Information Processing Letters 114 (2014) 524–528

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Fast algorithms for some dominating induced matching problems

Min Chih Lin [a], Michel J. Mizrahi [a,*], Jayme L. Szwarcfiter [b,c]

[a] Departamento de Computación, Universidad de Buenos Aires, Argentina
[b] Instituto de Matemática, NCE and COPPE, Universidade Federal do Rio de Janeiro, Brazil
[c] Instituto Nacional de Metrologia, Qualidade e Tecnologia, Brazil

ARTICLE INFO

Article history:
Received 24 February 2014
Received in revised form 24 April 2014
Accepted 24 April 2014
Available online 1 May 2014
Communicated by B. Doerr

Keywords:
Algorithms
Dominating induced matchings
Graph theory

ABSTRACT

We describe $O(n)$ time algorithms for finding the minimum weighted dominating induced matching of chordal, dually chordal, biconvex, and claw-free graphs. For the first three classes, we prove tight $O(n)$ bounds on the maximum number of edges that a graph having a dominating induced matching may contain. By applying these bounds, and employing existing $O(n+m)$ time algorithms we show that they can be reduced to $O(n)$ time. For claw-free graphs, we describe a variation of the existing algorithm for solving the unweighted version of the problem, which decreases its complexity from $O(n^2)$ to $O(n)$, while additionally solving the weighted version. The same algorithm can be easily modified to count the number of DIM's of the given graph.

1. Introduction

We consider undirected simple graphs G, denoting by $V(G)$ and $E(G)$, respectively, the sets of vertices and edges of G, $n = |V(G)|$ and $m = |E(G)|$. For $v \in V(G)$, $N(v)$ represents the set of neighbors of $v \in V(G)$, while $N[v] = N(v) \cup \{v\}$. For $S \subseteq V(G)$, $N(S) = \cup_{v \in S} N(v)$. We say a vertex $v \in V(G)$ such that $N[v] = V(G)$ is *universal*. Denote by $G[S]$ the subgraph of G induced by the vertices of S. If $G[S]$ is a 0-regular graph then S is an *independent set*, if it is a 1-regular graph then S is the set of vertices of an *edge independent set*. By $G + H$ we denote the disjoint union of two graphs G and H. We say that a graph G is H-free if G does not contain H as an induced subgraph. A vertex v is called *simplicial* if all its neighbors are adjacent to each other. An edge independent set is also known as an *induced matching*. For convenience, we may write induced matching to refer either to the set of edges or to its corresponding vertex set. Finally, we also employ the notation *matching* with its usual meaning of a set of pairwise non-adjacent edges.

Say that an edge $e \in E(G)$ *dominates* itself and every other edge adjacent to it. An *edge dominating set* of G is a set of edges $E' \subseteq E(G)$, such that every $e \in E(G)$ is dominated by some edge of E'. If each $e \in E(G)$ is dominated by exactly one edge of E' then E' is an *efficient edge dominating set*. In the latter situation, E' defines an induced matching, while the set of vertices not incident to E' form an independent set. For this reason, an efficient edge dominating set is also called *dominating induced matching (DIM)*. Not every graph admits a DIM. The *DIM problem* is to determine whether a graph has such a matching, and is known to be NP-complete [9]. We will consider graphs G with a weighting Ω, that assigns to each edge $vw \in E(G)$ a non-negative finite weight $\omega(vw)$. The aim is to find the minimum weight of a dominating induced matching of G, if any. We name this problem as $DIM_{\Omega}(G)$. Some of the existing algorithms for solving DIM problems are [3–5,10].

* Corresponding author.
E-mail addresses: oscarlin@dc.uba.ar (M.C. Lin), michel.mizrahi@gmail.com (M.J. Mizrahi), jayme@nce.ufrj.br (J.L. Szwarcfiter).

http://dx.doi.org/10.1016/j.ipl.2014.04.012
0020-0190/

Since the number of edges of any DIM of G, if existing, is invariant, it is straightforward to generalize the problem for edges with negative weights too.

Following the definition, the DIM problem can be viewed as to decide whether there is a partition of the vertices into two sets (say a *coloring* of the vertices in *white* and *black*) such that the white set is an independent set while the black one induces a 1-regular graph. Moreover, the black set defines a DIM of the graph [7]. A coloring is *partial* if only part of the vertices of G have been assigned colors, otherwise it is *total*. A black vertex is *single* if it has no black neighbor, and is paired if it has exactly one black neighbor. Each coloring, partial or total, can be *valid* or *invalid*.

A partial coloring is valid whenever any two white vertices are non-adjacent and each black vertex is either paired, or is single having some uncolored neighbor. A total coloring is valid whenever any two white vertices are non-adjacent and each black vertex is paired.

A valid partial coloring Γ might possibly extend into a coloring $\Gamma' \supseteq \Gamma$ by iteratively applying a set of coloring rules, compatible with Γ. In general, such rules would color some uncolored vertex v, whose color is uniquely determined by the colors of Γ. For instance, any uncolored neighbor of a white vertex must be colored as black, otherwise the coloring would be invalid. See [7] for a set of such rules. We refer to this process as *propagation*.

We prove that any chordal graph containing a DIM has at most $2n-3$ edges. Counting the edges and applying the $O(n+m)$ time algorithm by Lu, Ko and Tang [11] lead to an $O(n)$ time algorithm. For dually chordal graphs, by employing the similarity result *chordal - dually chordal* for DIM's by Brandstädt, Leitert and Rautenbach [2] also leads to solving the DIM problem in $O(n)$ time. For biconvex graphs, we prove that any $K_{3,3}$-free convex graph contains at most $2n-4$ edges. Additionally, that any biconvex graph containing a DIM is $K_{3,3}$-free. Using these two results, counting the number of edges of the given graph and employing the $O(n+m)$ time algorithm by Brandstädt, Hundt and Nevries [1] leads to solving the DIM problem for biconvex graphs in $O(n)$ time. Finally, for claw-free graphs, we describe a variation of the algorithm by Cardoso, Korpelainen and Lozin [7]. The latter solves the DIM problem, without weights, in $O(n^2)$ time, while the presently proposed algorithm requires $O(n)$ time for solving $DIM_\Omega(G)$.

A conference version of this paper has been presented at *LATIN' 2014* [6].

2. Chordal, dually chordal and biconvex graphs

In this section, we remark that computing $DIM_\Omega(G)$ for any graph G which is chordal, dually chordal or biconvex requires no more than $O(n)$ time.

Lemma 1. *(See [1].) If G contains a K_4 then G has no DIM's.*

Lemma 2. *Every K_4-free chordal graph G with at least 2 vertices has at most $2n-3$ edges. The bound is tight even if G is an interval graph.*

Proof. By induction on the number of vertices. For $n=2$, the result is trivial. Suppose the bound is valid for graphs with $n-1$ vertices, $n \geq 3$. Let G be an n-vertex chordal graph and v a simplicial vertex of it. Since $|E(G)| = |E(G \setminus \{v\})| + d(v)$, where $d(v)$ denotes the degree of v, by the induction hypothesis, the number of edges of $G \setminus \{v\}$ is bounded by $2n-5$. Since G is K_4-free, $d(v) \leq 2$, therefore $|E(G)| \leq 2n-5+2 = 2n-3$.

An interval graph having two universal vertices and the remaining ones having degree 2 has no K_4 and contains $2n-3$ edges, meaning that the bound is tight for interval graphs. □

Corollary 3. *The $DIM_\Omega(G)$ problem can be solved in $O(n)$ time for (dually) chordal graphs.*

Proof. Let G be a given chordal graph. First, count the number of edges of G, up to a limit of $2n-3$. If the bound has been exceeded then stop answering that G has no DIM's. Otherwise, apply the algorithm [11] which solves $DIM_\Omega(G)$ in $O(n)$ time. Finally, if a graph has a DIM then it is chordal if and only if it is dually chordal [2]. Consequently, $DIM_\Omega(G)$ can also be solved in $O(n)$ time for dually chordal graphs. □

Next, consider solving $DIM_\Omega(G)$ for biconvex graphs.

An ordering $<$ of X in a bipartite graph $G = (X, Y, E)$ has the *interval property* if for every vertex $y \in Y$, the vertices of $N(y)$ are consecutive in the ordering $<$ of X. A bipartite graph (X, Y, E) is *convex* if there is an ordering of X or Y that fulfills the interval property. Furthermore if there are orderings for both X and Y which fulfill the interval property the graph is *biconvex*.

Lemma 4. *Let G be a convex bipartite graph having no subgraph isomorphic to $K_{3,3}$. Then G contains at most $2n-4$ edges, for $n \geq 3$.*

Proof. By induction on n. If $n=3$, the graph has at most 2 edges, satisfying the bound. Let G be an arbitrary $K_{3,3}$-free convex graph, v its minimum degree vertex and G' the graph obtained from G by removing v.

- $d(v) \leq 2$: Clearly, G' is also $K_{3,3}$-free. By inductive hypothesis, G' has at most $2n-6$ edges. Consequently, G has at most $2n-6+d(v) \leq 2n-4$ edges.
- $d(v) > 2$: Every vertex in G has degree at least 3. Let $G = (X, Y, E)$ where X has the interval property. Thus for each vertex $y \in Y, N(y)$ consists of vertices that are consecutive. Let $\{x_1, \ldots, x_k\}$ be the ordering $<$ of X and w.l.o.g. let $\{y_1, y_2, y_3\} \subseteq N(x_1)$. Since y_1, y_2, y_3 have at least 3 neighbors and X has the interval property, it follows that $\{x_2, x_3\} \subseteq N(y_1) \cap N(y_2) \cap N(y_3)$. Therefore $\{x_1, x_2, x_3, y_1, y_2, y_3\}$ induces a $K_{3,3}$, a contradiction.

Hence, G contains indeed at most $2n-4$ edges. This bound is tight, $K_{2,n-2}$ is an example. □

We remark that bipartite graphs, not necessarily convex, which do not contain $K_{3,3}$ as a minor also have at most $2n-4$ edges [8](note that [8] employs the term *H-free* with the meaning that G does not contain the graph H as a minor). However, this bound does not apply to general bipartite graphs not containing $K_{3,3}$ as an induced subgraph, as shown by the example below described.

Let $G=(X,Y,E)$ be a bipartite graph where $X=\{x_0,x_1,\dots,x_{15}\}$ and $Y=\{y_0,y_1,\dots,y_7\}$. Add the edge x_iy_{2j}, if the binary representation of i has the digit 0 at position j, while if such a binary representation contains the digit 1 at j then add the edge x_iy_{2j+1}. It is easy to see that one of the edges x_iy_{2j}, x_iy_{2j+1} will exist for all $i,j: 0\le i\le 15, 0\le j\le 3$. We show that G is $K_{3,3}$-free: Suppose this is not true, and let $\{x_i,x_j,x_k,y_p,y_q,y_r\}$ be the vertices of an induced $K_{3,3}$. By the construction of G the binary representations of i, j, k have the same value for positions $\lfloor\frac{p}{2}\rfloor$, $\lfloor\frac{q}{2}\rfloor$, $\lfloor\frac{r}{2}\rfloor$. But i, j, k are distinct integers $0\le i,j,k\le 15$, which leads to a contradiction, since there are no three integers smaller than 16 with the above property. Consequently, G is $K_{3,3}$-free. To complete the example, note that G has 24 vertices and more than 44 edges.

Consider k copies of the graph defined above. Say x^i_j is the x_j vertex from the i-th copy. Add edges $y^i_7x^{(i+1)}_0$, $0\le i<k$. The number of vertices is $24k$ while $m=64k+k-1$. This resulting graph is $K_{3,3}$-free bipartite and has all vertices of degree at least 4. The bound $65k-1\le 48k-4$ is not satisfied for any k.

Lemma 5. *Let $G=(X,Y,E)$ be a biconvex graph which has a DIM. Then G is $K_{3,3}$-free.*

Proof. Suppose G contains a $K_{3,3}$ given by $X'=\{x_1,x_2,x_3\}\subseteq X$ and $Y'=\{y_1,y_2,y_3\}\subseteq Y$. Consider an arbitrary DIM of the graph and its corresponding black-white coloring of the vertices. Then the vertices of X' and Y' must have distinct colors. Suppose w.l.o.g. that the vertices X' are black and those of Y' are white. Let y^*_1, y^*_2, y^*_3 be the black neighbors of x_1, x_2, x_3, respectively. It follows that the graph induced by the nine vertices of $X'\cup Y'\cup\{y^*_1,y^*_2,y^*_3\}$ is not biconvex, a contradiction. □

Corollary 6. *The DIM problem for biconvex graphs can be solved in $O(n)$ time.*

Proof. Let G be a biconvex graph. If G contains a DIM, by Lemma 5, G is $K_{3,3}$-free. Therefore G has at most $2n-4$ edges, by Lemma 4. Consequently, given an arbitrary biconvex graph, count the number of its edges, up to $2n-4$. If the number of edges exceeds $2n-4$ then the graph does not contains any DIM, otherwise apply the algorithm [1], which solves the DIM problem in $O(n+m)$ time, for chordal bipartite graphs. Since convex graphs are contained in chordal bipartite, we can solve the DIM problem for biconvex graphs in $O(n)$ time. □

We remark that there are convex graphs having a quadratic number of edges that admit DIM's. For instance, $V(G)=V_1\cup V_2\cup V_3$, where $|V(G)|=n, |V_1|=|V_2|=|V_3|=\frac{n}{3}$. Let V_i be an independent set for $1\le i\le 3$, and let $V_1\cup V_2$ induce a complete bipartite graph, $V_1\cup V_3$ be an induced matching, and $V_2\cup V_3$ be an independent set. Such a graph is bipartite, with bipartition $(V_1, V_2\cup V_3)$, moreover it is convex bipartite since it admits an interval ordering. Also, it contains a quadratic number of edges. On the other hand, $V_1\cup V_3$ is a DIM of it.

3. Claw-free graphs

The problem of finding a DIM of a claw-free graph, if existing, has been solved in [7] by an $O(n^2)$ time algorithm. We review the ideas of this paper and propose an improvement of it.

We assume that the given graph $G=(V(G),E(G))$ is connected, and has neither an induced cycle nor an induced path. Clearly, if G is disconnected we can reduce the problem to its connected components, while if G is a cycle or a path the solution is trivial.

By [7], if a claw-free graph G has a DIM then each vertex v of G is one of the following six types: **(1)** degree 1; **(2)** degree 2 with two non-adjacent neighbors; **(3)** degree 2 with two adjacent neighbors; **(4)** degree 3 with $G[N(v)]$ inducing a K_1+K_2, and the two edges connecting v to the K_2 are called *heavy*, while the third incident edge to v is called *light*; **(5)** degree 3 with a $G[N(v)]$ inducing a P_3; **(6)** degree 4 with $G[N(v)]$ inducing a $2K_2$. Thus, we assume that each vertex of G falls into one of the above types. This implies $m\le 2n$, i.e. $m=O(n)$.

The algorithm [7] can be viewed as a sequence of the following distinct phases:

1. Handling three consecutive vertices of Type 2.
2. Handling vertices of Type 1 which are at distance at least 3 of some Type 4 vertex.
3. Coloring all vertices of Types 1, 2, 5 and 6.
4. Coloring the remaining vertices, of Types 3 and 4.

Our proposed algorithm describes new formulations for Phases 1, 2 and 4, while maintaining the original Phase 3 of the algorithm [7]. We proceed by describing each of the parts.

3.1. Phase 1

The purpose is to eliminate the occurrence of three consecutive Type 2 vertices v_1, v_2, v_3, such that $N(v_2)=\{v_1,v_3\}$, $N(v_1)=\{v_2,w_1\}$ and $N(v_3)=\{v_2,w_3\}$. Consider the following alternatives:

- $w_1=w_3$: In this case if $d(w_1)=2$ then $G=C_4$, which contradicts G not to be a cycle. Hence $d(w_1)\ge 3$, but then $G[N[w_1]]$ contains a claw, a contradiction. Thus this case does not occur.
- $w_1w_3\in E(G)$: If $d(w_1)=d(w_3)=2$ then $G=C_5$ again a contradiction. Hence we may suppose $\exists u\in N(w_1)\setminus\{v_1,w_3\}$. We know that $u\notin N(v_1)$, thus in order to avoid a claw in $G[N[w_1]]$ we must assume $u\in N(w_3)$. The latter implies that no more vertices can belong to the neighborhoods of w_1 and w_3, otherwise G would contain vertices outside the above six types, a contradiction.

  Any DIM of G must have exactly one edge of the triangle $\{w_1,u,w_3\}$. The edge w_1w_3 does not lead to a valid DIM since it forces v_2 to be a single black vertex without black neighbor. It is easy to verify that the possibilities are either: $\{w_1u, v_2v_3\}$ or $\{w_3u, v_1v_2\}$. Therefore we can eliminate vertices v_1, v_2, v_3 and sum the weight of edge v_1v_2 to that of w_3u, and sum the weight of v_2v_3 to that of w_1u. To guarantee that the edge w_1w_3 is not chosen to enter the DIM, we assign infinite weight to it.

- $w_1 \neq w_3$ and $w_1w_3 \notin E(G)$: In this case we use the original procedure of [7], which consists of replacing vertices v_1, v_2, v_3 for the edge w_1w_3. However, the algorithm [7] solves the DIM problem without weights, thus, in order to guarantee the correct solution for the new weighted graph, we need to consider the following additional possibilities:
  - w_1, w_3 are black: Then v_1, v_3 are black and v_2 is white. The weights of edges v_1w_1 and v_3w_3 must be added to the weight of w_1w_3
  - w_1 is black and w_3 is white: In this case, v_2 and v_3 are black while v_1 is white. Hence the weight of edge v_2v_3 must be added to the weight of each edge of the set of edges w_1z, where $z \neq v_1$
  - w_3 is black and w_1 is white: This case is symmetric to the previous one. The weight of edge v_1v_2 must be added to the weight of each edge of the set w_3z, where $z \neq v_3$.

These operations are repeated until no three consecutive vertices of Type 2 remains in the graph, leaving a new reduced graph $G' = (V(G'), E(G'))$. This can be achieved in $O(n)$ time. The algorithm now proceeds on G'.

3.2. Phase 2

In this phase, we eliminate the occurrence of Type 1 vertices, lying at distance at least 3 from some Type 4 vertex. Let $v \in V(G')$ such that $d(v) = 1$ and let $w \in V(G')$ be the vertex such that $d(w) \geq 3$ and the distance to v is minimum. Note that if there is no such w then G' is a path, a contradiction. Therefore there is a path $v - w$ where all vertices, except v, w are of Type 2. Since there are at most two consecutive vertices of Type 2, the distance between v and w is at most 3. It is easy to see that w is of Type 4, otherwise G' is not claw-free. Let v, u_1, u_2, w be any path of length 3 from a vertex $v \in V(G')$ to a vertex $w \in V(G')$, with $d(v) = 1$ and $d(w) = 3$. Let $\{z_1, z_2\}$ be the K_2 induced by $N(w)$, and G^* be the graph after deletion of vertices $\{v, u_1, u_2\}$. It is clear that any DIM M^* of G^* contains exactly one edge of the triangle $\{w, z_1, z_2\}$. In case M^* contains the edge z_1z_2, we add the edge u_1u_2 to M^* in order to obtain a DIM of G, hence to generate a DIM with the same weight in G^* we set $\omega(z_1z_2) = \omega(z_1z_2) + \omega(u_1u_2)$. In case that M^* contains wz_1 or wz_2 the edge vu_1 is added to M^*. In the latter situation, we set $\omega(wz_1) = \omega(wz_1) + \omega(vu_1)$ and $\omega(wz_2) = \omega(wz_2) + \omega(vu_1)$. We repeat this process for each vertex $v \in V(G')$ such that $d(v) = 1$. Finally, we assert that every vertex of Type 1 is at distance 1 or 2 from some vertex of Type 4. These computations can be completed in $O(n)$ time.

3.3. Phase 3

By applying convenient propagation rules, the algorithm [7] colors a subset of vertices of the graph, including all vertices of Types 1, 2, 5, and 6. Let Γ be the final coloring obtained in the algorithm. First, check its validity. If Γ is not valid, then G has no DIM's and the algorithm terminates. If Γ is valid and total, also terminate the algorithm, since the unique DIM of G has been found. Otherwise, proceed to Phase 4.

All the above operations can be completed in $O(n)$ time. At the end of this phase, the only possibly uncolored vertices are of Types 3 and 4. Observe that the obtained coloring is unique, so there is no choice to be made concerning weights, so far.

3.4. Phase 4

In this phase, we extend the coloring Γ, obtained by the previous phase, into a total valid coloring. It is assumed that Γ is a valid not total coloring, which cannot be extended by propagation. Let U be the set of uncolored vertices and S the set of single black vertices of the coloring Γ. Note that extending Γ is equivalent to extending the coloring Γ' of $G^*[U \cup S]$ (in Γ', only vertices of S are colored with black color). It can be verified that in any valid coloring, the following holds: $\forall s \in S$, $N[s]$ induces in $G^*[U \cup S]$ a $K_3 = \{u, v, s\}$ where $u, v \in U$. Since vertices of S and Type 3 vertices are simplicial in $G^*[U \cup S]$, any central vertex of induced P_3 in $G^*[U \cup S]$ must be an uncolored Type 4 vertex. Particularly, the vertices of a cycle $C_{k \geq 4}$ are central vertices of induced P_3's. Moreover, an edge of induced P_3 must be heavy and the other one must be light. It is easy to see that vertices of a light edge must have different colors. The following lemma is helpful to extend coloring C'.

Lemma 7. *Let Γ'' any total valid coloring extensible from Γ' and $P = (v_1, \ldots, v_t)$ be an induced path of $G^*[U \cup S]$ such that v_1, v_t are Type 4 vertices, v_1v_2 is a light edge and v_1 is a black vertex, then (i) v_iv_{i+1} is a light (heavy) edge if i is odd (even); (ii) v_i is black (white) if i is odd (even).*

Proof. Since P is an induced path, $v_2, \ldots, v_{t-1}$ are central vertices of induced P_3's, they are also Type 4 vertices and the edges of P are light and heavy alternately. Then (i) holds because the first edge is white. On the other hand, vertices of light edges must have different colors, while the same occurs for heavy edges if one vertex is white. Since v_1 is black and v_1v_2 is a light edge, then v_2 is white and v_3 is black. Again, we can check that v_3 satisfies the same properties as v_1 and v_4 will satisfy the same properties as v_2. Therefore there is a unique valid coloring for vertices of P which consists of alternating the colors of the vertices, where (ii) v_i is black if and only if i is odd. □

We proceed by finding a minimum weight DIM on each connected component G_i of $G^*[U \cup S]$:

3.4.1. G_i is a chordal graph

In this case, for each single black vertex s in G_i, its neighbors u and v form an edge and we set $\omega(uv) = \infty$. In this way any non infinity weight DIM of G_i will not contain this edge and s will be a black vertex as in C'. Apply the algorithm described in the previous section that computes $DIM_\Omega(G_i)$, if existing.

3.4.2. G_i has an induced cycle C_k, $k \geq 4$

As it was mentioned before, C_k is formed by light and heavy edges, where each light edge is adjacent to heavy edges and viceversa.

Lemma 8. *(See [7].) Let G^* be the resulting claw-free graph and Γ the partial valid coloring obtained after Phase 3. If the subgraph of G^* induced by uncolored vertices contains an induced cycle $C_{k\geq 4}$, then k is even. Moreover, if G^* admits a black-white partition, then the vertices of C_k are colored alternately black and white along the cycle, and furthermore, by switching the colors of vertices of C_k we again obtain a valid black-white partition of G^*.*

Lemma 9. *G_i admits exactly two DIM's or none.*

Proof. We extend the initial coloring choosing any alternate coloring for C_k and applying propagation rules. Let Γ_i be this result coloring. We will prove that Γ_i is invalid or is a total valid coloring. Clearly, if Γ_i is invalid then G_i has no DIM's by Lemma 8. If Γ_i is a total valid coloring, then switching the colors of vertices of C_k, we obtained another total valid coloring of G_i and they are the unique total valid colorings. Suppose that Γ_i is valid but there is some uncolored vertex u in G_i. Let $P = (v_0, \ldots, v_t = u)$ be the shortest path from a vertex v_0 in C_k, w.o.l.g. we can assume that $v_0, \ldots, v_{t-1}$ are colored vertices. Clearly, P is an induced path. On the other hand, $v_0 v_1$ is a heavy edge because v_1 must be adjacent to two consecutive vertices of C_k by the claw-freeness. Hence, v_1 must be a black vertex and $t \geq 2$. Then v_1, v_{t-1} are central vertices of induced P_3's which implies that v_1, v_{t-1} are Type 4 vertices and $v_1 v_2$ is a light edge. Clearly, $v_t = u$ must be Type 3 vertex because otherwise it must be colored applying Lemma 7. Hence, $v_{t-1} v_t$ is a heavy edge and $v_{t-2} v_{t-1}$ is a light edge which means that t is odd and v_{t-1} is white vertex. Therefore, v_t must be a black vertex which is a contradiction. Consequently, C_i is a total valid coloring. □

Using these two lemmas, we can determine in linear time all DIM's of G_i and return one of minimum weight (if existing).

As for the complexity of the last phase of the algorithm, observe that a cycle of length ≥ 4 of a non chordal graph can be obtained in linear time in the order of G, that is, $O(n)$ time. All the remaining steps can be completed in $O(n)$ time. It should be noted that the corresponding phase of the algorithm [7] requires $O(n^2)$ time. The main difference is that in the presently proposed algorithm, it is sufficient to find just one induced cycle of length ≥ 4, and propagate the coloring to its connected component, whereas the algorithm [7] requires the computation of $O(n)$ such cycles, in subgraphs not necessarily disjoint. Since each of them needs $O(n)$ time, the overall complexity of the latter algorithm is $O(n^2)$.

Our proposed formulation computes $DIM_\Omega(G)$ in $O(n)$ time. Observe that through the algorithm the input graph is modified, however the changes do not alter the value of the $DIM_\Omega(G)$ solution. As for the actual minimizing DIM, itself, there is no difficulty retrieving it in $O(n)$ time, by backwards computation.

Applying similar techniques of the above algorithm, the number of DIM's of the graph G can be obtained in $O(n)$ time.

Acknowledgements

We thank the unknown reviewer for his helpful comments.

The first and second authors were partially supported by UBACyT Grants 20020100100754 and 20020120100058, PICT ANPCyT Grant 1970 and PIP CONICET Grant 11220100100310. The third author was partially supported by CNPq, CAPES and FAPERJ, research agencies.

References

[1] A. Brandstädt, C. Hundt, R. Nevries, Efficient edge domination on hole-free graphs in polynomial time, in: LATIN'10 Latin American Conference on Theoretical Informatics, in: Lect. Notes Comput. Sci., vol. 6034, 2011, pp. 650–661.

[2] A. Brandstädt, A. Leitert, D. Rautenbach, Efficient dominating and edge dominating sets for graphs and hypergraphs, in: ISAAC'12 International Symposium on Algorithms and Computation, in: Lect. Notes Comput. Sci., vol. 7676, 2012, pp. 267–277.

[3] A. Brandstädt, R. Mosca, Dominating induced matchings for P7-free graphs in linear time, in: ISAAC'11 International Symposium on Algorithms and Computation, in: Lect. Notes Comput. Sci., vol. 7074, 2011, pp. 100–109.

[4] M.C. Lin, M.J. Mizrahi, J.L. Szwarcfiter, Exact algorithms for dominating induced matchings, CoRR, arXiv:1301.7602 [abs], 2013.

[5] M.C. Lin, M.J. Mizrahi, J.L. Szwarcfiter, An $O^*(1.1939^n)$ time algorithm for minimum weighted dominating induced matching, in: ISAAC'13 International Symposium on Algorithms and Computation, in: Lect. Notes Comput. Sci., vol. 8283, 2013, pp. 558–567.

[6] M.C. Lin, M.J. Mizrahi, J.L. Szwarcfiter, $O(n)$ time algorithms for dominating induced matching problems, in: LATIN'14 Latin-American Theoretical Informatics, in: Lect. Notes Comput. Sci., vol. 8392, 2014, pp. 399–408.

[7] D.M. Cardoso, N. Korpelainen, V.V. Lozin, On the complexity of the dominating induced matching problem in hereditary classes of graphs, Discrete Appl. Math. 159 (2011) 521–531.

[8] Z.-Z. Chen, S. Zhang, Tight upper bound on the number of edges in a bipartite $K_{3,3}$-free or K_5-free graph with an application, Inf. Process. Lett. 84 (2002) 141–145.

[9] D.L. Grinstead, P.J. Slater, N.A. Sherwani, N.D. Holmes, Efficient edge domination problems in graphs, Inf. Process. Lett. 48 (1993) 221–228.

[10] C.L. Lu, C.Y. Tang, Solving the weighted efficient edge domination problem on bipartite permutation graphs, Discrete Appl. Math. 87 (1998) 203–211.

[11] C.L. Lu, M.-T. Ko, C.Y. Tang, Perfect edge domination and efficient edge domination in graphs, Discrete Appl. Math. 119 (2002) 227–250.