IJE TRANSACTIONS B: Applications Vol. 37 No. 02, (February 2024) 298-311

International Journal of Engineering

Journal Homepage: www.ije.ir

A Complex Network-based Approach for Designing of Wide Area Measurement Systems in Smart Grids using Adam-Eve Like Genetic Algorithm

M. Shahraeini*, R. Soltanifar

Department of Electrical Engineering, Faculty of Engineering, Golestan University, Gorgan, Iran

P A P E R   I N F O

Paper history:
Received 07 August 2023
Received in revised form 21 September 2023
Accepted 23 September 2023

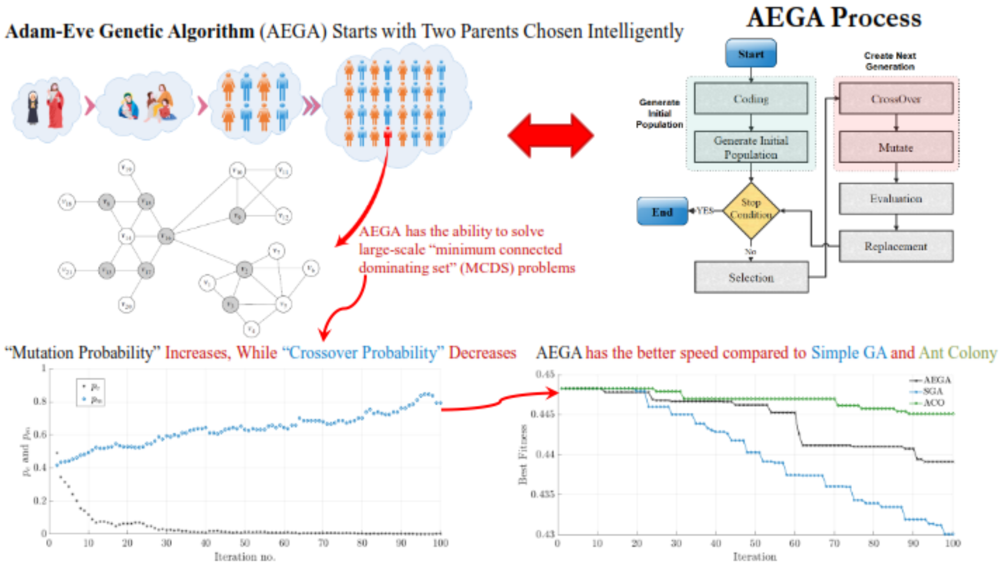
Keywords:
Wide Area Measurement System
Minimum Connected Dominating Set
Genetic Algorithm
Adaptive Genetic Algorithm
Ant Colony Optimization

A B S T R A C T

The wide area measurement system (WAMS) consists of two different measuring and communication infrastructures, which is respectively responsible for measuring power girds' data in the wide area and sending and processing them in the control centers. The design of WAMS can include the design of each of its infrastructures or target both infrastructures at the same time, the latter has been known as the WAMS comprehensive design. The WAMS comprehensive design means the simultaneous placement of measurement components and its required communication, which is known as minimum connected dominating set (MCDS) problem in graph theory and is formulated in the form of an optimization problem. Solving such a complex optimization problem is often done with evolutionary algorithms (e.g. genetic algorithm and ant colony), and the speed and efficiency of finding the solution has always been a challenge. This research proposes an adaptive genetic algorithm known as the Adam and Eve algorithm, which has the ability to solve the MCDS problem that arises from the WAMS comprehensive design. Through simulation results for IEEE 1354 bus network, we demonstrate that proposed algorithm is well-tuned to solved MCDS related to the power graphs. It is 30% faster than simple genetic algorithm, handles large-scale problems effectively, and outperforms both simple genetic algorithm and ant colony algorithm within a given timeframe.

***doi**: 10.5829/ije.2024.37.02b.07*

Graphical Abstract



*Corresponding Author Email: *m.shahr@gu.ac.ir* (M. Shahraeini)

Please cite this article as: Shahraeini M, Soltanifar R. A Complex Network-based Approach for Designing of Wide Area Measurement Systems in Smart Grids using Adam-Eve Like Genetic Algorithm, International Journal of Engineering, Transactions B: Applications. 2024;37(02):298-311.

1. INTRODUCTION

The term "Wide Area Measurement System" (WAMS) was first introduced to the power system literature in the late 20th century to describe a novel measurement system that has become a crucial component of smart grids. Such systems include new advanced digital measurement devices (e.g. PMUs), and modern and high-speed communication infrastructure that enable the efficient management of the complex behavior demonstrated by extensive power grids. In general, the WAMS consists of three interconnected subsystems: data acquisition, data transmission, and data processing (1, 2).

Normally, "WAMS design" may concentrate on any of the mentioned subsystems. For example, placing of the PMUs is considered as the design of the measurement subsystem, while the design of the communication infrastructure for such units implies the design of the communication subsystem of the WAMS. The former is known as the optimal PMU placement (OPP) problem (3), while the latter can be considered as the optimal OPGW placement (OOP) problem (4). It should be noted that the PMU has the ability to measure the voltage and current phasor with high resolution in the entire of the power grid, and OPGW optical fiber is preferred as the transmission medium for these units due to its low latency and high capacity. It can be installed as a part of transmission lines (2).

Both above mentioned problems (i.e. OPP and OOP) can be formulated as the optimization problems (5). In WAMS design, there is an opportunity to place different WAMS components (i.e. PMUs and OPGWs) of two subsystems at the same time, which we call it "*WAMS comprehensive design*" (6). This can be also considered as optimal PMU and OPGW placement problem (OPOP) and it is also can be formulated as the optimization problem (5).

As explained above, different WAMS design problems may be formulated as the optimization problems and should be solved by optimization problem-solving methods. In general, there are three main categories of methods for solving optimization problems (7): deterministic, stochastic, and hybrid. Deterministic methods, also known as Gradient-based methods, are generally faster computationally and can find local optimum. Examples of deterministic methods include Gradient Conjugate, Newton, and Dayton-Fletcher-Powell methods (8). On the other hand, stochastic methods (also known as evolutionary algorithms) are slower but have the potential to find global optimum. Examples of stochastic methods include genetic algorithm (GA), simulated annealing (SA), ant colony (ACO), and particle swarming algorithms (PSO). To address the limitations of both deterministic and stochastic methods, hybrid methods have been developed that combine the robustness of stochastic methods with the speed of deterministic ones (7).

Among the evolutionary algorithms introduced above, the genetic algorithm is a widely used and highly effective evolutionary algorithm that can tackle various problems across different domains. It is also a good choice for multi-objective problems (9-11). GAs may be classified into several categories, including simple genetic algorithm, parallel and distributed genetic algorithm, fast messy genetic algorithm, adaptive genetic algorithm, hybrid genetic algorithm, and independent sampling genetic algorithm (12, 13).

The design problems discussed earlier (i.e. WAMS design problems) have equivalents in the complex networks, with the OPP referred to as the "minimum dominating set" (MDS) problem and the OPOP known as the "minimum connected dominating set" (MCDS) problem (5). These are both considered to be NP-hard problems (14), and their effectiveness is a major concern.

In particular, the MCDS problem is a more complex problem because it performs the simultaneous placement of two different categories of WAMS components (i.e., PMUs and OPGWs) and is often a multi-objective optimization problem with various constraints in which effectiveness becomes more important in such a problem (15-18). Besides the efficiency in solving of MCDS (hereafter we call it "*performance*"), the speed of obtaining the solution (hereafter we call it "*speed*") is also a significant consideration, particularly when dealing with large problem sizes.

The complexity of the MCDS problem and the emphasis on performance and speed has led to the introduction of sequential techniques in addition to the comprehensive approach for solving the MCDS problem. Unlike the comprehensive approach that solves the entire MCDS problem in the form of an optimization problem, sequential methods first place PMUs and then, using complex network-based algorithms, OPGWs are placed to produce required WAMS communication infrastructure. In these methods i.e., sequential ones, the placement of PMUs is often in the form of optimization, while graph algorithms are generally based on the shortest path.

The problem of addressing the MCDS problem in smart grids and comprehensive designing of wide area measurement systems has been a concern for approximately a decade. For the first attempt made by Fan and Watson (19) and by using integer programming, were solved MCDS problem for simultaneous placement of PMUs and their required communication infrastructure but there; they didn't point to MCDS as the WAMS comprehensive design. At the same time, the WAMS comprehensive design was firstly introduced by Shahraeini et al. (6), while the authors didn't directly point to MCDS problem. There, simple genetic algorithm

(SGA) was used to solve proposed problem and the solution was of great importance, while performance and speed of SGA were not of main concern. After that, many researches concentrate on improvement of solutions for WAMS design, while many of them introduce sequential methods for WAMS design (15-18), that is, they first perform OPP and then place communication links by graph shortest path algorithms. Meanwhile, few studies (e.g. (20)) have focused on solving the MCDS problem for WAMS design, that is, they place PMUs and OPGWs in one run and by solving MCDS problem. In our recent works, we have introduced ACO for solving MCDS problem with some heuristics such as pheromone modification. There, we have dramatically improved performance compared to SGA results presented by Shahraeini et al. [6], but still the speed of reaching to the best solution has been not improved there. As a new approach, machine learning and deep learning algorithms (e.g. methods presented in literature (21-27)) are another approach to study MCDS; for instance, convolutional neural networks (CNNs) can be used to classify nodes in a graph as either part of the MCDS or not.

A review of previous works presented above indicates that due to NP-hardness of MCDS problem, performance and speed of methods used for solving of such a problem (either comprehensive methods or sequential ones) are still of main concern. On the other hands, stochastic method like GA and ACO are preferred methods in solving MCDS as the comprehensive approach since they are simple to implement and at the same time guarantee to find at least one local optimum; but they have been not examined yet in large-scale problems for the large-scale power grids. As a result of these facts, introducing new stochastic methods for improvement of performance and speed in solving MCDS problems that has the ability to solve such large-scale problems are still an opening issue for researchers and this is the main motivation of this study.

The main objective of this study is to propose an adaptive genetic algorithm to solve MCDS problem that is derived from WAMS comprehensive design. In our very last publication (28), we have introduced Adam-Eve Genetic Algorithm (AEGA) as an adaptive GA to solve OPP problem. There, we have shown the advantages of AEGA over other GA algorithms like SGA. In this study, we have designed previously proposed AEGA (28) to solve MCDS (i.e. OPOP) problem in large-scale power grids. To the best of our knowledge, this is the first attempt to implement an adaptive GA to solve MCDS problem in power graphs, and also the first study that addresses the solving of large-scale problems.

The main contributions of this paper are as follows:

- The design and setting of the parameters of an adaptive GA (e.g. AEGA) strongly depends on the nature and type of the problem and is inherently hard, which requires appropriate heuristics such as choosing initial population and well-tuning of crossover and mutation rates. In this study and for the first time, this is well done for the MCDS problem raised from large-scale power grids.
- Although MCDS problem is an offline design problem, algorithms such as ACO and SGA, whose high performance has been proven before [5-6], are unable to solve such large-scale problems in a reasonable amount of time. For this reason, the speed of reaching to the solution is very important in large-scale power grids, and the proposed AEGA has improved the speed by 30% compared to its basic version, i.e., SGA.
- The speed improvement in OPOP that has been done in the current research, along with the performance improvement in OPP that was done in our previous research (28), is a proof of the fact that adaptive GAs have a good ability to improve solving of the problems related to complex networks such and MDS and MCDS problems. Additionally, better performance during the same execution time than SGA and ACO in large-scale power graphs is another advantage of the proposed AEGA that has been shown in the current research.
- Choosing Adam and Eve at the beginning of AEGA that is proposed in this study makes the search space closer to the optimal solutions, and this is a considerable improvement in performance and speed. We have shown that this heuristic can be also used for other stochastic methods like SGA and ACO for performance improvement.

In short, our proposed AEGA is well-tuned to solved MCDS raised from power graphs, it is 30% faster than SGA, it has the ability to solve large-scale problems, and it achieves better performance than both of SGA and ACO within a specific timeframe.

The rest of this paper is organized as follows: Section 2 review genetic algorithm and its functions and operators. Classification of different kinds of GA algorithms is also provided in this section. Section 3 proposes simple genetic algorithm and its implementation details. Section 4 proposes Adam-Eve like genetic algorithm and details of its implementation, i.e., its functions and procedures. In section 5 first we have formulated MCDS as an optimization problem and then by setting parameters of the proposed Adam-Eve algorithm and defining some heuristics, the proposed method is adjusted to solve MCDS problem, which is derived from WAMS comprehensive design. Simulation results for IEEE 1354 test network, which are obtained by three different meta-heuristics algorithms (i.e. AEGA, SGA, and ACO), will be presented in section 6. This paper will be end with conclusion in section 7.

2. GENETIC ALGORITHM

Holland (29) introduced the genetic algorithm in 1962 and worked on its development with his colleagues during the 1960s and 1970s. In 1975, Holland (29) published a book called "Adaptation in Natural and Artificial Systems." Since then, extensive research has been conducted on this algorithm, resulting in the introduction of various types of genetic algorithms. The investigation's findings demonstrate the effectiveness of this algorithm in solving diverse problems (12, 13, 29).

The main idea of the genetic algorithm is based on the concepts of natural selection, inheritance, and the possibility of individual change. Its objective is to discover the most effective solution by starting with a set of answers known as the initial population, each represented as a chromosome. To generate the next generation, operators are employed. These operators include selection, mutation, crossover, and replacement, which are applied to each generation in sequence. By utilizing these operators, the genetic algorithm can be executed in a systematic manner (28, 29). An outline of the various steps involved in different genetic algorithms is presented in **Algorithm 1**. Indeed, genetic algorithm utilizes some functions and operators in the aforementioned steps.

Algorithm 1 Simple Genetic Algorithm

Inputs:
Gen Coding;
EVALUATE: Fitness Function;
SELECT: Selection Function;
REPLACEMENT: Replacement Method;
SC: Stop Condition;
n_{pop}: Population size;
p_c and p_m: Crossover and Mutation Probabilities.
Output:
$\mathbf{best}_{\text{sol}}$: Best Solution.

Initialize:
$[pop] \leftarrow$ Randomly generate n_{pop} number of individuals
EVALUATE($[pop]$) ▷ evaluate individuals by fitness function
gen $\leftarrow$ 1 ▷ first generation

while $\sim$ SC **do**
  $[pop_c] \leftarrow$ generate individuals by crossover
    ▷ individual selection is based on SELECT and p_c
  $[pop_m] \leftarrow$ generate mutated individuals
    ▷ individual selection is based on SELECT and p_m
  $[pop_{total}] \leftarrow [pop] \cup [pop_c] \cup [pop_m]$
    ▷ merge all new and old individuals
  EVALUATE($[pop_{total}]$)
    ▷ evaluate merged population
  $\mathbf{best}_{\text{sol}} \leftarrow$ best of $[pop_{total}]$
    ▷ store best solution
  $[pop] \leftarrow$ REPLACEMENT($[pop_{total}]$)
    ▷ select n_{pop} number of merged population
  gen $\leftarrow$ gen + 1
    ▷ preparing for next generation
end while
Return $\mathbf{best}_{\text{sol}}$

2. 1. Genetic Algorithm Functions and Operators

In the subsequent sections, we provide a brief explanation of the roles played by each functions and operators of genetic algorithm.

2. 1. 1. Coding The process of representing solutions to a problem as chromosomes is known as coding. In general, different problems require different coding methods, and the selection of a particular encoding method depends on the nature and characteristic of the problem. Binary encoding and permutation encoding are among the various types of coding available (30). Based on the coding techniques, GAs may be classified as: binary GA, real-valued GA, permutation GA, and tree-Based GA (28, 31).

2. 1. 2. Fitness Function The quality of chromosomes in each generation is determined by the fitness function, which provides a non-negative value for each chromosome based on its ability to solve the main problem. Chromosomes with higher fitness values are more likely to be chosen for the next generation (28, 29).

2. 1. 3. Selection The selection operator in genetic algorithms picks out chromosomes from the present population to act as parents for generating the next generation. Its primary objective is to enhance the quality of the population by selecting worthy chromosomes that will produce superior offspring. The selection process is based on the fitness function, which means that individuals with higher fitness values have a greater chance of being chosen. Various selection techniques are employed, such as roulette wheel, tournament, and random selection (28, 32).

2. 1. 4. Crossover The function of the crossover operator is to simulate inheritance in successive generations. In order to create the next generation, two chromosomes are chosen from the population as parents, and their chromosomes are merged using the crossover operator, resulting in two offspring. The probability of performing the crossover operator on the entire population is not fixed and is typically set between 60%-90% in SGA. Various methods of crossover exist, such as single point, double-point, and linear crossover, which vary depending on the coding and chromosome type (28, 32, 33).

2. 1. 5. Mutation The mutation operator is used to modify certain genes on a chromosome in order to explore new areas of the search space and discover fresh solutions. This operator has the ability to change the search direction in GA and consequently, this operator can prevent it from getting stuck in local optima. Similar to the crossover, the mutation operator is applied randomly to some chromosomes with a probability (p_m). In SGA, this probability is lower than that of the crossover and typically around 20%. The specific type of mutation employed may vary depending on the coding and chromosome specifications outlined in the algorithm (28-33).

2. 1. 6. Replacement By utilizing crossover and mutation operators, new offspring are generated. During the replacement process, a determination is made regarding which offspring should be included in the population and which parent should be replaced. There are two primary replacement techniques: "Generational Update" and "Steady State Update". In Generational Update, the number of chromosomes in the population determines the number of offspring produced. This method replaces the previous generation entirely and creates a new population. In Steady State Update, offspring are added to the next-generation population as soon as they are produced. To allow a new chromosome to join the population while maintaining a constant size in SGA, some chromosomes from the existing population must be removed (28, 32).

2. 1. 7. Stop Condition A genetic algorithm is an evolutionary algorithm that operates through iterations and necessitates a stopping point. Various stopping points include the number of generations, constant best fitness for a predetermined number of generations, reaching a predetermined level of fitness, duration of time, and others.

2. 1. 8. Heuristics The genetic algorithm involves creating an initial population randomly, using inheritance (crossover), allowing for generational change (mutation), and natural selection. Depending on the problem at hand, heuristic functions and operators can be utilized in each of these processes to increase efficiency in finding the optimal solution or avoiding local optima. The effectiveness of these heuristics is heavily influenced by factors such as the problem's nature, coding type, fitness function, and other genetic algorithm operators. Examples of commonly used heuristics include generating an initial population within the feasible set that is likely to be close to solutions, elitism replacement, and fascism replacement [23-24].

2. 2. Classification of Genetic Algorithms Previously, it has been noted that genetic algorithms may be classified based on their coding techniques. In general, genetic algorithms can be classified based on various concepts. In terms of implementation, they can be divided into two major categories: sequential and parallel. The sequential ones themselves can be classified as generational, steady-state, and messy (32).

However, genetic algorithms can also be classified based on their evolutions over the generations. In this case, they can be classified into adaptive and non-adaptive genetic algorithms. Adaptive genetic algorithms use feedback from the environment to adjust their parameters and improve performance over time. Non-adaptive genetic algorithms do not change their parameters during the optimization process (27).

3. SIMPLE GENETIC ALGORITHM

The Simple Genetic Algorithm (SGA) is a widely used type of GA that is easy to implement and has an acceptable performance. Several methods have been proposed to implement SGA. The main objective of this study is to implement AEGA, so we start by implementing SGA and then develop it into the Adam-Eve GA, that is to say the SGA presented here has been the basis of the proposed AEGA. Therefore, we need first to describe the details of the implemented SGA.

The flowchart of SGA can be seen in Figure 1. It is assumed that the coding type and fitness function are *a priori* defined. The proposed SGA starts by defining input parameters and then carries out four procedures: `P0`, `P1`, `P2`, and `T`, which generate the initial population, perform crossover, mutation, and replacement, respectively. The next subsections will provide a description of the SGA algorithm.

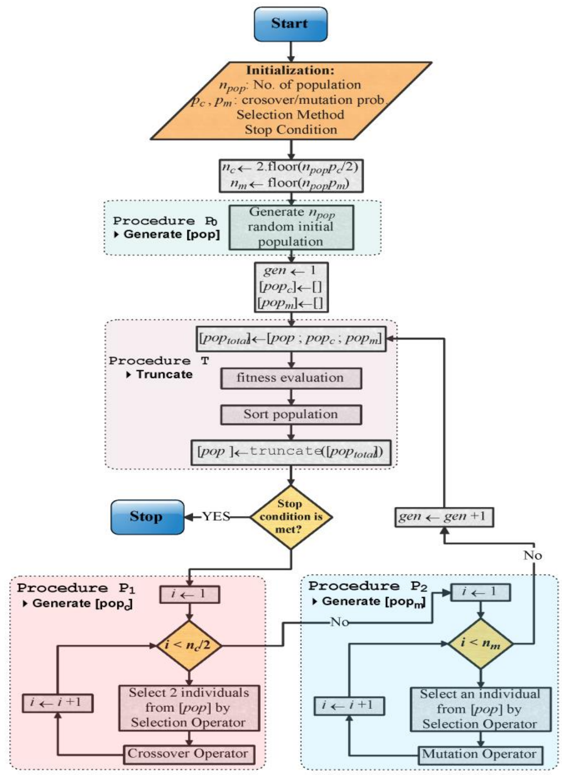


Figure 1. Flowchart of SGA with steady sate update replacement (27)

3. 1. Initialization To begin, the initial values for SGA must be specified. This includes determining the selection function, stop condition, population size (n_{pop}), crossover probability (p_c), and mutation probability (p_m). The proposed SGA utilizes the steady-state update for replacement. Consequently, during the crossover and mutation process, a certain number of offspring, n_c and n_m respectively, are generated as follows.

$$n_c = 2\left\lfloor \frac{n_{pop} \cdot p_c}{2} \right\rfloor, \quad n_m = \lfloor n_{pop} \cdot p_m \rfloor \tag{1}$$

where, ⌊ . ⌋ is floor function, and due to the creation of two offspring in crossover, the value of n_c is assumed to be even.

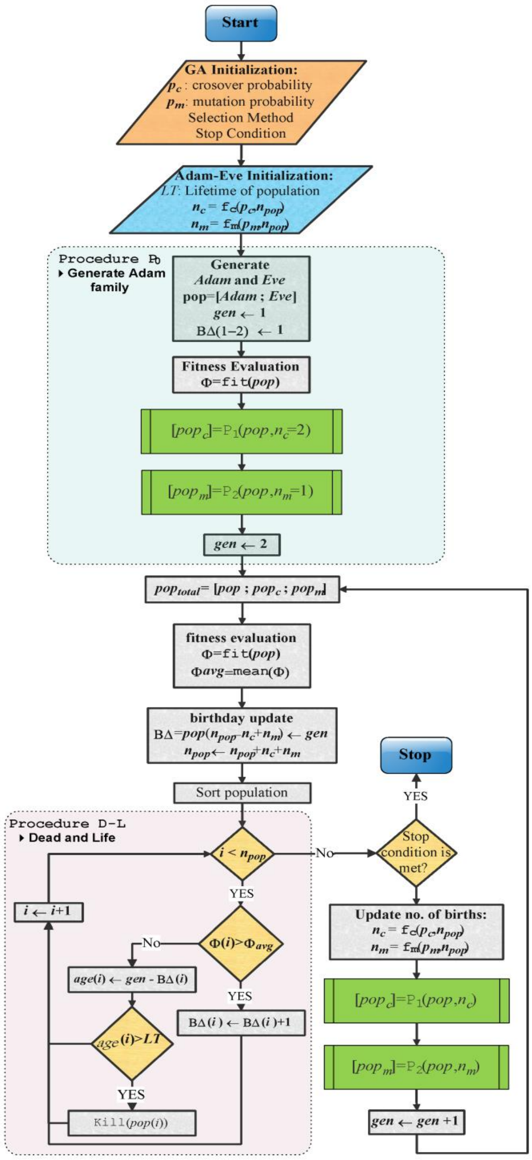


Figure 2. Flowchart of proposed Adam–Eve GA (27)

3. 2. Initial Population and Total Population In the first generation, the n_{pop} number of parents is randomly generated using the `P0()` procedure (population [pop]). At this stage, both the population of crossover offspring ([pop_c]) and mutated offspring ([pop_m]) are empty. The total population ([pop_{total}]) is obtained by combining these three populations. It is important to note that in the first generation; [pop_{total}] is equal to [pop].

As previously mentioned, one of the heuristics in GA involves generating a portion of the initial population within the feasible set area. This approach increases the likelihood of being closer to potential solutions and greatly impacts convergence speed and performance.

3. 3. Evaluation, Sorting, and Truncation The fitness function is used to evaluate the members of the population created in each generation. These members are then sorted based on their fitness values, from best to worst. In the proposed SGA, the steady state update method is used as the replacement method. From the total population ([pop_{total}]), which consists of n_{total}=n_{pop}+n_c+n_m members, n_{pop} members need to be chosen. This selection is done through the `T()` procedure, known as Truncation. The truncation allows for the implementation of *elitism* and *fascism* heuristics. By defining where n_{pop} members are selected from the sorted population, these heuristics can be applied.

3. 4. Crossover and Mutation Procedures The `P1()` and `P2()` procedures generate populations of crossover and mutated offspring respectively. The offspring are stored in [pop_c] and [pop_m], with n_c and n_m representing their respective numbers. As mentioned earlier, the crossover loop produces two individuals in each run, so n_c is always an even value and the loop iterates n_c/2 times.

4. ADAM-EVE LIKE GA AS AN ADAPTIVE GA

The Adam-Eve genetic algorithm is inspired by the biblical story of Adam and Eve, in such a way that the initial population is created by two individuals, referred to as Adam and Eve. These individuals are then used to generate the rest of the population through a process of crossover and mutation. Indeed, the concept of using two initial individuals to generate a population in genetic algorithms has been around since the early days of genetic algorithm research in the 1970s and 1980s. The term “Adam-Eve Genetic Algorithm” may have been coined later as a way to describe this approach.

This algorithm begins with two parents (i.e. Adam and Eve) and as new offspring are born in each generation, the population grows. However, the lifespan of each member of the population must be taken into account as it is natural for individuals to die after their lifetime. This helps to control the rate of population growth and allows for adjustments to be made to keep the population constant. In other words, by adjusting the Adam-Eve parameters effectively and through multiple generations, the number of births is almost balanced by the number of deaths.

After surveying the Adam-Eve algorithm and comparing it to the SGA proposed earlier, it can be inferred that modifying the initial population (`P0()`) and truncation (`T()`) procedures can convert SGA into the Adam-Eve algorithm. Additionally, each generation produces new offspring (n_c + n_m) which can be used as parameters for dynamic population control. Upon reviewing the AEGA process, it is evident that this algorithm is adaptive.

The proposed AEGA includes some variables and some procedures, some of which are comparable to the SGA while others are new. The upcoming sections will outline how this algorithm will be put into action, and a visual representation of the algorithm can be seen in Figure 2.

4. 1. Initialization At the beginning of the AEGA algorithm, initialization is carried out, but unlike SGA, the population size (n_{pop}) is not specified due to the algorithm's characteristics. Apart from SGA, the following elements must also be initialized:

- LT: Life-Time of people;
- $n_c(p_c,n_{pop})$: The function of the number of offspring caused by the crossover;
- $n_m(p_m,n_{pop})$: The function of the number of offspring caused by the mutation.

4. 2. Creation of Adam Family AEGA differs from SGA in that it starts with Adam and Eve as the first generation, instead of generating a fixed size population randomly. The second generation is then created through two offspring: the one that is created by crossover operator and another that is created by mutation operator. Procedure `P0()` in Figure 2 illustrates this process.

4. 3. Population merging, evaluation, birth registration, and life and death process of Adam Family First, it is necessary to document the birth details of the descendants in their official record. Then a new generation is created by combining the newly born offspring with their parents. After updating the population and similar to the SGA, it is crucial to assess the fitness of the population. The following are then updated:

- $\phi(i)$: the fitness of i^{th} individual in the population;
- ϕ_{avg}: average of the population fitness;
- $B\Delta(i)$: birth detail of i^{th} descendants in the population.

Next up is the Death and Life process (aka `D-L()` procedure), which involves comparing the fitness levels of individuals to the population average (i.e., ϕ_{avg}). Those with above-average fitness are granted more opportunities for life. After that, each member's age is determined using the following method:

$$\mathrm{age}(i) = \mathrm{gen} - \mathrm{B}\Delta(i) \tag{2}$$

where, gen is the current generation number, $B\Delta(i)$ is the generation in which the i^{th} individual was born, and age(i) is the age of the i^{th} individual in the population.

Finally, individuals who have surpassed a certain age threshold (age>LT) are ultimately removed from the population.

5. PROPOSED ADAM–EVE GA FOR WAMS PLANNING

In this section, we will first formulate WAMS comprehensive design as an optimization problem. Then we describe how we have implemented the proposed adaptive genetic algorithm from the previous section to solve this problem.

5. 1. Problem Formulation The power grids can be expressed by weighted undirected graph. Let $G(V,E,w)$; w:→R^+ be a weighted undirected graph representing the power grid, where V denotes the electrical buses (i.e. $B_i \in V$) and $ij \in E$ represents the transmission line connecting buses i to j with the length of w_{ij}. The number of buses is considered as the grid size ($|V|=n$), and number of grid connections is denoted by $|E|=M$.

The n-bus power graph can be expressed by n-squared adjacency matrix, where A_w is the weighted one, that is the i^{th} and j^{th} element of this matrix is considered to be w_{ij}, and A is its unweighted version.

The WAMS implemented in $G(V,E,w)$ can be represented by its “measurement infrastructure” and “communication infrastructure”, expressed in Equation 3. The measurement components can be expressed by vector X described in Equation 3.a, which indicates the location of measurements in the system buses. The communication infrastructure of WAMS can be represented by weighted graph $G_{OP}(V_{OP},E_{OP},w)$, as a subgraph of $G(V,E,w)$, where communication nodes, communication links, and length of the links are respectively represented by $h,k \in V_{OP}$; $hk \in E_{OP}$; and w_{hk}. Note that the lengths of a transmission line and its corresponding OPGW link are the same, that is $w=w_{hk}$. Thus, the WAMS can be graphically represented as follows (5):

$$W(G)=\begin{cases} X=[x_i]_{n\times 1} & (3.a)\\ G_{OP}(V_{OP},E_{OP},w),\ w:E_{OP}\rightarrow R^{+} & (3.b)\end{cases}$$

$$\text{such that:}\begin{cases} X\subseteq V_{OP} & (3.c)\\ V_{OP}\subset V, E_{OP}\subset E & (3.d)\\ G_{OP}\ \text{is connected} & (3.e)\end{cases}$$

where, n is the size of the network, X denotes PMU location in the system buses, and $G_{OP}(V_{OP},E_{OP})$ is OPGW subgraph.

In general, many applications can be performed by WAMS, such as damping of inter-area oscillation (33), but the main function of PMUs in the power grid has been state observation. The observability analysis indentifies that whether the entire power grid is observable with a set of PMUs or not. The topological observability analysis checks the observability of the entire system by graph theory concepts. The entire system is observable if there is no any zero element in the product of $A^+.X$, where $A^+=A+I_n$ and I_n is identity matrix of size n. It should be noted that "observation" in electrical engineering and "domination" in graph theory are equivalent concepts (5).

The comprehensive design of WAMS aims to place PMUs and OPGWs simultaneously. This can be formulated as the optimization problem as follows (5):

$$\underset{\substack{i\in V(G),\\ hk\in E(G)}}{\text{minimize}} \sum_{i=1}^{n} cp_i.x_i + \sum_{hk\in E_{OP}}\sum cf.w_{hk}.y_{hk} \quad (4.a)$$

subject to:

$$A^{+}.X \succ \hat{1} \quad (4.b)$$

$$V_{OP}\subset V, E_{OP}\subset E \quad (4.c)$$

$$X\subseteq V_{OP} \quad (4.d)$$

$$G_{OP}\ \text{is connected} \quad (4.e)$$

$$x_i, y_{hk}\in\{0,1\} \quad (4.f)$$

where, cp_i is the cost of PMU installation at the i^{th} bus, cf is the cost of installation for one kilometer of OPGW, first and second summations are respectively total costs of measurement and communication infrastructures of WAMS. $G_{OP}(V_{OP},E_{OP})$ is a subgraph of power graph G which demonstrates the routers i.e. $\{V_{OP}\}$ and transmission lines equipped by OPGW fiber, i.e. $\{E_{OP}\}$.

The solution of Equation 4 will be minimum connected dominating set of power graph $G(V,E,w)$, where the domination set denotes the PMU locations, and connections specify communication links in the power grids.

It is important to note, as mentioned earlier, that the main issue with MCDS is its multi-objective nature. It aims to minimize both the size of the dominating set and the length of connections between its members. In Equation 4, similar to other work (5, 6), the use of cp_i and cf allows us to transform the MCDS problem into a single-objective problem.

5. 2. Implementation of AEGA for WAMS Design

In this section, we will explain how we have implemented AEGA for the WAMS comprehensive design problem. Additionally, we will describe the heuristics used in the algorithm to improve speed and performance. Note that in adaptive GAs, due to their natures, i.e. adaptation, heuristics are more common than other GAs, especially SGA.

5. 2. 1. Chromosome Coding As explain before, power graph $G(V,E)$ (with the size $n=|V|$, and $M=|E|$ connections) can be expressed by adjacency matrix $A_{n\times n}$. The subgraph $G_{OP}(V_{OP},E_{OP})$ of graph G is also can be expressed by a adjacency matrix A_{OP}. Storing a solution in this way requires n^2 bits, which is not an optimal storage method. As power graph are usually sparse, in the proposed method a M-bit chromosome is suggested that represents connections in the G_{OP} subgraph. Coding and decoding routines enable the conversion of the chromosome to A_{OP} and vice versa.

5. 2. 2. Choice of Adam and Eve In AEGA, the choice of the first two parents (i.e., Adam and Eve) is of great importance. This choice causes AEGA to search the feasible space around these two parents, and their correct selection helps to quickly reach the local optimum. In other words, choosing Adam and Eve intelligently and close to a local optimum (whose its area is approximately known) guarantees finding an optimal solution within only a few generations after Adam and Eve, and this point is valuable in very large-scale problems such as WAMS design.

In the WAMS design, we are looking for the minimum connected dominating set in such a way that a minimum set specifying (aka PMU set) is obtained that are connected to each other, that is, to be connected to each other with the OPGW. On the other hand, we know that a "*Minimum Spanning Tree*" (MST) can be considered as a connected dominating set, because it includes all the nodes and is also connected; but it is not the minimum one. It is also clear that an MCDS does not contain any pendant nodes because it is easy to replace the only node connected to the pendant node and still maintain dominance and reduce the size of the connected subgraph. Based on the given explanations, we go through the following steps to find Adam and Eve:

1. We define the weighted graph $\tilde{G}=(\tilde{V},\tilde{E})$ in such a way that we remove all the pendant nodes from the weighted graph $G(V,E)$.
2. We extract the minimum spanning tree subgraph (which we call M_m) for the weighted graph $\tilde{G}=(\tilde{V},\tilde{E})$; Also, by removing the weight of the

edges of $\tilde{G}=(\tilde{V},\tilde{E})$ and extracting the minimum spanning tree for it, we reach the subgraph M_f.

3. Now we choose M_m as Adam and M_f as Eve.

5. 2. 3. Setting of Probabilities In general, the crossover operator makes big changes in the chromosomes of the parents, while the mutation operator is a type of fine tuning that makes small changes in the chromosome. The former is a time-consuming process, while the latter is executed faster. In the proposed algorithm, we provided the conditions that the crossover operator is executed with a higher probability at the beginning, and then after a few generations, the probability of crossover decreases and on the contrary, the probability of mutation increases, so that the search continues around the obtained best solution. To achieve this goal, the probability of p_c and p_m is defined as follows. Using these two probabilities, the number of people born by crossover and mutation (n_c and n_m) can be also calculated.

$$p_{c_{new}} = p_c\left(0.85^{\frac{-n_{pop}}{\alpha}}\right)$$
$$n_c = 2.\lfloor 0.5 n_{pop}.p_{c_{new}} \rfloor \tag{5}$$

where, n_{pop} is number of populations in the current generation. p_c and $p_{c_{new}}$ are respectively probabilities of crossover is the current and next generation. n_c is number of born individuals by crossover operator.

$$p_{m_{new}} = p_m\left(1.25^{\frac{-n_{pop}}{\alpha}}\right)$$
$$n_m = \lfloor n_{pop}.p_{m_{new}} \rfloor \tag{6}$$

where, p_m and $p_{m_{new}}$ are respectively probabilities of mutation is the current and next generation. n_m is number of born individuals by mutation operator.

5. 2. 4. Elitism In the genetic algorithm, elitism means giving more chances to elite people in the society. This action increases the speed of reaching the optimum, but at the same time increases the probability of getting stuck in the local optimum. In the proposed AEGA algorithm, it is done by increasing the lifespan of good people, which is fully described in **§4.3**.

5. 3. Innovation of Proposed Algorithm Compared to Previous One Both algorithms presented in the current research and (28) are Adam and Eve GAs, which are classified in the category of adaptive GAs; with this major difference that the algorithm presented in (28) is designed to solve OPP (i.e., minimum dominating set problem) and the current proposed algorithm to solve OPOP (i.e. minimum connected dominating set problem). As a result of this main difference, there are two basic differences in the algorithm implementation in here and our previous work. The first difference is that the adaptability only occurs in p_c during generations and p_m is a linear function of the population size, while in the current algorithm both p_c and p_m values are adaptive and defined to be exponential functions (5). The second difference is the use of innovative Adam and Eve in the current research, which makes the search process starts from areas closer to the optimal solution. These two mentioned differences cause the proposed algorithm to have the best speed while maintaining efficiency in solving MCDS, which will be shown in the next section with simulations.

6. CASE STUDY AND SIMULATION RESULTS

In order to show the ability of proposed method in WAMS design, IEEE 1354 test case is selected and MCDS problem has been solved by three different meta-heuristic algorithms; AEGA proposed in **§4,** SGA that is proposed in (6) and presented in **§3**, and ACO that is presented in (5). Note that ACO works based on pheromone and visibility of ants, in which evaporation is considered for pheromone. The terms α and β are relative importance power factors between pheromone and visibility function and the term ρ is the evaporation factor and usually is more than %90. For further information about ACO algorithm and its different parameters (α, β, ρ, and Q) we refer the readers to (5).

The parameters of different algorithms are shown in Table 1. We have set ACO parameters based on our previous findings (5). Also, parameter setting for SGA is based on our previous findings (6). Parameter setting in the proposed AEGA is based on Equations 5 and 6, which is discussed before, and the constant values are reported in Table 1. In order to provide same condition for all algorithms, we have used same heuristic in the initial populations of all examined algorithms, that is, we have putted two individuals obtained by the method presented in **§5.2.2** in their initial populations. The selection function is tournament selection in both GAs, the population size of SGA is set to be 70, and number of

TABLE 1. Parameter Settings for Different Algorithms

AEGA		SGA		ACO	
p_c	0.6	p_c	0.6	α	1
p_m	0.9	p_m	0.2	β	2
LT	2	n_{pop}	70	ρ	0.9
α	20	-	-	Q	1

ants set to be 20 in ACO. All algorithms have been iterated for 100 times and the evolutions of different parameters in different algorithms are examined during the iterations. Our observations show that AEGA is the most sensitive algorithm to exponential functions of p_c () and p_m (), and finding appropriate functions is another heuristic process. Meanwhile, the least sensitive algorithm is SGA, which has constant values of p_c and p_m, and in most problems, same values are used.

The first parameter is the best solution in each iteration. The best solution is defined as the normalized value of OPGW coverage in percent, i.e., the total length of OPGW fiber for the best solution divided by total length of transmission lines in the power grid. As it can be seen in Figure 3(a), improvement in best solutions occurs first in AEGA (iteration #12), then in SGA (iteration #20), and finally in ACO (iteration #24). This can be considered due to the adaptiveness of AEGA, which adjusts the values of p_c and p_m at the beginning of the search in such a way that the search process is directed faster towards the optimal solution. The rate of improvement in ACO is the lowest among the investigated problems, which is a large-scale network. This is due to the nature of the ant colony algorithm, where the presence of a satisfactory pheromone path (Adam or Eve) results in a low probability of ants moving in new paths, even with high evaporation. Also, the improvement rate in SGA is better than AEGA, which is due to the large number of populations in SGA and spending more time to produce each generation. For this reason, the next parameter that we have evaluated is the production time of each generation.

Figure 3(b) shows the execution times for each iteration in the investigated algorithms (in minutes). It can be seen that the execution time in ACO is almost constant because the ants' tour in a graph with a fixed structure takes a constant time. However, in AEGA, time grows due to population growth, and in SGA, time grows due to the similarity of chromosomes and the process of elitism and fascism. It can also be seen that the execution time for AEGA is always less than that of SGA and about 10% to 30% less than the execution time of SGA. This justifies the low rate of improvement of the best solution in AEGA (shown in Figure 3(a)) because it is able to improve the best solution by spending less time. For example, in the time of 300,000 seconds, AEGA, ACO, and SGA respectively execute 64, 54, and 47 iterations and respectively obtain the best solutions equal to 44.116%, 44.697%, and 44.178%, and this proves that the AEGA algorithm has the better performance at the same execution time.

The next items, which are shown in Figure 3(c), are p_c and p_m probabilities that are confirm the adaptiveness of the proposed AEGA. The decrease of p_c and increase of p_m during different generations can be seen in this figure.

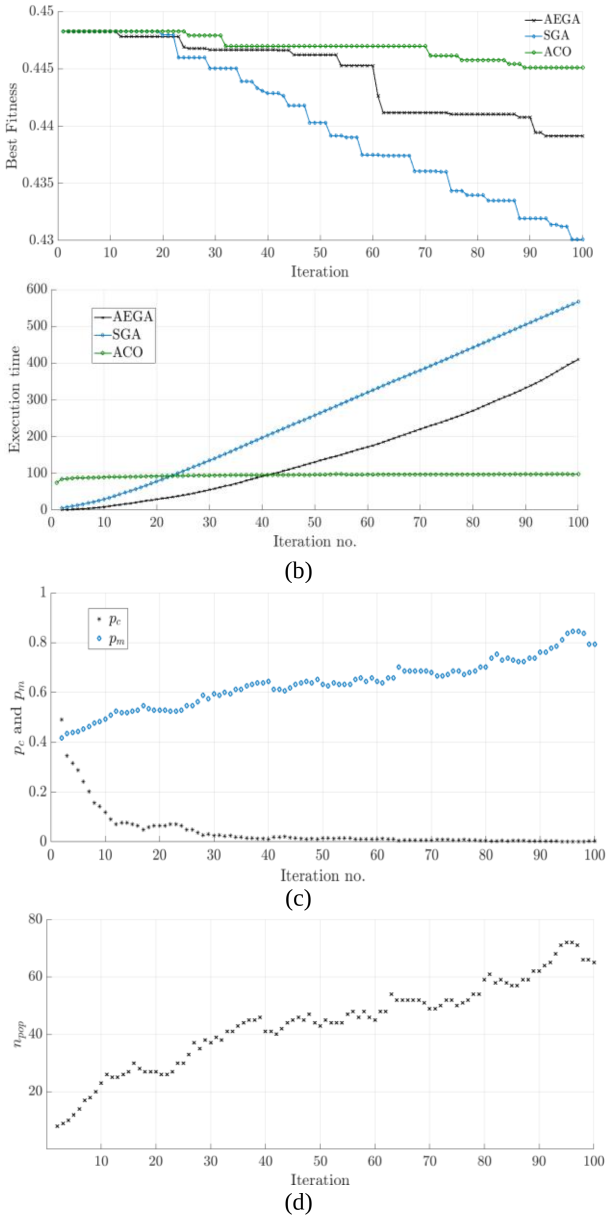


Figure 3. Evolutions of different parameters of Adam-Eve GA, classic GA, and ACO over different iterations: (a) Best fitness for both GAs and ACO; (b) Execution time for different iterations for both GAs and ACO; (c) Variations of p_c and p_m probabilities for different iterations in AEGA; (d) Number of populations over different generations in AEGA

Lastly, the population size in AEGA is shown in Figure 3(d) over different generations. As it can be seen, the population size grows over generations and is fixed at about 70 people in the 100th generation. Actually, this is when the birth and death rates in AEGA become almost equal, and it occurs around the 100th generation due to the adjustments we made in AEGA. That's why we set the population size in SGA to 70 so that the comparison conditions are the same for both GA algorithms.

The results are also shown in detail in Table 2. Considering the importance of changes in the initial generations, 10 iterations are first shown and then with a step of 10.

The closeness of the obtained solution to the optimal solution using the proposed algorithm is another point that should be quantified. Previously, it has been shown that in small and medium-scale power grids, ACO has the best recorded efficiency in finding the solution (5). However, in the current study, it loses its efficiency in large-scale grids due to its slowness. The best solutions for MDS (i.e. OPP reported by Shahraeini (5)) and MCDS (i.e. OPOP) problems are influenced by the structure of power grids and are normally scale-free. To prove this, we have solved the MCDS problem for three small-scale (30, 39, and 57 buses) and three medium-scale (118, 200, and 300) power grids with ACO and reported the results in Table 3. It can be observed that the coverage of edges belonging to MCDS (aka OPGW coverage) in power grids is scale-free and is regularly about 20% to 30% of the total length of power graph edges.

Having the vicinity of the optimal solution in the MCDS problem of the power grids, and considering provided information in **§5.2.2** that the proposed Adam and Eve are naturally shaped as "connected dominating sets" but not the minimum ones, it can be concluded that our heuristic approach in large-scale power grids finds two solutions that are relatively close to the optimal solution (44.828% reported in Table 2). Then, the next step is to implement an algorithm that can guide these solutions to the optimal solution in the shortest time with acceptable efficiency. The results presented in this study indicate that the proposed AEGA algorithm has the best

TABLE 2. Information of each iteration for the proposed AEGA and SGA

gen	Adam-Eve GA						Simple GA		ACO	
	n_{pop}	Best Fitness	Died	p_c	p_m	Execution Time (sec)	Best Fitness	Execution Time (sec)	Best Fitness	Execution Time (sec)
1	-	-	-	-	-	-	0.44828	181.4462	0.44828	4480.5533
2	8	0.44828	0	0.49127	0.41699	30.2831	0.44828	336.2431	0.44828	5049.0588
3	9	0.44828	2	0.34479	0.43469	70.234	0.44828	500.7388	0.44828	5124.0784
4	10	0.44828	2	0.31558	0.43924	115.8274	0.44828	635.3813	0.44828	5195.838
5	12	0.44828	1	0.28884	0.44383	163.4357	0.44828	797.9163	0.44828	5251.6817
6	14	0.44828	1	0.24198	0.45315	218.0468	0.44828	957.5509	0.44828	5247.2689
7	17	0.44828	0	0.20272	0.46268	278.5821	0.44828	1156.554	0.44828	5290.2011
8	18	0.44828	2	0.15544	0.47733	348.7935	0.44828	1334.2016	0.44828	5293.0065
9	20	0.44828	1	0.14227	0.48232	426.1163	0.44828	1548.3312	0.44828	5322.3575
10	23	0.44828	0	0.11919	0.49246	519.0996	0.44828	1767.5569	0.44828	5358.0243
20	27	0.44781	4	0.06414	0.52964	1765.4013	0.44796	4677.2578	0.44828	5531.1844
30	37	0.44666	4	0.024224	0.59381	3312.1561	0.44504	8089.361	0.44791	5648.0553
40	41	0.44666	8	0.011932	0.64531	5501.8556	0.44286	11812.5558	0.44697	5722.874
50	43	0.44622	4	0.014243	0.63203	7852.3392	0.44029	15479.6895	0.44697	5768.0213
60	45	0.44528	4	0.011932	0.64531	10300.7435	0.43747	19196.2619	0.44697	5806.1309
70	49	0.44116	5	0.0076646	0.67975	13217.683	0.43606	22819.7922	0.44697	5823.0418
80	59	0.44102	1	0.005877	0.70128	16200.4046	0.43396	26519.5783	0.44576	5824.9127
90	62	0.44076	3	0.0028947	0.76211	19920.6118	0.43192	30252.2871	0.44509	5841.1705
100	65	0.43914	6	0.0020316	0.79447	24565.5268	0.4289	33982.6806	0.44509	5846.1951

TABLE 3. Best Solution of MCDS for six small/medium grids

Test Case	OPGW Cov. (%)
30-bus	21.359
39-bus	37.285
57-bus	29.505
118-bus	23.357
200-bus	24.813
300-bus	15.672

speed to reach the solution while maintaining acceptable efficiency. However, the best solution is obtained with SGA at the cost of high time consumption. On the other hand, ACO lacks the necessary efficiency even with very high time consumption.

7. CONCLUSION

The WAMS comprehensive design aims to simultaneously place measurements and their required communication links. This can be formulated as an optimization problem and the output will be minimum connected dominating set, which has been a well-known graph problem and due to its NP-hardness, evolutionary algorithms are preferred for solving.

This study proposes an adaptive genetic algorithm to solve MCDS problem, which is knows as Adam-Eve like genetic algorithm. Designing an adaptive genetic algorithm is inherently more challenging than the simple genetic algorithm, and its success relies on the specific characteristics of the problem being studied, which is well done for MCDS problem in the current study.

We have designed the algorithm in such a way that it starts with two parents close to the optimum, and in the initial generations, they are more likely to crossover, and with the passage of generations, this probability decreases and, conversely, the probability of mutation increases in order to reach the optimum by fine tuning.

The simulation results for IEEE 1354 bus test network indicate that our AEGA has been up to 30% faster than SGA, while the efficiency of the algorithm is acceptable. This speed improvement can be used in other domain like wireless sensor networks, where the speed is more important than the efficiency. Also we have observed that ACO has not the capability of solving MCDS problems in the large-scale power graphs.

Our recent researches have demonstrated that the adaptive genetic algorithm is capable of enhancing the resolution of complex network problems, as evidenced by the speed and performance enhancement in simultaneous placement of WAMS components (i.e., PMUs and OPGWs) in the current study and the performance enhancement in PMU placement from our previous research.

Further investigation should be done to the appropriate selection of Adam and Eve, as well as the adaptability of the two functions of crossover and mutation and the process of death of individuals, in order to increase the efficiency of the proposed algorithm. The current study has been only focus on the WAMS comprehensive design, while adaptive GAs can be also used in the sequential methods that have previously proposed for WAMS design.

8. ACKNOWLEDGMENT

This paper is an extended version of the conference paper which was selected as one of the best paper award finalists in *2022 8th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, Mazandaran, Iran.

9. REFERENCES

1. Maheswari M, Suthanthira Vanitha N, Loganathan N. Wide-area measurement systems and phasor measurement units. Wide Area power systems stability, protection, and security. 2021:105-26. 10.1007/978-3-030-54275-7
2. Shahraeini M, Javidi MH, Haq Z. Wide area measurement systems. Advanced topics in measurements. 2012:303-22. 10.5772/35466
3. Maji TK, Acharjee P. Multiple solutions of optimal PMU placement using exponential binary PSO algorithm for smart grid applications. IEEE transactions on industry applications. 2017;53(3):2550-9. 10.1109/TIA.2017.2666091
4. Shahraeini M, Javidi MH, Ghazizadeh MS. Communication infrastructure planning for wide area measurement systems in power systems. International Journal of Communication Networks and Distributed Systems. 2013;10(4):319-34. 10.1504/IJCNDS.2013.054229
5. Shahraeini M, Kohsari G, Javidi MH, editors. Comparison of Meta-Heuristic Algorithms for Solving Dominating Set Problems in WAMS Design. 2022 8th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS); 2022: IEEE. 10.1109/ICSPIS56952.2022.10044037
6. Shahraeini M, Ghazizadeh MS, Javidi MH. Co-optimal placement of measurement devices and their related communication infrastructure in wide area measurement systems. IEEE Transactions on Smart Grid. 2012;3(2):684-91. 10.1109/TSG.2011.2178080
7. Kvasov DE, Mukhametzhanov MS. Metaheuristic vs. deterministic global optimization algorithms: The univariate case. Applied Mathematics and Computation. 2018;318:245-59. 10.1016/j.amc.2017.05.014
8. Colaço MJ, Dulikravich GS. A survey of basic deterministic, heuristic and hybrid methods for single-objective optimization and response surface generation. Thermal measurements and inverse techniques. 2011;1:355-405.
9. Khalkhali A, Narimanzadeh N, Khakshournia S, Amiri S. Optimal design of sandwich panels using multi-objective genetic algorithm and finite element method. International Journal of Engineering, Transactions C: Aspects 2014;27(3):395-402. 10.5829/idosi.ije.2014.27.03c.06
10. Arab R, Ghaderi S, Tavakkoli-Moghaddam R. Solving a new multi-objective inventory-routing problem by a non-dominated sorting genetic algorithm. International Journal of Engineering, Transactions A: Basics. 2018;31(4):588-96. 10.5829/ije.2018.31.04a.10
11. Khodadadi M, Kazemi A, Gorji M. Optimal thermodynamic design of turbofan engines using multi-objective genetic algorithm. International Journal of Engineering, Transactions C: Aspects. 2014;27(6):961-70. 10.5829/idosi.ije.2014.27.06c.15
12. Sivanandam S, Deepa S, Sivanandam S, Deepa S. Classification of genetic algorithm. Introduction to Genetic Algorithms. 2008:105-29. 10.1007/978-3-540-73190-0_5

13. Feyzbakhsh SA, Matsui M. Adam–Eve-like genetic algorithm: a methodology for optimal design of a simple flexible assembly system. Computers & Industrial Engineering. 1999;36(2):233-58. 10.1016/S0360-8352(99)00131-X
14. Li B, Zhang X, Cai S, Lin J, Wang Y, Blum C, editors. Nucds: An efficient local search algorithm for minimum connected dominating set. Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence; 2021. 10.24963/ijcai.2020/209
15. Mohammadi MB, Hooshmand R-A, Fesharaki FH. A new approach for optimal placement of PMUs and their required communication infrastructure in order to minimize the cost of the WAMS. IEEE Transactions on smart grid. 2015;7(1):84-93. 10.1109/TSG.2015.2404855
16. Ghasemkhani A, Monsef H, Rahimi-Kian A, Anvari-Moghaddam A. Optimal design of a wide area measurement system for improvement of power network monitoring using a dynamic multiobjective shortest path algorithm. IEEE Systems Journal. 2015;11(4):2303-14. 10.1109/JSYST.2015.2469742
17. Dubey R, Popov M, Muro JdJC. Cost effective wide area measurement systems for smart power network. IEEE Power and Energy Technology Systems Journal. 2018;5(3):85-93. 10.1109/JPETS.2018.2835646
18. Appasani B, Mohanta DK. Co-optimal placement of PMUs and their communication infrastructure for minimization of propagation delay in the WAMS. IEEE Transactions on Industrial Informatics. 2018;14(5):2120-32. 10.1109/TII.2018.2799659
19. Fan N, Watson J-P, editors. Solving the connected dominating set problem and power dominating set problem by integer programming. Combinatorial Optimization and Applications: 6th International Conference, COCOA 2012, Banff, AB, Canada, August 5-9, 2012 Proceedings 6; 2012: Springer. 10.1007/978-3-642-31770-5_33
20. Kucuktezcan CF, Genc VI. A new dynamic security enhancement method via genetic algorithms integrated with neural network based tools. Electric Power Systems Research. 2012;83(1):1-8. 10.1016/j.epsr.2011.09.004
21. Kaya M. The effects of two new crossover operators on genetic algorithm performance. Applied soft computing. 2011;11(1):881-90.
22. Hassanat A, Almohammadi K, Alkafaween Ea, Abunawas E, Hammouri A, Prasath VS. Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. Information. 2019;10(12):390. 10.3390/info10120390
23. Michalewicz Z. Heuristic methods for evolutionary computation techniques. Journal of Heuristics. 1996;1:177-206. 10.1007/BF00127077
24. Tovey CA. Nature-inspired heuristics: Overview and critique. Recent advances in optimization and modeling of contemporary problems. 2018:158-92. 10.1287/educ.2018.0187
25. Putnins M, Androulakis IP. Self-selection of evolutionary strategies: adaptive versus non-adaptive forces. Heliyon. 2021;7(5). 10.1016/j.heliyon.2021.e06997
26. Alexandrov IA, Kirichek AV, Kuklin VZ, Chervyakov LM. Development of an Algorithm for Multicriteria Optimization of Deep Learning Neural Networks. HighTech and Innovation Journal. 2023;4(1):157-73. 10.28991/HIJ-2023-04-01-011
27. lal Rajora G, Sanz-Bobi MA, Domingo CM. Application of Machine Learning Methods for Asset Management on Power Distribution Networks. Emerging Science Journal. 2022;6(4):905-20. 10.28991/ESJ-2022-06-04-017
28. Shahraeini M, Soltanifar R, editors. Performance Comparison between Simple and Adam—Eve-Like Genetic Algorithms in Optimal PMU Placement Problem. 2022 8th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS); 2022: IEEE. 10.1109/ICSPIS56952.2022.10043948
29. Holland J. Adaptation in neural and artificial system. Ann Arbor, Univeristy of Michigan Press. 1975.
30. Hasibuan A, Siregar WV, Isa M, Warman E, Finata R, Mursalin M. The Use of Regression Method on Simple E for Estimating Electrical Energy Consumption. HighTech and Innovation Journal. 2022;3(3):306-18. 10.28991/HIJ-SP2022-03-06
31. Goldberg D. Genetic Algorithms in Search, Optimization and Machine Learning,. Addition-Westly. Reading MA. 1989.
32. Deepa S, Sivanandam S. Introduction to genetic algorithms: Springer; 2010.
33. Asghari R, Mozafari S, Amraee T. Delay-scheduled controllers for inter-area oscillations considering time delays. International Journal of Engineering, Transactions B: Applications. 2018;31(11):1852-61. 10.5829/ije.2018.31.11b.08

COPYRIGHTS

©2024 The author(s). This is an open access article distributed under the terms of the Creative Commons Attribution (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, as long as the original authors and source are cited. No permission is required from the authors or the publishers.

Persian Abstract

چکیده

سیستم اندازه‌گیری ناحیه گسترده (WAMS) از دو زیرساخت مختلف اندازه‌گیری و مخابراتی تشکیل شده است که وظیفه اندازه‌گیری داده‌های شبکه در ناحیه گسترده و ارسال و پردازش آنها در مراکز کنترل را بر عهده دارد. طراحی WAMS می‌تواند شامل طراحی هر کدام از زیرساخت‌های آن باشد و یا هر دو زیرساخت را همزمان هدف قرار دهد که به طراحی جامع WAMS مشهور است. طراحی جامع WAMS به معنی جانمائی همزمان ادوات اندازه‌گیری و ارتباطات مورد نیاز آن است که در تئوری گراف‌ها به عنوان مجموعه متصل غالب کمینه (MCDS) شناخته می‌شود و در قالب یک مساله بهینه‌سازی فرمول‌بندی می‌شود. حل چنین مساله بهینه‌سازی پیچیده‌ای غالبا با الگوریتم‌های جمعیت‌گرا مانند الگوریتم ژنتیک انجام می‌شود و سرعت و کارائی یافتن پاسخ همواره یک چالش بوده است. این پژوهش یک الگوریتم ژنتیک تطبیقی موسوم به الگوریتم آدم و حوا پیشنهاد می‌کند که قابلیت حل مساله MCDS که از طراحی جامع WAMS ناشی شده را دارد. از طریق نتایج شبیه‌سازی برای شبکه IEEE ۱۳۵٤ باس نشان می‌دهیم که الگوریتم پیشنهادی به خوبی برای حل MCDS مرتبط با گراف‌های قدرت طراحی و تنظیم شده است. این الگوریتم ۳۰٪ سریعتر از الگوریتم ژنتیک ساده است، مسائل مقیاس-بزرگ را به طور موثر حل می‌کند، و در یک بازه زمانی معین از الگوریتم ژنتیک ساده و الگوریتم کلنی مورچه‌ها بهتر عمل می‌کند.