An Analysis of Unconventional Grid Sizes for Wind Farm Layout Optimization Using a Genetic Algorithm

Mohamed Mohandes
Professor
Electrical Engineering Department, and Interdisciplinary Research Center for Renewable Energy and Power Systems Engineering (IRC-REPS)
King Fahd University of Petroleum & Minerals Dhahran 31261
Saudi Arabia

Salman A. Khan
Professor
College of Computing & Info. Sciences
Karachi Institute of Economics and Technology Karachi
Pakistan

Shafiqur Rehman
Research Engineer II (Associate Professor)
Center for Engineering Research
Research Institute
King Fahd University of Petroleum & Minerals Dhahran 31261
Saudi Arabia

Ali Al-Shaikhi
Professor
Electrical Engineering Department, and Interdisciplinary Research Center for Communication and Systems Sensing (IRC-CSS)
King Fahd University of Petroleum & Minerals Dhahran 31261
Saudi Arabia

Kashif Iqbal
Graduate Student
College of Computing & Info. Sciences
Karachi Institute of Economics and Technology Karachi
Pakistan

Wind energy has emerged as an effective alternative to fossil fuels. At a commercial level, wind energy is harnessed through wind farms, which consist of several wind turbines arranged in a specific layout. Identifying the optimal layout of a wind farm is vital for maximum energy absorption from wind. Due to the sheer complexity of the problem, evolutionary and swarm intelligence algorithms have been employed for wind farm layout optimization. The performance of the optimal layout design is governed by several physical parameters, such as the wind speed, wind direction, and grid size. Most existing studies have focused on the effects of wind speed and wind direction on layout optimization. However, limited attention has been given to studying the impact of grid size. This study analyzes the effect of various unconventional grid sizes of higher dimension, ranging from 16×16, 17×17, up to 20×20, while using the genetic algorithm as the test bench. For a more comprehensive comparison, results of 10×10 to 15×15 were adopted from an earlier study. In contrast to previous studies, which focused on the assessment of the quality of results, a novel aspect of the present study also considers execution time as a performance measure. Another novel aspect is to analyze the performance of the generic algorithm for different grid sizes. Furthermore, unlike past studies, which mostly used hypothetical data, the present study employs real data from a potential site in Saudi Arabia. Results indicate that the maximum average conversion efficiency of 0.992 was obtained with a grid size of 19×19, while a grid size of 10×10 produced the minimum average conversion efficiency of 0.813. These results signify that an increase in the grid size positively affects conversion efficiency, which is used as a measure of the quality of the solution. However, the change in grid size has a negligible impact on the run time of the genetic algorithm. The findings of the study potentially pave the way for wind farm developers to select the best grid size for a given site while considering several practical issues, such as maintenance and operational costs, demographic and topographic structures, among other factors.

Keywords: *Wind farm optimization, Wind energy, Genetic algorithms, Optimization, Grid size*

1. INTRODUCTION

The last couple of decades have seen a tremendous gro–wth in the utilization of renewable energy. This growth is motivated by the adverse effects of the traditional, fossil fuels on the environment, as well as their quick depletion. Not only this, logistic and geopolitical issues in terms of availability of fossil fuels have also been a catalyst in shifting the focus of researchers towards renewable energy [1].

Among various renewable energy sources, wind energy has emerged as one of the most attractive areas of research and development [1-3]. The reason for the significant attention on wind energy utilization is due to its low development and operational costs, as well as its abundance globally. The access to wind energy is hardly affected by logistical issues or demographic boundaries, thus cutting the dependency on oil and gas-producing countries. Furthermore, environmental pollution in terms of greenhouse gases is also a minor concern with wind energy [4]. Although there are currently some issues in terms of the waste material produced by the decommissioned turbine blades, considerable attention is also given to coming up with environmentally fri–endly solutions to deal with this issue.

In order to utilize wind energy, wind farms need to be developed, where wind turbines are placed within the defined geographical boundaries of the farm. An

Received: June 2025, Accepted: July 2025
Correspondence to: Prof. Mohamed Mohandes
Electrical Engineering Department, King Fahd University of Petroleum & Minerals, Saudi Arabia
E-mail: mohandes@kfupm.edu.sa
doi: 10.5937/fme2503445M

efficient wind turbine system is governed by harmonization of several processes as highlighted by Rašuo et. al. [3]. These processes include design, materials and technology, manufacturing, verification testing, and regulations & standards [3]. Additionally, the importance of design, manufacturing, and verification testing for a turbine's performance is emphasized by Rašuo et. al. in several studies [5,6,7].

Currently, horizontal-axis wind turbines (HAWTs) are the most employed turbine types. Harnessing the maximum energy from wind primarily depends on how "well" the turbines are placed within the farm. Until this placement is done optimally, the wind absorption is not maximum. As such, this placement problem, generally known as the wind farm layout design (WFLD) problem, has been classified as NP-hard in the relevant literature. Accordingly, simple algorithms, such as linear search heuristics, cannot be employed. Therefore, researchers resort to algorithms inspired by various phenomena in nature. These algorithms are further classified into various sub-domains such as evolutionary computation (EC) and swarm intelligence (SI) algorithms [8]. These algorithms have proven to be effective in solving the WLFD problem. However, in contrast to simple algorithms, one major drawback of the EC and SI algorithms is their high computational time.

Any computational engineering problem has two aspects: the quality of the engineering design produced by the employed algorithm and the computational efficiency of the algorithm. Through the process of engineering design, the problem is defined, and potential solutions are generated. These solutions are then evaluated, and the best solution(s) are identified. Thus, the engineering design governs how a problem is modelled and the quality of solutions produced through the use of this problem model. As such, the engineering design aspect of the WFLD problem is concerned with how the problem is modelled and the quality of layout that is produced by an EC/SI algorithm. The other aspect focuses on the computational effort of the employed EC/SI algorithm. A plethora of studies have primarily focused on the first aspect, focusing on the quality of solutions produced by the EC/SI algorithm. This is quite logical since the main objective of the layout designer is to maximize energy generation or optimize other design requirements. However, the computational aspect has not been given due attention by researchers. From the computational point of view, it is also important to analyze how much execution time an EC/SI algorithm consumes. This importance is due to several factors. The more time an algorithm takes in execution, the more it consumes electricity, which is undesirable. Furthermore, the phenomenon of hardware stress has a negative impact on the processor. The more an algorithm utilizes the processor, the more it affects the remaining life of the processor [9,10]. Furthermore, when a processor is occupied by a task at hand, it cannot accept another task [11]. Considering the aforementioned issues, it is of utmost importance that an EA or SI be utilized in a way in which the execution time is minimized.

As mentioned earlier, the majority of existing EA and SI approaches focused on the quality improvement of the layout generated by the algorithms to optimize the optimization objective (such as power output, cost, etc.). The performance of these algorithms was evaluated under various testing scenarios, such as varying wind speeds and/or varying wind directions. To test the impact of these, traditionally, a grid-based layout configuration was used, which was typically arranged in a 10 × 10 configuration as shown in Figure 1. However, the impact of different grid sizes of higher dimensions, such as 16 × 16 up to 20 × 20, has not received due attention. This is evident from Table 1, which provides a chronological listing of all major studies covering the use of EC and SI algorithms for the WFLD problem. It is evident from the table that almost all studies focused on the use of a 10 × 10 grid in their analysis. To the best of our knowledge, there are only three studies that utilized other grid sizes. Two of these appeared in 2022, while the third is from 2024. These are Massoudi et. al. [12], which employed the genetic algorithm and analyzed 36 different grid sizes of disproportionate size (e.g., 10 × 9, 8 × 5, etc.). Another study was carried out by Koc [13], which made use of an invasive weed optimization algorithm while utilizing grid sizes of 10 × 10, 11 × 11, 12 × 12,..., 20 × 20. However, both the above studies, in addition to most others listed in Table 1, did not focus on the computational efficiency of the underlying algorithm with respect to grid size, which is a major novelty of the present study. Other novel aspects of the study are the analysis of the performance of the generic algorithm for different grid sizes and the use of real data from a potential site in Saudi Arabia. A recent study by Mohandes et. al. [14] carried out a preliminary investigation on the unconventional grid sizes while considering computational efficiency of the underlying algorithm, but they focused on grid sizes of 10 × 10 up to 15 × 15. As such, the present study extends the work of Mohandes et. al. [14] and analyzes the impact of higher dimension grid sizes on conversion efficiency and algorithm execution time.

Motivated by the above discussion, the major contributions of the present study are enumerated as follows:

1. The impact of various grid sizes is evaluated using quality of solutions produced and the executeon time of the employed algorithm. These grid sizes range from 16 × 16, 17 × 17, up to 20 × 20. Results are compared with an earlier study by Mohandes et. al. [14], which carried out an analysis of grid sizes ranging from 10 × 10 to 15 × 15.
2. A genetic algorithm is employed for the study. The reason for selecting GA is that it is the highest utilized algorithm for WFLD [8]. This is also evident from the coverage of the literature, as shown in Table 1. As per our knowledge, there is only one study that utilized GA for grid sizes greater than 10 × 10.
3. As opposed to many past studies, which utilized hypothetical data, the present study utilizes real data from a potential wind farm site in Turaif, a city located in northern Saudi Arabia.

The rest of the paper is organized as follows. Section 2 provides an overview of the WFLD problem, along

with the wake and cost models, as well as the optimization model. This is followed by a brief primer on the genetic algorithm in Section 3. Results and discussion are presented in Section 4. The paper ends with a conclusion and future directions in Section 5.

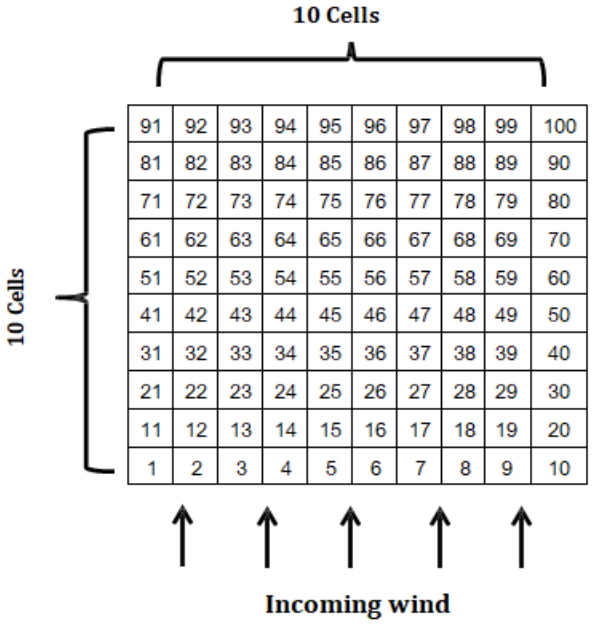


Figure 1. A wind with a grid size of 10 × 10.

2. OVERVIEW AND PROBLEM MODEL

The present study analyzes different grid sizes while employing a genetic algorithm as the test bench. Therefore, the focus of the study is neither on deve–loping a new problem model nor on the modification of the genetic algorithm. Therefore, the WFLD problem and associated concepts are briefly described in this section for a comprehensive view of the problem. First, the necessary basic information on the WFLD problem is provided. Then, a brief discussion on the wake models and the optimization function used in the study is provided. The section also gives a short overview of the genetic algorithm.

2.1 Wind Farm Layout Design Problem

In the WFLD problem, the aim is to place wind turbines within a wind farm in an optimal configuration. A wind farm is a geographical area within a defined perimeter having a wind corridor. In a wind farm, the turbines absorb the wind energy and convert it into electrical power.

As used in many studies, a conventional wind farm model assumes a discrete search space, where the layout is designed considering a square shape. This square is generally structured as an $x \times x$ grid. An example of a 10×10 grid is shown in Figure 1, which consists of 100 equally sized cells in a 10×10 grid configuration. These 100 cells result in 2^{100} possible solutions [34]. Similarly, 2^{121} possible solutions exist for 11×11, and so on. Placement of turbines is done at the center of a cell. Furthermore, the distance between two turbines in any direction is maintained at 3D to 5D (where D is the turbine diameter).

2.2 Wake Model

Several wake effect models have been proposed and employed over the years. In this research, a wake effect model used in a recent study by Ju and Liu [39] is adopted. The model assumes N turbines, which are to be placed in the wind farm. A unidirectional incoming wind with a speed of v_0 is assumed. The turbines in the front row (cells 1 to 10 in Figure 1 are directly exposed to incoming wind and therefore are not influenced by any wake effect. As a result, wind speed remains unaffected at these turbines. Turbines affected by the wake encounter a wind speed of v_i (i = 1, 2,. . . , N) and $v_i < v_0$. The value of v_i depends on whether a concerned turbine encounters a single wake or multiple wakes. If a turbine i is affected by the wake of another single turbine j, then the resulting wake at turbine i is mathematically calculated as follows:

$$v_{i,j} = v_0 \left[1 - \frac{2}{3} \left(\frac{R_j}{r_j} \right)^2 \right] \tag{1}$$

where $v_{i,j}$ denotes wind speed at wind turbine i under the wake effect of turbine j, and R_j is the rotor radius of wind turbine j. Furthermore, r_j is the wake radius and is represented as follows.

$$r_j = R_j + \alpha d_{i,j} \tag{2}$$

In Eq. (2), α represents the entrainment factor while $d_{i,j}$ is the downward distance.

If a turbine i is affected by multiple wakes, the wake is calculated using the following equation.

$$v_i = v_0 \left[1 - \sqrt{\sum_{j \in \Phi_j} \left(1 - \frac{v_{i,j}}{v_0} \right)^2} \right] \tag{3}$$

where Φ_i is a set that includes all indices of turbines that are upwind of turbine i. A detailed discussion on the single and multiple wake models can be found in the study by Ju and Liu [39].

2.3 Optimization Model

Since the purpose of the study is not to evaluate the optimization model or its effectiveness, only necessary details are provided here. Interested readers are referred to Ju and Liu [39] for further details.

It is assumed that the number of wind turbines in the wind farm is fixed, and the farm encounters a unid–irectional wind with a known direction. The aim is to harness wind energy with maximum efficiency. There–fore, the objective function is defined to maximize the conversion efficiency, represented as follows:

$$Conversion\ Efficiency = P_{current} / P_{total} \tag{4}$$

where $P_{current}$ is the total power generated by turbines under the wake effect in the current layout, and P_{total} is the ideal total power generated by all turbines without the impact of any wake.

Table 1. Summary of previous studies.

Reference	Year	Algorithm	Grid Size
Mosetti et. al. [15]	1994	Genetic Algorithm	10 × 10
Grady et. al. [16]	2005	Genetic Algorithm	10 × 10
Huang [17]	2007	Genetic Algorithm	10 × 10
Sisbot et. al. [18]	2009	Genetic Algorithm	10 × 10
Huang [19]	2009	Genetic Algorithm	10 × 10
Wan et. al. [20]	2009	Genetic Algorithm	10 × 10
Wang et. al. [21]	2009	Genetic Algorithm	10 × 10
Wang et. al. [22]	2009	Genetic Algorithm	10 × 10
Herbert-Acero et. al. [23]	2009	Genetic Algorithm	10 × 10
Emami and Noghreh [24]	2010	Genetic Algorithm	10 × 10
Kusiak and Song [25]	2010	Genetic Algorithm	10 × 10
Bilbao and Alba [26]	2010	Genetic Algorithm	10 × 10
Gonzalez et. al. [27]	2010	Genetic Algorithm	10 × 10
Rašuo et. al. [28,29]	2010	Differential Evolution	10 × 10
Saavedra et. al. [30]	2011	Genetic Algorithm	10 × 10
Kwong et. al. [31]	2012	Genetic Algorithm	10 × 10
Eroglu and Seckiner [32]	2013	Ant Colony Optimization	10 × 10
Yang et. al. [33]	2015	Genetic Algorithm	10 × 10
Rehman et. al. [34]	2016	Cuckoo Search	10 × 10
Afanasyeva et. al. [35]	2018	Genetic Algorithm, Cuckoo Search	10 × 10
Kirchner-bossi et. al. [36]	2018	Genetic Algorithm	10 × 10
Khanali et. al. [5]	2018	Genetic Algorithm	10 × 10
Charhouni et. al. [37]	2019	Genetic Algorithm	10 × 10
Wang [38]	2019	Genetic Algorithm	10 × 10
Ju and Liu [39]	2019	Genetic Algorithm	10 × 10
Ju et. al. [40]	2019	Genetic Algorithm	10 × 10
Gao et. al. [41]	2020	Genetic Algorithm	10 × 10
Wu et. al. [42]	2020	Particle Swarm Optimization	10 × 10
Wen et. al. [43]	2020	Genetic Algorithm	10 × 10
Liu et. al. [44]	2020	Genetic Algorithm	10 × 10
Rehman et. al. [4]	2020	Particle Swarm Optimization	10 × 10
Aggarwal et. al. [45]	2021	Various EC/SI algorithms	10 × 10
Al Sheriqi et. al. [46]	2021	Genetic Algorithm	10 × 10
Kirchner-bossi et. al. [47]	2021	Genetic Algorithm	10 × 10
Asfour et. al. [48]	2022	Genetic Algorithm	10 × 10
Guoqing et. al. [49]	2022	Genetic Algorithm	10 × 10
Masoudi et. al. [12]	2022	Genetic Algorithm	Various disproportional grid sizes smaller than 10 × 10
Koc [13]	2022	Invasive Weed Optimization	10 × 10 up to 20 × 20
Khan [50]	2022	Simulated Evolution	10 × 10
Mohandes et. al. [51]	2023	Genetic Algorithms	10 × 10

3. GENETIC ALGORITHMS

The Genetic algorithm (GA) is a non-deterministic iterative algorithm that is inspired by the theory of evolution. Originally proposed by Fraser [52] in 1957, the algorithm received notable attention through the work of Holland [53]. GAs operate on a set of solutions, known as a *population*. A solution in GA is denoted as a *chromosome* and consists of a string of individual elements called *genes*. In every iteration, a new set of

chromosomes is generated. These specific chromosomes are called *offspring*.

The search in GA is governed by two processes known as *exploration* and *exploitation* [54]. Exploration allows the algorithm to search through the solution space in the hope of discovering new potential regions in which optimal solution(s) could be found. The purpose of exploitation is to focus on an intensified search in a smaller region of the solution space. The balance between exploration and exploitation allows GA to carry out the search efficiently, leading to an optimal or near-optimal solution. A high exploration rate makes the search inefficient since the algorithm can miss good solutions. This results in a higher execution time to reach convergence. In contrast, a high exploitation rate can result in the loss of population diversity and may lead to premature convergence, thus resulting in low-quality solutions [55 - 57].

In GA, the *crossover* and *mutation* functions implement exploitation and exploration, respectively. The level of exploitation through crossover is controlled by a parameter called the 'crossover rate'. Similarly, exploration is controlled through another variable known as the 'mutation rate'. The values of these two parameters are user-defined. For GA to produce an optimal solution, sufficient time (and, equivalently, a large number of iterations) is required to reach convergence. If this is not ensured, then exploration and exploitation are not fully utilized, and the algorithm fails to produce an optimal solution.

4. RESULTS AND DISCUSSION

In this section, empirical results are presented and discussed to evaluate the performance of the various grid sizes, which include 16 × 16, 17 × 17, 18 × 18, 19 × 19, and 20 × 20. For the sake of comprehensiveness, results of 10 × 10 to 15 × 15 were adopted from a recent study by Mohandes et. al. [14]. For simulations, real data collected from a potential site of Turaif was used. The site is located in the northern part of Saudi Arabia at an elevation of 827 meters above sea level. Following an earlier study [58], a wind speed of 6.94 m/s at a height of 130 meters was used in the experimentation for Turaif.

After parameter tuning, the values of different parameters of the genetic algorithm were set as follows: population size = 30, crossover rate = 0.6, and mutation rate = 0.1. A fixed number of 20 turbines was used. GE 1.5sle turbine is employed as an example since it is frequently used in real wind farms [39]. Note that GA is a non-deterministic algorithm, suggesting that the result of a single run cannot lead to the best efficiency and therefore, a single run is not sufficient to reach a decision. As such, in accordance with the established approach suggested in the literature [8], 30 independent runs were carried out for all experiments, and results were reported as the average of the best efficiency in 30 runs as well as standard deviations of these 30 runs, with the corresponding execution times.

All experiments were done using the same initial population for GA. To give ample time for convergence, simulations were run for 2000 iterations. However, the results are reported considering the iteration in which the best result is first encountered. For example, if the best result is found in iteration number 900, then the algorithm would still run 2000 iterations, but the time at which 900 iterations were completed is reported.

In addition to the above, uniform simulation conditions (with regard to background processes and software platform) and hardware setup were used. Simulations were carried out using the open-source Python package developed by Ju and Liu [39], and necessary modifications were made to the code.

4.1 Effect of Grid Size on Conversion Efficiency

Tables 2 and 3 present the results of the conversion efficiency for the different grid sizes. The results are presented with respect to the best, worst, and average conversion efficiency (of 30 runs), as well as the standard deviation of the 30 runs. As can be seen from the two tables, the grid size of 19 × 19 produced the highest average conversion efficiency of 0.992 while maintaining a very low standard deviation of 0.001. In contrast, the worst results were generated by the grid size of 10×10 where an average conversion efficiency of 0.813 was obtained, with a standard deviation of 0.006, which was the highest among the results of all grid sizes.

Table 2. Results of conversion efficiency for grids of 10 × 10 to 15 × 15 (adopted from Mohandes et. al. [14]).

	10×10	11×11	12×12	13×13	14×14	15×15
Max Conv Eff.	0.832	0.866	0.895	0.917	0.918	0.953
Min Conv Eff.	0.801	0.846	0.882	0.909	0.908	0.948
Avg Conv Eff.	0.813	0.855	0.888	0.912	0.913	0.950
St dev.	0.006	0.004	0.003	0.002	0.002	0.001

Table 3. Results of conversion efficiency for grids of 16 × 16 to 20 × 20.

	16×16	17×17	18×18	19×19	20×20
Max Conv Eff.	0.965	0.977	0.985	0.994	0.991
Min Conv Eff.	0.962	0.968	0.983	0.991	0.964
Avg Conv. Eff	0.964	0.975	0.984	0.992	0.987
St dev.	0.001	0.002	0.001	0.001	0.005

Figure 2 provides further details about the different grid sizes. The figure illustrates the distribution of the best solution for 30 runs with regard to each grid size. These solutions are divided into five ranges of conversion efficiency. As observed from the figure, on the one hand, higher grid sizes (18×18, 19×19, and 20×20) had all solutions falling in the highest efficiency ranges of 0.98 and above. On the other hand, the lowest grid size of

10 × 10 had all solutions in the lowest efficiency range (0.8 to 0.84).

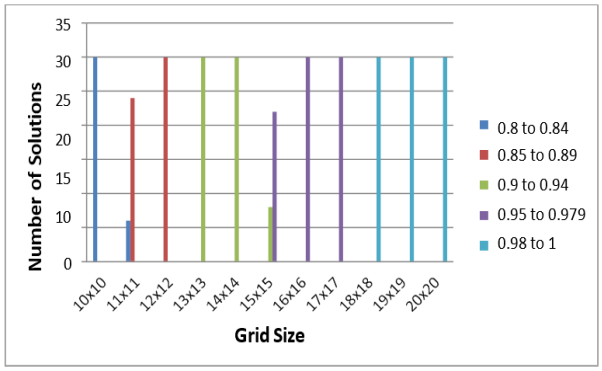


Figure 2. Conversion efficiency count for different grid sizes (results of 10 × 10 to 15 × 15 are adopted from Mohandes et. al. [11]).

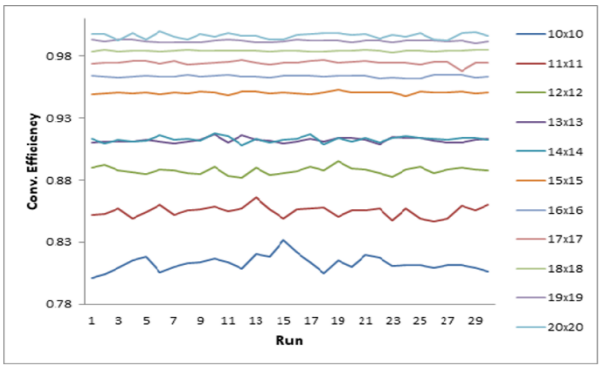


Figure 3. Conversion efficiency of 30 runs with different grid sizes (results of 10 × 10 to 15 × 15 are adopted from Mohandes et. al. [14]).

4.2 Effect of Grid Size on Execution Time

Concerning the impact of grid size on the execution time of GA, experiments were carried out considering the execution time per single run and execution time per iteration within a single run. With regard to execution time per single run, the results are summarized in Tables 4 and 5. The tables indicate that the highest execution time (average of 30 runs) of 3759.8 seconds was obtained for a grid size of 17 × 17, and as the grid size is increased beyond this, the average execution time decreases. Furthermore, the lowest average execution time was obtained for a grid size of 11 × 11. Another observation is that the grid size of 18 × 18 showed the highest standard deviation, which means that for this grid size, the iterations (which correspond to the execution time) at which the best results (of 30 runs) were obtained fluctuated tremendously. This is also obvious from the maximum and minimum execution times of 6738.2 seconds to 189.4 seconds, respectively.

Figure 4 gives a pictorial view of the trends on execution time, showing division of times into 7 different ranges, from 0 - 999 seconds, to 6000 - 6999 seconds. The illustration indicates that the execution times of all grid sizes are spread across all time ranges, more or less. As such, there is no clear trend whether the execution time increases or decreases with the variation in grid size. This observation is further confirmed by plots of execution times for 30 runs in Figure 5. In this figure, the run times for all grid sizes are sorted and plotted against each of the 30 runs. One common observation from the plots is that the run times are spread between very low to very high run times. Furthermore, all plots demonstrate a somewhat linear trend. However, a clear pattern is that the increase in grid size does not have a significant impact on the runtime, as the run times for all grid sizes vary more or less between 50 seconds and 6800 seconds.

Table 4. Results of execution times (in seconds) for grids of 10 × 10 to 15 × 15 (adopted from Mohandes et. al. [14]).

	10×10	11×11	12×12	13×13	14×14	15×15
Max Time	4027.3	5272.0	5201.6	6104.8	5323.9	4093.8
Min Time	196.1	187.0	187.6	372.4	89.7	190.4
Avg Time	2300.9	2222.5	2666.2	3255.6	2637.4	2372.9
St dev.	1259.7	1451.6	1409.2	1636.3	1537.1	1154.4

Table 5. Results of execution times (in seconds) for grids of 16 × 16 to 20 × 20

	16×16	17×17	18×18	19×19	20×20
Max Time	6243.6	5845.1	6738.2	6135.3	5448.8
Min Time	799.6	176.5	189.4	50.6	53.4
Avg Time	3471.7	3759.8	3212.4	3043.8	2396.8
St dev.	1824.6	1312.7	2021.6	1935.6	1379.9

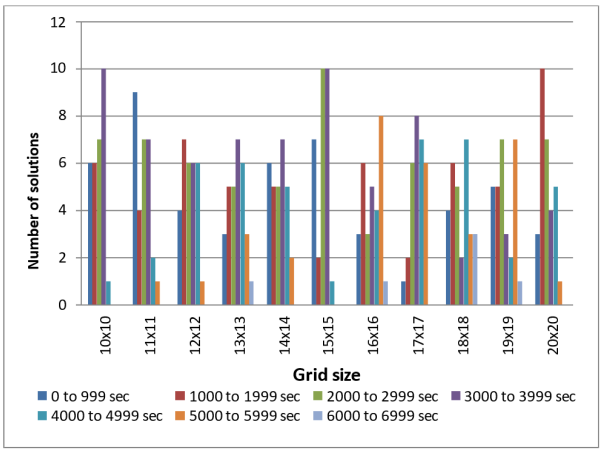
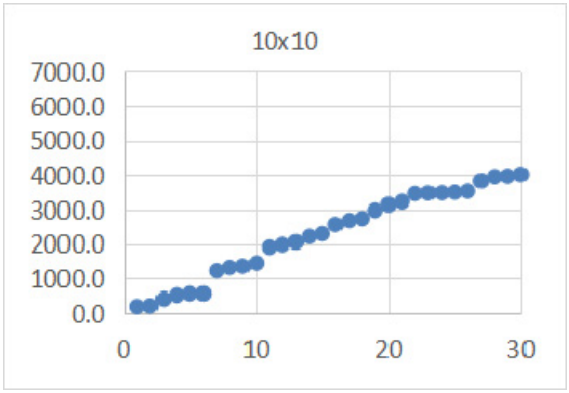


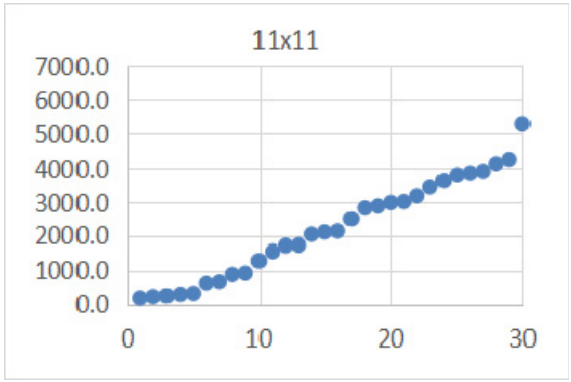
Figure 4. Execution time count for different grid sizes (results of 10 × 10 to 15 × 15 are adopted from Mohandes et. al. [14]).

The impact of grid size on execution time was further analyzed through the measurement of average execution time per iteration. That is, for each of the best 30 runs, where the total execution time is reported above, the average execution time per iteration was also measured. Tables 6 and 7 provide these averages along with the maximum and minimum execution time per iteration for different grid sizes.

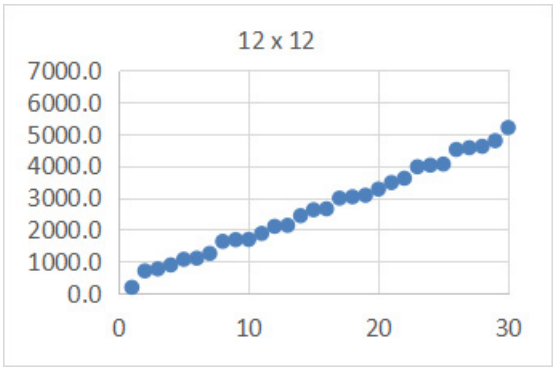
The tables indicate that the highest (average) exe–cution time of 3.24 seconds per iteration was observed for a grid size of 18 × 18, while the lowest (average) execution time per iteration was 2.06 seconds for a grid size of 10 × 10. Furthermore, the average execution time per iteration was most stable for grid size 15 × 15 with a standard deviation of 0.04, indicating that for each of the 30 runs for this grid size, the average runtime per iteration remained more or less the same.



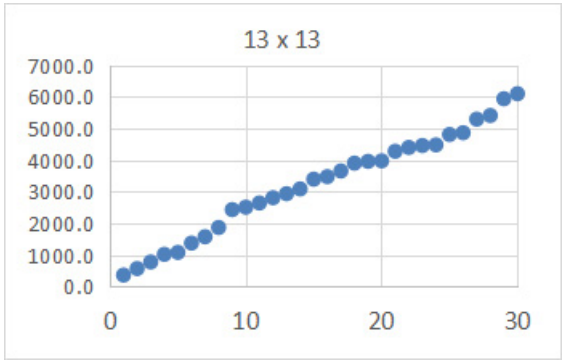
(a)



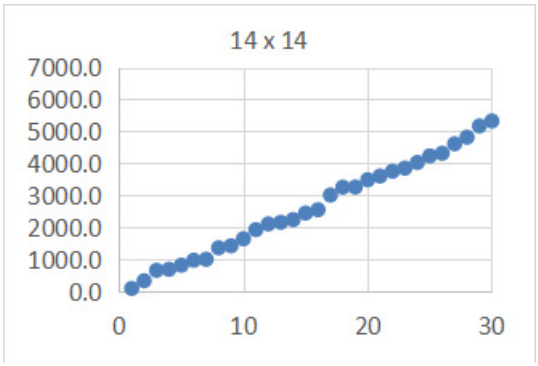
(b)



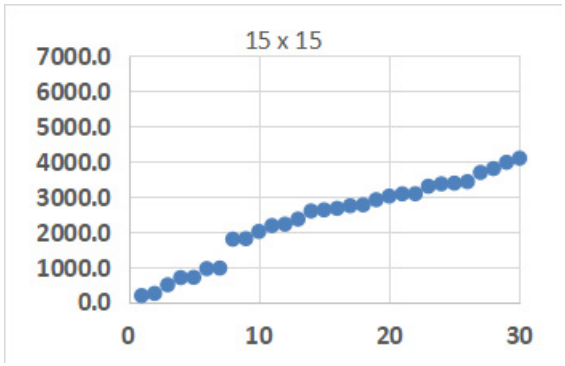
(c)



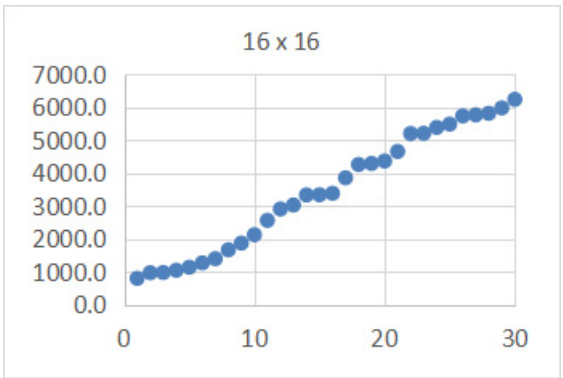
(d)



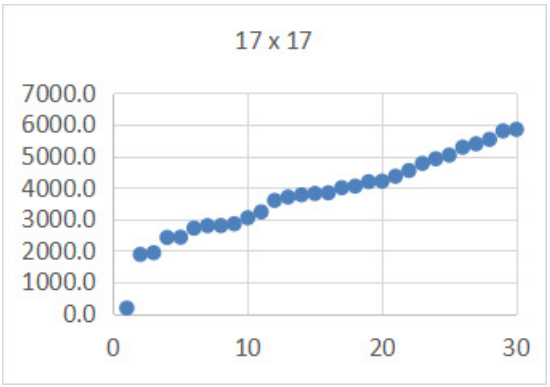
(e)



(f)

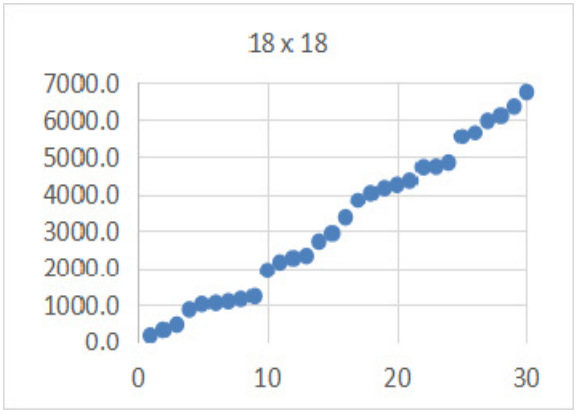


(g)

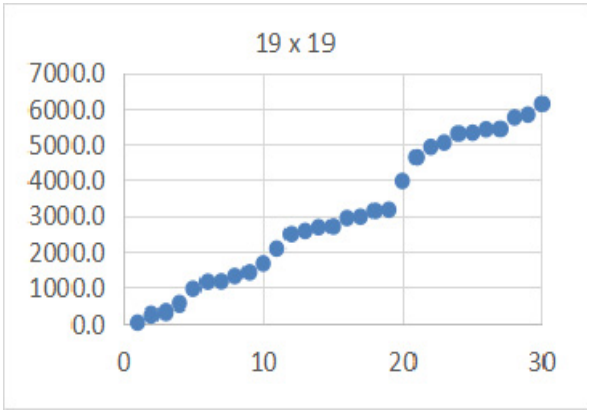


(h)

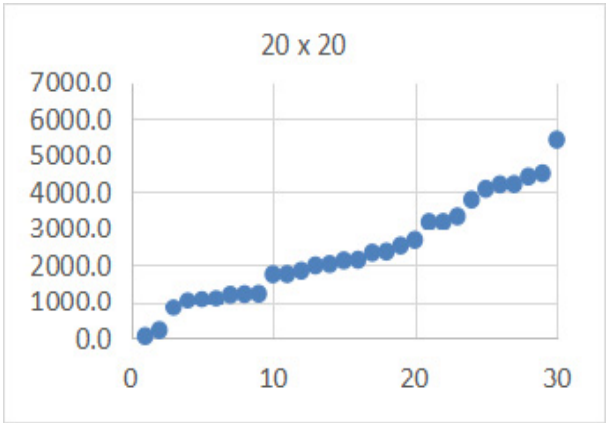
Figure 5. Runtime for 30 runs for different grid sizes. (a) 10 x 10 (b) 11 x 11 (c) 12 x 12 (d) 13 x 13 (e) 14 x 14 (f) 15 x 15 (g) 16 x 16 (h) 17 x 17.



(i)



(j)



(k)

Figure 5 (contd..). Runtime for 30 runs for different grid sizes. (i) 18 x 18 (j) 19 x 19 (k) 20 x 20.

In contrast, grid sizes 13 × 13 and 18 × 18 showed a high variation in execution time per iteration, since the standard deviation was the highest (0.27) for both of them. Overall, the table indicates that the grid size of 18 × 18 was the worst performer due to its highest exe–cution time per iteration, while indicating an unstable behavior with the highest standard deviation.

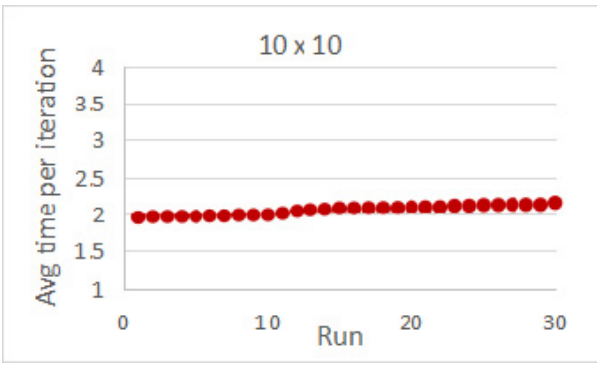
Figure 6 illustrates the iteration-wise run times for all grid sizes. The iteration-wise runtimes are sorted and plotted for each of the 30 runs. A general trend in the plots is that there is not much variation in iteration-wise run times for different grid sizes; the values vary between 2.06 and 3.86 seconds per iteration. The most stable behavior is for grid sizes of 10 × 10 and 15 × 15, where the iteration-wise run time remained almost constant for the 30 runs. This is also confirmed by the low standard deviations for the two grid sizes given in Table 6. Furthermore, for each grid size, a typical trend is that plots are more or less show a linear behavior.

Table 4. Results of execution times (in seconds) per iteration for grids of 10 × 10 to 15 × 15.

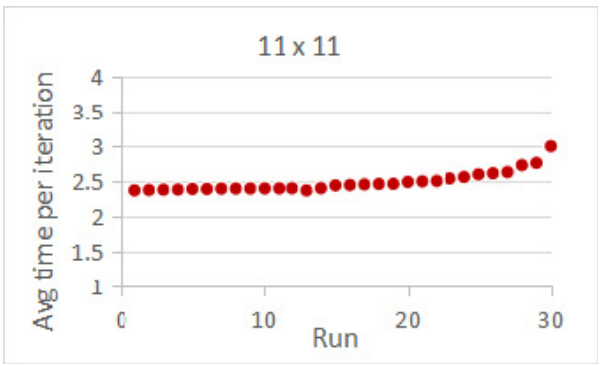
	10×10	11×11	12×12	13×13	14×14	15×15
Max Time	2.16	3.00	2.76	3.22	3.32	2.29
Min Time	1.96	2.37	2.13	2.29	2.36	2.08
Avg Time	2.06	2.49	2.47	2.70	2.81	2.16
St dev.	0.07	0.14	0.18	0.27	0.24	0.04

Table 5. Results of execution times (in seconds) for grids of 16 × 16 to 20 × 20.

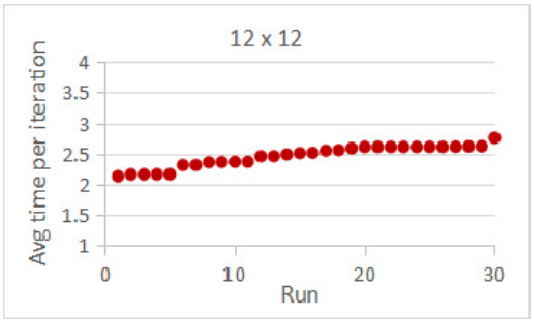
	16×16	17×17	18×18	19×19	20×20
Max Time	3.30	3.31	3.69	3.24	2.84
Min Time	2.89	2.60	2.89	2.29	2.27
Avg Time	3.06	3.09	3.24	2.86	2.43
St dev.	0.13	0.15	0.27	0.24	0.15



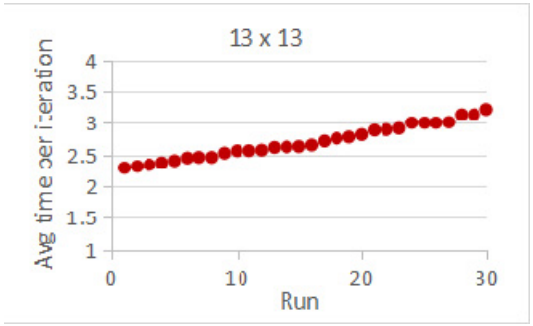
(a)



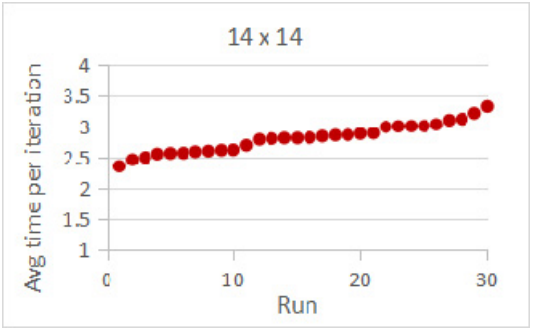
(b)



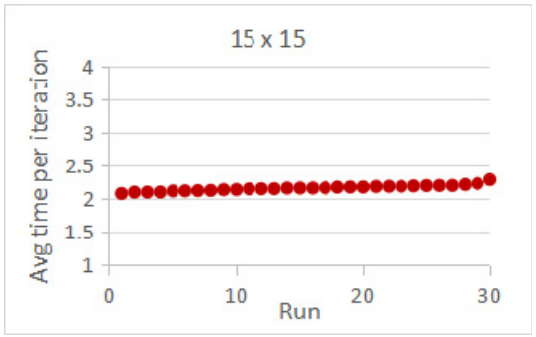
(c)



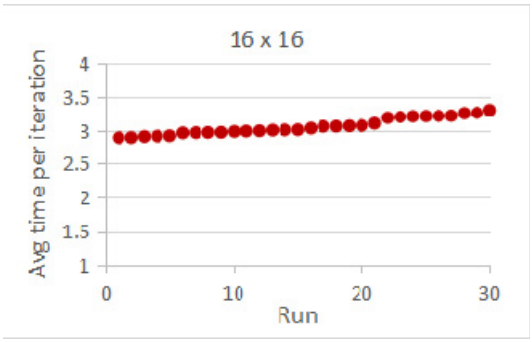
(d)



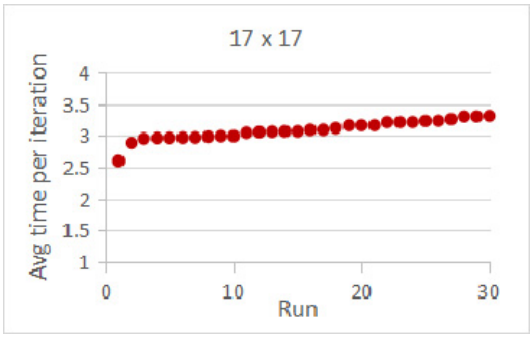
(e)



(f)

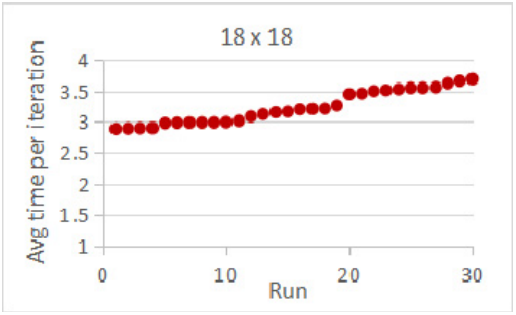


(g)

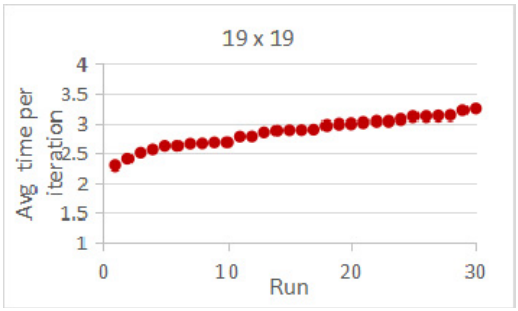


(h)

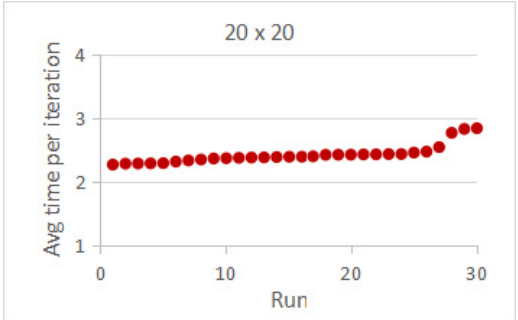
Figure 6. Average runtime per iteration (in seconds) for 30 runs for different grid sizes. (a) 10 x 10 (b) 11 x 11 (c) 12 x 12 (d) 13 x 13 (e) 14 x 14 (f) 15 x 15 (g) 16 x 16 (h) 17 x 17.



(i)



(j)



(k)

Figure 6 (contd..). Average runtime per iteration (seconds) for 30 runs for different grid sizes. (a) 18 x 18 (b) 19 x 19 (c) 20 x 20.

5. DISCUSSION

From the results in Sections 4.1 and 4.2, certain observations can be noticed. First, it is clearly observed that an increase in the conversion efficiency is directly proportional to the grid size. This is logical because a higher grid size gives more options for placement if the number of turbines is fixed (which is the case assumed in the present study). This allows the genetic algorithm to explore a higher number of possible solutions. The results indicated that the grid size of 19 × 19 produced the highest conversion efficiency. Surprisingly, the study by Koc [13] also revealed that the grid size of 19 × 19 also produced the best results when they carried out empirical performance evaluation of their proposed binary invasive weed optimization algorithm.

When it comes to the execution time, the trends do not go line in line with the trends on conversion efficiency. There is no clear trend in which grid size gives the minimum execution times; the execution time varied aggressively for all grid sizes. For each grid size, sometimes the optimal results were found very early during the execution (within the first 100 iterations), while in some cases the optimal results were found just near the end of 2000 iterations. Furthermore, the iteration-wise execution time did not show much variation with respect to the grid size. This indicates that, in general, the performance of GA in terms of execution time is not affected by the grid size. This could possibly be explained by the non-deterministic (stochastic) nature of GA. It is quite possible that, due to this stochastic behavior, the algorithm is able to explore the search areas that are closer to the optimal solution in some instances. Consequently, the algorithm would be able to reach the optimal solution in less time. In other cases, where the algorithm traverses search areas that are far from the optimal solutions, more execution time is required to reach the optimal solution. However, the grid size has a significant correlation with the conversion efficiency, which is due to the fact that with a higher grid size, the algorithm has more options to explore (the search space gets bigger), which allows the algorithm to perform exploration and exploitation more effectively.

6. CONCLUSION AND FUTURE DIRECTIONS

In harnessing energy from wind, the configuration of a wind farm plays a crucial role. One important factor that contributes to this configuration is the grid structure and size of the wind farm. While past studies have focused on many other aspects of wind farm design, research on the impact of grid size has not been carried out with due attention. The novel aspects of this study include the utilization of a genetic algorithm to study the impact of various grid sizes, both in terms of the quality of solution produced, which is measured by conversion efficiency, as well as the execution time of the genetic algorithm. The novelty also lies in the use of real data collected from a potential wind farm size in Saudi Arabia. It has been found that while grid size has a notable impact on the conversion efficiency, the execution time of the genetic algorithm is hardly dependent on the grid size.

In terms of the theoretical aspects, the findings in this study can be expanded into several dimensions. While the present study has utilized the genetic algorithm, other algorithms from the domain of evolutionary computation and swarm intelligence, such as differential evolution, particle swarm optimization, ant colony optimization, cuckoo search, among many others, can be studied. Different wind scenarios, considering multiple wind speeds and/or wind coming from multiple directions, can also be of interest for further research. In addition, different mathematical models with regard to wake and cost modelling can be employed and mutually compared.

From an application point of view, the relationship between different numbers of turbines and grid sizes can be studied. Furthermore, while the study considered a single turbine, several turbine models with various rated capacities can be evaluated. This will assist wind engineers in selecting the most appropriate turbine and its count for a specific grid size. The impact of grid size on other practical issues, such as maintenance and operation costs, as well as grid connectivity, can also be studied.

ACKNOWLEDGEMENT

The authors acknowledge King Fahd University of Petroleum & Minerals for supporting this study through IRC-SES grant # INRE 2217.

REFERENCES

[1] Rehman, S., Khan, S. A., and Alhems, L. M.: A rule-based fuzzy logic methodology for multi-criteria selection of wind turbines. Sustainability, Vol. 12, No. 20, p. 8467, 2020.

[2] Khanali, M., Ahmadzadegan, S., Omid, M., Keyhani Nasab, F., Chau, K.W.: Optimizing layout of wind farm turbines using genetic algorithms in Tehran province, Iran. International Journal of Energy and Environmental Engineering. Vol. 9, No. 4, pp. 399–411, 2018.

[3] Rašuo, B., Dinulović, M., Veg, A., Grbović, A., and Bengin, A.: Harmonization of new wind turbine rotor blades development process: A review. Renew. Sust. Energ. Rev.. Vol. 39, pp. 874-882, 2014.

[4] Rehman, S., Khan, S. A., and Alhems, L. M.: The effect of acceleration coefficients in particle swarm optimization algorithm with application to wind farm layout design. FME Trans., Vol. 48, No. 4, pp. 922-930. 2020.

[5] Dinulović, M. R., Trninić, M. R., Rašuo, B. P., and Kožović, D. V.: Methodology for aeroacoustic noise analysis of 3-bladed h-Darrieus wind turbine. Thermal Science, Vol. 27 (1 Part A), pp. 61-69, 2023.

[6] Rašuo, B. P., and Veg, A. D.: Design, fabrication, and verification testing of the wind turbine rotor

blades from composite materials, in: *Proceedings of the ICCM-16*, Japan, 2007, pp. 1-4.

[7] Parezanovic, V., Rašuo, B., and Adzic, M.: Design of airfoils for wind turbine blades. In: *Proceedings of the French-Serbian European Summer University: Renewable Energy Sources And Environment-Multidisciplinary Aspect,* 2006, pp. 195-200.

[8] Khan, S. A., and Rehman, S.: Iterative nondeterministic algorithms in on-shore wind farm design: A brief survey, Renew. Sust. Energ. Rev.. Vol. 19, No. 3, pp. 370–384, 2013.

[9] Gupta, T., Bertolini, C., Heron, O., Ventroux, N., Zimmer, T., Marc, F.: Impact of power consumption and temperature on processor lifetime reliability. Journal of Low Power Electronics, Vol. 8, No. 1, pp. 83–94, 2012.

[10] Yousuf, S., Khan, S.A., Khursheed, S.: Remaining useful life (RUL) regression using long–short term memory (LSTM) networks. Microelectronics Reliability, Vol. 139, p.114772, 2022.

[11] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT press, 2022.

[12] Masoudi, S.M., Baneshi, M.: Layout optimization of a wind farm considering grids of various resolutions, wake effect, and realistic wind speed and wind direction data: A techno-economic assessment. Energy, Vol. 244, p. 123188, 2022.

[13] Koc, I.: A comprehensive analysis of grid-based wind turbine layout using an efficient binary invasive weed optimization algorithm with Levy flight. Expert Systems with Applications, Vol. 198, p. 116835, 2022.

[14] Mohnades, M., Khan, S.A., Rehman, S., Al-Shaikhi, A., Liu, B., Iqbal, K.: A preliminary analysis of different grid sizes for wind farm micrositing using and evolutionary algorithm for Turaif, Saudi Arabia. In: *IEEE 6th International Symposium on Advanced Electrical and Communication Technologies*, 2024, pp. 1-6.

[15] Mosetti, G., Poloni, C., Diviacco, B.: Optimization of wind turbine positioning in large wind farms by means of a genetic algorithm. J. Wind Eng. Indust. Aerodyn. Vol. 51. pp.105–116. 1994.

[16] Grady, S.A., Hussaini, M,Y., Abdullah, M. M.: Placement of wind turbines using genetic algorithms. Renew. Energ., Vol 30, pp. 259–270, 2005.

[17] Huang, H.: Distributed Genetic Algorithm for Optimization of Wind Farm Annual Profits. In: *Proceedings of IEEE International Conference on Intelligent Systems Applications to Power Systems*, 2007, pp. 1–6.

[18] Şişbot, S., Turgut, O., Tunc¸, M., Çamdali, U.: Optimal positioning of wind turbines on Gökçeada using multi-objective genetic algorithm. Wind Energy, Vol. 13, No. 4, pp. 297–306, 2009.

[19] Huang, H.S.: Efficient hybrid distributed genetic algorithms for wind turbine positioning in large wind farms. In: *Proceedings of IEEE International Symposium on Industrial Electronics*, 2009, pp. 2196–2201.

[20] Wan, C., Wang, J., Yang, G., Li, X., Zhang, X.: Optimal micro-siting of wind turbines by genetic algorithms based on improved wind and turbine models. *In: Proceedings of the 48th IEEE Conference on Decision and Control*, 2009, pp. 5092–5096.

[21] Wang, F., Liu, D., Zeng, L.: Modeling and simulation of optimal wind turbine con- figurations in wind farms. In: *Proceedings of IEEE World Non-Grid-Connected Wind Power and Energy Conference*, 2009, pp. 1–5.

[22] Wang, F., Liu, D., Zeng, L.: Study on Computational Grids in Placement of Wind Turbines Using Genetic Algorithm. In: *Proceedings of IEEE World Non-Grid- Connected Wind Power and Energy Conference*, 2009, pp. 1–4.

[23] Herbert-Acero, J., Franco-Acevedo, J., Valenzuela-Rendón, M., Probst- Oleszewski, O.: Linear Wind Farm Layout Optimization through Computational Intelligence. In: *Proceedings of MICAI 2009, Lecture Notes in AI*, 2009, pp. 692–703.

[24] Emami, A., Noghreh, P.: New approach on optimization in placement of wind turbines within wind farm by genetic algorithms. J. Renew. Energ. Vol. 25, pp. 1559–64, 2010.

[25] Kusiak, A., and Song, Z.: Design of wind farm layout for maximum wind energy capture. Renew. Energ., Vol. 35, No. 3, pp. 685–694, 2010.

[26] Bilbao, M., Alba, E.: CHC and SA applied to wind energy optimization using real data. *In: IEEE Congress on Evolutionary Computation*, 2010, pp. 1–8.

[27] González, J., Santos, J., Payan, M.: Wind Farm Optimal Design Including Risk. In: *IEEE International Symposium on Modern Electric Power Systems*, 2010, pp. 1–6.

[28] Rašuo, B., Bengin, A., and Veg, A.: On aerodynamic optimization of wind farm layout. PAMM, Vol. 10, No. 1, pp. 539–540, 2010.

[29] Rašuo, B., Bengin, A.: Optimization of wind farm layout. FME Trans., Vol. 38, pp. 107–114, 2010.

[30] Saavedra-Moreno, B., Salcedo-Sanz, S., Paniagua-Tineo, A., Prieto, L., Portilla- Figueras, A.: Seeding evolutionary algorithms with heuristics for optimal wind turbines positioning in wind farms. Renew. Energ. Vol. 36, No. 11, pp. 2838–2844, 2011.

[31] Kwong, W.Y., Zhang, P.Y., Romero, D., Moran, J., Morgenroth, M., Amon, C.: Wind farm layout optimization considering energy generation and noise propagation. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers. Vol. 45028, 2012, pp. 323–332.

[32] Eroğlu, Y., Seckiner, S.U.: Design of wind farm layout using ant colony algorithm. Renew. Energ. Vol. 44, pp. 53–62, 2012.

[33] Yang, J., Zhang, R., Sun, Q., Zhang, H.: Optimal wind turbines micrositing in onshore wind farms using fuzzy genetic algorithm. Mathematical Problems in Engineering, p. 24203, 2015.

[34] Rehman, S., Ali, S.S., Khan, S.A.: Wind farm layout design using cuckoo search algorithms. Applied Artificial Intelligence, Vol. 30, No. 10, pp. 899–922, 2016.

[35] Afanasyeva, S., Saari, J., Pyrhönen, O., Partanen, J.: Cuckoo search for wind farm optimization with auxiliary infrastructure. Wind Energy, Vol. 21, No. 10, pp. 855–875, 2018.

[36] Kirchner-Bossi, N., Port´e-Agel, F.: Realistic wind farm layout optimization through genetic algorithms using a Gaussian wake model. Energies Vol. 11, No. 12, pp. 3268, 2018.

[37] Charhouni, N., Sallaou, M., Mansouri, K.: Rea–listic wind farm design layout optimization with different wind turbines types. Int. J. Energ. Env. Eng., Vol. 10, No. 3, pp. 307–318, 2019.

[38] Wang, L.: Comparative study of wind turbine pla–cement methods for flat wind farm layout opti–mization with irregular boundary. Appl. Sci., Vol. 9, No. 4, p.639, 2019.

[39] Ju, X., and Liu, F.: Wind farm layout optimization using self-informed genetic algorithm with infor–mation guided exploitation. Appl. Energ., Vol. 248, pp. 429–445, 2019.

[40] Ju, X., Liu, F., Wang, L., Lee, W.J.: Wind farm layout optimization based on support vector regre–ssion guided genetic algorithm with consideration of participation among landowners. Energy Conversion and Management, Vol. 196, pp. 1267–1281, 2019.

[41] Gao, X., Li, Y., Zhao, F., žSun, H.: Comparisons of the accuracy of different wake models in wind farm layout optimization. Energy Explor. Exploit., Vol. 38, No. 5, pp. 1725–1741, 2020.

[42] Wu, X., Hu, W., Huang, Q., Chen, C., Jacobson, M. Z., and Chen, Z.: Optimizing the layout of onshore wind farms to minimize noise. Appl. Energ., Vol. 267, p. 114896, 2020.

[43] Wen, Y., Song, M., Wang, J.: A customized binary-coded genetic algorithm for wind-farm layout optimization. In: *IEEE Conference on Control Technology and Applications*, 2020, pp. 1–6.

[44] Liu, F., Ju, X., Wang, N., Wang, L., Lee, W.-J.: Wind farm macro-siting optimization with insig–htful bi-criteria identification and relocation mec–hanism in genetic algorithm. Energy Conversion and Management, Vol. 217, p. 112964, 2020.

[45] Aggarwal, S.K., Saini, L. M., and Sood, V.: Large wind farm layout optimization using nature inspired meta-heuristic algorithms. IETE J. Res., pp. 1–18, 2021.

[46] Al Shereiqi, A., Mohandes, B., Al-Hinai, A., Bakhtvar, M., Al-Abri, R., El Moursi, M., and Albadi, M.: Co-optimisation of wind farm micrositing and cabling layouts. IET Renew. Power Gener., Vol. 15, No. 8, pp. 1848–1860, 2021.

[47] Kirchner-Bossi, N., Port -Agel, F.: Wind farm area shape optimization using newly developed multi-objective evolutionary algorithms. Energies, Vol. 14, No. 14, p. 4185, 2021.

[48] Asfour, R., Brahimi, T., El-Amin, M.: Wind farm layout: modeling and optimization using genetic algorithm. In: *Proceedings of the IOP conference Series: Earth and Environmental Science,* Vol. 1008, 2022, p. 012004. IOP Publishing.

[49] Guoqing, H., Zhang, S., Li, K.: Optimization of wind farm regular layout based on grid coordinate genetic algorithm. In: *Proceedings of the World Congress on Advances in Civil, Environmental, & Materials Research (ACEM22)*, Seoul, Korea, 2022.

[50] Khan, S. A.: Adaptation of the simulated evolution algorithm for wind farm layout optimization. FME Transact, 50(4), 664-673.

[51] Mohandes, M., Khan, S.A., Rehman, S., Al-Shaikhi, S.A., Liu, B., Iqbal, K.: GARM: A stoc–hastic evolution based genetic algorithm with re–warding mechanism for wind farm layout opti–mization. FME Transactions Vol. 51, No. 4, pp. 575–584, 2023.

[52] Fraser, A.S.: Simulation of genetic systems by automatic digital computers - introduction. Aust–ralian journal of biological sciences, Vol. 10, No. 4, pp. 484–491, 1957.

[53] Holland, J. H.: Genetic algorithms and adapta–tion. Adaptive control of ill-defined systems, Vol. 16, pp. 317-333, 1984.

[54] Crepinšek, M., Liu, S. H., Mernik, M.: Exploration and exploitation in evolutionary algorithms: A survey. ACM computing surveys (CSUR), Vol. 45, No. 3, pp. 1–33, 2013.

[55] Fister, I., Strnad, D., Yang, X. S.: Adaptation and hybridization in nature- inspired algorithms. In: *Adaptation and Hybridization in Computational Intelligence*, pp. 3–50. Springer, 2015.

[56] Engelbrecht, A.P.: Fundamentals of computational swarm intelligence. John Wiley & Sons, 2005.

[57] Khan, S.A. et al.: Fuzzy Logic Based Evaluation of Hybrid Termination Criteria in the Genetic Algorithms for the Wind Farm Layout Design Problem. Computers, Materials & Continua, Vol. 84, No. 1, pp. 553-581, 2025.

[58] Rehman, S., Khan, S. A., and Alhems, L. M.: Application of TOPSIS approach to multi-criteria selection of wind turbines for on-shore sites. Appl. Sci., Vol. 10, No. 21, pp. 7595, 2020.

NOMENCLATURE

$v_{t,j}$	Wind speed at turbine i under the wake of turbine j
v_0	Mean wind speed (prevailing wind)
v_i	Wind speed at turbine i
R_j	Rotor radius of turbine j
r_i	Wake radius
α	Entrainment factor
$d_{i,j}$	Distance downstream from turbine j to turbine i (i.e., distance between the current turbine and

	the turbine creating the wake effect on it)
θ	Angle of prevailing wind to the front of the farm
N	Total number of turbines
C	Number of cells in the layout grid
$P_{current}$	Total power generated by turbines
P_{ideal}	Ideal power generated by turbines
GA	Genetic algorithms
PSO	Particle Swarm optimization
DE	Differential Evolution
EC	Evolutionary Computation
SI	Swarm Intelligence
WFLD	Wind Farm Layout Design

АНАЛИЗА НЕКОНВЕНЦИОНАЛНИХ ВЕЛИЧИНА МРЕЖЕ ЗА ОПТИМИЗАЦИЈУ РАСПОРЕДА ВЕТРОЕЛЕКТРАНА КОРИШЋЕЊЕМ ГЕНЕТСКОГ АЛГОРИТМА

М. Мохандес, С.А. Кан, Ш. Рехман, А. Ал-Шаики К. Икбал

Енергија ветра се појавила као ефикасна алтернатива фосилним горивима. На комерцијалном нивоу, енергија ветра се користи кроз ветроелектране, које се састоје од неколико ветротурбина распоређених у специфичном распореду. Идентификовање оптималног распореда ветроелектране је од виталног значаја за максималну апсорпцију енергије из ветра. Због саме сложености проблема, за оптимизацију распореда ветроелектрана коришћени су еволутивни и алгоритми ројне интелигенције. Перформансе оптималног дизајна распореда регулисане су неколико физичких параметара, као што су брзина ветра, смер ветра и величина мреже. Већина постојећих студија фокусирала се на утицај брзине ветра и смера ветра на оптимизацију распореда. Међутим, ограничена пажња је посвећена проучавању утицаја величине мреже. Ова студија анализира утицај различитих неконвенционалних величина мреже веће димензије, у распону од 16×16, 17×17, па све до 20×20, користећи генетски алгоритам као тестну лабораторију. За свеобухватније поређење, резултати од 10×10 до 15×15 су усвојени из раније студије. За разлику од претходних студија, које су се фокусирале на процену квалитета резултата, нови аспект ове студије такође разматра време извршавања као меру учинка. Још један нови аспект је анализа учинка генеричког алгоритма за различите величине мреже. Штавише, за разлику од претходних студија, које су углавном користиле хипотетичке податке, ова студија користи стварне податке са потенцијалне локације у Саудијској Арабији. Резултати показују да је максимална просечна ефикасност конверзије од 0,992 добијена са величином мреже 19×19, док је величина мреже 10×10 произвела минималну просечну ефикасност конверзије од 0,813. Ови резултати указују на то да повећање величине мреже позитивно утиче на ефикасност конверзије, која се користи као мера квалитета решења. Међутим, промена величине мреже има занемарљив утицај на време извршавања генетског алгоритма. Резултати студије потенцијално отварају пут програмерима ветроелектрана да изаберу најбољу величину мреже за дату локацију, узимајући у обзир неколико практичних питања, као што су трошкови одржавања и рада, демографске и топографске структуре, између осталих фактора.