Discussiones Mathematicae
Graph Theory 21 (2001) 293–301

AN ATTRACTIVE CLASS OF BIPARTITE GRAPHS

Rodica Boliac and Vadim Lozin

RUTCOR, Rutgers University
640 Bartholomew Rd. Piscataway NJ 08854-8003 USA

e-mail: (boliac,lozin)@rutcor.rutgers.edu

Abstract

In this paper we propose a structural characterization for a class of bipartite graphs defined by two forbidden induced subgraphs. We show that the obtained characterization leads to polynomial-time algorithms for several problems that are NP-hard in general bipartite graphs.

Keywords: bipartite graphs, structural characterization, polynomial algorithm.

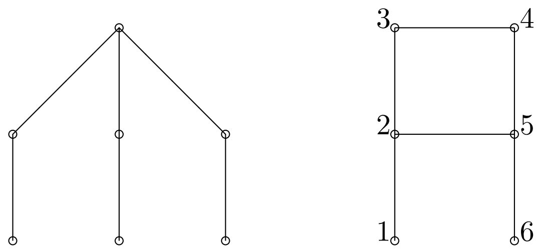
2000 Mathematics Subject Classification: 05C75.

1. Introduction

All graphs in this paper are simple (undirected and loopless) and bipartite. A bipartite graph $G = (W, B, E)$ consists of a set W of white vertices, a set B of black vertices, and a set of edges $E \subseteq W \times B$. For a bipartite graph $G = (W, B, E)$, we denote by $\widetilde{G}$ the bipartite complement to G, i.e., $\widetilde{G} = (W, B, (W \times B) - E)$. The neighborhood of a vertex x, i.e. the set of vertices adjacent to x, is denoted $N(x)$. The degree of x is $|N(x)|$. A vertex of degree 1 is called *pendant*.

As usual, C_n and P_n denote, respectively, a chordless cycle and a chordless path on n vertices. A complete bipartite graph with parts of size n and m is denoted $K_{n,m}$. In addition, by $S_{i,j,k}$ we denote a tree with exactly three vertices of degree one being at distance i, j, k from the only vertex of degree three. In this notation, $S_{1,1,1} = K_{1,3}$ is a claw and $S_{1,1,2}$ is a fork. The graph $S_{2,2,2}$ is depicted in Figure 1(a).

We call a bipartite graph G *almost complete* if for every vertex x in G, there is at most one unlike-colored vertex non-adjacent to x. Obviously, a bipartite graph G is almost complete if and only if $\widetilde{G}$ is a graph with vertex degree at most 1. The class of graphs with maximum degree 1 is exactly the class of $K_{1,2}$-free bipartite graphs, i.e. bipartite graphs containing no $K_{1,2}$ as an induced subgraph. Similarly, $K_{1,3}$-free ($S_{1,1,1}$-free) bipartite graphs have vertex degree at most two. In other words, every connected $S_{1,1,1}$-free bipartite graph is either a cycle or a path. A generalization of $S_{1,1,1}$-free bipartite graphs, the class of $S_{1,1,2}$-free graphs, has been characterized recently as follows [1]: every connected $S_{1,1,2}$-free bipartite graph is either a cycle or a path or an almost complete bipartite graph. In the present paper we study an extension of $S_{1,1,2}$-free bipartite graphs defined by two forbidden induced subgraphs: $S_{2,2,2}$ and A (Figure 1).



(a) graph $S_{2,2,2}$ (b) graph $A(1,2,3,4,5,6)$

Figure 1

The class of $(S_{2,2,2}, A)$-free bipartite graphs also generalizes bipartite interval graphs that can be characterized in terms of forbidden induced subgraphs as $S_{2,2,2}$-free graphs without cycles. Kötzig proposed in [8] the following characterization of bipartite interval graphs: the connected bipartite interval graphs are exactly the caterpillars, defined as follows.

Definition 1. A *caterpillar* is a tree that becomes a path by removing the pendant vertices.

In this paper we extend the notion of a caterpillar in the following way.

Definition 2. A *circular caterpillar* G is a graph that becomes a cycle C_k by removing the pendant vertices. We call G a long circular caterpillar if $k > 4$.

The key idea in characterizing the class of $(S_{2,2,2}, A)$-free bipartite graphs is the notion of a prime graph. To introduce it, let us call two vertices *similar* if they have the same neighborhood. Clearly similarity is an equivalence relation.

Definition 3. A bipartite graph G is called *prime* if every similarity class in G is of size 1.

It is not hard to see that any bipartite graph has a unique (up to an isomorphism) maximal prime induced subgraph that can be obtained by choosing exactly one vertex in each similarity class. We say that vertex x *distinguishes* vertices y and z if x has exactly one neighbor in $\{y, z\}$. Obviously a graph is prime if and only if for every two vertices of the same color, there is a third vertex that distinguishes them.

2. Structural Characterization

Theorem 4. *A connected prime $(S_{2,2,2}, A)$-free bipartite graph $G = (W, B, E)$ is either a caterpillar or a long circular caterpillar or an almost complete bipartite graph.*

Proof. If G does not contain any cycle, then it is a tree and hence a caterpillar due to $S_{2,2,2}$-freeness. In what follows we assume that G contains a cycle.

1. Suppose first that G contains a cycle of length more than 6 induced by vertices $C = \{1, 2, \ldots, 2k\}$, where $k > 3$. Assume vertex $x \notin C$ has a neighbor $i \in C$. Our goal is to prove that i is the only neighbor of x, i.e., x is a pendant vertex.

We show first that x is adjacent neither to $i-2$ nor to $i+2$. To prove this, assume the contrary: $(x, i-2) \in E$. Now, in order to be prime, G must contain a vertex y that distinguishes x and $i-1$. Without loss of generality, we let y be adjacent to x but not to $i-1$. Then y is adjacent to $i-3$, otherwise G contains the induced subgraph $A(i-3, i-2, i-1, i, x, y)$. Now if $y \neq i-4$, then G contains the induced subgraph $A(i-1, i-2, x, y, i-3, i-4)$, and if $y = i-4$, then G contains the induced subgraph $A(i-4, x, i-2, i-1, i, i+1)$. The contradiction in both cases proves that x is not adjacent to $i-2$. Symmetrically, x is not adjacent to $i+2$.

Now we are able to show that i is the only neighbor of x in G. Indeed, assume y is another neighbor of x. Then y is not adjacent to $i+1$ else G contains the induced subgraph $A(i-1, i, x, y, i+1, i+2)$. Similarly y is not adjacent to $i-1$. But then vertices $i-2, i-1, i, i+1, i+2, x, y$ induce a subgraph $S_{2,2,2}$ in G.

The above arguments permit us to conclude that if G contains a cycle of length more than 6, then G is a long circular caterpillar.

2. Assume G contains a cycle on 6 vertices $C = \{1, 2, 3, 4, 5, 6\}$. Denote by B_j the subset of vertices outside C that have exactly j neighbors in C. Obviously, for $j > 3$, B_j is empty since G is bipartite. We show now that

2.1. $B_0 = \emptyset$,

2.2. $B_2 = \emptyset$,

2.3. $B_1 = \emptyset$ or $B_3 = \emptyset$.

To derive equality 2.2, we apply arguments similar to those used in case 1. Assume vertex x has exactly two neighbors in C, say 1 and 3. Then, to be prime, G must contain a vertex y adjacent, say, to x but not to 2. Now G contains either the induced subgraph $A(y, x, 3, 2, 1, 6)$ (if y is not adjacent to 6) or $A(2, 1, x, y, 6, 5)$ (if y is adjacent to 6).

To prove 2.3, assume G contains both a vertex x in B_1 and a vertex y in B_3. Without loss of generality we let y be adjacent to all odd vertices in C. If x is adjacent to an odd vertex, say 1, then G contains the induced subgraph $A(x, 1, 6, 5, y, 3)$. If x is adjacent to an even vertex, say 2, then G contains either the induced subgraph $A(x, 2, 3, y, 1, 6)$ or $A(x, y, 1, 6, 5, 4)$ depending on the adjacency of x to y. Thus, either B_1 or B_3 is empty.

To show 2.1, consider a nearest to C vertex x in B_0. Let y be a neighbor of x on a shortest path connecting x to C. Due to 2.2, y belongs either to B_1 or to B_3. If y is in B_1, say y is adjacent to 1, then vertices $5, 6, 1, 2, 3, y, x$ induce a subgraph $S_{2,2,2}$ in G. If y is in B_3, say y is adjacent to the odd vertices, then G contains the induced subgraph $A(x, y, 3, 4, 5, 6)$.

To complete the case, we show next that

2.4. if B_3 is empty, then G is a long circular caterpillar;

2.5. if B_1 is empty, then G is an almost complete bipartite graph.

To prove 2.4, assume, to the contrary, that B_1 contains two adjacent vertices x and y whose neighbors in C are i and j, respectively. Since G is bipartite, vertices i and j are at odd distance in the cycle. If $i = j-1$, then G contains the induced subgraph $A(i-1, i, x, y, j, j+1)$. If i and j are at distance 3 in the cycle, then vertices $i-2, i-1, i, i+1, i+2, x, y$ induce a subgraph $S_{2,2,2}$ in G.

To show 2.5, we assume, by contradiction, there is a vertex x in G that has two non-neighbors y and z in the opposite part of the graph. Since B_1 and B_2 are empty, x, y, z belong to B_3. Moreover, there must be a vertex a in G that distinguishes y and z. Suppose a is adjacent to y but not to z. Obviously a is in B_3. But then G contains the induced subgraph $A(x, 3, a, y, 2, z)$.

3. Suppose G contains a C_4 induced by vertices $1, 2, 3, 4$ and does not contain any cycle of length more than 4. To distinguish vertices 2 and 4 in G, we let vertex 5 be adjacent to 4 but not to 2. Similarly, to distinguish 1 and 3, we let vertex 6 be adjacent to 1 but not to 3. Then 5 is adjacent to 6, otherwise G contains the induced subgraph $A(5, 4, 3, 2, 1, 6)$. Now G is an almost complete bipartite graph. To prove this, we denote by C the set of vertices $\{1, 2, 3, 4, 5, 6\}$ and by B_j the subset of vertices outside C that have exactly j neighbors in C. Then

- B_1 is empty. Indeed, if the only neighbor in C for a vertex $x \in B_1$ is 1 or 3, then vertices $1, 2, 3, 4, 5, x$ induce a subgraph A in G.
- B_2 is empty. Indeed, if a vertex $x \in B_2$ is adjacent to 3 and 5, then vertices $1, 2, 3, x, 5, 6$ induce a cycle on 6 vertices, contradicting the assumption. If x is adjacent to 1 and 3, then the contradiction follows by applying the proof of case 2.2.
- B_0 is empty. For the proof we refer the reader to case 2.1.

Now the conclusion is the same as for case 2.5. ■

3. Algorithms

In this section we use the obtained characterization for the class of $(S_{2,2,2}, A)$-free bipartite graphs in order to derive polynomial-time algorithms for two problems that are NP-hard in general bipartite graphs. Both problems are related to the notion of a matching, i.e., a subset of edges, no two of which have a vertex in common.

3.1. The jump number

The jump number problem came to graph theory from the theory of partial orders. In the class of bipartite graphs the jump number problem coincides with the problem of finding a maximum alternating cycle-free matching [4].

A matching in a graph is called *alternating cycle-free* (AC- free for short) if no cycle in the graph has exactly half of its edges in the matching. We denote the number of edges in a maximum AC-free matching in a graph G by $J(G)$.

Müller has shown that the problem of finding a maximum AC-free matching is NP-complete even for chordal bipartite graphs [9]. Nevertheless, efficient methods for jump-minimization have been found for several subclasses of bipartite graphs, like bipartite permutation [10], biconvex [2], convex [5] and distance-hereditary bipartite graphs [9]. Note that all the listed classes are subclasses of chordal bipartite graphs, i.e., bipartite graphs without induced cycles of length more than 4. We now extend the above list with the class of $(S_{2,2,2}, A)$-free bipartite graphs that contains all bipartite cycles.

Obviously, for the jump number problem, we may be restricted to connected graphs. Moreover, the following lemma permits us to restrict ourselves to prime graphs.

Lemma 5. *Let H be a maximal prime induced subgraph of a graph G, then $J(G) = J(H)$.*

Proof. Any AC-free matching M covers at most one vertex in each similarity class of G, otherwise there is an alternating C_4 for the matching. Without loss of generality we may assume that vertices covered by M belong to H. Therefore, $J(G) \leq J(H)$. The inverse inequality is trivial. ■

Due to Lemma 5 and Theorem 4, we consider the problem for the following graphs: caterpillars, long circular caterpillars and almost complete bipartite graphs.

To solve the problem for almost complete bipartite graphs, we use the following general result.

Lemma 6. *For a $\widetilde{K}_{1,n}$-free bipartite graph G, $J(G) \leq n$.*

Proof. Suppose $J(G) \geq n + 1$ and let M be an AC-free matching in G with $n + 1$ edges $(a_1, b_1), \ldots, (a_{n+1}, b_{n+1})$.

We show by induction on k that vertex b_k, $1 \leq k \leq n$, is adjacent to at least one vertex a_j with $j > k$. Indeed, if b_1 has no neighbors in $\{a_2, \ldots, a_{n+1}\}$, then vertices $b_1, a_2, \ldots, a_{n+1}$ induce a $K_{1,n}$ in the bipartite complement to G. Without loss of generality we let b_1 be adjacent to a_2.

To make the inductive jump, we assume that vertex b_i is adjacent to vertex a_{i+1} for each $i = 1, \ldots, k$. Then b_{k+1} has no neighbors in set $\{a_1, \ldots, a_k\}$, otherwise G has an alternating cycle with respect to M. Therefore, in order to be $\widetilde{K}_{1,n}$-free, b_{k+1} must be adjacent to at least one vertex a_j with $j > k + 1$. We let b_{k+1} be adjacent to a_{k+2}.

Now the contradiction follows for $k = n+1$. On the one hand, vertex b_{n+1} must have a neighbor a_j with $j < n+1$, otherwise G contains an induced $\widetilde{K}_{1,n}$. On the other hand, if b_{n+1} has a neighbor a_j with $j < n+1$, then the cycle $(a_j, b_j, a_{j+1}, \ldots, b_{n+1})$ is alternating for M. ■

Recall from the introduction that if G is an almost complete bipartite graph, then it is $\widetilde{K}_{1,2}$-free and hence $J(G) \leq 2$. Assume now G is a (circular) caterpillar. If G does not contain a pendant vertex, then obviously G is a cycle C_k with $k \geq 6$. In that case, solution is trivial: $J(C_k) = k/2 - 1$. If G contains a pendant vertex, we use the following simple lemma that remains valid for arbitrary bipartite graphs.

Lemma 7. *For every pendant vertex a with the only neighbor b, there is a maximum AC-free matching in G that contains edge (a, b).*

Proof. Let M be a maximum AC-free matching in graph G. If M does not contain (a, b), then it must contain an edge (b, c), otherwise M is not maximum. But then $M' = (M - \{(b, c)\}) \cup \{(a, b)\}$ is a matching of the same cardinality. Obviously M' is AC-free, because a is pendant. ■

We now present a rough description of the algorithm as follows. Given a connected $(S_{2,2,2}, A)$-free bipartite graph G, find a maximal prime induced subgraph H of G. If H is an almost complete bipartite graph, then $J(G) = J(H) \leq 2$. If H does not contain a pendant vertex, then $H = C_k$ and therefore $J(H) = k/2 - 1$. If a is a pendant vertex with neighbor b, then $H' = H - \{a, b\}$ is a cycle-free graph, i.e. H' has a pendant vertex as well. We hence apply Lemma 7 to compute $J(H)$ recursively: $J(H) = J(H') + 1$.

With some care, the algorithm above can be implemented in time $O(n^2)$ for graphs on n vertices. Hence the conclusion

Theorem 8. *Given an $(S_{2,2,2}, A)$-free bipartite graph G with n vertices, one can find a maximum alternating cycle-free matching in G in time $O(n^2)$.*

3.2. Maximum induced matching

A matching in a graph is induced if no two edges in the matching have a third edge connecting them. The number of edges in a maximum size induced matching of a graph G is denoted $i\mu(G)$. The problem of finding a maximum induced matching has been introduced in [3], where the author proved its NP-hardness in the class of bipartite graphs. On the other hand, the problem is known to be solvable in the class of trees in linear time [6, 11] (see [7] for more solvable cases for the problem). Below we use the solution for trees in order to develop an efficient procedure to solve the problem in the class of $(S_{2,2,2}, A)$-free bipartite graphs.

Again, like for the jump number problem, it is obviously sufficient to consider only connected graphs. Moreover, we can apply the proof of Lemma 5 with no extra arguments to conclude that $i\mu(H) = i\mu(G)$, where H is a maximal prime induced subgraph of a graph G. Consequently, we study the problem for caterpillars, long circular caterpillars and almost complete bipartite graphs.

To solve the maximum induced matching problem for almost complete bipartite graphs, we denote by mK_2 a regular graph of degree 1 with $2m$ vertices. It is not hard to see that $i\mu(G) = m$ if and only if G contains mK_2, but not $(m+1)K_2$, as an induced subgraph.

Lemma 9. *If G is an almost complete bipartite graph, then $i\mu(G) \leq 2$.*

Proof. The bipartite complement to G is a graph with maximum degree 1. Hence $\widetilde{G}$ is C_6-free. It is not hard to verify that $\widetilde{C_6} = 3K_2$. Thus G is a $3K_2$-free graph and hence $i\mu(G) \leq 2$. ∎

Suppose now that G is a (circular) caterpillar. If G contains no cycle, then we can apply the algorithm in [6] to solve the problem. If G is an even cycle C_n, then $i\mu(G) = [k/3]$. If G contains both a cycle C_k and a pendant vertex a, we denote by G_a the subgraph obtained from G by deleting vertex a together with the only neighbor b of a and all the vertices adjacent to b. Then

$$i\mu(G) = \max\{[k/3], i\mu(G_{a_1}) + 1, \ldots, i\mu(G_{a_s}) + 1\},$$

where $a_1, \ldots, a_s$ is the list of all pendant vertices in G.

Every G_{a_j} is obviously a tree. So, we can solve the problem for G applying the algorithm in [6] at most n times. All the above arguments lead to

Theorem 10. *Given a $(S_{2,2,2}, A)$-free bipartite graph G with n vertices, one can find a maximum induced matching in G in time $O(n^2)$.*

References

[1] V.E. Alekseev, *A polynomial algorithm for finding maximum stable sets in fork-free graphs*, Discrete Analysis and Operations Research, Ser.1, **6** (4) (1999) 3–19 (in Russian).

[2] A. Brandstädt, *The jump number problem for biconvex graphs and rectangle covers of rectangular regions*, Lecture Notes in Computer Science **380** (1989) 68–77.

[3] K. Cameron, *Induced matchings*, Discrete Appl. Math. **24** (1989) 97–102.

[4] G. Chaty and M. Chein, *Ordered matchings and matchings without alternating cycles in bipartite graphs*, Utilitas Mathematica **16** (1979) 183–187.

[5] E. Dahlhaus, *The computation of the jump number of convex graphs*, Lecture Notes in Computer Science **831** (1994) 176–185.

[6] G. Fricke and R. Laskar, *Strong matchings on trees*, Congr. Numer **89** (1992) 239–243.

[7] M.C. Golumbic and M. Lewenstein, *New results on induced matchings*, Discrete Appl. Math. **101** (2000) 157–165.

[8] A. Kötzig, *Paare Hajös the Graphen*, Časopis Pěst. Mat. **88** (1963) 236–241.

[9] H. Müller, *Alternating cycle free matchings in chordal bipartite graphs*, Order **7** (1990) 11–21.

[10] G. Steiner and L. Stewart, *A linear time algorithm to find the jump number of* 2*-dimensional bipartite orders*, Order **3** (1987) 359–367.

[11] M. Zito, *Linear time maximum induced matching algorithm for trees*, Nordic J. Computing **7** (2000) 58–63.

Received 3 March 2001
Revised 19 May 2001