

On the extension of bipartite to parity graphs ^{1,2}

Serafino Cicerone, Gabriele Di Stefano *

Dipartimento di Ingegneria Elettrica, Università degli Studi di L’Aquila,
I-67040 Monteluco di Roio, L’Aquila, Italy

Received 4 November 1997; received in revised form 10 July 1998; accepted 23 December 1998

Abstract

Parity graphs form a superclass of bipartite and distance-hereditary graphs. Since their introduction, all the algorithms proposed as solutions to the recognition problem and other combinatorial problems exploit the structural property of these graphs described by Burlet and Uhry (Berge and Chvátal (Eds.), Topics on Perfect Graphs, Ann. Discrete Math., vol. 21, North-Holland, Amsterdam, 1984, pp. 253–277). This paper introduces a different structural property of parity graphs: split decomposition returns exactly, as building blocks of parity graphs, cliques and bipartite graphs. This characterization, together with the observation that the split decomposition process can be performed in linear time, allows us to provide optimum algorithms for both the recognition problem and the maximum weighted clique. Moreover, it can also be used to show that the maximum weighted independent set problem can be solved by an algorithm whose worst complexity occurs when the parity graph considered is bipartite. A remarkable consequence of this work is that the extension of bipartite graphs to parity graphs does not increase the complexity of the basic problems we have considered, since the worst case occurs when the parity graph under consideration is an undecomposable bipartite graph. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Parity graphs; Split composition; Recognition problem; Maximum weighted clique problem; Maximum weighted independent set problem

1. Introduction

Graphs represent a very general model to describe interconnection networks, circuit layouts, ordering and logical problems, properties of database schemes, and other computer science situations. For many applications, the general model can be restricted to very special classes of graphs with properties that allow us to efficiently solve common

¹ A preliminary version of this paper has been presented at the 8th International Symposium on Algorithms and Computation (ISAAC'97), December 17–19, Singapore, 1997 [9].

* Corresponding author.

E-mail addresses: cicerone@infolab.ing.univaq.it (S. Cicerone), gabriele@infolab.ing.univaq.it (G.D. Stefano)

and basic algorithmic problems. Some of these problems require deciding whether a graph G belongs to a special class or not, or ask to compute some invariants such as the independence number $\alpha(G)$, the clique covering number $\theta(G)$, the clique number $\omega(G)$, and the chromatic number $\chi(G)$.

In the class of perfect graphs [18], the equality $\alpha(G) = \theta(G)$ holds and, as a consequence of the well-known Perfect Graph Theorem [24], also $\omega(G) = \chi(G)$ holds.

In [19], Grötschel, Lovász and Schrijver showed that the ellipsoid method for solving linear programming problems can be applied to obtain a polynomial algorithm for the maximum weighted independent set and the maximum weighted clique problem for perfect graphs, although the same authors pointed out that the proposed method is not intended to compete with the special purpose algorithms designed to solve these problems for special classes of perfect graphs.

On the other hand, to efficiently recognize a perfect graph is still an open problem. This problem has been solved for many interesting subclasses of the class of perfect graph: comparability graphs, cographs, permutation graphs, interval graphs and distance-hereditary graphs, to name just a few. All these graph classes are used in a wide spectrum of practical applications in different areas of computer science and operations research. For example, distance-hereditary graphs (in which all the induced paths joining any pair of vertices have the same length) have been used to design interconnection network topologies [11,16,17].

A fairly natural generalization of both distance-hereditary graphs and bipartite graphs are parity graphs. A graph is called *parity graph* if and only if, for any pair of vertices, the lengths of all the induced paths joining them have the same parity. Parity graphs are perfect [28], and in fact they are a subclass of Meyniel graphs [25]. They have been extensively studied, and several characterizations and properties have been discovered. Bandelt and Mulder [5] gave characterizations of parity graphs in term of forbidden isometric subgraphs and of distance functions. Burlet and Uhry [8] noticed that parity graphs can be obtained from a single node by applying construction rules, namely creation of true and false twins, and extension by bipartite graphs. Moreover, they observed that parity graphs have an interesting breadth first search layering structure, and, based on such a structure, they gave the first recognition algorithm for parity graphs working in $O(n^2)$ time, as well as algorithms to solve maximum weighted clique problem and maximum weighted independent set problem, both taking $O(n^3)$ time.

From then on, some attempts have been done to improve the recognition time for parity graphs [1,14,27]. It is important to remark that all these results are based on the same characterization given by Burlet and Uhry and that they provide parallel algorithms. The best parallel recognition algorithm takes $O(\log^2 n)$ with $O(n+m)$ processors [14]. In [27] an $O(n^2)$ sequential algorithm to compute the clique number for parity graphs is also devised.

Significant improvements of algorithmic solutions are often obtained as a consequence of a better understanding of the structural properties of the graphs under consideration. Hence, in this paper, we give a new characterization of parity graphs based

on split decomposition. Split decomposition was introduced by Cunningham [13] to generalize the well-understood substitution decomposition theory, whose applications in discrete mathematics are widely recognized [26]. In particular, we prove that split decomposition returns exactly cliques and bipartite graphs as building blocks of parity graph. This result, together with the observation that the split decomposition process can be performed in linear time [15], give us a promising technique for solving combinatorial (optimization) problems in the class of parity graphs.

In particular, the following algorithmic results have been achieved:

1. A linear time algorithm that solves the recognition problem for parity graphs. This improves the result shown in [8] that requires $O(n^2)$ time;
2. A linear time algorithm that solves the maximum weighted clique number problem for parity graphs. This improves the result shown in [27] that requires $O(n^2)$ time.
3. An algorithm that solves the maximum weighted independent set problem for parity graphs. This result is based on a technique discussed in [13]. The algorithm takes the same time complexity ($O(n^{2.5})$) of the best algorithm solving the same problem in bipartite graphs ([22], by matching). This result improves the result shown in [8] that requires $O(n^3)$ time.

A remarkable consequence of these results is that the extension of bipartite graphs to parity graphs does not increase the complexity of the basic problems we have considered, since the worst case occurs when the parity graph under consideration is an undecomposable bipartite graph.

The remainder of this paper is organized as follows. In Section 2, we give the notation used in the paper and recall the basic concepts regarding split decomposition. In Section 3, the new characterization for parity graphs is given, and in Section 4 we use such a characterization to develop algorithms for the recognition problem, the maximum weighted clique problem, and the maximum weighted independent set problem.

2. Notation and basic concepts

In this work we consider finite, simple, loopless, undirected graphs, $G = (V, E)$ with vertex set V and edge set E . We use standard terminologies from [21], some of which are slightly reviewed here.

A *subgraph* of G is a graph having all its vertices and edges in G . Given a subset S of V , the *induced subgraph* $\langle S \rangle$ of G is the maximal subgraph of G with vertex set S . On the other hand, the removal of S from G results in a graph denoted by $G - S$, that is, the induced subgraph $\langle V \setminus S \rangle$ of G , and we write $G - x$ when $S = \{x\}$. S is *independent* if $\langle S \rangle$ has no edges. By $|G|$ we denote the cardinality of V .

If T and S are subsets of V , by $N_T(S)$ we denote the *neighborhood* of S in T , that is, the set of vertices in T that are adjacent to some vertex in S , and with $N_T[S] = N_T(S) \cup S$ the *closed neighborhood* of S in T . For the sake of simplicity, we omit T if T coincides with V , and we write $N(x)$ and $N[x]$ when $S = \{x\}$.

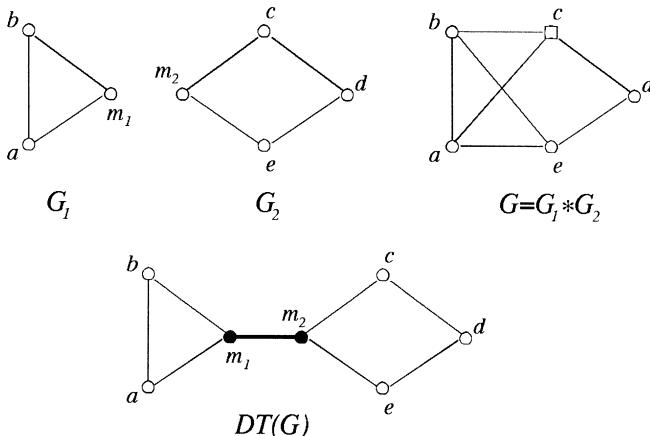


Fig. 1. The split composition operation $G = G_1 * G_2$ and the decomposition tree $DT(G)$. Graphs G_1 and G_2 are the components of $D(G)$.

Two vertices x and y are *twins* if they have the same neighborhood; we distinguish between *false twins* when $N(x) = N(y)$ and *true twins* when $N[x] = N[y]$.

A sequence of pairwise distinct vertices (x_0, \dots, x_k) is *path* in G if $(x_i, x_{i+1}) \in E$ for $i = 0, \dots, k - 1$. A path (x_0, \dots, x_k) is an *induced path* if and only if $\langle \{x_0, \dots, x_k\} \rangle$ has k edges. A graph G is *connected* if and only if for each pair of vertices x and y of G there is a path from x to y in G . A graph G is *bipartite* if its vertex set can be partitioned into two subsets V_1 and V_2 such that every edge in E joins V_1 to V_2 . A *clique* K_n is a graph having every pair of its n vertices adjacent. A *star* is a connected bipartite graph with $|V_1| = 1$, and it is denoted by $\{x; x_1, x_2, \dots, x_n\}$ when $V_1 = \{x\}$ and $V_2 = \{x_1, x_2, \dots, x_n\}$.

We now recall the split decomposition terminology defined by Cunningham [13] and also introduce some new terms. Let G_1 , G_2 be graphs having vertex sets $V_1 \cup \{m_1\}$, $V_2 \cup \{m_2\}$ and edge sets E_1 , E_2 , respectively, where $\{V_1, V_2\}$ is a partition of V and $m_1, m_2 \notin V$. The *split composition* of G_1 and G_2 is the graph $G = G_1 * G_2$ having vertex set V and edge set $E = E'_1 \cup E'_2 \cup \{(x, y) | x \in N(m_1), y \in N(m_2)\}$, where $E'_i = \{(x, y) \in E_i | x, y \in V_i\}$ for $i = 1, 2$. An example of split composition is shown in Fig. 1.

Split composition has an inverse operation. If $G = G_1 * G_2$ and $|V_1|, |V_2| \geq 2$, then we say that $\{G_1, G_2\}$ is a *simple decomposition* of G . We call $\{V_1, V_2\}$ the *split* of G associated with the simple decomposition $\{G_1, G_2\}$, and m_1 , m_2 the associated *marked* vertices. In other words, a split of $G = (V, E)$ is a partition of V into two subsets V_1 and V_2 with at least two elements, such that all vertices in V_1 that have neighborhoods in V_2 have the same neighborhoods in V_2 .

The *split decomposition* of a graph G is the set $D(G)$ of graphs obtained by the following recursive procedure:

- if G has a split $\{V_1, V_2\}$, then apply the split decomposition to graphs G_1 and G_2 obtained by the simple decomposition $\{G_1, G_2\}$.
- If G does not have a split then G is called *prime*.

Each element of $D(G)$ is called *component*. An important property of simple decomposition is that every component of $D(G)$ is isomorphic to an induced subgraph of G . For example, graph G_2 in Fig. 1 is isomorphic to the induced subgraph $\langle \{a, c, d, e\} \rangle$ of G .

Though cliques and stars are not uniquely split decomposable, Cunningham proved the following uniqueness result:

Theorem 2.1 (Cunningham [13]). *Each connected graph has a unique split decomposition into prime graphs, stars, and cliques with a minimum number of components.*

From now on, $D(G)$ will denote the unique split decomposition of a graph G . In this paper we also use the version of Bouchet [6] of simple decomposition of a graph G , where marked vertices m_1 and m_2 are joined by a marked edge. In this way, we can also consider the *decomposition tree* $DT(G)$ of G , whose vertices are the components of $D(G)$ and whose edges are the marked edges (see Fig. 1).

It is easy to verify that, for a generic graph G , the total number of vertices at the end of the decomposition process cannot be greater than $3(n - 2)$, where $n = |G|$. This maximum value can be achieved when G is a distance-hereditary graph, and, for this reason, these graphs are also called *completely separable graphs* [20].

3. Parity graphs and split decomposition

The basic definition of parity graphs states that a graph is a parity graph if and only if for any pair of vertices the lengths of all the induced paths joining them have the same parity. This definition also shows that the class of parity graphs is hereditary, that is, it is closed under induced subgraphs. In the following, we show that parity graphs have a useful characterization based on split decomposition. Let us review the recursive definition of parity graphs via generative rules.

Definition 3.1. Let G be a graph, x a vertex of G , and $\{x_1, x_2, \dots, x_n\}$ a set of false twins of G . Let B be a bipartite graph with vertex set $V_1 \cup V_2$. We introduce the following three operations:

1. $\beta(G, x)$ extends G by creating a true twin of x .
2. $\gamma(G, x)$ extends G by creating a false twin of x .
3. $\phi(G, B, \{x_1, x_2, \dots, x_n\})$ extends G by the bipartite graph B , identifying certain vertices of V_1 with the set of false twins $\{x_1, x_2, \dots, x_n\}$ (possibly $n = 1$).

Vertex x in cases 1 and 2, and vertices x_1, x_2, \dots, x_n in case 3 are called *extension vertices* of the corresponding operation.

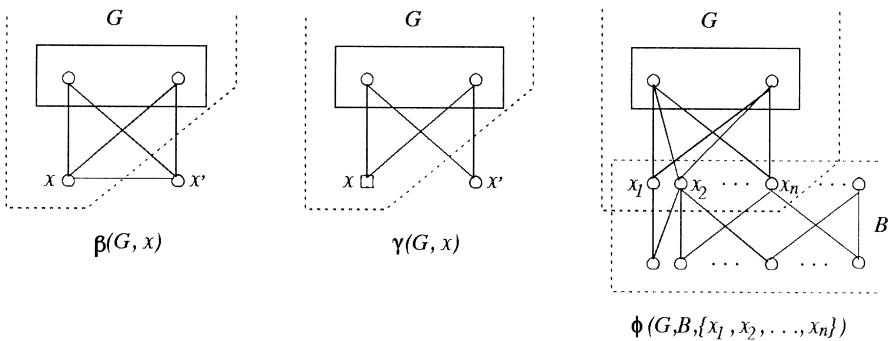
Fig. 2. Extension of a graph G by β , γ and ϕ operations.

Fig. 2 shows how to extend a graph G by β , γ and ϕ when the extension vertices are x and x_1, x_2, \dots, x_n . Without loss of generality, we assume that every bipartite graph B used by ϕ operations is connected, and also that $|B| \geq 2$. Moreover, in this paper we only consider connected parity graphs, since the extension of the obtained results to the disconnected case is straightforward.

Operations in the above definition have been used to characterize in a generative way the following subclasses of parity graphs:

- distance-hereditary graphs (where ϕ adds only K_2) [4];
- cographs (where ϕ is not used) [12];
- ptolemaic graphs (where ϕ adds K_2 and γ is applied only to vertices whose neighborhoods are nonempty complete subgraphs) [23];
- (6,2)-chordal graphs (where ϕ adds K_2 and β is not used) [2].

Burlet and Uhry proved the following theorem.

Theorem 3.1 (Burlet and Uhry [8]). *Every connected parity graph is obtained from a single vertex by applying operations β , γ , and ϕ successively and in any order.*

As a consequence of this theorem, if G is a parity graph, then there exists a sequence of operations $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$ such that $G_0 = K_1$, $G_i = \sigma_i(G_{i-1})$, $\sigma_i \in \{\gamma, \beta, \phi\}$ for all $i = 1, 2, \dots, n$, and $G_n = G$. Notice that, in general, there are several different sequences of operations that generate the same parity graph and that these operations can be replaced by composition of clique and bipartite graphs (see also [7], where the authors noticed this fact by studying a restriction of the *amalgam* operation used to build Meyniel graphs). Our results have been obtained by investigating the consequences of applying both composition and decomposition operations to parity graphs.

The following lemma can be used to decide whether a generative operation induces a split in the obtained graph.

Lemma 3.2. Let $G = (V, E)$ be a connected graph and $\sigma \in \{\beta, \gamma, \phi\}$. Let $G' = \sigma(G)$ and $X \subseteq V$ be the corresponding set of extension vertices. If $|G - X| \geq 2$ then G' has a split.

Proof. Let $\sigma \in \{\beta, \gamma\}$. In this case $X = \{x\}$ and x' is the twin of x created by σ . Then $\{V \setminus \{x\}, \{x, x'\}\}$ is a split of G' , because $|G - \{x\}| \geq 2$ by hypothesis and vertices x and x' have the same neighborhoods in $G - \{x\}$ by definition of twins.

On the other hand, if $G' = \phi(G, B, X)$ then $\{G - X, B\}$ is a simple decomposition of G' (see Fig. 2). In fact, $|G - X| \geq 2$ by hypothesis and $|B| \geq 2$ by assumption on ϕ operation. Moreover, X contains all the vertices in B that have neighborhoods in $G - X$, and, since vertices in X are false twins in G , any vertex in X have the same neighborhoods in $G - X$. This proves that G' has a split. \square

The above lemma and the following theorem are used to prove the new characterization of parity graphs given in Theorem 3.4.

Theorem 3.3. *The class of parity graphs is closed under split composition.*

Proof. Let G_1, G_2 be parity graphs having vertex sets $V_1 \cup \{m_1\}$, $V_2 \cup \{m_2\}$, respectively, where $\{V_1, V_2\}$ is a partition of V and $m_1, m_2 \notin V$. Let $G = G_1 * G_2$ be the split composition of G_1 and G_2 having vertex set V . We prove that G is a parity graph by showing that all the induced paths from a given vertex $a \in V_1$ to a given vertex $b \in V_2$ have the same parity.

Let us denote by $P(x, y)$ the parity of any induced path between vertices x and y , and by $\bar{P}(x, y)$ the opposite parity. Since G_1 is a parity graph, then all the induced paths between a and m_1 have the same parity. This implies that all the possible induced paths between a and all vertices in $N(m_1)$ have the same parity, that is

$$P(a, m_1) = \bar{P}(a, x) \quad \text{for each } x \in N(m_1).$$

Symmetrically, the same relation holds in G_2 with respect to a given vertex $b \in V_2$:

$$P(b, m_2) = \bar{P}(b, y) \quad \text{for each } y \in N(m_2).$$

By definition of split composition, every induced path in G from a vertex $a \in V_1$ to a vertex $b \in V_2$ has the form $(a, \dots, x, y, \dots, b)$, where:

- (a, \dots, x) is an induced path in $\langle V_1 \rangle$ and $x \in N(m_1)$;
- (y, \dots, b) is an induced path in $\langle V_2 \rangle$ and $y \in N(m_2)$;
- the edge (x, y) exists in G .

This shows that all the induced paths between a and b have the same parity, that is, if $P(a, x) = P(y, b)$ then $P(a, b)$ is odd, otherwise $P(a, b)$ is even. This proves that G is a parity graph. \square

In [10] the same authors introduced a family of graph classes forming an infinite lattice with respect to inclusion, whose top and bottom elements are parity graphs and distance-hereditary graphs, respectively. Each defined class differs from the others in that ϕ adds graphs belonging to a specific subclass of bipartite graphs. In particular, distance-hereditary graphs are obtained by adding K_2 , and parity graphs are obtained by adding any bipartite graph. In the same paper, a general theorem based on split

decomposition characterizes each class in the family, and the following result could be also thought as a consequence of this general theorem applied to parity graphs.

Theorem 3.4. *Let G be a connected graph and $D(G)$ its split decomposition. G is a parity graph if and only if every component in $D(G)$ is bipartite or is a clique.*

Proof. (*If case*). Since every component in $D(G)$ is bipartite or is a clique then $D(G)$ contains only parity graphs. To prove that G is a parity graph it is sufficient (by Theorem 3.3) to apply the split composition to the components in $D(G)$ until G itself is rebuilt.

(*Only if case*). By Theorem 2.1, the split decomposition components in $D(G)$ are prime graphs, stars and cliques. Since stars are bipartite graphs, we have to prove that every prime component $G_i \in D(G)$ is bipartite. If $|G_i| \leq 3$ then G_i is prime and hence is bipartite or a clique. Let us suppose that $|G_i| \geq 4$. G_i has no twins, otherwise Lemma 3.2 would imply that G_i has a split. This fact, and the observation that G_i is a parity graph, proves that ϕ is the last operation in every generative sequence for G_i . This implies

$$G_i = \phi(G', B, \{x_1, x_2, \dots, x_n\}).$$

Using again Lemma 3.2, we obtain that $|G' - \{x_1, x_2, \dots, x_n\}| = 1$, otherwise G_i would have a split. Then G' is a star $\{x; x_1, x_2, \dots, x_n\}$. By definition of ϕ operation, the extension of a star by a bipartite graph trivially produces a bipartite graph. \square

4. Basic problems on parity graphs

Theorem 3.4 provides a basis to devise algorithms solving basic combinatorial problems on parity graphs in a straightforward way. In this section we provide three algorithms to solve the recognition problem, and to calculate the maximum weighted clique, and the maximum weighted independent set.

The solutions of the last two optimization problems also provide the clique number $\omega(G)$ and the independence number $\alpha(G)$, of a parity graph G . Moreover, recalling that parity graphs are perfect, the same algorithms supply a solution to the chromatic number $\chi(G)$ and clique covering number $\theta(G)$ problems, respectively.

4.1. Recognition problem

Problem. Given a graph $G = (V, E)$ and a graph class, the *recognition problem* consists in verifying whether G belongs to the class or not.

Algorithm. The characterization given in Theorem 3.4 suggests solving this problem by decomposing G and verifying that all the components are clique or bipartite graphs.

The following procedure performs this test.

```

procedure TEST;
input: a graph  $G$ .
output: true if and only if  $G$  is a parity graph.
1. begin
2.   Compute  $D(G)$ 
3.   bool=true
4.   for each  $G_i \in D(G)$ 
5.     if  $G_i$  is not clique and  $G_i$  is not bipartite then bool=false
6.   return(bool)
7. end
```

Complexity. The split decomposition can be performed in linear time [15] (step 2). Since every element of $D(G)$ can be verified in linear time on the size of the component (step 5), then testing all the components (step 4) takes linear time in the size of the input graph G . Hence, the following theorem can be stated.

Theorem 4.1. *Let G be a graph. It can be decided in linear time whether G is a parity graph.*

This result represents the optimal solution for the recognition problem of parity graphs, and improves the result achieved by Burlet and Uhry in [8] (a sequential algorithm working in $O(n^2)$ time).

4.2. Maximum weighted clique

Problem. Given a graph $G = (V, E)$ and a weight $w_v \in Z^+$ for all $v \in V$ (Z^+ is the set of positive integers), the *maximum weighted clique problem* consists in finding a clique subgraph of G such that the sum of the weights of its vertices (denoted $\omega_w(G)$) is as large as possible. The *clique problem* is the unweighted version, and in turn it consists in finding the clique having the largest number $\omega(G)$ of vertices.

The following result, based on Theorem 3.4, shows how to calculate the clique number $\omega(G)$ for a parity graph G .

Lemma 4.2. *Let G be a parity graph. Then*

$$\omega(G) = \max\{\omega(G_i) \mid G_i \in D(G)\}.$$

Proof. By Theorem 3.4, we know that every component in $D(G)$ is a clique or bipartite. If all the components in $D(G)$ are bipartite, G itself is a bipartite graph and then $\omega(G) = \max\{\omega(G_i) \mid G_i \in D(G)\} = 2$.

Let us suppose that G_i and G_j are two components of $D(G)$ such that G_i is a clique, $\{G_i, G_j\}$ is a simple decomposition, and m_i, m_j are the corresponding marked vertices. In this case the component G_j must be a bipartite component, otherwise $G_i * G_j$ would be a clique and $D(G)$ would not be minimal, as stated in Theorem 2.1. Furthermore, since all the vertices in $N(m_j)$ are pairwise not adjacent in G , then G_i is a maximal clique in G . This implies that the largest clique in $D(G)$ is the maximum clique in G . \square

Notice that, since parity graphs are perfect, this lemma also provides the value of the chromatic number $\chi(G)$, that is the minimum number of colors needed to color all the vertices in such a way that no two adjacent vertices have the same color.

In the case of a weighted graph G , Lemma 4.2 cannot be used to compute $\omega_w(G)$ directly, because marked vertices are unweighted. This problem can be overcome in the following way.

Let us consider a graph G having a split $\{V_1, V_2\}$. A simple decomposition based on such a split decomposes G into two induced subgraphs G_1 and G_2 whose vertex sets are $V_1 \cup \{m_1\}$ and $V_2 \cup \{m_2\}$, respectively. By definition of split, the marked vertex m_1 in G_1 plays the role of all the vertices in $N(m_2)$, that is, the neighborhoods of the marked vertex in G_2 . Symmetrically, the marked vertex in G_2 plays the role of all the vertices in $N(m_1)$. For example, in Fig. 1 marked vertex in G_1 can be viewed as either vertex c or vertex e , and marked vertex in G_2 can be viewed as either vertices b or a . If G has weighted vertices, in order to compute $\omega_w(G)$, every marked vertex has to play the role of the vertex with the largest weight. In this way, the clique component corresponds to an induced maximal weighted clique in G . So, $\max\{w_x \mid x \in N(m_2)\}$ and $\max\{w_x \mid x \in N(m_1)\}$ have to be the weights of the marked vertices m_1 and m_2 , respectively. It is easy to see that the weights of all the marked vertices can be computed during the decomposition process without increasing the final time bound. In Fig. 3 it is shown the decomposition process of a weighted graph.

Algorithm. The following procedure computes the maximum weighted clique set and the maximum weighted clique number taking as input a parity graph G , by calculating the $DT(G)$ having weights on marked vertices as stated before. The basic ideas of this algorithm are also similar to those used by Cunningham in [13].

```

procedure CLIQUE;
input: a parity graph  $G$ .
output: the maximum weighted clique and  $\omega_w(G)$ 

```

1. **begin**
2. Compute $DT(G)$ with weights on marked vertices
3. $m = 0$
4. **for each** $G_i \in D(G)$
5. **if** G_i is clique
6. **then** $G' = G_i$; $k = \sum_{v \in G_i} w_v$

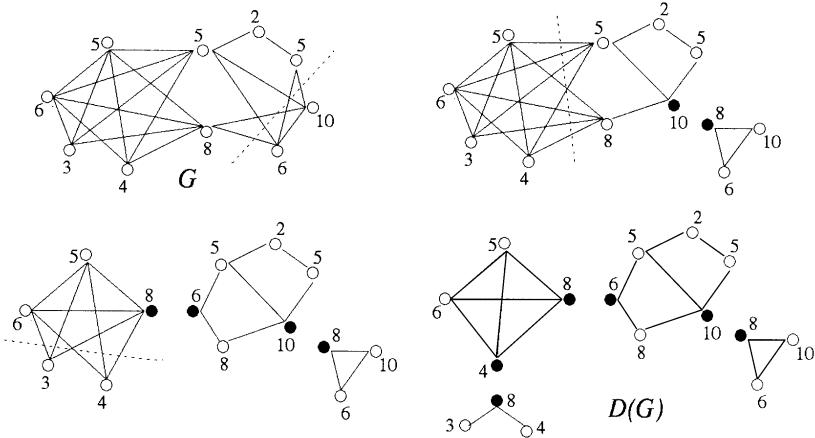


Fig. 3. The split decomposition process of a weighted parity graph G . Each simple decomposition gives rise to a pair (m_1, m_2) of marked vertices whose weights are $w_{m_1} = \max\{w_x \mid x \in N(m_2)\}$ and $w_{m_2} = \max\{w_x \mid x \in N(m_1)\}$, respectively. It is clear from the resulting $D(G)$ that $\omega_w(G) = 24$ and $\omega(G) = 4$.

```

7.      else  $G' =$  the induced  $K_2$  of  $G_i$  s.t.  $k = \omega_w(G')$  is maximal
8.      if  $m < k$  then  $m = k$ ;  $H = G'$ 
9.      return  $(H, m)$ 
10. end

```

The CLIQUE procedure, after calculating the weighted decomposition tree (step 2), tests each component (step 4) by calculating the relative maximum weighted clique and maximum weighted clique number both when the component is a clique (step 6) and when it is a bipartite graph (step 7). At the same time, the procedure (step 8) keeps memory (by variables m and H) of the best result achieved after each step of the cycle.

Complexity. The decomposition tree can be calculated in linear time [15] (step 2). Since the maximum weighted clique and the maximum weighted clique number of every component in $DT(G)$ can be computed in linear time on the size of the component (steps 6 and 7), then visiting all the components (cycle at step 4) takes linear time in the size of the input graph G . Hence, the following theorem can be stated.

Theorem 4.3. *Let $G = (V, E)$ be a parity graph having a weight $w_v \in \mathbb{Z}^+$ for all $v \in V$. The maximum weighted clique problem can be solved in linear time.*

From the previous result it follows that $\omega_w(G)$, and the clique number $\omega(G)$ can be calculated in linear time. Furthermore, as parity graphs are perfect, also the chromatic number $\chi(G)$ can be calculated in linear time. This improves the result shown in [27] requiring $O(n^2)$ time.

4.3. Maximum weighted independent set

Problem. Given a graph $G=(V,E)$ and a weight $w_v \in Z^+$ for all $v \in V$, the *maximum weighted independent set problem* consists in finding an independent set $\text{Mis}(G) \subseteq V$ such that the sum of the weights (denoted by $\alpha_w(G)$) of its vertices is as large as possible. The *independence number problem* is the unweighted version, and it consists in finding the largest number of independent vertices in G . The solution of the independence number problem is denoted by $\alpha(G)$.

We prove now that, when G is a parity graph, $\text{Mis}(G)$ can be efficiently computed using the characterization given in Theorem 3.4. We recall in the following lemma a general technique discussed in [13] to solve the same problem in any split decomposable graph.

Lemma 4.4 (Cunningham [13]). *Let $\{G_1, G_2\}$ be a simple decomposition of a graph G , and (m_1, m_2) the marked edge connecting G_1 to G_2 . Assuming that $w_{m_1} = \alpha_w(G_2 - m_2) - \alpha_w(G_2 - N[m_2])$, then*

$$\text{Mis}(G) = \begin{cases} \text{Mis}(G_1 - N[m_1]) \cup \text{Mis}(G_2 - m_2) & \text{if } m_1 \in \text{Mis}(G_1), \\ \text{Mis}(G_1 - m_1) \cup \text{Mis}(G_2 - N[m_2]) & \text{otherwise.} \end{cases}$$

This result can also be used to calculate $\text{Mis}(G)$ for a generic graph G when $|D(G)| > 2$. In fact, let us suppose that $\{G_1, G_2\}$ is a simple decomposition of G such that G_2 is prime. By Theorem 3.4, $\text{Mis}(G_2)$ can be efficiently computed (in linear time if G_2 is a clique, and in $|G_2|^{2.5}$ otherwise [22]). Now, if the weight of the marked vertex of G_1 is assigned as in the previous lemma, then $\text{Mis}(G)$ can be calculated by recursively computing $\text{Mis}(G_1)$, and by joining this partial solution with $\text{Mis}(G_2)$ following the criteria of Lemma 4.4.

Algorithm. For a given parity graph G , the following recursive procedure computes $\text{Mis}(G)$. It takes the decomposition tree $DT(G)$ of G as input, and calls the subroutine `mis_comp`. Such a subroutine takes a component $G_2 \in D(G)$ as input, verifies whether G_2 is bipartite or a clique, and computes the relative maximum weighted independent set.

procedure STABLE;

input: $DT(G)$, the decomposition tree of a weighted graph G .

output: $\text{Mis}(G)$.

1. **if** $|D(G)| = 1$
2. **then return** (`mis_comp($DT(G)$)`)
3. **else**
4. **begin**
5. Let G_2 be a leaf of $DT(G)$
6. Let (m_1, m_2) be the edge connecting a component G_1 of $DT(G)$ to G_2
7. $S_1 = \text{mis_comp}(G_2 - m_2)$

```

8.    $S_2 = \text{mis\_comp}(G_2 - N[m_2])$ 
9.    $w_{m_1} = \sum_{v \in S_1} w_v - \sum_{v \in S_2} w_v$ 
10.   $M = \text{STABLE}(DT(G) - G_2)$ 
11.  if  $m_1 \in M$ 
12.    then return( $M \setminus \{m_1\} \cup S_1$ )
13.    else return( $M \cup S_2$ )
14.  end
15. end

```

For example, the procedure STABLE applied to the decomposition tree of graph G shown in Fig. 3 returns that $\alpha_w(G) = 19$, and that $\text{Mis}(G)$ contains vertices whose weights are 10, 4, 3, and 2, respectively.

Complexity. Let n be the number of vertices of G . The number of times that the procedure STABLE calls itself equals the number of components in $DT(G)$. Each time the procedure is executed it calls two times the procedure mis_comp. The procedure mis_comp requires linear time if applied to a clique component, and $O(n_i^{2.5})$ time if applied to a bipartite component having size n_i (by using the Hopcroft–Karp matching algorithm [22]). Then, to visit all the components, the STABLE procedure takes $O(n^{2.5})$ time. Hence, the following theorem holds.

Theorem 4.5. *Let $G = (V, E)$ be a parity graph having a weight $w_v \in Z^+$ for all the n vertices $v \in V$. The weighted independent set problem can be computed in $O(n^{2.5})$ time.*

From this result, it follows that $\alpha_w(G)$ and the independence number $\alpha(G)$, which coincides with the clique covering number $\theta(G)$, can be calculated in $O(n^{2.5})$ time when G is a parity graph. This improves the result shown in [8] requiring $O(n^3)$ time.

Notice that the procedure STABLE can also be applied to distance-hereditary graphs. We know that G is a distance-hereditary graph if and only if each component of $D(G)$ is a star or a clique [10,3]; in both cases the procedure mis_comp requires linear time. This implies that the main work to compute $\text{Mis}(G)$ is due to the split decomposition, that is, $\text{Mis}(G)$ can be computed in linear time. This result equals the result achieved by Hammer and Maffray in [20].

5. Conclusions and future works

This work provides a new characterization of parity graphs based on split decomposition. In particular, we proved that the split decomposition returns exactly, as building blocks of parity graphs, cliques and bipartite graphs. Since the split decomposition process can be performed in linear time [15], this characterization supplies a new technique for solving combinatorial problems in the class of parity graphs.

By using the new characterization, the following algorithmic results have been achieved: a linear time algorithm that solves the *recognition* problem, a linear time algorithm that solves the *maximum weighted clique number* problem, and an algorithm that solves the *maximum weighted independent set* problem taking $O(n^{2.5})$.

Not only do these algorithmic results improve the best known solutions, but they also reduce the computational complexity of the considered problem for parity graphs to the computational complexity of the same problem for bipartite graphs. In other words, the extension of bipartite graphs to parity graphs does not increase the complexity of the combinatorial problems we have considered, since the worst case occurs when the parity graph is an undecomposable bipartite graph.

As this characterization provides a better understanding of the structural properties of parity graphs, we believe that it might be also applied to improve the computational time bounds of other combinatorial problems. Other optimization problems on parity graphs may also be reducible to problems on bipartite graphs.

References

- [1] G.S. Adhar, S. Peng, Parallel algorithms for cographs and parity graphs with applications, *J. Algorithms* 11 (2) (1990) 252–284.
- [2] G. Ausiello, A. D'Atri, M. Moscarini, Chordality properties on graphs and minimal conceptual connection in semantic data models, *J. Comput. System Sci.* 33 (2) (1986) 179–202.
- [3] H.J. Bandelt, A. D'Atri, M. Moscarini, H.M. Mulder, A. Schultze, Operations on distance hereditary graphs, Technical Report 226, CNR, Istituto di Analisi dei Sistemi e Informatica del CNR, Rome, Italy, 1988.
- [4] H.J. Bandelt, M. Mulder, Distance-hereditary graphs, *J. Combin. Theory, Ser. B* 41 (2) (1986) 182–208.
- [5] H.J. Bandelt, M. Mulder, Metric characterization of parity graphs, *Discrete Math.* 91 (1991) 221–230.
- [6] A. Bouchet, Transforming trees by successive local complementations, *J. Graph Theory* 4 (1988) 196–207.
- [7] M. Burlet, J. Fonlupt, Polynomial algorithm to recognize meyniel graphs, in: C. Berge, V. Chvátal (Eds.), *Topics on Perfect Graphs*, number 21 Ann. *Discrete Math.* North-Holland, Amsterdam, 1984, pp. 225–252.
- [8] M. Burlet, J.P. Uhry, Parity graphs, in: C. Berge, V. Chvátal (Eds.), *Topics on Perfect Graphs*, number 21 Ann. *Discrete Math.* North-Holland, Amsterdam, 1984, pp. 253–277.
- [9] S. Cicerone, G. Di Stefano, On the equivalence in complexity among basic problems on bipartite and parity graphs, in: eighth International Symposium on Algorithms and Computation (ISAAC'97), vol. 1350, *Lecture Notes in Computer Science*, Springer, Berlin, December 1997, pp. 354–363.
- [10] S. Cicerone, G. Di Stefano, Graph classes between parity and distance-hereditary graphs, in: First Discrete Mathematics and Theoretical Computer Science (DMTCS'96), Combinatorics, Complexity, and Logic, Springer, Berlin, 1996, pp. 168–181.
- [11] S. Cicerone, G. Di Stefano, Port and node support to compact routing, Technical Report R.97-17, Dipartimento di Ingegneria Elettrica, Università di L'Aquila, L'Aquila, Italy, 1997.
- [12] D.G. Corneil, H. Lerchs, L. Stewart-Burlingham, Complement reducible graphs, *Discrete Appl. Math.* 3 (1981) 163–174.
- [13] W.H. Cunningham, Decomposition of directed graphs, *SIAM J. Alg. Disc. Meth.* 3 (1982) 214–228.
- [14] E. Dahlhaus, An efficient parallel recognition algorithm for parity graphs, in: O. Abou-Rabia et al. (Eds.), *ICCI '93*, 1993, pp. 82–86.
- [15] E. Dahlhaus, Efficient parallel and linear time sequential split decomposition, in: 14th Foundations of Software Technology and Theoretical Computer Science (FSTTCS'94), vol. 880, *Lecture Notes in Computer Science*, 1994.

- [16] G. Di Stefano, A routing algorithm for networks based on distance-hereditary topologies, in: Third International Colloquium on Structural Information and Communication Complexity (SIROCCO'96), 1996.
- [17] A.H. Esfahanian, O.R. Oellermann, Distance-hereditary graphs and multideestination message-routing in multicompilers, *J. Combin. Math. Combin. Comput.* 13 (1993) 213–222.
- [18] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, NY, 1980.
- [19] M. Grötschel, L. Lovász, A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica* 1 (1981) 169–197.
- [20] P.L. Hammer, F. Maffray, Completely separable graphs, *Discrete Appl. Math.* 27 (1990) 85–99.
- [21] F. Harary, Graph Theory, Addison-Wesley, Reading, MA, 1969.
- [22] J.E. Hopcroft, R.M. Karp, An $O(n^{5/2})$ algorithm for maximum matchings in bipartite graphs, *SIAM J. Comput.* 2 (4) (1973) 225–231.
- [23] E. Howorka, A characterization of ptolemaic graphs, *J. Graph Theory* 5 (1981) 323–331.
- [24] L. Lovasz, A characterization of perfect graphs, *J. Combin. Theory Ser. B* 13 (1972) 95–98.
- [25] H. Meyniel, The graphs whose odd cicles have at least two cords, in: C. Berge, V. Chvátal (Eds.), Topics on Perfect Graphs, number 21 Ann. Discrete Math., North-Holland, Amsterdam, 1984, pp. 115–119.
- [26] R.H. Möhring, F.J. Radermacher, Substitution decomposition for discrete structure and connection with combinatorial optimization, *Ann. Discrete Math.* 19 (1984) 257–356.
- [27] T. Przytycka, D.G. Corneil, Parallel algorithms for parity graphs, *J. Algorithms* 12 (1) (1991) 96–109.
- [28] H. Sachs, On the berge conjecture concerning perfect graphs, in: Combinatorial Structures and their Applications, Gordon and Breach, New York, 1970, pp. 377–384.