Communications in Combinatorics and Optimization
Vol. xx, No. x (xxxx), pp. 1-21
https://doi.org/10.22049/cco.2024.28647.1650

CCO
Commun. Comb. Optim.

Research Article

Total Roman domination and total domination in unit disk graphs

Sasmita Rout[1,†], Pawan Kumar Mishra[2] and Gautam Kumar Das[1,*]

[1] Department of Mathematics, Indian Institute of Technology Guwahati, Assam, India
†sasmita18@iitg.ac.in
*gkd@iitg.ac.in

[2] Department of Computer Science and Engineering, IIIT Guwahati, Assam, India
pawan.mishra@iiitg.ac.in

Received: 12 May 2023; Accepted: 24 January 2024
Published Online: 5 February 2024

Abstract: Let $G = (V, E)$ be a simple, undirected and connected graph. A Roman dominating function (RDF) on the graph G is a function $f : V \rightarrow \{0, 1, 2\}$ such that each vertex $v \in V$ with $f(v) = 0$ is adjacent to at least one vertex $u \in V$ with $f(u) = 2$. A total Roman dominating function (TRDF) of G is a function $f : V \rightarrow \{0, 1, 2\}$ such that (i) it is a Roman dominating function, and (ii) the vertices with non-zero weights induce a subgraph with no isolated vertex. The total Roman dominating set (TRDS) problem is to minimize the associated weight, $f(V) = \sum_{u \in V} f(u)$, called the total Roman domination number ($\gamma_{tR}(G)$). Similarly, a subset $S \subseteq V$ is said to be a total dominating set (TDS) on the graph G if (i) S is a dominating set of G, and (ii) the induced subgraph $G[S]$ does not have any isolated vertex. The objective of the TDS problem is to minimize the cardinality of the TDS of a given graph. The TDS problem is NP-complete for general graphs. In this paper, we propose a simple 10.5-factor approximation algorithm for TRDS problem in UDGs. The running time of the proposed algorithm is $O(|V| \log k)$, where k is the number of vertices with weights 2. It is an improvement over the best-known 12-factor approximation algorithm with running time $O(|V| \log k)$ available in the literature. Next, we propose another algorithm for the TDS problem in UDGs, which improves the previously best-known approximation factor from 8 to 7.79. The running time of the proposed algorithm is $O(|V| + |E|)$.

Keywords: total domination, total Roman domination, unit disk graphs, approximation algorithm.

AMS Subject classification: 05C69

* *Corresponding Author*

1. Introduction

Let $G = (V, E)$ be a simple, undirected, and connected graph, where $V(G)$ and $E(G)$ are the vertex set and edge set of G, respectively[1]. Given a vertex $v \in V(G)$, a set $N_G(v)$ denotes the open neighborhood of v, and it is defined as $N_G(v) = \{u \in V : uv \in E(G)\}$. On the other hand, the closed neighborhood $N_G[v]$ of v is defined as $N_G[v] = N_G(v) \cup \{v\}$. For any subset $S \subseteq V$, $G[S]$ represents the subgraph induced by the vertex set S in G (i.e., for each $x, y \in S$, $xy \in E(G[S])$ if and only if $xy \in E(G)$). The boundary of a set $T \subseteq V(G)$ is the set $B(T)$ such that $B(T) = N_G(T) \setminus T$. A subset $D \subseteq V(G)$ is said to be a dominating set (DS) of G if each vertex $v \in V(G)$, $|N_G[v] \cap D| \geq 1$. We denote the domination number as $\gamma(G)$, and it is the minimum cardinality among all dominating sets in G. A vertex $v \in V(G)$ dominates $N_G[v]$, and a subset $S \subseteq V(G)$ dominates $\bigcup_{v \in S} N_G[v]$. A subset $D_t \subseteq V(G)$ is said to be a total dominating set (TDS) of G if (i) D_t is a dominating set of G (domination property), and (ii) the induced subgraph $G[D_t]$ does not have any isolated vertex (total property). The cardinality of the minimum total dominating set is called the total domination number. We denote the total domination number of the graph G as $\gamma_t(G)$.

The Roman dominating set (RDS) is an ordered partition of $V(G)$, say (V_0, V_1, V_2) induced by a function $f : V \to \{0, 1, 2\}$ called Roman dominating function (RDF) such that (i) $V_i = \{v \in V(G) : f(v) = i\}$, for $i = 0, 1, 2$, and (ii) for each $v \in V_0$, there exists at least a vertex $u \in V_2$ such that $uv \in E(G)$. The RDF with minimum weight, $f(V) = \sum_{u \in V(G)} f(u)$, is called the Roman domination number, and it is denoted by $\gamma_R(G)$. The total Roman dominating set (TRDS) is an ordered partition of V, say (V_0, V_1, V_2) induced by a function, $f : V \to \{0, 1, 2\}$ called total Roman dominating function (TRDF) such that (i) f is a Roman dominating function (Roman property), and (ii) the induced subgraph $G[V_1 \cup V_2]$ does not contain any isolated vertex (total property). The TRDF with minimum weight, $f(V) = \sum_{u \in V(G)} f(u)$, is called the Roman domination number. We define the total Roman domination number $\gamma_{tR}(G)$ as the minimum weight among all TRDFs on G. We often refer to the weight of a vertex as the Roman value of the vertex.

A unit disk graph (UDG) is an intersection graph formed from a collection of unit disks on the Euclidean plane. Let $D = \{D_1, D_2, \ldots, D_n\}$ be a set of unit disks and $C = \{c_1, c_2, \ldots, c_n\}$ be the set of corresponding centers, where c_i is the center of the disk D_i. A graph $G = (V, E)$ is said to be a geometric UDG if a one-to-one correspondence exists between each $v_i \in V(G)$ with $c_i \in C$ and an edge $v_i v_j \in E(G)$ if and only if $d(c_i, c_j) \leq 1$, where $d(.,.)$ is the Euclidean distance between two points in $\mathbb{R}^2$.

An ad-hoc (or IoT) network consists of heterogeneous nodes, where each node is associated with an inherent cost. The efficacy of the network is enhanced by installing

[1] *In this article, we refer $V(G)$ and V interchangeably.*

additional security features to the node in terms of hardware and/or software, often called monitoring devices, such as cameras, sensors, intrusion detection systems (IDSs), etc. Nodes with such devices are called monitoring nodes. Monitoring node with added security features consumes additional operational costs, where the operational cost is directly proportional to the level of security provided by the node to the network. It may be noted that the nodes without any security devices have negligible operational costs. Let each node in the network is mapped to a security function $f \rightarrow \{0, 1, 2\}$ such that each node with security level 0 is directly surveillance by a node with security level 2 and a node with security level 1 surveillance itself only. In addition, each monitoring node actively exchanges "still alive" messages at regular intervals with other monitoring nodes in its neighborhood. Since each node acts within a specific range, the properties of these types of networks can be studied through a UDG. The RDS problem helps the network to locate the installation sites for the monitoring devices such that the incurred cost is minimized, and the inclusion of total property to the RDS problem ensures that at least one of the monitoring nodes remains informed about the faulty monitoring node(s) (if any) in the network.

1.1. Related Work

The domination problem and its variations are studied extensively in the literature [11–13]. In 1990, Clark et al. [6] showed that the minimum dominating set problem is NP-complete when the graph is restricted to UDGs. Subsequently, Marathe et al. [17] gave a 5-factor approximation algorithm for the dominating set problem in UDGs. Cockayne et al. [7] introduced the concept of the total dominating set and showed that $\gamma_t \leq \frac{2}{3}n$ for a connected graph with $n > 3$ vertices. The detailed literature on TDS can be found in [14, 15]. In 2004, Cockayne et al. [8] introduced a new variation on domination called Roman domination, which was motivated by article [24] and was based on legion deployment to enhance security with limited resources. Some more variations on Roman domination can be found in [3–5, 9, 20]. Shang et al. [22] introduced the concept of Roman dominating set (RDS) in UDGs and gave 5-factor and 7.5- factor approximation algorithms for Roman dominating set (RDS) problem and connected Roman dominating set (CRDS) problem, respectively, in UDGs. However, the latter was achieved using a distributed algorithm from [25]. One of the variations, called total Roman domination, was introduced in [16]. In [1], authors established lower and upper bounds on total Roman dominating set (TRDS) and related the total Roman domination number ($\gamma_{tR}(G)$) to domination parameters such as the domination number ($\gamma(G)$), Roman domination number ($\gamma_R(G)$) and total domination number ($\gamma_t(G)$). In [18], authors gave a new lower and upper bound for $\gamma_{tR}(G)$, which was even tighter than the well known bound, $2\gamma(G) \leq \gamma_{tR}(G) \leq 3\gamma(G)$. In 2020, A. Poureidi [19] gave a linear time algorithm to compute the total Roman domination number for proper interval graphs. For general graphs, more results and variations on TRDS can be found in [2, 10, 23].

The connected Roman dominating set (CRDS) problem is closely related to the TRDS problem since every CRDS is a TRDS. As seen in [22], the authors gave a 7.5-factor approximation algorithm for the CRDS problem in UDGs. However, the factor was achieved through a distributed algorithm that appeared in [25], which has an extra message passing overhead. We observe that the heuristic in [21] can be modified to obtain a 12-factor approximation algorithm for the TRDS problem in UDG. So any constant factor algorithm below 12 for the TRDS problem without message passing overhead is noteworthy.

Recently, in 2021, Jena and Das [21] showed that the TDS problem in UDGs is NP-complete and gave an 8-factor approximation algorithm which runs in $O(|V| \log |D_t|)$ time, where $|D_t|$ is the size of the output. In the same paper, the authors presented a polynomial time approximation scheme (PTAS) that runs in $O(k^2 n^{2(\lceil 2\sqrt{2}k \rceil)^2})$ time to compute a total dominating set of size at most $(1 + \frac{1}{k})^2 |D_t^*|$, where $k \geq 1$ and D_t^* is the minimum TDS. The scheme calculates a total dominating set of size at most $4|D_t^*|$ in $O(n^{18})$ time, which is quite high, and for better approximation, the time complexity even worsens. So, there is scope for improvements in the approximation factor and running time.

1.2. Our Contribution

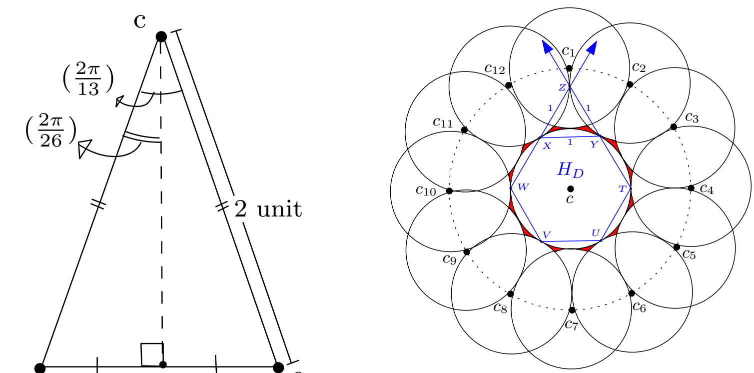
The remaining part of this paper is organized as follows. In Section 2, we introduce the required preliminaries. We also establish some new lemmas and observations pertinent to the article. In Section 3, we detail the approach for finding a 10.5-factor approximation algorithm for the TRDS problem in UDGs, which is an improvement over the previously best-known approximation factor 12 [21]. In Section 4, we propose a 7.79-factor approximation algorithm for the TDS problem in UDGs, which is an improvement over the best known 8-factor approximation algorithm [21]. Finally, we conclude the paper in Section 5.

2. Preliminaries

In this section, we define some notations and definitions that are pertinent to the article. We revisit some of the already-known facts and properties of the unit disk graphs (UDGs) and the total Roman dominating set (TRDS) in general graphs. Here, we also establish some lemmas that are relevant to the article.

UDG: Let $G = (V, E)$ be a unit disk graph with the vertex set V and edge set E, where $V = \{p_1, p_2, \ldots, p_n\} \subseteq \mathbb{R}^2$ is the set of disk centers and $p_i p_j \in E$ iff $d(p_i, p_j) \leq 1$. In this article, we often refer to a point as a vertex or node. Let $\Delta(p)$ denote the disk of radius 1 centered at the point $p \in V$ and $\Delta(P) = \{\Delta(p) : p \in P\}$. The set of disks $\Delta(P)$ is said to be independent if for any pair $p, q \in P$, $p \notin \Delta(q)$. In a graph G, we use the symbol $x \overset{G}{\rightsquigarrow} y$ to denote a path between x and y consisting of multiple edges, and we use multiple paths to denote a cycle. For an example, the symbol $x \overset{G}{\rightsquigarrow} y \overset{G}{\rightsquigarrow} x$ represents a cycle consisting of paths $x \overset{G}{\rightsquigarrow} y$ and $y \overset{G}{\rightsquigarrow} x$.

Lemma 1. *[21] Consider two points $p, q \in \mathbb{R}^2$ such that $d(p,q) \leq 1$. If S is the set of independent disks of radius 1 such that each disk in S contains the points p and/or q, then $|S| \leq 8$.*



(a) Measuring distance of $\overline{c_1c_2}$

(b) Illustration of independent unit disks inside a disk of radius 2

Figure 1. **Illustration of Lemma 2.**

Lemma 2. *Let D be a unit disk centered at c. If $S = \{c_1, c_2, \ldots, c_t\}$ is the set of centers of the independent set of disks $\Delta(S)$ such that $1 < d(c, c_i) \leq 2$ for $1 \leq i \leq t$, then $t = |S| \leq 18$.*

Proof. Let $\mathscr{D}$ and D be the disks of radius 2 and 1, respectively, centered at a single point c. We have to show that at most 18 independent unit disks exist whose centers lie on the disk $\mathscr{D}$ and are also independent from the disk D. We prove the result in two steps. In the first step, we show that there exists at most 12 independent unit disks' centers on the periphery of $\mathscr{D}$. Let $S_1 = \{c_1, c_2, \ldots, c_{12}\}$ be the set of corresponding disks' centers such that $d(c, c_i) = 2$ for $i = 1, 2, \ldots, 12$. In the second step, we show that if $d(c, c_i) = 2$ for $i = 1, 2, \ldots, 12$; then there exists at most 6 independent unit disks, say $S_2 = \{c'_1, c'_2, \ldots, c'_6\}$ such that $\Delta(S)$ is independent where $S = S_1 \cup S_2$ and $1 < d(c, c'_i) < 2$ for $i = 1, 2, \ldots, 6$.
(i) On contrary assume that $S_1 = \{c_1, c_2, \ldots, c_{13}\}$ is the set of 13 points such that $d(c, c_i) = 2$, for $1 \leq i \leq 13$. Let the points c_1, $c_2, \ldots$, c_{13} be placed sequentially in clockwise order on the periphery of the disk $\mathscr{D}$ (i.e., $d(c, c_i) = 2$ for $1 \leq i \leq 13$). Let $\overline{\mathrm{cc_i}}$ be the line segment joining the point c with c_i, where $1 \leq i \leq 13$. Then there exists at least one pair of consecutive segments $\overline{\mathrm{cc_j}}$ and $\overline{\mathrm{cc_k}}$ such that $\angle c_j c c_k \leq \frac{2\pi}{13}$. Without loss of generality, let the two segments be $\overline{\mathrm{cc_1}}$ and $\overline{\mathrm{cc_2}}$ such that $\angle c_1 c c_2 \leq \frac{2\pi}{13}$. We need to show that the points c_1 and c_2 are not independent. Let's consider the triangle $\Delta c c_1 c_2$ as shown in Figure 1(a). Let $\overline{\mathrm{cp}}$ be the perpendicular bisector of $\overline{\mathrm{c_1c_2}}$. Then $|\overline{\mathrm{c_1c_2}}| = 4\sin(\frac{2\pi}{13\times 2}) < 1$. This leads to the contradiction that c_1 and c_2 are independent. This proves that $|S_1| \leq 12$ (refer to Figure 1(b) for an orientation of 12

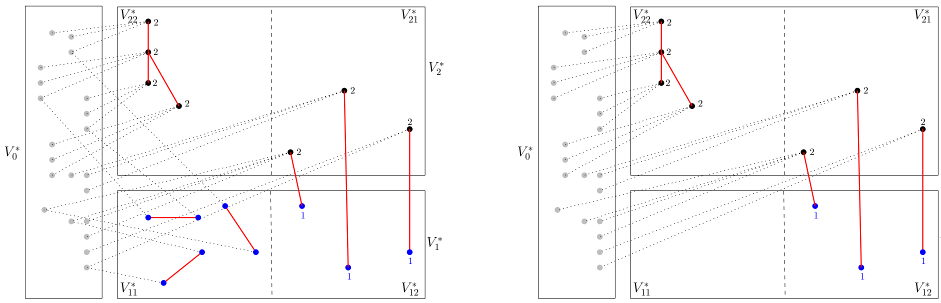
independent disks whose centers are placed on the periphery (dotted line) of the disk $\mathscr{D}$).
(ii) From the first step, it is proved that there can be at most 12 independent unit disks whose centers lie on the periphery of a disk of radius 2. Let $S_1 = \{c_1, c_2, \ldots, c_{12}\}$ be the set of disks' centers that lie on the periphery of $\mathscr{D}$ as shown in Figure 1(b). Let D be a unit disk centered at point c; then there exists a set of 12 regions, say $R = \mathscr{D} \setminus D \setminus \bigcup_{i=1}^{12} \Delta(c_i)$ (the 12 regions are shaded in Red colour in Figure 1(b), which are still independent from S_1. Now, we have to show that at most 6 independent unit disk centers can lie in these regions. Let H_D $(TUVWXY)$ be the hexagon inscribed in disk D as shown in Figure 1(b). Since D is a unit disk, each arm of the hexagon H_D is of unit length. Let us extend the line segments $\overline{WX}$ and $\overline{TY}$ such that they meet at point Z. Now, the ΔXYZ is an equilateral triangle with side length of 1 unit. Since each such triangle contains two red regions, no consecutive regions can contain the centers of two independent unit disks (since the maximum distance between any two points in an equilateral triangle of side length 1 is upper bounded by 1). Therefore, 12 regions can contain at most 6 independent unit disks. Let $S_2 = \{c'_1, c'_2, \ldots, c'_6\}$ be the independent unit disks on those 12 regions.
From the first step, there exists at most 12 independent unit disks at a distance 2 from the point c. If the independent disks become nearer to c, then the independency among the disk decreases along with the decrease in the area of the 12 regions. Hence, placing 12 disks' centers on the periphery ensures the maximum value of $|S| = |S_1 \cup S_2| \leq 18$. □

TRDS: For a given graph $G = (V, E)$, let $F^* = \{f_1^*, f_2^*, \ldots, f_m^*\}$ be the set of all TRDFs with $W(f_i^*) = \gamma_{tR}(G)$ for $1 \leqslant i \leqslant m$. Let $f^* \in F^*$ be a TRDF and let (V_0^*, V_1^*, V_2^*) be the ordered partition of V induced by f^* such that f^* attains minimum $|V_1^*|$. We partition the set V_1^* into subsets V_{11}^* and V_{12}^* such that V_{12}^* is the set of vertices in V_1^* which have neighbors in V_2^* and $V_{11}^* = V_1^* \setminus V_{12}^*$. Similarly, we partition V_2^* into subsets V_{22}^* and V_{21}^* such that V_{21}^* is the set of vertices in V_2^* which have neighbors in V_1^* and $V_{22}^* = V_2^* \setminus V_{21}^*$. In Figure 2(a) (respectively, Figure 2(b)), 3 rectangles enclosed within solid line represent the sets V_0^*, V_1^* and V_2^*. The dashed lines in Figure 2(a) (respectively, Figure 2(b)) segregate V_1^* into V_{11}^* and V_{12}^*, and V_2^* into V_{22}^* and V_{21}^*.

Lemma 3. *[1] If $f^* = (V_0^*, V_1^*, V_2^*)$ is an optimal TRDF on a graph $G = (V, E)$ such that f^* attains minimum $|V_1^*|$, then either (i) V_2^* is a dominating set (DS) of G, or (ii) $G[V_{11}^*] = \alpha K_2$ for some integer $\alpha \geq 1$, where K_2 represents the complete graph of two vertices.*

From Lemma 3, the following observations can be noted.

Observation 1. If $f^* = (V_0^*, V_1^*, V_2^*)$ is an optimal TRDF on the graph $G = (V, E)$ such that f^* attains minimum $|V_1^*|$, then for each edge $pq \in V_{11}^*$, $B(\{p, q\}) \subseteq V_0^*$ (refer to Figure 2(a)).



(a) V_2^* is not a DS of G (b) V_2^* is a DS of G

Figure 2. **Illustration of** $G[V_1^* \cup V_2^*]$.

Proof. It is sufficient to prove that there does not exist any vertex $v \in B(\{p,q\})$ such that $f(v) = 1$ or $f(v) = 2$. On the contrary, assume that there exists a vertex $v \in B(\{p,q\})$ with $f(v) = 1$. Let $vp \in E(G)$. If so, then we define another optimal TRDF $f^{*\prime} = (V_0^{*\prime}, V_1^{*\prime}, V_2^{*\prime})$, where $V_0^{*\prime} = V_0^* \cup \{v\}$, $V_1^{*\prime} = V_1^* \setminus \{v,p\}$ and $V_2^{*\prime} = V_2^* \cup \{p\}$ such that $W(f^{*\prime}) = W(f^*)$ and $|V_1^{*\prime}| < |V_1^*|$. This leads to a contradiction that $|V_1^*|$ is the minimum.
To prove the second part, assume that there exists a vertex $v \in B(\{p,q\})$ with $f(v) = 2$. Let $vp \in E(G)$. If so, then we define another optimal TRDF $f^{*\prime} = (V_0^{*\prime}, V_1^{*\prime}, V_2^{*\prime})$, where $V_0^{*\prime} = V_0^* \cup \{q\}$, $V_1^{*\prime} = V_1^* \setminus \{p,q\}$ and $V_2^{*\prime} = V_2^* \cup \{p\}$ such that $W(f^{*\prime}) = W(f^*)$ and $|V_1^{*\prime}| < |V_1^*|$. This leads to a contradiction that $|V_1^*|$ is the minimum. □

Observation 2. Let $f^* = (V_0^*, V_1^*, V_2^*)$ is an optimal TRDF on a graph $G = (V,E)$ such that f^* attains minimum $|V_1^*|$. If V_2^* is a DS of $G = (V,E)$, then $G[V_1^* \cup V_2^*]$ does not contain an edge with Roman value $(1,1)$, i.e., $V_{11}^* = \emptyset$ (refer to Figure 2(b)).

Proof. On the contrary, assume $V_{11}^* \neq \emptyset$, i.e., there exists an edge $pq \in E(G[V_{11}^*])$ (as f^* satisfies the total property). Since V_2^* is a dominating set of G, each vertex $v \in V(G)$ is either in V_2^* or there exists an edge $uv \in E(G)$ such that $u \in V_2^*$. Since the sets V_2^* and V_{11}^* are mutually exclusive, for each vertex $u \in V_{11}^*$, there exists a vertex $v \in V_2^*$ such that $uv \in E[G]$. Let s and t dominate p and q, respectively, where $s,t \in V_2^*$. If $s = t$ (single vertex which dominates p and q), then we define another TRDF $f^{*\prime} = (V_0^{*\prime}, V_1^{*\prime}, V_2^{*\prime})$, where $V_0^{*\prime} = V_0^* \cup \{q\}$, $V_1^{*\prime} = V_1^* \setminus \{q\}$ and $V_2^{*\prime} = V_2^*$ such that $W(f^{*\prime}) < W(f^*)$. This leads to a contradiction that f^* is optimal; otherwise (i.e., $s \neq t$), p and q would have been in V_{12}^* not in V_{11}^*. □

Observation 3. Let $f^* = (V_0^*, V_1^*, V_2^*)$ is an optimal TRDF on a graph $G = (V,E)$ such that f^* attains minimum $|V_1^*|$. If V_2^* is not a DS of G, then $G[V_1^* \cup V_2^*]$ contains edges with Roman values $(2,2)$, $(2,1)$ and $(1,1)$.

Proof. It follows from Observation 1 and Observation 2. □

From Observation 2 and Observation 3, the induced graph $G[V_1^* \cup V_2^*]$ exhibits the following *five* structures only: *(i)* $2-2$: An edge with Roman value 2 for each vertex in the edge, *(ii)* Bouquet of $2-2$: A connected component with Roman value 2 for each vertex in the component, *(iii)* $2-1$: An edge with Roman value 2 for one vertex and Roman value 1 for another vertex, *(iv)* Flower of $2-1$: A connected component with a star structure, where the center vertex carries Roman value 1 and the remaining carries Roman value 2, *(v)* $1-1$: An edge with Roman value 1 for each vertex in the edge. For reference, see Figure 3 for the complete illustration of each possible type of structure in the induced graph $G[V_1^* \cup V_2^*]$.

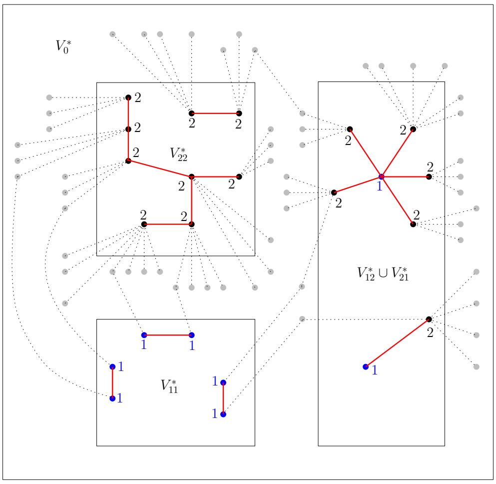


Figure 3. **Illustration of $G[V_{22}^*]$, $G[V_{12}^* \cup V_{21}^*]$ and $G[V_{11}^*]$.**

3. A *10.5* factor Approximation Algorithm

In this section, we propose a 10.5 - factor approximation algorithm called *TRDF-UDG* for the TRDS problem in geometric UDGs. The algorithm runs on a graph with no isolated vertex. If the graph is disconnected, each component can run it to obtain the TRDF.

3.1. TRDF-UDG: The Proposed Algorithm

Given a unit disk graph $G = (V, E)$, where $V = \{p_1, p_2, \ldots, p_n\} \subseteq \mathbb{R}^2$ is the set of disk centers, the algorithm finds a TRDF $f = (V_0, V_1, V_2)$ of G. First, it finds a maximal independent set $V_2 \subseteq V$ of G to satisfy the Roman property. Next, to satisfy the total property, it chooses a set of neighboring vertices $V_1 \subseteq V \setminus V_2$ such that for each $u \in V_2$, there exists a vertex $u' \in V_1$ and $u' \in \Delta(u)$. See Algorithm 1 (*TRDF-UDG*) for the pseudocode, Lemma 4 for the correctness and Lemma 6 for time complexity analysis/implementation details of the algorithm.

Algorithm 1 TRDF-UDG

Require: A unit disk graph, $G = (V, E)$ with known disk centers
Ensure: A TRDF $f = (V_0, V_1, V_2)$ and the corresponding weight $W(f)$
1: $V_0 = \emptyset$, $V_1 = \emptyset$, $V_2 = \emptyset$, $V' = V$
2: **while** $V' \neq \emptyset$ **do** ▷ Roman property of TRDF
3: choose a vertex $v \in V'$
4: $V_2 = V_2 \cup \{v\}$ and $f(v) = 2$
5: $V' = V' \setminus N_G[v]$
6: **end while**
7: **for** each $u \in V_2$ **do** ▷ total property of TRDF
8: choose a vertex $u' \in N_G(u)$
9: $V_1 = V_1 \cup \{u'\}$ and $f(u') = 1$
10: **end for**
11: $V_0 = V \setminus (V_1 \cup V_2)$
12: **for** each $u \in V_0$ **do**
13: $f(u) = 0$
14: **end for**
15: **return** $f = (V_0, V_1, V_2)$ and $W(f) = 2 \times |V_2| + |V_1|$

Lemma 4. *The function $f = (V_0, V_1, V_2)$ in TRDF-UDG is a TRDF of G.*

Proof. *TRDF-UDG* runs in two phases. In the first phase, it finds a maximal independent set V_2 of G (since every maximal independent set is a dominating set) and then assigns Roman value 2 to each vertex in V_2 (see lines 2-6 of Algorithm 1), which ensures the *Roman property* of TRDF. To ensure the *total property* of TRDF, it finds another set V_1 by adding a neighbor vertex for each vertex in V_2 and assigns Roman value 1 (see lines 7-10 of Algorithm 1). The remaining vertices carry Roman value 0 (see lines 12-14 of Algorithm 1). Therefore, combinedly the nominated points in V_2 and V_1 satisfies the *Roman* and *total* properties. Hence, the function $f = (V_0, V_1, V_2)$ is a TRDF of G. □

Lemma 5. *A cell of size 1×1 contains the centers of at most 3 independent unit disks.*

Proof. Since the perimeter of a cell is 4 unit, the number of independent unit disks that a cell can contain is at most 3; otherwise, the disks are no longer independent. □

Lemma 6. *TRDF-UDG runs in $O(|V| \log k)$ time, where $k = |V_2|$.*

Proof. Let $V = \{p_1, p_2, \ldots, p_n\}$ be the set of disks' centers corresponding to graph $G = (V, E)$. Let $\mathbb{R}$ be a rectangular plane containing the set of points V, where the extreme left and extreme bottom arms of the rectangle represent the x- and y-axis, respectively. Split the plane $\mathbb{R}$ into horizontal strips and then vertical strips of width one unit each, resulting in a grid of cell size 1×1. Let index each cell as $[x, y]$, where

$x, y \in \mathbb{N} \cup \{0\}$. If any point $p \in V$ is located at co-ordinate (p_x, p_y) in the given plane, then it belongs to a cell with index $[\lfloor p_x \rfloor, \lfloor p_y \rfloor]$.
In phase *one*, *TRDF-UDG* constructs a maximal independent dominating set V_2 of the input graph G. To do so efficiently, each non-empty cell maintains a list that keeps the points of V that are chosen for inclusion in V_2, and they are located within that cell. While considering a point $p \in V$ as a candidate for the set V_2, it only probes into 9 cells. That means if p is located at co-ordinate (p_x, p_y), then it searches in each $[i, j]$ cell, where $\lfloor p_x \rfloor - 1 \leqslant i \leqslant \lfloor p_x \rfloor + 1$ and $\lfloor p_y \rfloor - 1 \leqslant j \leqslant \lfloor p_y \rfloor + 1$.[2] If there does not exist any point $q \in V_2$ in those 9 cells such that $p \in \Delta(q)$, then p is included in V_2. A height balance binary tree containing non-empty cells is used to store the points that are in V_2. Since each cell of size 1×1 can contain at most 3 independent unit disks (see Lemma 5), the processing time to decide whether a point is in V_2 or not requires $O(\log k)$ time. Thus the time taken to process $|V|$ points is $O(|V| \log k)$, where $k = |V_2|$ (see lines 2-6 of Algorithm 1). In phase *two*, it finds a neighboring vertex u' for each $u \in V_2$ and assigns Roman value 1. Let the set of neighboring vertices be V_1. Since, $|V_1| = |V_2| = k$, the time taken in phase *two* is $O(k)$. (see lines 7-10 of Algorithm 1). Then it assigns Roman value 0 to the remaining vertices of V (excluding V_1 and V_2) in $O(|V|)$ time (see lines 12-14 of Algorithm 1). So in total, *TRDF-UDG* (Algorithm 1) runs in $O(|V| \log k)$ time. □

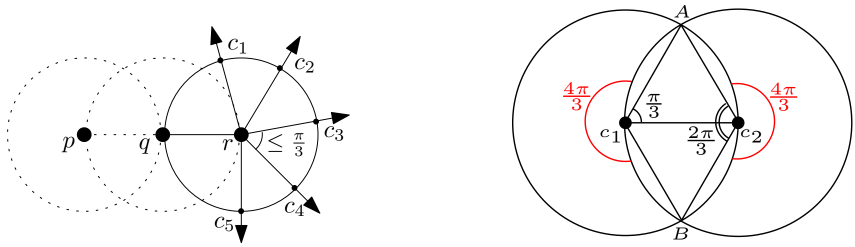
Lemma 7. *Consider two points $p, q \in \mathbb{R}^2$ such that $d(p, q) \leq 1$. If there exists a point $r \in \mathbb{R}^2$ such that either $d(r, p) \leq 1$ and/or $d(r, q) \leq 1$ and S' is the set of independent unit disks that contains at least one element from $\{p, q, r\}$, then $|S'| \leq |S| + 4$, where S is the set of independent unit disks containing p and/or q.*

Proof. Without loss of generality, assume that $d(q, r) \leq 1$. Suppose there exists 5 independent unit disks that contains r but neither p nor q, i.e., $|S'| = |S| + 5$. Let c_1, c_2, c_3, c_4 and c_5 are the centers of those 5 unit disks. Let r_i denotes the ray $\overrightarrow{rc_i}$ as depicted in Figure 4(a), where $1 \leq i \leq 5$. Without loss of generality, let q lie between the centers c_1 and c_5. Since, $\angle c_1 rq + \angle c_5 rq > \frac{2\pi}{3}$, so at least one of $\angle c_1 r c_2$, $\angle c_2 r c_3$, $\angle c_3 r c_4$ and $\angle c_4 r c_5$ is less than $\frac{\pi}{3}$. This leads to a contradiction that the disks at centers c_1, c_2, c_3, c_4, and c_5 are independent. Thus $|S'| \leq |S| + 4$. □

Lemma 8. *Consider two points $p, q \in \mathbb{R}^2$ such that $d(p, q) \leq 1$. Let S be the set of independent disks of radius 1 such that S contains the points p and/or q. If $|S| > 7$, then S contains at least one disk having center in $\Delta(p) \cap \Delta(q)$.*

Proof. It is sufficient to prove that if $|S| = 8$, then S contains a disk having its center in $\Delta(p) \cap \Delta(q)$. Let $c_1, c_2, \ldots, c_8$ are the disk centers in S. On the contrary assume that

[2] *Any point outside these 9 cells is independent from p*



(a) Illustration of independent disks orientation of a pendant vertex

(b) Illustration of boundary length of $(\Delta(p) \setminus \Delta(q)) \cup (\Delta(q) \setminus \Delta(p))$, when $d(c_1, c_2) = 1$

Figure 4. **Illustration of Lemma 7 and Lemma 8**

$c_i \notin \Delta(p) \cap \Delta(q)$ for $i = 1, 2, \ldots, 8$. Since $\Delta(c_1), \Delta(c_2), \ldots, \Delta(c_8)$ are independent, so without loss of generality, we assume that $c_1, c_2, \ldots, c_8$ lie on the boundary of $(\Delta(p)\setminus\Delta(q))\cup(\Delta(q)\setminus\Delta(p))$. The length of the boundary in $(\Delta(p)\setminus\Delta(q))\cup(\Delta(q)\setminus\Delta(p))$ is at most $8\pi/3$ as $d(p, q) \leq 1$ (refer to Figure 4(b)). Since $\Delta(c_1), \Delta(c_2), \ldots, \Delta(c_8)$ are independent, then mutual distance among $c_1, c_2, \ldots, c_8$ must be more than $\pi/3$, which leads to a contradiction. □

Let $f^* = (V_0^*, V_1^*, V_2^*)$ be an optimal TRDF with minimum $|V_1^*|$ and $f = (V_0, V_1, V_2)$ be the solution returned by *TRDF-UDG* (Algorithm 1). By Observation 2 and Observation 3, each edge $pq \in E(G[V_1^* \cup V_2^*])$ has Roman value either $(2, 2)$ or $(2, 1)$ or $(1, 1)$. Let $(V_{11}^*, V_{12}^* \cup V_{21}^*, V_{22}^*)$ be a partition of the set $V_1^* \cup V_2^*$ such that V_{11}^*, V_{12}^*, V_{21}^* and V_{22}^* are the sets as defined earlier in Section 2. The inner 3 rectangles in Figure 3 represent the sets V_{11}^*, $V_{12}^* \cup V_{21}^*$ and V_{22}^*, and the red solid lines in each rectangle illustrate the corresponding induced graph.[3]

Lemma 9. *If $f^* \in F^*$ is an optimal TRDF, which attains minimum $|V_1^*|$, then TRDF-UDG charges at most weight 24 against an edge $pq \in E(G[V_{22}^*])$.*

Proof. Let $f = (V_0, V_1, V_2)$ be the TRDF returned by *TRDF-UDG* with weight $W(f)$. Consider an arbitrary edge $pq \in E(G[V_{22}^*])$. From Lemma 1, there are at most 8 independent unit disks in V_2 which contain the points p and/or q. To satisfy the Roman property of TRDF, *TRDF-UDG* assigns Roman value 2 (see line 4 in Algorithm 1) to each of the 8 vertices, which are the elements of V_2. So, our algorithm may invest $2 \times 8 = 16$ from $W(f)$ to dominate (Roman) both p and q. Next, to satisfy the total property, *TRDF-UDG* chooses a neighbor for each selected vertex in phase *one* from V_1 and assigns Roman value 1 (see line 9 in Algorithm 1), which requires an additional 8 investment from $W(f)$. So in total, *TRDF-UDG* invests at most

[3] *In Figure 3, the grey colored (light shaded) disks represent the vertices in V_0^* and the dotted lines represent the edges between the set V_0^* and $V_1^* \cup V_2^*$*

24 from $W(f)$ weight. So for any edge $pq \in E(G[V^*_{22}])$ having Roman value $(2,2)$, $f = (V_0, V_1, V_2)$ dominates (total Roman) p and q by investing at most 24 weight from $W(f)$. □

Lemma 10. *If $f^* \in F^*$ is an optimal TRDF which attains minimum $|V_1^*|$, then TRDF-UDG charges at most weight 24 against an edge $(p,q) \in E(G[V^*_{12} \cup V^*_{21}])$.*

Proof. Proof of the lemma is similar to Lemma 9. □

Lemma 11. *Let $f^* = (V_0^*, V_1^*, V_2^*)$ is an optimal TRDF, which attains minimum $|V_1^*|$. If S is the set of independent unit disks of radius 1 such that S contains p and/or q, where $pq \in E(G[V^*_{11}])$, then $|S| \leq 7$.*

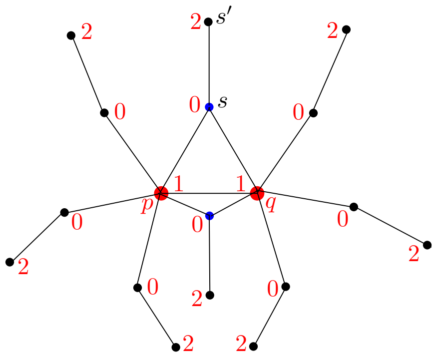


Figure 5. **Illustration of Lemma 11.**

Proof. Let $|S| \geq 8$. So, by Lemma 8, there exists at least one disk in S having its center in $\Delta(p) \cap \Delta(q)$. Let the disk's center be s (see Figure 5). From Observation 1, $f^*(s) = 0$. This implies there exist a vertex $s' \in N_G(s)$ such that $f^*(s') = 2$. Now, we define another optimal TRDF $f^{*\prime} = (V_0^{*\prime}, V_1^{*\prime}, V_2^{*\prime})$, where $V_0^{*\prime} = V_0^* \setminus \{s\} \cup \{p,q\}$, $V_1^{*\prime} = V_1^* \setminus \{p,q\}$ and $V_2^{*\prime} = V_2^* \cup \{s\}$ such that $W(f^{*\prime}) = W(f^*)$ and $|V_1^{*\prime}| < |V_1^*|$. This leads to a contradiction that $|V_1^*|$ is the minimum. □

Lemma 12. *If $f^* \in F^*$ is an optimal TRDF which attains minimum $|V_1^*|$, then TRDF-UDG charges at most 21 against an edge $pq \in E(G[V^*_{11}])$.*

Proof. Let $f = (V_0, V_1, V_2)$ be the TRDF returned by *TRDF-UDG* with weight $W(f)$. Consider an arbitrary edge $pq \in E(G[V^*_{11}])$. From Lemma 11, there are at most 7 independent disks which contain p and/or q. So in the worst case, at most 7 independent disks in V_2 contain the point p and/or q. The Roman value assigned to each vertex in V_2 is 2 (see line 4 in Algorithm 1). So, *TRDF-UDG* may invest $2 \times 7 = 14$ from $W(f)$ to dominate (Roman) both p and q. Now, to satisfy the total property, *TRDF-UDG* further selects a neighbor for each $v \in V_2$ from V_1 and assigns Roman value 1 (see line 9 in Algorithm 1), which requires additional 7 investment from $W(f)$. So in total, *TRDF-UDG* invests at most 21 from $W(f)$ weight. So for any edge $pq \in E(G[V_{11}^*])$ having Roman value $(1,1)$, $f = (V_0, V_1, V_2)$ dominates (total Roman) p and q by investing at most 21 weight from $W(f)$. □

Lemma 13. *If A ($|A| = \ell$) is the set of vertices in the bouquet/flower (mentioned in Section 2) and S is the set of independent unit disks that contains A, then $|S| \leq 4\ell$.*

Proof. Consider an edge pq in the bouquet/flower. Then there exists at most 8 independent unit disks that can contain p and/or q (by Lemma 1). Now consider another vertex r attached either to p or q, then at most, 4 independent unit disks can contain only r (by Lemma 7). So, at most, $8 + 4 = 12$ independent disks can contain p and/or q and/or r. Let $A' \subseteq A$ be a set of vertices in the bouquet/flower apart from p and q, i.e., $A' = A \setminus \{p, q\}$. Note that $|A'| = \ell - 2$. So, by Lemma 7, there exists at most 4 independent disks for each vertex in A', thus at most $8 + 4(\ell - 2) = 4\ell$ independent disks can contain all the vertices in A. □

Let C^* be a connected component in $G[V_1^* \cup V_2^*]$. For any component C^*, we denote $W(C^*)$ and $W^*(C^*)$ as the weights incurred by *TRDF-UDG* and f^*, respectively.

Lemma 14. *If $f^* = (V_0^*, V_1^*, V_2^*)$ is an optimal TRDF such that $|V_1^*|$ attains minimum value and $f = (V_0, V_1, V_2)$ is the TRDF returned by TRDF-UDG , then for each component $C^* \in G[V_1^* \cup V_2^*]$, $W(C^*) \leq 10.5 \times W^*(C^*)$.*

Proof. Let $f^* = (V_0^*, V_1^*, V_2^*)$ be an optimal TRDF which attains minimum $|V_1^*|$. Consider the ordered partition $(V_{11}^*, V_{12}^* \cup V_{21}^*, V_{22}^*)$ of the set $V_1^* \cup V_2^*$ as defined in Section 2. From Section 2, the induced graph $G[V_1^* \cup V_2^*]$ exhibits *five* different structures only (refer to Figure 3). To make the analysis easier, we consider each of the induced graphs $G[V_{22}^*]$, $G[V_{12}^* \cup V_{21}^*]$ and $G[V_{11}^*]$ separately.

Case 1. $G[V_{22}^*]$.
By Lemma 13, there are 4ℓ independent disks that can contain vertices of a bouquet. So, in the first phase, *TRDF-UDG* can pick at most 4ℓ independent disks. *TRDF-UDG* invests $3 \times 4\ell$ weight from $W(f)$. But in the optimal, 2ℓ weight out of $W(f^*)$ is required for the bouquet (since each vertex carries Roman value 2). So, *TRDF-UDG* invests $\frac{12\ell}{2\ell} = 6$ times more if compared to the optimal solution f^*.

Case 2. $G[V_{12}^* \cup V_{21}^*]$.
By Lemma 13, There are 4ℓ independent disks that can contain vertices of a flower. So, in the first phase, *TRDF-UDG* can pick at most 4ℓ independent disks. *TRDF-UDG* invests $3 \times 4\ell$ weight from $W(f)$. But in the optimal, $3 + 2(\ell - 2) = 2\ell - 1$ weight out of $W(f^*)$ is required for the flower. Note that $\frac{12\ell}{2\ell-1}$ is maximum when the flower consists of only $\ell = 2$ vertices, and in f^*, the Roman value assigned to the two vertices are 2 and 1. The more is the number of vertices in a flower, the less is the

value of $\frac{12\ell}{2\ell-1}$ (since the function $\frac{12\ell}{2\ell-1}$ is a monotonically decreasing function). So, at maximum, *TRDF-UDG* invests 8 times more if compared to the optimal solution f^*.

Case 3. $G[V_{11}^*]$.

From Lemma 3, each component $C^* \in G[V_{11}^*]$ is a K_2. *TRDF-UDG* invests at most 21 against C^* by Lemma 12. Hence, *TRDF-UDG* invests $\frac{21}{2} = 10.5$ times more for $C^* \in G[V_{11}^*]$ in f compared to f^*. □

Lemma 15. *$W(f) \leq 10.5 \times \gamma_{tR}(G)$, where $W(f)$ and $\gamma_{tR}(G)$ are the weights associated with f (in Algorithm 1) and f^* (optimal TRDF) of G, respectively.*

Proof. Since $V_{11}^* \cup (V_{12}^* \cup V_{21}^*) \cup V_{22}^* = V_1^* \cup V_2^*$, therefore, the theorem follows from Lemma 14. □

Theorem 4. *The proposed algorithm (TRDF-UDG) gives a 10.5-factor approximation algorithm for the TRDS problem in UDGs, which runs in $O(|V| \log k)$ time, where k is the number of vertices with Roman value 2.*

Proof. The approximation factor follows from Lemma 15, and the time complexity result follows from Lemma 6. □

4. Total Dominating Set (TDS) in UDGs Revisited

In this section, we propose an algorithm called *TDS-UDG* (see Algorithm 2 for the pseudocode) for the TDS problem in geometric UDGs with an approximation factor 7.79. The algorithm runs on a graph with no isolated vertex. If the graph is disconnected, each component can run it to obtain the TDS.

4.1. *TDS-UDG*: The Proposed Algorithm

We briefly describe the algorithm for finding a total dominating set (TDS) D_t of the given geometric UDG G. *TDS-UDG* consists of three phases. The first phase segregates the set of vertices V into $k+1$ subsets, i.e., $S = \{S_i : 0 \leq i \leq k\}$ such that $V = \bigcup_{i=0}^{k} S_i$. The second phase selects a set of vertices I from S_i (where $0 \leq i \leq k$) such that I satisfies the *domination property* of graph G. At the end, the third phase identifies another set of vertices $T \subseteq V(G)$ that meets the *total property*. Finally, *TDS-UDG* reports the *total dominating set* D_t as the collection of the two sets I and T, i.e., $D_t = I \cup T$. Now, we discuss each of the phase of the algorithm elaborately.

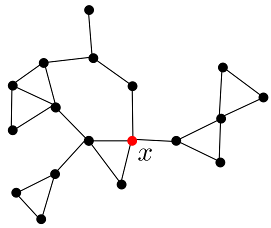
Algorithm 2 TDS-UDG

Require: A unit disk graph $G = (V, E)$ with known disk centers
Ensure: A TDS D_t

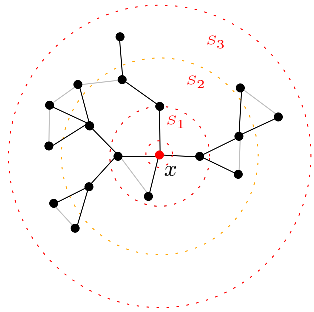
```
1: Choose a vertex x ∈ V(G)
2: x.explored = true, x.parent = NIL
3: for each vertex u ∈ V \ {x} do
4:     u.explored = false, u.parent = NIL
5: end for
6: I = ∅, T = ∅, D_t = ∅, k = 0, count = 0, S_k = {x}, I_k = {x}
7: while count ≠ |V(G)| do
8:     S_{k+1} = ∅
9:     for each node u ∈ S_k do
10:        for each node v ∈ N_G(u) do
11:            if v.explored = false then
12:                v.explored = true, S_{k+1} = S_{k+1} ∪ {v}
13:                v.parent = u
14:            end if
15:        end for
16:        count = count + 1
17:    end for
18:    k = k + 1
19: end while
20: k = k − 1
21: if k = 1 then
22:    choose any vertex y ∈ N(x)
23:    T = T ∪ {y}
24: else
25:    for i = 2 to k do
26:        I_i = ∅, T_i = ∅
27:        while S_i ≠ ∅ do
28:            Choose a vertex p ∈ S_i
29:            I_i = I_i ∪ {p}
30:            T_i = T_i ∪ {p.parent}
31:            S_i = S_i \ N_G[p], S_{i+1} = S_{i+1} \ N_G[p]
32:        end while
33:        I = I ∪ I_i
34:        T = T ∪ T_i
35:    end for
36: end if
37: D_t = I ∪ T
38: return D_t
```

4.1.1. Phase I

In phase I, *TDS-UDG* segregates the vertex set V into $k + 1$ subsets namely S_0, $S_1, \ldots, S_k$. The subsets are formed by constructing a breadth first search (BFS) tree rooted at any vertex x. Let T_x be the tree constructed at x for G (see Figure 6(a) and the corresponding rooted tree T_x of G in Figure 6(b)). If i is the level of the vertex y in T_x, then $y \in S_i$, where x is at level 0, and the level of a vertex in T_x is the number of edges in the path (in T_x) that connects to the root. Hence, S_i can be interpreted as the collection of all the vertices that are at i^{th} level in T_x (see lines 1-19 of Algorithm 2).



(a) $G = (V, E)$



(b) Rooted BFS tree T_x at x of G

Figure 6. **Illustration of S_i in G.**

Observation 5. Every vertex $v \in S_i$ has a neighbor in S_{i-1} for $i = 1, 2, \ldots k$.

Proof. In T_x, i represents the level of a node and S_i is the collection of nodes with level i. Since T_x is connected, every vertex $v \in S_i$ has a neighbor in S_{i-1}. □

Observation 6. If T_x is the rooted BFS tree constructed at vertex x of G, then for any vertex $u \in S_i$, there does not exist any vertex $v \in S_i$ such that $uv \in E(T_x)$, where $0 \leq i \leq k$.

Proof. On the contrary assume that there exists two vertices $u, v \in S_i$ such that $uv \in E(T_x)$. Since the vertices u and v are in S_i (i.e., at level i in T_x), there exists paths $u \overset{T_x}{\rightsquigarrow} x$ and $v \overset{T_x}{\rightsquigarrow} x$ in T_x. Now, if $uv \in E(T_x)$, then the edge uv forms a cycle $x \overset{T_x}{\rightsquigarrow} u \overset{T_x}{\rightsquigarrow} v \overset{T_x}{\rightsquigarrow} x$ in T_x. This leads to a contradiction that T_x is a tree. □

4.1.2. Phase II

In phase II, *TDS-UDG* finds a set of independent vertices I_i for each S_i such that $I = \bigcup_{i=0}^{k} I_i$ is a maximal independent set (MIS) of G. Initially, I_0 contains x as S_0 contains the only vertex x (see line 6 in Algorithm 2). Since each vertex in S_1 is a neighbor of x, $I_1 = \emptyset$. For the remaining S_i for $2 \leq i \leq k$, the set I_i is constructed on an incremental basis, i.e., I_i is constructed only after constructing I_{i-1}. For each S_i, the algorithm selects any random vertex $p \in S_i$ as a candidate for I_i. Then, it deletes the closed neighborhood of p in G (i.e., $N_G[p]$) from S_i and S_{i+1} (see line 31 in Algorithm 2). The neighborhood deletion from S_i and S_{i+1} ensures that the next candidate for I_i remains independent from the vertices already in I_i and the set $I_i \cup I_{i+1}$ remains an independent set. This process continues for the remaining vertices in S_i till S_i exhausts. Once the i value reaches $k + 1$, the set I becomes an MIS of G (see line 33 in Algorithm 2). Since every MIS of a graph is a dominating set of the graph, the MIS I is a dominating set of G. Thus I satisfies the *domination property*.

4.1.3. Phase III

In phase III, *TDS-UDG* finds a set T that satisfies the *total property* when taken with the independent set I. T is the collection of all T_i. Each T_i is a collection of the parent node for each node selected in I_i (see line 30 in Algorithm 2). Since line 13 keeps track of the parent vertex for each node present in S_i, the algorithm updates the set T by adding the parent node for each node selected in I (see lines 29-30 in Algorithm 2). The set I along with T ensures that none of the vertices in $G[I \cup T]$ is isolated.

Lemma 16. *Algorithm 2 returns (D_t) a TDS of the geometric UDG G.*

Proof. Since $D_t = I \cup T$ and from Phase I, II, and III, it is clear that the set D_t satisfies both domination and total properties. Hence the lemma follows. □

Lemma 17. *Algorithm 2 runs in $O(|V| + |E|)$ time.*

Proof. The complexity of *TDS-UDG* is primarily dominated by the *nested loop* used for segregating the set V into k subsets (see lines 7-19 in Algorithm 2). In the worst case, the algorithm checks each vertex and the corresponding adjacency list for segregation. Hence, the time complexity of *TDS-UDG* is $O(|V| + |E|)$. □

Lemma 18. *In Algorithm 2, if $|I| \geq 2$, then for each vertex $u \in I$, there exists at least one vertex $v \in I$, such that $d(u, v) \leq 2$.*

Proof. Since Algorithm 2 constructs an MIS of G on an incremental basis, i.e., it constructs the MIS I_i of S_i only after constructing the MIS I_{i-1} of S_{i-1}, where $I = \bigcup_{i=0}^{k} I_i$. So the lemma can be proved using induction on k with respect to the rooted tree T_x at x, i.e., the algorithm finds the MIS of the induced subgraph $G[S_0 \cup S_1 \cup \cdots \cup S_i]$ from $G[S_0 \cup S_1 \cup \cdots \cup S_{i-1}]$ to find an MIS of G since $G = G[S_0 \cup S_1 \cup \cdots \cup S_k]$. Let $P(k)$ be the proposition that for each vertex $u \in \bigcup_{i=0}^{k} I_i$, there exists at least one vertex $v \in \bigcup_{i=0}^{k} I_i$ such that $d(u, v) \leq 2$ for $k \geq 2$.

Basis step: We have to show that $P(2)$ is true, i.e., when $k = 2$, for each $u \in \bigcup_{i=0}^{2} I_i$, there exists at least one vertex $v \in \bigcup_{i=0}^{2} I_i$ such that $d(u, v) \leq 2$. Since, $I_0 = \{x\}$ and $I_1 = \emptyset$ ($S_1 = \emptyset$, since *TDS-UDG* removes $N_G(x)$ from S_1). Let I_2 be the MIS of S_2. Since the graph is connected, each vertex $v \in I_2$ has at least one vertex w in S_1 (see Observation 5). This implies $d(v, w) \leq 1$. Since every vertex present in S_1 is a neighbor of x, $w \in N_G(x)$. So $d(x, w) \leq 1$. From the triangle inequality, we have $d(v, x) \leq d(v, w) + d(w, x) \leq 2$. Hence $P(2)$ is true.

Inductive Hypothesis: When $k = m$, $P(m)$ is true, i.e., for each $u \in \bigcup_{i=0}^{m} I_i$, there exists at least one vertex $v \in \bigcup_{i=0}^{m} I_i$ such that $d(u, v) \leq 2$.
Inductive Step: We have to show that when $k = m + 1$, $P(m + 1)$ is true, i.e., for each $u \in \bigcup_{i=0}^{m+1} I_i$, there exists at least one vertex $v \in \bigcup_{i=0}^{m+1} I_i$ such that $d(u, v) \leq 2$. Since $P(m)$ is true, the following is sufficient to prove the lemma: for each vertex $u \in I_{m+1}$; there exists at least one vertex $v \in I_{m-1}$ such that $d(u, v) \leq 2$.
Without loss of generality, let us consider a vertex $u \in I_{m+1}$. Then the vertex u has one neighbor $w \in S_m$ (see Observation 5). Since $uw \in E$, $d(u, w) \leq 1$. $u \in I_{m+1}$ implies $u \in S_{m+1}$. u is still in S_{m+1} because none of the neighbors of u is in I_m; otherwise, u would have been removed from S_{m+1} and hence u should not have appeared as a candidate in I_{m+1}. Since $P(m)$ is true and $w \notin I_m$, there exists a vertex $v \in I_{m-1}$ due to which w was removed from S_m. This implies $d(w, v) \leq 1$. Due to triangle inequality $d(u, v) \leq d(u, w) + d(w, v) \leq 2$. Hence the proposition holds. □

Lemma 19. *In the worst case,* 19 *vertices in* I *share* 18 *nodes of* T *as intermediate nodes.*

Proof. Since I is an independent set and for each node $u \in I$, another node $v \in I$ exists, such as $d(u, v) \leq 2$. From Lemma 2, we know that a disk of radius 2 contains at most 19 independent unit disks. So in the worst case, those 19 vertices can share 18 nodes of T as intermediate nodes. □

Observation 7. If $|k| = 1$, then $|D_t| = |D_t^*|$, where D_t is the TDS returned by algorithm *TDS-UDG* and D_t^* is the optimal TDS of G.

Proof. If $k = 1$, the rooted tree T_x is a star graph with center vertex x. Hence $|D_t| = |D_t^*| = 2$. □

Lemma 20. $|D_t| \leq 7.79|D_t^*|$, *where* $|D_t|$ *is the size of the TDS returned by algorithm TDS-UDG and* $|D_t^*|$ *is the size of the optimal TDS of the graph* G.

Proof. The set D_t in *TDS-UDG* is a TDS of G, where $D_t = I \cup T$ (see Lemma 16). Since D_t^* is a TDS of G, there does not exist any isolated vertex in $G[D_t^*]$. For each $pq \in E(G[D_t^*])$, there exists at most 8 independent vertices in I which contains the vertices p and/or q (see Lemma 1). In worst case, there are at most $\lceil |D_t^*|/2 \rceil$ number of edges in $G[D_t^*]$. Therefore, size of I is at most $8 \times \lceil |D_t^*|/2 \rceil = 4|D_t^*|$, i.e.,

$$|I| \leq 4|D_t^*|.$$

The set T in *TDS-UDG* satisfies the total property when added to the independent set I. In the worst case, we know that 19 vertices in I share 18 vertices from T as intermediate nodes (see Lemma 19). So there are at most $\frac{18}{19} \times 4|D_t^*|$ vertices in T for $4|D_t^*|$ number of vertices in I, i.e.,

$$|T| \leq \frac{72}{19}|D_t^*|.$$

Therefore, from Lemma 16,

$$|D_t| = |I| + |T| \leq 4|D_t^*| + \frac{72}{19}|D_t^*| = \frac{148}{19}|D_t^*| \simeq 7.79|D_t^*|.$$

□

Theorem 8. *The proposed algorithm (TDS-UDG) gives a* 7.79*-factor approximation algorithm for the TDS problem in UDGs, which runs in* $O(|V| + |E|)$ *time.*

Proof. The approximation factor follows from Lemma 20, and the time complexity result follows from Lemma 17. □

5. Conclusion

In this paper, we studied the total Roman dominating set problem. We also revisited the total dominating set problem in unit disk graphs. We proposed a 10.5-factor approximation algorithm for the TRDS problem in UDGs with time complexity $O(|V| \log k)$, where k is the number of vertices with Roman value 2. For the TDS problem in UDGs, we proposed a 7.79-factor approximation algorithm, which runs in $O(|V| + |E|)$ time.

Authors Contribution. The author confirms sole responsibility of the research work presented in this article.

Conflict of interest. The author declares no conflict of interest.

Data Availability. Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

References

[1] H.A. Ahangar, M.A. Henning, V. Samodivkin, and I.G. Yero, *Total Roman domination in graphs*, Appl. Anal. Discret. Math. **10** (2016), no. 2, 501–517. http://doi.org/10.2298/AADM160802017A.

[2] J. Amjadi, S. Nazari-Moghaddam, S.M. Sheikholeslami, and L. Volkmann, *Total Roman domination number of trees*, Australas. J. Comb. **69** (2017), no. 2, 271–285.

[3] R.A. Beeler, T.W. Haynes, and S.T. Hedetniemi, *Double Roman domination*, Discrete Appl. Math. **211** (2016), 23–29. https://doi.org/10.1016/j.dam.2016.03.017.

[4] P. Chakradhar and P.V.S. Reddy, *Algorithmic aspects of total Roman {2}-domination in graphs*, Commun. Comb. Optim **7** (2022), no. 2, 183–192. https://doi.org/10.22049/cco.2021.27063.1189.

[5] M. Chellali, N. Jafari Rad, S.M. Sheikholeslami, and L. Volkmann, *Varieties of Roman domination*, Structures of Domination in Graphs (T.W. Haynes, S.T. Hedetniemi, and M.A. Henning, eds.), Springer, 2021, pp. 273–307.

[6] B.N. Clark, C.J. Colbourn, and D.S. Johnson, *Unit disk graphs*, Discrete Math. **86** (1990), no. 1-3, 165–177. https://doi.org/10.1016/0012-365X(90)90358-O.

[7] E.J. Cockayne, R.M. Dawes, and S.T. Hedetniemi, *Total domination in graphs*, Networks **10** (1980), no. 3, 211–219. https://doi.org/10.1002/net.3230100304.

[8] E.J. Cockayne, P.A. Dreyer Jr, S.M. Hedetniemi, and S.T. Hedetniemi, *Roman domination in graphs*, Discrete Math. **278** (2004), no. 1–3, 11–22. https://doi.org/10.1016/j.disc.2003.06.004.

[9] O. Favaron, H. Karami, R. Khoeilar, and S.M. Sheikholeslami, *On the Roman domination number of a graph*, Discrete Math. **309** (2009), no. 10, 3447–3451. https://doi.org/10.1016/j.disc.2008.09.043.

[10] G. Hao, L. Volkmann, and D.A. Mojdeh, *Total double Roman domination in graphs*, Commun. Comb. Optim. **5** (2020), no. 1, 27–39. https://doi.org/10.22049/cco.2019.26484.1118.

[11] T.W. Haynes, S. Hedetniemi, and P. Slater, *Fundamentals of Domination in Graphs*, CRC press, 1998.

[12] T.W. Haynes, S.T. Hedetniemi, and M.A. Henning, *Topics in Domination in Graphs*, Springer, 2020.

[13] S.T. Hedetniemi and R.C. Laskar, *Bibliography on domination in graphs and some basic definitions of domination parameters*, Discrete Math. **86** (1990), no. 1-3, 257–277. https://doi.org/10.1016/0012-365X(90)90365-O.

[14] M.A. Henning, *A survey of selected recent results on total domination in graphs*, Discrete Math. **309** (2009), no. 1, 32–63. https://doi.org/10.1016/j.disc.2007.12.044.

[15] M.A. Henning and A. Yeo, *Total Domination in Graphs*, Springer, 2013.

[16] C.H. Liu and G.J. Chang, *Roman domination on strongly chordal graphs*, J. Comb. Optim. **26** (2013), no. 3, 608–619. https://doi.org/10.1007/s10878-012-9482-y.

[17] M.V. Marathe, H. Breu, H.B. Hunt III, S.S. Ravi, and D.J. Rosenkrantz, *Simple heuristics for unit disk graphs*, Networks **25** (1995), no. 2, 59–68.

https://doi.org/10.1002/net.3230250205.

[18] A.C. Martinez, S. Cabrera Garcia, and A. Carrion Garcia, *Further results on the total Roman domination in graphs*, Mathematics **8** (2020), no. 3, Article ID: 349 https://doi.org/10.3390/math8030349.

[19] A. Poureidi, *Total Roman domination for proper interval graphs*, Electron. J. Graph Theory Appl. **8** (2020), no. 2, 401–413. https://dx.doi.org/10.5614/ejgta.2020.8.2.16.

[20] A. Poureidi and J. Fathali, *Algorithmic complexity of triple Roman dominating functions on graphs*, Commun. Comb. Optim. **9** (2024), no. 2, 217–232. https://doi.org/10.22049/cco.2023.27884.1385.

[21] K.J. Sangram and K.D. Gautam, *Total Domination in Geometric Unit Disk Graphs*, Proceedings of the 33rd Canadian Conference on Computational Geometry, 2021, August 10-12, 2021, 2021, pp. 219–227.

[22] W. Shang, X. Wang, and X. Hu, *Roman domination and its variants in unit disk graphs*, Discrete Math. Algorithms Appl. **2** (2010), no. 1, 99–105. https://doi.org/10.1142/S1793830910000504.

[23] Z. Shao, J. Amjadi, S.M. Sheikholeslami, and M. Valinavaz, *On the total double roman domination*, IEEE Access **7** (2019), 52035–52041. https://doi.org/10.1109/ACCESS.2019.2911659.

[24] I. Stewart, *Defend the Roman empire!*, Sci. Am. **281** (1999), no. 6, 136–138.

[25] P.J. Wan, K.M. Alzoubi, and O. Frieder, *Distributed construction of connected dominating set in wireless ad hoc networks*, Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, IEEE, 2002, pp. 1597–1604. https://doi.org/10.1109/INFCOM.2002.1019411.