A Variant of Connected Dominating Set in Unit Disk Graphs for Applications in Communication Networks

Djamel Djenouri, Miloud Bagaa
CERIST Research Center, Algiers, Algeria
Email: ddjenouri@acm.org, bagaa@mail.cerist.dz

***Abstract*—This paper considers a variant of the connected dominating set (CDS) problem in a unit disk graph $G = (V, E)$. The considered problem consists in minimizing the number of CDS vertices that belong to a subset $V' \subseteq V$. As far as we know, this problem has not been treated in the literature. Nevertheless, its resolution would be useful in many communication network applications, such as the selection of relay nodes in heterogenous wireless ad hoc networks where only a subset of powerful nodes (e.g., energy or memory rich nodes) may form the network backbone act as relays, or where it is preferable to select relays from these nodes and minimize the number of non-powerful nodes that act as relays. Replacement of non-powerful nodes might be necessary either at the initialization (after deployment), or during the network lifetime, which justifies the need to minimize their number. The problem is first modeled and reduced to the minimum weighted connected dominating set (WCDS) problem in a vertex weighted graph, and then it is resolved by taking advantage of the simple form of the weight function using integer linear programming (ILP). A heuristic is also proposed for large scale resolution. Simulation results confirms closeness of the proposed heuristic to the optimal solution obtained by the ILP, and scalability of the heuristic.**

I. Introduction and Background

A dominating set (DS) D of a $G = (V, E)$ is a subset of V, where every vertex $n \in V \setminus D$ is adjacent to at least one vertex in D. A CDS C of a graph G is a DS of G, where the subgraph $G[C]$ induced by C is connected. Finding the minimum connected dominating set (CDS) is very useful to construct backbones of relays/routers in communication networks. A typical example is wireless ad hoc and sensor networks, where the connected dominating set has largely been used to construct the backbone for cluster-based and hierarchal routing [1], [2]. Many centralized and distributed heuristics have been proposed for bounded graphs in general, and unit disk graphs in particular that represnt typical model for communication networks, e.g., [3], [4], [5]. While the fundamental problem of CDS has largely been treated by the graph theory community, as well as by engineering communities (computer network and wireless communication communities), finding the CDS that minimizes the number of elements from a subset of the connection graph ($V' \subseteq V$) has not been considered (to the best of our knowledge). This may be useful in many applications such as hybrid wireless networks, e.g, a WSN with energy harvesting enabled nodes and regular nodes, where it is preferable for power efficiency to avoid the regular (non-harvesting) nodes to be included in the CDS [6].

The minimum weighted connected dominating set (MWCDS) in vertex weighted graph has also been treated recently by the graph theory and operational research communities, and many heuristics has been proposed, e.g., [7], [8], [9], [10], [11], [12]. The MWCDS is a CDS variant (and can also be considered as a general form of CDS) where vertices are assigned weights, and the objective function is to minimize the weight of the set, instead of simply minimizing the cardinality of set (number of its vertices). In this paper, the problem of finding a CDS with a minimum number of vertices from a subset of the graph is modelled as a variant of the MWCDS, and resolved with an exact (optimal) solution and a heuristic. Optimal resolution is derived using integer linear programming (ILP), and the heuristic is given as approximative solution for large scale scenarios. The simple form of the weight function is exploited both in the ILP and the heuristic to reduce runtime. The proposed solution is evaluated by simulations.

The remaining of the paper is organized as follow. The problem formulation is given in Sec. II, followed by the problem transformation in Sec. III. The problem resolution by ILP then a heuristic is presented in Sec. IV. The simulation analysis is summarized in Sec. V, and finally Sec. VI concludes the paper.

II. Problem Formulation

Let $CDS(G)$ denotes all the connected dominating set in $G = (V, E)$. The problem considered herein is to find a connected dominating set ($S \in CDS(G)$) with minimum number of vertices from $V' \subseteq V$. Let us denote this problem by $P1$, with instantiation $G = (V, V', E)$. This problem is very similar to the traditional minimum connected dominating set problem (MCDSP), which is known to be NP-hard. The only difference is to minimize the number of elements in the MCDS from a subset of V ($V' \subseteq V$), instead of minimizing the total number of the MCDS elements.

It is easy to demonstrate the general form of this variant is also NP-hard by using the traditional MCDSP. In other words, we should prove that the ordinary MCDSP reduces to $P1$. Searching an $MCDS(G = (V, E))$ is equivalent to $P1$, with instance $G' = (V, V, E)$, i.e., $V = V$. That is, the solution of $P1$ with instance G', say S, is a solution to the traditional MCDSP ($S \in MCDS(G)$). Therefore, $MCDSP <_p P1$[1], and thus The problem $P1$ is NP-hard.

III. Problem Transformation

The problem is transformed to the search for a minimum-weighted connected dominating set in a vertex weighted graph. The graph, $G = (V, V', E)$, of $P1$ is transformed into a vertex weighted graph, $G_w = (V, E, W)$, as follows.

$$\forall u \in \{V \setminus V'\}, W(u) = 0, \forall v \in_{V'}, W(v) = 1. \quad (1)$$

The problem $P1$ then reduces to finding a *minimum-weight* connected dominating set ($S \in MWCDS(G_w)$). In the following, the new problem is denoted by $P2$.

Theorem 1: If S is a MWCDS for $P2$ (i.e, S $\in MWCDS(Gw)$), then it is an optimal solution to $P1$.

Proof: The aforementioned theorem may be proved by contradiction. Suppose $S \in MWCDS(Gw)$ for $P2$ (optimal solution), i.e.,

$$W(S) = \sum_{u \in S} W(u) = \min_{\zeta \in CDS(G)} \sum_{u \in \zeta} W(u), \quad (2)$$

and assume it is not an optimal solution to $P1$. This means,

$$\exists S' \in CDS(G), |S' \cap (V')| < |S \cap (V')| \quad (3)$$

From Eq. 1, it results that:

$$W(S) = |S \cap \{V'\}|, \quad (4)$$

$$W(S') = |S' \cap \{V'\}| \quad (5)$$

Therefore, Eq. 3 yields: $W(S') < W(S)$. This is contradictory with the initial assumption that $S \in MWCDS(Gw)$ (Eq. 2). □

Theorem 1 yields the following corollary:

Corollary 1: Resolving $P1$ is thus equivalent to resolving $P2$.

Theorem 2: Let the problem P_3 be defined for $G = (V, V', E)$ as follows: Find S such that,

$$S = \underset{\xi \cup \{V \setminus V'\} \in CDS(G)}{\arg\min} |\xi|.$$

If S is a solution to P_3, then $S \cup \{V \setminus V'\}$ is a solution to $P1$.

Proof: Since any solution to $P2$ is such for $P1$, (Theorem 1), we just need to prove that $S \cup \{V \setminus V'\}$ is a solution to $P2$ in the extended graph Gw, i.e., $S \cup \{V \setminus V'\} \in MWCDS(Gw)$, to deduce it is such for $P1$.

We proceed by contradiction. Let S be a solution to P_3, and suppose $S \cup \{V \setminus V'\} \notin MWCDS(Gw)$. From the definition of the weight function (Eq. 1),

$$W(S \cup \{V \setminus V'\}) = W(S) = |S|. \quad (6)$$

Now consider $S' \in MWCDS(Gw)$, and let it be divided into two disjoint subsets: i) S'_{in}, the subset of vertices that belong to V', and ii) S'_{out}, the subset of remaining vertices that belong to

$$\{V \setminus V'\}$$

.

From the definition of the weight function (Eq. 1),

$$W(S') = |S'_{in}|. \quad (7)$$

Since $S \cup \{V \setminus V'\} \notin MWCDS(Gw)$, and $S \cup \{V \setminus V'\} \in CDS(G)$, it results that $W(S') < W(S)$. From Eq. 6 and Eq. 7, $W(S') < W(S)$ implies

$$|S'_{in}| < |S|. \quad (8)$$

Since $S'_{out} \subseteq \{V \setminus V'\}$, it results:

$$S'_{in} \cup \{V \setminus V'\} \in CDS(G). \quad (9)$$

Eq. 8 and Eq. 9 represent a contradiction with S definition ($S = \underset{\xi \cup \{V \setminus V'\} \in CDS(G)}{\arg\min} |\xi|$).

Thus inevitably, $W(S') = W(S)$, which means $S \cup \{V \setminus V'\} \in MWCDS(Gw)$. □

It results from Theorem 2 the following:

Corollary 2: Resolving $P1$ (or $P2$) is equivalent to adding a minimal set, say S, to $\{V \setminus V'\}$, to construct a solution $S \cup \{V \setminus V'\} \in MCDS(G)$ (respectively $S \cup \{V \setminus V'\} \in MWCDS(Gw)$).

This will be useful in the following to reduce the search space.

IV. Problem Resolution

A. Exact Solution

To resolve $P1$, we use P_3 that helps reducing the number of variables and thus the search space compared to $P2$. The problem P_3 can be modeled by the following mixed linear program:

$$\min \sum_{i \in V'} X_i. \quad (10)$$

S.t,

$$X_i + \sum_{j \in \mathcal{N}(i)} X_j \geq 1, \forall i \in V \quad (11)$$

$$\sum_{i \in \mathcal{N}(1)} F_{1,i} = \sum_{i \in V, i \neq 1} X_i \quad (12)$$

$$\sum_{j \in \mathcal{N}(i)} F_{j,i} - \sum_{j \in \mathcal{N}(i)} F_{i,j} = X_i, \forall i \in V, i \neq 1 \quad (13)$$

[1]MCDSP is not more difficult to a polynomial order than $P1$

$$0 \le F_{i,j} \le nX_j, \forall (i,j) \in E, j \neq 1 \quad (14)$$

$$\sum_{i \in N(1)} Y_i \le 1 + X_1(|N(1)| - 1) \quad (15)$$

$$F_{1,i} \le nY_i, \forall i \in N(1) \quad (16)$$

$$F_{i,j} = 0, \forall (i,j) \notin E, \text{ or } j = 1 \quad (17)$$

$$X_i = 1, \forall i \in \{V \setminus V'\}, \quad (18)$$

ILP has as input the graph $G = (V, E, H)$, from which the set of adjacent vertices (neighboring vertices), $N(i)$, can be deduced for every vertex, v_i. The outputs are: i) a vector of booleans, X, which represents the decision variables, i.e., $X_i = 1$ iff vertex, $v_i \in V$, is selected in the MCDS. Variables on X are only for $v_i \in V'$, while entries for $v_i \in \{V \setminus V'\}$ are fixed a priori to 1 (Eq. 18). ii) The flow matrix of integers, $F_{i,j}$, $(v_i, v_j) \in E$, as well as the vector Y_i, for $v_i \in N(1)$, are additive variables used to model the connectivity as it will be explained hereafter.

The objective function, Eq. 10, is to minimize the total weight of vertices in the selected CDS, to achieve MWCDS. The constraint represented by Eq. 11 is to guaranty that either v_i is in the CDS ($X_i = 1$), or it has some edge towards some vertex in the CDS (at least one of the terms X_j should equal 1). Constraints represented by Eq. 12 to Eq. 16 are for modeling the connectivity requirements. The principle is to generate a flow, only from an arbitrary vertex v_1. The amount of this flow is the exact amount to cover the CDS (Eq. 12), i.e., it should be $\sum X_i$ if v_1 is out of the set ($X_1 = 0$), or $\sum X_i - 1$ if it belongs to the CDS (one of the dominating vertices). In the former case, v_1 inevitably would have at least one edge towards a dominating vertex. The generated follow traverse the dominating vertices and at every one, a single unit of the flow fades (Eq. 13). Eq. 14 verifies that every flow is bounded by 0 and n, and that no flow goes to dominated vertices. This is as the term $F_{i,j}$ vanishes when $X_i = 0$. Note here that a more strict upper bound that would reduce the search space is $\sum Xi$ instead of n, but this would make the inequalities non-linear. Also note that the latter condition, combined with Eq. 12 when $X_i = 0$, ensures no flow will be generated from vertices out of the CDS.

Constraints Eq. 15 and Eq. 16 are used to limit the number of neighbors to which node v_1 can transfer its flow. A binary vector, Y, is added to the outputs such that $Y_i = 1$ iff flow is permitted from node, v_1 to node, v_i. Constraint Eq. 16 ensures that flow can only be transferred from node v_1 to v_j if $Y_j = 1$, while Eq. 15 forces Y_j to be equal to one for only one adjacent vertex, v_j, in case, $v_1 \notin CDS$. Otherwise, it is bounded by the number of v_1's adjacent vertices. Final, conditions expressed by Eq. 17 are to ensure the flow travels only through existing edges, and no flows enters v_1, and Eq. 18 to set X entries to 1 for vertices in $\{V \setminus V'\}$ (constants). Note that the latter conditions (Eq. 17 and Eq 18) are just to reduce the number of the ILP variables, and do not represent constraints to be verified by the ILP resolver. This reduces the search space and thus the runtime.

This ILP represent an exact solution to the problem, and can be used for limited size graphs. We checked the resolution with CPLEX for graphs up to 150 vertices using a single PC. Parallel implementation can be used for larger scales. However, even the parallel implementation would have limitations, giving the exponential worst case complexity of the ILP. To overcome this limitation, a heuristic is given in the following.

B. Heuristic

We propose a heuristic for unit disk graphs based on [7] and [12], while considering the particularity of the problem, as illustrated by Algorithm 1. The principal of [7] is used to find an approximation of the minimum weighted dominating set (MWDS), say χ_1. This starts by initializing χ_1 to $\emptyset$. The latter is progressively augmented by adding local dominating vertices (D_{k+2}) of $(k+2) - neighborhood$ from a pivotal vertex v (the loop from line 4 to line 20). If there is a vertex where all its adjacent vertices are in V', then it is select as v. Otherwise, v is chosen arbitrary (lines 5 through line 9). The inner loop (repeat loop) searches for local dominating set for the $k+2$ vicinity from v ($\mathcal{N}_{k+2}(v)$), and it increases k until the condition of line 17 is fulfilled. The condition states that the weight of the local dominating set for $\mathcal{N}_{k+2}$ is no more than that for $\mathcal{N}_k$ when multiplied by the factor $(1+\epsilon)$. The result of Theorem 2 is used to accelerate the search, and all the vertices with null weight in the appropriate vicinity are initially picked up in every iteration (line 13 and line 15 for k neighborhood, $k+2$ neighborhood, respectively) instead of performing an exhaustive search among all k, (respectively $k+2$) neighbors. Those with non-null weight ($\mathcal{N}_k(v) \cap (V')$, and $\mathcal{N}_{k+2}(v) \cap (V')$) are added progressively if necessary to construct the local MWDS with a minimum addition. The local MWDS is added to χ, and the loop continues by selecting another pivotal vertex until covering the set V. At that point (termination of the $while$ loop), χ_1 will represent an approximation of MWDS, but not necessary connected.

The connection of ($\chi_1$second part of the algorithm) is similar to the solution used in [12] to connect the resulted approximation of the MWDS and construct an approximation of the MWCDS. Although the solution of [12] used a different approach to calculate a MWDS, the connection algorithm proposed by the authors is general and connects any DS. The calculation of the connector set, λ, start by extracting the connected parts in χ_1, then every connected part is clustered in c_i. After clustering, an auxiliary multiple graph $\hat{G}$ is constructed from G as follows: i) The vertices are the clusters (c_i). ii) Between every couple of vertices, $(c_i, c_j) \in \hat{G}^2$, and for every path, $p \in G$, of length not exceeding three hops, an edge between c_i and c_j is added. The resulted graph is a multiple graph with possible multiple edges between a couple of vertices. Note that every p includes only vertices that do not belong to χ_1, and which will possibly be used to connect vertices of χ_1. A minimum spanning tree of $\hat{G}$ is then calculated. λ is then constructed from vertices in G that form every single edge in the calculated spanning tree. χ_1 is augmented with λ to construct the connected χ.

Algorithm 1: Heuristic for $G = (V, V', E, W)$

Init:
$\chi = \chi_1 = \emptyset$
$\lambda = \emptyset$
Contruction of a Dominating Set χ_1
while $V \neq \emptyset$ **do**
  if $\exists u \in V, \forall \omega \in \mathcal{N}_1(u), v \in V'$ **then**
    $v = u$
  else
    Chose arbitrary vertex v
  end if
  $k = -1$
  repeat
    $k = k + 1$
    $D_k = \mathcal{N}_k(v) \cap (V \setminus V')$
    add progressively to D_k a minimum number of non-null weight vertices ($u \in \mathcal{N}_k(v) \cap (V')$); until D_k dominates $\mathcal{N}_k(v)$
    $D_{k+2} = \mathcal{N}_{k+2}(v) \cap (V \setminus V')$
    add progressively to D_{k+2} a minimum number of non-null weight vertices ($u \in \mathcal{N}_{k+2}(v) \cap (V')$); until D_{k+2} dominates $\mathcal{N}_{k+2}(v)$
  until $\sum_{u \in D_{k+2}} W(u) \leq (1 + \epsilon) \sum_{u \in D_k} W(u)$
  $\chi_1 = \chi_1 \cup D_{k+2}$
  $V = V \setminus \mathcal{N}_{k+2}(v)$
end while
Connecting χ_1
Extract the connected components in χ_1 and cluster them, i.e. denote every cluster c_i
Construct an auxiliary graph, $\hat{G} = \{\hat{V}, \hat{E}, \hat{W}s\}$ from G as follows:
$\hat{V} = \{c_i\}$
For every path, $p \in G$, of length 3 or less that connects a vertex from c_i to another one from c_j, add an edge, e, to $\hat{E}$, and set $\hat{W}(e) = W(p)$
Compute a minimum spanning tree MST of $\hat{G}$
For every edge $e \in MST$, add the vertices that form e to λ
$\chi = \chi_1 \cup \lambda$

V. Simulation Analysis

The proposed solution is evaluated by simulation in this section. Both the exact resolution (ILP) and the heuristic algorithm are evaluated. The number of vertices is varied in random generated graphs, and the objective function (number of vertices from V' in the respective CDS, i.e., $CDS \cap V'$), as well as the runtime, have been measured. We used NetworkX environment to implement the heuristic, and CPLEX to resolve the ILP. We run the simulation on an $i7$, $3.4GH$ CPU computer, with $8GB$ of RAM. Every point in the following plots is the average of extensive repetitions with different random generated graphs, and the error bars are presented with 95% confidence interval.

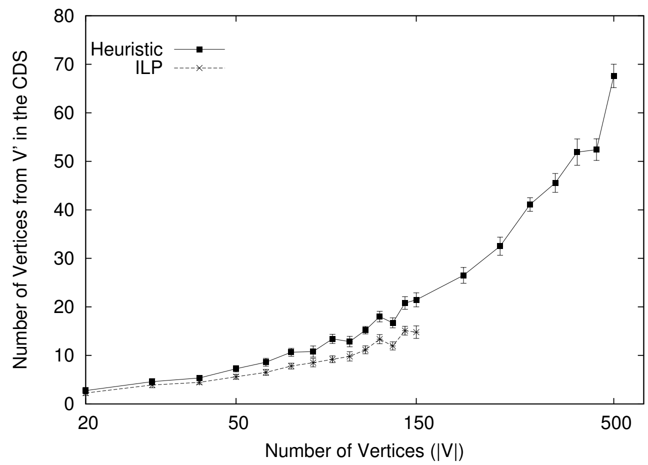

Fig. 1: Number of vertices from V' in the CDS vs. number of vertices ($|V|$)

Fig .1 plots the achieved values of the object function (number of vertices from V' in the obtained CDS). It shows that the inevitable increase of the cost vs. the number of vertices is smooth, and it confirms scalability of the heuristic whose cost does not exceed 68 vertices for 500 vertices. The ILP cost is even lower, which is obvious as it represents the optimum. But note that it was not possible with our setting to assess it for scenarios beyond 150 vertices due to memory overflow, contrary to the heuristic. It is worthy to note here the closeness of the two solutions with respect to this metric, e.g, 18 vs. 13 for 120 vertices, and 21 vs. 14 for 150 vertices. However, the picture is different with Fig. 2 where the difference is between the ILP and the heuristic is much more relevant. The plots in this figure confirm scalability of the heuristic and the exponential increase of the runtime for the ILP. The latter has been reasonable up to 100 vertices, than it increases dramatically with the number of vertices. It was not possible to simulate scenarios beyond 150 for the ILP due to memory overflow. A parallel version would help rising a bit this limit, but ILP solution will always have a limit. Nonetheless, the ILP can be useful for low scale applications, e.g., local area or ad hoc sensor networks, but not in large scale scenarios.

VI. Conclusion

A variant of the connected dominating set (CDS) problem in a graph $G = (V, E)$ is considered in this paper, where the objective function is to minimize the number of vertices from a subset $V' \subseteq V$, in the CDS. This problem resolution has application in communication and computer networks, such as the selection of relay nodes in heterogonous wireless ad

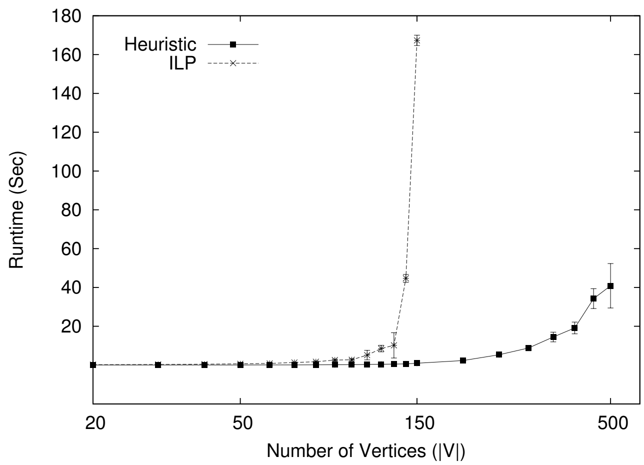

Fig. 2: Runtime vs. number of vertices ($|V|$)

hoc and sensor networks where only a subset of powerful nodes (e.g., energy or memory rich nodes) may take part in the networks backbone to act as relays. This NP-hard problem is modeled and reduced to the minimum weighted connected dominating set problem in a vertex weighted graph. It is then resolved by taking advantage of the simple form of the weight function using integer linear programming (ILP), and then with a heuristic. The proposed solution has been analyzed by simulation, and the results confirm closeness of the proposed heuristic to the optimal solution obtained by the ILP. And more importantly, it confirms scalability of the heuristic. the ILP is still useful to derive optimal solution for low scale applications. With a single computer, we were able to simulate scenarios up to 150 vertices. A parallel implementation would help rising a bit this limit, which represent a potential perspective. Application of the proposed solution to relay node selection in future generation energy harvesting WSN is another research perspective in our agenda.

REFERENCES

[1] J. Yu, N. Wang, G. Wang, and D. Yu, "Connected dominating sets in wireless ad hoc and sensor networks a comprehensive survey," *Computer Communications*, vol. 36, pp. 121–134, 2013.

[2] J. Yu, N. Wang, and G. Wang, "Constructing minimum extended weakly-connected dominating sets for clustering in ad hoc networks," *J. Parallel Distrib. Comput.*, vol. 72, no. 1, pp. 35–47, 2012.

[3] Y. P. Chen and A. L. Liestman, "Maintaining weakly-connected dominating sets for clustering ad hoc networks," *Ad Hoc Netw.*, vol. 3, no. 5, pp. 629–642, 2005.

[4] K. Sakai, S. C. H. Huang, W.-S. Ku, M.-T. Sun, and X. Cheng, "Timer-based cds construction in wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 10, pp. 1388–1402, 2011.

[5] H.-Y. Yang, C.-H. Lin, and M.-J. Tsai, "Distributed algorithm for efficient construction and maintenance of connected k-hop dominating sets in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 4, pp. 444–457, 2008.

[6] D. Djenouri and M. Bagaa, "Energy harvesting aware relay node addition for power-efficient coverage in wireless sensor networks," in *IEEE International Conference on Communications (ICC)*, 2015.

[7] Z. Wang, W. Wang, J.-M. Kim, B. Thuraisingham, and W. Wu, "Ptas for the minimum weighted dominating set in growth bounded graphs," *Journal of Global Optimization*, vol. 54, no. 3, pp. 641–648, Nov 2012.

[8] M. Hassinen, J. Kaasinen, E. Kranakis, V. Polishchuk, J. Suomela, and A. Wiese, "Analysing local algorithms in location-aware quasi-unit-disk graphs," *Discrete Appl. Math.*, vol. 159, no. 15, pp. 1566–1580, Sep. 2011.

[9] F. Zou, Y. Wang, X. Xu, X. Li, H. Du, P.-J. Wan, and W. Wu, "New approximations for minimum-weighted dominating sets and minimum-weighted connected dominating sets on unit disk graphs," *Theoritical Computer Science*, vol. 412, no. 3, pp. 198–208, 2011.

[10] G. D. da Fonseca, C. M. H. de Figueiredo, V. G. P. de Sá, and R. C. S. Machado, "Efficient sub-5 approximations for minimum dominating sets in unit disk graphs," *Theor. Comput. Sci.*, vol. 540, pp. 70–81, 2014.

[11] D. Kim, Z. Zhang, X. Li, W. Wang, W. Wu, and D.-Z. Du, "A better approximation algorithm for computing connected dominating sets in unit ball graphs," *IEEE Trans. Mob. Comput.*, vol. 9, no. 8, pp. 1108–1118, 2010.

[12] Y. Huang, X. Gao, Z. Zhang, and W. Wu, "A better constant-factor approximation for weighted dominating set in unit disk graph," *Journal of Combinatory Optimimization*, vol. 18, no. 2, pp. 179–194, 2009.