



LUND UNIVERSITY

Adaptive Scatter Search to Solve the Minimum Connected Dominating Set Problem for Efficient Management of Wireless Networks

Abdulaziz, Shada; Hedar, Abdel-Rahman; Sewisy, Adel; El-Sayed, Gamal

Published in:
Algorithms

DOI:
[10.3390/a13020035](https://doi.org/10.3390/a13020035)

2020

Document Version:
Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):
Abdulaziz, S., Hedar, A-R., Sewisy, A., & El-Sayed, G. (2020). Adaptive Scatter Search to Solve the Minimum Connected Dominating Set Problem for Efficient Management of Wireless Networks. *Algorithms*, 13(2). <https://doi.org/10.3390/a13020035>

Total number of authors:
4

General rights

Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Article

Adaptive Scatter Search to Solve the Minimum Connected Dominating Set Problem for Efficient Management of Wireless Networks

Abdel-Rahman Hedar ^{1,2,*}, Shada N. Abdulaziz ^{3,4}, Adel A. Sewisy ² and Gamal A. El-Sayed ^{1,5}

¹ Department of Computer Science in Jamoum, Umm Al-Qura University, 25371 Makkah, Saudi Arabia; gaelsayed@uqu.edu.sa

² Department of Computer Science, Faculty of Computers & Information, Assiut University, 71526 Assiut, Egypt; sewisy@aun.edu.eg

³ Center for Middle Eastern Studies, Lund University, SE-22100 Lund, Sweden; shada.abdulaziz@cme.lu.se

⁴ Department of Mathematics, Faculty of Science, Assiut University, 71516 Assiut, Egypt

⁵ Electrical Engineering Department, Assiut University, 71516 Assiut, Egypt; gamal@eng.au.edu.eg

* Correspondence: ahahmed@uqu.edu.sa or hedar@aun.edu.eg; Tel.: +966-55-0086-411 or +20-10-0070-4940

Received: 28 December 2019; Accepted: 3 February 2020; Published: 4 February 2020

Abstract: An efficient routing using a virtual backbone (VB) network is one of the most significant improvements in the wireless sensor network (WSN). One promising method for selecting this subset of network nodes is by finding the minimum connected dominating set (MCDS), where the searching space for finding a route is restricted to nodes in this MCDS. Thus, finding MCDS in a WSN provides a flexible low-cost solution for the problem of event monitoring, particularly in places with limited or dangerous access to humans as is the case for most WSN deployments. In this paper, we proposed an adaptive scatter search (ASS-MCDS) algorithm that finds the near-optimal solution to this problem. The proposed method invokes a composite fitness function that aims to maximize the solution coverness and connectivity and minimize its cardinality. Moreover, the ASS-MCDS methods modified the scatter search framework through new local search and solution update procedures that maintain the search objectives. We tested the performance of our proposed algorithm using different benchmark-test-graph sets available in the literature. Experiments results show that our proposed algorithm gave good results in terms of solution quality.

Keywords: dominating sets; minimum connected dominating sets; scatter search; local search; wireless network design

1. Introduction

One of the most important variations of the dominating set (DS) problem is its connected version, which is a subset of connected nodes (vertices), where each node in the graph is in the connected dominating set, or adjacent to some nodes in the connected dominating set. Recently, the minimum connected dominating set (MCDS) problem has obtained wide popularity due to its applications in wireless sensor networks (WSN) and mobile ad hoc networks. Generally, in real-life problems that can be represented as a MCDS problem, it is not necessary to acquire the optimal solution, as near-optimal solutions are adequate in most applications [1]. Unlike traditional computer networks, there is no pre-defined fixed infrastructure as a hierarchical structure for WSNs. Consequently, it is difficult to achieve routing scalability and efficiency [2]. Using a connected dominating set (CDS) to form a virtual backbone (VB) in WSNs is a promising technique to enhance the performance and improve the efficiency of routing protocols. All network nodes are dominated by the set of nodes in CDS, and packets can be forwarded from the source to the destination node through the CDS. Thus, using CDS

as VB reduces the average message load of a WSN and improves routing performance. In addition to that, the routing algorithm adapts quickly to any changes in network topology [3]. The improvement of routing performance in wireless networks results in energy saving and interference reduction as well as flooding restriction in the network. Such improvements require minimizing the size of CDS as possible. Finding a minimum CDS (MCDS) is usually an NP-hard problem and, in general, cannot be solved exactly in polynomial time [4].

Recently, meta-heuristics have become among the most successful computational methods to find good solutions for real-life optimization problems in many fields such as science, engineering, economics, and business, by exploring the solution search space of these large problems. Those algorithms reduce the effective size of the space and explore that space efficiently [5]. In fact, meta-heuristics are classified into three different categories. The first includes point-to-point techniques such as simulated annealing and tabu search, the second includes population-based techniques such as genetic algorithms and scatter search, and the third includes hybrid techniques that combine techniques from the first two categories such as memetic algorithms [6].

Scatter search (SS) is one of the most popular meta-heuristic algorithms based on the population-based technique that has effectively been applied to many hard optimization problems. The foundation, basic concepts, and principles of that technique were initially proposed in the 1970s [5]. The foundations of scatter search derive from previous strategies for combining decision rules and constraints. Usually, combining elements yields solutions better than one based only on original elements. Unlike genetic algorithms, scatter search is based on the hypothesis that systematic designs and methods create new solutions more than those derived from randomization. Furthermore, scatter search uses strategies for search diversification and intensification that have effectively been applied in varied optimization problems. Actually, the SS method is very effective and flexible as it is not restricted to a single uniform design and all SS elements can be implemented in many ways and varied degrees of sophistication [7].

In this paper, we propose an adaptive scatter search algorithm for finding the minimum connected dominating set, which is abbreviated to ASS-MCDS. It uses an on/off variable representation of solutions in searching for the MCDS and uses a composite fitness function to measure the quality of the solution. To measure the quality of each solution, the fitness function takes into account the size of the domination set as well as the connectivity between the nodes in the graph. Local search is the intensification search method that is used by ASS-MCDS to improve trail solutions. By applying the proposed method in a wireless network, we can derive the skeleton of a virtual backbone network to improve routing efficiency. This skeleton can be used to easily manage and control the network topology when a node is added to or removed from the network. We compare the results of the proposed method with some other standard methods in the literature. Several instances of the MCDS problem are used to test the performance and effectiveness of the proposed method.

The rest of this paper is organized as follows. In Section 2, an overview of the MCDS problem is briefly presented. Section 3 describes the related works on the MCDS problem. Section 4 presents the solution representation, design of a new evaluation function for the MCDS problem, and presents the main components of the ASS-MCDS in addition to its formal algorithm. The implementation of the proposed method, its initial and control parameters setting, as well as its comparisons against some benchmark methods are illustrated in Section 5. Finally, the conclusion makes up Section 6.

2. Minimum Connected Dominating Set Problem

Given a simple undirected graph $G = (V, E)$, where $V(G)$ is the set of vertices (or nodes), these nodes connect by the set of edges $E(G)$. A path P from vertex u to vertex v is an alternating consecutive sequence of vertices and edges that connects vertices $u, v \in V(G)$ such that no vertex is repeated in the path from u to v , where the number of edges in that path represent the length of the path P . In graph G every vertex v is said to dominate itself and any vertex adjacent to it. A dominating set is a set of

vertices that collectively dominates all other vertices in the graph G [8]. The dominating set D in a connected graph must satisfy two conditions:

- A path from any vertex in D to another vertex in D stays entirely within D . Thus, D represents a connected subgraph of G ;
- Every vertex in G is either in D or adjacent to a vertex in D .

The MCDS problem seeks to find a connected dominating set of minimum cardinality, which is also called the domination number of G and written as $\gamma(G)$.

The MCDS problem is one of the classical MDS problem variations, which is classified as a NP-hard problem. Thus, it is a hard combinatorial problem and cannot exactly be solved in polynomial time. Therefore, acquiring an effective solution for the minimum connected dominating set problem is one of the significant research areas in graph theory [1]. Recently, several researches are moving towards the MCDS problem because of its various applications in wireless networks, facility locations, and operational research in computer science [9–12].

The concept of the MCDS problem is significant and useful in wireless networks, particularly for networks with very large scales like a WSN. In such networks, the minimum CDS works as a virtual backbone in the network and dominates all the nodes in the network. Consequently, it is easy to forward the packets from the source node to the destination node through the VB nodes [2].

3. Related Works

The CDS and its variant MCDS are among the fundamental covering problems in graph theory related to various wireless network problems such as network monitoring, network control, and resource allocation. As this problem has been shown to be NP-hard in [13], there exists no known polynomial-time algorithm for exactly identifying the connected domination number of any arbitrary graph. Consequently, the design of an effective approximation algorithms has become an important issue for the study of CDSs. Guha and Khuller [14] proposed two polynomial-time algorithms to solve the MCDS problem for a general graph with these algorithms being greedy and centralized. Ruan et al. [15] presented a new approximation algorithm, which is a one-step greedy algorithm, with a performance ratio $\ln \delta + 2$, where δ is the maximum degree in the input graph. Wu et al. [16] introduced a distributed CDS construction algorithm that relies on the relation between the maximal independent set (MIS) and the connected dominating set.

For many years, researchers focused on developing heuristics to solve the MCDS problem in artificial intelligence and graph theory. These heuristic methods can be classified into two categories. The first category aims to find the maximal independent set of disconnected nodes that can be connected by a steiner tree or minimum spanning tree [17]. The second category aims to evolve a CDS by growing a small trivial CDS [18,19]. A maximal independent set is a dominating set, where the nodes in the MIS are non-adjacent and no node can be added to this set to maintain the independence of the nodes. Constructing a CDS depends on connecting the nodes in the MIS through some nodes not in the MIS [16,20].

In recent years, developing meta-heuristics to construct MCDS has attracted many researchers. A greedy randomized adaptive search procedure for solving the MCDS problem is designed by Li et al. [21]. In [1], authors proposed ant colony optimization algorithms for the MCDS. For the minimum weighted connected dominating set problem a hybrid population-based iterated greedy algorithm and a genetic algorithm are proposed by [22]. Two meta-heuristics, memetic, and simulated annealing with intensification search methods are presented in [11,12] to solve the MCDS problem for wireless networks design and management.

4. Methodology

In this section, the components of the proposed method are described. First, the representation of solutions and the fitness function are explained. Then, the scatter search operators, diversification

generation, improvement, reference set update, subset generation, and solution combination are defined. Moreover, the main scatter search modifications and local search are demonstrated. Finally, the formal algorithm of ASS-MCDS method is presented at the end of this section.

4.1. Solution Representation and Evaluation

In this section, the solution representation in the ASS-MCDS method is described. Then, to evaluate the solution quality for the MCDS problem a composite fitness function fit is presented.

4.1.1. Solution Representation

The ASS-MCDS method uses an on/off variable representation of solutions. Specifically, a trial solution X will be represented as a bit vector $(x_1, x_2, \dots, x_{V_n})$, where V_n is the number of nodes in the graph. The subscript/index values $1, 2, \dots, V_n$, refer to the corresponding nodes in the graph vertices set V . If a component x_i of X has the value 0, then the node represented by the i -th column in V_n is not contained in the node subset represented by the solution X . Otherwise, the solution X contains the i -th node.

4.1.2. Solution Evaluation

A fitness function $fit(\cdot)$ is a particular type of an objective function that is used to measure the quality of the solution that increases the chance of finding good solutions and reaches higher coverage for the graph. Thus, there is a direct relationship between the fitness function and the solution quality, so better solutions have a higher fitness function value than worse ones. The fitness function is used to measure the solution effectively.

$$fit(x) = \alpha \frac{V_D}{V_n} + \beta \frac{C_n}{\sum_i^{V_n} x_i} + \gamma \frac{V_n - \sum_i^{V_n} x_i}{V_n}, \quad (1)$$

where V_D represents number of vertices dominated by the subset of vertices D in the solution x and V_n is the total number of vertices in the graph, C_n is the number of vertices in largest connected set in the solution x , and α , β and γ are weights with $\alpha + \beta + \gamma = 1$. The fitness function fit consists of three parts. The first part V_D/V_n , reflects the size of domination on G by x . If x represents a dominating set, then this part is equal to 1. The second part, $C_n/\sum_i^{V_n} x_i$, reflects the connectivity between the vertices in x . This part is equal to 1 if x represents a connected set. While the third part, $(V_n - \sum_i^{V_n} x_i)/V_n$, distinguishes between two solutions that are equal in the first part and the second part according to the percentage of vertices which are not contained in each of them such that better solutions have a higher percentage than worse ones.

4.2. Adaptive Scatter Search Algorithm for the MCDS Problem

In this section, an adaptive scatter search method is designed to solve the considered problem and is called ASS-MCDS shortly. The ASS-MCDS algorithm begins with a well-distributed random population of individuals (solutions). Then, the fitness function which is previously defined is repeatedly invoked to evaluate the fitness of the initial solutions and rank them. Five methods represent the well-known SS template: Diversification generation, improvement, reference set update, subset generation, and the solution combination method. Unlike other evolutionary methods such as genetic algorithms, SS depends on using a small population, known as the reference set, whose solutions are combined to create new solutions that are acquired in a systematic way. These new solutions can be effectively improved by applying a local search method before terminating the search process. The following subsections describe the five methods of the SS method then, the algorithm is formally presented.

4.2.1. Scatter Search Method

In this section, the basic design to apply SS depending on its five well-known methods is presented. The fundamental feature of SS is related to the flexibility in implementing these five methods, where these mechanisms do not have a single uniform design. This expands the exploration strategies that may improve the efficiency of the solutions in a particular application. The implementation of SS consists of the following five methods:

- The diversification generation method builds a large set population of diverse solutions;
- The improvement method transforms a trial solution into another with higher quality. If there is no improvement to the input trial solution results, then the enhanced solution is considered to be the same as the input solution [23];
- The reference set (*RefSet*) update method is a collection of both high quality solutions and diverse solutions. The best solutions in the population P are added to *RefSet* and then deleted from P . For each solution in $(P - \text{RefSet})$, the minimum of distances to the solutions in *RefSet* is computed. Then, the solution with the maximum of these minimum distances is selected and added to *RefSet* [5];
- The generation method operates on the reference set to produce all pairs of reference solutions. That is, the method would focus on subsets of size 2 resulting $(b^2 - b)/2$ subsets of solutions, where b is the size of *RefSet*;
- The solution combination method generates members of the new population. The reference set update method is applied once again. The updated reference set consists of the best solutions in $\text{RefSet} \cup P$. The algorithm is terminated after submitting all subsets that are generated within the current iteration to the combination method, and *RefSet* does not admit the improved trial solutions under the rules of the reference set update method [5,24].

4.2.2. Local Search

The local search mechanism (LS) is used to improve the solutions subsets by adding or deleting some vertices in the solution. This process is repeated n_{step} times. The formal description of the LS mechanism is stated in Algorithm 1.

Algorithm 1 Local Search

1. Repeat the following Steps (2-6) n_{step} times.
 2. Set $x^{best} = x$.
 3. If $V_D/V_n = 1$, randomly select a component x_i^{best} with value 1. This selection is inversely proportional to the degree of its corresponding vertex. Set $x_i^{best} = 1 - x_i^{best}$.
 4. If $V_D/V_n < 1$, randomly select a component x_i^{best} with value 0. This selection is proportional to the degree of its corresponding vertex. $x_i^{best} = 1 - x_i^{best}$.
 5. If $fit(x^{best}) > fit(x)$, set $x = x^{best}$.
-

It is worthwhile mentioning that the decision of adding or deleting nodes in local search explained by Algorithm 1 is related to the first part of the fitness function. However, this directly affects the other two parts of the fitness function. On the one hand, adding a node increases the domination and may increase the connectivity of the nodes in the solution. On the other hand, deleting a node minimizes the cardinality of the solution. Specifically, Algorithm 1 works in two manners:

- It increases the coverness if the node set represented by the solution does not cover the whole graph as in Step 4 of Algorithm 1;
- It decreases the cardinality if the node set represented by the solution covers the whole graph as in Step 3 of Algorithm 1.

4.2.3. ASS-MCDS Algorithm

The ASS-MCDS algorithm starts with a population P that contains P_{size} diverse trial solutions, which are generated by the diversification generation method using arbitrary trial solutions (or seed solutions) as inputs. During each generation, the quality of each solution in the population is evaluated by the fitness function fit as defined in Equation (1). The ASS-MCDS method applies the LS Algorithm 1 to improve the generated solutions. From these generated solutions, the $RefSet$ with size equal to Ref_{Size} is constructed of r_1 high-quality solutions, which represent 80% of the Ref_{Size} , and r_2 diverse solutions, which represent the remaining 20%. Thus, the size of the reference set is denoted by $Ref_{Size} (= r_1 + r_2)$. The construction of the initial reference set starts with the selection of the best r_1 solutions from P . These solutions are added to $RefSet$ and then deleted from P . The minimum of the distances for each solution in $P - RefSet$ to the solutions in $RefSet$ is computed. Then, the solution with the maximum of these minimum distances is added to $RefSet$ and deleted from P .

A subset generation method operates on the reference set $RefSet$, to generate all pairs of reference solutions. Thus, the method would focus on subsets of size 2 resulting in $(Ref_{Size}^2 - Ref_{Size})/2$ new subsets. The pairs in the new subsets are selected one at a time, so that the solution combination method is applied on each subset of solutions which is produced by the generation method to generate two trial solutions.

The local search (Algorithm 1) is used to update the current population. During the implementation of the local search method, the connected domination solutions in a pool P_s is maintained. The reference set update method is applied once again. The updated reference set consists of 80% of the best solutions in the reference set solutions and 20% of the connected domination solutions in the P_s . Trial solutions that are created as a combination of reference solutions are collected in a pool, denoted by T_{pool} . After the implementation of both the combination method and the improvement method, the T_{pool} is filled and the reference set is updated.

The ASS-MCDS algorithm terminates if none of the improved trial solutions are admitted to $RefSet$ under the rules of the reference set update method. The formal steps of the proposed method are stated in Algorithm 2 as well as Figure 1.

Algorithm 2 Adaptive Scatter Search (ASS-MCDS)

1. **Initialization.** Set values of P_{Size} and Ref_{Size} . Set P_s to be an empty set. Generate an initial population P of size P_{Size} .
 2. **Local Search.** Evaluate the fitness of all solutions in P by using Equation (1). Then, apply Algorithm 1 to improve the solutions in P .
 3. **Promising Selection Generation.** Construct P_s by adding all connected domination solutions in P .
 4. **Reference Set Update.** Construct $RefSet$ by adding the best and the maximally diverse solutions from P .
 5. **Subset Generation.** Generating all pairs of the reference set solutions.
 6. **Solution Combination.** Generate new trial solutions T_{pool} .
 7. **Local Search.** Improve the new solutions in T_{pool} using Algorithm 1.
 8. **Promising Selection Update.** Update P_s .
 9. **Reference Set Update.** Update the $RefSet$.
 10. **Stopping Condition.** Terminate the algorithm if there are no new solutions added to the $RefSet$. Otherwise, go to Step 5.
-

The MCDS problem is usually a NP-hard problem [4,13]. Therefore, there is no efficient algorithm to solve such a problem in its general form. The scatter search and meta-heuristics in general, are practical solvers and generally polynomial algorithms [25,26]. The proposed SS-based methods follow the main structures of standard SS with additive procedures which also have a polynomial order performance.

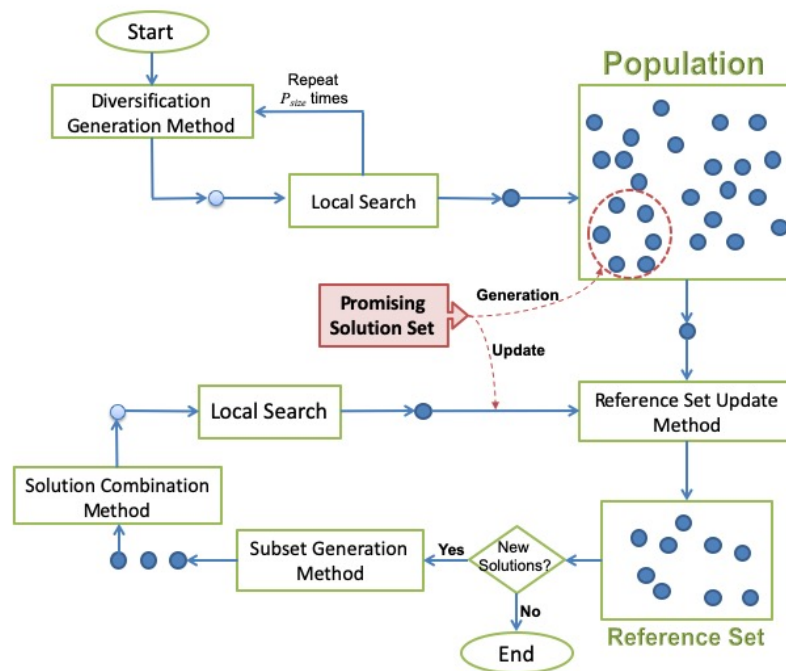


Figure 1. ASS-MCDS (adaptive scatter search) flowchart

5. Experimental Results

The ASS-MCDS method is programmed using MATLAB. In this experimental section, the effectiveness of ASS-MCDS method is analyzed on several test graphs from the literature [1,11,12,21]. These graphs are given as ad hoc wireless network clustering instances, which are described in Table 1. As shown in Table 1, eight different networks (instances) are used and all of these networks occupy the same area with different number of nodes (vertices) for each network starting from 80 nodes up to 400 nodes. The experiments were run 20 times for variant sizes of networks with the shown transmission ranges R .

Table 1. Ad hoc network instances

Network	Area ($L \times L$)	No. Vertices (V)	Range (R)
Net 1	400 × 400	80	120
Net 2	600 × 600	100	120
Net 3	700 × 700	200	120
Net 4	1000 × 1000	200	160
Net 5	1500 × 1500	250	160
Net 6	2000 × 2000	300	230
Net 7	2500 × 2500	350	230
Net 8	3000 × 3000	400	240

5.1. Parameter Setting and Tuning

Table 2 summarizes all parameters used in ASS-MCDS with their assigned values. These values are chosen according to our numerical experiments, or well-known common settings in the literature. In parameter tuning, we tried to find the best parameter values with moderate costs and good performance. The performance of the ASS-MCDS with different values of the parameters were studied to demonstrate the relationship between tuning parameter values and the network size. The following settings are tested to identify the best setting of the parameter values.

- The population size (P_{Size}) = 60, 80, 100, 120;

- The number of iterations in local search (n_{step}) = 4, 8, 10, 14;
- The reference set size (Ref_{Size}) = 10, 15, 20.

Table 2. Parameter setting

Parameter	Definition	Value
P_{Size}	Population size	$60 + (V_n/200) \times 20$
n_{step}	Number of iterations in local search	10
Ref_{Size}	Size of reference set	15

The best solutions vary according to the tuning parameter values which significantly affect solution qualities. In Figure 2, when the value of P_{Size} increases from 60 to 120, the results are improved significantly until a specific limit is reached where the quality of the results is not affected with the increment of the P_{Size} values. For moderate-size networks Net 1, Net 2, Net 3, and Net 4, the value 60 for P_{Size} is enough to acquire the best domination number for the networks. However, this value is not enough for Net 5 and Net 6, so it is increased to 80 to acquire the best domination number. This was also applied for Net 7 and Net 8 when the network became bigger, requiring an increment in P_{Size} values. Consequently, setting P_{Size} equal to 60 for small-size networks helped the proposed algorithm to obtain results with the same quality as those obtained by higher values of P_{Size} . Maintaining small values for this P_{Size} contributed to maintaining the computation cost and processing time. However, the P_{Size} value for large-size networks should gradually be increased according to the size of the network in order to maintain the solution qualities. According to these experimental results, the best value for P_{Size} could be calculated as the following:

$$P_{Size} = \begin{cases} 60, & \text{if } V_n \leq 200, \\ 60 + (20 \times V_n/200), & \text{if } V_n > 200. \end{cases} \quad (2)$$

where V_n represents the number of vertices in the graph.

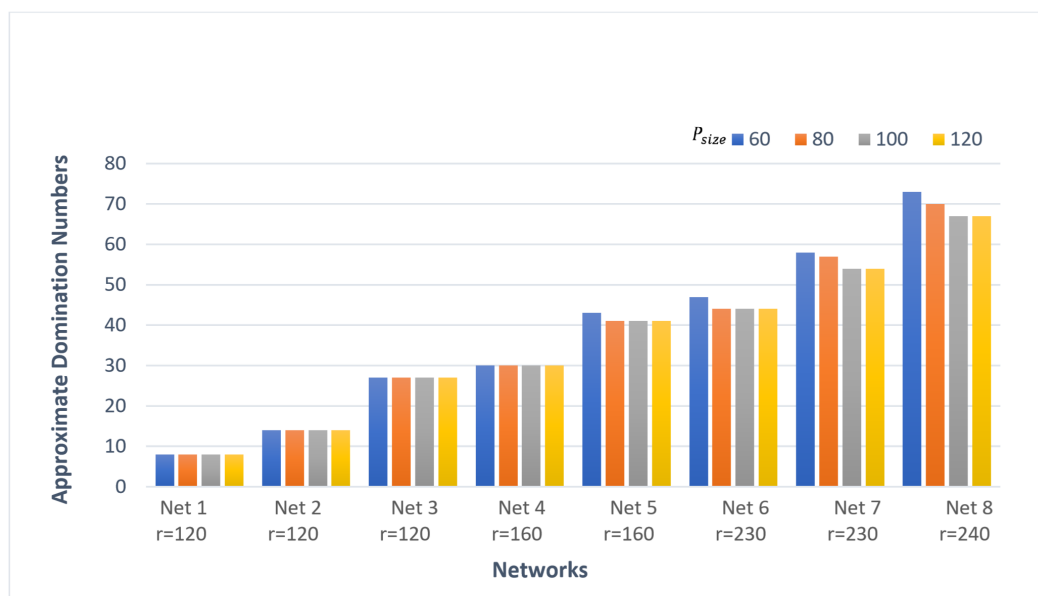


Figure 2. The performance of the ASS-MCDS with different P_{Size} values.

In Figure 3, when the value of n_{step} increases from 2 to 14 the results are significantly improved for all networks. As noted, the ASS-MCDS acquires the best domination number for all networks

when the number of iterations in local search n_{step} equals 10. In general, the experiment results proved that choosing the best parameter value increased the efficiency of ASS-MCDS for solving the MCDS problem.

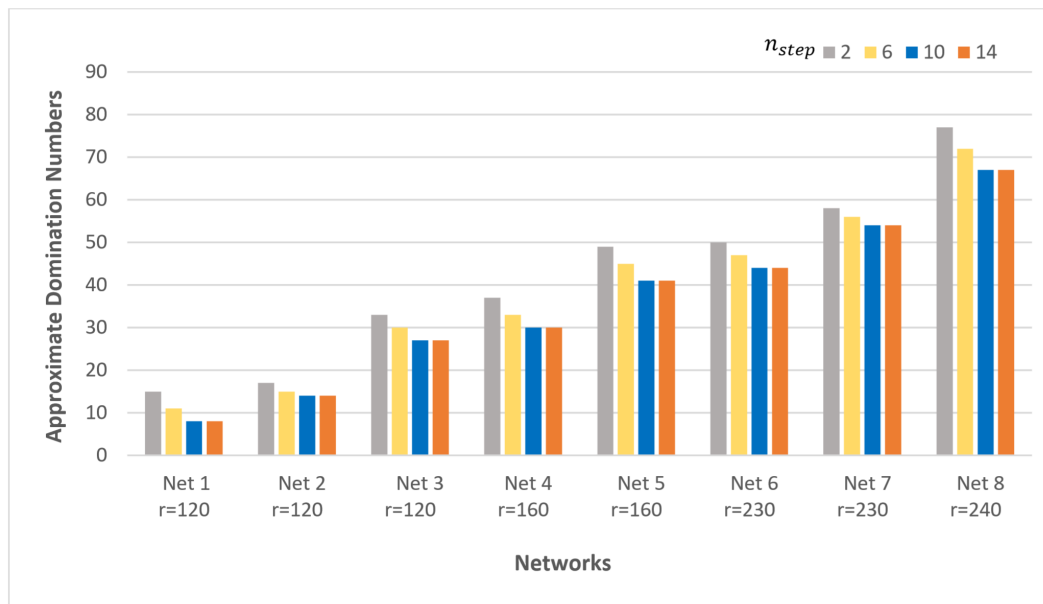


Figure 3. The performance of the ASS-MCDS with different n_{step} values

According to known common settings in the literature, the value of Ref_{Size} is typically small, e.g., no more than 20 [5]. Figure 4 shows how the performance of ASS-MCDS in acquiring the best domination is related to the size of the reference set Ref_{Size} . Thus, increasing the Ref_{Size} increases the ASS-MCDS performance. However, this resulted in more time consumption as summarized in Figure 5, where ASS-MCDS1, ASS-MCDS2, and ASS-MCDS3 are the ASS-MCDS with Ref_{Size} equal to 10, 15, and 20, respectively.

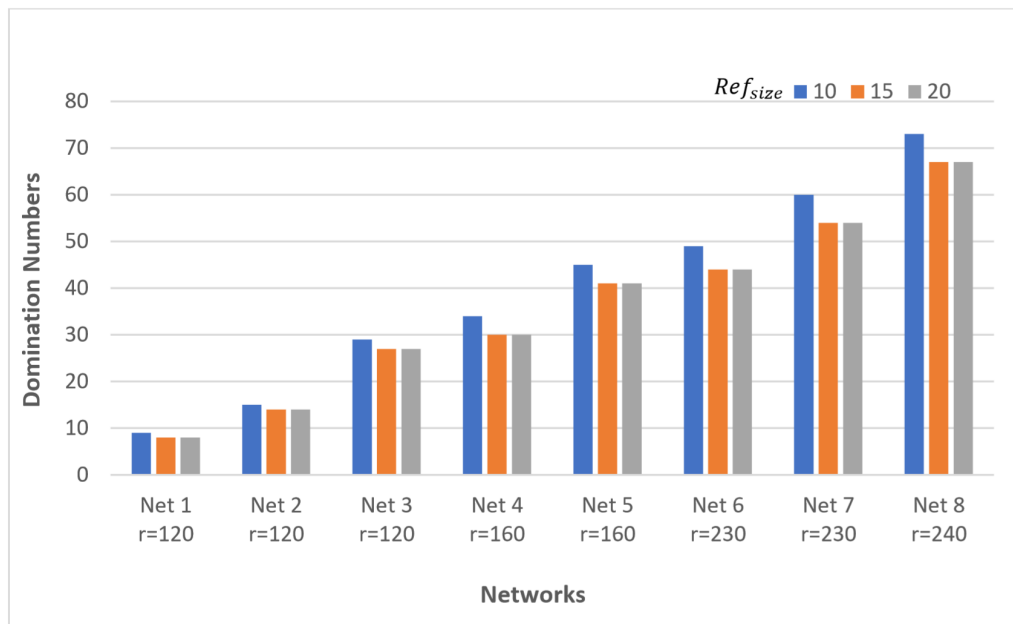


Figure 4. The performance of the ASS-MCDS with different Ref_{Size} values.

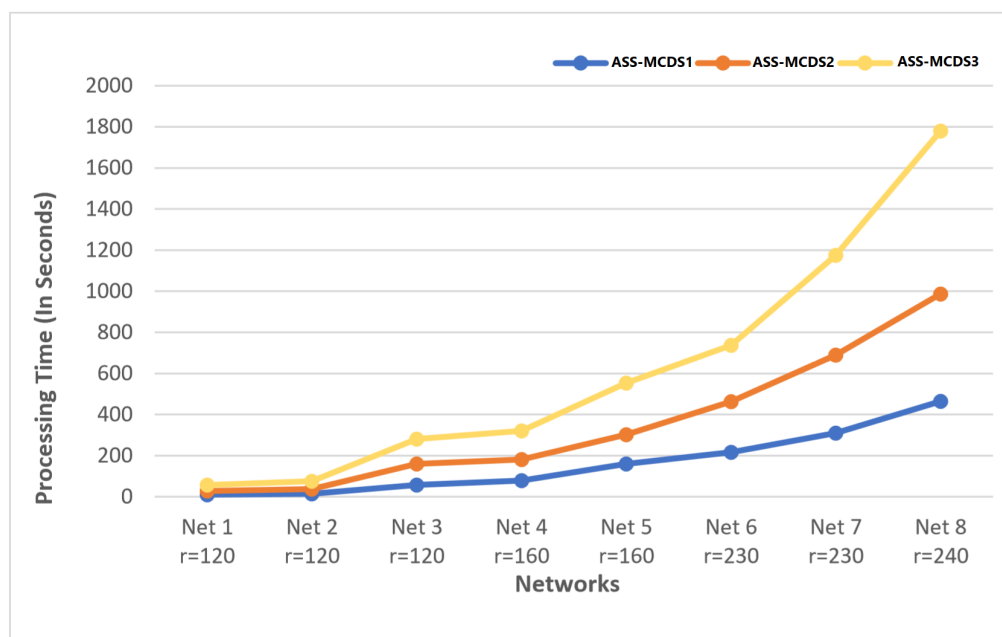


Figure 5. The average processing time of the ASS-MCDS with different Ref_{Size} values.

The performance of the fitness function weights α , β , and γ of Equation (1) was studied using several setting values. The obtained best settings of these parameters are the following values:

- **Setting 1:** $\alpha = 1/2$, $\beta = 1/4$, and $\gamma = 1/4$;
- **Setting 2:** $\alpha = 1/4$, $\beta = 1/2$, and $\gamma = 1/4$;
- **Setting 3:** $\alpha = 1/4$, $\beta = 1/4$, and $\gamma = 1/2$;
- **Setting 4:** $\alpha = 1/3$, $\beta = 1/3$, and $\gamma = 1/3$.

The performance of these different settings of the fitness function weights is shown in Figure 6 using four different networks. There is almost no major difference in results obtained by these values but the

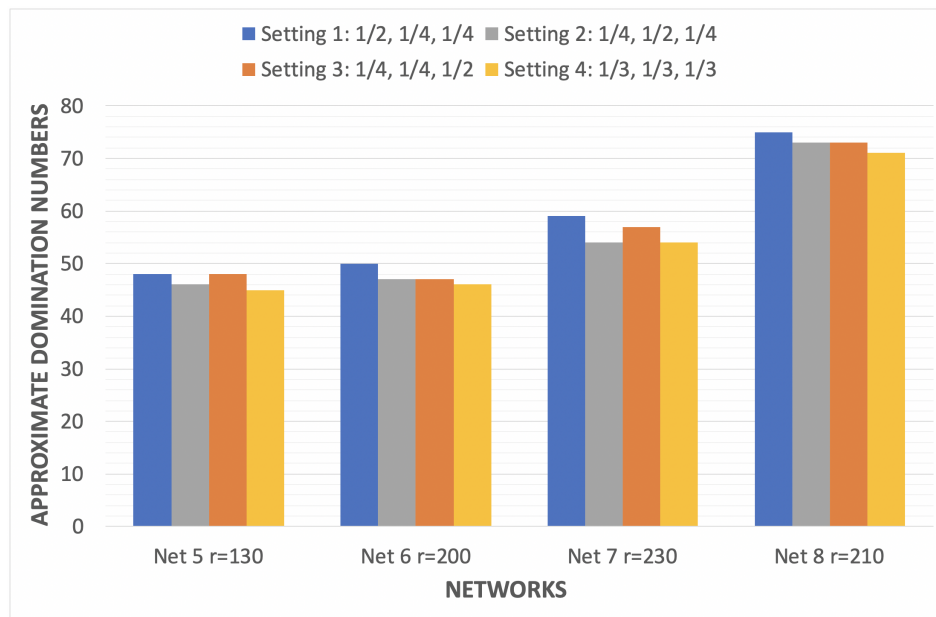


Figure 6. The performance of the different settings of the fitness function weights.

best one is the fourth setting. Using this best setting of equal weights, the fitness function defined by Equation (1) takes the following formula after adjusting its weight parameters.

$$fit(x) = \frac{1}{3} \left[\frac{V_D}{V_n} + \frac{C_n}{\sum_i^{V_n} x_i} + \frac{V_n - \sum_i^{V_n} x_i}{V_n} \right]. \tag{3}$$

For simplicity, the fitness function form can be equivalently rewritten as:

$$fit(x) = \frac{V_D}{V_n} + \frac{C_n}{\sum_i^{V_n} x_i} + \frac{V_n - \sum_i^{V_n} x_i}{V_n}. \tag{4}$$

5.2. Numerical Results

In this section, the algorithmic performance of the proposed method introduced in Algorithm 2 is investigated by comparing the results of ASS-MCDS against the results of other benchmark methods presented in [1,11,12,21]. To measure the performance of the ASS-MCDS, two different quantities are used to make comparisons:

- Minimum number (Min.): This measure gives the minimum connected domination number found in all independent runs and represents the minimum number of connected nodes in the best solution;
- Average number (Avg.): This measure calculates the average number of dominant connected nodes in best solutions found in independent runs.

The maximum number of iterations for the ASS-MCDS is 20 for each graph. The comparison experiment is set to test ASS-MCDS results with network instances described in Table 1. Table 3 displays the comparison results obtained by the ASS-MCDS against the three following methods:

- A greedy randomized adaptive search procedure [21];
- Ant colony optimization algorithms [1];
- The memetic algorithm [11,12].

Table 3. Network clustering comparison results.

Network ID	ACO+PCS		GRASP		MA-MCDS		ASS-MCDS	
	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.
<i>Net</i> _{1,60}	21.2	19	19.8	19	18.0	17	17.2	16
<i>Net</i> _{1,70}	16.2	15	15.1	14	15.3	15	14.5	14
<i>Net</i> _{1,80}	13.1	12	12.0	12	11.9	11	12.2	11
<i>Net</i> _{1,90}	11.6	11	10.6	10	11.7	11	11.8	11
<i>Net</i> _{1,100}	8.9	8	8.2	8	11.2	10	10.8	10
<i>Net</i> _{1,110}	8.5	8	7.8	7	8.4	8	8.7	8
<i>Net</i> _{1,120}	7.2	7	6.1	6	8.9	8	8.4	8
<i>Net</i> _{2,80}	23.6	22	22.9	22	15.5	14	15.4	14
<i>Net</i> _{2,90}	23.6	21	20.7	20	19.4	18	18.0	16
<i>Net</i> _{2,100}	19.0	17	17.9	17	19.4	18	18.4	18
<i>Net</i> _{2,110}	16.8	15	15.9	15	20.9	18	15.5	15
<i>Net</i> _{2,120}	15.5	14	13.8	13	14.6	14	15.2	14
<i>Net</i> _{3,70}	49.6	46	46.5	45	37.3	36	36.5	36
<i>Net</i> _{3,80}	43.9	41	37.5	35	37.4	35	36.4	36
<i>Net</i> _{3,90}	35.7	33	30.9	30	34.9	34	34.5	33
<i>Net</i> _{3,100}	31.0	23	25.8	25	31.4	29	30.8	28
<i>Net</i> _{3,110}	26.4	22	22.7	22	30.0	28	29.7	28
<i>Net</i> _{3,120}	23.4	21	19.1	18	27.8	27	27.4	27
<i>Net</i> _{4,100}	49.6	46	46.5	45	34.4	33	35.2	35
<i>Net</i> _{4,110}	44.8	42	39.5	37	40.6	39	39.1	38
<i>Net</i> _{4,120}	39.8	37	35.4	34	34.8	34	36.2	36
<i>Net</i> _{4,130}	34.9	32	30.5	29	35.8	35	36.5	36
<i>Net</i> _{4,140}	31.3	29	34.3	25	33.6	33	32.6	32
<i>Net</i> _{4,150}	28.8	26	24.3	23	30.8	29	32.2	30
<i>Net</i> _{4,160}	26.5	25	22.3	22	30.8	30	31.2	30
<i>Net</i> _{5,130}	64.3	60	85.6	57	48.6	43	46.4	45
<i>Net</i> _{5,140}	57.0	52	52.3	50	50.5	47	47.2	46
<i>Net</i> _{5,150}	54.4	51	48.5	46	44.2	43	44.6	44
<i>Net</i> _{5,160}	49.8	45	43.7	43	41.6	41	42.3	41
<i>Net</i> _{6,200}	58.8	52	50.4	49	61.5	59	48.6	46
<i>Net</i> _{6,210}	52.8	50	46.1	45	46.8	43	46.8	46
<i>Net</i> _{6,220}	48.4	45	42.1	40	49.2	48	48.3	47
<i>Net</i> _{6,230}	46.9	44	39.8	39	44.9	43	44.9	44
<i>Net</i> _{7,200}	81.5	79	75.4	73	56.2	54	55.6	55
<i>Net</i> _{7,210}	78.2	74	70.0	67	69.2	64	66.1	65
<i>Net</i> _{7,220}	73.8	69	67.0	62	64.4	61	62.3	61
<i>Net</i> _{7,230}	68.9	66	60.9	59	57.7	55	54.6	54
<i>Net</i> _{8,210}	104.0	98	94.7	90	75.0	70	72.6	71
<i>Net</i> _{8,220}	97.6	91	87.9	82	77.3	73	74.2	73
<i>Net</i> _{8,230}	90.3	86	81.6	78	70.8	69	70.6	69
<i>Net</i> _{8,240}	84.1	80	76.1	74	68.0	65	70.2	67

Table 3 displays the minimum number (Min.) and average (Avg.) values of the network instances obtained by applying the ASS-MCDS and the above-mentioned three methods. Generally, it can be concluded that the proposed method (ASS-MCDS) acquired the best results with the network instances including in instances with a large number of wireless nodes. Figure 7 displays the comparison results of applying different methods on the networks with a large sizes. Although the proposed method performance was close to the other methods, it achieved better results in some networks. On the other hand, the other methods were partially better in some small size networks. In general, the ASS-MCDS method was more practical and promising than other methods in designing and managing wireless networks.

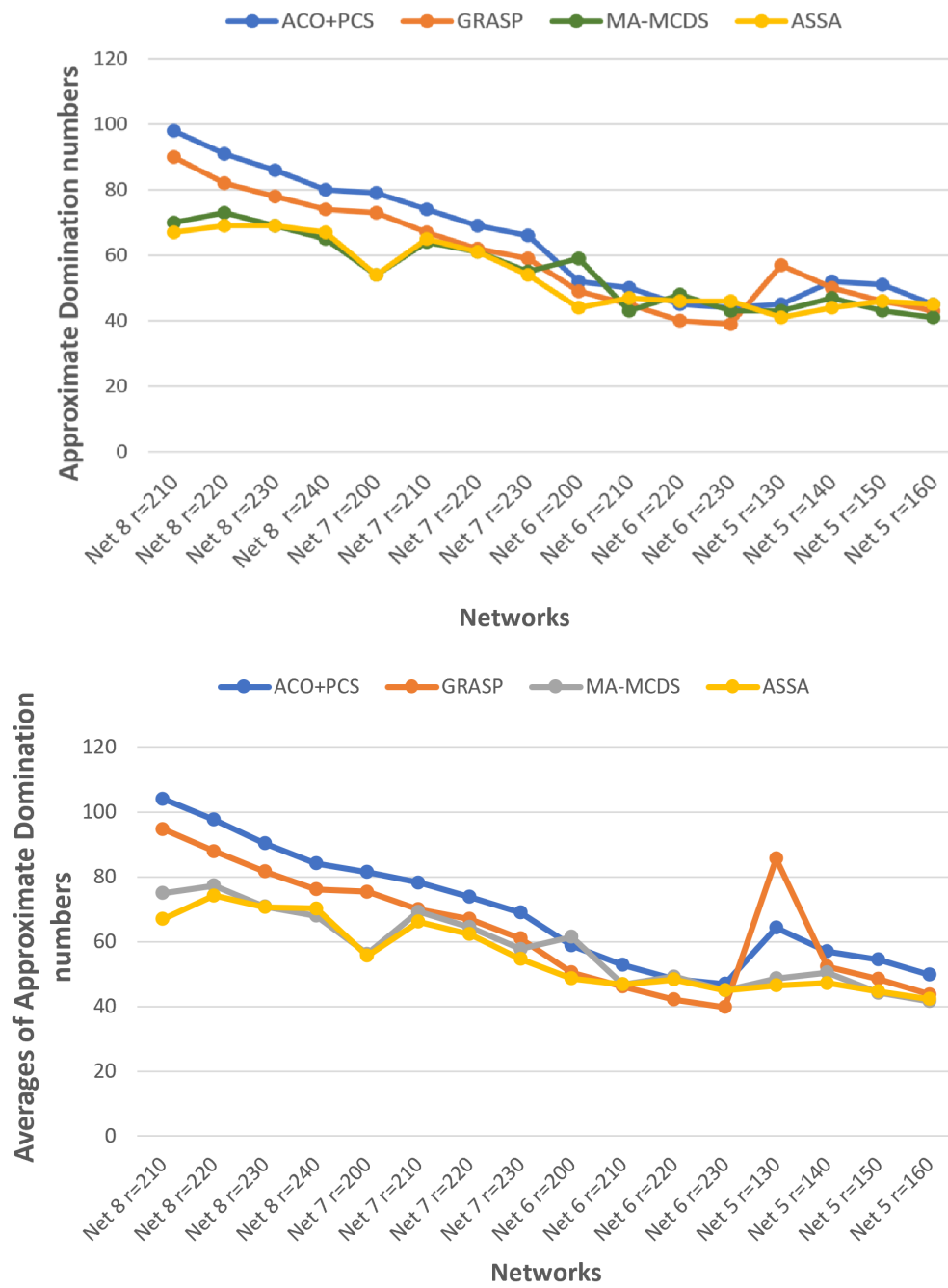


Figure 7. Comparison of results for big-size networks.

6. Conclusion

In this paper, the MCDS problem and its application in wireless sensor network management was investigated. A new algorithm to compute the MCDS was proposed, called the adaptive scatter search algorithm for solving the MCDS problem (ASS-MCDS). Additionally, an extended fitness function with multiple objectives used by the algorithm to evaluate the solutions qualities and achieve a better performance was invoked. In general, numerical experiments on ad hoc wireless networks showed that the proposed method outperformed other methods, particularly in large size wireless networks by identifying the wireless network virtual backbone structure by reformulating it to the MCDS problem. Test results demonstrated that ASS-MCDS was very efficient in finding solutions with high quality to identify MCDS using different standard benchmarks test graphs.

Author Contributions: Conceptualization, A.-R.H., G.A.E.-S.; Methodology, A.-R.H., S.N.A., A.A.S., G.A.E.-S.; Programming and implementation, A.-R.H., S.N.A., G.A.E.-S.; Writing—original draft preparation, A.-R.H., S.N.A., G.A.E.-S.; Writing—review and editing, A.-R.H., A.A.S., G.A.E.-S.; Funding acquisition, A.-R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Plan for Science, Technology, and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, the Kingdom of Saudi Arabia, award number (13-INF544-10).

Acknowledgments: The authors would like to thank King Abdulaziz City for Science and Technology, the Kingdom of Saudi Arabia, for supporting project number (13-INF544-10).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Jovanovic, R.; Tuba, M. Ant colony optimization algorithm with pheromone correction strategy for the minimum connected dominating set problem. *Comput. Sci. Inf. Syst.* **2013**, *10*, 133–149.
- He, J.S.; Ji, S.; Pan, Y.; Cai, Z. Load-balanced virtual backbone construction for wireless sensor networks. In *Combinatorial Optimization and Applications*; Lin, G., Eds.; Springer: Berlin, Germany, 2012; pp. 1–12.
- Nimisha, T.; Ramalakshmi, R. Energy efficient connected dominating set construction using ant colony optimization technique in wireless sensor network. In Proceedings of the 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 19–20 March 2015; pp. 1–5.
- Yu, J.; Wang, N.; Wang, G. Heuristic algorithms for constructing connected dominating sets with minimum size and bounded diameter in wireless networks. In *Wireless Algorithms, Systems, and Applications*; Pandurangan, G., Anil K.V.S., Ming, G., Liu Y., Li, Y., Eds.; Springer: Berlin, Germany, 2010; pp. 11–20.
- Martí, R.; Laguna, M.; Glover, F. Principles of scatter search. *Eur. J. Oper. Res.* **2006**, *169*, 359–372.
- Gendreau, M.; Potvin, J.-Y. *Handbook of metaheuristics*; Springer: New York, NY, USA, 2010; Volume, 2.
- Kar, A.K. Bio inspired computing—A review of algorithms and scope of applications. *Expert Syst. Appl.* **2016**, *59*, 20–32.
- Chiarelli, N.; Milanic, M. Linear Separation of Connected Dominating Sets in Graphs. *arXiv* **2014**, arXiv:1610.06539. Available online: <https://arxiv.org/abs/1610.06539> (accessed on 1 December 2019).
- Downey, R.G.; Fellows, M.R.; McCartin, C.; Rosamond, F. Parameterized approximation of dominating set problems. *Inform. Process. Lett.* **2008**, *109*, 68–70.
- Thai, M.T.; Tiwari, R.; Du, D.Z. On construction of virtual backbone in wireless ad hoc networks with unidirectional links. *IEEE Trans. Mob. Comput.* **2008**, *7*, 1098–1109.
- Hedar, A.-R.; Ismail, R.; Sayed, G.A.E.; Khayyat, K.M.J. Two meta-heuristics for the minimum connected dominating set problem with an application in wireless networks. In Proceedings of the 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence, Okayama, Japan, 12–16 July 2015; pp. 355–362.
- Hedar, A.R.; Ismail, R.; El-Sayed, G.A.; Khayyat, K.M.J. Two Meta-Heuristics Designed to Solve the Minimum Connected Dominating Set Problem for Wireless Networks Design and Management. *J. Netw. Syst. Manag.* **2019**, *27*(3), pp. 1–41.
- Al-Nabhan, N.; Zhang, B.; Cheng, X.; Al-Rodhaan, M.; Al-Dhelaan, A. Three connected dominating set algorithms for wireless sensor networks. *Int. J. Sens. Netw.* **2016**, *21*, 53–66.

14. Guha, S.; Khuller, S. Approximation algorithms for connected dominating sets. *Algorithmica* **1998**, *20*, 374–387.
15. Ruan, L.; Du, H.; Jia, X.; Wu, W.; Li, Y.; Ko, K.I. A greedy approximation for minimum connected dominating sets. *Theor. Comput. Sci.* **2004**, *329*, 325–330.
16. Wu, W.; Du, H.; Jia, X.; Li, Y.; Huang, S.C.H. Minimum connected dominating sets and maximal independent sets in unit disk graphs. *Theor. Comput. Sci.* **2006**, *352*, 1–7.
17. Al-Nabhan, N.; Zhang, B.; Al-Rodhaan, M.; Al-Dhelaan, A. Two connected dominating set algorithms for wireless sensor networks. In *Wireless Algorithms, Systems, and Applications*; Wang, X., Zheng, R., Jing, T., Xing, K. Eds.; Springer: Berlin, Germany, 2012; Volume 7405, pp. 705–713.
18. Rai, M.; Verma, S.; Tapaswi, S. A power aware minimum connected dominating set for wireless sensor networks. *JNW* **2009**, *4*, 511–519.
19. Rai, M.; Verma, S.; Tapaswi, S. A heuristic for minimum connected dominating set with local repair for wireless sensor networks. In *Proceedings of the Eighth International Conference on Networks*, Gosier, Guadeloupe, 1–6 March 2009; pp. 106–111.
20. Balaji, S.; Revathi, N. An efficient heuristic for the minimum connected dominating set problem on ad hoc wireless networks. *Proceedings of WASET* **2012**, *68*, 1045–1051.
21. Li, R.; Hu, S.; Gao, J.; Zhou, Y.; Wang, Y.; Yin, M. GRASP for connected dominating set problems. *Neural Comput. Appl.* **2017**, *28*, 1059–1067.
22. Dagdeviren, Z.A.; Aydin, D.; Cinsdikici, M. Two population-based optimization algorithms for minimum weight connected dominating set problem. *Appl. Soft Comput.* **2017**, *59*, 644–658.
23. Laguna, M. Scatter search. In *Search Methodologies*; Springer: Berlin, Germany 2014; pp. 119–141.
24. Laguna, M.; Marti, R. *Scatter Search: Methodology and Implementations in C*; Springer Science & Business Media: Berlin, Germany, 2012; Volume 24.
25. Talbi, E.G. *Metaheuristics: From Design to Implementation*; John Wiley & Sons, Chichester, UK, 2009; Volume 74.
26. Yang, X.S. Metaheuristic optimization: algorithm analysis and open problems. In *Experimental Algorithms*; Pardalos, P.M., Rebennack, S., Eds.; Springer: Berlin, Germany, 2011; Volume 6630, pp. 21–32.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).