Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2019) 27: 4659 – 4672

doi:10.3906/elk-1901-192

A new approach for wind turbine placement problem using modified differential evolution algorithm

Hüseyin HAKLI*
Department of Computer Engineering, Faculty of Engineering and Architecture, Necmettin Erbakan University, Konya, Turkey

Received: 28.01.2019 • **Accepted/Published Online:** 26.08.2019 • **Final Version:** 26.11.2019

Abstract: Energy use is increasing worldwide with industrialization and advancing technology. Following this increase, renewable energy resources are increasingly preferred to reduce the costs of energy production. Wind energy is preferred as a renewable energy resource because it is clean and safe. Wind turbines are used to meet the demand for wind energy. They are placed close to each other to generate higher amounts of energy. However, the wake effect problem arises in these types of layouts, and this hinders the turbines from producing the desired yield. A modified differential evolution (MDE) algorithm was proposed in this study to solve the placement problem for wind turbines, and employed a binary-real-coded method – obtained by combining binary coding and real coding. The proposed method contains three different modifications: generation of the initial population, regeneration, and mutation. The effective distribution of the wind turbines on land was achieved with a preliminary operation proposed to generate the initial population. In addition, with the MDE method, population regeneration and elitism were carried out to increase the diversity of population and to preserve the success of the method. Finally, a mutation operation was performed on the individuals to alternate the presence or absence of wind turbines. To investigate the performance of the MDE method in solving the wind turbine placement problem, the method was applied to a study area of 2 x 2 km. The results were compared with those obtained with other methods used in the published literature for the wind turbine placement problem. The most successful and productive placement was achieved using the proposed method, and experimental results showed that the MDE is an efficient and successful tool to solve the wind turbine placement problem.

Key words: Wind turbine placement, binary-real coding, differential evolution algorithm, optimization

1. Introduction

Electrical energy demand increases with increasing population and developing technology. To meet the fundamental needs of humans and contribute to the economic growth of a country, electrical energy is required [1]. In addition to gradually diminishing fossil fuel resources, other factors like climate change, air pollution, and the greenhouse gas effect have also raised the importance of renewable energy resources. The use of renewable energy resources is gradually becoming more prevalent thanks to its advantages of being safe, fuel-free, and clean. Solar, wind, geothermal, hydroelectric, and wave power are classified as renewable energy resources. Wind energy is one of the clean, low-cost, and commercially feasible energy types, and is therefore preferred worldwide. According to data from the Global Wind Energy Council, its cumulative capacity has reached 486.8 GW in total following a 12.5% increase, and is projected to reach a cumulative installed capacity of 800 GW in 5 years [2]. In light of this data, the increase in energy generation from wind indicates that wind turbines

*Correspondence: hhakli@erbakan.edu.tr

are more frequently used. Thus, the importance of wind turbine placement becomes more important. In a wind farm, to obtain potential wind energy, the turbines are placed in appropriate positions using a process called microplacement to achieve maximum energy generation. The microplacement optimization problem has an important role in the convenient and successful placement of turbines to achieve the highest energy yield.

There are certain limitations in the turbine placement problem, which include the interturbine distance, the service life of the turbines, the turbine noise, and the turbine (vortex) wake [3]. These limitations present challenges in formulating the problem. The approximate results obtained by commercial software packages that design turbines on a wind farm do not reach the desired levels, and therefore necessitate the development of improved solutions to the turbine placement problem. Optimal control theory has a pivotal role in both science and engineering [4]. Many researchers and practitioners have carried out studies to achieve better turbine placements by employing various optimization algorithms. The first example of these studies was carried out by Mosetti et al. in 1994 [5] in which the efficiency of genetic algorithms (GA) in solving the problem was demonstrated. In their study, by considering generations with greater population numbers and of greater problem size, Grady et al. [6] allowed enough time for candidate solutions to unify and solved the turbine placement problem using GA. For the wind turbine placement problem, Gao et al. [7] obtained a solution that aimed to generate maximum power with minimum investment cost by employing a multipopulation genetic algorithm. By using a Jensen's wake-based Gaussian model, Gao et al. [8] developed a 2-D analytical wind turbine wake model and applied a multipopulation genetic algorithm to it. In their study, Emami and Noghreh [9] optimized the turbine placement process with GA by developing an objective function used in the placement of wind turbines on wind farms and compared the results to those obtained in previous studies. Moreover, by assuming one turbine placement in each cell, they reported that the interturbine distance could be sufficient to eliminate the wake effect. Wan et al. [10] used real coding and GA to place a certain number of turbines in a certain area and showed that better results were achieved by comparing their results to those obtained by Grady. To solve the optimum placement of wind turbines in a 100 square cell area, Pookpunt and Ongsakul [11] used the binary particle swarm optimization (BPSO) algorithm and showed that the cost per power output was lower. Chen et al. [12] carried out wind turbine placement by employing the binary-real coding method, which combines binary coding and real coding methods. For their wind turbine arrangements, the linear wake model was used to minimize the cost per unit of power output; or, to maximize the profitability of a wind turbine farm, the placement process was optimized using genetic algorithms. Moreover, different optimization algorithms such as the artificial algae algorithm [13], the viral-based optimization algorithm [14], and the cuckoo search [15] have all been used to solve the wind turbine placement problem.

Today, various commercial software packages that offer wind farm designs and that identify installation problems are available. Most of these software packages solve the microplacement problem by assuming that the total number of turbine on a wind farm is fixed. In this case, the problem takes the form of a nonlinear, single-decision variable problem in which the ideal power capacity of a wind farm is calculated by taking the total number of turbines and all the turbine coordinates into account. In many turbine placement processes, the wake effect is ignored; thus, microplacement of an excessive number of turbines is carried out. To overcome this issue, a careful and meticulous approach should be adopted when establishing a wind farm, and attention should be paid to their limitations [12]. Furthermore, in some studies, the farm is divided into a certain number of cells, and the cell center is calculated by accepting it as the only turbine position in that specific cell, and by ignoring any limitations. Chen et al. [12] turned the wind turbine placement problem into a nonlinear, two-decision variable problem by using both binary and continuous variable methods to simultaneously decide ideal

turbine numbers and their placement. Thus, the presence of the turbines is decided with binary coding using "yes" and "no" questions, and the coordinates of the turbines are decided using real coding, which increases the size and complexity of the problem [12].

In this study, the optimum turbine number determination and the optimum placement process were simultaneously carried out using binary-real coding by taking the limitations into account. In its solution, the initial distribution of the turbines is as important as the performance of the algorithm. The modified differential evolution (MDE) algorithm was proposed to solve the problem with the three modifications. First, a preliminary operation was used to establish the effective and efficient distribution of the initial population over the search space. Secondly, improvements were performed to increase the diversity of population and to preserve the success of the method. To achieve population diversity after a certain number of iterations, the population regeneration process were added to the proposed method. In addition, the elitism operation was used to prevent the loss of best solutions after the regeneration. Finally, a mutation operation was performed on the individuals to overcome the disadvantages of binary-real coding. The proposed method was applied to a test site of 2 x 2 km and compared to other studies in the literature. In the trial, the limitations in energy calculations, such as interturbine distances and the wake effect, were taken into account [3].

2. Materials and methods

2.1. Problem formulation

The wake effect, which stems from the interaction of wind turbines with each other, was formulated by Jensen in 1983 [16]. The formulation introduced by Jensen was first used by Mosetti in 1994 to optimize the layout of wind turbines. Figure 1 shows the Jensen wake model employed in various studies.

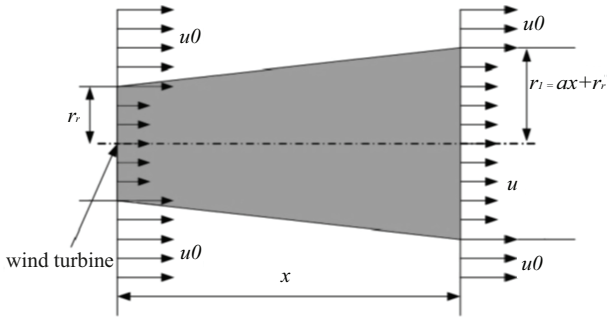

Figure 1. Jensen wake model [13, 16, 17].

The wind striking the first turbine induces the wake effect behind it. The velocity of the wind striking the first turbine is reduced and although the wind recovers after travelling a certain distance to resume its original velocity; its velocity when it reaches the second turbine is determined using the wake model given in Figure 1. The formula for the wake model is:

$$u = u_0 \left[1 - 2a \left(1 + \frac{\alpha x}{r_r} \right)^{-2} \right], \tag{1}$$

where u is the velocity in the wake at downstream distance x, u_0 represents the free wind velocity, a represents the axial induction factor, r_r represents the rotor radius, and α is the entrainment constant:

$$\alpha = \frac{0.5}{\ln(z/z_0)}. \tag{2}$$

Here z represents the hub height of the wind turbine and z_0 represents the surface roughness of the wind turbine. The thrust coefficient of the wind turbine is calculated as:

$$C_T = 4a(1-a). \tag{3}$$

The downstream rotor radius, r_1, is calculated as:

$$r_1 = r_r\sqrt{\frac{1-a}{1-2a}}. \tag{4}$$

Assuming the kinetic energy deficit of a mixed wake is equal to the sum of the energy deficits, the resulting velocity downstream of N turbines can be calculated as [18]:

$$\left(1-\frac{u}{u_0}\right)^2 = \sum_{i=1}^{N}\left(1-\frac{u_i}{u_0}\right)^2. \tag{5}$$

Total energy generation in the wind farm is:

$$P_{Total} = \sum_{i=1}^{N} 0.3 \times u^3. \tag{6}$$

The cost model frequently used in the literature to determine the cost of a wind farm [3, 5, 11, 12, 19] is preferred and this model is presented as:

$$Cost = N\left[\frac{2}{3} + \frac{1}{3}e^{-0.00174N^2}\right]. \tag{7}$$

The objective function given in Equation 8 is employed to carry out minimization: to produce maximum turbine power with minimum investment cost in a wind farm [12, 19]. This function enables the minimum cost per unit energy generation to be calculated.

$$Objective = \frac{Cost}{P_{Total}} \tag{8}$$

2.2. Coding methods for wind turbine placement

Although swarm intelligence and evolution-based methods are developed by drawing inspiration from different structures found in nature, they all have common properties; these include: 1) preparing an initial population, 2) evaluating the fitness function, 3) changing poor solutions, and 4) producing new solutions [20]. During their application to real-world problems through these stages, they may not be used directly depending on the structure of the problem, the coding type, and whether the method requires improvement. As previously mentioned, the wind turbine placement problem was modeled in different coding forms comprising binary, real, and binary-real coding, which are demonstrated in Figures 2a–2c, respectively [12]. Binary coding is obtained through combining the grids given in Figure 2a with each other. Each grid either contains a value of one or zero. Each value of "one" in the grids denotes the presence of a wind turbine, while each value of "zero" denotes the absence of a wind turbine. For this purpose, wind turbine placement is converted into a series of ones and zeros. With binary coding, the turbines are only placed in the middle of the grids. Since it fails to place the turbines in positions within continuous values, this operation is at a disadvantage in the turbine placement problem. On

the other hand, binary coding allows a change in the number of turbines, i.e. the algorithm decides how many turbines will be placed in a specific area. Real coding is represented in Figure 2b. In real coding, turbines are positioned within continuous coordinates, but the number of turbines is constant. Therefore, real coding is disadvantageous in terms of turbine numbers, while offering advantages in terms of coordinate placement. This coding type proposes a better solution to the turbine placement process through using continuous coordinate values. However, since the number of the wind turbines is constant in real coding, a different method can be employed to determine the number of the turbines to achieve an optimal layout [12]. Binary-real coding is shown in Figure 2c. It is developed by combining binary coding with real coding. While the positions of the turbines are determined by each value of x and y, the presence or absence of the turbines is determined by the values of 1 and 0. Both the positions and the number of wind turbines can be adjusted for a better layout. The binary-real coding method was preferred in the MDE method because of the advantages offered by not assigning a turbine number and providing positioning through continuous coordinates. The application of these methods to the problem is discussed in the relevant following sections.

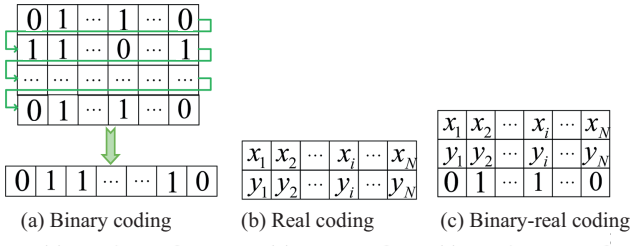

Figure 2. Diagrams of the coding methods [12].

2.3. Differential evolution algorithm

The differential evolution algorithm developed by Storn and Price in 1995 [21] is commonly used in optimization problems. In addition to its easy coding, differential evolution (DE) has a strong searching capability. Population-based DE can yield effective results in solving the optimization problems in a continuous search space. In the DE method, an advanced and effective mutation operation is applied [22]. In the mutation operation, three different individuals are selected and these individuals are used to obtain the mutation individual as:

$$V_{m,t+1} = x_{r_3,t} + F \times (x_{r_1,t} - x_{r_2,t}). \tag{9}$$

Here $V_{m,t+1}$ represents the mutation individual at the t+1; $x_{r_1,t}$, $x_{r_2,t}$, and $x_{r_3,t}$ are randomly selected individuals and they are not equal to each other and the current individual. F is scaling factor which controls the impact of differential variation.

Moreover, the crossover operation is applied in addition to the mutation operation to generate a trial chromosome from a parent chromosome [23]. After the mutation process, crossover phase is performed as:

$$x_{j,u,t+1} = \begin{Bmatrix} v_{j,m,t+1} & if rand[0,1] \leq CR \ or \ j = j^{rand} \\ x^{j,i,t} & otherwise. \end{Bmatrix} \tag{10}$$

A random number is determined for each gene, if the random number is less than crossover ratio (CR) the gene is added to new individual from the mutant individual, otherwise from the current individual. After generating the new individual, fitness values of new and current individuals are compared in the selection

phase. If the new individual is better than the current individual, the new individual is transferred to the next generation, otherwise the current is kept in memory. The DE has certain advantages over other methods such as its rapid operation, applicability to large-scale complex problems, and its requirement for a small number of control parameters. The DE has been applied in various fields including machine design [24], traffic flow models [25], pattern recognition [26], energy demand estimation [1], the training of artificial neural networks [27], the solution of chemical engineering problems [28] and the planning of unbalanced radial distribution systems [29].

2.4. Implementation of the MDE

The generation of the initial population directly affects the convergence rate and quality of the final solution; therefore, it is a process of great importance in evolutionary methods [30, 31]. If general information on the solution is not available, the initial population is randomly generated [30]. In the wind turbine placement problem, the wind turbines should be spread over the land in an orderly fashion. Determining turbine positions through random initial populations may cause clustering in certain regions, which can lead to late convergence and decreasing solution quality. A strategy was developed to achieve the complete distribution of the initial population over the land. As seen in Figure 3, the land was divided into 400 squares in the form of a 20 x 20 grid. By taking the initial population and number of sizes into consideration, turbine positions are determined in a way that allows each square to contain an equal number of turbines. For example, for 40 individuals and 10 sizes, the X and Y positions of the turbines are generated at specified intervals so that each square from (40 x 10)/400 contains one turbine. Thus, an even distribution of the wind turbines over the land is achieved by avoiding clustering in a certain region. Figure 3 shows the turbine positions of the initial populations created using the random and diversification generation methods for 40 individuals and 10 sizes. Figure 3a reveals that, in the randomly generated population, no turbines were placed in some squares and clustering emerged in some regions, whereas as seen in Figure 3b turbine placement was achieved in each square and the turbines were evenly distributed over the search space.

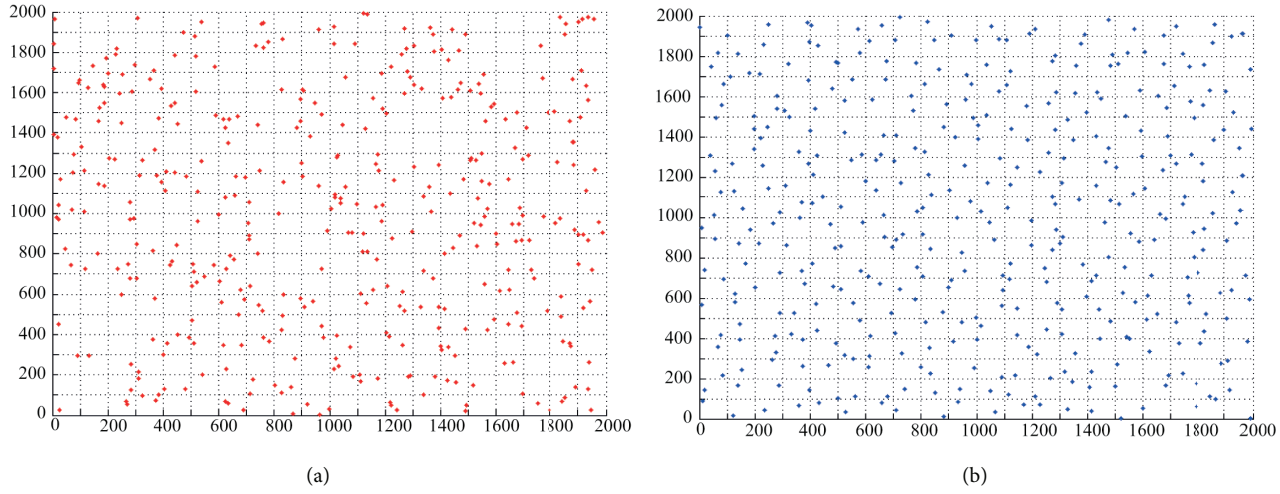

Figure 3. Generating the initial population with (a) random and (b) diversification generation methods.

The wind turbines were suitably distributed over the search space for the final solution through the strategy employed in the MDE by benefiting from the contribution of the efficient determination of the initial population. After preparing the initial population, their quality was determined using the objective function.

Following this stage, the basic mutation and crossover operations of the MDE method were performed. However, the equations used in the original DE method results in the formation of continuous data, whereas binary-real coding contains both continuous and binary data. There are three different data for each gene in the individuals, and these data are evaluated as a whole. This structure will be disrupted if the equations in the original DE method are used. Since each gene was considered as a whole, the original DE was represented by discrete values and then applied to the problem. Kitayama et al. developed a technique for the implementation of the DE in a discrete search space [32]. This technique was used in the proposed method to solve the wind turbine placement problem with binary-real coding. In the MDE method, each individual in the initial population was individually subjected to the operation. Three individuals that are different from each other and from the current individual are randomly selected and then put through certain operations to obtain a mutated individual. First, two randomly selected individuals are subjected to the operation by considering the F and a temporary individual is obtained. Instead of the equation used in the original DE method, a random number in the range [0,1] is generated for each gene, and genes from the individual are copied depending on whether the number is greater or smaller than the scaling factor. The same process is repeated for the temporary individual and the three randomly determined individuals to obtain the mutated individual, this time using a value of 0.5. Figure 4 shows the process for obtaining the mutated individual.

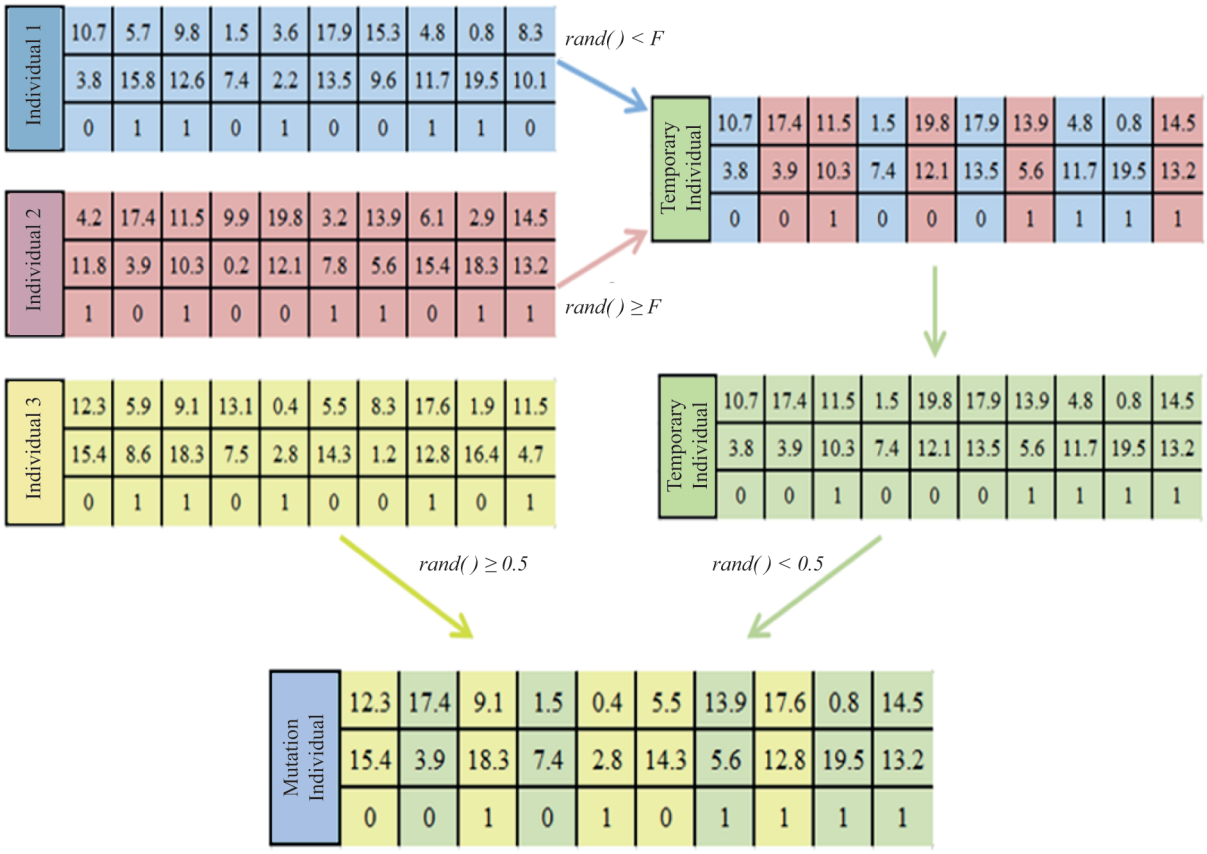

Figure 4. Obtaining the mutated individual for the MDE method.

After obtaining the mutation individual, a random number is generated for each gene of the current individual and mutation individual. The new individual is obtained by copying genes from the mutation individual if the number is below the CR and by copying genes from the current individual if the number is

above the crossover ratio. Figure 5 shows the process for obtaining the new individual. To subject the activity and passivity of the wind turbines of the new individual to change, the mutation modification given in Figure 6 is applied. The fitness value of the new individual is calculated using the objective function, and if the new individual is of higher quality than the current individual, the current individual is cancelled and the new individual is stored in the memory, otherwise no change is made.

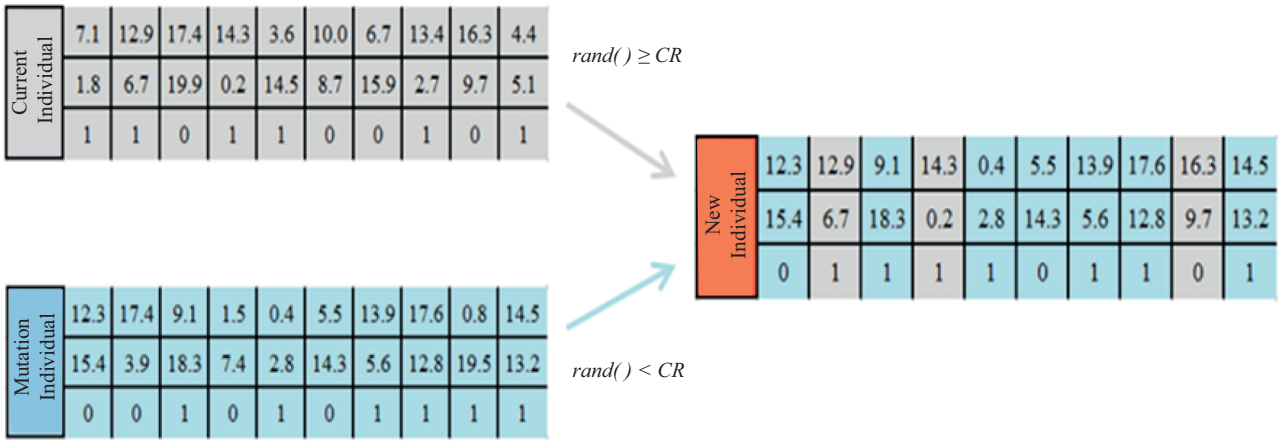

Figure 5. Obtaining the new individual for the MDE method.

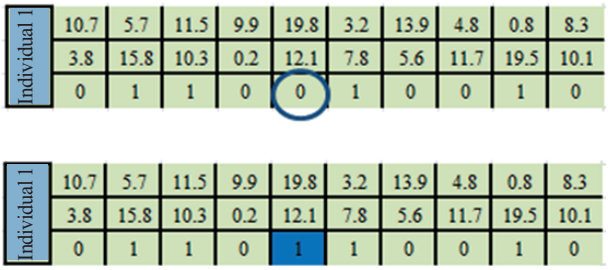

Figure 6. Implementation of mutation modification for the MDE method.

Since each gene is handled as a whole, the initially generated turbine positions are not changed. In this case, both this problem and the rapid convergence of the MDE method result in rapidly diminishing population diversity. Hence, population regeneration is carried out after a certain number of iterations to overcome this issue, while inevitably transferring a certain part of the best solutions determined by this process to the new population through elitism. Through this operation, both the population with increased similarity is diversified and the new turbine positions are introduced into the population. This process is only repeated until the specified iteration number is reached. Figure 7 shows the operation diagram for the MDE method used in the wind turbine placement problem.

3. Experimental results

In this study, the MDE method was used for the wind turbine placement problem to carry out the microplacement process. A square area of 2000 x 2000 m was used as the turbine site. Surface roughness was accepted to be 0.3 m. The common properties of the wind farm, such as power curve, wake model and optimal curve were obtained from scientific literature [3, 5, 6, 12]. The main purpose in using these properties is to minimize the

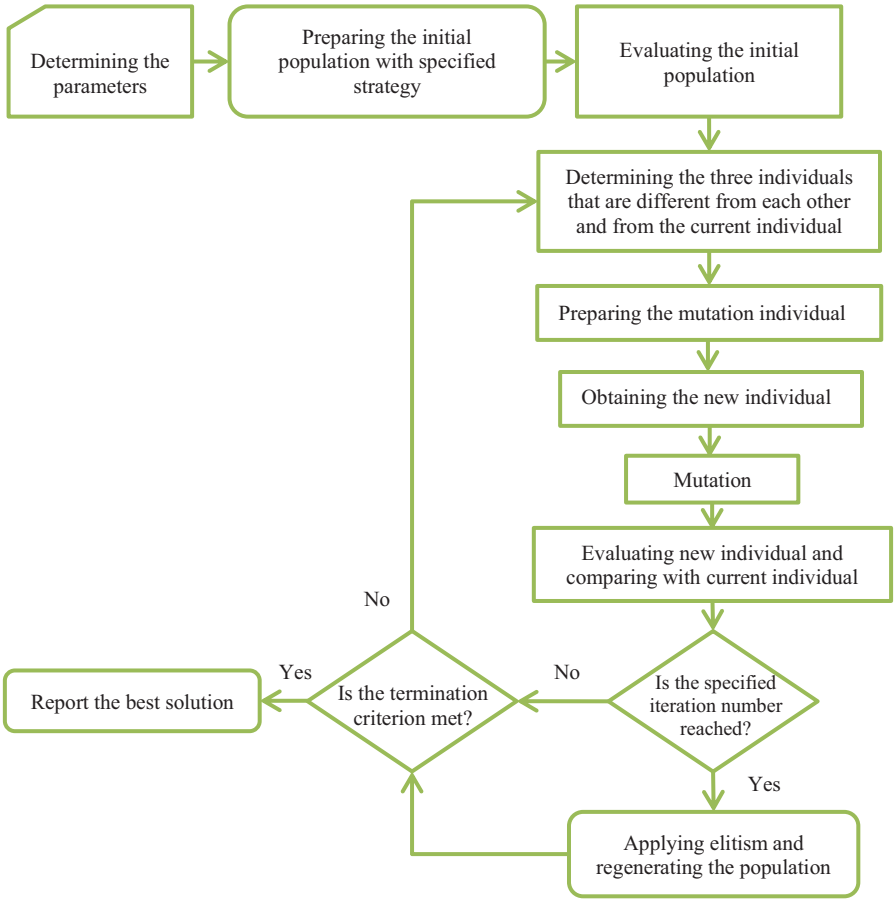

Figure 7. Operation diagram of the MDE method for the wind turbine placement problem.

wake effect of turbines on each other by using the condition of not allowing an interturbine distance to be below 5D or 200 m. With the wake effect, by keeping total energy (P_{Total}) at a maximum level, and cost ($COST$) at a minimum level, the objective function ($COST\,/\,P_{Total}$) is minimized [3, 12]. Table 1 shows the wind farm, wind turbine and wake model characteristics.

Table 1. Characteristics of the wind farm, wind turbine, and wake model [3].

Wind farm information	
Farm area (m^2)	2000 x 2000
Wind turbine specifications	
Turbine diameter (m)	40
Turbine rated power (Pr) (kW)	630
Hub height (Z)(m)	60
Coefficient of thrust (C_T)	0.88
Surface roughness (Z_0)(m)	0.3
Wake model information	
Model	Jensen
Jensen constant (k_w)	0.0944

In order to evaluate the effects of the improvements and to show its performance, the MDE was firstly compared with the basic DE. Also, modifications, performed for the MDE method, were tested on the basic DE method separately to investigate the effects of modifications. The basic DE method was respectively modified with the generation of initial population (DE-InitPop), mutation (DE-Mut), and regeneration (DE-Reg). Three modifications were used jointly on the MDE method. In the experiments for all methods, the crossover rate was 0.8, the mutation rate was set to 0.2, and F was determined as 0.5. The maximum number of function evaluations (FEs) was used as a termination condition and it was set to 3×10^5. All the variants were run 30 times for the wind turbine placement problem and the best, the worst, and the mean values were given in Table 2.

Table 2. A comparison of DE variants and the MDE algorithm.

	DE	DE-InitPop	DE-Mut	DE-Reg	MDE
Best	0.0013606	0.0013617	0.0013471	0.0013434	0.0013372
Worst	0.0014032	0.0013928	0.0013843	0.0013813	0.0013678
Mean	0.0013795	0.0013773	0.0013672	0.0013629	0.0013545

When Table 2 is considered, it is seen that the MDE method has a better performance than the basic DE algorithm and its variants. The proposed method obtains the best results for three cases, while the basic DE has the worst mean value of 30 runs. The primary reason of this is that the basic DE algorithm is exposed to stagnation due to the similarity of individuals and the lack of population diversity. It is clearly seen that mutation and regeneration modifications, used to overcome these problems, improve the performance of basic DE considering the experimental results given in Table 2. DE-InitPop is slightly better than the basic DE, while the regeneration modification (DE-Reg) provides the most significant improvement on the basic DE algorithm. The best performance for wind turbine placement problem is obtained when the three modifications are used jointly as the MDE method. For the proposed method, the regeneration and mutation modifications have a pivotal role on the diversification of population. In addition, efficiently generated initial population helps to achieve solution quickly for the MDE method. According to experimental results given in Table 2, the modifications on the MDE are beneficial and these modifications increase its performance on wind turbine placement problem.

A more suitable fitness value ($COST\,/\,P_{Total}$), which is given in Table 3, was determined for the MDE by comparing the methods to the previous approaches [3, 5, 6, 10, 12, 13]. As seen in the Table 3, two methods (Chen et al.'s method and the proposed method), in which the binary-real coding was used, outperformed the other methods in terms of fitness value. Binary-real coding determines both how many turbines are needed on a certain size of wind farm land and enables placement of the turbines in the desired positions because the coordinate system is made up of continuous values. The $COST\,/\,P_{Total}$ value is an indicator of the higher success of the result in the placement of the turbines, which was achieved by applying the MDE method to the problem by also using binary-real coding. The higher amount of total power generation obtained with the MDE method is also an indicator of the greater success of the methods compared with the others. In the placement carried out by taking the interturbine distance into account, the MDE method placed 2 more turbines compared to the method applied by Chen et al.. The effective placement achieved by the MDE method yielded maximum energy generation through successfully managing the wake effect, albeit with an increase in cost. By further decreasing the fitness value in comparison to other methods, the MDE method increased the efficiency of the turbine layout. Wan et al. used the real coding method and although obtaining the same turbine number and

layout as Grady et al., the total power output and efficiency obtained by Wan et al. were better than those obtained by Grady et al.. However, Wan et al. performed calculations by ignoring the interturbine distance factor and Figure 8 reveals the excessive proximity of the turbines to each other [12]. The MDE method used in this study yielded results better than those of Mosetti et al., Grady et al., Wan et al., Mittal et al., Beskirli et al., and Chen et al. by 21.1%, 15.0%, 8.2%, 7.6%, 5.1%, and 0.6%, respectively. Moreover, the FEs value used in the studies of Chen et al. and Beskirli et al. is twice as much as the FEs value used for the proposed method. Considering the algorithms given in Table 3, all studies implemented the GA algorithm to wind turbine placement problem except Beskirli et al. and the present study.

Table 3. A comparison of other studies within the present study.

	Mosetti et al.	Grady et al.	Wan et al.	Mittal et al.	Beskirli et al.	Chen et al.	Present study MDE
Cost/P_{Total}	0.0016197	0.0015436	0.0014475	0.0014386	0.0014054	0.0013456	0.0013372
Total Power	12352.00	14310.00	15262.00	20742.54	23422.00	22624.30	23682.60
Number of turbines	26	30	30	44	49	45	47
Algorithm	GA	GA	GA	Hybrid-GA	AAA	GA	DE
Coding format	Binary	Binary	Real	Gradient based solver	Binary	Binary-real	Binary-real
FEs	8×10^4	1.8×10^6	NA	3.15×10^5	6×10^5	6×10^5	3×10^5

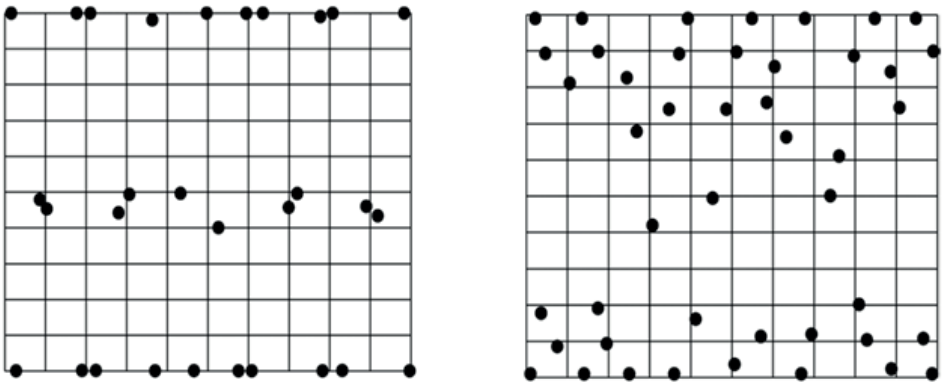

Figure 8. Optimal layout according to (a) Wan et al. [10] and (b) Chen et al. [12].

The positions of the turbines in the turbine placement problem optimized using the MDE and other methods are simulated in Figure 9. The MDE method placed 47 turbines on the turbine site.

The efficiency obtained in the study by Mosetti et al., in which 26 turbines were placed in the study area, is far behind the efficiencies obtained in other studies, and it fails to effectively use the site. Although the study by Grady et al. yielded relatively better results than that of Mosetti, the single-line and parallel layout of the wind turbines of that study resulted in an unsuccessful wind turbine distribution. Mittal et al. managed to increase the efficiency by achieving a layout of 44 turbines but failed to effectively use the middle section of the study area, which was subject to higher levels of wake effect. The MDE method increased the turbine

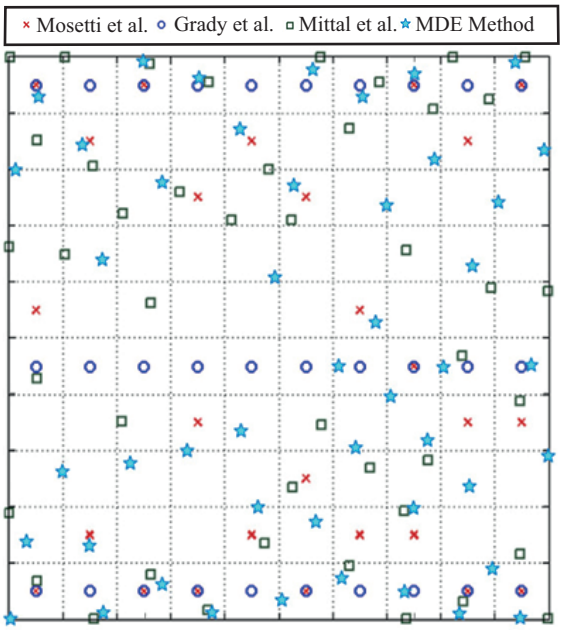

Figure 9. Turbine layouts for MDE and other methods [3].

number and achieved effective placement of the turbines, thereby achieving a successful wind turbine layout. The result obtained by Grady et al. is far away from those of the other methods. Although the result obtained by Wan et al. was considerably promising, the study optimized a fixed number of turbines and the interturbine distance was ignored. A constant number of turbines is not an optimal situation but can be remedied by using binary-real coding, which provides ideal turbine numbers and a better layout [12]. The result obtained by Chen showed a good performance thanks to the layout they introduced by carrying out binary-real coding. In their study, Chen argued that they obtained better results through randomly generating the initial population and thereby achieving population diversity. The MDE method outperformed the other methods according to the results given in Table 3 and the placement shown in Figure 9. The successful results obtained in this study are attributable to the use of the MDE, and binary-real coding methods, generating the initial population with the strategy, and performing improvements on the methods.

4. Conclusion and future works

In this study, the wind turbine placement process was carried out by modifying the DE method, which have not been previously used by other studies in scientific literature to solve the wind turbine placement problem, and by using the continuous coordinate system with binary-real coding and without pre-specifying the number of turbines. A preliminary operation was applied to obtain the efficient distribution of the turbines on the site and the success rate of the method was increased. In the MDE method, stagnation was avoided by applying the elitism and population regeneration operations after a certain number of iterations. The methods were applied to a trial site of 2 x 2 km by taking interturbine distance into account. The results obtained with the placement were simulated and compared to the results obtained in other studies. The results obtained by the MDE method achieved a more effective layout than the results obtained by other studies in the literature. The MDE method achieved maximum energy generation that produced an effective turbine layout by reducing the wake effect in spite of the increase in cost. In conclusion, the MDE method is a successful and efficient tool for

solving the wind turbine placement problem and, it can be implemented to various problems as a future work. Moreover, wind turbine placement problem can be solved with restriction of total power instead of limitation on the study area.

Acknowledgment

This study has been supported by a Scientific Research Project of Necmettin Erbakan University.

References

[1] Beskirli M, Hakli H, Kodaz H. The energy demand estimation for Turkey using differential evolution algorithm. Sadhana-Acad P Eng S 2017; 42: 1705-1715.

[2] The Global Wind Energy Council, Global Wind Report 2016.

[3] Mittal P, Kulkarni K, Mitra K. A novel hybrid optimization methodology to optimize the total number and placement of wind turbines. Renew Energ 2016; 86: 133-147.

[4] Mahmudov EN. Approximation and Optimization of Discrete and Differential Inclusions. Elsevier: Boston, MA, USA; 2011.

[5] Mosetti G, Poloni C, Diviacco B. Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm. J Wind Eng Ind Aerod 1994; 51: 105-116.

[6] Grady S, Hussaini M, Abdullah MM. Placement of wind turbines using genetic algorithms. Renew Energ 2005; 30: 259-270.

[7] Gao X, Yang H, Lin L, Koo P. Wind turbine layout optimization using multi-population genetic algorithm and a case study in Hong Kong offshore. J Wind Eng Ind Aerod 2015; 139: 89-99.

[8] Gao X, Yang H, Lin L. Optimization of wind turbine layout position in a wind farm using a newly-developed two-dimensional wake model. Appl Energ 2016; 174: 192-200.

[9] Emami A, Noghreh P. New approach on optimization in placement of wind turbines within wind farm by genetic algorithms. Renew Energ 2010; 35: 1559-1564.

[10] Wan C, Wang J, Yang G, Zhang X. Optimal siting of wind turbines using real-coded genetic algorithms. In: Proceedings of European Wind Energy Association Conference and Exhibition. Marseille, France: EWEC; 2009. pp. 3710-3715.

[11] Pookpunt S, Ongsakul W. Optimal placement of wind turbines within wind farm using binary particle swarm optimization with time-varying acceleration coefficients. Renew Energ 2013; 55: 266-276.

[12] Chen K, Song M, Zhang X. Binary-real coding genetic algorithm for wind turbine positioning in wind farm. J Renew Sustain Ener 2014; 6: 053-115.

[13] Beskirli M, Koc I, Hakli H, Kodaz H. A new optimization algorithm for solving wind turbine placement problem: Binary artificial algae algorithm. Renew Energ 2018; 121: 301-308.

[14] Ituarte-Villarreal CM, Espiritu JF. Optimization of wind turbine placement using a viral based optimization algorithm. Procedia Comput Sci 2011; 6: 469-474.

[15] Rehman S, Ali SS, Khan SA. Wind Farm Layout Design Using Cuckoo Search Algorithms. Appl Artif Intell 2018; 32: 956-978.

[16] Jensen NO. A note on wind generator interaction, 1983.

[17] Wan C, Wang J, Yang G, Li X, Zhang X. Optimal micrositing of wind turbines by genetic algorithms based on improved wind and turbine models. In: Proceedings of the 48th IEEE Conference on Decision and Control. Shanghai, P.R.China: IEEE; 2009. pp. 5092-5096.

[18] Moorthy CB, Deshmukh M. A new approach to optimise placement of wind turbines using particle swarm optimisation. Int J Sust Energ 2015; 34: 396-405.

[19] Mittal A. Optimization of the layout of large wind farms using a genetic algorithm. MSc, Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Ohio, USA, 2010.

[20] Fister D, Fister I, Šafarič R. Parameter tuning of PID controller with reactive nature-inspired algorithms. Robot Auton Syst 2016; 84: 64-75.

[21] Storn R, Price K. Differential Evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. Berkeley, USA: International Computer Science Institute; 1995.

[22] Gämperle R, Müller SD, Koumoutsakos P. A parameter study for differential evolution, Advances in intelligent systems, fuzzy systems. Evolutionary Computation 2002; 10: 293-298.

[23] Karaboğa D. Yapay Zeka Optimizasyon Algoritmalari. 6th ed. Ankara, Turkey: Nobel Akademik; 2014.

[24] Ronkkonen J, Kukkonen S, Price K. Real-parameter optimization with differential evolution. In: IEEE Congress on Evolutionary Computation. Edinburgh, Scotland: IEEE; 2005. pp. 506-513.

[25] Porfyri KN, Delis AI, Nikolos IK, Papageorgiou M. Calibration and validation of a macroscopic multi-lane traffic flow model using a differential evolution algorithm. In: Transportation Research Board 96th Annual Meeting. Washington DC, USA: TRB; 2017. pp. 1-17.

[26] Ilonen J, Kamarainen JK, Lampinen J. Differential evolution training algorithm for feed-forward neural networks. Neural Process Lett 2003; 17: 93-105.

[27] Bas E. The training of multiplicative neuron model based artificial neural networks with differential evolution algorithm for forecasting. Journal of Artificial Intelligence and Soft Computing Research 2016; 6: 5-11.

[28] Dragoi EN, Curteanu S. The use of differential evolution algorithm for solving chemical engineering problems. Rev Chem Eng 2016; 32: 149-180.

[29] Samal P, Ganguly S, Mohanty S. Planning of unbalanced radial distribution systems using differential evolution algorithm. Energ Syst 2017; 8: 389-410.

[30] Gao W, Liu S. Improved artificial bee colony algorithm for global optimization. Inform Process Lett 2011; 111: 871-882.

[31] Bajer D, Martinović G, Brest J. A population initialization method for evolutionary algorithms based on clustering and Cauchy deviates. Expert Syst Appl 2016; 60: 294-310.

[32] Kitayama S, Arakawa M, Yamazaki K. Discrete differential evolution for mixed discrete non-linear problems. J Civil Eng Arch 2012; 6: 594.