqslant \frac{1}{\alpha}\}$, where $\alpha = \max\{k_1,k_2\}$, to get an approximation algorithm for Minimum Π_{k_1,k_2} Vertex Deletion. Similarly, modifying the proof of the above theorem by replacing k with α throughout will give an approximation factor of α.

Theorem 2. *Algorithm 1 is an α-approximation algorithm for* Minimum Π_{k_1,k_2} Vertex Deletion, *where* $\alpha = \max\{k_1,k_2\}$.

From Algorithm 1, Theorems 1 and 2 we have the following theorem.

Theorem 3. Minimum Chain Vertex Deletion, Minimum Dissociation Vertex Deletion *and* Minimum Bipartite Threshold Vertex Deletion *are approximable within a factor of* 2. Minimum Bipartite Claw Vertex Deletion *is approximable within a factor of* 3. Minimum Bi-Complement Vertex Deletion *is approximable within a factor of* 4.

2.2. Maximum $\Pi_{k,k}$ subgraph

Using the vertex deletion algorithms developed so far, we improve the approximation factor to $2 - \frac{1}{k}$, where k is the approximation factor of the corresponding vertex deletion algorithm. It should be observed that for any non-trivial hereditary property in a bipartite graph, the finite forbidden characterization cannot contain an independent set. An independent set in the set of forbidden subgraphs would imply that the maximum size of the bipartitions is bounded, hence contradicting the non-triviality. Therefore, returning one of the bipartitions will give a factor 2 approximation algorithm for Maximum $\Pi_{k,k}$ Subgraph, for $k \geqslant 2$.

Algorithm 2: Factor $(2 - \frac{1}{k})$ Approximation Algorithm for Maximum $\Pi_{k,k}$ Subgraph

Input: A bipartite graph $G = (A \cup B, E)$, $w : A \cup B \to \mathbb{Q}^+$ and a graph class $\mathcal{G}$ which is in $\Pi_{k,k}$ (characterized by finite forbidden structures);
Output: A vertex set S such that $G[S]$ is in the graph class $\mathcal{G}$;

Compute a solution P to Minimum $\Pi_{k,k}$ Vertex Deletion for G using Algorithm 1;
$S = \text{arg-max}\{w((A \cup B) - P), w(A), w(B)\}$ (Gives a set with the max argument);
Return S;

Theorem 4. Maximum $\Pi_{k,k}$ Subgraph *can be approximated within a factor of* $2 - \frac{1}{k}$.

Proof. Let S be the set returned by Algorithm 2. Clearly S is a feasible solution as $G[S]$ is a graph in class $\mathcal{G}$. We know that $w(P) \leqslant kw(P_o)$, where P_o is an optimal number of vertices to be deleted from G such that the resulting graph is in graph class $\mathcal{G}$. Also $S_o = (A \cup B) - P_o$ is an optimal solution for Maximum $\Pi_{k,k}$ Subgraph for the given input. Since $w(P) \leqslant kw(P_o)$, we have $w(V) - w(S) \leqslant k[w(V) - w(S_o)]$. From this inequality we have $k\frac{w(S_o)}{w(S)} \leqslant 1 + (k-1)\frac{w(V)}{w(S)}$. Since $w(V) \leqslant 2w(S)$, we have $\frac{w(S_o)}{w(S)} \leqslant 2 - \frac{1}{k}$. □

The arguments given in the proof of the above theorem can be modified to obtain results for Maximum Π_{k_1,k_2} Subgraph, by replacing k with $\max\{k_1, k_2\}$ in the proof.

Theorem 5. Maximum Π_{k_1,k_2} Subgraph *can be approximated within a factor of* $2 - \frac{1}{\alpha}$, *where* $\alpha = \max\{k_1, k_2\}$.

From Theorems 4 and 5, we have

Theorem 6. Maximum Induced Chain Subgraph, Maximum Induced Dissociation Subgraph *and* Maximum Induced Bipartite Threshold Subgraph *are approximable within a factor of* $\frac{3}{2}$. Maximum Induced Bipartite Claw Subgraph *is approximable within a factor of* $\frac{5}{3}$. Maximum Induced Bi-Complement Subgraph *is approximable within a factor of* $\frac{7}{4}$.

3. Hardness results

In this section, we give lower bound results on approximation factors for unweighted versions of Minimum Chain Vertex Deletion and Maximum Induced Chain Subgraph. These lower bounds are obtained by establishing reductions from other problems with known lower bounds. First, we state a known result on Maximum 3SAT.

Lemma 3.1. *(See [1].) If NP* $\subseteq$ *PCP*$(\log n, 1)$, *then there exist an integer* $k \geqslant 3$ *and a positive constant c such that there is a reduction* τ *from* SAT *to* Maximum 3SAT *that ensures the following for some fixed* $\epsilon > 0$.

- *For a given instance* ϕ *of* SAT, τ *constructs an instance* $\tau(\phi)$ *of* Maximum 3SAT *with* $N = kn^c + (k-2)2^k n^c$ *boolean variables and* $M = (k-2)2^k n^c$ *clauses.*
- $\phi \in$ SAT $\Longrightarrow$ Maximum 3SAT$(\tau(\phi)) = (k-2)2^k n^c$,
  $\phi \notin$ SAT $\Longrightarrow$ Maximum 3SAT$(\tau(\phi)) < \frac{1}{1+\epsilon}(k-2)2^k n^c$.
- *If* $\phi \notin$ SAT, *then at least* $\frac{n^c}{2}$ *clauses of* $\tau(\phi)$ *are not satisfied.*

In [14], Yannakakis proved that Minimum Chain Vertex Deletion is *NP*-complete by establishing a reduction from 3SAT. Given an instance ϕ (with m clauses and n variables) of 3SAT this reduction constructs a bipartite graph $G(\phi)$ with $(4n + 6m + 2)$ vertices, an instance of Minimum Chain Vertex Deletion, with the following property.

$\phi \in$ 3SAT $\Longrightarrow$ Minimum Chain Vertex Deletion$(G(\phi)) = n + 2m$,
$\phi \notin$ 3SAT $\Longrightarrow$ Minimum Chain Vertex Deletion$(G(\phi)) > n + 2m$.

By composing these two reductions we have a gap-reduction from SAT to Minimum Chain Vertex Deletion with the following property.

$\phi \in$ SAT $\Longrightarrow$ Minimum Chain Vertex Deletion$(G(\tau(\phi))) = N + 2M$,
$\phi \notin$ SAT $\Longrightarrow$ Minimum Chain Vertex Deletion$(G(\tau(\phi))) \geqslant N + 2M + \frac{n^c}{2}$.

Therefore, Minimum Chain Vertex Deletion cannot be approximated within a factor smaller than $\frac{N+2M+\frac{n^c}{2}}{N+2M} = 1 + \frac{n^c}{2(N+2M)} = 1 + \frac{1}{2[k+3(k-2)2^k]}$. From the above arguments, we have the following result.

Theorem 7. Minimum Chain Vertex Deletion *is APX-complete.*

Since the Minimum Chain Vertex Deletion is a special case of the Minimum Π_{k_1,k_2} Vertex Deletion, the above theorem coupled with the constant factor approximation algorithms obtained in the previous sections imply the following theorem.

Theorem 8. Minimum Π_{k_1,k_2} Vertex Deletion, *for* $\max\{k_1, k_2\} \geqslant 2$, *is APX-complete on bipartite graphs.*

The above composition can also be used to prove that Maximum Induced Chain Subgarph is *APX*-complete. We observe the following properties of $G(\phi)$ where $G(\phi)$ is the graph as constructed in the reduction from 3SAT to Mininimum Chain Vertex Deletion in [14].

$$\phi \in 3\text{SAT} \Longrightarrow \text{Maximum Induced Chain Subgraph}(G(\phi)) = 3n + 4m + 2,$$
$$\phi \notin 3\text{SAT} \Longrightarrow \text{Maximum Induced Chain Subgraph}(G(\phi)) < 3n + 4m + 2.$$

Again by composing this construction with the reduction for Maximum 3SAT, we have a gap reduction for Maximum Induced Chain Subgraph with the following properties.

$$\phi \in \text{SAT} \Longrightarrow \text{Maximum Induced Chain Subgraph}(G(\tau(\phi))) = 3N + 4M + 2,$$
$$\phi \notin \text{SAT} \Longrightarrow \text{Maximum Induced Chain Subgraph}(G(\tau(\phi))) \leqslant 3N + 4M + 2 - \frac{n^c}{2}.$$

Therefore, Maximum Induced Chain Subgraph cannot be approximated within a factor greater than $\frac{3N+4M+2-\frac{n^c}{2}}{3N+4M+2} = 1 - \frac{n^c}{2(3N+4M+2)} > 1 - \frac{1}{2[3k+7(k-2)2^k+2]}$. Hence it follows that,

Theorem 9. Maximum Induced Chain Subgraph *is APX-complete.*

Similar to the generalization we made for the *APX*-completeness of the node deletion problem above, we can also generalize the above theorem by coupling it with the constant factor approximation algorithm for the maximum vertex induced subgraph problem and the following theorem follows.

Theorem 10. Maximum Π_{k_1,k_2} Subgraph, *for* $\max\{k_1, k_2\} \geqslant 2$, *is APX-complete on bipartite graphs.*

4. Conclusion

In the paper, we described algorithms for the node deletion problems for some non-trivial hereditary properties when the input is a bipartite graph. Books by Golumbic [7] and Brandstadt et al. [3] provide a nice survey of different graph classes and it will be interesting to systematically study the approximability of optimization problems associated with these graph classes. Notable ones include deleting minimum number of vertices to get interval graphs or chordal graphs. We leave the approximability of these problems open. An important problem in the context of results in this paper would be to improve the approximation bounds and bring them as close as possible to the hardness lower bounds or *vice-versa*. Moreover, for the problems that we consider, we see that the approximation factor can be reduced by a considerable value when the given graph is restricted to be bipartite while the result does not extend to general graphs or graphs which are close to bipartite. In these cases, we could look for an approximation algorithm with running time exponential in some parameter related to the non-bipartiteness of the graph, like the length of the longest odd cycle or the number of nodes to be deleted to make the graph bipartite. We could also look for an approximation factor which is better than the existing ones but a function of these parameters.

Acknowledgements

The authors thank the referees for their valuable comments and suggestions which helped revise and reorganize this paper. One of the authors, N. Safina Devi, thanks the University Grants Commission, New Delhi, India for the financial support provided in the form of Senior Research Fellowship.

References

[1] S. Arora, C. Lund, Hardness of approximations, in: Dorit Hochbaum (Ed.), Approximation Algorithms for NP-Hard Problems, PWS Publishing, 1996.
[2] V. Bafna, P. Berman, T. Fujito, A 2-approximation algorithm for the undirected feedback vertex set problem, SIAM J. Discrete Math. 12 (3) (1999) 289–297.
[3] A. Brandstadt, V.B. Le, J.P. Spinrad, Graph Classes – A Survey, SIAM Monogr. Discrete Math. Appl., SIAM, Philadelphia, 1999.
[4] V. Chvatal, P.L. Hammer, Set-packing and threshold graphs, Univ. Waterloo Res. Report, CORR 73-21, 1973.
[5] U. Feige, S. Goldwasser, L. Lovász, S. Safra, M. Szegedy, Interactive proofs and the hardness of approximating cliques, J. ACM 43 (2) (1996) 268–292.
[6] T. Fujito, A unified approximation algorithm for node-deletion problems, Discrete Appl. Math. 86 (1998) 213–231.
[7] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980.
[8] J.M. Lewis, M. Yannakakis, The node-deletion problem for hereditary properties is NP-complete, J. Comput. System Sci. 20 (2) (1980) 219–230.
[9] V.V. Lozin, On a generalization of bi-complement reducible graphs, in: 25th International Symposium, MFCS 2000, in: Lecture Notes in Comput. Sci., vol. 1893, 2000, pp. 528–538.
[10] C. Lund, M. Yannakakis, The approximation of maximum subgraph problems, in: Automata, Languages and Programming, in: Lecture Notes in Comput. Sci., vol. 700, 1993, pp. 40–51.
[11] S. Mishra, M. Kumar, N. Safina Devi, S. Saurabh, Approximation algorithms for minimum chain vertex deletion, in: Proceedings of WALCOM, in: Lecture Notes in Comput. Sci., vol. 6552, 2011, pp. 21–32.
[12] J.H. Tu, F. Yang, The vertex cover P3 problem in cubic graphs, Inform. Process. Lett. 113 (2013) 481–485.
[13] V.V. Vazirani, Approximation Algorithms, Springer, 2001.
[14] M. Yannakakis, Node deletion problems on bipartite graphs, SIAM J. Comput. 10 (2) (1981) 310–327.