Institute of Advanced Engineering and Science

IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 2, No. 4, December 2013, pp. 179~186
ISSN: 2252-8938

Memetic Algorithm for the Minimum Edge Dominating Set Problem

Abdel-Rahman Hedar*, Shada N. Abdel-Aziz*, and Adel A. Sewisy**
Faculty of Computers and Information, Assiut University, Egypt

Article Info

Article history:

Received Jun 16, 2013
Revised Sep 05, 2013
Accepted Nov 03, 2013

Keyword:

Minimum edge dominating set
Graph theory
Genetic algorithm
Memetic Algorithm
Local search

ABSTRACT

The minimum edge dominating set (MEDS) is one of the fundamental covering problems in graph theory, which finds many practical applications in diverse domains. In this paper, we propose a meta-heuristic approach based on genetic algorithm and local search to solve the MEDS problem. Therefore, the proposed method is considered as a memetic search algorithm which is called Memetic Algorithm for minimum edge dominating set (MAMEDS). In the MAMEDS method, a new fitness function is invoked to effectively measure the solution qualities. The search process in the proposed method uses intensification schemes beside the main genetic search operations in order to achieve faster performance. The experimental results proves that the proposed method is promising in solving the MEDS problem.

Copyright © 2013 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Abdel-Rahman Hedar
Faculty of Computers and Information, Assiut University
Assiut University, Assiut 71526, Egypt
+20-10-00704940
Email: hedar@aun.edu.eg

1. INTRODUCTION

The Minimum Edge Dominating Set (MEDS) is a subset of edges of minimum cardinality, where each edge is be in the edge dominating set, or adjacent to some edges in the edge dominating set [1, 2, 3]. The MEDS problem is a hard combinatorial problem, classified as NP-hard [3], and in general cannot be solved exactly in polynomial time. Several research works have been introduced on exponential-time algorithms for some natural and basic problems, such as independent set [4, 5], coloring [6]. Nowadays, there are a great interest of proposing efficient algorithms for domination problems in graph especially those are coming from networking area [7, 8, 9]. Actually, different types of domination problems; such as dominating set [10], edge dominating set [11] and feedback set [12], have also drawn much attention in this line of research. These domination problems in graphs have been subject of many studies in graph theory, and have many applications in operations research, resource allocation and network routing, as well as in coding theory [1, 2]. The Edge Dominating Set (EDS) problem is a fundamental problem in graphs which simultaneously generalizes and melds vertex cover and edge cover into a restricted version of the total cover problem [13, 14].

There are many algorithms proposed for solving MEDS, these algorithms takes two tracks exact algorithm and parameterized algorithm. Although these algorithms guarantee the optimality of the solutions they find, they may fail to give a solution within reasonable time for large instances. As the size of the problem increases, these methods become futile. Meta-heuristics are powerful search methods which can be efficiently in providing satisfactory solutions to large and complex problems such as vertex cover [15], dominating set [16] and edge coloring [17] in a reasonable time. However, up to the authors' knowledge, there are no studies up to day used meta-heuristic techniques for solving MEDS problem. Genetic Algorithms (GAs) are the most popular meta-heuristic algorithms that have been employed in wide variety of problems

[18]. Actually, GAs are able to incorporate other techniques within its framework to produce a hybrid method that brings more promising one. One direction of such hybridization is to use local search which can accelerate the search process in a pure GA. This modification yields another search approach which is called the Memetic Algorithm (MA) [19].

Several meta-heuristic methods have been developed to solve different problems in graph theory and combinatorial optimization [20, 21]. However, the number of contributions that deal with the graph domination problems is very limited. In this paper, we propose a memetic algorithm for finding the minimum edge dominating set, called shortly MAMEDS. It uses a 0-1 variable representation of solutions in searching for the MEDS, and invokes a new fitness function to measure the solution qualities. Two intensification search schemes are used beside local search in order to enhance the performance of the MAMEDS method.

The paper is organized as follows. The next section gives a brief description about the MEDS problem as preliminaries needed throughout the paper, and highlights the related works in solving the considered problem. Section 3 describes the proposed method steps in details. Section 4 reports numerical experiments and results. Finally, the conclusions make up Section 5.

2. PROBLEM FORMULATION AND RELATED WORKS

The domination problems in graph are mainly classified as vertex dominating set (or dominating set) and edge dominating set. The latter can be defined as follows [3]. Given an undirected graph $G = (V, E)$, without loops and multiple edges, where V is the set of nodes (or vertices) and E the set of edges. An edge $\{u, v\}$ of G is said to dominate itself and any edge adjacent to it in G. An edge dominating set in a graph G is a subset of the edges $D \subseteq E$ such that every edge in E is in D or adjacent to at least one edge in D. The MEDS problem is that of finding an edge dominating set of minimum cardinality in the given graph. The edge domination number of G is denoted by $\acute{\gamma}(G)$ and defined as the minimum number of edges in a set D such that every edge not in D has a vertex in common with at least one edge in D. One can note that the MEDS may not be unique.

The problems of finding the minimum dominating set and minimum edge dominating set have been considered in the literature. For the dominating set problem, there are many studies focus on solving this problem, see for example [16, 22] and references therein. The edge dominating set problem is a basic problem introduced in Garey and Johnson's work [23] on NP-completeness. Yannakakis and Gavril [3] proved that the edge dominating set problem is NP-hard even in planar or bipartite graphs of maximum degree 3. Randerath and Schiermeyer [24] designed the first nontrivial exact algorithm for the minimum edge dominating set problem of time complexity $O(1.4423^m)$, where m is the number of edges in the graph. Later Raman et al. [25] gave an $O(1.4423^n)$ algorithm, and Fomin et al. [26] improved this to $O(1.4082^n)$. Rooij and Bodlaender [11] got an $O(1.3226^n)$ algorithm by using the "measure and conquer" method, which was further improved to $O(1.3160^n)$ [27], where n is the number of vertices.

In terms of parameterized algorithms with parameter k being the size of the solution, there are also a long list of contributions to the upper bound of the running time. Let us quote the $O^*(2{:}6181^k)$-time algorithm by Fernau [28], the $O^*(2{:}4181^k)$-time algorithm by Fomin et al. [26], the $O^*(2{:}3819^k)$-time algorithm by Binkele-Raible and Fernau [29], and finally the $O^*(2{:}3147^k)$-time algorithm by Xiao et al. [30]. Although these algorithms provide the optimal solution, they are too slow on graphs with few hundreds of nodes. Therefore, when deals with a very large graphs, these algorithm become impractical. This motivates us to consider meat-heuristics to design more efficient algorithm to solve the MEDS problem.

3. PROPOSED METHOD

In this section, we describe the components of MAMEDS, and then state its formal algorithm at the end of this section. The MAMEDS method is an evolutionary algorithm, therefore, we first start by describing the solution representation and the fitness function. Then, the genetic operators; selection, crossover and mutation are defined. The main memetic search element, local search, is stated after that. Finally, our intensification schemes are explained.

3.1. Graph Representation

The graph represented as $n_V \times n_V$ adjacency matrix, where n_V is the number of vertices in the graph. Form an adjacency matrix, we create edges matrix E_m which include all edges in the graph. Edge matrix dimension is n_E x 2, where n_E is the number of edges in the graph, the two columns are the vertex numbers which represent the endpoints of edges.

3.2. Solution Representation and Fitness Function

A solution s will be represented as a bit vector with size equal to the number of edges in the graph. Therefore, s is equal to (s_1, s_2, … , s_{nE}), as shown in Figure 1. The subscript numbers 1, 2, …, n_E, are related to the corresponding edges in E_m. If a component s_i of s, i = 1, …, n_E, has the value 1, then the edge represented by the i-th row in E_m is contained in the edge subset represented by solution s. Otherwise, the solution s does not contain the i-th edge.

s_1	s_2	s_3	$\cdots$	s_{n_E}

Figure 1. Solution Representation

Fitness function *fit* is a function designed to measures the quality of a solution which plays a major role in the selection process. The main idea in designing the fitness function is that better solutions will have a higher fitness function value than worse one.

$$fit(s) = \frac{n_D}{n_E} + \frac{1}{\grave{\gamma}_S(G) x n_E} \qquad (1)$$

where n_D is the number of edges dominated by the sub set of edges D represented by the solution s, and $\grave{\gamma}s(G)$ is the number of edges in D.

The fitness function *fit* consist of two parts, the first part n_D/n_E, reflects the size of domination on G by s. If s represents an edge dominating set, then this part is equal to 1. While the second part $1/(\grave{\gamma}s(G)$ x $n_E)$ distinguishes between solutions that have the same values of the first part based on the number of nodes contained in each of them. It is worthwhile to mention that the second term is designed to make $fit(s_1) < fit(s_2)$ in only two cases:

- $c_1 < c_2$, where c_1 and c_2 are the number of edges covered by s_1 and s_2 respectively, or
- $c_1 = c_2$ and $\grave{\gamma}s_1(G) > \grave{\gamma}s_2(G)$.

3.3. Genetic Operators

The parent selection mechanism first produces an intermediate population, say P0 from the initial population P: $P' \subseteq P$ as in the canonical GA. For each generation, P' has the same size as P but an individual can be present in P' more than once. The individuals in P are ranked with their fitness function values based on the *linear ranking selection mechanism* [31, 32]. Indeed, individuals in P' are copies of individuals in P depending on their fitness ranking: the higher fitness an individual has, the more the probability that it will be copied is. This process is repeated until P' is full while an already chosen individual is not removed from P.

The crossover operation has an exploration tendency, and therefore it is not applied to all parents. First, for each individual in the intermediate population P', the crossover operation chooses a random number from the interval (0, 1). If the chosen number is less than the crossover probability $p_c \in (0, 1)$, the individual is added to the parent pool. After that, two parents from the parent pool are randomly selected and mated to produce two children c_1 and c_2, which are then replacing their parents in P'. These procedures are repeated until all selected parents are mated. The standard one-point crossover [33] is used in MAMEDS to compute children.

For each gene each in all individuals in the intermediate population P', a random number from the interval (0, 1) is associated. If the associated number is less than the mutation probability p_m, then the individual is mutated using the standard uniform mutation operation [33].

3.4. Local Search

In *Local Search* mechanism, we add or delete some edges to improve the best solution s^{best} found so far, and this process is repeated n_l times. The formal description of this mechanism is shown in Procedure 1.

Procedure 1 (Local Search)

1. *Repeat the following steps n_l times.*
2. *Set $\tilde{s}^{best} = s^{best}$.*
3. *If $fit(\tilde{s}^{best}) < 1$, select a component $\tilde{s}_i^{best}$ with value 0. This selection is randomly and proportional to the degree of its corresponding node. Set $\tilde{s}_i^{best} = 1 - \tilde{s}_i^{best}$.*
4. *If $fit(\tilde{s}^{best}) < 1$, select a component $\tilde{s}_i^{best}$ with value 1. This selection is randomly and inversely proportional to the degree of its corresponding node. Set $\tilde{s}_i^{best} = 1 - \tilde{s}_i^{best}$.*
5. *If $fit(\tilde{s}^{best}) > fit(s^{best})$, set $s^{best} = \tilde{s}^{best}$.*

3.5. Intensification Schemes

Two mechanisms are used in MAMEDS to reduce the cardinality of the solution computed. The best edge dominating set s^{best} will be refine with another intensification scheme called *Filtering*. This echanism basically checks if an edge contained in s^{best} can be removed without losing the coverage.

Procedure 2 (Filtering)

1. *If* $fit(s^{best}) < 1$, *return*.
2. *Compute the set* $X = \{x1, \ldots, x|X|\}$ *of all positions of value one in* s^{best}.
3. *Repeat the following steps for* $j = 1, \ldots, |X|$.
4. *Set* $s_{xj}^{best} = 0$*, and compute the new fitness value.*
5. *If the fitness value is increased, update* s^{best}.

The final intensification mechanism is called Elite Edge Dominating Sets Inspiration. In this mechanism the best edge dominating set which have been visited are saved in a set called Edge Dominating Sets (*EDS*). A trial solution s^{Core} is define as the intersection of the n_{Core} best edge dominating sets in *EDS*, where n_{Core} is a pre-specified number. If the number of edges contained in s^{Core} is less than that in s^{best} by at least two, then the zero position in s^{Core} which gives the highest edge-degree is updated to be one. This mechanism is repeated until the number of edges contained in s^{Core} becomes less than that in s^{best} by one.

Procedure 3 (Elite Inspiration)

1. *If EDS is empty, then return.*
2. *Set* n_F *equal to the number of edges contained in* s^{best}*, and set* s^{Core} *equal to the intersection of the* n_{Core} *best edge dominating sets in EDS.*
3. *If* $\sum_{i=1}^{|E|} s_i^{Core} < n_F - 1$*, then go to Step 4. Otherwise, return.*
4. *If* $fit(s^{Core}) \geq 1$*, then return.*
5. *Update the zero position in* s^{Core} *which gives the highest fitness, and go to Step 3.*

3.6. MAMEDS algorithm

MAMEDS starts with an initial population of chromosomes P_0 generated randomly. Each chromosome represents a trial solution to the MEDS problem. During each generation, the quality of each chromosome in the population is evaluated by using a fitness function *fit* (see Equation (4.1)). MAMEDS applies Procedure 1 to improve the best solution. In each generation, the population is updated through genetic operators. Linear ranking selection algorithm uses to select parents for standard one-point crossover and uniform mutation to generate members of the new population [16]. MAMEDS invokes LocalSearch Procedure to update the current population. If a certain number of consecutive generations without improvement is achieved, MAMEDS invokes Procedure 2 to improve the best edge dominating set s^{best} obtained so far, if it exists. The search will be terminated if the number of generations exceeds g_{max}, or the number of consecutive generations without improvement exceeds a pre-specified number. Finally, Elite Edge Dominating Set Inspiration Procedure is applied as a final intensification mechanism.

Algorithm 4 (MAMEDS)

1. *Initialization. Set values of* P_{size}, g_{max}, n_{step}, n_{Core}. *Set the crossover and mutation probabilities* $p_c \in (0, 1)$ *and* $p_m \in (0, 1)$*, respectively. Set EDS to be an empty set. Generate an initial population* P_0 *of size* P_{size}.
2. *Local Search. Evaluate the fitness function of all chromosomes in* P_0 *by using the Equation (4.1), and then apply Procedures 1 to improve the best trial solution in* P_0*. Set the generation counter* $t := 0$.
3. *Parent Selection. Select an intermediate population* $\acute{P}_t$ *from the current population* P_t *using the linear ranking selection*.
4. *Crossover. Apply the standard one-point crossover to chromosomes in* $\acute{P}_t$*, and update* P_t.
5. *Mutation. Apply the standard uniform mutation to chromosomes in* $\acute{P}_t$*, and update* $\acute{P}_t$.
6. *Survival Selection. Evaluate the fitness function of all generated children in the updated* $\acute{P}_t$*, and set* $P_t +1 = \acute{P}_t$*. If the best solution in* $P_t +1$ *is worse than the best solution in* $\acute{P}_t$*, then replace the worst solution in* $\acute{P}_t +1$ *by the best solution in* $\acute{P}_t$.
7. *Local Search. Apply Procedure 1 to improve the* s^{best}*, update EDS.*
8. *Filtering. If* s^{best} *represents an edge dominating set, then apply Procedure 2 to improve it, update EDS.*
9. *Stopping Condition. If* $t > g_{max}$*, then go to Step 10. Otherwise, set* $t := t + 1$*, and go to Step 3.*
10. *Final Intensification. Apply Procedure 3 to obtain* s^{Core}*. Update EDS by* s^{Core} *if a better solution is found, and terminate.*

4. NUMERICAL EXPERIMENTS

The MAMEDS algorithm was programmed using MATLAB. In this experimental section, we technically discuss the implementation of the MAMEDS code as well as its results. This section also shows how the test graphs used in the numerical simulations are generated.

4.1. Graph Generation

In order to measure the performance of MAMEDS we apply it on number of graphs with different sizes. The previous works in solving MEDS did not implemented for special types of graphs. Thus, the graphs which we used in our experiments are randomly generated with a known edge domination number $\acute{\gamma}$(G). The following algorithm describe how these graphs are constructed.

Algorithm 5 (Graph Generation)

1. *Set the maximum number of edges* $max_E = n_V \times (n_V - 1)/2$, *and the number of edges* $n_E = max_E \times d$, *where d is the density of edges in the graph which is set to be in* (0, 1), *and* n_V *is the number of vertices.*
2. *Divide the vertices into two groups*:
   - V_{ED} *with size equal to* $\acute{\gamma}(G)$ x 2, *and has vertices incident to dominant edges. Therefore, each pair of them is connected*.
   - V_E *with size equal to* n_V - ($\acute{\gamma}(G)$ x 2), *and has vertices not incident to dominant edges*.
3. *Add edges to connect the graph vertices to reach the edge density d. This edge adding process should satisfy the following condition in order to maintain the edge domination number equal to* $\acute{\gamma}(G)$.
   - *No edge connects two vertices belong to different pairs in* V_{ED}.

MAMEDS was applied 9 instances of MEDS problems created from the three graphs G1-G3, see Table 1. These three graphs generated randomly with a number n_V of vertices and different number n_E of edges depending on the density number *d* for each instance. For each problem instance, the edge domination number $\acute{\gamma}$(G) was known and the code was run 10 times.

4.2. Parameter Setting and Tuning

Table 3 summarize all parameters setting used in MAMEDS with their assigned values. These chosen values are based on our numerical experiments or their universal setting. In parameter tuning, we have tried to find the best parameter values with moderate costs and good performance.

Table 1. Test Problems

Test graphs	No. of Nodes	No. of Edges	d	$\acute{\gamma}$(G)
$G_{0.1}^{20}$	20	19	0.1	4
$G_{0.3}^{20}$	20	57	0.3	4
$G_{0.5}^{20}$	20	95	0.5	4
$G_{0.1}^{50}$	50	123	0.1	15
$G_{0.3}^{50}$	50	368	0.3	15
$G_{0.5}^{50}$	50	613	0.5	15
$G_{0.1}^{150}$	150	1118	0.1	55
$G_{0.1}^{150}$	150	3353	0.3	55
$G_{0.1}^{150}$	150	5588	0.5	55

Table 2. MAMEDS Parameter setting

Parameter	Definition	Value
P_{size}	Population size	40
p_c	Crossover probability	0.8
p_m	Mutation probability	0.01
n_{step}	Number of iterations in LocalSearch	2
$n_{emphEDS}$	Max number of the best edge dominating sets used to update EDS	10
n_{Core}	The number of the best edge dominating sets used to compute s^{Core}	3
g_{max}	Max number of generations	100

4.3. Results and Discussion

The performance of the proposed MAMEDS is tested on 9 instances of the MEDS problem created from the three graphs G1 - G3, as shown in Table 1. The results of the MAMEDS are reported in Tables 4 - 6. The program runs 10 times for each instance. The average number of the best minimum edge dominating sets (*Ave*.) which we obtained for all instances of the MEDS problem will be compared with edge domination number $\acute{\gamma}$(G). The (*rate*) shows how many times MAMEDS acquires an optimal solution $\acute{\gamma}$(G). These best solutions are different according to tuning parameter values which significantly affects solution qualities. In Table 4, when the value of P_{size} increases for 40 to 100 the results are improved relatively. In Table 5, the results are better than that in Table 4 and when the value of g_{max} increases the results are improved significantly. The best results which we obtained from our experiments are presented in Table 6, when the value of n_{step} increased form 2 to 6. In G1 instances, the tuning parameters P_{size}, g_{max} and n_{step} with values 100, 100 and 2, respectively, are enough to acquire the optimal solution. In G2 and G3 instances the tuning parameters P_{size}, g_{max} and n_{step} with values 100, 100 and 6, respectively, are enough to acquire the optimal or near optimal solution. In large graphs, these values must be increased proportionally with raph size. The results that are obtained can be demonstrate the relationship between tuning parameter values and graph size. In addition the efficiency of the MAMEDS in the instances of the same graph are different according to its number of edges n_E. In general the experiment results prove the efficiency of MAMEDS algorithm for solving MEDS problem.

5. CONCLUSION

The minimum dominating set problem in graph theory has been studied in this paper. A hybrid GA-based method, called memetic algorithm for minimum dominating set (MAMEDS), has been proposed to solve the considered problem. New fitness function and intensification elements have been applied in MAMEDS to achieve better performance and to fit the problem. The values of tuning parameter are significantly affect the results. In general numerical experiments on 9 instances of graphs have are show the efficiency of MAMEDS.

Table 3. Results of MAMEDS on G1, G2 and G3 (g_{max} = 100, n_{step} = 2)

Graph no	n_E	P_{size}	$\dot{\gamma}$(G)	*Avg*	*rate*
$G_{0.1}^{20}$	19	40	4	4	10
$G_{0.1}^{20}$	19	100	4	4	10
$G_{0.3}^{20}$	57	40	4	4.8	6
$G_{0.3}^{20}$	57	100	4	4	10
$G_{0.5}^{20}$	95	40	4	5.2	4
$G_{0.5}^{20}$	95	100	4	4.2	8
$G_{0.1}^{50}$	123	40	15	16.2	6
$G_{0.1}^{50}$	123	100	15	15.8	7
$G_{0.3}^{50}$	368	40	15	17	4
$G_{0.3}^{50}$	368	100	15	16.8	4
$G_{0.5}^{50}$	613	40	15	17.8	4
$G_{0.5}^{50}$	613	100	15	17.4	5
$G_{0.1}^{150}$	1118	40	55	66	4
$G_{0.1}^{150}$	1118	100	55	64	3
$G_{0.3}^{150}$	3353	40	55	70	3
$G_{0.3}^{150}$	3353	100	55	68	3
$G_{0.5}^{150}$	5588	40	55	73	1
$G_{0.5}^{150}$	5588	100	55	68	0

Table 4. Results of MAMEDS on G1, G2 and G3 (P_{size} = 100, n_{step} = 2)

Graph no	n_E	G_{max}	$\dot{\gamma}$(G)	*Avg*	*rate*
$G_{0.1}^{20}$	19	120	4	4	10
$G_{0.1}^{20}$	19	150	4	4	10
$G_{0.3}^{20}$	57	120	4	4.8	5
$G_{0.3}^{20}$	57	150	4	4.2	9
$G_{0.5}^{20}$	95	120	4	4.2	9
$G_{0.5}^{20}$	95	150	4	4.2	8
$G_{0.1}^{50}$	123	120	15	15.4	6
$G_{0.1}^{50}$	123	150	15	15	10
$G_{0.3}^{50}$	368	120	15	16.4	4
$G_{0.3}^{50}$	368	150	15	15.2	8
$G_{0.5}^{50}$	613	120	15	15.8	5
$G_{0.5}^{50}$	613	150	15	16	2
$G_{0.1}^{150}$	1118	120	55	64	3
$G_{0.1}^{150}$	1118	150	55	65	1
$G_{0.3}^{150}$	3353	120	55	64.4	2
$G_{0.3}^{150}$	3353	150	55	65	2
$G_{0.5}^{150}$	5588	120	55	66.8	0
$G_{0.5}^{150}$	5588	150	55	66	1

Table 5. Results of MAMEDS on G1, G2 and G3 (g_{max} = 100, P_{size} = 100)

Graph no	n_E	n_{step}	$\dot{\gamma}$(G)	*Avg*	*rate*
$G_{0.1}^{20}$	19	2	4	4	10
$G_{0.1}^{20}$	19	6	4	4	10
$G_{0.3}^{20}$	57	2	4	4.8	6
$G_{0.3}^{20}$	57	6	4	4	10
$G_{0.5}^{20}$	95	2	4	5.2	4
$G_{0.5}^{20}$	95	6	4	4.2	9
$G_{0.1}^{50}$	123	2	15	16.2	6
$G_{0.1}^{50}$	123	6	15	15	10
$G_{0.3}^{50}$	368	2	15	17	4
$G_{0.3}^{50}$	368	6	15	15	10
$G_{0.5}^{50}$	613	2	15	17.8	4
$G_{0.5}^{50}$	613	6	15	15.8	6
$G_{0.1}^{150}$	1118	2	55	66	4
$G_{0.1}^{150}$	1118	6	55	62	2
$G_{0.3}^{150}$	3353	2	55	70	3
$G_{0.3}^{150}$	3353	6	55	62.4	0
$G_{0.5}^{150}$	5588	2	55	73	0
$G_{0.5}^{150}$	5588	6	55	62.8	1

REFERENCES

[1] TW Haynes, ST Hedetniemi, and PJ Slater. *Domination in graphs: advanced topics*. Marcel Dekker New York. 1998; 40.

[2] ——, *Fundamentals of Domination in Graphs*. CRC PressI Llc. 1998; 208.

[3] M Yannakakis and F Gavril. "Edge dominating sets in graphs". *SIAM Journal on Applied Mathematics*. 1980; 38(3): 364-372.

[4] FV Fomin, F Grandoni, and D Kratsch. "*Measure and conquer: a simple o (2 0.288 n) independent set algorithm*". in Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm. ACM. 2006: 18-25.

[5] RE Tarjan and AE Trojanowski. "Finding a maximum independent set". *SIAM Journal on Computing*. 1977; 6(3): 537-546.

[6] R Beigel and D Eppstein. "3-coloring in time o (n1. 3289)". *Journal of Algorithms*. 2005; 54(2): 168-204.

[7] V Anitha and M Sebastian. "Application oriented connected dominating set-based cluster formation in wireless sensor networks". *Journal of the Brazilian Computer Society*. 2011; 17(1): 3-18.

[8] Z Liu, B Wang, and L Guo. "A survey on connected dominating set construction algorithm for wireless sensor networks". *Information Technology Journal*. 2010; 9(6): 1081-1092.

[9] R Misra and C Mandal. "Minimum connected dominating set using a collaborative cover heuristic for ad hoc sensor networks". *IEEE Transactions on Parallel and Distributed Systems*. 2010; 21(3): 292-302.

[10] FV Fomin, F Grandoni, and D Kratsch. "Measure and conquer: domination–a case study". in *Automata, Languages and Programming*. Springer. 2005: 191-203.

[11] JM Van Rooij and HL Bodlaender. "Exact algorithms for edge domination". in *Parameterized and Exact Computation*. Springer. 2008: 214-225.

[12] I Razgon. "Exact computation of maximum induced forest". in *Algorithm Theory–SWAT 2006*. Springer. 2006: 160-171.

[13] R Carr, T Fujito, G Konjevod, and O Parekh. "A {{10}}-approximation algorithm for a generalization of the weighted edge-dominating set problem". *Journal of Combinatorial Optimization*. 2001; 5(3): 317-326.

[14] O Parekh. "*Polyhedral techniques for graphic covering problems*". Ph.D. thesis, Carnegie Mellon University. 2002.

[15] O Ugurlu. "*New heuristic algorithm for unweighted minimum vertex cover*". in IV International Conference "Problems of Cybernetics And Informatics". Conference Proceedings. 2012.

[16] AR Hedar and R Ismail. "Hybrid genetic algorithm for minimum dominating set problem". in Computational Science and Its Applications–ICCSA 2010. *Springer*. 2010: 457-467.

[17] T Januario, S Urrutia, B Horizonte, and M Gerais-Brazil. "*An edge coloring heuristic based on vizings theorem*". in Sejam bem-vindos aos pr-anais XVI CLAIO. Conference Proceedings. 2012.

[18] LD Chambers. *Practical handbook of genetic algorithms: complex coding systems*. CRC press. 2010; 3.

[19] P Moscato, C Cotta, and A Mendes. "Memetic algorithms". in New optimization techniques in engineering. *Springer*. 2004: 53-85.

[20] C Blum and A Roli. "Metaheuristics in combinatorial optimization: Overview and conceptual comparison". *ACM Computing Surveys (CSUR)*. 2003; 35(3): 268-308.

[21] EG Talbi. Metaheuristics: from design to implementation. *Wiley*. 2009: 74.

[22] A Potluri and A Singh. "Two hybrid meta-heuristic approaches for minimum dominating set problem". in Swarm, Evolutionary, and Memetic Computing. *Springer*. 2011: 97-104.

[23] M Garey and J DS. "Computers and intractability a guide to the theory of np-completeness". JIS, M Tech CSE Syllabus. 1979.

[24] B Randerath and I Schiermeyer. "Exact algorithms for minimum dominating set". *Zentrum für Angewandte Informatik Köln, Lehrstuhl Speckenmeyer, Tech. Rep.* 2004.

[25] V Raman, S Saurabh, and S Sikdar. "Efficient exact algorithms through enumerating maximal independent sets and other techniques". *Theory of Computing Systems*. 2007; 41(3): 563-587.
[26] FV Fomin, S Gaspers, S Saurabh, and AA Stepanov. "On two techniques of combining branching and treewidth". *Algorithmica*. 2009; 54(2): 181-207.
[27] M Xiao and H Nagamochi. "A refined exact algorithm for edge dominating set". in Theory and Applications of Models of Computation. *Springer*. 2012: 360-372.
[28] H Fernau. "Edge dominating set: Efficient enumeration-based exact algorithms". in Parameterized and Exact Computation. *Springer*. 2006: 142-153.
[29] D Binkele-Raible and H Fernau. "Enumerate and measure: improving parameter budget management". in Parameterized and Exact Computation. *Springer*. 2010: 38-49.
[30] M Xiao, T Kloks, and SH Poon. "New parameterized algorithms for the edge dominating set problem". *Theoretical Computer Science*. 2012.
[31] JE Baker. "*Adaptive selection methods for genetic algorithms*". in Proceedings of the 1st International Conference on Genetic Algorithms. L. Erlbaum Associates Inc. 1985: 101-111.
[32] AR Hedar and M Fukushima. "Minimizing multimodal functions by simplex coding genetic algorithm". *Optimization Methods and Software*. 2003; 18(3): 265-282.
[33] F Herrera, M Lozano, and JL Verdegay. "Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis". *Artificial intelligence review*. 1998; 12(4): 265-319.

BIOGRAPHY OF AUTHORS

	**Abdel-Rahman Hedar** received his B.Sc. and M.Sc. (Mathematics) from Assiut University, Egypt in 1993 and 1997, respectively. He also holds a Doctor of Informatics degree from Kyoto University, Japan in 2004. He is currently an associate professor of computer science in Assiut University. He is also the director of quality assurance unit at the faculty of computing and informatics at Assiut University, Egypt, and he was a member of Egyptian Cloud Computing Center at Assiut, Egypt. His research includes meta-heuristics, global optimization, machine learning, data mining, bioinformatics and parallel programming. He has published over 40 papers in international journals and conferences and his h-index is 10. He is a member of the ACM and IEEE (Computer Society). From July 2005 to July 2007, he was a JSPS research fellow in Kyoto University, Japan.