Polynomial Algorithms for Special Cases of the Balanced Complete Bipartite Subgraph Problem*

Claudio Arbib
Università degli Studi di L'Aquila
Dipartimento di Matematica Pura e Applicata
via Vetoio, 67010 Coppito (L'Aquila)
Italia,
and
II Università degli Studi di Roma "Tor Vergata"
Centro Vito Volterra
viale della Ricerca Scientifica, 00133 Roma
Italia
email: arbib@smaq20.univaq.it

Raffaele Mosca
II Università degli Studi di Roma "Tor Vergata"
Centro Vito Volterra
viale della Ricerca Scientifica, 00133 Roma
Italia

ABSTRACT. A balanced bipartite graph is a bipartite graph (U, V, E) such that $|U| = |V|$. Particular balanced bipartite subgraph problems have applications in fields such as VLSI design and flexible manufacturing. An example of such problems is the following: given a graph G and a positive integer m, does G contain a balanced complete bipartite subgraph with at least $2m$ vertices? This problem is NP-complete for several classes of graphs, including bipartite graphs. However, the problem can be solved in polynomial time for particular graphs classes. We aim to contribute to the characterization of "easy" classes of instances of the problem, and to individuate graph-theoretic properties that can be useful to develop solution algorithms for the general case. A simple polynomial algorithm can be devised for bipartite graphs with no induced P_5 on the basis of a result of Bacsó and Tuza. We generalize the result to particular subclasses of i) graphs with no odd cycles of given size, ii) paw-free graphs, iii) diamond-free graphs.

*Work supported by Progetto Finalizzato Trasporti 2, Contract n. 93.01781.PF74, of the Italian National Research Council (CNR).

1 The complete bipartite subgraph problem

1.1 Preliminary definitions

Throughout the paper, a (simple) graph will be denoted as an ordered pair $G = (V, E)$, where V and E indicate the vertex and the edge set, respectively. Graphs are assumed with no loops (i.e., irreflexive) and, unless otherwise specified, undirected. The vertex set (the edge set) of a graph G will also be denoted as $V(G)$ (as $E(G)$). For any subset S of $V(G)$, we let $E(S)$ denote the set of all the edges of G having both endpoints in S. A graph is called H-free if no subset S of its vertices induces a subgraph isomorphic to a given graph H.

P_m, C_m and K_m stand for path, cycle and complete graph of m vertices. A multipartite graph is denoted as an ordered $(k+1)$-ple $B = (V_1, \ldots, V_k, F)$, where $\{V_1, \ldots, V_k\}$ is a partition of $V(B)$ into k stable sets, and $F = E(B)$ is the edge set of B. We set in particular $V_i = V_i(B)$, $i = 1, \ldots, k$. The vertex sets $V_1, \ldots, V_2$ are called the *shores* of B. The complete bipartite graph with $m + n$ vertices is denoted as $K_{m,n}$. A complete bipartite graph $K_{m,n}$ is called *balanced* if $m = n$.

For $m \geq 2$, $n \geq 3$, $(K_{m,n} - 2e)$ denotes a bipartite graph obtained by deleting any two non-adjacent edges from $K_{m,n}$. Since $(K_{m,n} - 2e) \equiv (K_{n,m} - 2e)$, with no loss of generality we can from now on assume $m \leq n$. In particular, $(K_{2,3} - 2e) \equiv P_5$, i.e., every P_5-free graph is also $(K_{m,n} - 2e)$-free. A *triangle* is a complete graph with three vertices. A *paw* is a graph with 4 vertices x, y, w, z, and 4 edges xy, xw, yw, yz. A *diamond* is a graph with 4 vertices x, y, w, z, and 5 edges xy, xw, yw, yz, wz (see Figure 1).

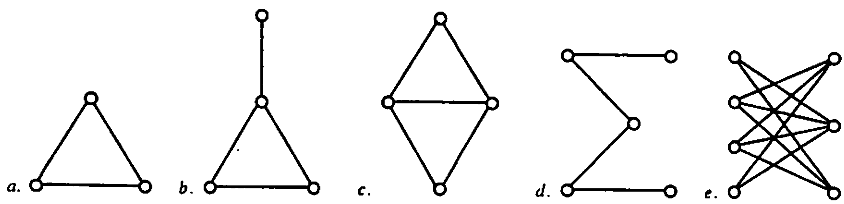

Figure 1

a) triangle; b) paw; c) diamond; d) $P_5 \equiv (K_{2,3} - 2e)$; e)$(K_{n,m} - 2e)$

This work focuses on the problem of finding a maximum size balanced complete bipartite subgraph of a given graph. More formally:

> Given a graph $G = (V, E)$ with n vertices and a positive integer $m \leq n$, find, if one, a (not necessarily induced) balanced complete bipartite subgraph H of G with at least $2m$ vertices.

This problem is here referred to as the *complete bipartite subgraph* problem.

1.2 Applications

The *complete bipartite subgraph* problem deserves particular interest in such fields as VLSI design [11] and flexible manufacturing [3]. In the former case, it formulates a class of design problems known as PLA *folding* [5, 8], which consist of reducing the area of a programmed logic array (PLA) through an efficient circuit layout (for a more detailed description of PLAs, see for instance [11]). In the latter, it models concurrent task and resource assignment problems in bi-processor systems subject to resource constraints [6].

In most applications, a problem instance consists of a $0-1$ rectangular matrix **A**.

In PLA folding, **A** represents the so-called *topological matrix* of the PLA (see Figure 2a): rows and columns of **A** correspond to "wires" obtained by diffusion onto a polysilicon layer, and the element a_{ij} of **A** situated at the cross-point of row i and column j is equal to 1 if and only if the two wires are connected by means of an active device (e.g., a transistor) called *personalization.* The row-wires are generally called *gates*, and each gate of a PLA is associated with a particular input or output signal.

Minimizing the area occupied by devices and components in an integrated circuit is crucial issue in VLSI design[1]. In fact, the more compact the circuit, the faster its electrical transitions. Moreover, the smaller the semiconductor chip supporting the circuit, the higher the chances that it is free of impurity that can affect correct circuit response. On the other hand, the number of personalizations needed to implement a boolean function via a PLA is usually a very small fraction of all the possible cross-points in the grid: an analysis due to Wood [16] sets this fraction between 4% and 10% (since this range depends on the combinatorial nature of the circuits to be realized, and not on the particular solid-state technology adopted, we can rely on Wood's report although it dates back to 1979). This sparse structure suggests the possibility of reducing the device area through particular layout methods. One of such methods consists of allowing distinct signals to share the same track, and is known as *folding.* In particular, two gates can share the same horizontal track (which, to this purpose, is interruped at a suitable point) only if the sets of column-wires they are connected to do not intersect each other. The problem is to assign as much as possible gate pairs to horizontal tracks.

More in detail (refer for instance to Figure 2b), one can partition the set C of the columns of the topological matrix **A** into 2 parts, C_1 and C_2. The row set R can in turn be partitioned into 3 sets, R_1, R_2, $R \backslash (R_1 \cup R_2)$, so that $|R_1| = |R_2|$, and set R_1 (set R_2) contains rows i such that $a_{ik} = 0$

[1] Computer programs for PLA optimization are included as benchmark in SPEC96INT standards for computer performance.

for all $k \in C_1$ (for all $k \in C_2$). By doing so, matrix $\mathbf{A}$ turns out to be partitioned into 5 blocks, two of which (namely, $R_1 \times C_1$ and $R_2 \times C_2$) have all elements equal to 0 and can be suppressed, thus reducing the device area. In the example, $C_1 \equiv \{4, 1\}$, $C_2 \equiv \{2, 5, 3\}$, $R_1 \equiv \{6, 7\}$, $R_2 \equiv \{4, 8\}$.

a.

	1	2	3	4	5
1	1	0	1	0	0
2	1	1	0	0	0
3	0	1	1	0	1
4	0	0	0	1	0
5	1	1	0	1	0
6	0	1	1	0	0
7	0	0	0	0	1
8	0	0	0	1	0

b.

	4	1	2	5	3	
4	1	0	1	0	1	6
8	1	0	0	1	0	7
1	0	1	0	0	1	
2	0	1	1	0	0	
5	1	1	1	0	0	
3	0	0	1	1	1	

Figure 2

a) Example of topological matrix of a PLA; b) example of layout

The above model can be given a different interpretation in terms of concurrent task and resource assignment (see e.g. [6]). According to this interpretation, each row of $\mathbf{A}$ is associated with a task, and each column with a resource. Element a_{ij} is equal to 1 if and only if the execution of task i requires resource j. Given a two-processor system, one problem is to simultaneously assign tasks and resources to processors so to maximize the workload of the least busy processor (i.e., the system utilization). When only one copy of each resource is available, two tasks can be executed in parallel if and only if they require non intersecting resource sets. Figure 2b illustrates the case of 2 parallel processors with static resource assignment for constant tasks lengths. In practical cases, the objective is related to maximizing the time interval between two consecutive tool loading. This objective is relevant in flexible manufacturing applications (see for instance [3, 9]), where a wide variety of products is to be manufactured by the same tool machine group.

The problems described can be formulated as *complete bipartite subgraph* by introducing the following notion of *compatibility graph G*.

Definition 1. *A graph $G = (V, E)$ is the* compatibility graph *of some rectangular matrix $\mathbf{A} \in \{0, 1\}^{m \times n}$ if its vertices are in a one-to-one correspondence with the rows of $\mathbf{A}$, and $u_i u_j \in E$ if and only if $a_{ik} \neq a_{jk}$ for $1 \leq k \leq n$.*

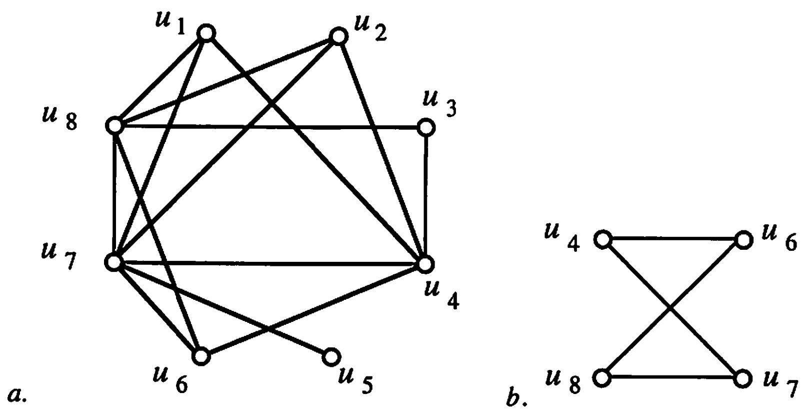

Figure 3
a) The compatibility graph of the matrix of Figure 2a;
b) a complete bipartite subgraph

The compatibility graph G of the matrix $\mathbf{A}$ depicted in Figure 2a is shown in Figure 3a, (vertex u_i biunivoquely correspond to the i-th row of $\mathbf{A}$, $1 \leq i \leq 8$). The layout of Figure 2b corresponds to the balanced complete bipartite subgraph of G of Figure 3b.

Observe that in many applications G happens to have a peculiar structure. For instance, in VLSI layout, wiring requirements derived from the external circuit layout may impose that, for each row, the associated gate must lay on a particular side of the PLA. In this case, no two gates assigned to the same side can be folded, even if compatible. Similarly, in resource assignment problems, one may have to face situations in which tasks have been a-priori assigned to the machines (e.g., according to particular machine capabilities), and one wants to assign the resources so to maximize the system utilization. In both cases, we can limit our attention to solving *complete bipartite subgraph* in a particular bipartite subgraph of G.

1.3 Scope of the paper

As far the complexity of the problem is concerned, we have the following results:

Theorem 1. (Johnson, 1987). Complete bipartite subgraph *polynomially reduces from* clique.

Proof: See [10].

From the proof of Theorem 1, it follows that *complete bipartite subgraph* is NP-complete even if G is bipartite. We incidentally observe that complete bipartite subgraph is NP-complete for other classes of graphs, including regular graphs of degree 3 or more [13], complement of split graphs [5] and edge-disjoint graphs of regular graphs [2] (reduction from graph bisection). Conversely, the problem is in P for graphs with no induced $K_{1,3}$ and (K_5-e) (in particular, for edge-graphs) [1], for partial k-trees (see [13]) and for other classes of graphs (see [5, 6]). Other results [15] seem to provide indications for non-approximability.

Similarly to [1, 5, 6, 13], aim of this paper is to contribute to the characterization of "easy" classes of instances of the problem, and to individuate graph-theoretic properties that can be useful to devise solution algorithms for the general case. In particular, we show that, for fixed $m \geq 2$, $n > 2$ integers, *complete bipartite subgraph* can be solved in polynomial time when G has no induced subgraph isomorphic to $(K_{m,n} - 2e)$ and, in alternative:

i) no cycle of a given odd number of vertices, or

ii) no induced paw, or

iii) no induced diamond.

The paper is organized as follows: in Section 2 we introduce some further notation and preliminary results, and give a characterization of those $(K_{m,n}-2e)$-free graphs that are also triangle-free or diamond-free; a polynomial algorithm for *complete bipartite subgraph* on $(K_{m,n}-2e)$- and triangle-free graphs is described in Section 3; Section 4 is devoted to extend the result to larger classes of graphs.

2 Notation and preliminary results

2.1 Notation

Let G be a graph. For any subset S of $V(G)$, the neighbor set $N(S)$ of S is the set of all the vertices of $V(G)\backslash S$ adjacent to a vertex of S. For brevity, we set

$$N(S_1, \ldots, S_m, v_1, \ldots, v_p) = N(S_1 \cup \cdots \cup S_m \cup \{v_1, \ldots, v_p\})$$

We will also make extensive use of the following notation:

$$D(S) \equiv \bigcap_{v \in S} N(v)$$

Each element of $D(S)$ is adjacent to every vertex of S (in other words, each vertex in $D(S)$ *dominates* S). Of course, also S and $D(S)$ are mutually disjoint, and similarly to above we set

$$D(S_1, \ldots, S_m, v_1, \ldots, v_p) = D(S_1 \cup \cdots \cup S_m \cup \{v_1, \ldots, v_p\})$$

and also

$$d(S_1, \ldots, S_m, v_1, \ldots, v_p) = |D(S_1, \ldots, S_m, v_1, \ldots, v_p)|$$

For example, in the graph depicted in Figure 3a, $\{u_4, u_8\}$ is dominated by u_1, u_2, u_3, u_6, u_7, hence $d(u_4, u_8) = 5$. In general, we clearly have $D(v) \equiv D(v, v, \ldots, v) \equiv N(v)$. Notice that, for any $S \subseteq V(G)$ with s elements, S and $D(S)$ are the shores of a (not necessarily induced) subgraph of G isomorphic to $K_{s,d(S)}$.

For every subset S of $V(G)$, let V_S denote the following subset of $V(G)$:

$$V_S \equiv \{u \in V \backslash S : D(S, u) \neq \emptyset\} \tag{1}$$

In particular, for $S \equiv \emptyset$, the set $V_\emptyset \equiv \{u \in V : N(u) \neq \emptyset\}$ coincides with $V(G)$ if and only if G has no isolated vertices. For example, in the graph depicted in Figure 3a, $V_{\{u_4, u_8\}} = \{u \in V, u \neq u_4, u \neq u_8, d(u_4, u_8, u) > 0\} = \{u_1, u_2, u_5, u_6, u_7\}$.

Let us also give the following definition:

Definition 2. (subgraph induced by S and T). *Let S and T be disjoint subsets of V. The subgraph $G_{S,T}$ induced on G by S and T is the subgraph induced by $S \cup T$ on the graph $G = (V, E \backslash (E(S) \cup E(T)))$, i.e., on the graph obtained from G by deleting all the edges having both endpoints in S or in T.*

Notice that $G_{S,T}$ is bipartite by definition.

2.2 Preliminary results

Graphs with no induced P_5 and C_5 have been characterized in [4]:

Theorem 2. (Bacsó and Tuza, 1990). *In a graph G, every connected subgraph contains a dominating clique if and only if G is P_5-free and C_5-free.*

The absence of induced P_5 can help solving hard graph problems such as *maximum stable set* [12] or, as we will see later, *complete bipartite subgraph.* It is therefore interesting to individuate cases in which such a property holds:

Theorem 3. *If for $p \geq 2$, $q > 2$, $q \geq p$ integers a bipartite graph G is $(K_{p,q} - 2e)$-free, then for any $H = (V_1, V_2, F)$ contained in G and isomorphic to $K_{q-2,q-2}$ the subgraph induced by $D(V_1) \cup D(V_2)$ is P_5-free.*

Proof: Let $G(H)$ denote the subgraph of G induced by $D(V_1) \cup D(V_2)$. Suppose that $G(H)$ contains an induced P_5 with vertices $u_1, \ldots, u_5$ and edges $u_i u_{i+1}$ for $i = 1, \ldots, 4$. With no loss of generality, assume u_i adjacent to V_1 for i even. Let $X = V_1 \cup \{u_1, u_3, u_5\}$, $Y = V_2 \cup \{u_2, u_4\}$. Then, $G_{X,Y}$ is isomorphic to $K_{q+1,q} - 2e$, and hence G contains a $K_{p,q} - 2e$. □

$(K_{p,q} - 2e)$-free graphs that are also triangle-free can be characterized through the following theorem:

Theorem 4. *Let G be triangle-free. Then, for any two integers $p > 2$, $q > 2$, G is $(K_{p,q} - 2e)$-free if and only if for every subset S of $V(G)$ such that $|S| \geq p - 2$ and any $u, v \in V(G)$ such that $d(S, u, v) \geq q - 2$, either $D(S, u) \supseteq D(S, v)$, or $D(S, v) \supseteq D(S, u)$.*

Before proving the theorem, let us point out what follows.

Observation 1. Theorem 4 cannot be extended to the case $p = 2$ ($S = \emptyset$), and in particular to P_5-free graphs. In fact, consider the P_5- and K_3-free graph of Figure 4. One has $D(u_1, u_3) \neq \emptyset$ (that is, $d(u_1, u_3) \geq q - 2 = 1$), but neither $D(u_1) \supseteq D(u_3)$, nor the opposite. □

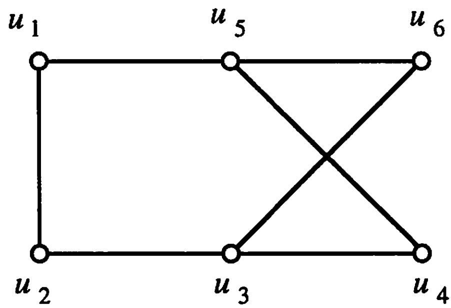

Figure 4 A counterexample to Theorem 4 with $p = 2$

Proof of Theorem 4: (if part). Assume that for every subset S of V with at least $p - 2$ elements and for any $u, v \in V(G)$ such that $d(S, u, v) \geq q - 2$, one has either $D(S, u) \supseteq D(S, v)$ or the opposite. By contradiction, assume then that G contains an induced $(K_{p,q} - 2e)$, say H. Let $w_i \neq z_i \in V_i = V_i(H)$, $i = 1, 2$, and $w_1 w_2$, $z_1 z_2$ be the missing edges of H. The set $S \equiv V_1 \backslash \{w_1, z_1\}$ has $p - 2$ vertices. The vertices w_1, z_1, and those in S are all adjacent in H to a subset T of V_2 with $q - 2$ vertices, i.e., $d(S, w_1, z_1) = |D(S, w_1) \cap D(S, z_1)| \geq q - 2$. Moreover, since $w_1 w_2, z_1 z_2 \notin E(H)$, one has $w_2 \in D(S, z_1) \backslash D(S, w_1)$ and $z_2 \in D(S, w_1) \backslash D(S, z_1)$. Hence neither $D(S, w_1) \cap V_2 \supseteq D(S, z_1) \cap V_2$, nor the opposite, which is a contradiction.

(Only if part). The theorem is obvious for $d(S) \leq 2$. Assume $d(S) > 2$, $|S| \geq p - 2$, and let $J \equiv \{u \in V \mid D(S, u) \neq \emptyset\}$. Since G is triangle-free, $J \cap N(S) \equiv \emptyset$: in fact, $u \in J \cap N(S)$ implies the existence of two vertices $v \in S$, $w \in D(S)$ (and hence $w \in N(S)$) both adjacent to u; but, by definition of $D(S)$, $(v,w) \in E(G)$, i.e., G contains a triangle (contradiction). Observe also that, for the same reason, both S and $D(S)$ are stable sets of G.

Let $u_1, u_2 \in V(G)$ be such that $T = D(S, u_1, u_2)$ has at least $q-2$ elements, and assume by contradiction that there exist $w_1 \in D(S,u_1)\backslash D(S,u_2)$, $w_2 \in D(S,u_2)\backslash D(S,u_1)$, $w_1 \neq w_2$. Observe that the existence of such vertices imply $u_1, u_2 \in S$. Also, u_1 and u_2 are non-adjacent, since G is triangle-free and T contains at least one vertex adjacent to both u_1 and u_2. Thus, $U = S \cup \{u_1, u_2\}$ has at least p elements and is stable. On the other hand, $|T| \geq q-2$, and $w_1, w_2 \notin T$. Hence, $W = T \cup \{w_1, w_2\}$ has at least q elements and, as a subset of $D(S)$, is stable as well. But then $G_{U,W}$ contains a subgraph isomorphic to $(K_{p,q} - 2e)$. □

From the proof of Theorem 4, it turns out that the assumption of G triangle-free is not necessary to prove the if part. Under different assumptions on set S and vertices u, v we can characterize $(K_{p,q} - 2e)$-free graphs also in the class of diamond-free graphs:

Theorem 5. *Let G be diamond-free. Then, for any two integers $p > 2$, $q \geq 2$, G is $(K_{p,q} - 2e)$-free if and only if for every stable subset S of $V(G)$ such that $|S| \geq \max\{p-2, 2\}$ and any $u, v \in V(G)$ such that $d(S,u,v) \geq \max\{q-2, 2\}$, either $D(S,u) \supseteq D(S,v)$, or $D(S,v) \supseteq D(S,u)$.*

Observe that in this case S is still non-empty, but stable.

Proof of Theorem 5: The if part follows from the proof of Theorem 4. To prove the converse, let us proceed as follows. The following three facts immediately follow from G being diamond-free. First, since S is stable and contains at least 2 elements, also $N(S)$ is stable. Secondly, $d(S,u,v) \geq \max\{q-2, 2\}$ implies $u, v \notin N(S)$: if not, as $d(S,u) \geq d(S,u,v) \geq 2$ and $N(S)$ is stable and contains $D(S,u)$, G would admit an induced diamond. Third, for any $u, v \notin N(S)$ such that $d(S,u,v) \geq 2$, $uv \notin E(G)$. Hence, for such vertices u, v, $X = S \cup \{u,v\}$ and $Y = D(S,u,v)$ are both stable. Then, with an argument similar to that used in the proof of Theorem 4, one can show that if there exist $w \in D(S,u)\backslash D(S,v)$, $z \in D(S,v)\backslash D(S,u)$, then $G_{X,Y}$ is an induced $(K_{p,q} - 2e)$ of G. □

Consider the following binary relation between elements of V_S:

$$\forall\, u, v \in V_S, u \underline{\searrow} v \text{ if and only if } D(S,u) \supseteq D(S,v) \qquad (2)$$

For example, referring to the graph of Figure 3a and assuming $S = \{u_4, u_8\}$, we have $V_S = \{u_1, u_2, u_5, u_6, u_7\}$, and $D(S,u_7) = \{u_1, u_2, u_6\}$, $D(S,u_1) = D(S,u_2) = D(S,u_5) = D(S,u_6) = \{u_7\}$: hence no pair of elements in V_S are in the relation (2). If on the other hand we let $S = \{u_1\}$, we see that in $V_S = V(G)\backslash\{u_1\}$ one has:

$$u_2 \setminus u6 \quad u_6 \setminus u_2 \quad u_2, u_6 \setminus u_3, u_7 \quad u_2, u_6 \setminus u_4, u_5, u_8$$
$$u_3 \setminus u_7 \quad u_7 \setminus u_3 \quad u_4 \setminus u_5 \setminus u_8 \setminus u_4.$$

We can prove the following:

Theorem 6. *If a bipartite graph G is P_5-free, then for any non-empty subset S of V the relation '$\setminus$' is a total order of the elements of V_S.*

In other words, under the assumptions of Theorem 6, for any subset S of $V(G)$ there exists a total order of the elements of $V(G)\backslash S$. The proof can easily be obtained through arguments similar to those used in Theorem 4, and is left to the reader. It is not difficult to see that, if G is connected, then Theorem 6 also holds for $S \equiv \emptyset$. We have in particular:

Theorem 7. *If a connected bipartite graph $G = (V_1, V_2, E)$ is P_5-free, then the relation '$\setminus$' is a total order of the elements of both $V_\emptyset \backslash V_1$ and $V_\emptyset \backslash V_2$.*

Let us now focus on graphs without odd cycles of a given size. The following lemmas will be used in Section 4 to limit the search of the maximum $K_{m,m}$ of G to particular bipartite subgraphs of G. The first result gives a straightforward property of graphs with no C_{2k+1}.

Lemma 1. *Let G be a graph with no cyles of $2k+1$ vertices and H a balanced complete bipartite subgraph of G with $2m$ vertices, $m \geq k$. Then, H is an induced subgraph of G.*

The next lemma applies to non-adjacent disjoint subsets S and U of $V(G)$, and gives a sufficient condition for U and $D(S) \cap N(U)$ being stable.

Lemma 2. *Let G contain no cycle with $2k+1$ vertices, $k > 1$. Then, for any two disjoint subsets S and U of $V(G)$ such that*

i) $|S| \geq k-1$

ii) *for any $u \in U$, $v \in S$, $(u,v) \notin E(G)$*

iii) $\forall\, u \in U$, $d(S,u) > k$

U and $D(S) \cap N(U) \equiv \cup_{u \in U} D(S,u)$ are stable sets of G.

Proof: With no loss of generality, let $S \equiv \{v_1, \dots, v_{k-1}\}$. Let us first show that U is stable. Let therefore $u_1, u_2 \in U$, and assume by contradiction that u_1 and u_2 are adjacent. Since $d(S, u_i) > k$, $i = 1, 2$, G contains at least k vertices $w_1, \dots, w_k$ adjacent to each vertex of S. Moreover, u_1 and u_2 are respectively adjacent to two distinct vertices of $N(S)$, say w_1 and w_k. Then, G contains a cycle $\{u_2, u_1, w_1, v_1, w_2, v_2, \dots, w_{k-1}, v_{k-1}, w_k, u_2\}$ with $2k+1$ vertices.

Now assume by contradiction that two vertices of $D(S) \cap N(U)$, say w_1 and w_2 are adjacent. Let in particular $w_1 \in D(S,u)$, $u \in U$. Since $d(S,u) > k > 1$, there exist $k-1$ vertices $w_3, \ldots, w_{k+1}$ adjacent to all of the vertices of S, and one of them, say w_{k+1}, is also adjacent to u. Then, G contains a cycle $\{w_1, w_2, v_1, w_3, v_2, w_4, \ldots, v_{k-1}, w_{k+1}, u, w_1\}$ with $2k+1$ vertices. □

3 The complete bipartite subgraph problem on triangle- and $(K_{p,q} - 2e)$-free graphs

In this section we will show that if G is triangle- and $(K_{p,q} - 2e)$-free, then *complete bipartite subgraph* can be solved in $O(|V(G)|^h)$ time, where $h = \max\{p, q-3\}$.

Let us give the following definition:

Definition 3. *Let S be a subset of $V(G)$. A bipartite subgraph B of G is said to be* supported *by S if $V_1(B) \supseteq S$.*

For any graph G, let S denote a subset of $V(G)$, and consider a partition of $V(G)\backslash S$ into equivalence classes A_i, where A_i corresponds to some vertex $u_i \in V(G)\backslash S$ and contains all the $v \in V(G)\backslash S$ such that $D(S,v) \equiv D(S,u_i) \neq \emptyset$. Recalling that $D(S) \supseteq D(S,u)$ for any $u \in V(G)\backslash S$, we can construct a directed acyclic graph Ω_S with node set $\{u_0, u_{i_1}, u_{i_2}, \ldots, u_{i_t}\}$, where u_{i_k} represents set A_{i_k}, $k = 1, \ldots, t$, and node u_0 represents $A_0 \equiv \emptyset$.

The arc set of Ω_S is such that the transitive closure of Ω_S draws the partial order relation '$\searrow$' defined by (2) between pairs u, w of vertices of Ω_S representing distinct A_i's. In practice, excluding the nodes u_i such that $D(S,u_i) \neq \emptyset$, Ω_S is a strongly connected tree with root in u_0.

Example 1. Consider the P_5- and triangle-free graph G of Figure 5a, and set $S = \{u_8\}$. One has

$$D(S) = \{u_2, u_4, u_6\}, \qquad D(S,u_i) = \emptyset \text{ for } i = 2,4,6.$$

$$D(S,u_1) = \{u_2\},\ D(S,u_3) = \{u_2,u_4,u_6\},\ D(S,u_5) = D(S,u_7) = \{u_4,u_6\}$$

We can partition $V(G)\backslash S$ into $A_1 = \{u_1\}$, $A_3 = \{u_3\}$, $A_5 = \{u_5, u_7\}$ and $A_4 = \{u_2, u_4, u_6\}$: $\Omega_{\{u_8\}}$ will contain exactly one node for each of them, plus one root associated with $A_0 = \emptyset$. Graph $\Omega_{\{u_8\}}$ is drawn in Figure 5b: each of its nodes carries the indication of the corresponding sets A_i and $D(S,u_i)$ ($\{2,4,6\}$ is a shorthand for $\{u_2,u_4,u_6\}$, etc.). □

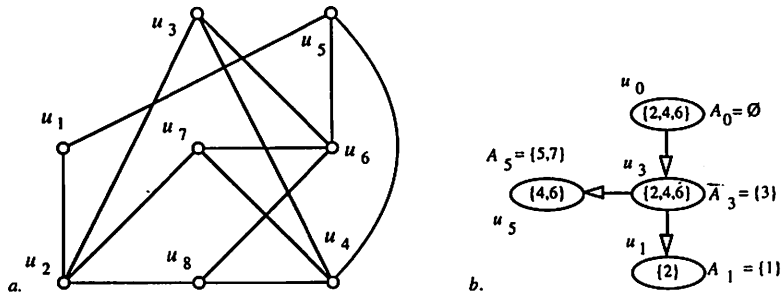

Figure 5 Sample graphs G and Ω_S

Let us point out a first consequence of Theorem 4:

Corollary 1. *Let G be triangle- and $(K_{p,q} - 2e)$-free with $p > 2$, $q > 2$. Then, for any subset S of $V(G)$ with at least $p - 2$ vertices and any two nodes u_i, u_j of Ω_S that cannot be reached from u_0 through the same path, $d(S, u, w) < q - 2$ for any $u \in A_i$, $w \in A_j$.*

Let $(u_{i_1}, u_{i_2}, \ldots, u_{i_r})$, with $i_1 = 0$, be a path of Ω_S originating in u_0, and let $T_k \equiv D(S, A_{i_1}, \ldots, A_{i_k})$, $1 \le k \le r$. For instance, referring to Figure 5b and to the path (u_0, u_3, u_1), one has $T_0 = D(S) = \{u_2, u_4, u_6\} = T_3$, $T_1 = D(S, A_0, A_3, A_1) = D(S, A_1) = \{u_2\}$. The following is another immediate consequence of Theorem 4.

Corollary 2. $D(S) \equiv D(S, A_{i_1}) \supseteq D(S, A_{i_2}) \supseteq \ldots \supseteq D(S, A_{i_k}) \equiv T_k$.

Consider the complete bipartite graph $B_r = (S_r, T_r, S_r \times T_r)$ induced by the above defined set T_r and by the set $S_r \equiv S \cup A_{i_1} \cup \cdots \cup A_{i_r}$ (recall that, since G has no loops, S_r and T_r are mutually disjoint). Based on the above Corollaries 1 and 2, let us prove the following theorem:

Theorem 8. *Let G be triangle- and $(K_{p,q} - 2e)$-free, with $p > 2$, $q > 2$. Let S be a subset of $V(G)$, and H be the maximum $K_{m,m}$ of G supported by S (if one). Then, either $m < q - 2$, or there exists a path $(u_{i_1}, u_{i_2}, \ldots, u_{i_r})$ of Ω_S originating in $u_{i_1} = u_0$ such that $B_r \supseteq H$.*

Proof: Let $V_1(H) \equiv S \cup U$, with $S \cap U \neq \emptyset$ and $|S \cup U| = m$. Clearly, $G_{S \cup U, D(S,U)} \supseteq H$. Consider the set $A \equiv \{v \in V(G) \backslash S \colon \forall\, u \in U, D(S, v) \equiv D(S, u)\}$. From the definition of A, one has $S \cup A \supseteq S \cup U$ and $D(S, A) \equiv D(S, U)$. Hence $G_{S \cup A, D(S,A)} \supseteq H$, and for every two subsets W and Z of A and $D(S, W)$ with $m - |S|$ and m vertices, respectively, $G_{S \cup W, Z} \equiv K_{m,m}$.

Let A be partitioned into equivalence classes $A_{l_1}, \ldots, A_{l_t}$ as previously described, and let $\Lambda \equiv \{u_{l_1}, \ldots, u_{l_t}\}$ denote the set of nodes of Ω_S corresponding to $A_{l_1}, \ldots, A_{l_t}$. Consider the minimum subtree of Ω_S that spans $\Lambda \cup \{u_0\}$, and let $N \supseteq \Lambda \cup \{u_0\}$ denote the node set of such a subtree. From

Corollary 2, we have $D(S,N) \equiv D(S,A)$ and therefore $G_{S\cup N,D(S,N)} \supseteq H$. Moreover, for every two subsets W and Z of N and $D(S,W)$ with $m-|S|$ and m vertices, respectively, $G_{S\cup W,Z} \equiv K_{m,m}$.

Let $\pi_1,\ldots,\pi_s$ denote all the paths originating in u_0 and spanning the nodes of N. Let us prove that $m \geq q-2$ implies $s=1$. Suppose not. Let $\pi_1 = \{u_{i_1},\ldots,u_{i_r}\}$, $\pi_2 = \{u_{k_1},\ldots,u_{k_s}\}$, $i_1 = k_1 = 0$. Since $\pi_1 \neq \pi_2$, there exist at least two distinct vertices u_{i_h}, u_{k_j} that are not connected by any directed path of Ω_S. Hence, by Corollary 1, $|D(S,u,v)| < q-2$ for any $u \in A_{i_h}$, $v \in A_{k_j}$. Let W be a subset of N with $m-|S|$ elements that contains u and v and is such that $W \cap S = \emptyset$. For any subset Z of $D(S,W)$ with m elements, we have $G_{S\cup W,Z} \equiv K_{m,m}$. Since $D(S,u,v) \supseteq D(S,W) \supseteq Z$, we have also $m = |Z| < q-2$, which is a contradiction. □

Let us associate two weights a_i and b_i with each node u_i of Ω_S, with $a_i = |A_i|$ and $b_i = d(S,A_i)$. From Theorem 8 and the definition of T_i, S_i, it follows:

Corollary 3. *Let G be triangle- and $(K_{p,q}-2e)$-free $(p,q>2)$, and S be a subset of $V(G)$ with $|S| \geq p-2$. Let $(u_{i_1},u_{i_2},\ldots,u_{i_r})$ denote a path of Ω_S originating in $u_{i_1} = u_0$. Then the maximum $K_{m,m}$ supported by S and contained in B_r has size*

$$m(i_r) = \min\{|S| + \sum_{k=1}^{r} |A_{i_k}|, d(S,A_{i_r})\} \tag{3}$$

Proof: It suffices to observe that, for every positive integer $k<r$, $T_k \supseteq T_{k+1}$ and $S_{k+1} \supseteq S_k$. As $A_{i_1},\ldots,A_{i_k}$ are mutually disjoint, $|S_k| = |S| + \sum_{j=1..k}|A_{i_j}|$, and, by definition and by Corollary 1, $|T_k| = d(S,A_{i_k})$ □

From Theorem 8 and Corollary 3, one can derive an algorithm to find, if one, a $K_{m,m}$ supported by a given subset S of the vertices of a triangle- and $(K_{p,q}-2e)$-free graph with $2<p\leq q\leq m+2$. The algorithm organizes a depth-first search on Ω_S starting from node u_0. For each node u_i visited, the algorithm computes $m(i)$ through equality (3), i.e., the size of a maximum balanced complete subgraph of B_i, and updates, if necessary, the maximum size m^* found so far. The visit of the subtree rooted in the current node is abandoned if the value $m(i)$ of its root is less than or equal to m^*.

To find the maximum $K_{m,m}$ contained in G, one can proceed as follows:

Algorithm 1.

1. Construct and search a graph Ω_S for each subset S of V with $p-2$ elements.

2. If step 1 has failed, then by Theorem 6 $m<q-2$: in particular, either no $K_{m,m}$ of G is supported by any S with $p-2$ elements

(i.e., $m < p-2$), or G contains one such $K_{m,m}$ (and in this case $p-2 \leq m < q-2$).

For any subset S of V with $p-2$ elements, step 1 of Algorithm 1 can be organized into two phases as follows:

Phase 1. (*Partition* $V \backslash S = \{u_1, \ldots, u_n\}$ into equivalence classes A_i)
(*initialization*) Assigned := $\emptyset$; for $0 \leq i \leq n$ do $A_i := \emptyset$;
for $i, j \notin$ *Assigned*, $j \geq i$ do
if $D(S, u_i) \equiv D(S, u_j)$
then $A_i := A_i \cup \{u_j\}$; *Assigned* := *Assigned* $\cup \{j\}$.
Phase II. (*Search graph* Ω_G)
(*initialization*) $m^* := 0$; $a(0) := |S|$;
search(0); output m^*

where procedure *search(i)* is recursively defined as follows:

```
procedure search(i)
        begin
        for any j that minimizes |D(S, A_i)\D(S, A_j)| with
        D(S, A_i)\D(S, A_j) ≠ ∅
        do    begin
              a(j) := a(i) + |A_j|;
              m(j) := min{a(j), d(S, u_j)};
              if m(j) > m*
              then begin m* := m(j); search(j) end;
              end;
        end search
```

The complexity of searching Ω_S (phase II of step 1) is linear in the number of nodes of Ω_S, i.e., in $|V(G)|$. This bound is however dominated by the complexity of constructing Ω_S (phase I of step 1) which is $O(|V(G)|^2)$. Summarizing, step 1 of Algorithm 1 takes $O(|V(G)|^p)$ time. As for step 2, the search for a maximum $K_{m,m}$ takes in any case $O(|V(G)|^{q-3})$ time.

In conclusion, we have the following result:

Theorem 9. *For $p > 2$, $q > 2$, if a graph G is triangle- and $(K_{p,q} - 2e)$-free, then finding in G an optimum solution to* complete bipartite subgraph *requires $O(|V(G)|^{\max\{p,q-3\}})$ time.*

Observation 2. Recalling that $(K_{2,3} - 2e) \equiv P_5$, Theorem 9 implies in particular the existence of an $O(|V(G)|^2)$ solution algorithm for bipartite P_5-free graphs. For such graphs, however, a polynomial algorithm can be directly derived from Theorem 2. In fact, if G is bipartite and P_5-free, then by Theorem 2 G contains a dominating edge $e = uv$. If we delete u and v from G, together with all the edges incident on them, we obtain a new graph consisting of several connected components $H_1, \ldots, H_p$, all dominated by e and, as subgraphs of G, bipartite and P_5-free. We can therefore recursively apply the same argument to each component H_i, and construct in this way a directed strongly connected tree T, in which the root of each subtree is associated with a pair (H, f), where H is a connected component of G, and $f \in E(H)$ dominates H. With an argument similar to that used in Theorem 4, one can prove that every maximal $K_{m,m}$ of G is individuated by the edges f associated with the nodes of a path from the root to any leaf of T. Moreover, based on Theorem 3, one can extend the above solution approach to bipartite graphs with no induced $(K_{p,q} - 2e)$. □

Example 2. Let us find a maximum $K_{m,m}$ in the P_5- and K_3-free graph of Figure 5a. Recalling that a P_5-free graph is also $(K_{3,3} - 2e)$-free, we can apply Algorithm 1 with $p = q = 3$. Sets $D(S, u)$ are listed in Table 1 for $S = \{u_j\}$, $u = u_i$, $1 \le i \le j \le 8$ (sets with $i > j$ are not indicated, as $D(\{u_i\}, u_j) = D(\{u_j\}, u_i)$).

$\{u_1\}$	$\{u_2\}$	$\{u_3\}$	$\{u_4\}$	$\{u_5\}$	$\{u_6\}$	$\{u_7\}$	$\{u_8\}$	S/u
u_2u_5	$\emptyset$	u_2	u_5	$\emptyset$	$\emptyset$	u_2	u_2	u_1
	$u_1u_3u_7u_8$	$\emptyset$	$u_3u_7u_8$	u_1	$u_3u_7u_8$	$\emptyset$	$\emptyset$	u_2
		$u_2u_4u_6$	$\emptyset$	u_4u_6	$\emptyset$	$u_2u_4u_6$	$u_2u_4u_6$	u_3
			$u_3u_5u_7u_8$	$\emptyset$	$u_3u_5u_7u_8$	$\emptyset$	$\emptyset$	u_4
				$u_1u_4u_6$	$\emptyset$	u_4u_6	u_4u_6	u_5
					$u_3u_5u_7u_8$	$\emptyset$	$\emptyset$	u_6
						$u_2u_4u_6$	$u_2u_4u_6$	u_7
							$u_2u_4u_6$	u_8

Table 1

From each column of the table, it is immediate to derive a directed rooted tree Ω_S. These trees are depicted in Figure 6. Similarly to Figure 5a, their nodes carry the indication of both sets $D(S, u_j)$ and A_i. Nodes with $D(S, u_i) = \emptyset$ have however not been drawn, since unnecessary by equality (3).

For each tree, we compute $m(r)$ by equality (3) for every path with $r \ge 1$ nodes and root in u_0 (in the present case, $|S| = 1$). Consider for instance $\Omega_{\{u_2\}}$ (the second tree in Figure 6): the paths (u_0), (u_0, u_4), (u_0, u_5) respectively yield $m(1) = 1$, $m(2) = 3$, $m(2) = 1$. Hence, G admits a $K_{3,3}$ with shores $\{u_2, u_4, u_6\}$, $\{u_3, u_7, u_8\}$. The same subgraph can be obtained from $\Omega_{\{u_3\}}$, $\Omega_{\{u_7\}}$, $\Omega_{\{u_8\}}$. □

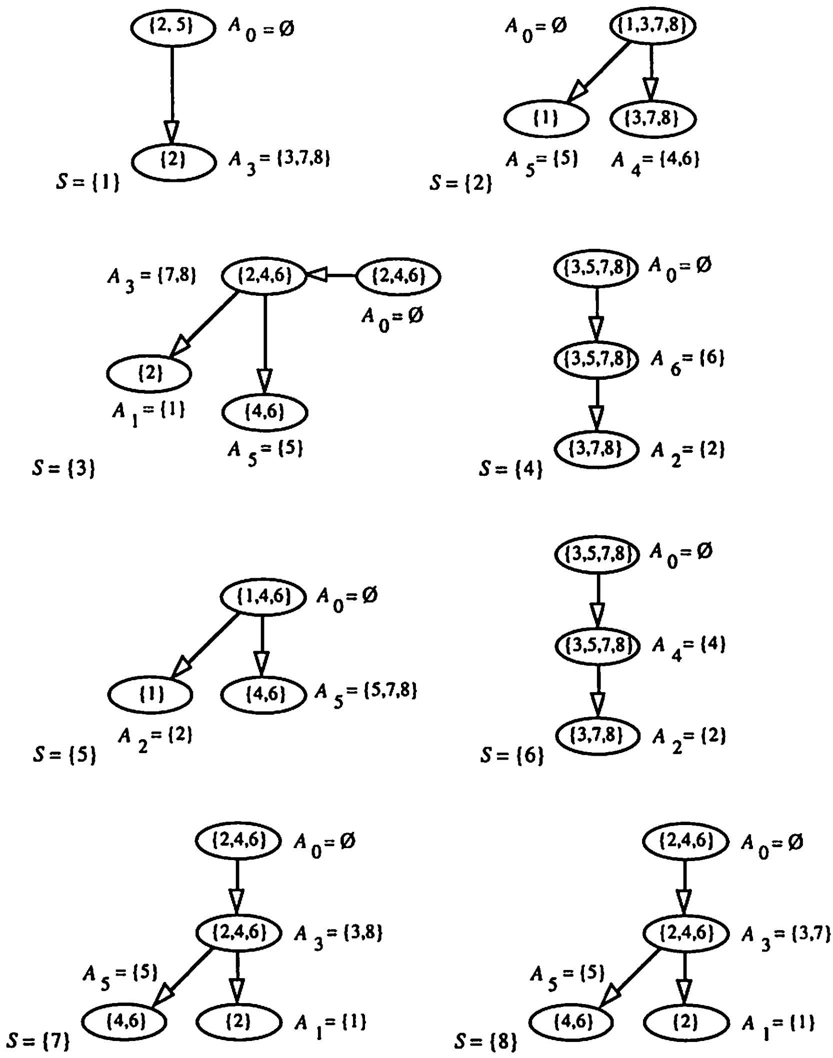

Figure 6
Trees to be searched to find a maximum $K_{m,m}$ of the graph of Figure 5a

4 Extensions of the method

4.1 Extension to diamond- and paw-free graphs

Algorithm 1 can be extended with similar arguments to diamond-free graphs with no induced $(K_{p,q} - 2e)$. Let Ψ_S denote the rooted digraph obtained by suppressing the nodes u_i of Ω_S (leaves) such that $d(S, u_i) = 1$. Based on Theorem 5, Corollary 1 and Theorem 8 rewrite as follows:

Corollary 4. *Let G be diamond- and $(K_{p,q} - 2e)$-free. Then, for any stable subset S of $V(G)$ with at least $p - 2 \geq 2$ vertices and any two nodes u_i, u_j of Ψ_S that cannot be reached from u_0 through the same path, $d(S, u, w) < \max\{q - 2, 2\}$ for any $u \in A_i$, $w \in A_j$.*

Theorem 10. *Let G be diamond- and $(K_{p,q} - 2e)$-free, with $p \geq 2$, $q > 2$. Let S be a stable subset of $V(G)$ with at least 2 elements, and H be the maximum $K_{m,m}$ of G supported by S (if one). Then, either $m < \max\{2, q - 2\}$, or there exists a path $(u_{i_1}, u_{i_2}, \ldots, u_{i_r})$ of Ψ_S originating in $u_{h_1} = u_0$ such that $B_r \supseteq H$.*

Observe that if G is diamond-free, then, for $m > 2$, every $K_{m,m}$ of G is contained either in a clique, or in an induced bipartite subgraph of G. In fact, let $G \supseteq B \equiv K_{m,m}$ with $m > 2$, and let $\{u_1, u_2, u_3, u_4\}$ be a C_4 contained in B, with $u_1, u_3 \in V_1(B)$ and $u_2, u_4 \in V_2(B)$: then, $u_1u_3 \in E(G)$ if and only if $u_2u_4 \in E(G)$, otherwise G contains a diamond. If the maximum $K_{m,m}$ is contained in an induced bipartite subgraph of G, then we can apply Algorithm 1, provided that Ψ_S is constructed from a stable set S with at least 2 vertices. Otherwise, we simply observe that the maximal cliques of a diamond-free graph G are $O(|V(G)|^2)$.

Observation 3. Olariu [14] proved that G is a paw-free graph if and only if is triangle-free or complete multipartite. This suggests an immediate extension of Algorithm 1 to paw-free graphs with no induced $(K_{p,q} - 2e)$. In fact, if G is complete multipartite with shores $V_1, \ldots, V_k$, a maximum balanced complete bipartite subgraph of G is obtained by partitioning the set $\{1, \ldots, k\}$ into 2 classes Θ_1 and Θ_2 such that the absolute value of the difference between $|\bigcup_{i \in \Theta_1} V_i|$ and $|\bigcup_{i \in \Theta_2} V_i|$ is minimum. Finding Θ_1 and Θ_2 is a polynomially solvable instance of *subset sum*. □

4.2 Extension to graphs with no odd cycles of given length

After the proof of Lemmas 1 and 2 (Section 2), and of Theorem 9, we are in a position to extend Algorithm 1 to $(K_{p,q} - 2e)$-free graphs with no C_{2k+1}, for any fixed $k > 1$.

Consider two disjoint subsets S and U of $V(G)$ such that S is stable with $k - 1$ elements, and U contains all the vertices u that are neither in S nor adjacent to any element of S, and are such that $d(S, u) > k$, i.e.

$$U \equiv \{u \in V(G) \setminus (S \cup N(S)) : d(S, u) > k\}$$

Consider also the subgraph $G(S)$ of G induced by $S \cup U \cup (D(S) \cap N(U))$. Based on Lemmas 1 and 2, we can prove the following:

Theorem 11. *Let G be a graph with no cycles of $2k+1$ vertices, S a stable set of G with $k-1$ vertices, and H a maximum $K_{m,m}$ of G supported by S, if one. Then, either $m \leq k$, or $G(S) \supseteq H$.*

Proof: By Lemma 1, H is an induced subgraph of G. By Lemma 2, and since S is stable, $G(S)$ is bipartite with shores $S \cup U$, $D(S) \cap N(U)$. Assume $m \geq k+1$. Let $v \in V_1(H)$. If $v \notin S$, then $v \in U$: in fact, $v \notin N(S)$ (otherwise H would not be induced) and $d(S,v) \geq k+1$. Let $w \in V_2(H)$. Then $w \in D(S) \cap N(U)$: in fact, as H is supported by S, w is adjacent to all the vertices in S; moreover, since $m \geq k+1$ and $|S| = k-1$, there exists some $v \in V_1(H) \backslash S$ such that $(v,w) \in E(G)$, and we just proved that $v \in V_1(H) \backslash S$ implies $v \in U$. The thesis follows. □

By Theorem 11, finding a maximum $K_{m,m}$ supported by a stable set S in a graph G without cycles of $2k+1$ vertices polynomially reduces to finding a maximum $K_{m,m}$ in a suitable bipartite subgraph $G(S)$ of G. If G has no induced $(K_{p,q} - 2e)$, then $G(S)$ has no induced $(K_{p,q} - 2e)$ either, and we can therefore apply Algorithm 1 to $G(S)$. The whole solution algorithm is as follows:

Algorithm 2.

1. For every stable set S of G with $k-1$ vertices, find through Algorithm 1 a maximum $K_{m,m}$ (if one) contained in $G(S)$.

2. If step 1 has failed, then by Theorem 11 $m \leq k$: in particular, either no $K_{m,m}$ of G is supported by any S with $k-1$ elements (i.e., $m < k$), or G contains one such $K_{m,m}$ (and in this case $m = k$).

For each stable set S of G with $k-1$ vertices, step 1 takes in the worst case $O(|V(G_S)|^{\max\{p,q-3\}})$ time. In case of failure, the search for a maximum $K_{m,m}$ required by step 2 takes $O(|V(G)|^k)$ time. Apparently, the complexity of Algorithm 2 is therefore $O(|V(G_S)|^{k+\max\{p,q-3\}-1})$. However, we can prove the following theorem:

Theorem 12. *If a graph G is $(K_{p,q} - 2e)$-free and without cycles of $2k+1$ vertices $(p, q \geq 2, k > 1)$, then the time required to find in G an optimum solution to complete bipartite subgraph is $O(|V(G)|^{\max\{k,p,q-3\}})$.*

Proof: Suppose $k-1 \leq p-2$, i.e., $k \leq p-1$. For every stable set S of G with $p-2$ vertices, construct and search graph Ω_S to find a maximum $K_{m,m}$, if one, supported by S. This takes $O(|V(G)|^p)$ time. If the search succeeds, then the $K_{m,m}$ found is the maximum supported by any stable set A of G with $k-1$ elements. If in fact A is a not contained in S, then A cannot support a maximum $K_{m,m}$, since, by Lemma 1, any balanced complete bipartite subgraph of G with $2(p-2)$ vertices is an induced subgraph of G. If the above search fails, then $m < q-2$, and we have to apply step 2 of

Algorithm 1, which takes $O(|V(G)|^{q-3})$ time. Hence, in this case the time bound is $O(|V(G)|^{\max\{p,q-3\}}) = O(|V(G)|^{\max\{k,p,q-3\}})$.

Assume now $k > p - 1$. To find whether G contains a $K_{m,m}$ with $m = k-1$, we apply an exhaustive search on G: this takes $O(|V(G)|^{k-1})$. If the search fails, then we stop. Otherwise, we must check whether G contains a $K_{m,m}$ with $m \geq k$. To this aim, we observe that, since G is $(K_{p,q} - 2e)$-free, G is also $(K_{k,q} - 2e)$-free. Then we can apply Algorithm 1 starting from stable sets S with $k-2$ elements. This takes $O(|V(G)|^{\max\{k,q-3\}})$. The overall complexity is therefore again $O(|V(G)|^{\max\{k,p,q-3\}})$. □

Conclusions

In this paper we showed how to solve in polynomial time, for any fixed integers $m \geq 2$, $n > 2$, $k > 0$, the complete bipartite subgraph problem for graphs with no induced $(K_{m,n} - 2e)$ that either i) do not contain C_{2k+1}, or ii) are paw-free, or iii) are diamond-free. The proposed algorithms are based on an orderability property enjoyed by $V(G)$ under the above assumptions. The proposed algorithms yield a feasible solution to the problem also if $V(G)$ does not enjoy the orderability property expressed (provided that a restricted definition of graph Ω_S is given). This characteristics make the algorithms suitable to construct good heuristics for the general case.

Further investigation is needed to evaluate the implications of the above results on other combinatorial problems, such as maximum clique or maximum stable set. In particular, it is easy to see that, if G is bipartite, then complete bipartite subgraph can be formulated in terms of maximum stable set with additional constraints on the bipartite complement of G.

Acknowledgement. We thank Anna Galluccio for pointing out Bacsó and Tuza's work.

References

[1] C. Arbib, Two Polynomial Problems in PLA Folding, in: R.H. Möhring, ed., *Proceedings of the 16th International Workshop on Graph Theoretic Concepts in Computer Science*, *Lecture Notes in Computer Science* (Springer-Verlag, Berlin 1991) **484**, 119–129.

[2] C. Arbib, M. Lucertini, S. Nicoloso, Polynomial and NP-complete Problems in Programmed Logic Arrays Folding, *Istituto di Analisi dei Sistemi e Informatica*, CNR, report N. 284 (1990).

[3] R.G. Askin, C.R. Standridge, *Modelling and Analysis of Manufacturing Systems*, J. Wiley & Sons, New York (1992).

[4] G. Bacsó, ZS. Tuza, Dominating Cliques in P_5-free graphs, *Periodica Mathematica Hungarica* **21**,4 (1990), 303–308.

[5] P. Damaschke, PLA Folding in Special Graph Classes, *Discrete Applied Mathematics* **51** (1994), 63–74.

[6] D. De Werra, On an Optimization Problem Occurring in FMSs: A Hypergraph-theoretical Formulation, *Discrete Applied Mathematics* **55** (1994), 107–111.

[7] J.R. Egan, C.L. Liu, Bipartite Folding and Partitioning of a PLA, *IEEE Transactions on Computer Aided Design* CAD-3 (1984), 191–199.

[8] T.C. Hu, Y.S. Kuo, Graph Folding and Programmable Logic Arrays, *Networks* **17** (1987), 19–37.

[9] E.J. Lee, P.B. Mirchandani, Concurrent Routing, Sequencing, and Setups for a Two-Machine Flexible Manufacturing Cell, *IEEE Transactions on Robotics and Automation* **4**, 3 (1988), 256–264.

[10] D.S. Johnson, The NP-completeness Column: An Ongoing Guide, *Journal of Algorithms* **8** (1987), 445–446.

[11] C. Mead, L. Conway, *Introduction to VLSI Systems*, Addison Wesley, Reading MA, (1980).

[12] R. Mosca, The Maximum Stable Set Problem on Graphs with no Induced P_5 and $(K_4 - e)$, working paper (1994).

[13] R.H. Möhring, Graph Problems Related to Gate Matrix Layout and PLA Folding, in: G. Tinhofer et al., eds., *Computing Supplementum* **7** (1990), 170–51.

[14] S. Olariu, Paw-free Graphs, *Information Processing Letters* **28** (1988), 53–54.

[15] S.S. Ravi, E.L. Lloyd, The Complexity of Near-optimal Programmable Logic Array Folding, *SIAM Journal on Computing* **17** (1988), 696–710.

[16] R.A. Wood, A High Density Programmable Logic Array Chip, *IEEE Transactions on Computing* C-28, **9** (1979), 602–608.