Algorithmica (2014) 68:998–1018
DOI 10.1007/s00453-012-9709-4

Dominating Induced Matchings for P_7-Free Graphs in Linear Time

Andreas Brandstädt · Raffaele Mosca

Received: 8 September 2011 / Accepted: 29 October 2012 / Published online: 10 November 2012

Abstract Let G be a finite undirected graph with edge set E. An edge set $E' \subseteq E$ is an *induced matching* in G if the pairwise distance of the edges of E' in G is at least two; E' is *dominating* in G if every edge $e \in E \setminus E'$ intersects some edge in E'. The *Dominating Induced Matching Problem* (*DIM*, for short) asks for the existence of an induced matching E' which is also dominating in G; this problem is also known as the *Efficient Edge Domination* Problem.

The DIM problem is related to parallel resource allocation problems, encoding theory and network routing. It is $\mathbb{NP}$-complete even for very restricted graph classes such as planar bipartite graphs with maximum degree three. However, its complexity was open for P_k-free graphs for any $k \geq 5$; P_k denotes a chordless path with k vertices and $k-1$ edges. We show in this paper that the weighted DIM problem is solvable in linear time for P_7-free graphs in a robust way.

Keywords Dominating induced matching · Efficient edge domination · P_7-free graphs · Linear time algorithm · Robust algorithm

1 Introduction

Let G be a simple undirected graph with vertex set V and edge set E. A subset M of E is an *induced matching* in G if the G-distance of every pair of edges $e, e' \in M$, $e \neq e'$, is at least two, i.e., $e \cap e' = \emptyset$ and there is no edge $xy \in E$ with $x \in e$ and $y \in e'$. A subset $M \subseteq E$ is a *dominating edge set* if every edge $e \in E \setminus M$ shares an

A. Brandstädt (✉)
Fachbereich Informatik, Universität Rostock, A.-Einstein-Str. 21, 18051 Rostock, Germany
e-mail: ab@informatik.uni-rostock.de

R. Mosca
Dipartimento di Scienze, Universitá degli Studi "G. D'Annunzio", Pescara 65121, Italy
e-mail: r.mosca@unich.it

endpoint with some edge $e' \in M$, i.e., if $e \cap e' \neq \emptyset$. A *dominating induced matching* (*d.i.m.* for short) is an induced matching which is also a dominating edge set.

Let us say that *an edge $e \in E$ is matched by M* if $e \in M$ or there is an $e' \in M$ with $e \cap e' \neq \emptyset$. Thus, M is a d.i.m. of G if and only if every edge of G is matched by M but no edge is matched twice.

The *Dominating Induced Matching Problem* (*DIM*, for short) asks whether a given graph has a dominating induced matching. This can also be seen as a special 3-colorability problem, namely the partition into three independent vertex sets A, B, and C such that $G[B \cup C]$ is an induced matching: If $M \subseteq E$ is a d.i.m. of G then the vertex set has the partition $V = A \cup V(M)$ with independent vertex set A, and independent sets B, C with $B \cup C = V(M)$.

Dominating induced matchings are also called *edge packings* in some papers, and DIM is known as the *Efficient Edge Domination Problem* (*EED* for short). A brief history of EED as well as some applications in the fields of resource allocation, encoding theory and network routing are presented in [14] and [16].

Grinstead et al. [14] showed that EED is $\mathbb{NP}$-complete in general. EED remains hard for bipartite graphs [18]. In particular, [17] shows the intractability of EED for planar bipartite graphs and [10] for very restricted bipartite graphs with maximum degree 3 (the restrictions are some forbidden subgraphs). In [4], it is shown that the problem remains $\mathbb{NP}$-complete for planar bipartite graphs with maximum degree 3 but is solvable in polynomial time for hole-free graphs (in [9, 17], the complexity of EED was mentioned as an open problem for weakly chordal graphs which are a subclass of hole-free graphs). Some other new results for EED are given in [7]. In [9], as another open problem, it is mentioned that for any $k \geq 5$, the complexity of DIM is unknown for the class of P_k-free graphs. Note that the complexity of the related problems Maximum Independent Set and Maximum Induced Matching is unknown for P_5-free graphs, and a lot of work has been done on subclasses of P_5-free graphs.

In this paper, we show that for P_7-free graphs, DIM is solvable in linear time. Actually, we consider the edge-weighted optimization version of DIM, namely the *Minimum Dominating Induced Matching Problem* (*MDIM*), which asks for a dominating induced matching M in $G = (V, E)$ of minimum weight with respect to some given weight function $\omega : E \to \mathbb{R}$ (if existent). For P_5-free graphs, DIM is solvable in time $\mathcal{O}(n^2)$ as a consequence of the fact that the clique-width of (P_5,gem)-free graphs is bounded [5, 6] and a clique-width expression can be constructed in time $\mathcal{O}(n^2)$ [3]. In [9], it is mentioned that DIM is expressible in a certain kind of Monadic Second Order Logic, and in [12], it was shown that such problems can be solved in linear time on any class of bounded clique-width assuming that the clique-width expressions are given or can be determined in the same time bound. It is well known that the clique-width of cographs (i.e., P_4-free graphs) is at most 2 (and such clique-width expressions can be determined in linear time) and thus the DIM problem can be solved in linear time on cographs. In Sect. 4 we give a simple characterization of cographs having a d.i.m.

Our algorithm for P_7-free graphs is based on a structural analysis of such graphs having a d.i.m. It is robust in the sense of [20] since it is not required that the input graph is P_7-free; our algorithm either determines an optimal d.i.m. correctly or finds out that G has no d.i.m. or is not P_7-free.

The paper is structured as follows: In Sect. 2, we give further basic notions. In Sect. 3, we develop some tools for the main algorithm, in Sect. 4 we solve the DIM problem for cographs in a simple way in linear time, in Sect. 5 we describe structural properties of a P_7-free graph with d.i.m. and in particular discuss its distance levels with respect to an edge in a d.i.m. In Sect. 6, in procedure Check(xy), for a candidate edge xy (for which it is still unknown whether it is in a d.i.m.), it is analyzed whether its distance levels fulfill the properties of the distance levels described in Sect. 5, and one either obtains a d.i.m. or the answer that the input graph has no d.i.m. or is not P_7-free. In Sect. 7, as another preparing step, we solve the DIM problem for P_7-free bipartite graphs in linear time, and finally, in Sect. 8 we solve the DIM problem for P_7-free graphs robustly in linear time. If one only aims for a polynomial time algorithm then one can carry out Check(xy) for every edge xy in G; most of the tools are only necessary for obtaining a linear time algorithm. In particular, Check(xy) is done only for a fixed number of candidate edges.

2 Further Basic Notions

Let G be a finite undirected graph without loops and multiple edges. Let V denote its vertex set and E its edge set; let $|V| = n$ and $|E| = m$. For $v \in V$, let $N(v) := \{u \in V \mid uv \in E\}$ denote the *open neighborhood of* v, and let $N[v] := N(v) \cup \{v\}$ denote the *closed neighborhood of* v. If $xy \in E$, we also say that x and y *see each other*, and if $xy \notin E$, we say that x and y *miss each other*. A vertex set S is *independent* (or *stable*) in G if for every pair of vertices $x, y \in S$, $xy \notin E$. A vertex set is a *clique* in G if for every pair of vertices $x, y \in S$, $x \neq y$, $xy \in E$ holds. For $uv \in E$ let $N(uv) := N(u) \cup N(v) \setminus \{u, v\}$ and $N[uv] := N[u] \cup N[v]$. Distinct vertices x and y are *true twins* if $N[x] = N[y]$.

For $U \subseteq V$, let $G[U]$ denote the *induced subgraph* of G with vertex set U, hence, the graph which contains exactly the edges $xy \in E$ with both vertices x and y in U. Throughout this paper, subgraphs are meant to be induced subgraphs.

Let $\overline{G}$ (or co-G) denote the *complement graph* of $G = (V, E)$, i.e., $\overline{G} = (V, \overline{E})$ with $xy \in \overline{E}$ if and only if $x \neq y$ and $xy \notin E$.

Let A and B be disjoint vertex sets in G. If every vertex from A sees (misses, respectively) every vertex from B, we denote this by $A \textcircled{1} B$ (by $A \textcircled{0} B$, respectively).

A set H of at least two vertices of a graph G is called *homogeneous* if $H \neq V(G)$ and every vertex outside H is adjacent to all vertices in H or to no vertex in H. Obviously, H is homogeneous in G if and only if H is homogeneous in the complement graph $\overline{G}$.

A homogeneous set H is *maximal* if no other homogeneous set properly contains H. It is well known that in a connected graph G with connected complement $\overline{G}$, the maximal homogeneous sets are pairwise disjoint and can be determined in linear time (see, e.g., [19]).

A *chordless path* P_k (*chordless cycle* C_k, respectively) has k vertices, say $v_1, \dots, v_k$, and edges $v_i v_{i+1}$, $1 \leq i \leq k-1$ (and $v_k v_1$, respectively). We say that P_k has length $k-1$ and C_k has length k. Let K_i denote the clique with i vertices. Let $K_4 - e$ or *diamond* be the graph with four vertices and five edges, say vertices a, b, c, d and edges ab, ac, bc, bd, cd; its *mid-edge* is the edge bc. Let W_4 denote the

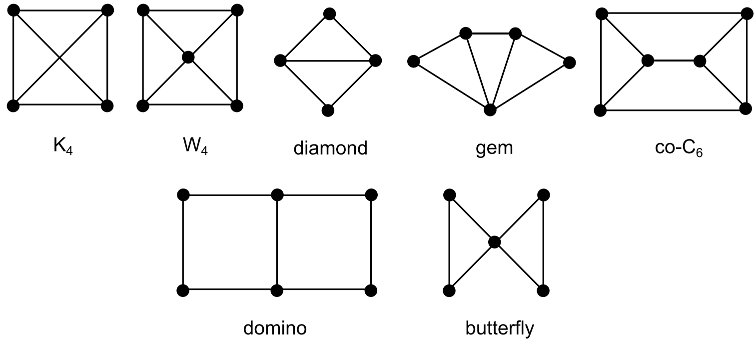


Fig. 1 K_4, W_4, *diamond*, *gem*, *co-*C_6, *domino* and *butterfly*

graph with five vertices consisting of a C_4 and a universal vertex (see Fig. 1). Let $K_{1,k}$ denote the star with one universal vertex and k independent vertices. A star is *nontrivial* if it contains a P_3 or an edge, otherwise it is *trivial*.

For two vertices $x, y \in V$, let $\text{dist}_G(x, y)$ denote the *distance between* x *and* y *in* G, i.e., the length of a shortest path between x and y in G. The *distance of two edges* $e, e' \in E$ is the length of a shortest path between e and e', i.e., $\text{dist}_G(e, e') = \min\{\text{dist}_G(u, v) \mid u \in e, v \in e'\}$. In particular, this means that $\text{dist}_G(e, e') = 0$ if and only if $e \cap e' \neq \emptyset$. For a vertex x, let $N_i(x)$ denote the *distance levels of* x: $N_i(x) := \{v \mid \text{dist}_G(v, x) = i\}$. Thus, $N_1(x) = N(x)$. For an edge xy, let $N_i(xy)$ denote the *distance levels of* xy: $N_i(xy) := \{z \mid \text{dist}_G(z, xy) = i\}$. Thus, $N_1(xy) = N(xy)$.

A *connected component* of G is a maximal vertex subset $U \subseteq V$ such that all pairs of vertices of U are connected by paths in $G[U]$. A 2*-connected component* of G is a maximal vertex subset $U \subseteq V$ such that all pairs of vertices of U are connected by at least two vertex-disjoint paths in $G[U]$. The 2-connected components are also called *blocks*. A vertex v is a *cut-vertex* of a connected graph G if $G - v$ is disconnected. A block of G is a *leaf block* if it contains only one cut-vertex of G, otherwise it is an *internal block*. It is well known that the blocks of a graph can be determined in linear time [15] (see also [1]).

For a set $\mathcal{F}$ of graphs, a graph G is called $\mathcal{F}$*-free* if G contains no induced subgraph from $\mathcal{F}$. A *hole* is a C_k for some $k \geq 5$. A graph is *hole-free* if it is C_k-free for all $k \geq 5$. A graph is *chordal* if it is C_k-free for all $k \geq 4$. A graph is *weakly chordal* if it is C_k-free and $\overline{C_k}$-free for all $k \geq 5$.

If M is a d.i.m., an edge is *matched by* M if it is either in M or shares a vertex with some edge in M. Likewise, a vertex is *matched* if it is in $V(M)$.

Note that M is a d.i.m. in G if and only if it is a dominating vertex set in the line graph $L(G)$ and an independent vertex set in the square $L(G)^2$. Thus, the DIM problem is simultaneously a packing and a covering problem.

3 Some Basic Tools

3.1 Reducing the Graph by Mandatory Edges

If an edge $e \in E$ is contained in any d.i.m. of G, we call it *mandatory* (or *forced*) in G. If an edge xy is mandatory, we can reduce the graph as follows: Delete x and y and all edges incident to x and y, and give all edges in distance one to xy the weight ∞. This means that these edges are not in any d.i.m. of finite weight in G. Let us call the resulting graph $\text{Reduced}(G, \{xy\})$.

For a set M of mandatory edges, let $\text{Reduced}(G, M)$ denote the reduced graph by applying the reduction step $\text{Reduced}(G, \{xy\})$ to each edge $xy \in M$ (in any order) as defined above. Obviously, this graph is an induced subgraph of G and can be determined in linear time for given G and M. Moreover:

Observation 1 *Let M' be an induced matching which is a set of mandatory edges in G. Then G has a d.i.m. M if and only if $\text{Reduced}(G, M')$ has a d.i.m. $M \setminus M'$.*

We can also color red all vertices in distance 1 to a mandatory edge; subsequently, an edge ab with a red vertex a cannot be matched in vertex a; it has to be matched in vertex b. If also b is red then G has no d.i.m.

$\text{Reduced}(G, M)$ is used in Algorithm P_7-Free-DIM of Sect. 8.

3.2 Reducing Singular Triangle Leaf Blocks

In Algorithm P_7-Free-DIM of Sect. 8, we also need another kind of reduction which is described subsequently. Let c be a cut-vertex of a leaf block consisting of the triangle abc. We call this a *triangle leaf block*. If c is a cut-vertex of only one such leaf block and no other leaf block has c as its cut-vertex, we call $G[\{a, b, c\}]$ a *singular triangle leaf block*. For graph G, let G^* denote the graph obtained from G by omitting all such singular triangle leaf blocks. Obviously, G^* can be constructed in linear time.

There, we also need the following transformation: For every singular triangle leaf block abc with cut-vertex c and corresponding edge weights $w(ab)$, $w(ac)$, $w(bc)$, let $\text{Tr}(G, abc)$ be the graph with the same cut-vertex c where the triangle is replaced by a path $a'b'c$ with new vertices a', b', and weights $w(ab)$ for edge $a'b'$ and $\min(w(ac), w(bc))$ for edge $b'c$. Let $\text{Tr}(G)$ be the result of applying $\text{Tr}(G, abc)$ to all singular triangle leaf blocks abc of G. Obviously, G has a d.i.m. if and only if $\text{Tr}(G, abc)$ has a d.i.m., and the optimal weights of d.i.m.'s in G and $\text{Tr}(G, abc)$ are the same. The only problem is the fact that the new graph is not necessarily P_7-free when G is P_7-free. We will apply this construction only in one case, namely when the internal blocks of G form a distance-hereditary bipartite graph; then $\text{Tr}(G)$ is also distance hereditary bipartite.

3.3 Finding Some Mandatory Edges

The following observations are helpful, in particular for obtaining mandatory edges (some of them are mentioned e.g. in [4]):

Observation 2 *Let M be a d.i.m. in G.*

(i) *M contains at least one edge of every odd cycle C_{2k+1} in G, $k \geq 1$, and exactly one edge of every odd cycle C_3, C_5, C_7 of G.*
(ii) *No edge of any C_4 can be in M.*
(iii) *If C is a C_6 then either exactly two or none of the C-edges are in M.*

Proof (i): Let C be an odd cycle C_{2k+1} in G, $k \geq 1$, with vertices $v_1, \dots, v_{2k+1}$ and edges $v_i v_{i+1}$, $i \in \{1, \dots, 2k+1\}$ (index arithmetic modulo $2k+1$). Suppose first that none of the edges of C are in M. Then the edge $v_1 v_2$ must be matched by an M-edge, say by $v_1 x$, $x \neq v_2, v_{2k+1}$. Now the edge $v_2 v_3$ must be matched in v_3 and so on, until finally the edge $v_{2k} v_{2k+1}$ must be matched in v_{2k+1} but now two M-edges are in distance one-contradiction.

Now for C_3's and C_5's in G, obviously not more than one edge can be in M. If for a C_7, two edges would be in M, say $v_1 v_2 \in M$ and $v_4 v_5 \in M$ then $v_6 v_7$ cannot be matched-contradiction.

(ii): If (v_1, v_2, v_3, v_4) is a C_4 in G then if $v_1 v_2 \in M$, $v_3 v_4$ is not matchable.

(iii): This condition obviously holds. □

Let us denote by *butterfly* (see Fig. 1) a graph of five vertices, say a, b, c, d, e, such that a, b, c and c, d, e induce a triangle. Let ab and de be the *peripheral edges* of the butterfly.

Observation 3 *The mid-edge of any diamond in G is mandatory. Moreover, the peripheral edges of any butterfly are mandatory.*

Subsequently, as a kind of preprocessing, some of the mid-edges of diamonds will be determined. Since for a linear-time algorithm it would be too time-consuming to determine all diamonds in G, we will mainly find such diamonds whose mid-edges are edges between true twins having at least two common neighbors. These are contained in maximal homogeneous sets which can be found in linear time.

3.4 Neighborhood Properties

Since the edges of any d.i.m. must have pairwise distance at least 2, we obtain by Observation 2:

Observation 4 *If G has a d.i.m. then for all vertices v, $G[N(v)]$ is cycle-free and P_4-free.*

Thus, the neighborhood of every vertex is the disjoint union of stars. Let S induce a star in $N(v)$ containing a P_3 with vertices a, b, c and edges ab, bc. Then $G[S \cup \{v\}]$ is called a *diamond-star* with *mid-edge* vb.

Observation 5 *If G has a d.i.m. then for all vertices v, one of the following three cases holds*:

(i) *$G[N(v)]$ is the disjoint union of exactly one star with P_3, and of isolated vertices. In this case, the mid-edge of the corresponding diamond-star is in M.*
(ii) *$G[N(v)]$ is the disjoint union of at least two edges and of isolated vertices. In this case, all the edges are in M.*
(iii) *$G[N(v)]$ is the disjoint union of at most one edge and of isolated vertices.*

Proof Let G have a d.i.m. M. Then by Observation 2(i), M contains an edge of every triangle, and by Observation 3, any P_3 abc in $N(v)$ generates a mandatory edge bv, and $N(v)$ can not contain two stars with P_3 since the mid-edge of any diamond-star is mandatory. Moreover, if in Case (ii), there are at least two edges in $G[N(v)]$ then all the edges are in M. □

From the previous observations, it follows (see Fig. 1 for K_4, W_4, gem, and $\overline{C_6}$):

Corollary 1 *If G has a d.i.m. then G is K_4-free, W_4-free, gem-free and $\overline{C_k}$-free for any $k \geq 6$.*

3.5 Homogeneous Sets

Now we deal with homogeneous sets in G.

Proposition 1 *Let G have a d.i.m. and let H be a homogeneous set in G.*

- (i) *If H contains an edge then $N(H)$ is stable.*
- (ii) *If $|N(H)| \geq 2$ then H is either a stable set or a disjoint union of edges.*
- (iii) *Vertices x and y are true twins with at least two common neighbors in G if and only if they appear as an edge in a homogeneous set H with $|N(H)| \geq 2$.*

Proof Let G have a d.i.m. and let H be a homogeneous set in G.

(i): If H contains an edge then since by Corollary 1, G is K_4-free, $N(H)$ is stable.

(ii): If $|N(H)| \geq 2$ then by Observation 5 and Corollary 1, H must be P_3-free, i.e., is a disjoint union of cliques. Since G is K_4-free, these cliques are edges or vertices. If there is an edge uv in H and there is a component in H consisting of a single vertex w then by Observation 3, uv is a mandatory edge and for any $a \in N(H)$, the edge aw cannot be matched-contradiction.

(iii): If x and y are true twins then x, y are contained in a (maximal) homogeneous set. On the other hand, if x and y with $xy \in E$ appear in a P_3-free homogeneous set H (by the proof of (ii), H is P_3-free) then x and y are true twins. □

The following procedure uses Observation 5 and the fact that for a homogeneous set H with $|N(H)| = 1$, say $N(H) = \{z\}$, all connected components of H together with z are leaf blocks in G.

Procedure Hom-1-DIM(H)
Given: A non-stable homogeneous set H in G with $N(H) = \{z\}$.
Task: Determine some mandatory edges or find out that G has no d.i.m.

- (a) If H contains a cycle or P_4 then STOP-G has no d.i.m.
- (b) (*Now H is a P_4-free forest.*) If H contains at least two stars with P_3 then STOP-G has no d.i.m.
- (c) (*Now H is a P_4-free forest which contains at most one star with P_3.*) If H contains exactly one star with P_3, say P_3 abc then $M := M \cup \{bz\}$. If another connected component of H contains an edge then STOP-G has no d.i.m.

(d) (*Now H is a P_3-free forest, i.e., a disjoint union of edges $E'(H)$ and isolated vertices $V'(H)$.*) If $E'(H)$ contains at least two edges then $M := M \cup E'(H)$. If $V'(H) \neq \emptyset$ then STOP-G has no d.i.m.
(e) (*Now H is the disjoint union of exactly one edge and isolated vertices $V'(H)$.*) If there is an edge ab in H and $V'(H) \neq \emptyset$ then $M := M \cup \{az\}$ or $M := M \cup \{bz\}$ (depending on the better weight).

We postpone the discussion of the final case in (e), namely $E'(H) = \{ab\}$ and $V'(H) = \emptyset$ (i.e., the case of a singular triangle leaf block). Since cographs can be recognized in linear time [8, 11], the following holds:

Lemma 1 *Procedure* Hom-1-DIM(H) *is correct and can be carried out in linear time.*

3.6 Checking the d.i.m. Property in Linear Time

In Sect. 8, we need the following:

Proposition 2 *For a given set E' of edges, it can be tested in linear time whether E' is a d.i.m., and likewise, whether E' is an induced matching.*

Proof For $E' \subseteq E$, in an array of all vertices in V, count the number $m(x)$ of appearances of each vertex of V in the edges of E' by going through all edges in E' once.

(1) Two edges of E' intersect if and only if one of the vertices appears in more than one edge of E', i.e., if there is a vertex x with $m(x) \geq 2$.
(2) Two edges of E' have distance 1 if and only if for an edge $xy \in E \setminus E'$, both $m(x) \geq 1$ and $m(y) \geq 1$.
(3) E' is dominating if and only if for each edge $xy \in E$, $m(x) \geq 1$ or $m(y) \geq 1$.

Obviously, steps (1)–(3) can be done in time $\mathcal{O}(n+m)$. The first two steps are checking whether E' is an induced matching. □

3.7 Identifying a C_3, C_5, C_7 or P_7 in a Non-bipartite Graph

In Algorithm P_7-Free-DIM of Sect. 8, we need the following:

Procedure Find-Odd-Cycle-Or-P_7
Given: A connected non-bipartite graph G.
Task: Determine an odd cycle C_3, C_5, C_7 or a P_7 in G.

(a) Choose a vertex x and determine the distance levels $N_1, N_2, \ldots$ with respect to x. If $N_6 \neq \emptyset$ then STOP-G contains a P_7.
(b) If there is an edge $ab \in E$ in N_1 then xab is a C_3. Else N_1 is stable.
(c) If there is an edge $ab \in E$ in N_2 then either abc is a C_3 for a common neighbor $c \in N_1$ of a, b or for neighbors $a' \in N_1$ of a and $b' \in N_1$ of b, $xaba'b'$ is a C_5. Else N_2 is stable.

(d) If there is an edge $ab \in E$ in N_3 then either abc is a C_3 for a common neighbor $c \in N_2$ of a, b or for neighbors $a' \in N_2$ of a and $b' \in N_2$ of b, and a common neighbor $c \in N_1$ of a', b', $caba'b'$ is a C_5 or for neighbors $a'' \in N_1$ of a' and $b'' \in N_1$ of b', $xa''b''a'b'ab$ is a C_7. Else N_3 is stable.
(e) If there is an edge $ab \in E$ in N_4 then either abc is a C_3 for a common neighbor $c \in N_3$ of a, b or for neighbors $a' \in N_3$ of a and $b' \in N_3$ of b, and a common neighbor $c \in N_2$ of a', b', $caba'b'$ is a C_5 or for neighbors $a'' \in N_2$ of a' and $a''' \in N_1$ of a'', $xa'''a''a'abb'$ is a P_7. Else N_4 is stable.
(f) (*Now N_5 must contain an edge, otherwise G is bipartite.*) For an edge ab in N_5, let a_4 denote a neighbor of a in N_4 and let $a_{i-1} \in N_{i-1}$ denote a neighbor of $a_i \in N_i$, $i = 2, 3, 4$. Then either a_4ab is a C_3 or $xa_1a_2a_3a_4ab$ is a P_7.

Obviously, the following holds:

Lemma 2 *Procedure* Find-Odd-Cycle-Or-P_7 *is correct and runs in linear time.*

4 DIM for Cographs in Linear Time

Recall that G is a cograph if and only if G is P_4-free. It is well known that a graph is a cograph if and only if its clique-width is at most 2. Thus, for solving the DIM problem on cographs, one could use the clique-width argument (as mentioned in the Introduction). Here we give a simple direct way. By Corollary 1, the following holds:

Corollary 2 *If G has a d.i.m. and $\overline{G}$ is not connected then G is a cograph.*

For the subsequent characterization of cographs with d.i.m., we need the following notion:

G is a *super-star* if G contains a universal vertex u such that $G[V \setminus \{u\}]$ is the disjoint union of a star and a stable set. Note that every super-star has a d.i.m. M, namely if the star contains a P_3 with central vertex c then M consists of the single edge uc, and if the star consists of only one edge ab, then $\{ua\}$ and $\{ub\}$ are both d.i.m.'s, and the choice of an optimal d.i.m. depends on the edge weights. If there is no edge in $G[V \setminus \{u\}]$ then any edge uv is a d.i.m., and the choice of an optimal d.i.m. depends on the edge weights.

For cographs having a d.i.m., there is the following simple characterization:

Proposition 3 *A connected cograph G has a d.i.m. if and only if it is either a super-star or the join $G = G_1 \textcircled{1} G_2$ of a disjoint union of edges G_1 and a stable set G_2.*

Proof Let G be a connected cograph with a d.i.m. M. Then, since G is K_4-free, $G = G_1 \textcircled{1} G_2$ for some triangle-free (i.e., bipartite) subgraphs G_1 and G_2.

Case 1. G_1 (or G_2) contains only one vertex; without loss of generality say $V(G_1) = \{u\}$.

Then by Observation 5, G_2 is the disjoint union of at most one star with P_3, of edges and vertices. If exactly one of the connected components of G_2 contains a P_3

then this component is a star, say with central vertex c, and $uc \in M$. Now the other components of G_2 must be isolated vertices since in every triangle, exactly one edge is in M. This shows that in this case, G is a super-star, and an optimal d.i.m. can be chosen as described above.

If none of these connected components contain P_3 then the connected components of G_2 are edges and vertices. If at least two such edges exist then all the connected components are edges, otherwise there is no d.i.m. This corresponds to the second case in Proposition 3.

If exactly one of the connected components is an edge, say ab, and all the others are vertices then ua and ub are possible d.i.m.'s. This is again a special super-star. If there is no edge in G_2 then G is simply a star.

Case 2. G_1 and G_2 contain at least two vertices.

If none of G_1, G_2 contains an edge then if both G_1 and G_2 contain at least two vertices, every edge is in a C_4 and therefore not in M-contradiction.

If G_1 contains an edge then by Proposition 1(i), G_2 is edgeless, and by Proposition 1(ii), G_1 is a disjoint union of edges. In this case, the uniquely determined d.i.m. of G is the set of edges in G_1.

Conversely, it is easy to see that any super-star has a d.i.m., and likewise any join of a disjoint union of edges and a stable set has a d.i.m. □

Corollary 3 *Cographs with d.i.m. can be recognized in linear time.*

The following uses Proposition 3:

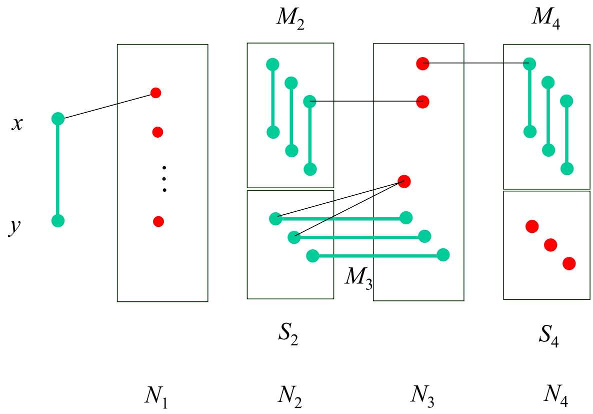
Procedure Cograph-DIM
Given: A connected cograph G with edge weights.
Task: Decide whether G has a d.i.m. and if yes, determine an optimal d.i.m. of G.

(a) Check whether G is either a super-star or the join of a disjoint union of edges and a stable set. If yes then G has a d.i.m. as described above, otherwise STOP-G has no d.i.m.

5 Structure of P_7-free Graphs with Dominating Induced Matching

Throughout this section, let $G = (V, E)$ be a connected P_7-free graph having a d.i.m. Recall that if M is a d.i.m. of G then the vertex set V has the partition $V = I \cup V(M)$ with independent vertex set I. We suppose that $xy \in M$ is an edge in an induced P_3 of G and consider the distance levels $N_i = N_i(xy)$, $i \geq 1$, with respect to the edge xy (see Fig. 2). Note that every edge of a hole C_5, C_6, or C_7 is part of an induced P_3. For triangles abc, this is not fulfilled if a and b are true twins. However, according to Proposition 1, true twins with at least two common neighbors will lead to mandatory edges as mid-edge of a diamond (or K_4 if there is an edge in their neighborhood), and true twins a, b with only one common neighbor c form a leaf block abc which will be treated by procedure Hom-1-DIM or will be temporarily omitted by constructing G^* (as described in Sect. 3.2) and looking for an odd cycle in G^*. Thus we can assume in this section that xy is an edge in M which is part of an induced P_3.

Fig. 2 Distance levels



5.1 Distance Levels with Respect to an M-Edge xy

We refer to the partition $V = V(M) \cup I$ with d.i.m. M and independent set I. Since we assume that $xy \in M$, clearly, $N_1 \subseteq I$ and thus:

$$N_1 \text{ is a stable set.} \tag{1}$$

Moreover, no edge between N_1 and N_2 is in M. Since $N_1 \subseteq I$ and all neighbors of vertices in I are in $V(M)$, we have:

$$N_2 \text{ is the disjoint union of some edges and isolated vertices.} \tag{2}$$

Let M_2 denote the set of edges in N_2 and let S_2 denote the set of isolated vertices in N_2; $N_2 = V(M_2) \cup S_2$. Obviously:

$$M_2 \subseteq M \quad \text{and} \quad S_2 \subseteq V(M). \tag{3}$$

Let M_3 denote the set of M-edges with one endpoint in S_2 (and the other endpoint in N_3).

Since xy is contained in a P_3, i.e., there is a vertex r such that y, x, r induce a P_3, we obtain some further properties:

$$N_5 = \emptyset. \tag{4}$$

Proof of (4) If there is a vertex $v_5 \in N_5$ then there is a shortest path $(v_5, v_4, v_3, v_2, v_1)$, $v_i \in N_i$, $i = 1, \dots, 5$, connecting v_5 and a neighbor v_1 of x or y. If $v_2 r \in E$ then $v_5, v_4, v_3, v_2, r, x, y$ is a P_7, and if v_2 is nonadjacent to any personal neighbor of x with respect to y then $v_5, v_4, v_3, v_2, v_1, x, r$ is a P_7 or $v_5, v_4, v_3, v_2, v_1, y, x$ is a P_7-a contradiction which shows (4). □

This kind of argument will be used later again-we will say that the subgraph induced by $x, y, N_1, v_2, v_3, v_4, v_5$ contains an induced P_7.

Obviously, by (3) and the distance condition, the following holds:

$$\text{No edge in } N_3 \text{ and no edge between } N_3 \text{ and } N_4 \text{ is in } M. \tag{5}$$

Furthermore the following statement holds.

$$N_4 \text{ is the disjoint union of edges and isolated vertices.} \tag{6}$$

Proof of (6) The proof is very similar to the one of (4): Let uv be an edge in N_4 and let $w \in N_3$ see u; then w must see also v since G is P_7-free (recall the existence of r in a P_3 with x and y). Then N_4 must be P_3-free—otherwise any neighbor $w \in N_3$ of a P_3 abc in N_4 would induce a diamond w, a, b, c and then edge wb is mandatory in contradiction to Observation 3 and condition (5). Moreover, N_4 is triangle-free (otherwise there is a K_4 in contradiction to Corollary 1). Then N_4 is a disjoint union of edges and vertices which shows (6). □

Let M_4 denote the set of edges in N_4 and let S_4 denote the set of isolated vertices in N_4; $N_4 = V(M_4) \cup S_4$. Note that by (4) and (5), $S_4 \subseteq I$.

Since every edge ab in N_4 together with a predecessor c in N_3 forms a triangle, and $ac, bc \notin M$, by (5) necessarily:

$$M_4 \subseteq M. \tag{7}$$

By Observation 2(i), in every odd cycle C_3, C_5 and C_7 of G, exactly one edge must be in M. Thus, (5) implies:

$$N_3 \cup S_4 \text{ is bipartite.} \tag{8}$$

Note that in general, N_3 is not a stable set.

5.2 Matching the S_2-Vertices by N_3-Neighbors

By the previous conditions and in particular, by (5), we obtain:

$$M = \{xy\} \cup M_2 \cup M_3 \cup M_4. \tag{9}$$

From the algorithmic point of view, determining M_2 and M_4 for a candidate edge xy is easy since these are the edges in N_2 (N_4, respectively) if (2) ((6), respectively) is fulfilled for xy. The crucial point, however, is the problem how to match the vertices in S_2 by edges with neighbors in N_3, and the remaining part of this section is dealing with the conditions under which this is possible.

Let $T_{\text{one}} := \{t \in N_3 : |N(t) \cap S_2| = 1\}$, and $T_{\text{two}} := \{t \in N_3 : |N(t) \cap S_2| \geq 2\}$. Note that if uv is an edge with $u \in T_{\text{two}}$ then $uv \notin M$ and uv must be matched by an M-edge at v since it cannot be matched at u because of the distance condition; in particular, $T_{\text{two}} \subseteq I$.

In general, (5) will lead to some forcing conditions since the edges in N_3 and between N_3 and N_4 have to be matched. If an edge $uv \in E$ cannot be matched at u then it has to be matched at v—in this case, as described later, we color the vertex v

green if it has to be matched by an M_3 edge. (For an algorithm checking the existence of a d.i.m., it is useful to observe that if vertices in distance one get color green then no d.i.m. exists.)

Let $S_3 := (N(M_2) \cap N_3) \cup (N(M_4) \cap N_3) \cup T_{\text{two}}$. Then by definition, $S_3 \subseteq N_3$, and obviously $S_3 \subseteq I$ holds. Furthermore, since $S_4 \subseteq I$, one obtains:

$$S_3 \cup S_4 \text{ is a stable set.} \tag{10}$$

Let $T^*_{\text{one}} := T_{\text{one}} \setminus S_3$. Then $N_3 = S_3 \cup T^*_{\text{one}}$ is a partition of N_3. In particular, T^*_{one} contains the M-mates of the vertices of S_2. Recall that M_3 denotes the set of M-edges with one endpoint in S_2 (and the other endpoint in T^*_{one}).

Let $S_2 = \{u_1, u_2, \dots, u_k\}$, and let $T_i := T^*_{\text{one}} \cap N(u_i)$, $i = 1, \dots, k$. Then $T^*_{\text{one}} = T_1 \cup \dots \cup T_k$ is a partition of T^*_{one}. The following condition is necessary for the existence of M_3:

$$\text{For all } i = 1, \dots, k, T_i \neq \emptyset, \text{ and exactly one vertex of } T_i \text{ is in } V(M_3). \tag{11}$$

Recall that by Observation 5, $G[T_i]$ is the disjoint union of at most one star with P_3, and of edges and isolated vertices. Furthermore, by Observation 5, $G[T_i]$ cannot contain two edges, i.e., the following statement holds for all $i = 1, \dots, k$:

$$G[T_i] \text{ is the disjoint union of isolated vertices and at most one star } Y_i \text{ with an edge.} \tag{12}$$

Proof of (12) Assume that there are two edges, say ab and $a'b'$, in T_i. Then by Observation 5, ab and $a'b'$ are mandatory, but $u_i \in V(M)$-contradiction. □

Assume that T_i contains the star Y_i with an edge.

$$\text{For all } i, j = 1, \dots, k, i \neq j, Y_i \text{ sees no vertex of } T_j. \tag{13}$$

Proof of (13) Let $t'_i t''_i$ be an edge of Y_i. By contradiction assume that a vertex $t_j \in T_j$, $i \neq j$, is adjacent to Y_i, say t_j sees t''_i. Then, since by (8), $G[T^*_{\text{one}}]$ is triangle-free, t_j is nonadjacent to t'_i, and now $x, y, N_1, u_j, t_j, t''_i, t'_i$ induce a subgraph of G containing a P_7. □

5.3 Pairing the N_3-Neighborhoods of Vertices in S_2

Claim 1 *For all $i = 1, \dots, k$, there is at most one j with $j \neq i$ such that a vertex in T_i sees a vertex in T_j.*

Proof of Claim 1 By contradiction assume that there are two indices $j \neq h$ such that some vertices in T_i see vertices in T_j and T_h.

Case 1. If there is a vertex $t_i \in T_i$ which sees a vertex $t_j \in T_j$ and $t_h \in T_h$ then, since there is no triangle in N_3, t_j misses t_h, and then $x, y, N_1, u_h, t_h, t_j, t_i$ induce a subgraph of G containing a P_7 (recall the existence of a P_3 with x, y and vertex $r \in N_1$).

Case 2. Thus, assume that there are two vertices $t'_i, t''_i \in T_i$ such that t'_i sees a vertex $t_j \in T_j$ and t''_i sees a vertex $t_h \in T_h$. Clearly, by (13), $t'_i t''_i \notin E$, and by Case 1, $t'_i t_h \notin E$, $t''_i t_j \notin E$. Moreover, $t_j t_h \notin E$, otherwise we are in Case 1 again. Now $u_j, t_j, t'_i, u_i, t''_i, t_h, u_h$ induce a P_7-contradiction. □

Let us say that T_i *sees* T_j if there are vertices in T_i and T_j which see each other. Now by Claim 1, for every $i = 1, \dots, k$, T_i either sees no T_j, $j \neq i$, and in this case let us say that T_i *is isolated*, or sees exactly one T_j, $j \neq i$, in which case we say that *T_i and T_j are paired.*

Claim 2 *If T_i and T_j are paired then $G[T_i \cup T_j]$ contains at most two components among the four following ones*: *Y_i (defined above), Y_j (defined above), Y'_i which is a star with center in T_i and the other vertices in T_j, Y'_j which is a star with center in T_j and the other vertices in T_i; in particular, at most one from $\{Y_i, Y_j\}$ does exist.*

Proof of Claim 2 By (11) and since each edge of G must be matched by M, $G[T_i \cup T_j]$ contains at most two components among the above ones. By (12) and (13) it is enough to focus on the possible components of $G[T_i \cup T_j]$ with vertices in both T_i and T_j. In particular, by (12) each such component is a star with center in T_i (in T_j, respectively) and the other vertices in T_j (in T_i, respectively); if any of such stars contains a P_3 then its center c belongs to $V(M_3)$ (in fact otherwise, c would have two neighbors in T_i or in T_j, and such neighbors should belong to $V(M)$, a contradiction to (11)); then if such stars exist and contain P_3, their centers belong to T_i and T_j respectively; then one obtains the stars described in the claim. Finally, since $G[T_i \cup T_j]$ contains at most two components, by (13) and by definition of paired sets one has that at most one from $\{Y_i, Y_j\}$ does exist. □

Claims 1 and 2 are useful tools to detect M_3. Observe that:

(i) if a vertex $t_i \in T_i$ sees a vertex of $S_3 \cup S_4$, then $u_i t_i \in M_3$;
(ii) if a vertex $t_i \in T_i$ is the center of the star Y_i or Y'_i (in case of paired sets), with a P_3 then $u_i t_i \in M_3$.

Let us say that a vertex $t_i \in T_i$ is *green* if it enjoys one of the above two conditions (i), (ii). Then the following statement holds for all $i = 1, \dots, k$:

$$G[T_i] \text{ contains at most one green vertex, say } t_i^* \tag{14}$$

and

$$G\big[T_i \setminus N\big(t_i^*\big)\big] \text{ is edgeless.} \tag{15}$$

6 Procedure Check(xy)

In our algorithm P_7-Free-DIM in Sect. 8, we carry out a fixed number of times the subsequent

Procedure Check(xy)
Given: A (candidate) edge xy which is in an induced P_3 of G.
Task: Determine a minimum weight d.i.m. M of G with $xy \in M$ or unsuccessfully STOP, i.e., return a proof that G has no d.i.m. M with $xy \in M$ or G is not P_7-free.

(a) Determine the distance levels $N_1, N_2, \dots$ with respect to xy.
(b) Check if all the conditions (1), (2), (4), (6), (8), (10)–(13) of Sects. 5.1 and 5.2 are fulfilled. If one of them is not fulfilled then unsuccessfully STOP. Otherwise, set $M := \{xy\} \cup M_2 \cup M_4$. If $S_2 = \emptyset$, then STOP and return M.
(c) Check if Claim 1 of Sect. 5.3 holds. If not, then unsuccessfully STOP. Otherwise classify the T_i sets into isolated ones and paired ones.
(d) Check if Claim 2 of Sect. 5.3 holds. If not, then unsuccessfully STOP.
(e) Color green every vertex t_i of T_i such that either t_i sees a vertex of $S_3 \cup S_4$ or t_i is the center of the star Y_i or Y_i' (in case of paired sets) with Y_i or Y_i' containing P_3.
(f) Check if conditions (14)–(15) of Sect. 5.3 hold. If not, then unsuccessfully STOP.
Notation. For any subset T_i' of any T_i set introduced in Sect. 5.3, let us say that a vertex t_i' is a *best* vertex in T_i' if $w(u_i t_i') \le w(u_i t_i'')$ for any $t_i'' \in T_i'$.
(g) For all isolated T_i, proceed as follows: If T_i has a green vertex t_i^*, then set $M := M \cup \{u_i t_i^*\}$. Otherwise set $M := M \cup \{u_i t_i'\}$ where t_i' is a best vertex in Y_i (if Y_i does exist) or is a best vertex in T_i (otherwise).
(h) For all paired T_i and T_j, proceed as follows.
(h.1) If T_i and T_j have a green vertex, respectively t_i^* and t_j^*, then: if t_i^* misses t_j^*, and if $G[(T_i \cup T_j) \setminus (N(t_i^*) \setminus N(t_j^*))]$ is edgeless then set $M := M \cup \{u_i t_i^*\} \cup \{u_j t_j^*\}$; otherwise unsuccessfully STOP.
(h.2) If T_i has a green vertex t_i^*, and if T_j has no green vertex, then: If $G[(T_i \cup T_j) \setminus N(t_i^*)]$ has at least one vertex and contains most one component (i.e., Y_j' or Y_j), then set $M := M \cup \{u_i t_i^*\} \cup \{u_j t_j\}$ where t_j is, in this order, either the vertex in $Y_j' \cap T_j$ (if), or a best vertex in Y_j (if), or a best vertex in T_j. Otherwise unsuccessfully STOP. If T_j has a green vertex t_j^*, and if T_i has no green vertex, then proceed similarly by symmetry.
(h.3) If T_j and T_j has no green vertex (according to Claim 2 and to the above, $G[T_i \cup T_j]$ contains isolated vertices, at most two isolated edges, and at least one isolated edge, say $t_i t_j$, between T_i and T_j), then proceed as follows:

- If there exists another edge, say pq, in T_i or T_j then: If $p, q \in T_i$ (or $p, q \in T_j$) then set $M := M \cup \{u_i z\} \cup \{u_j t_j\}$ where z is a best vertex in $\{p, q\}$ (or $M := M \cup \{u_i t_i\} \cup \{u_j z\}$ where z is a best vertex in $\{p, q\}$); if $p \in T_i$ and $q \in T_j$, then either set $M := M \cup \{u_i p\} \cup \{u_j t_j\}$ or set $M := M \cup \{u_i t_i\} \cup \{u_j q\}$, depending on the best alternative.
- Otherwise: If $(T_i \setminus \{t_i\}) \cup (T_j \setminus \{t_j\}) = \emptyset$, then unsuccessfully STOP; if $T_i \setminus \{t_i\} \neq \emptyset$ and $T_j \setminus \{t_j\} = \emptyset$, then set $M := M \cup \{u_i z_i\} \cup \{u_j t_j\}$ where z_i is a best vertex in $T_i \setminus \{t_i\}$; if $T_i \setminus \{t_i\} = \emptyset$ and $T_j \setminus \{t_j\} \neq \emptyset$, then set $M := M \cup \{u_i t_i\} \cup \{u_j z_j\}$ where z_j is a best vertex in $T_j \setminus \{t_j\}$; if $T_i \setminus \{t_i\} \neq \emptyset$ and $T_j \setminus \{t_j\} \neq \emptyset$, then either set $M := M \cup \{u_i z_i\} \cup \{u_j t_j\}$ where z_i is a best

vertex in $T_i \setminus \{t_i\}$, or set $M := M \cup \{u_i t_i\} \cup \{u_j z_j\}$ where z_j is a best vertex in $T_j \setminus \{t_j\}$, depending on the best alternative.

(j) STOP and return M.

Theorem 1 *Procedure* Check(xy) *is correct and runs in linear time.*

Proof Correctness: The correctness of Procedure Check(xy) follows from the structural analysis of P_7-free graphs with d.i.m. described in Sects. 5.1, 5.2 and 5.3.

Time bound: (a): Determining the distance levels N_i with respect to edge xy can be done in linear time, e.g. by using BFS.

(b): Likewise, concerning conditions (1), (2), (4), (6), (8), (10)–(13), we can test in linear time if N_1 is a stable set, N_2 is the disjoint union of edges and isolated vertices, $N_5 = \emptyset$, N_4 is the disjoint union of edges and isolated vertices and $N_3 \cup S_4$ is bipartite. The assignments can be done in linear time: This is obvious for M, S_2 and S_4. Then determine the degree of all vertices in N_3 with respect to S_2, and assign degree one vertices to T_{one} and degree ≥ 2 vertices to T_{two}. Obviously, a vertex in N_3 which misses S_2 has a predecessor in M_2, and thus S_3 and $T^*_{\text{one}} = T_{\text{one}} \setminus S_3$ form a partition of N_3. Obviously, it can be checked in linear time whether $N_3 \cup S_4$ is a bipartite subgraph and whether $S_3 \cup S_4$ is a stable set.

(c)–(j): All these steps can obviously be done in linear time. □

In the other case when an edge xy is not in any P_3, it follows that x and y are true twins, and this case will be treated by determining the maximal homogeneous sets of G.

7 DIM for P_7-Free Bipartite Graphs in Linear Time

In this section, as a further preparing step for the general case, we show how to solve the DIM problem on P_7-free bipartite graphs in linear time. A *domino* (see Fig. 1) is a bipartite graph having six vertices, say $x_1, x_2, x_3, y_1, y_2, y_3$ such that x_1, y_1, x_2, y_2, x_3 induce a P_5 with edges $x_1y_1, y_1x_2, x_2y_2, y_2x_3$ and y_3 sees exactly x_1, x_2 and x_3.

Observation 6 *Let M be a d.i.m. of a bipartite P_7-free graph B.*

(i) *If C is a C_6 in B then exactly two C-edges are in M.*
(ii) *B is domino-free.*

Proof (i): Assume to the contrary that the statement is not true. Let C be a C_6 in B with vertices $v_1, \dots, v_6$ and edges v_iv_{i+1}, $i \in \{1, \dots, 6\}$ (index arithmetic modulo 6). Then by Observation 2(iii), none of the C-edges are in M. Then since every edge of B is matched by M, exactly three vertices of C, say v_1, v_3, v_5, belong to $V \setminus V(M)$, while v_2, v_4, v_6 belong to $V(M)$: let v'_2, v'_4, v'_6 be respectively their M-mates. Then by definition of M and since B is bipartite, $v'_2, v_2, v_3, v_4, v_5, v_6, v'_6$ induce a P_7-contradiction.

(ii): If D is a domino in B then by Observation 2(ii), the edges of the two C_4's of D must be matched from outside but now obviously there is a P_7-contradiction. □

If moreover, B is C_6-free, it is $(6, 2)$-chordal bipartite, i.e., distance hereditary and bipartite (see e.g. [2]). In this case, DIM can be easily solved in linear time by using the clique-width argument [12, 13] since the clique-width of distance-hereditary graphs is at most 3 (and 3-expressions can be determined in linear time). We want to give a robust linear-time algorithm for P_7-free bipartite graphs for solving the DIM problem. If a bipartite graph B is given, the algorithm either solves the DIM problem optimally or shows that there is a domino or P_7 in B. The algorithm constructs the distance levels starting from an arbitrarily chosen vertex. Then it checks whether B is distance hereditary as in [2]. If a domino or P_7 is found, the algorithm unsuccessfully stops, and if a C_6 C is found, one of the pairs of opposite edges in C must be in M, say v_1v_2 and v_4v_5, and in this case, it is checked by Check(v_1v_2) whether the distance levels starting from v_1v_2 have the required properties.

For making this paper self-contained, we repeat Corollary 5 of [2]:

Corollary 4 (Bandelt, Mulder [2]) *Let G be a connected graph, and let u be any vertex of G. Then G is bipartite and distance hereditary if and only if all distance levels $N_k(u)$ are edgeless, and for any vertices $v, w \in N_k(u)$ and neighbors x and y of v in $N_{k-1}(u)$, we have*

(∗) $N(x) \cap N_{k-2}(u) = N(y) \cap N_{k-2}(u)$, *and further,*
(∗∗) $N(v) \cap N_{k-1}(u)$ *and* $N(w) \cap N_{k-1}(u)$ *are either disjoint, or one is contained in the other.*

We have to check level by level beginning with the largest index, whether conditions (∗) and (∗∗) are fulfilled. If one of them is violated, we obtain a hole or domino.

This leads to the following procedure for the bipartite case which includes a certifying recognition algorithm:

Procedure P_7-Free-Bipartite-DIM
Given: A connected bipartite graph B with edge weights.
Task: Determine a d.i.m. M in B of minimum weight (if existent) or unsuccessfully STOP, i.e., find out that B has no d.i.m. or is not P_7-free.

(a) Choose a vertex $u \in V$ and determine the distance levels $N_1(u), N_2(u), \ldots$ with respect to u. If $N_6(u) \neq \emptyset$ then STOP-B is not P_7-free.
(b) For all levels $N_k(u)$, $k \leq 5$, beginning with $N_5(u)$, check whether conditions (∗) and (∗∗) are fulfilled. If one of them is violated, we obtain an obstruction which is either a hole C_8 or C_{10} (in the case of a C_8 or C_{10} STOP-B is not P_7-free), or a C_6 C (in which case we have to proceed with C) or a domino-STOP-B has no d.i.m. or is not P_7-free.
(c) If for all levels, conditions (∗) and (∗∗) are fulfilled, B is distance hereditary and bipartite. Apply the clique-width approach for solving the DIM problem.

(d) (*Now B is not distance hereditary and C is a C_6 in B.*) For three consecutive edges ab of C, carry out Check(ab). If none of them ends successfully then STOP-B has no d.i.m., otherwise we obtain an optimal d.i.m. (among the at most three solutions).

Procedure Check(ab) assumes that ab is in a C_6 of the bipartite graph B. In this case we have some additional properties, and the procedure could be simplified:

Let $N_{1a} = N(a) \cap N_1(ab)$ ($N_{1b} = N(b) \cap N_1(ab)$, respectively). Obviously, the following is a partition of $N_1(ab)$ if B is bipartite:

$$N_1(ab) = N_{1a} \cup N_{1b} \tag{16}$$

As before, $N_1(ab)$ has to be stable, and $N_2(ab)$ is a disjoint union of edges M_2 and isolated vertices S_2. Since ab is in a C_6, we have that $M_2 \neq \emptyset$.

Since B is P_7-free and assuming that $ab \in M$, obviously:

$$S_2 = \emptyset \quad \text{and} \quad N_4(ab) = \emptyset. \tag{17}$$

Moreover:

$$N_3(ab) \text{ is edgeless.} \tag{18}$$

Finally, since B is P_7-free, we obtain:

$$\text{Vertices in } M_2 \text{ of the same color have the same neighborhood in } N_1(ab). \tag{19}$$

Proof of (19) Let $ef \in M_2$ and $gh \in M_2$ with e and g in the same color class, and suppose that e sees $x \in N_{1a}$ while g misses x. Then there is $y \in N_{1b}$ such that $yf \in E$. Since $N_1(ab)$ is stable, $xy \notin E$. Since g misses x, there is a neighbor $z \in N_{1a}$ of g. Since h, g, z, a, x, e is no P_7, $ze \in E$. Again, since $N_1(ab)$ is stable, $yz \notin E$. If $hy \in E$ then x, e, z, g, h, y, b is a P_7. Thus, $hy \notin E$ but now h, g, z, a, b, y, f is a P_7—a contradiction which shows (19). □

Obviously, $\{ab\} \cup M_2$ is a d.i.m. of B if all conditions are fulfilled.

Lemma 3 *Procedure P_7*-Free-Bipartite-DIM *is correct and runs in linear time.*

Proof The correctness of the procedure follows from the structural analysis of bipartite P_7-free graphs with d.i.m. The time bound follows from the fact that procedure Check(xy) is carried out only for a fixed number of candidate edges, and each step of the procedure can be done in linear time. □

8 The DIM Algorithm for the General P_7-Free Case

In the previous chapters we have analyzed the structure of P_7-free graphs having a d.i.m. Now we are going to use these properties for an efficient algorithm for solving the DIM problem on these graphs.

Algorithm P_7-Free-DIM
Given: A connected graph $G=(V,E)$ with edge weights.
Task: Determine a d.i.m. in G of finite minimum weight (if existent) or find out that G has no d.i.m. or is not P_7-free.

(a) If G is bipartite then carry out procedure P_7-Free-Bipartite-DIM.
(b) (*Now G is not bipartite.*) If G is a cograph then apply procedure Cograph-DIM. If G is not a cograph but $\overline{G}$ is not connected then STOP-G has no d.i.m.
(c) (*Now G is neither bipartite nor a cograph, and $\overline{G}$ is connected.*) Let $M := \emptyset$. Determine the maximal homogeneous sets $H_1, \dots, H_k$ of G. For all $i \in \{1, \dots, k\}$ do the following steps (c.1), (c.2):
(c.1) If $|N(H_i)| = 1$ and H_i is not a stable set then carry out procedure Hom-1-DIM(H_i).
(c.2) In the case when $|N(H_i)| \geq 2$ and H_i is not a stable set then check whether $N(H_i)$ is stable and H_i is a disjoint union of edges; if not then STOP-G has no d.i.m., otherwise, for all edges xy in H_i, let $M := M \cup \{xy\}$.
(d) If $M \neq \emptyset$ then construct $G' = \mathit{Reduced}(G, M)$ as described in Sect. 3.1.
(e) For every connected component C of G', do:
  (e.1) If C is bipartite then carry out procedure P_7-Free-Bipartite-DIM for C. Otherwise:
  (e.2) Construct C^* as described in Sect. 3.2 (where the singular triangle leaf blocks are temporarily omitted) and carry out Find-Odd-Cycle-Or-P_7 for C^*.
  (e.3) If an odd cycle C_3, C_5 or C_7 is found, carry out Check(ab) in the component C for all (at most seven) edges of the odd cycle. Add the resulting edge set to the mandatory edges from steps (c.1), (c.2), respectively.
  (e.4) If however, C^* is bipartite then with procedure P_7-Free-Bipartite-DIM for C^*, find out if the procedure unsuccessfully stops or if there is a C_6 in C^*; in the last case, do Check(ab) in the component C for all edges of the C_6.
  (e.5) Finally, if C^* is distance hereditary bipartite, construct Tr(C) as described in Sect. 3.2 (the omitted triangle leaf blocks are attached as P_3's and the resulting graph is distance hereditary bipartite) and solve DIM for this graph using the clique-width argument (or using the linear time DIM algorithm for chordal bipartite graphs given in [4]).
(f) Finally check once more whether M is a d.i.m. of G. If not then G has no d.i.m., otherwise return M.

Theorem 2 *Algorithm P_7-Free-DIM is correct and runs in linear time.*

Proof Correctness: The correctness of the algorithm follows from the structural analysis of P_7-free graphs with d.i.m. In particular, if G is bipartite (a cograph, respectively) then procedure P_7-Free-Bipartite-DIM (Cograph-DIM, respectively) correctly solves the DIM problem.

If $\overline{G}$ is not connected, i.e., $G = G_1 \textcircled{1} G_2$ for some nonempty G_1, G_2 and G has a d.i.m. then by Corollary 2, G must be a cograph.

For the maximal homogeneous sets $H_1, \dots, H_k$ of G, there are two cases $|N(H_i)| = 1$ or $|N(H_i)| \geq 2$. By Proposition 1 and Lemma 1, steps (c.1) and (c.2) are correct, and G can be correctly reduced by using the obtained set M of forced edges. Since in procedure Hom-1-DIM, in the last case, the corresponding singular triangle leaf blocks are postponed, in the reduced graph, every odd cycle contains only edges in P_3's. Thus, it is correct to apply Check(ab) for the edges of some odd cycle in the (non-bipartite) reduced graph. Finally one has to add the postponed edges and solve the DIM problem on these graphs.

Time bound: Step (a) can be done in linear time since procedure P_7-Free-Bipartite-DIM takes only linear time. Step (b) can be done in linear time since it can be recognized in linear time whether G is a cograph (see [8, 11]) and procedure Cograph-DIM can be done in linear time. Step (c) can be done in linear time since modular decomposition can be done in linear time and finds the maximal homogeneous sets [19]. There is only a linear number of true twins, and the corresponding reduced graph can be determined in linear time.

In the reduced graph $G' = \text{Reduced}(G, M)$, procedure Check($xy$) is carried out only for a fixed number of edges xy, and the procedures P_7-Free-Bipartite-DIM and Find-Odd-Cycle-Or-P_7 can be done in linear time. □

9 Conclusion

In this paper we solve the DIM problem in linear time for P_7-free graphs which answers an open question from [9]. Actually, we solve the minimum weight DIM problem in a robust way in the sense of [20]: Our algorithm either solves the problem correctly or finds out that the input graph has no d.i.m. or is not P_7-free. This avoids to recognize whether the input graph is P_7-free; the known recognition time bound is much worse than linear time. It is a challenging open question whether for some k, the DIM problem is $\mathbb{NP}$-complete for P_k-free graphs.

Acknowledgements The first author gratefully acknowledges a research stay at the LIMOS institute, University of Clermont-Ferrand, and the inspiring discussions with Anne Berry on dominating induced matchings.

References

1. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The Design and Analysis of Computer Algorithms. Addison-Wesley, Reading (1974)
2. Bandelt, H.-J., Mulder, H.M.: Distance-hereditary graphs. J. Comb. Theory **41**, 182–208 (1986)
3. Bodlaender, H.L., Brandstädt, A., Kratsch, D., Rao, M., Spinrad, J.: On algorithms for (P_5, gem)-free graphs. Theor. Comput. Sci. **349**, 2–21 (2005)
4. Brandstädt, A., Hundt, C., Nevries, R.: Efficient Edge Domination on Hole-Free graphs. In: Polynomial Time, Conference Proceedings LATIN 2010. Lecture Notes in Computer Science, vol. 6034, pp. 650–661 (2010)
5. Brandstädt, A., Kratsch, D.: On the structure of (P_5, gem)-free graphs. Discrete Appl. Math. **145**, 155–166 (2005)
6. Brandstädt, A., Le, H.-O., Mosca, R.: Chordal co-gem-free and (P_5, gem)-free graphs have bounded clique-width. Discrete Appl. Math. **145**, 232–241 (2005)

7. Brandstädt, A., Leitert, A., Rautenbach, D.: Efficient dominating and edge dominating sets for graphs and hypergraphs (2012). Extended abstract accepted for ISAAC 2012, Taiwan. arXiv:1207.0953v2 [cs.DM]
8. Bretscher, A., Corneil, D.G., Habib, M., Paul, Ch.: A simple linear time LexBFS cograph recognition algorithm. SIAM J. Discrete Math. **22**(4), 1277–1296 (2008)
9. Cardoso, D.M., Korpelainen, N., Lozin, V.V.: On the complexity of the dominating induced matching problem in hereditary classes of graphs. Discrete Appl. Math. **159**, 521–531 (2011)
10. Cardoso, D.M., Lozin, V.V.: Dominating induced matchings. In: "Graph Theory, Computational Intelligence and Thought", A Conference Celebrating Marty Golumbic's 60th Birthday, Jerusalem, Tiberias, Haifa, 2008. Lecture Notes in Computer Science, vol. 5420, pp. 77–86 (2009)
11. Corneil, D.G., Perl, Y., Stewart, L.K.: A linear recognition algorithm for cographs. SIAM J. Comput. **14**, 926–934 (1985)
12. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique width. Theory Comput. Syst. **33**, 125–150 (2000)
13. Golumbic, M.C., Rotics, U.: On the clique-width of some perfect graph classes. Int. J. Found. Comput. Sci. **11**, 423–443 (2000)
14. Grinstead, D.L., Slater, P.L., Sherwani, N.A., Holmes, N.D.: Efficient edge domination problems in graphs. Inf. Process. Lett. **48**, 221–228 (1993)
15. Hopcroft, J.E., Tarjan, R.E.: Efficient algorithms for graph manipulation [H]. Commun. ACM **16**(6), 372–378 (1973)
16. Livingston, M., Stout, Q.: Distributing resources in hypercube computers. In: Proceedings 3rd Conf. on Hypercube Concurrent Computers and Applications, pp. 222–231 (1988)
17. Lu, C.L., Ko, M.-T., Tang, C.Y.: Perfect edge domination and efficient edge domination in graphs. Discrete Appl. Math. **119**(3), 227–250 (2002)
18. Lu, C.L., Tang, C.Y.: Efficient domination in bipartite graphs. Manuscript (1997)
19. McConnell, R.M., Spinrad, J.P.: Modular decomposition and transitive orientation. Discrete Math. **201**, 189–241 (1999)
20. Spinrad, J.P.: Efficient Graph Representations. Fields Institute Monographs. Am. Math. Soc., Providence (2003)