An Efficient Heuristic for the Minimum Connected Dominating Set Problem on Ad Hoc Wireless Networks

S. Balaji and N. Revathi

Abstract—Connected dominating set (CDS) problem in unit disk graph has signi£cant impact on an ef£cient design of routing protocols in wireless sensor networks, where the searching space for a route is reduced to nodes in the set. A set is dominating if all the nodes in the system are either in the set or neighbors of nodes in the set. In this paper, a simple and ef£cient heuristic method is proposed for £nding a minimum connected dominating set (MCDS) in ad hoc wireless networks based on the new parameter support of vertices. With this parameter the proposed heuristic approach effectively £nds the MCDS of a graph. Extensive computational experiments show that the proposed approach outperforms the recently proposed heuristics found in the literature for the MCDS.

Keywords—ad hoc wireless networks, dominating sets, unit disk graphs, heuristic.

I. INTRODUCTION

LOCAL search (or local improvement) is a practical tool and a common technique for £nding near-optimal solutions in reasonable time for combinatorial optimization problems. In many cases, local search can be incorporated into more sophisticated methods called meta-heuristics, in order to obtain more high-quality solutions. The basic idea of local search is start from a feasible solution x and repeatedly replace x with better x' which is selected from neighborhood of x de£ned as the set of neighbor solutions that can be reached by making slight modi£cations to x. If no better solutions can be found in its neighborhood, local search immediately stops and returns as £nal the best solution found during search.

To keep the computation time within a reasonable limit, we can use variable depth search (VDS), a well known generalization of local search methods. The idea of VDS was £rst applied by Lin and Kernighan to the traveling salesman problem (TSP) [21] in 1973 and graph partitioning problem (GPP) [19] in 1970. The basic concept is to search a portion of the large neighborhood within a reasonable amount of computation time.

A. Local search on optimization problems

Recently, for TSP and GPP, the VDS-based heuristics have been incorporated into several metaheuristic frameworks, such as iterated local search [2, 17] and evolutionary algorithm [24, 25]. Generally, the performance of metaheuristics embedded

S. Balaji is with the Department of Mathematics, SASTRA University, Thanjavur, 613 401, INDIA, e-mail: (balaji@maths.sastra.edu).

N.Revathi is with the Department of Mathematics, RVS Institutions, Coimbatore, 641 402, INDIA, e-mail: (reva17787@yahoo.co.in).

with VDS based local search is remarkably effective for the hard problems TSP and GPP. For some other hard problems, effective VDS algorithms have been proposed. For the generalized assignment problem, Yagiura et al. [36] suggested an algorithm. For the unconstrained binary quadratic programming problem (UBQP), an effective local search based on VDS was proposed by [24]. Merz and Katayama [26] recently proposed a memetic algorithm with the variant VDS-based local search for UBQP and reported that the memetic algorithm is highly effective.

More recently K. Katayama et al. [18] proposed a local search algorithm inspired by VDS for the maximum clique problem (MCP) and they claimed that their algorithm capable of £nding better average solutions than compared metaheurstics. Pullan [28, 29] proposed a phased local search algorithm for both MCP and weighted MCP and he claimed that it achieves state-of-the-art performance for those problems. Judging from these contributions, we can expect that metaheuristics embedded with local search heuristics to offer promising approaches to other hard problems, such as the minimum vertex cover problem and its associated decision problems.

The variable depth search (VDS) based local search algorithms are known to be highly effective for several combinatorial optimization problems. It is worth while considering a new local search based on an ancient idea of VDS for solving the other hard problems. With this motivation, we attempted to develop a heuristic algorithm inspired by VDS, the proposed algorithms, for solving the graph optimization problems.

B. MCDS on Adhoc wireless networks

Given an undirected graph $G(V, E)$ with the set of vertices V and the set of edges E, a dominating set (DS) is a set $D \subseteq V$ such that each vertex in $V - D$ is adjacent to at least one vertex in D. If the subgraph induced by D (denoted by $G[D]$) is connected, D is called connected dominating set (CDS). The minimum CDS problem asks for £nding a CDS of a graph with its size as small as possible. However, £nding a CDS (MCDS) for arbitrary graphs has shown to be NP-complete [12].

The initial topology of ad hoc wireless networks can be modeled as a unit disk graph (UDG) which consists of hosts in a wireless network as a node set V and an edge set E such that any edge $(u, v) \in E$ if and only if the distance between u and v is not more than r, where r means the transmission

World Academy of Science, Engineering and Technology
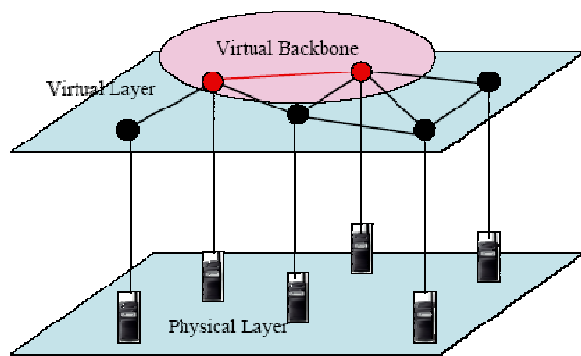International Journal of Mathematical and Computational Sciences
Vol:6, No:8, 2012

Fig. 1. Connected dominating set virtual back bone in a unit disk graph

range of the nodes (see Fig. 1). For the purpose of efficient communication, it is necessary to set up a CDS as virtual backbone. However, even in UDG the problem of finding an MCDS is still NP-complete [3]. Therefore, polynomial-time efficient heuristics are of interest for constructing small size CDS. Heuristics developed for CDS can be divided into two sets. The first set of heuristics strives to find disconnected maximum independent set (MIS) of nodes that are joined through minimum spanning tree or Steiner tree [1]. The second type of heuristics concentrates on evolving a CDS by growing a small trivial CDS [11]. Different techniques have been proposed for the MCDS problem in the recent years [4, 5, 9, 22, 27, 35].

One set of algorithms were based on the idea of creating a dominating set incrementally. The other set of algorithms uses initial set as CDS, recursively remove vertices using Steiner tree etc.,[5]. Some approaches try to construct a MCDS by finding a maximal independent set, which is then expanded to a CDS by adding (connecting) vertices [4][27]. Other than that some approaches follows geometrical properties of nodes such as angle area [22] and some others obtaining partial Steiner tree during the construction of independent set [27] for constructing MCDS.

A common feature of the currently available techniques for solving the MCDS problem is that the algorithms create the CDS from scratch, adding some vertices at each iteration according to a greedy criterion. For the general dominating set problem, the only exception known to the authors is briefly explained in [31], where a solution is created by sequentially removing vertices. Such type of procedure may cause more energy consumption, additional set up time and suffer more time delay when forwarding packets. Another disadvantage of the existing approaches is that frequently they use complicated strategies in order to achieve a good performance guarantee. This fact implies that the current approximation factors do not reflect the actual performance of the algorithm outputs. In addition, some worst case (time or message) complexities, though have theoretical means, do not reflect the actual implementation cost of the algorithm.

In this paper, a new heuristic algorithm for computing approximate solutions to the MCDS is proposed. In particular, the application of this algorithm to the MCDS problem in unit-disk graphs discussed in detail. The algorithm defined with the

parameter called support of vertices, to identify a dominating node to cover large number of vertices and hence for the MCDS. With this parameter the proposed heuristic effectively finds the connected dominating set of a graph and moreover by simple local search approach, the proposed heuristic finds the optimal solution for the MCDS. This approach also has the advantage of being simple to implement, interms of its computational complexity, less set up cost and providing good solution compared to other existing algorithms in the literature.

The rest of the paper is organized as follows. Works related to the present study is described in section II. Section III drives the advantages of taken parameter. Section IV provides the preliminaries of Graph Theory. In section V, description of the proposed SLS algorithm is presented. Detailed study of experiments and their related results are described in section VI and the paper concludes in section VII.

II. RELATED WORKS

In early 1960s, Claude Berge first formulated the mathematical concept of domination in graphs. Half century has passed since then and thousands of research papers have been published on this topic [15]. There are several polynomial-time algorithms for the MCDS problem. For instance, Guha and Khuller [13] propose an algorithm with approximation factor of $H(\Delta) + 2$, where Δ is the maximum degree of a graph and $H(n) = 1 + 1/2 + 1/3 + ... + 1/n$ is the harmonic function. Other approximation algorithms are given in [4, 23]. Other centralized algorithms were also proposed [23] with different approximation factors. However, large wireless sensor networks with limited energy are not amenable to centralized algorithms. Distributed techniques that need limited hop information are required for such networks. In [20], a distributed algorithm was proposed that utilized maximal independent set for CDS construction in UDG. However, the technique requires large amount of message exchanges and transmission for the construction of spanning tree for constructing MCDS. In [32][33], a distributed technique was given that constructed a CDS which is twice the size of the corresponding MCDS. In [1], a constant factor algorithm was given without the use of spanning tree construction. Some techniques use the Steiner tree to connect the maximal independent set nodes [7][16]. In [16], a CDS is considered and the nodes are removed recursively to discover an MCDS. However, the construction of the CDS is not given. These techniques not localized and may require the complete network information depending on the status of the neighboring nodes. In [8] construction of a k-connected k-dominating set (k-CDS) is proposed as a backbone to balance efficiency and fault tolerance. In which four localized k-CDS construction protocols developed and they claimed over simulation results that these protocols can select a small k-CDS with relatively low overhead. In [30] a timer based CDS protocols introduced, which first elect a number of initiators distributively and then utilize timers to construct a CDS from initiators with the minimum localized information. In [38] diameter between two communication parties considered as a quality factor for CDS construction, based on that two new heuristic algorithms developed for

World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:6, No:8, 2012

CDS. Recent developments of wireless networking, dominating and packing problems were discussed in [34]. In [37], a connected dominating set based localized mechanism is proposed. Initially a basic backbone is constructed, then when a query is issued, new sensors are activated locally such that to meet the requirements of the query and global connectivity. The £rst polynomial time approximation scheme designed in [11] for d-hop connected dominating set (d-CDS) problem in growth bounded graphs. The main strategy used on the scheme is clustering partition. In [6] a constant-ratio approximation algorithm for MCDS problem in unit ball graphs is proposed £rst and then introduced and analyzed the edge-weighted bottleneck CDS problem, which seeks a minimum edge weight in a graph such that the corresponding bottleneck subgraph has a CDS of size k. The work in [39] investigate the trade-offs among multiple factors in CDS construction, such as fault tolerance, size, diameter and running time. They introduced multi-factors model studying a joint optimization problem in which the objective is to optimize the CDS size, network latency or running time while keeping the fault tolerance. A novel zone-based distributed algorithm for CDS presented in [14]. They combined the zone and level concepts and partition the wireless network into different zones, constructed a dominating tree for each zone and connected adjacent zones by inserting additional connectors into the £nal CDS and they claimed their zone-based algorithm is more effective than previous approaches.

Most existing heuristics for the MCDS problem work by selecting vertices to be a part of the dominating set and adding them to the £nal solution. In this situation we should pay attention to following conditions:

1) In what optimistic way we may select a vertex has been a position to cover (pass messages) most of the nodes in to a CDS;
2) In what way to check the speci£c vertices which get covered by one vertex and whether or not they could be covered by some other vertices in alternate way;
3) In what simplest way we can remove the redundant nodes in the construction of CDS in order to obtain a MCDS.

By considering the above conditions and to make simple and ef£cient heuristic, the following algorithm is designed with the parameter *support* of vertices to approximate the MCDS of a graph.

III. DRIVEN PARAMETER: ADVANTAGE

From the recent literature on MCDS algorithms, it seems that, most algorithm have been constructed by taking into account of some troublesome parameters and vertices with maximum degree or minimum degree were added into a feasible solution. These selection process yields more number of random selections (due to tie in maximum or minimum degree) and it indeed yields more number of combinations of feasible solution. Therefore time taken by an algorithm to get a near optimal solution from these more number of feasible solutions becomes very high. In order to avoid more number of random selections as much as possible and to make a heuristic
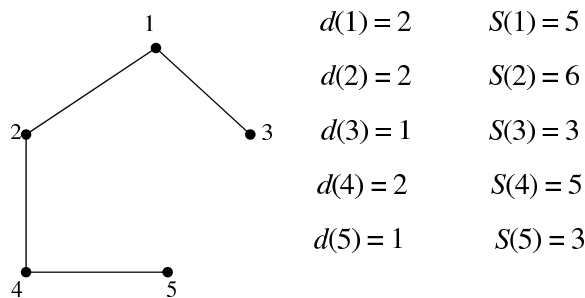

$$d(1) = 2 \qquad S(1) = 5$$
$$d(2) = 2 \qquad S(2) = 6$$
$$d(3) = 1 \qquad S(3) = 3$$
$$d(4) = 2 \qquad S(4) = 5$$
$$d(5) = 1 \qquad S(5) = 3$$

Fig. 2. Selection based on *support* value

better, a parameter called *support* of a vertex is de£ned and implemented in this research work.

For each $v \in V$, *support* of a vertex is de£ned by
$$s(v) = d(v) + \Sigma_{u \in N(v)} d_G(u)$$
The quantity $\Sigma_{u \in N(v)} d_G(u)$ is the sum of the degree of vertices which are adjacent to v.

It is worthy to note that the selection of vertices with maximum *support* value or minimum *support* value into a feasible solution decidedly reduce the number of random selections and the number of trials and also the execution time to get a near optimal solution.

1) Example: To illustrate how the $degree$ based selection and the new parameter *support* based selection of vertices differs, Fig. 2 gives an example for selection of a vertex based on maximum *support* value of a vertex. In this £gure vertices are numbered 1 to 5. When selecting vertices with maximum $degree$, there is a tie situation with the vertices 1, 2 and 4, each of them shares same $degree$ 2. But selecting vertices based on maximum *support* value of a vertex selects the only one vertex 2 as maximum *support* value vertex. i.e., there is no tie situation. In this way the new parameter reduces the number of random selection among vertices as much as possible and also the number of iterations to get a near optimal solution.

IV. PRELIMINARIES

Let $G = (V, E)$ be an undirected graph, where $V = \{v_1, v_2, ..., v_n\}$ is the set of vertices and $E \subseteq V \times V$ (not in ordered pairs) is the set of edges with cardinality of $|V| = n$ and $|E| = m$ and the complement graph of $G(V, E)$ is the graph $\overline{G}(V, \overline{E})$, where $\overline{E} = \{(v_i, v_j) \in V, v_i \neq v_j \text{ and } (v_i, v_j) \notin E\}$. Then we have the following basic de£nitions relative to the forthcoming chapters:

Neighborhood of a vertex: For each $v \in V$, the neighborhood of v is de£ned by $N(v) = \{u \in V/u \text{ is adjacent to } v\}$ and the closed neighbourhood of v is de£ned by $N[v] = \{v\} \cup N(v)$.

Degree of a vertex: The degree of a vertex $v \in V$, denoted by $d(v)$ and is de£ned by the number of neighbors of v i.e., $d(v) = |N(v)|$.

Dominating set: A dominating set for a graph $G(V, E)$ is a subset D of V such that every vertex not in D is adjacent with atleast one member of D. The minimum cardinality

World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:6, No:8, 2012

of a dominating set is denoted by $\gamma(G)$ and is called the domination number of G

Minimal dominating set: Minimal dominating set (mDS) is a dominating set (DS) such that any proper subset of mDS is not a DS; in other words, for any $v \in$ mDS either v is an isolate of the mDS or there exists at least one node $u \in V-$mDS such that u is not dominated by any node in mDS except v.

Connected dominating set: A dominating set D of a graph $G(V, E)$ is said to be a connected dominating set (CDS) if the subgraph induced by D is connected.

Minimal connected dominating set: Minimal connected dominating set (mCDS) is a CDS such that removing any node from this set will make it no longer a CDS.

V. ALGORITHM

The following algorithm is designed to approximate the MCDS of an undirected graph $G(V, E)$. The proposed algorithm is divided into three phases. In the £rst phase a dominating set is constructed and to make the dominating set as connected one, in the 2nd phase, connector nodes are found with the help of Steiner tree. In the £nal phase exhaustive local search procedure is applied to reduce the number of nodes in the CDS, make it as an near optimal minimum connected dominating set. The proposed SLS is as follows:

A. Support based Local Search (SLS): Algorithm for MCDS

Input: A graph $G(V, E)$ of n vertices.
Output: MCD Set D' of a graph $G(V, E)$
1. $D' \leftarrow \phi$; $D \leftarrow \phi$
2. $\forall v \in V$
3. **while** $V \neq \phi$ **do**
4. $d(v) = |N(v)|$ and $s(v) = d(v) + \Sigma_{u \in N(v)} d_G(u)$
5. $u \leftarrow max_{v \in V} s(v)$ if multiple vertices with same maximum s(v) is found   then select one vertex randomly among them.
6. $D \leftarrow D \cup \{u\}$
7. $V \leftarrow V - N[u]$
8. **end while**
9. **while** D is connected **do**
10. $v \leftarrow max\{|N(V - D) \cap D|/u \in V - D\}$
11. $D \leftarrow D \cup \{u\}$
12. **end while**
13. (V, E, D) /* Local search procedure /*
14. **repeat** $\forall v \in G[D]$, update $d(v)$ and $s(v)$
15. **while** D' is connected **do**
16. $w \leftarrow min_{v \in D} s(v)$ if multiple vertices with the same miniimum s(v) is   found then select one vertex randomly among them.
17.   **if** $N[w] \subset \bigcup_{v \in D} N[v]$
18.   then $D' \leftarrow D - \{w\}$; $D \leftarrow D - \{w\}$
19.   **else**
20. $D' \leftarrow D \cup \{w\}$; $D \leftarrow D - \{w\}$
21. **end while**
22. **until** $D \neq \phi$
23. end

24. return updated D'

The algorithm operates as follows: The proposed algorithm proceeds in three phases. Initially the minimum connected dominating set D is empty. The £rst phase of the algorithm starts in the line 3 and end in the line 8. In this phase *degree* $d(v)$ and the *support* $s(v)$ of each vertex is calculated. With the help of the parameter *support* of vertices a vertex with maximum *support* value is added in to the set D which would cover maximum number of previously uncovered vertices. If more than one vertex satis£es these criteria a random selection is executed to add a vertex in to the set D. The vertex added in to the set D and its neighboring nodes are removed from the vertex set V. This process is repeated until the vertex set of a graph is empty.

Lines 9 to 12 deliver the 2nd phase of the algorithm. In this phase, the dominating set found in the £rst phase get connected using Steiner tree. It is a tree, interconnecting all the nodes in dominating set by adding new nodes between them. The nodes that are in the Steiner tree but not in the dominating set are called Steiner nodes. In this phase a node in $V - D$ is connected to the maximum number of nodes in D, identi£ed as a Steiner node. Set of connector nodes with the given D could be found using Steiner tree. A connected dominating set D is constructed using these Steiner nodes. It is to be noted that in a MCDS the number of Steiner nodes should be a minimum.

Third phase of the algorithm explained in the lines 13-20 is a pruning technique. In this phase we apply a local search procedure. This variant was inspired by the local search procedure used in the GRASP program for maximum clique designed by Feo et al. [9]. The number of iterations required for the local search proposed in [9] is very large. Therefore, instead of using these rules following procedure is applied for the local search.

An exhaustive search procedure is applied to determine whether it is possible to remove any two vertices from D, and replace them with one or no vertices, while still remaining a dominating set. Select a vertex having minimum *support* value from the connected dominating set D. If one or more vertices have the same criteria, a random selection is made among them. Then the pruning technique takes the following steps to minimize the number of vertices in the MCDS. If the vertex set of all closed neighbors of selected minimum *support* value vertex, is a subset of union of all closed neighbors of the elements of D then delete the corresponding minimum *support* value vertex from the CDS D. Otherwise keep the minimum *support* value vertex in the set D' for the MCDS but not to be considered in the next iterations. i.e., D is updated by deleting the corresponding minimum *support* value vertex. Whenever we succeed in decreasing the size of D in this manner, the local search procedure is called once again in the lines 14 and 22 on the smaller set D.

Theorem 5.1: The proposed SLS algorithm returns a connected dominating set and has the time complexity of $O(mn)$.
Proof: Let $G(V, E)$ be an undirected connected graph. By the algorithm procedure, it is clear that £rst phase of the algorithm produces dominating set and the second phase of the

World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:6, No:8, 2012

algorithm connecting the dominating set using Steiner nodes and produces a connected dominating set D. But in the third phase, the algorithm removes some more vertices from D and produces another CDS D'. Therefore it is enough to prove that D' is still a CDS.

To remove or maintain a vertex from D in the third phase, at each iteration a vertex with minimum *support* value $w \in D$ is selected. If these selected vertices fails to satisfy the condition in step 17, the proposed algorithm returns as $D' = D$, in this case D' is a CDS.

Suppose the selected vertex w satisfes the condition in step 17 i.e., if the closed neighbourhood elements of w is a subset of union of all closed neighbourhood elements of $\forall v \in D - \{w\}$, we can find atleast one adjacent vertex in $D' = D - \{w\}$ neighbour to all the closed neighbouring elements of w. This implies that D' is a connected dominating set.

To determine the time complexity of the algorithm, note that in the first phase while loop executed at most n times. Each step of the second phase determine whether adding one or no vertices connects the graph or not. To do this we need $O(m(n-d))$ time where d is the size of a dominating set. The third phase of the algorithm executed at most d times. Thus, the total time complexity of algorithm is given by $O(mn)$.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

In order to evaluate the performance of SLS algorithm, for the MCDS problem, extensive computational experiments were carried out on two groups of simulation identifed below. All the procedures of SLS have been coded in C++ language. The experiments were carried out on a PC of Intel Pentium Core2 Duo processor with 1.6 GHz CPU and 1 GB of RAM.

The simulation is performed using the following parameters. v represents the number of mobile hosts in the network (number of nodes), γ represents cardinality of MCDS in the network and r represents the radius of the mobile host's transmission area. In the computational experiments, the testing instances were created randomly in a 100×100 square units of 2-D simulation area, by throwing a certain number of mobile hosts (nodes). i.e., a unit disk graph corresponding to sensor networks was constructed. An adjacency matrix is used for representing the unit disk graph. Any pair of nodes have link connection if their distance is less than radius r. Degree of each node was decided by a random number generator i.e., for each row in a matrix, a random number is generated for deciding the number of ones in that row, for each column of a row another random number is generated. If the random number for a column is less than the random number for that row, a one is placed at that particular column in corresponding row otherwise a zero is placed. If generated graph is disconnected, simply discard the graph. Otherwise continue the simulation. To benchmark the proposed SLS algorithm, it is compared with the algorithms proposed in [14, 27]. They are referred in the results (Fig. 3 and Fig. 4) as NH and CCH respectively.

Experiments were performed in two groups of simulation. In the first group, the set up of mobile host's transmission range r has four different values: 10, 30, 50 and 70. In this way, we can control the density of generated graphs, since
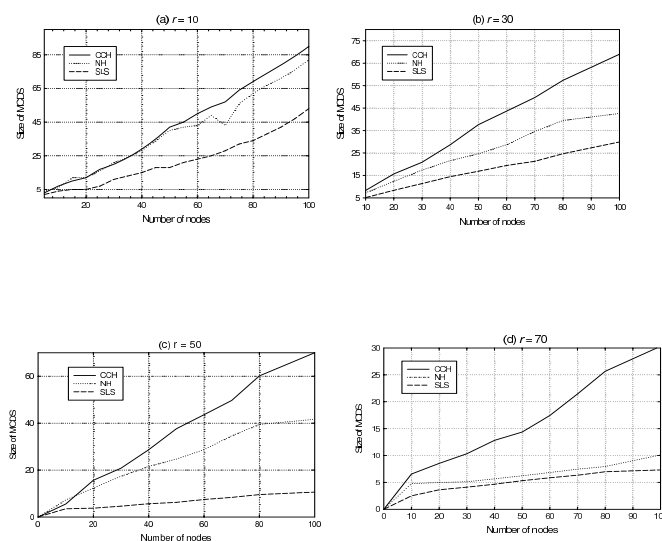

Fig. 3. Average size of MCDS relative to the nodes v

the density of generated graphs increases as r increases. For each r, the number of nodes v varied from 0 to 100. For each v, a random connected graph generated 10 times. For each parameter set, the average value of MCDS returned by each algorithm over 10 different random graphs of that type have been taken. In the second group, the number of nodes (mobile hosts) in the network set to four different values: 100, 200, 300 and 400. For each node 100 and 200, the radius of the transmission range r varied from 10 different values of 20 to 65 in the interval difference value of five. For each node 300 and 400, r varied from 7 different values of 50 to 80 in the interval difference value of five. For each r, random connected graphs generated 10 times and the average size of a MCDS found by each algorithm over 10 different random graphs of that type have been taken.

Fig. 3 (a), (b), (c), (d) show the average size of the MCDS versus the number of nodes in the network for the increasing order of radius r. These figures show that the performance of the proposed algorithm much better than the recently proposed heuristics [14, 27] when the radius of the transmission range is either too small or too large. When r increases the proposed approach outperformed other two compared algorithms. In order to assess the change in the relative performance between the approaches, the second groups of experiments were conducted.

Fig. 4 (a), (b), (c), (d) show the average size of the MCDS with respect to radius r for the increasing order of number of nodes v. We can see that the prposed approach outperforms the compared heuristics when sizes of the nodes are small or large. Proposed heuristic get coincide with algorithm proposed in [14] when the number of nodes and transmission range are small. For large values of r and v, the proposed approach SLS performs very well than other approaches.
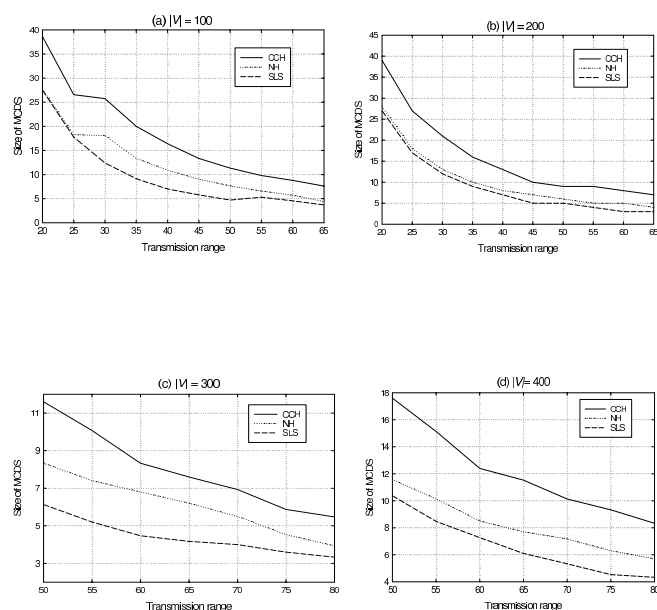
World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:6, No:8, 2012

Fig. 4. Average size of MCDS relative to the radius r

VII. CONCLUSIONS

In this paper, a new approach proposed for the minimum connected dominating set problem. The proposed heuristic algorithm applied to ad hoc wireless networks, which are modeled as unit disk graphs. The proposed SLS algorithm ef£ciently £nds the MCDS of a graph with the help of the parameter *support* of vertices. The model of the proposed heuristic is very simple and also easy to implement. An experimental study has been conducted to study the performance of the proposed algorithm with algorithms those proposed in proposed in in [14] and [27] in terms of the size of MCDS. It shows that the proposed heuristic outperformed compared algorithms from small to large values of r and v . The above mentioned advantages and the simplicity of the proposed heuristic make it an attractive alternative when solving the MCDS problem in dynamic environments.

REFERENCES

[1] K. M. Alzoubi, P -J. Wan, O. Frieder: Distrbuted heuristics for connected dominating set in ad hoc wireless networks. IEEE ComSoc/KICS Journal on communication networks, 4(1) (2002) 22-29.
[2] D. Applegate, W. Cook, A. Rohe, Chained Lin-Kernighan for large traveling salesman problems, Informs Journal of Computing, 15(1) (2003) 82-92.
[3] B. S. Baker: Approximation algorithms for NP-complete problems on planar graphs. Journal of the ACM, 41(1), (1994) 153-180.
[4] S. Butenko, X. Cheng, D.-Z. Du, P. M. Pardalos: On the construction of virtual backbone for ad hoc wireless networks, In S. Butenko, R. Murphey and P. M. Pardalos, editors, Cooperative control: Models, Application and Algorithms, Kluwer Academic publishers, 43-54 (2004).
[5] S. Butenko, X. Cheng, C. A. S. Oliveira, P. M. Pardalos: A new heuristics for the minimum connected dominating set problem on ad hoc wireless networks, In S. Butenko, R. Murphey and P. M. Pardalos, editors, Recent developments in cooperative control and optimization, Kluwer Academic publishers, 61-73 (2004).
[6] S. Butenko, S. K-Anderoglu, O. Ursulenko: On connected domination in unit ball graphs, Optimization Letters, 5(2) (2011) 195-205.
[7] D. Chen, X. Mao, X. Fei, K. Xing, F. Liu, M. Song: A convex -hull based algorithm to connect the maximal independent set in unit disk graphs, Wireless algorithms: systems and applications (2006) 36-370.
[8] F. Dai, J. Wu: On constructing k-connected k-dominating set in wireless ad hoc and sensor networks, Journal of parallel and distributed computing, 66 (2006) 947-958.
[9] B. Das, V. Bharghavan: Routing in adhoc networks using minimum connected dominating sets. In International conference on communications, 376-380 (1997).
[10] T. S. Feo, M. G. C. Resende, S. H. Smith: A greedy randomized adaptive search procedure for maximum independent set, Operations Research, 42(5) (1994) 860-978.
[11] X. Gao, W. Wang, Z. Zhang, S. Zhu, W. Wu: A PTAS for minimum d-hop connected dominating set in growth bounded graphs, Optimization Letters, 4 (2010) 321-333.
[12] M. R. Garey, D. S. Johnson: Computers and Intractability: A Guide to the theory NP - completeness, San Francisco: Freeman (1979).
[13] S. Guha and S. Khuller: Approximation algorithms for connected dominating sets, Algorithmica, 20(4) (1998) 374-387.
[14] B. Han: Zone-based virtual backbone formation in wireless ad hoc networks, Ad Hoc Networks, 7 (2009) 183-200.
[15] T. W. Haynes, S. T. Hedetniemi, P. J. Slater: Fundamentals of domination in graphs, New Slater: Fundamentals of domination in graphs, New ork, Marcel Dekker Inc., (1998).
[16] K. Islam, S. G. Akl, H. Meijr: A constant factor localized algorithm for computing connected dominatng sets in wireless sensor networks, Proc. of the 14th IEEE international conference on parallel and distributed systems (2008).
[17] K. Katayama, H. Narihisa, Iterated local search approach using genetic transformation to the traveling salesman problem, Proceedings of the Genetic and Evolutionary Computation Conference, (1999) 321-328
[18] K. Katayama, A. Hamamoto, H. Nirihisa, An effective local search for the maximum clique problem, Information Processing Letters, 95 (2005) 503-511.
[19] B. W. Kernighan, S. Lin, An ef£cient heuristic procedure for partitioning graphs, Bell System Techn. J. 49 (1970) 291-307.
[20] Y. Li, S. Zhu, M. Thai, D. Du: Localized construction of connected dominating set in wireless networks, In Proc. YAWN (2004)
[21] S. Lin, B. W. Kernighan, An effective heuristic algorithm for traveling salesman problem, Oper. Res. 21 (1973) 498-516.
[22] G. Lu, M. T. Zhou, Y. Tang, M. Y. Zhao, X. Z. Niu, K. She: Approximation algorithms for the connected dominating set problem in unit disk graphs, Journal of electronic science and technology of china, 7(3) (2009) 214-222.
[23] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, D. J. Rosenkrantz: Simple heuristics for unit disk graphs, Networks, 25 (1995) 59-68.
[24] P. Merz, B. Freisleben, Fitness landscapes, memetic algorithms and greedy operators for graph partitioning, Evolutionary Computing, 8(1) (2000) 61-91.
[25] P. Merz, B. Freisleben, Memetic algorithms for the traveling salesman problem, Complex Systems, 13(4) (2001) 297-345.
[26] P. Merz, K. Katayama, Memetic algorithms for the unconstrained binary quadratic programming problem, Biosystems, 78(1-3) (2004) 99-118.
[27] R. Misra, C. Mandal: Minimum connected dominating set using a collaborative cover heuristic for ad hoc sensor networks, IEEE transactions on parallel and distributed systems, 21(3) (2010) 292-302.
[28] W. Pullan, Phased local search for the maximum clique problem, J. Comb. Optim. 12 (2006) 303-323.
[29] W. Pullan, Approximating the maximum vertex/edge weighted clique using local search, J. Heuristics, 14 (2008) 117-134.
[30] K. Sakai, S. C-H. Huang, W-S. Ku, M-T. Sun, X. Chang: Timer-based CDS construction in wireless Ad Hoc networks, IEEE Transactions on Mobile Computing, 10(10) (2011) 1388-1402.
[31] L. A. Sanchis: Experimental analysis of heuristic algorithms for the dominating set problem, Algorithmica, 33 (2002) 59-68.
[32] I. Stojmenovic, M. Seddigh, J. Zunic: Dominating sets and neighbor elimination based routing algorithms in wireless networks. IEEE Transaction Parallel distributed systems, 13(1) (2002) 14-25.
[33] P -J. Wan, K. M. Alzoubi, O. Frieder: Distributed construction of connected dominating set in ad hoc wireless networks. In Proc. INFOCOM (2002).
[34] W. Wu, X. Gao, P. M. Pardalos, D -Z. Du: Wireless networking, dominating and packing, Optimization Letters, 4 (2010) 347-358.
[35] J. Wu, H. Li: On calculating connected dominating set for ef£cient routing in ad hoc wireless networks, In proceedings of the 3rd ACM international workshop on Discrete Algorithms and methods for Mobile computing and communications, 7-14 (1999).

World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:6, No:8, 2012

[36] M. Yaguira, T. Yamaguchi, T. Ibaraki, A variable depth search algorithm for the generalized assignment problem, in S. Voss, S. Martello, I. H. Osman, C. Roucairol(Eds.), Meta-Heuristics: Advance and Trends in Local Search Paradigms for Optimization, Kluwer, Dordrecht, (1999) 459-471.

[37] Y. Yang, M. Cardei: Adaptive energy ef£cient sensor scheduling for wireless sensor networks, Optimization Letters, 4 (2010) 359-369.

[38] J. Yu, N. Wang, G. Wang: Heuristic algorithms for constructing connected dominating sets with minimum size and bounded diameter in wireless networks, LNCS 6221 (2010) 11-20.

[39] W. Zhang, I. Shin, F. Zou, W. Wu, M. T. Thai: Construction of virtual backbone with multiple factors constraints in wireless ad-hoc network, Ad-Hoc and sensor wireless networks, 10(1) (2010) 27-59.