Formalization of the generalized domination structure in graphs with combinatorial analysis

Misa Nakanishi *

Abstract

Following [3], we provide a technical paper. This paper addresses the minimum dominating set problem and generalizes the results for graphs with maximum degree 3 to general graphs. In addition, it complements the technical aspects of the results.
keywords : minimum dominating set, structure, NP-complete
MSC : 05C69

1 Introduction

This technical paper addresses the minimum dominating set problem and aims to determine the minimum dominating set of a graph, complementing and extending the results of [3]. A dominating set of a graph G is a set S of vertices of G such that every vertex v of G is either in S or adjacent to a vertex of S. A minimum dominating set is a dominating set of minimum size.

The minimum dominating set problem for cubic graphs is known to be NP-complete [1], which implies that the problem is NP-complete for general graphs as well. In this work, we extend the order-decreasing sequence used in the proof of the theorem in [3], thereby generalizing the result from graphs of maximum degree 3 to arbitrary graphs.

As a main part, this paper introduces three polynomial-time algorithms for graphs and provides technical refinements, ultimately leading that the problem lies in P, thus establishing that P = NP.

2 Notation

In this paper, a graph G is finite, undirected, and simple with the vertex set V and edge set E. We follow the notations presented in [2]. For a vertex $v \in V(G)$, the open neighborhood, denoted by $N_G(v)$, is $\{u \in V(G) \colon\ uv \in E(G)\}$, and the closed neighborhood, denoted by $N_G[v]$, is $N_G(v) \cup \{v\}$, also for a set $W \subseteq V(G)$, let $N_G(W) = \bigcup_{v \in W} N_G(v)$ and $N_G[W] = N_G(W) \cup W$. A *dominating set* $X \subseteq V(G)$ is such that $N_G[X] = V(G)$. A minimum dominating set, called

*E-mail address : nakanishi@mx-keio.net

a *d-set*, is a dominating set of minimum size. A vertex is *dominated* by its adjacent vertex in the dominating set. *X-3-path* is a path that has *labels* on the vertices so that the distance between the labels is 3 if the length of the path is at least 2, otherwise X-3-path is a path of length 1 that has at most one label on the vertices. Two cycles C_1 and C_2 are said to be *connecting without seams* if X-3-paths can be assigned to C_1 and C_2 as $C_1 \cap C_2$ is one X-3-path.

3 Algorithms and complements

We consider a connected graph G, otherwise consider each component one by one. First, we introduce the algorithm Scheme **K** as follows.

Algorithm 1: Scheme **K**: construction applied to a connected graph G - Part 1

Input : A connected graph G
Output: A resulting graph $\mathbf{K}(G)$ and a tree $T(G)$

if $|V(G)| \leq 2$ **then**
  $T \leftarrow \varnothing$;
  assign a label to $v \in V(G)$;
  add a vertex to T;
  return G *as* $\mathbf{K}(G)$ *and* T *as* $T(G)$;
$G_0 \leftarrow G$ and $k \leftarrow 0$;
$T_0 \leftarrow \varnothing$ and $l \leftarrow 0$;
repeat
  Let x be a cut vertex of G_k.
  foreach *pairs of distinct components* C_1, C_2 *of* $G_k - x$ **do**
    foreach *pairs of vertices* $u \in V(C_1)$, $v \in V(C_2)$ **do**
      add an edge uv to G_k;
      $k \leftarrow k + 1$;
until *there exists a cut vertex in* G_k
Find an induced cycle C in G_k.
if *length of* C *is 0 mod 3* **then**
  assign X-3-path to C in G_k, following the procedure for labeling $(*0)$;
  add C as a vertex to T_l;
  $l \leftarrow l + 1$;
else if *length of* C *is 2 mod 3* **then**
  take a 4-path $x_1x_2x_3x_4 \subseteq C$;
  add edges x_1x_3, x_2x_4, x_1x_4 to G_k;
  $k \leftarrow k + 1$;
  assign X-3-paths to 0 mod 3 induced cycles connecting without seams in the formed subgraph (call this subgraph (a)), following the procedure for labeling $(*0)$;
  add each 0 mod 3 induced cycle as a vertex and one of its connections with the 0 mod 3 induced cycles in the subgraph (a) as an edge to T_l;
  $l \leftarrow l + 1$;
else
  take a 4-path $x_1x_2x_3x_4 \subseteq C$;
  add edges x_1x_3 and x_2x_4 to G_k;
  $k \leftarrow k + 1$;
  assign X-3-paths to 0 mod 3 induced cycles connecting without seams in the formed subgraph (call this subgraph (b)), following the procedure for labeling $(*0)$;
  add each 0 mod 3 induced cycle as a vertex and one of its connections with the 0 mod 3 induced cycles in the subgraph (b) as an edge to T_l;
  $l \leftarrow l + 1$;
Find a shortest path P over all two vertices v_1 and v_2 lying on previously assigned X-3-path in G_k such that X-3-path has not been assigned to $\mathring{v}_1 P \mathring{v}_2$ yet.

Algorithm 2: Scheme **K** - Part 2

if *such a path P exists* **then**
  Let P' be the previously assigned X-3-path of length at least 2 containing v_1 and v_2.
  if *$v_1Pv_2P'v_1$ forms a cycle of length 0 mod 3 in G_k* **then**
    assign X-3-path to $v_1Pv_2P'v_1$ in G_k while maintaining or deleting the labels of existing X-3-path;
    add $v_1Pv_2P'v_1$ as a vertex and one of its connections with the 0 mod 3 cycles which have a part of $v_1P'v_2$ and contained as the vertices in T_l already as an edge to T_l;
    $l \leftarrow l + 1$;
  else if *$v_1Pv_2P'v_1$ forms a cycle of length 2 mod 3 in G_k* **then**
    take a 4-path in $v_1Pv_2P'v_1$ and add three edges to G_k as in subgraph (a);
    $k \leftarrow k + 1$;
    if *possible* **then**
      assign X-3-paths to 0 mod 3 induced cycles connecting without seams in the subgraph (a) while maintaining or deleting the labels of existing X-3-path, following the procedure for labeling ($*0$);
    else
      $k \leftarrow k - 1$ and **goto** step 8;
    add each 0 mod 3 induced cycle as a vertex and one of its connections with the 0 mod 3 induced cycles in the subgraph (a) (or one of its connections with the 0 mod 3 cycles which have a part of $v_1P'v_2$ and contained as the vertices in T_l already) as an edge to T_l;
    $l \leftarrow l + 1$;
  else if *$v_1Pv_2P'v_1$ forms a cycle of length 1 mod 3 in G_k* **then**
    take a 4-path in $v_1Pv_2P'v_1$ and add two edges to G_k as in subgraph (b);
    $k \leftarrow k + 1$;
    if *possible* **then**
      assign X-3-paths to 0 mod 3 induced cycles connecting without seams in the subgraph (b) while maintaining or deleting the labels of existing X-3-path, following the procedure for labeling ($*0$);
    else
      $k \leftarrow k - 1$ and **goto** step 17;
    add each 0 mod 3 induced cycle as a vertex and one of its connections with the 0 mod 3 induced cycles in the subgraph (b) (or one of its connections with the 0 mod 3 cycles which have a part of $v_1P'v_2$ and contained as the vertices in T_l already) as an edge to T_l;
    $l \leftarrow l + 1$;
  goto step 34 of Part 1 (continue the loop);
else
  break the loop;
return *G_k as $\mathbf{K}(G)$ and T_l as $T(G)$*;

If there are multiple ways to label vertices during the labeling process, we take the procedure for labeling ($*0$): Let U_0 be the set of all unlabeled vertices not dominated by the labeled vertices. For the two vertices y_1 and y_2, select y_1 if $N_{G_k}[y_2] \cap U_0 \subseteq N_{G_k}[y_1]$, select y_2 if $N_{G_k}[y_1] \cap U_0 \subseteq N_{G_k}[y_2]$, select the first taken vertex y_1 or y_2 otherwise.

Proposition 3.1 ([2])**.** *A graph is 2-connected if and only if it can be constructed from a cycle by successively adding H-paths to graphs H already constructed.*

Fact 3.1. *In a 2-connected graph, any two vertices are connected by a path along edges in some ear of its ear decomposition.*

Proof. The statement is proved by induction on ears. As a base case, if v and w in first ear (cycle), path exists trivially. Assume path exists for first k ears, add $(k+1)$-th ear if needed. 2-connectedness ensures every vertex is in some ear. We can concatenate paths along endpoints of ears. □

Proposition 3.2. *In the algorithm Scheme* **K** *- Part 1 step 34, it is possible to find previously assigned X-3-path between v_1 and v_2.*

Proof. By Fact 3.1, the claim holds. □

Proposition 3.3. *In the algorithm Scheme* **K** *- Part 1 step 34, create a cycle by choosing the shortest path connecting the two endpoints of X-3-path. If the length of the cycle is 1 mod 3 or 2 mod 3, choose a 4-path and add edges according to the rule. For some 4-path in the cycle, X-3-paths can be assigned to 0 mod 3 induced cycles connecting without seams in the subgraph while maintaining or deleting the labels of existing X-3-paths.*

Proof. Given the length of the existing X-3-path X (0, 1, 2 mod 3), the labeling method Y (3 rotation), the length of the cycle Z (1, 2 mod 3), and the 4-path selection method A ($\mathcal{O}(n^4)$ for $n = |V(G)|$), we obtain the proposition through simple case analysis of $X \times Y \times Z \times A$. □

Note that $\mathbf{K}(G)$ and $T(G)$ are not unique and constructed from G arbitrarily. By the scheme, $\mathbf{K}(G)$ is 2-connected with $V(G) = V(\mathbf{K}(G))$ and $E(G) \subseteq E(\mathbf{K}(G))$, and all edges in $\mathbf{K}(G)$ are covered by at least one set of X-3-paths but may be covered by other set of X-3-paths with the rotation of labels. We have a tree-structured list $T(G)$ of 0 mod 3 cycles connecting without seams. The label rotation follows this list. Choose an initial vertex to label in $V(G)$, and from there, assign X-3-path to the cycle in the list.

Consider labeling vertices by assigning X-3-paths to the cycles of length 0 mod 3 connecting without seams in $\mathbf{K}(G)$ following $T(G)$. Note that certain cycles may not be counted for the labeling, and have as few labeled vertices as possible.

Figure 1. Labeling vertices by the procedure for labeling ($*1$).

Remark 3.1 ([3])**.** *Let X be a dominating set of G. Every subset $D \subseteq X$ is a d-set of $G[N_G[D]]$ if and only if X is a d-set of G.*

Proposition 3.4. *(i) For every labeling, the set of all labeled vertices is a minimal dominating set of $\mathbf{K}(G)$. In addition, by selecting the first vertex to label, the remaining labeled vertices are uniquely determined. (ii) For at least one labeling, the set of all labeled vertices is a d-set of $\mathbf{K}(G)$.*

Proof. Since every edge in $\mathbf{K}(G)$ is covered by X-3-path, for every labeling, the set of all labeled vertices is an independent dominating set of $\mathbf{K}(G)$, that is, a minimal dominating set of $\mathbf{K}(G)$. After selecting the vertex to label first, if there are multiple ways to label vertices during the labeling process, those vertices are two adjacent vertices in $K^4 - e \subseteq \mathbf{K}(G)$ for some $e \in E(K^4)$. Now, we take the procedure for labeling ($*1$): Let U_0 be the set of all unlabeled vertices not dominated by the labeled vertices. For the two vertices y_1 and y_2, select y_1 if $N_{\mathbf{K}(G)}[y_2] \cap U_0 \subseteq N_{\mathbf{K}(G)}[y_1]$, select y_2 if $N_{\mathbf{K}(G)}[y_1] \cap U_0 \subseteq N_{\mathbf{K}(G)}[y_2]$, select the first taken vertex y_1 or y_2 otherwise. Then the remaining labeled vertices are determined following $T(G)$ uniquely. Therefore, the statement (i) holds. By Remark 3.1, the statement (ii) obviously holds. □

Let Y be a d-set of $\mathbf{K}(G)$ that is obtained by labeling. Let $\mathcal{Y}$ be the set of all Y. Suppose G has a cut vertex v and $w \in N_G(v)$ is not a cut vertex. For $Y_1, Y_2 \in \mathcal{Y}$ such that $Y_1 \cap Y_2 \neq \emptyset$, if $v \in Y_1 \setminus Y_2$ and $w \in Y_2 \setminus Y_1$, then let $\mathcal{Y}$ be $\mathcal{Y} \setminus Y_2$. Repeat this operation while they occur and let $\mathcal{A} = \mathcal{Y}$.

Proposition 3.5. *Suppose G has a cut vertex v and $w \in N_G(v)$ is not a cut vertex. For $Y_1, Y_2 \in \mathcal{Y}$ such that $Y_1 \cap Y_2 \neq \emptyset$, if $v \in Y_1 \setminus Y_2$ and $w \in Y_2 \setminus Y_1$, then Y_1 is better for a d-set of G.*

Proof. In the 0 mod 3 induced cycles constructed from v by the edge addition rules of the algorithm Scheme **K** - Part 1 step 8, w does not dominate any other neighbor of v in G. □

Definition 3.1. *For $\mathcal{A}' \subseteq \mathcal{A}$, define $\equiv$ so that two sets $A_1, A_2 \in \mathcal{A}'$ are equivalent if and only if every pair of sets in $\mathcal{A}'$ has nonempty intersection.*

Proposition 3.6. *Let X be a d-set of G. Let $\mathcal{X}$ be the set of all X. For some $\mathcal{B} \in \mathcal{A}/\equiv$ and for each $Y \in \mathcal{B}$, there exists $X \in \mathcal{X}$ such that $Y = X$, otherwise, for each $Y \in \mathcal{A}$, there exists $X \in \mathcal{X}$ such that $Y \subseteq X$.*

Proof. Let $\mathcal{B} \in \mathcal{A}/\equiv$. Consider labeling vertices in $\mathbf{K}(G)$. For any $Y_1, Y_2 \in \mathcal{B}$ such that $Y_1 \neq Y_2$, the vertices of $Y_1 \setminus Y_2$ and $Y_2 \setminus Y_1$ are interchangeable and equivalent. By the definition of $\mathbf{K}(G)$ and $\mathcal{B}$, it suffices that considering subgraph (a) or (b) of $\mathbf{K}(G)$. Now, $N_{\mathbf{K}(G)}[Y_1 \setminus Y_2] = N_{\mathbf{K}(G)}[Y_2 \setminus Y_1]$. If Y_1 is a dominating set of G, then for some $X \in \mathcal{X}$, $X = Y_1$. Now, $N_G[Y_2 \setminus Y_1] \setminus N_G[Y_1 \setminus Y_2] \subseteq N_G[Y_1] = V(G)$. By considering the subgraph (a), we have $N_G[Y_1 \setminus Y_2] \setminus N_G[Y_2 \setminus Y_1] \subseteq N_G[Y_2]$. Hence, $N_G[Y_1] = N_G[Y_2]$. Suppose that for all $Y \in \mathcal{A}$, Y is not a dominating set of G. Now, each Y_1 and Y_2 is not a dominating set of G. By considering the subgraph (b), we have $N_G[Y_2 \setminus Y_1] \setminus N_G[Y_1 \setminus Y_2] \not\subseteq N_G[Y_1]$ and $N_G[Y_1 \setminus Y_2] \setminus N_G[Y_2 \setminus Y_1] \not\subseteq N_G[Y_2]$. Hence, $N_G[Y_1] \setminus N_G[Y_2] \neq \emptyset$ and $N_G[Y_2] \setminus N_G[Y_1] \neq \emptyset$. Let $\mathcal{B}_1, \mathcal{B}_2 \in \mathcal{A}/\equiv$ with $\mathcal{B}_1 \neq \mathcal{B}_2$. For any $Y_3 \in \mathcal{B}_1$ and any $Y_4 \in \mathcal{B}_2$, since $Y_3 \cap Y_4 = \emptyset$ and each Y_3 and Y_4 is not a dominating set of G, it follows that $N_G[Y_3] \setminus N_G[Y_4] \neq \emptyset$ and $N_G[Y_4] \setminus N_G[Y_3] \neq \emptyset$. Now, Y is a d-set of $G[N_G[Y]]$. Let W be a subset of $V(G) \setminus Y$ with minimum size such that $Y \cup W$ is a dominating set of G. By Remark 3.1, $Y \cup W$ is a d-set of G. □

Proposition 3.7. *Every $\mathcal{B} \in \mathcal{A}/\equiv$ is determined in polynomial time and $|\mathcal{A}/\equiv| \leq |V(G)|$.*

Proof. By the algorithm Scheme **K**, $\mathbf{K}(G)$ and $T(G)$ are constructed from G in polynomial time. By Proposition 3.4, for each $v \in V(G)$ to label first, the equivalence class $\mathcal{B} \in \mathcal{A}/\equiv$ is defined as for the label set $B(v)$, $\{B(v)\} = \mathcal{B}$, if $B(v)$ is a d-set of $\mathbf{K}(G)$ and $B(v) \in \mathcal{A}$. Therefore, $\{B(v)\} = \mathcal{B} \in \mathcal{A}/\equiv$ is determined in polynomial time and $|\mathcal{A}/\equiv| \leq |V(G)|$. □

Second, we introduce the algorithm Labeling-and-Prune procedure as follows. It returns a label set $L(G)$ of G, which is $Y \in \mathcal{B} \in \mathcal{A}/\equiv$ contained as the subset in a d-set of G and is a d-set of $\mathbf{K}(G)$.

Algorithm 3: Labeling-and-Prune procedure: labeling method on $\mathbf{K}(G)$

Input : A connected graph G, $\mathbf{K}(G)$, $T(G)$
Output: A label set $L(G)$

if $|V(G)| \leq 2$ **then**
  $L \leftarrow \varnothing$;
  add a vertex in $V(G)$ to L;
  return L *as* $L(G)$;
$\mathcal{L} \leftarrow \varnothing$;
foreach $v \in V(G)$ **do**
  choose v as the initially labelled vertex;
  create a temporary labelling L;
  assign X-3-path to an 0 mod 3 cycle C contained as the vertex in $T(G)$ such that $v \in C$ and add labels to L;
  for *each 0 mod 3 cycle* D *contained as the adjacent vertex in* $T(G)$ **do**
    assign X-3-path to D while maintaining or deleting the labels of existing X-3-path, following the procedure for labeling $(*1)$;
    add labels to L;
  add L to $\mathcal{L}$ and record its size $|L|$;
$\mathcal{Y} \leftarrow \{Y \in \mathcal{L} \colon |Y| \text{ is minimum}\}$;
repeat
  foreach *cut vertex* v *of* G **do**
    foreach *neighbor* $w \in N_G(v)$ *with* w *not a cut vertex* **do**
      foreach $Y_1, Y_2 \in \mathcal{Y}$ *with* $Y_1 \cap Y_2 \neq \varnothing$ **do**
        if $v \in Y_1 \setminus Y_2$ ***and*** $w \in Y_2 \setminus Y_1$ **then**
          $\mathcal{Y} \leftarrow \mathcal{Y} \setminus \{Y_2\}$;
until *no change to* $\mathcal{Y}$
$\mathcal{A} \leftarrow \mathcal{Y}$;
foreach $Y \in \mathcal{A}$ **do**
  if Y *is a dominating set of* G **then**
    return Y *as* $L(G)$;
choose any $Y \in \mathcal{A}$ and **return** Y *as* $L(G)$;

Proposition 3.8. *In the algorithm Labeling-and-Prune procedure,* $L(G)$ *is computed in polynomial time.*

Proof. By Proposition 3.4, $|\mathcal{L}| \leq |V(G)|$. The combinations of a cut vertex and its neighbor are $\mathcal{O}(n^2)$ for $n = |V(G)|$. By Proposition 3.6 and Proposition 3.7, $|\mathcal{A}| \leq |V(G)|$, so the claim holds. □

Third, we introduce the algorithm extendWithProp3.9 as follows. For a graph F, it returns a minimum dominating set of F, which is written by $X(F)$.

Proposition 3.9 ([3]). *Let $Y = L(G)$. Let G' be a graph obtained by deleting Y from, and adding edges to all vertex pairs of $\bigcup_{y\in Y} N_G(y)$ to G. Let Z_1 be a d-set of G'. Let $G'' = G - N_G[Y]$ and Z_2 be a d-set of G''. If $|Z_1| < |Z_2|$, then $Y \cup Z_1$ is a d-set of G, and $Z_1 \cap \bigcup_{y\in Y} N_G(y) \neq \emptyset$. If $|Z_1| \geq |Z_2|$, then $Y \cup Z_2$ is a d-set of G.*

Algorithm 4: extendWithProp3.9: determination of the remaining vertices for a d-set

Input : A graph F, $L(G)$ for a component G of F
Output: A minimum dominating set $X(F)$

if $F = \varnothing$ **then**
  return $\varnothing$ *as* $X(F)$;
$W \leftarrow \varnothing$;
foreach *component* G *of* F **do**
  $S \leftarrow L(G)$;
  if $V(G) \setminus N_G[S] = \varnothing$ **then**
    $W \leftarrow W \cup S$;
    goto step 4;
  Let G' be constructed by deleting S, and for every pair $w_1, w_2 \in \bigcup_{s\in S} N_G(s)$, adding an edge w_1w_2 to G.
  $Z_1 \leftarrow$ `extendWithProp3.9`(G');
  $G'' \leftarrow G - N_G[S]$;
  $Z_2 \leftarrow \varnothing$;
  foreach *component* C *of* G'' **do**
    if $|V(C)| = 1$ **then**
      $Z_2 \leftarrow Z_2 \cup V(C)$;
    else if C *is a cycle* **then**
      set Z to a d-set of C;
      $Z_2 \leftarrow Z_2 \cup Z$;
    else if C *is a path* **then**
      set Z to a d-set of C;
      $Z_2 \leftarrow Z_2 \cup Z$;
    else
      $Z \leftarrow$ `extendWithProp3.9`(C);
      $Z_2 \leftarrow Z_2 \cup Z$;
  if $|Z_1| < |Z_2|$ **then**
    $Z_{\text{chosen}} \leftarrow Z_1$;
  else
    $Z_{\text{chosen}} \leftarrow Z_2$;
  $W \leftarrow W \cup S \cup Z_{\text{chosen}}$;
return W *as* $X(F)$;

Theorem 3.1. *The d-set of a graph is determined in polynomial time.*

Proof. It follows the algorithm extendWithProp3.9. The computation of $L(G)$ for a component G of the graph F is in polynomial time by Proposition 3.8. The remaining vertices of a d-set of F from $\bigcup L(G)$ are derived from Lamma 3.9 with the recursion of extendWithProp3.9 for G' and G'', which have the order-decreasing sequence. The number of recursions in extendWithProp3.9 is $\mathcal{O}(n^2)$ for $n = |V(G)|$. Hence, all computations in extendWithProp3.9, including for the number of components and the definitions of G' and G'', can be performed in polynomial time. □

References

[1] P. Alimonti and V. Kann: Some APX-completeness results for cubic graphs, Theoretical Computer Science, 237 (2000), pp. 123-134.

[2] R. Diestel: Graph Theory Fourth Edition. Springer (2010)

[3] M. Nakanishi: Generalized domination structure in cubic graphs, Journal of Combinatorial Mathematics and Combinatorial Computing, 124 (2025), pp. 731-736.