Chapter 12
CDS in Planar Graphs

Simple, geometric forms and planar surfaces
define Jeep Patriot's timeless, purpose-built design.
TREVOR CREED

12.1 Motivation and Overview

Although MIN-CDS in general graphs is hard to approximate, the restriction to certain special graph classes admits much better approximation results. MIN-CDS in planar graphs remains NP-hard even for planar graphs that are regular of degree 4 [57]. The related problem, MIN-DS in planar graphs, is also NP-hard even for planar graphs with maximum vertex degree 3 and planar graphs that are regular of degree 4 [57]. It is well known that MIN-DS in planar graphs possesses a polynomial-time approximation scheme (PTAS) based on the shifting strategy [3]: For any constant $\varepsilon > 0$, there is a polynomial-time $(1+\varepsilon)$-approximation algorithm. Thus, it is immediate to conclude that MIN-CDS in planar graphs can be approximated within a factor $3+\varepsilon$ for any $\varepsilon > 0$ in polynomial time. However, the degree of the polynomial grows with $1/\varepsilon$ and hence, the approximation scheme is hardly practical.

In this chapter, we present a simple heuristic for MIN-CDS in general graphs developed in [105]. When running on graphs excluding K_m (the complete graph of order m) as a minor, the heuristic has an approximation ratio of at most 7 if $m = 3$, or at most $\frac{m(m-1)}{2} + 5$ if $m \geq 4$. In particular, if running on a planar graph, the heuristic has an approximation ratio of at most 15. The remaining of this chapter is organized as follows. In Sect. 12.2, we introduce some related graph-theoretic concepts and parameters. In Sect. 12.3, we describe the heuristic for MIN-CDS in general graphs. In Sect. 12.4, we provide an upper bound on the cardinality of the CDS output by the heuristic.

D.-Z. Du and P.-J. Wan, *Connected Dominating Set; Theory and Applications*, Springer Optimization and Its Applications 77, DOI 10.1007/978-1-4614-5242-3_12,

12.2 Preliminaries

Let $G = (V,E)$ be a graph. We sometimes write $V(G)$ instead of V and $E(G)$ instead of E. For any $U \subseteq V$, we use $G[U]$ to denote the subgraph of G induced by U. The distance $\mathrm{dist}_G(u,v)$ in G of two vertices $u,v \in V(G)$ is the length of a shortest path between u and v in G. The distance between a vertex v and a set $U \subseteq V(G)$ is

$$\min_{u\in U} \mathrm{dist}_G(u,v).$$

The distance between two subsets U and W of $V(G)$ is

$$\min_{u\in U}\min_{w\in W} \mathrm{dist}_G(u,w).$$

A vertex set $U \subseteq V(G)$ is a k-independent set (k-IS) of G if the distance between any pair of vertices in U is greater than k. The *k-independence number* of G, denoted by $\alpha_k(G)$, is the largest cardinality of a k-IS. Note that a 1-IS is a usual IS and $\alpha_1(G)$ is the usual independence number $\alpha(G)$. The domination number of G, denoted by $\gamma(G)$, the connected domination number of G, denoted by $\gamma_c(G)$, and $\alpha_2(G)$ are related by the following inequality [44].

$$\alpha_2(G) \le \gamma(G) \le \gamma_c(G).$$

To see why $\alpha_2(G) \le \gamma(G)$, let $U \subseteq V(G)$ be a maximum 2-IS of G. For each $u \in U$, let $N_G[u]$ denote the closed neighborhood of u in G. Then the closed neighborhoods $N_G[u]$ for all $u \in U$ are pairwise disjoint. Thus, each dominating set of G must contain at least one vertex from each $N_G[u]$. This implies that $\gamma(G) \ge \alpha_2(G)$.

A *contraction* of an edge (u,v) in G is made by identifying u and v with a new vertex whose neighborhood is the union of the neighborhoods of u and v (with resulting multiple edges and self-loops deleted). A *contraction* of G is a graph obtained from G by a sequence of edge contractions. A graph H is a *minor* of G if H is the contraction of a subgraph of G. G is *H-free* if G has no minor isomorphic to H. For example, by Kuratowski's theorem, a graph is planar if and only if it is both K_5-free and $K_{3,3}$-free. In this chapter, we focus on K_m-free graphs. Our algorithm would find a CDS of size at most

$$\left(\frac{m(m-1)}{2}+5\right)\alpha_2(G)-5$$

of a K_m-free graph G for any $m \ge 4$. This implies that if G is K_m-free for some $m \ge 4$, then

$$\gamma_c(G) \le \left(\frac{m(m-1)}{2}+5\right)\alpha_2(G)-5.$$

In particular, for a planar graph G,

$$\gamma_c(G) \le 15\alpha_2(G)-5.$$

12.3 Algorithm Description

We first give a brief overview on the algorithm design. The algorithm is presented as a color-marking and time-stamping process. Each vertex maintains one of the three colors: black, gray, and white, which is initially white. In addition, each vertex maintains a set of stamps, which is initially empty. The algorithm runs in proceeds in iterative phases. In the kth phase, a subset B_k of nonblack vertices are marked with black, and all of their gray neighbors are stamped with the phase number (interpreted as the time) k while keeping all previous stamps, and all of their white neighbors are marked gray and stamped with the current phase number k. At the end of the kth phase, all black nodes have to be connected, and each white vertex, if there is any left, has a gray neighbor with time-stamp j for every $1 \leq j \leq k$. The algorithm ends when no white vertex is left and outputs all black vertices which form a CDS.

Now, we describe the algorithm. For the simplicity of description, we introduce some new terms and notations. Given a color marking of all vertices of G, the *deficiency graph* is the graph obtained from G by first removing all black vertices and those gray vertices without white neighbors, and then removing edges between gray vertices. Thus, each vertex of a deficiency graph is either white or gray, and each connected component of a deficiency graph must have at least one white vertex. Given a vertex v and a positive integer k, we use **MarkStamp**(v,k) to denote the basic operation which marks v black and all white neighbors of v gray and stamps v and all its nonblack neighbors with k.

Consider a connected graph H and a positive integer k which satisfy the following properties: Each vertex of H is either white or gray and at least one vertex is white. If $k = 1$, then all vertices are white; and otherwise, every white vertex is adjacent to a gray vertex stamped with j for every $1 \leq j \leq k-1$. Such pair (H,k) is referred to as a *residue pair*. A *restricted connected 2-dominating set* (RC2DS) of a residue pair (H,k) is a subset of vertices U of H satisfying that:

- $H[U]$ is connected.
- Every white vertex not in U, if there is any, is at a distance of exactly two from U.
- And for every $1 \leq j \leq k-1$, at least one vertex in U has a stamp j

We present a simple procedure, called **RC2DS**(H,k), which takes a residue pair (H,k) as input and produces a RC2DS for (H,k) which are marked black and a color marking and time-stamping of the remaining vertices. The procedure **RC2DS**(H,k) consists of four steps:

- Step 1: Initialization. If $k \geq 2$, let $a_j = 0$ for $j = 1,\ldots,k-1$.
- Step 2: Sorting. Build a spanning tree T of H rooted at a white vertex, and compute a breadth-first-search order $v_1, v_2, \ldots, v_s$ of all white vertices in H with respect to T.
- Step 3: Coloring and Stamping. **MarkStamp**(v_1,k). For $i = 2$ to s, if v_i is white and has no gray neighbors stamped with k, proceed as follows:

(a) Set $l = 1$, $u_1 = v_i$. Repeat the following iteration until u_l is black: If $k \geq 2$ and u_l is gray, set $a_j = 1$ for each stamp $j < k$ of u_l. If u_l has a black neighbor, set u_{l+1} to any such neighbor; otherwise, if u_l has a gray neighbor stamped with k, set u_{l+1} to any such neighbor; otherwise, set u_{l+1} to its parent in T. Increment l by 1.
(b) Repeat the following iteration until $l = 1$: Decrement l by 1 and invoke **MarkStamp**(u_l, k).

- Step 4: Post-processing. If $k \geq 2$, perform the following processing. For $j = 1$ to $k-1$, if $a_j = 0$, choose a gray neighbor u of v_1 stamped with j, set $a_t = 1$ for each stamp $t < k$ of u, and then **MarkStamp**(u, k).

The $k-1$ boolean variables a_j for $1 \leq j \leq k-1$ indicate whether at least one black vertex has a stamp j. They are initialized to zero in Step 1. Whenever a gray vertex with stamp j is marked black at Step 3 or Step 4, a_j is set to one. Step 4 ensures that all these boolean variables are one eventually.

The for-loop in Step 3 guarantees that in the end, every white vertex is adjacent to a gray vertex stamped with k, and thus is exactly two hops away from some black vertex. The inner loop in Step 3(a) establishes a path from a white vertex v_i without gray neighbors stamped with k to some black vertex. Let P_i be the subpath of this path excluding the black end-vertex. The inner loop in Step 3(b) invokes **MarkStamp**(u, k) for all vertices in P_i. We claim that P_i consists of either three or four vertices. Indeed, u_1 is v_i, and since v_i is white and has no gray neighbors stamped with k, u_2 is always set to the parent of v_i. Depending on the color of u_2, we consider two cases:

Case 1: u_2 is white. Then u_2 must have a gray neighbor stamped with k as early as when u_2 is examined, for otherwise, it would have been marked black. Thus, u_3 is a gray neighbor of u_2 stamped with k, and hence P_i consists of the three vertices u_1, u_2 and u_3.

Case 2: u_2 is gray. Then every stamp of u_2 is less than k. We further consider two subcases.

Subcase 2.1: At least one gray neighbor of u_2 has stamp k. Then u_3 is one of such gray neighbors and P_i just consists of the three vertices u_1, u_2 and u_3.

Subcase 2.2: None of the gray neighbors of u_2 has stamp k. As u_2 is not adjacent to any black vertex, u_3 is the parent of u_2. Since no gray vertices with stamps less than k are adjacent in H, u_3 must be white. Then at least a gray neighbor of u_3 is stamped with k as early as when u_3 is examined, for otherwise, u_3 would have been marked black. Thus, u_4 is one of such gray neighbors, and P_i just consists of the four vertices u_1, u_2, u_3, and u_4.

In summary, the path consists of either three vertices or four vertices. Furthermore, if the path consists of four vertices, then k must be greater than one and at least one a_j is set to one for some $1 \leq j \leq k-1$ in Step 3(a).

Now we are ready to describe the algorithm, denoted by **MarkStamp**(G), for finding a CDS of G. Initially, $k = 0$, and all vertices of G have white colors. Repeat the following iteration while there are some white vertices left:

- Increment k by 1 and construct the deficiency graph G_k.
- For each connected component H of G_k, apply **RC2DS**(H,k).

Let B denote the set of black vertices produced by **MarkStamp**(G). It is easy to see that B is a CDS of G. In the next section, we will provide an upper bound on $|B|$ if the graph G is free of K_m-minor for some $m \geq 3$.

12.4 Performance Analysis

The main theorem of this section is given below.

Theorem 12.4.1. *Suppose that G is free of K_m-minor for some $m \geq 3$. If $m = 3$, then*

$$|B| \leq 7\alpha_2(G) - 4.$$

If $m \geq 4$, then

$$|B| \leq \left(\frac{m(m-1)}{2} + 5\right)\alpha_2(G) - 5.$$

By Kuratowski's theorem, a planar graph has no K_5-minor. So we have the following corollary of Theorem 12.4.1.

Corollary 12.4.2. *If G is a planar graph, then*

$$|B| \leq 15\alpha_2(G) - 5.$$

Since

$$\alpha_2(G) \leq \gamma(G) \leq \gamma_c(G),$$

Theorem 12.4.1 implies that when running on a graph G excluding K_m as a minor, the algorithm **MarkStamp**(G) has an approximation ratio of at most 7 if $m = 3$ or at most $\frac{m(m-1)}{2} + 5$ if $m \geq 4$. In particular, if running on a planar graph, the algorithm has an approximation ratio of at most 15. The remaining of this section is dedicated to the proof for Theorem 12.4.1.

Let H be a graph in which every vertex is either white or gray and there is at least one white vertex. A *restricted 2-independent set* (R2IS) is a 2-IS of H which consists of only white vertices. The *restricted* 2*-independence number* of H, denoted by $\alpha_2'(H)$, is the largest cardinality of an R2IS of H. Obviously, $\alpha_2'(H) \leq \alpha_2(H)$. The next lemma presents the "monotonic" properties of the deficiency graphs.

Lemma 12.4.3. *Suppose that **MarkStamp**(G) runs in l iterations. Then*

$$G = G_1 \supset G_2 \supset \cdots \supset G_l;$$

$$\alpha_2(G) = \alpha_2'(G_1) \geq \alpha_2'(G_2) \geq \cdots \geq \alpha_2'(G_l).$$

Proof. It is obvious that $G_1 = G$ and $\alpha_2'(G_1) = \alpha_2(G)$. Fix a k between 1 and $l-1$. We prove that $G_{k+1} \subset G_k$ and $\alpha_2'(G_{k+1}) \leq \alpha_2'(G_k)$.

We first show that $V(G_{k+1}) \subset V(G_k)$. Note that all white vertices of G_{k+1} must have been white in the previous iteration and thus are white vertices of G_k as well. In addition, all gray vertices of G_{k+1} which are white in the previous iteration must be white vertices of G_k. So it is sufficient to show that each gray vertex of G_{k+1} which is also gray in the previous iteration is also a vertex of G_k. Let v be a gray vertex of G_{k+1} which is also gray in the previous iteration. Then v has a white neighbor, denoted by u, in G_{k+1}. Since u is also a white vertex of G_k, v must be also a gray vertex of G_k.

Next, we show that $E(G_{k+1}) \subset E(G_k)$. Consider any edge uv of G_{i+1}. Then at least one of its endpoints is white. By symmetry, assume v is white. If u is also white, then the edge uv also appears in G_k. If u is gray, then u is either white or gray in G_k. In either case, the edge uv appears in G_k.

Finally, we show that $\alpha_2'(G_{k+1}) \leq \alpha_2'(G_k)$. Let w_1 and w_2 be any pair of white nodes of G_{k+1}. As G_{k+1} is a subgraph of G_k,

$$\mathrm{dist}_{G_{k+1}}(w_1, w_2) \geq \mathrm{dist}_{G_k}(w_1, w_2).$$

We claim that, however, if $\mathrm{dist}_{G_k}(w_1, w_2) \leq 2$, then

$$\mathrm{dist}_{G_{k+1}}(w_1, w_2) = \mathrm{dist}_{G_k}(w_1, w_2).$$

The claim is true if $\mathrm{dist}_{G_k}(w_1, w_2) = 1$. So we assume that $\mathrm{dist}_{G_k}(w_1, w_2) = 2$. Then $\mathrm{dist}_{G_{k+1}}(w_1, w_2) \geq 2$. Let v be a common neighbor of w_1 and w_2 in G_k. Then v must remain as a vertex of G_{k+1}, for otherwise, v would have been marked black in the previous iteration and both w_1 and w_2 would have become gray in G_{k+1}. Thus, $\mathrm{dist}_{G_{k+1}}(w_1, w_2) = 2$. So our claim is true. From the claim, we conclude that if $\mathrm{dist}_{G_{k+1}}(w_1, w_2) > 2$, then $\mathrm{dist}_{G_k}(w_1, w_2) > 2$. This implies that $\alpha_2'(G_{k+1}) \leq \alpha_2'(G_k)$. □

The lemma below gives an upper bound on the total number of iterations if the graph G is free of K_m-minor.

Lemma 12.4.4. *If G is free of K_m-minor for some $m \geq 3$, then* ***MarkStamp***(G) *runs in at most $m-1$ iterations.*

Proof. We prove the lemma by contradiction. Assume that G is free of K_m-minor but **MarkStamp**(G) runs in at least m iterations. Let H_m^* be an arbitrary connected component of G_m. By Lemma 12.4.3, for each $1 \leq k \leq m-1$, G_k has a unique connected component, denoted by H_k^*, which contains H_m^* as a subgraph. Obviously,

$$H_1^* \supset H_2^* \supset \cdots \supset H_m^*.$$

For each $1 \leq k \leq m$, let B_k^* be the set of black vertices of H_k^* marked by the procedure **RC2DS**(H_k^*, k). Then for any $1 \leq i < j \leq m$, B_i^* and B_j^* are disjoint and separated by one hop as at least one vertex in B_j^* has a stamp i. Since each B_k^* is connected, the m sets $B_1^*, B_2^*, \ldots, B_m^*$ give rise to a K_m-minor in G, which is a contradiction. Thus, the lemma holds. □

The next lemma provides an upper bound on the number of black vertices produced by the procedure **RC2DS**(H,k).

Lemma 12.4.5. *The number of black vertices produced by the procedure* ***RC2DS***(H,k) *is at most* $3\alpha_2'(H)-2$ *if* $k=1$, *and at most* $4\alpha_2'(H)+k-4$ *if* $k \geq 2$.

Proof. Let $v_1, v_2, \ldots, v_s$ be the ordering of the white vertices of H produced by Step 2 of the procedure **RC2DS**(H,k). Let I be the set of integers i in $\{2,\ldots,s\}$ such that when v_i is examined in the for-loop of Step 3, v_i is white and has no gray neighbors stamped with k. It is obvious that $\{v_i : i \in \{1\} \cup I\}$ form an R2IS of H. Thus,

$$1+|I| \leq \alpha_2'(H).$$

Next, we count the number of vertices marked black during each iteration i with $i \in I$ in the for-loop of Step 3. Fix an $i \in I$. From the explanation after the procedure **RC2DS**(H,k) in the previous section, either three or four vertices are marked black during iteration i. In addition, if four vertices are marked black in this iteration, then k must be greater than one and at least one a_j is set to one for some $1 \leq j \leq k-1$.

Finally, we count the total number of black vertices. Note that v_1 is always marked black. If for each $i \in I$, the iteration i of the for-loop at Step 3 marks exactly three vertices black, then Step 4 marks at most $k-1$ additional vertices black. So the total number of black vertices is at most

$$\begin{aligned} &1+3|I|+k-1 \\ &\quad = 3(1+|I|)+k-3 \\ &\quad \leq 3\alpha_2'(H)+k-3. \end{aligned}$$

If for some $i \in I$, the iteration i of the for-loop at Step 3 marks four vertices black, then $k > 1$ and Step 4 marks at most $k-2$ additional vertices black. So the total number of black vertices is at most

$$\begin{aligned} &1+4|I|+k-2 \\ &\quad = 4(1+|I|)+k-5 \\ &\quad \leq 4\alpha_2'(H)+k-5. \end{aligned}$$

Thus, if $k=1$, the total number of black vertices is at most

$$3\alpha_2'(H)+1-3 = 3\alpha_2'(H)-2.$$

If $k \geq 2$, the total number of black vertices is at most

$$\max\left\{3\alpha_2'(H)+k-3, 4\alpha_2'(H)+k-5\right\}$$
$$\leq 4\alpha_2'(H)+k-4.$$

Therefore, the lemma holds. □

The next lemma gives upper bounds on the number of black vertices produced in each iteration of **MarkStamp**(G).

Lemma 12.4.6. *Let B_k be the set of black vertices produced in the kth iteration of* ***MarkStamp***(G)*. Then*

$$|B_1| \leq 3\alpha_2(G)-2,$$
$$|B_2| \leq 4\alpha_2(G)-2,$$
$$|B_3| \leq 4\alpha_2(G)-1,$$
$$|B_k| \leq k\alpha_2(G), \quad k \geq 4.$$

Proof. From Lemmas 12.4.5 and 12.4.3, $|B_1| \leq 3\alpha_2(G)-2$. So we assume that $k > 1$. Suppose that G_k has t connected components, denoted by $H_{k,1}, \ldots, H_{k,t}$. Since each connected component contains at least one white vertex,

$$1 \leq t \leq \sum_{i=1}^{t} \alpha_2'\left(H_{k,i}\right) = \alpha_2'(G_k).$$

For each $1 \leq i \leq t$, let $B_{k,i}$ be the vertices of $H_{k,i}$ produced by the procedure **RC2DS**$\left(H_{k,i}, k\right)$. Then

$$B_k = B_{k,1} \cup \cdots \cup B_{k,t};$$

and by Lemma 12.4.5,

$$\left|B_{k,i}\right| \leq 4\alpha_2'\left(H_{k,i}\right)+k-4$$

for each $1 \leq i \leq t$. Thus, if $k = 2$ or 3, by Lemma 12.4.3, we have

$$|B_k| = \sum_{i=1}^{t}\left|B_{k,i}\right|$$
$$\leq 4\sum_{i=1}^{t}\alpha_2'\left(H_{k,i}\right)+(k-4)t$$
$$= 4\alpha_2'(G_k)+(k-4)t$$
$$\leq 4\alpha_2(G)+(k-4).$$

If $k \geq 4$, by Lemma 12.4.3 we have

$$\begin{aligned} |B_k| &= \sum_{i=1}^{t} |B_{k,i}| \\ &\leq 4 \sum_{i=1}^{t} \alpha_2'\left(H_{k,i}\right) + (k-4)t \\ &\leq 4\alpha_2'(G_k) + (k-4)\alpha_2'(G_k) \\ &= k\alpha_2'(G_k) \\ &\leq k\alpha_2(G). \end{aligned}$$

So, the lemma holds. □

Now we are ready to give the proof of Theorem 12.4.1. By Lemma 12.4.4, the total number of iterations is at most $m-1$. If $m=3$, then by Lemma 12.4.6,

$$|B| \leq (3\alpha_2(G) - 2) + (4\alpha_2(G) - 2) \leq 7\alpha_2(G) - 4.$$

If $m = 4$, then by Lemma 12.4.6,

$$\begin{aligned} |B| &\leq (7\alpha_2(G) - 4) + (4\alpha_2(G) - 1) \\ &= 11\alpha_2(G) - 5 \\ &= \left(\frac{m(m-1)}{2} + 5\right)\alpha_2(G) - 5. \end{aligned}$$

If $m > 4$, by Lemma 12.4.6,

$$\begin{aligned} |B| &\leq (11\alpha_2(G) - 5) + \sum_{k=4}^{m-1} k\alpha_2(G) \\ &= 11\alpha_2(G) - 5 + \left(\frac{m(m-1)}{2} - 6\right)\alpha_2(G) \\ &= \left(\frac{m(m-1)}{2} + 5\right)\alpha_2(G) - 5. \end{aligned}$$

This completes the proof of Theorem 12.4.1. □