Wind Farm Layout Design Using Cuckoo Search Algorithms

S. Rehman[a,*], S.S.Ali[b,] and S. A. Khan[c]

[a]Electrical & Electronic Engineering Department, Research Institute, King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia;
[b]Center for Engineering Research, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, Tronoh, Malaysia;
[c]Computer Science Department, University of Pretoria, Pretoria, South Africa

*Correspondence to: Research Institute, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia. email: srehman@kfupm.edu.sa

Abstract

Wind energy has emerged as a strong alternative to fossil fuels for power generation. To generate this energy, wind turbines are placed in a wind farm. The extraction of maximum energy from these wind farms requires optimal placement of wind turbines. Due to complex nature of micrositing of wind turbines, the wind farm layout design problem is considered a complex optimization problem. In the recent past, various techniques and algorithms have been developed for optimization of energy output from wind farms. The present study proposes an optimization approach based on the cuckoo search (CS) algorithm, which is relatively a recent technique. A variant of CS is also proposed that incorporates a heuristic-based seed solution for a better performance. The proposed CS algorithms are compared with genetic and particle swarm optimization (PSO) algorithms, which have been extensively applied to wind farm layout design. Empirical results indicate that the proposed CS algorithms outperformed the genetic and PSO algorithms for the given test scenarios in terms of yearly power output and efficiency.

Introduction

Globally increasing population, fast technological development, and luxurious and materialistic life styles have resulted in un-proportionate increase in power requirements. Hence new and renewable sources of energy in addition to regular means of power generation are being explored to meet the increasing demands. Exploitation and utilization of clean energy sources reduce the dependence on fossil fuels, which means reduction in greenhouse gases (GHG) emissions, and at the same time it facilitates energy supply at places where there is no national or regional electrical grid. The fast developing and widely used sources of clean energy include wind, solar thermal, solar photovoltaic (PV), hydro, geothermal, and biomass. Of these clean sources, wind energy has been accepted commercially due to its availability, ease of maintenance, and low cost of operation. The global cumulative wind power installed capacity reached 369.597 GW by the end of 2014 compared to 318.644 GW in 2013, an increase of 16%, (GWEC: Global Wind Energy Council annual report 2015). The global annual cumulative wind power growth is shown in Figure 1. With cumulative installed capacity of 91.413 GW, China remained the leader in wind power industry as of December 2014. The USA, Germany, Spain, and India remained at 2nd, 3rd, 4th, and 5th place with total cumulative wind power installed capacities of 65.879 GW, 39.165 GW, 22.987 GW, and 22.465 GW, respectively. With respect to new additions in 2014, China was number one with 23.196 GW (45.1%) and Germany at number two with 5.279 GW (10.2%) new installations. However, USA, Brazil, and India remained 3rd, 4th, and 5th with new capacity additions of 4.854 GW (9.4%), 2.472 GW (4.8%), and 2.315 GW (4.5%), respectively.

The wind farm layout optimization is the process of finding out the optimal positions of wind turbines within a wind farm to maximize and/or minimize a single objective or multiple objectives, while satisfying certain constraints (Feng and Shen 2015). Although there are many commercially available software packages for wind farm layout design, researchers have also directed their interests in employing computational intelligence techniques for the purpose. It is due to the fact that, despite their elegance, these software packages simply provide assistant to human designers, and the responsibility of an efficient design mainly lies on the experience and intelligence of the designer. This may lead to inefficient designs. On the other hand, computational intelligence techniques have proven successful for a variety of complex optimization problems. The primary reason is that these techniques are least dependent on human intervention and are capable of generating efficient solutions due to their built-in intelligence.

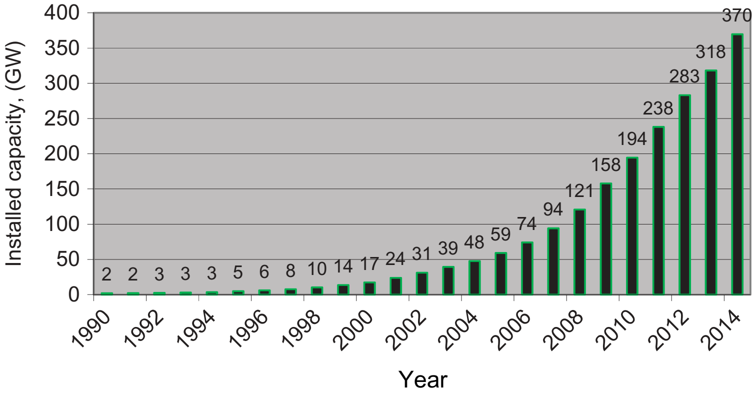


Figure 1. Global cumulative annual wind power installed growth, GWEC (GWEC: Global Wind Energy Council annual report 2015).

Several computation intelligence techniques, such as genetic algorithms, particle swarm optimization (PSO), differential evolution, and simulated annealing, have been applied to the wind farm layout design problem. However, there is a wide range of potentially good algorithms that have not been exploited and tested for the wind farm design problem. One such algorithm is cuckoo search (CS), which is a recent algorithm and has shown promising results for complex optimization problems in comparison with other established algorithms, such as genetic algorithm and PSO (Civicioglu and Besdok 2011; Guerrero, Castillo, and García 2015; Kumar and Chakarverty 2011; Yang and Deb 2010).

There are several reasons that make CS more efficient and distinct from other iterative algorithms. It has been shown that CS satisfies the global convergence requirements and thus guarantees global convergence properties (Wang et al. 2012). Furthermore, CS possesses two search capabilities: local search and global search. These search directions are controlled by a switching/discovery probability. According to (Yang and Deb 2014), the local search is very intensive with about 1/4 of the search time while global search takes about 3/4 of the total search time. This allows the search space to be explored more efficiently on the global scale, and consequently the global optimality can be found with a higher probability (Yang and Deb 2014). Another advantage of CS is that its global search uses Lévy flights or process, instead of standard random walks. As Lévy flights have infinite mean and variance, CS can explore the search space more efficiently than algorithms that follow standard Gaussian process (Yang and Deb 2014). This advantage, combined with both local and global search capabilities and guaranteed global convergence, makes CS very efficient (Yang and Deb 2014).

This paper is motivated by the above observations and proposes a CS-based algorithm for efficient wind farm layout design, which will be the first such attempt to the best of our knowledge. Another contribution of this paper is the use of heuristic-based seed solutions that further enhance the performance of CS algorithm. Furthermore, to assess the performance of the CS and for benchmarking, comparisons are done with genetic algorithm and PSO algorithm.

The rest of the paper is organized as follows. In Section 2, a brief review of the existing literature is presented. The description of the wind farm layout optimization problem is given in Section 3. Section 4 describes the wake and cost models used in this study. This is followed by a discussion on the proposed CS algorithm in Section 5. Section 6 provides the results and discussion, followed by a conclusion in Section 7.

Brief overview of existing literature

Various computational intelligence techniques have been employed for optimal design of wind farms, with genetic algorithms (GAs) (Goldberg 1989) being the first and the highest utilized algorithm so far (Khan and Rehman 2013). Early researchers such as Mosetti et al. (Mosetti, Poloni, and Diviacco 1994) and Grady et al. (Grady, Hussaini, and Abdullah 2005) employed GA for wind farm design. These algorithms have received noteworthy attention by many other researchers for wind farm design problems. Emami and Noghreh (Emami and Noghreh 2010) utilized matrix binary chromosomes which reduced computational time and improved the quality of results. They also proposed a new objective function, which allowed for more control on cost, power, and efficiency of the wind farm. The GA proposed by Gonzalez et al. (Gonzalez, Santos, and Payan 2010) considered investment financial risk as the optimization factor. Results suggested that with risk analysis included in the optimization process, the wind farm produced solutions, which were less sensitive to uncertainty than the deterministic solution. Wang et al. (Wang et al. 2015) utilized a GA for optimization while considering lands belonging to different owners. They claimed that if the division of land is complex, then the optimization with traditional penalty-based approach is not efficient. With the proposed approach, the obtained results were satisfactory, and in some scenarios, optimal or near-optimal solutions were achieved. Huang (Huang 2009) proposed a GA that was hybridized with hill-climbing property. The main achievement of this was reduction in computational time in the range of 88–92% when compared with simple GA on the same test scenarios.

Wang et al. (Wang, Liu, and Zeng 2009a) proposed a nonlinear wake model, combined with benefit evaluation model to find the optimal configuration of the wind farm. The models were incorporated in a GA for optimization. A GA proposed by Sorkhabi et al. (Sorkhabi et al. 2016) considered energy and noise as the optimization objectives. The most appealing finding of the study was that variation in the severity of land use constraints does not affect the energy generation to the same extent that they affect noise propagation. Mora et al. (Mora et al. 2007) assumed a variable-size chromosome-based GA while considering the economical aspect of optimization through Net Present Value (NPV) function. The experimentation showed that the proposed GA resulted in profit on the investment in an optimal way. Sisbot et al. (Sisbot et al. 2009) proposed a multiobjective GA with Pareto ranking to design a wind farm for a real site. The results revealed that the proposed GA could predict optimal turbine placement. Wan et al (Wan et al. 2009) proposed a GA with improved wind and turbine models. The results were compared with a previous model by Grady et al. (Grady, Hussaini, and Abdullah 2005) and demonstrated a better performance by the GA based on new model. Herbert-Acero et al. (Herbert-Acero et al. 2009) employed a virtual gene GA with the objective being maximization of the power generated by the wind farm.

Kusiak and Song (Kusiak and Song 2010) utilized a bi-criteria GA considering maximization of power and minimization of certain constraints. The proposed scheme was tested in an industrial setup and was found to be effective. Bilbao and Alba (Bilbao and Alba 2010) employed CHC-GA, which exhibits the elitist approach in the search process. This produced high-quality results. Gonzalez et al. (Gonzalez et al. 2010; González et al. 2010; Gonzalez, Santos, and Payan 2012) proposed two nested GAs for optimizing NPV for the wind farm. Performance evaluation of the proposed scheme showed suitability of the two-level GAs to find the optimum configuration of the wind farm. A GA proposed by Saavedra-Moreno et al. (Saavedra-Moreno et al. 2011) employed a greedy heuristic, which generated an initial solution of reasonable quality. This approach resulted in layouts of higher quality compared to ones that were produced by simple GA, resulting in increased economic benefits from the wind farm. Yang et al. (Yang et al. 2015) proposed a fuzzy genetic algorithm to the design of the layout while considering wake loss, terrain effect, and economic benefits. The results demonstrated that the proposed algorithm produced results of better quality with reduced computational cost when compared with simple GA. Song et al. (Song, Zhang, and Chen 2016) utilized SPEA which is a multiobjective GA and investigated the possibility of maximizing the expected wind farm power output through optimizing the layout of wind turbines as well the respective hub heights. The proposed model indicated that the power output could be increased by choosing wind turbines with different heights.

PSO (Kennedy and Eberhart 1995) is another technique that has been utilized for wind farm layout design. Rahmani et al. (Rahmani et al. 2010) made the first attempt to solve the wind farm layout design problem using PSO, considering cost per unit energy produced as the optimization objective. The results were compared with GA and indicated the effectiveness and efficiency of PSO. Chowdhury et al. (Chowdhury and Zhang 2010; Chowdhury et al. 2012) adapted PSO to design a wind farm to explore the influences of the number of turbines, the farm size, and the use of a combination of turbines with differing rotor diameters, on the optimal power generated by a wind farm. Their findings indicated that the use of an optimal combination of turbines with differing rotor diameters significantly increases the net power generation. The PSO proposed by Wan et al. (Wan et al. 2010) assumed a continuous space for turbine placement as opposed to previous studies, which assumed discrete positions for turbine placement. Maximization of generated power was sought and the results were of notable quality. The PSO algorithm by Song et al. (Song et al. 2016) employed computational fluid dynamics and the virtual particle model for the simulation of turbine wake flow and proposed a sensitivity index to quantitatively evaluate the variation of power generation under varying wind direction. The results indicated that regularly arranged turbine layouts are not suitable for stable power production. Case studies on flat terrain and complex terrain both demonstrated the effectiveness of the proposed method.

Pookpunt and Ongsakul (Pookpunt and Ongsakul 2013) proposed a binary PSO with time-varying acceleration coefficients to optimize the placement of wind turbines within a wind farm for maximum power output. The results indicated that the investment cost of power generation for both uniform and nonuniform wind speed with variable wind direction were lower than those obtained from genetic and evolutive algorithms leading to maximum power extraction with least investment. Rehman and Ali (Rehman and Ali 2015) proposed a PSO algorithm, which incorporated heuristic-based initial solutions. The objective was to minimize the total cost versus total power generated for a given number of turbines. Results were compared with GA and basic version of PSO, which indicated that the heuristic-based PSO outperformed both GA and basic PSO.

Apart from GAs and PSO, several other algorithms have been applied to variations of wind farm layout design problem. Herp et al. (Herp, Poulsen, and Greiner 2015) proposed a sequential optimization algorithm and compared it with PSO algorithm. They proposed a co-operative control strategy between the turbines. Results revealed that both algorithms were able to reach the optimal solutions, but the sequential optimization algorithm was efficient in terms of computational time. Salcedo-Sanz et al. (Salcedo-Sanz et al. 2014) presented a novel algorithm for wind farm layout optimization using the Coral Reefs Optimization algorithm and showed the proposed method outperformed the results of other methods such as Evolutionary Approaches, Differential Evolution, or Harmony Search algorithms. Eroğlu and Seçkiner (Eroğlu and Seçkiner 2013) used particle filtering approach to obtain an optimized layout of a wind farm having minimum wake effects and maximum power generation. The results showed that the particle filtering approach can compete with ant colony and evolutionary strategy algorithms available in the literature. Feng et al. (Feng and Shen 2015) used a random search algorithm based on continuous formulation to improve the wind farm layout iteratively in the feasible solution space. The optimized layouts consistently showed better performance in power production than the original layout, despite of considerable variations in wind direction and speed. Simulated annealing algorithm was used by Herbert-Acero et al. (Herbert-Acero et al. 2009) for one-dimensional arrangement of turbines, i.e. arrangement of turbines in one line. The optimization objective was to maximize the total power generated by a wind farm. Results were compared with GA and were found to be of similar quality. Bilbao and Alba (Bilbao and Alba 2010) employed simulated annealing and compared with their self-proposed CHC-GA. The results, however, showed better performance by CHC-GA compared with simulated annealing. Rasuo and Bengin (Rasuo and Bengin 2010a, 2010b) employed differential evolution algorithm while considering energy output and investment costs as the optimization objectives. Ant colony optimization was used by Eroğlu and Seçkiner (Eroğlu and Seçkiner 2013) while considering maximization of expected energy output. Results were promising.

Wind farm layout optimization problem

The wind farm layout design problem is concerned with optimal placement of wind turbines in a wind farm. This optimal placement is vital for minimizing the power losses due to various effects such as wake decay, transmission line loss, and turbulence. Furthermore, an optimal placement of wind turbines also plays a pivotal role in minimizing the costs associated with installation, functioning, and maintenance of these turbines. Considering these issues, numerous design objectives can be defined, noting the fact that the fundamental aims of a wind farm are to minimize capital and operating costs and to maximize energy production (Rasuo and Bengin 2010b). Collectively, these two objectives relate to maximizing profit from the farm, which is considered another important objective.

The area where turbines are placed can be characterized through a discrete or a continuous representation. In either case, the solution space (i.e. number of all possible solutions) is huge. Assume a wind farm layout design problem with an area of 40D × 40D, where D represents the rotor diameter of a wind turbine. If discrete representation is used and the area is divided into 100 equal size squares, then each square will be comprised of an area of $16D^2$. This will result in the turbine placement with 2^{100} possible solutions, considering whether a square contains a turbine or not. Thus, the wind farm layout design problem can be characterized as an NP-hard problem. In this scenario, exact algorithms would be inefficient due to their high computation time. Therefore, it would be necessary to employ some heuristic to intelligently search the solution space and reach an optimal or quasi-optimal solution within a reasonable amount of time.

Wake and cost modeling

The assumptions made in this paper are the same as used in the previous studies in the domain (Bilbao and Alba 2010; Emami and Noghreh 2010; Gonzalez et al. 2010; González et al. 2010; Gonzalez, Santos, and Payan 2010; Grady, Hussaini, and Abdullah 2005; Huang 2009, 2007; Mittal 2010; Mosetti, Poloni, and Diviacco 1994; Rahmani et al. 2010; Wan et al. 2010; Wang, Liu, and Zeng 2009a, 2009b). Accordingly, a simplified version of Jensen model is used in this paper to find the optimal layout design of a wind farm. The following notations have been used in the present study:

The schematic of the wake model is shown in Figure 2. The typical wind farm grid used in the present work is shown in Figure 3. For fair comparison of the proposed CS algorithm with other techniques, the grid size and other

A	Axial induction factor
E	Entrainment factor
z_0	Surface roughness
Z	Hub height
C_T	Thrust coefficient
x_{ij}	Distance downstream from turbine *j* to turbine *i* (i.e., distance between the current turbine and the turbine creating wake effect on it)
u_i	Wind speed downstream under multiple wakes
N	Total number of turbines
m_i	Set of all turbines creating wake effect on turbine *i*
r_{d0}	Wake radius immediately downstream of the wind turbine
r_{d1}	Wake radius at *x* distance downstream of the wind turbine
K	Number of rows and columns that exist in the solution space

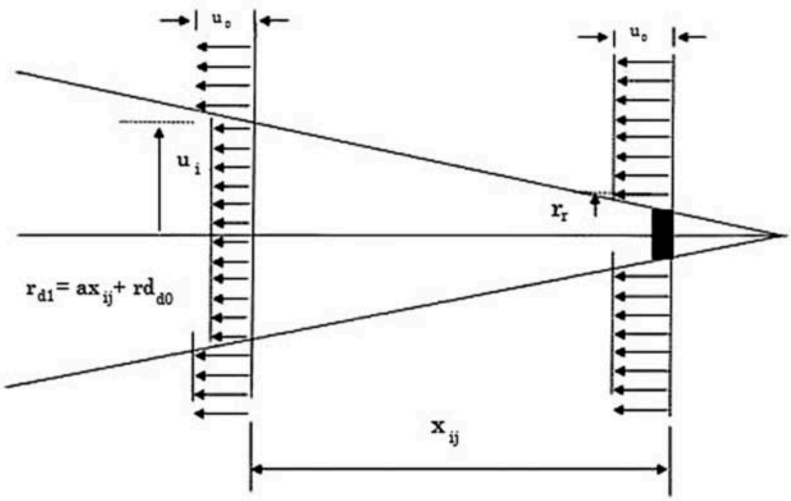


Figure 2. Schematic of the wake model.

properties are the same as used in previous studies (Bilbao and Alba 2010; Emami and Noghreh 2010; Gonzalez et al. 2010; González et al. 2010; Gonzalez, Santos, and Payan 2010; Grady, Hussaini, and Abdullah 2005; Huang 2009, 2007; Mittal 2010; Mosetti, Poloni, and Diviacco 1994; Rahmani et al. 2010; Wan et al. 2010; Wang, Liu, and Zeng 2009a, 2009b). Following these properties, the grid is divided into 100 possible turbine locations. A turbine can be placed at the center of a cell. The size of each cell is taken as five times the rotor diameter (D). More precisely, since a rotor diameter of 40 m is assumed, a cell size becomes 200 m. A hub of the turbine, directly facing the wind direction, is not under effect of any wake. Therefore, the wind speed remains unaffected as visible in Figure 3. The equations to calculate the wake, generated power, and optimization objectives (Equations (1)–(12)) have been adopted from Mosetti et al. (Mosetti, Poloni, and Diviacco 1994) (since the same model was followed by various other studies Bilbao and Alba 2010; Emami and Noghreh 2010; Gonzalez et al. 2010;

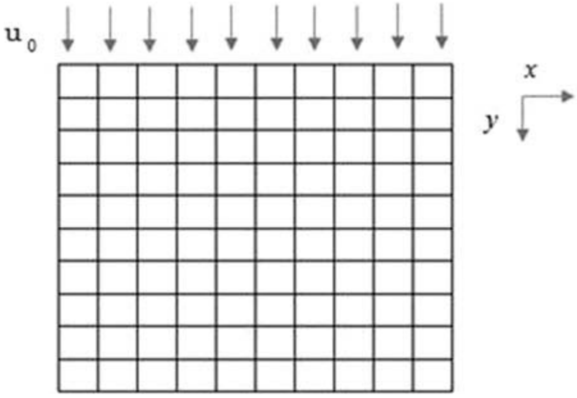


Figure 3. A 10 x 10 wind farm grid.

González et al. 2010; Gonzalez, Santos, and Payan 2010; Grady, Hussaini, and Abdullah 2005; Huang 2009, 2007; Mittal 2010; Rahmani et al. 2010; Wan et al. 2010; Wang, Liu, and Zeng 2009a, 2009b) and are presented below for the sake of clarity and completion. An interested reader may find more details in Mosetti et al. (Mosetti, Poloni, and Diviacco 1994) on the wake and power efficiency model. According to this model, we have

$$u_i = u_0 \tag{1}$$

If the hub is subjected to only one wake, then the wind speed is affected according to

$$u_i = u_0 \left[1 - \frac{2A}{\left(1 + E\left(\frac{x_{ij}}{r_{d0}}\right)\right)^2} \right] \tag{2}$$

However, if any hub is subjected to multiple wakes, then the wind speed is determined by

$$u_i = u_0 \left[1 - \sqrt{\sum_{j \in m_i} \left(\left[1 - \frac{2A}{\left(1 + E\left(\frac{x_{ij}}{r_{d0}}\right)\right)^2} \right] \right)} \right] \tag{3}$$

The radius $\mathrm{r_{d0}}$ of the wake downstream immediately after a turbine is calculated using

$$r_{d0} = r_r \sqrt{\frac{1 - A}{1 - 2A}} \tag{4}$$

Furthermore, the radius rd1 of the wake at a distance x_{ij} downstream of any wind turbine is calculated using the following equation:

$$r_{d1} = Ex_{ij} + r_{d0} \tag{5}$$

The relationship between thrust coefficient and axial induction factor is given by

$$C_T = 4A(1 - A) \tag{6}$$

The thrust coefficient is normally known for the system. Therefore, we can calculate axial induction factor a instead of CT. (The solution of Equation (6) gives two values of *A*. We select one which gives a real value for $\mathrm{r_{d0}}$ in Equation (4)). Finally, the entrainment factor *E* is found out using

$$E = \frac{0.5}{\ln\left({}^{z}\!/_{z_0}\right)} \tag{7}$$

Total cost of placing *N* turbines in the grid is calculated using the following equation:

$$Cost = N\left(\frac{2}{3} + \frac{1}{3}e^{-0.00174N^2}\right) \tag{8}$$

Total power generated by *N* turbines under multiples wakes is calculated using the following equation:

$$P_{actual} = \sum_{N}^{i} z_0 u_i^3 \tag{9}$$

Total power generated by *N* turbines without any wake is calculated using the following equation:

$$\mathbf{P_{ideal}} = \sum_{\mathbf{N}}^{\mathbf{i}} \mathbf{z_0 u_0^3} \tag{10}$$

The efficiency of the wind power generation is calculated using the following equation:

$$Efficiency = \frac{P_{actual}}{P_{ideal}} \tag{11}$$

With the above equations, the wind farm layout design problem is fundamentally the wind turbine placement problem, where the objective is to minimize the total cost versus total power generated for *N* number of turbines. Therefore, the objective of this optimization problem can be stated as

$$\mathbf{Objective} = \mathbf{min}\left(\frac{\mathbf{Cost}}{\mathbf{P_{actual}}}\right) \tag{12}$$

Cuckoo search algorithm for wind farm layout design

This section discusses the proposed CS algorithm. The section first provides a brief background of the CS algorithm, which is followed by the proposed adaptation of the CS algorithm for the wind farm layout design. Furthermore, modified CS algorithm that incorporates a heuristic-based seed solution is also discussed.

CS algorithm

CS algorithm was originally proposed by Yang and Deb (Yang and Deb 2009) as an optimization tool for numerical functions and continuous problems. The algorithm is based on the brooding parasitism of cuckoo species in natural habitat. Some cuckoo species lay their eggs in the nests of other host birds (of other species). Some host birds can engage in direct conflict with the intruding cuckoos. The original CS algorithm evolves from the following three behavioral patterns of real cuckoos (Yang and Deb 2009):

- Each cuckoo lays one egg at a time. The egg is dumped in a nest randomly chosen by the cuckoo.
- The best nests with high quality of eggs (solutions) will carry over to the next generations.
- The number of available host nests is fixed, and a host can discover an alien egg with probability p_a □ (0,1). In this situation, the host bird can either throw the egg out of its nest or abandon the nest in order to build a completely new nest in a new location.

Each nest represents a potential solution in the search space. The CS algorithm also determines how to update the position of an egg laid by a cuckoo. Each cuckoo updates its position of laying egg based on current step size via Lévy flights. Lévy flight is a natural phenomenon noticed in some birds and fruit flies. It is a combination of short and very long steps, with sudden turns (typically around 90°). These sudden turns are of essential importance for the CS algorithm, and determine the next position of the bird/fly using the following equation:

$$x_i(t+1) = x_i(t) + \alpha * Levy(\lambda) \tag{13}$$

where $\alpha > 0$ represents a step size. This step size should be closely related to the scale of the test function that the algorithm is applied on. In most cases, α can be set to the value of 1 (Rasuo and Bengin 2010a). It has been shown that the use of Lévy flight is much more efficient in exploring the search space as its step length is significantly longer when a large number of steps are performed compared to a simple random walk. The random step length is drawn from a Lévy distribution, which has an infinite variance with an infinite mean:

$$Levy{\sim}u = t^{-\lambda} = \lambda \in (0, 3] \tag{14}$$

The consecutive positions generated through steps/iterations of a cuckoo create a random walk process, which obeys a power-law step length distribution with a heavy tail.

Proposed cuckoo search algorithm

As mentioned in Section 1, wind turbine placement is an optimization problem with n × n dimensional search space. Each dimension has two possible values (i.e., 0 or 1). If a turbine is placed on any one of the n × n locations (i.e., n × n search space) then a value of "1" is assigned to that location, otherwise a "0" is assigned.

The objective of CS algorithm for wind turbine placement problem is to perform the Lévy flight in n × n dimensional search space to find the best egg among all eggs in nests. In CS, each nest contains one egg at a time and each egg represents a possible solution to a given problem. Therefore, for the wind farm layout design problem, each egg represents a solution in n × n dimensional space, with n × n locations in the wind farm. In this study, n = 10 is used, resulting in a farm of dimensions 10 x 10. Moreover, for each iteration there are 10 nests with 1 egg each. Hence, 10 solutions are generated in each iteration. Since CS is a population-based algorithm, seed solutions need to be generated for each nest. These seed solutions are generated randomly.

Steps of the cuckoo search algorithm

The following are the steps of the proposed CS algorithm:

- The objective function given in Equation (12) is minimized for any fixed number of turbines N (where N have any value between 1 and n × n).
- A turbine present at any grid position is represented by a "1" and absence is represented by a "0".
- Apply Cuckoo algorithm using Lévy distribution [Equations (13) and (14)] to update all current nest's egg (i.e., each nest contains only one egg and that egg represents a solution) in n × n dimensional search space using the difference between each solution "s" and the overall best solution "b". Here, a nest's egg in each of the n × n dimensional search space represents one possible turbine position (i.e., having either "1" if turbine is placed or "0" if turbine is not placed at the selected grid

position) and all eggs represent one complete possible solution to wind turbine placement problem.

- The original CS was designed for problems in continuous domain (i.e., an egg in each of n × n dimensional space has any real value). However, the wind turbine placement problem is a discrete problem, considering a "1" or a "0" for each dimension. Therefore, the algorithm has to be transformed to work in the discrete domain. To deal with this issue, we used the following rules in the given order for conversion from continuous domain to two-valued discrete domain of "0" and "1".
  Rule # 1: If $s_{ij} < 0$ then $s_{ij} =$ randomly assign either 0 or 1
  Rule # 2: If $s_{ij} > 0$ then $s_{ij} =$ randomly assign either 0 or 1
  Rule # 3: If $s_{ij} < 0.5$ then $s_{ij} = 0$
  Rule # 4: If $s_{ij} \geq 0.5$ then $s_{ij} = 1$

- In each nest, discover alien eggs using p_a (i.e., alien discovery probability). Generate new nests for all nests containing alien eggs using random walk around the alien egg.
- Compare each new nest with the actual nest. If the updated nest has improved value of objective function then replace the actual nest with the updated nest otherwise keep the actual nest for the next generation.
- If turbines placed in the grid are greater than total turbines N then randomly remove extra turbines from the grid.
- If turbines placed in the grid are less than total turbines N then place the remaining wind turbines in the grid using the following approach:
  - Determine the nest egg, which contains the best solution of the updated population. Compare the population best solution with the global best solution (i.e., overall best solution from population of all generations). If the population best solution is better than the global best solution then update the global best solution, otherwise keep the global best solution unchanged. The global best solution will be used by the CS algorithm to update the nest egg of each member of the population.

Cuckoo search algorithm with heuristic-based seed solution

In order to have faster convergence, the CS algorithm described above is augmented with a seed solution, which is generated using a heuristic instead of a random solution. For this purpose, the following heuristic is proposed: a special configuration of wind farm grid is assumed which would serve as the seed solution. This configuration is [1 10 6 4 8 3 5 7 9 2]. In this seed solution, each value represents a row in the farm grid. The turbines are first placed in all columns of row "1" from left to right followed by row "10" and at row “2” in the end. This heuristic is best if the wind is coming from the north (i.e., at angle of 0 degree). So, turbines are first placed according to the configuration, and further solutions are generated after rotating the grid at various angles (for e.g., if the grid is rotated at 90 degrees clockwise then the turbine in row “1” will be moved to column “10” in the respective order).

Results and discussion

The performance of the proposed CS algorithm was evaluated empirically through simulations. Two sets of experiments were performed. The first set of experiments provided a preliminary comparison between CS with random initial solution and CS with heuristic-based seed solution. In the second set of experiments, a comparative analysis of CS with GAs and PSO algorithm was conducted. All comparisons measured three aspects: fitness of solution (calculated using Equation (12)), yearly power output, and efficiency of the wind farm with the obtained configuration. Two test scenarios were used depicting different wind conditions and directions. These scenarios have been used in several earlier studies (Emami and Noghreh 2010; Gonzalez, Santos, and Payan 2010; Grady, Hussaini, and Abdullah 2005; Herp, Poulsen, and Greiner 2015; Huang 2009; Mosetti, Poloni, and Diviacco 1994; Wang, Liu, and Zeng 2009a; Wang et al. 2015). The scenarios are briefly discussed below for the sake of completeness. Furthermore, 50 independent runs were made for each scenario, and the run which gave the best results (out of 50 runs) was reported. In order to have fair comparisons between algorithms, CS, GA, and PSO were executed for the same amount of time.

Case A

This scenario assumes that the wind is coming from all the directions with equal probability, while considering mean wind speed of 12 m/s. For simplified calculations, wind directions were divided into 36 equal intervals with 10-degree differences (i.e., 0 degree, 10 degrees, 20 degrees, …, 350 degrees). It is also implicitly assumed that each turbine in the grid rotates along with the prevailing wind direction, while it is installed at the center of the cell in the grid. Thus, each turbine is facing the prevailing wind direction. The turbines affected by wake from preceding turbines will receive downstream wind speeds as per Equations (2) and (3) for single and multiple wakes, respectively. It is important to mention that since the wind directions may be approaching from all directions, it is required to determine the wake effects geometrically on the turbines downstream.

Case B

In this scenario, wind is assumed to be coming from all possible directions with equal probability but with varying mean wind speeds of 8, 12, and 17 m/s. This case is similar to Case A except for the wind speeds. Therefore, as in Case A, wind direction was divided in to 36 equal intervals with an angle difference of 10 degrees (i.e., 0 degree, 10 degrees, 20 degrees, …, 350 degrees). Furthermore, turbine installation and calculations of wake effect remain similar to Case A. The complexity of Case A is intensified by the fact that the probability of having wind direction may be different for different mean wind speeds. In particular, previous studies (Grady, Hussaini, and Abdullah 2005; Mosetti, Poloni, and Diviacco 1994) have used the probability distribution as shown in Figure 4, where it is observed that wind distributions from 270 degrees to 350 degrees are higher than the remaining wind directions, with the peak at around 310 degrees. The same distribution was used to evaluate the performance of the CS algorithm and comparison with GA.

Comparison of basic cuckoo search and heuristic-based cuckoo search algorithms

A preliminary comparison of the basic CS and CS with heuristic (CSWH) was performed. Tables 1 and 2 provide the results for the two algorithms with respect to the two scenarios and different number of turbines. As seen in Table 1, for Case A with 19 turbines, CS and CSWH were almost equal in terms of power generation and efficiency. The power generation by CS was only 3 KW more than that of CSWH. However, for 39 turbines, the difference in power generation by CS was slightly higher than that of CSWH, with 70 KW additional by CS. The difference in efficiency was also slight. As far as scenario B is concerned, results in Table 2 are slightly favorable for CSWH. For both 15 and 39 turbines, CSWH was able to produce more power and higher efficiency than CS. For 15 turbines, CSWH produced 31 KW more than CS, and for 39 turbines, this number was 15 KW. These results indicate

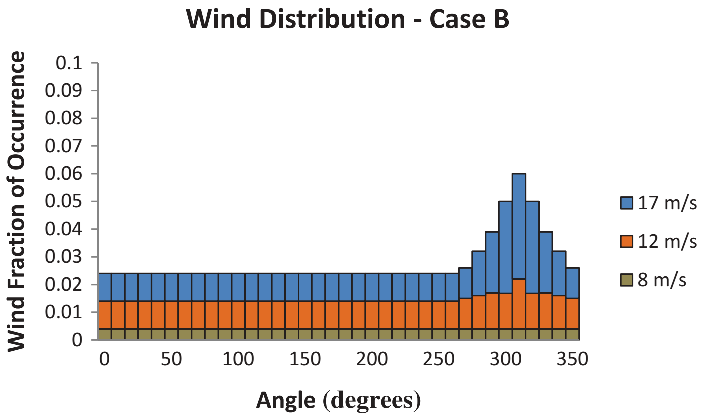


Figure 4. Varying wind speeds for different directions (developed from (Mosetti, Poloni, and Diviacco 1994)).

Table 1. Comparison of CS and CSWH for Case A using 19 and 39 turbines.

Attributes	CS	CSWH	CS	CSWH
Total (kw/year)	9385	9382	17861	17791
Efficiency (%)	95.29	95.26	88.34	88.00
No. of turbines	19	19	39	39
Run time (sec)	2584	2584	2428	2428

Table 2. Comparison of CS and CSWH for Case B using 15 and 39 turbines.

Attributes	CS	CSWH	CS	CSWH
Total (kw/year)	14,769	14,800	34,548	34,563
Efficiency (%)	97.61	97.82	87.82	87.86
No. of turbines	15	15	39	39
Run time (sec)	2463	2463	5271	5271

that, overall, CS and CSWH performed more or less the same, although the performance of CSWH was slightly better than that of CS for scenario B.

In order to further understand the behavior of the two CS algorithms, the search pattern of the two with respect to the two scenarios is depicted in Figure 5. The figure shows the search pattern for the best run (the run that

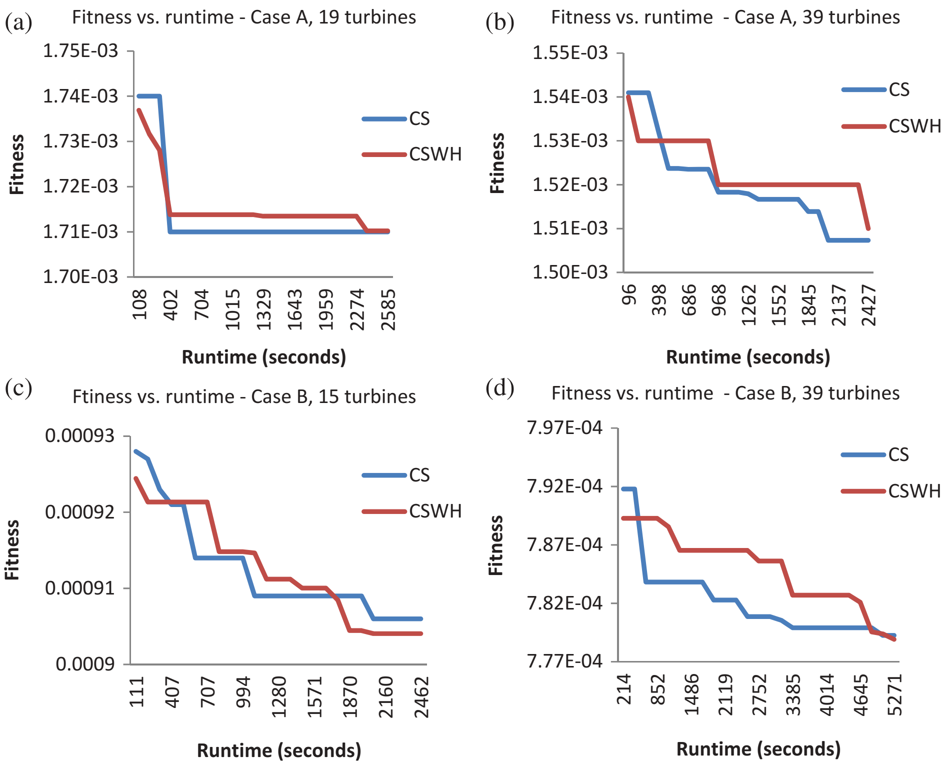


Figure 5. Progression of fitness versus runtime for CS and CSWH with (a) Case A, 19 turbines; (b) Case A, 39 turbines; (c) Case B, 15 turbines; and (d) Case B, 39 turbines.

Table 3. Comparison of CS, CSWH, GA, and PSO for Case A using 19 turbines.

Attributes	CS	CSWH	GA	PSO
Total (kw/year)	9385	9382	9245	9334
Efficiency (%)	95.29	95.26	93.86	94.77
No. of turbines	19	19	19	19
Runtime (sec)	2584	2584	2584	2584

Table 4. Comparison of CS, CSWH, GA, and PSO for Case A using 39 turbines.

Attributes	CS	CSWH	GA	PSO
Total (kw/year)	17,861	17,791	17,220	17,737
Efficiency (%)	88.34	88.00	85.17	87.73
No. of turbines	39	39	39	39
Runtime (sec)	2428	2428	2428	2428

Table 5. Comparison of CS, CSWH, GA, and PSO for Case B using 15 turbines.

Attributes	CS	CSWH	GA	PSO
Total (kw/year)	14,769	14,800	13,460	14,700
Efficiency (%)	97.61	97.82	94.62	97.16
No. of turbines	15	15	15	15
Runtime (sec)	2463	2463	2463	2463

Table 6. Comparison of CS, CSWH, GA, and PSO for Case B using 39 turbines.

Attributes	CS	CSWH	GA	PSO
Total (kw/year)	34,548	34,563	32,038	34,715
Efficiency (%)	87.82	87.86	86.62	88.25
No. of turbines	39	39	39	39
Runtime (sec)	5271	5271	5271	5271

generated the best results out of the 50 runs) for scenarios A and B with different number of turbines. For all four plots depicted in the figure, it is observed that the CSWH was able to start with better quality of initial solutions for both scenarios and turbines. This is evident from better fitness values in the early stages of the search (note that objective is to minimize the fitness value). For example, in Figure 5(a) during the first 400 seconds, the fitness value of CSWH is better than that of CS; even the fitness value of CSWH at the very start is better than that of CS. Similar patterns are observed in Figures 5(b), (c), and (d), where CSWH was able to reach solutions with higher fitness values than CS during the early stages of the search. However, as the search progresses and continues for longer duration, both CS and CSWH reach the same quality of solution in Figures 5(a) and (d), while CSWH reaches better quality of solution in Figure 5(c). A different trend is observed in Figure 5(b) where CS was able to reach solution with better fitness than CSWH, despite the fact that CSWH started with better fitness value initially.

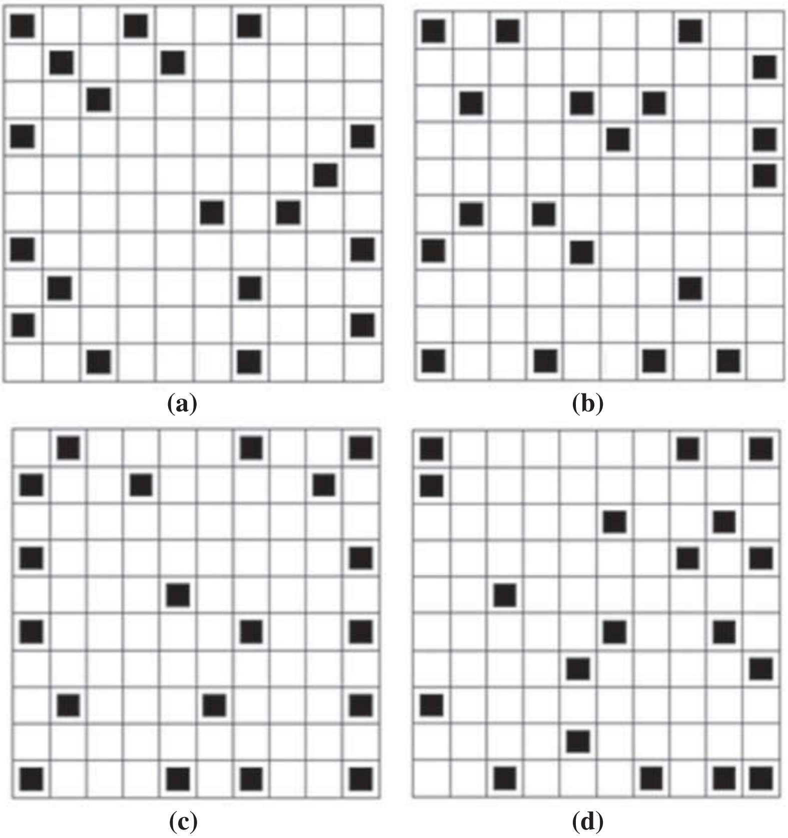
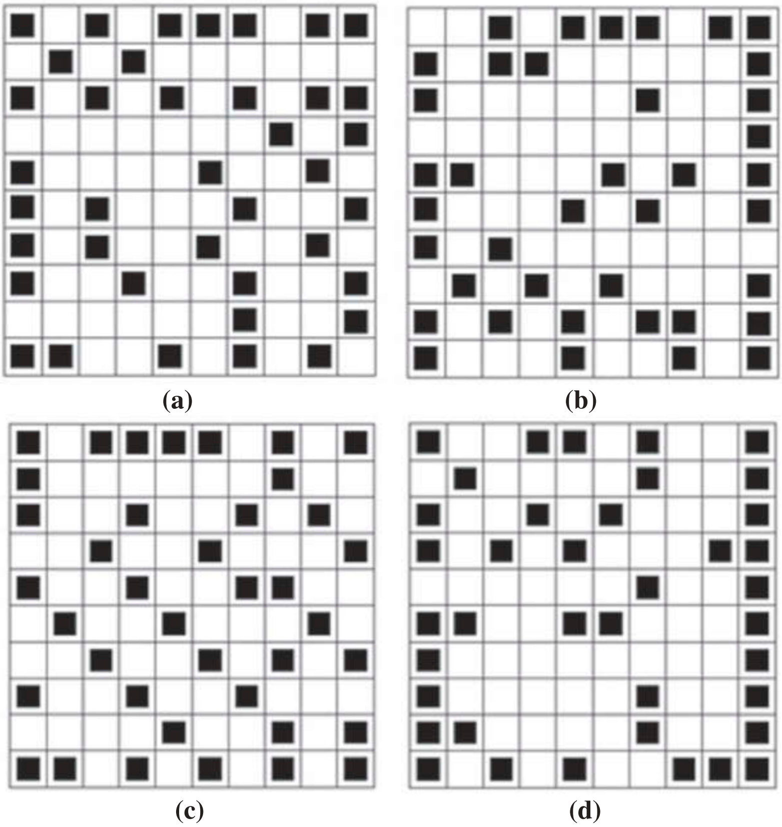


Figure 6. Best layouts for Case A with 19 turbines using (a) CS, (b) CSWH, (c) GA, and (d) PSO.

The above results and discussion indicate that CS with heuristic-based seed solution was able to provide a better start to the algorithm as compared to simple CS (which uses a random seed to start), and could be helpful if the algorithm was run for shorter durations. However, given a much longer duration, both CS and CSWH reach the same level of performance. Thus, the advantage of the heuristic seed solution is more prominent for execution for shorter durations.

Comparison of cuckoo search with genetic algorithm and particle swarm optimization algorithm

In order to benchmark the performance of the proposed CS algorithms, comparison with genetic algorithm and PSO algorithm were also performed. GA and PSO have already been applied to the same wind farm layout design problem (Rasuo and Bengin 2010b; Rehman and Ali 2015). Therefore, a comparison with the proposed CS algorithms would be more logical.



(a) (b) (c) (d)

Figure 7. Best layouts for Case A with 39 turbines using (a) CS, (b) CSWH, (c) GA, and (d) PSO.

The genetic algorithm used in this study is the same as used in various previous studies (Bilbao and Alba 2010; Grady, Hussaini, and Abdullah 2005; Mittal 2010; Wan et al. 2009). The solution representation is a binary string representing the 2-dimensional structure of the grid. Crossover probability was varied between 0.6 and 0.9 with a step size of 0.1 (i.e. 0.6, 0.7, 0.8, and 0.9), while mutation probability was varied between 0.01 and 0.1 with a step size of 0.03 (i.e. 0.01, 0.04, 0.07, and 0.1). After experimentation with the said values, values of 0.9 for crossover and 0.07 for crossover were found suitable. These values were used in the subsequent experiments.

For PSO, the parameters to tune are inertia weight, w, and acceleration coefficients c_j and c_g. For acceleration coefficients, various combinations as follows were tried: $c_j = 4$ and $c_g = 2$, $c_j = 2$ and $c_g = 4$, $c_j = c_g = 2$, and $c_j = c_g = 4$. The best results for PSO were found for both scenarios when $c_j = 2$ and $c_g = 4$. For inertia weight, values of 0.5, 0.72, and 0.9 were tried. The value of 0.72 was specifically chosen since it has been widely used in literature. It turned out that a value of 0.9 gave the best results. Therefore, all empirical work was done using $c_j = 2$, $c_g = 4$, and $w = 0.9$.

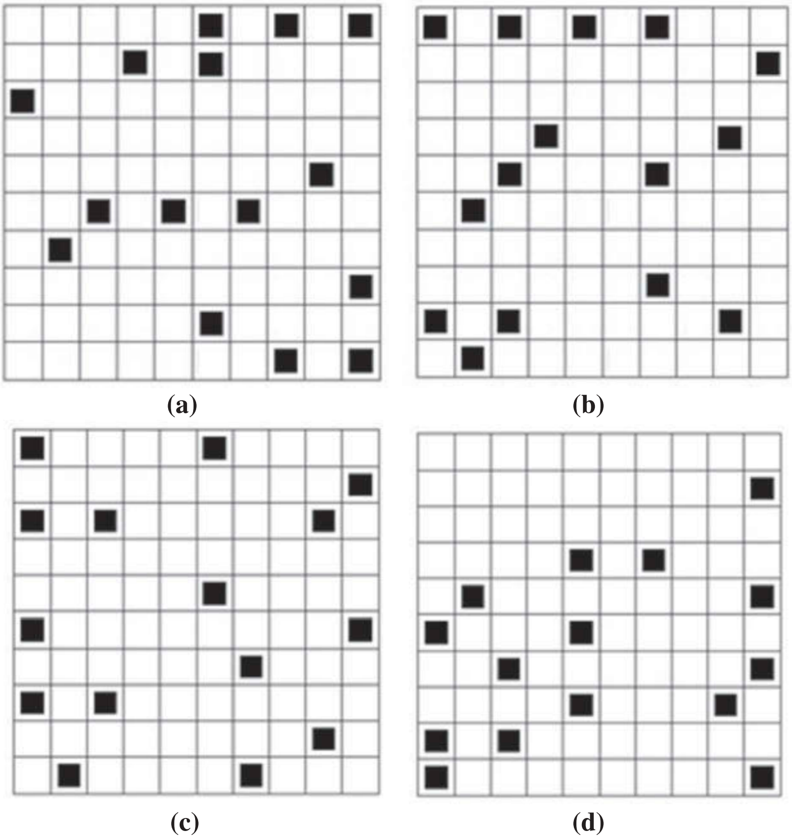
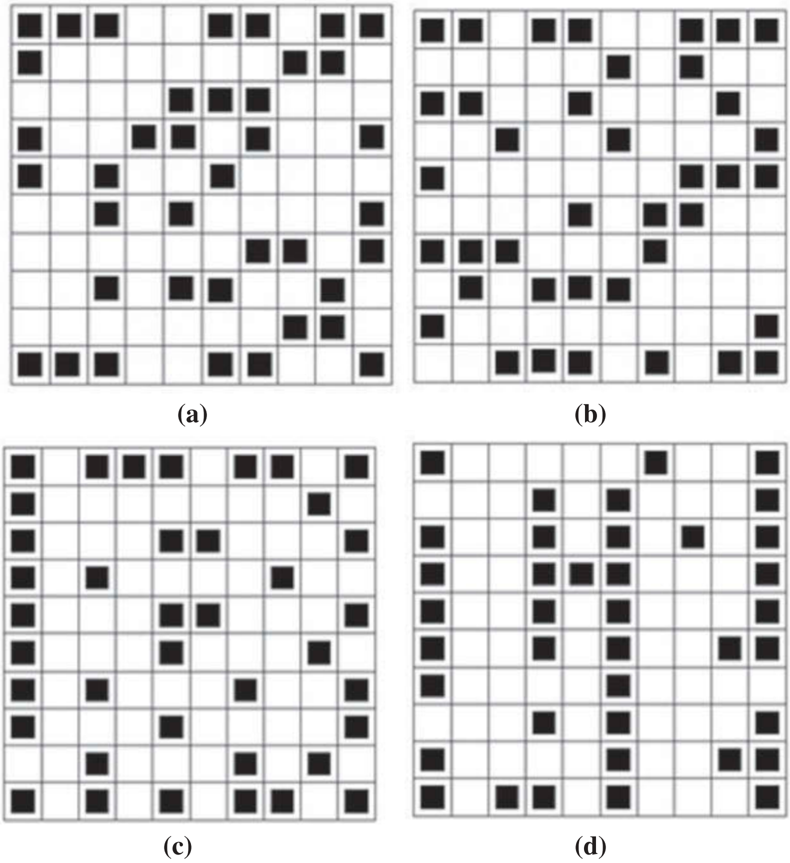


Figure 8. Best layouts for Case B with 15 turbines using (a) CS, (b) CSWH, (c) GA, and (d) PSO.

Tables 3–6 display the results produced by CS, CSWH, GA, and PSO (the results for CS and CSWH are reproduced here for the sake of completeness). It is observed from these tables that, in general, both CS and CSWH were able to produce better results (in terms of yearly power generation and efficiency) than GA and PSO for both test scenarios and the number of turbines considered. There is one exception (Case B) where PSO generated the best yearly power output and efficiency using 39 turbines. Furthermore, GA demonstrated the worst performance for all test scenarios and turbines. Figures 6–9 display the best layouts generated by CS, CSWH, GA, and PSO for different test cases.

The superior performance of CS algorithms could be attributed to the fact that CS is much more efficient in finding the global optima than GA or PSO (Chowdhury and Zhang 2010). Furthermore, CS has higher capability than PSO in terms of providing more robust and precise solutions (Chowdhury et al. 2012) and provides faster convergence compared to GA (Song et al. 2016; Wan et al. 2010). This is due to the three essential components of CS which are selection of the best, exploitation by local random walk, and


(a) (b)

(c) (d)

Figure 9. Best layouts for Case B with 39 turbines using (a) CS, (b) CSWH, (c) GA, and (d) PSO.

exploration by randomization via Lévy flights globally. These characteristics are achieved through only one parameter, p_a, which requires tuning. In comparison, GA and PSO both have a number of parameters to tune to achieve the same (Chowdhury and Zhang 2010).

Conclusions

This paper presented a novel approach for optimization of a wind farm layout. A recent optimization technique, namely, the CS algorithm, was engineered to optimize the wind farm layout design. A variant of CS was also proposed, which incorporated a heuristic-based seed solution. The proposed CS algorithms were mutually compared and it was found that the heuristic-based CS provides some leverage to the search process during the early phase of algorithm execution. Furthermore, comparison was done with genetic algorithm and PSO algorithm, which have been previously used in many studies to solve different variations of the wind turbine layout design problem. The results revealed that the proposed CS algorithms produced higher yearly energy output and better efficiency for all the considered test scenarios and different number of wind turbines. This signifies that the CS algorithm was more efficient than genetic algorithm and PSO algorithms in traversing the search space, which resulted in better solutions by CS.

Funding

This work was supported by Deanship of Research at King Fahd University of Petroleum and Minerals under project number IN131012.

References

Bilbao, M., and E. Alba. 2010. CHC and SA applied to wind energy optimization using real data. Proceedings of the IEEE conference on evolutionary computation, 1–8.

Chowdhury, S., and J. Zhang. 2010. Exploring key factors influencing optimal farm design using mixed-discrete particle swarm optimization. Proceedings of 13th AIAA/ISSMO multidisciplinary analysis and optimization conference, 1–16.

Chowdhury, S., J. Zhang, A. Messac, and L. Castillo. 2012. Unrestricted wind farm layout optimization (UWFLO): Investigating key factors influencing the maximum power generation. *Renewable Energy* 38:16–30. doi:10.1016/j.renene.2011.06.033.

Civicioglu, P., and E. Besdok. 2011. A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artificial Intelligence Review*, 39 (4):315–346. Springer.

Emami, A., and P. Noghreh. 2010. New approach on optimization in placement of wind turbines within wind farm by genetic algorithms. *Renewable Energy* 35:1559–64. doi:10.1016/j.renene.2009.11.026.

Eroğlu, Y., and S. U. Seçkiner. 2013. Wind farm layout optimization using particle filtering approach. *Renewable Energy* 58:95–107. doi:10.1016/j.renene.2013.02.019.

Feng, J., and W. Z. Shen. 2015. Solving the wind farm layout optimization problem using random search algorithm. *Renewable Energy* 78:182–92. doi:10.1016/j.renene.2015.01.005.

Goldberg, D. E. 1989. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Publishing Company, New York, United States.

Gonzalez, J., A. Rodriguez, J. Mora, M. Payan, and J. Santos. 2010. Overall design optimization of wind farms. *IEEE Systems Journal* 99:539–40.

González, J., A. Rodriguez, J. Mora, J. Santos, and M. Payan. 2010. Optimization of wind farm turbines layout using an evolutive algorithm. *Renewable Energy* 35:1671–81. doi:10.1016/j.renene.2010.01.010.

Gonzalez, J., J. Santos, and M. Payan. 2010. Wind farm optimal design including risk. *Proceedings of IEEE international symposium modelling electric power systems*, 1–6.

Gonzalez, J., J. Santos, and M. Payan. 2012. Optimization of wind farm turbine layout including decision making under risk. *IEEE Systems Journal* 6 (1):94–102. doi:10.1109/JSYST.2011.2163007.

Grady, S. A., M. Y. Hussaini, and M. M. Abdullah. 2005. Placement of wind turbines using genetic algorithms. *Renewable Energy* 30 (2):259–70. doi:10.1016/j.renene.2004.05.007.

Guerrero, M., O. Castillo, and M. García. 2015. Cuckoo search via lévy flights and a comparison with genetic algorithms. In *Fuzzy logic augmentation of nature-inspired* *optimization metaheuristics, studies in computational intelligence*, eds. O. Castillo, and P. Melin, Springer International Publishing, Switzerland, Vol. 574, 91–104.

GWEC: Global Wind Energy Council annual report. 2015. Global Wind Energy Council. http://www.gwec.net/wp-content/uploads/2012/06/GLOBAL_INSTALLED_WIND_POWER_CAPACITY_MW_–_Regional_Distribution.jpg (accessed July 13, 2015)

Herbert-Acero, J., J. Franco-Acevedo, M. Valenzuela-Rendon, and O. Probst-Oleszewski. 2009. Linear wind farm layout optimization through computational intelligence. Proceedings of the Mexican international conference on artificial intelligence, Lecture notes in Artificial Intelligence, 692–703.

Herp, J., U. V. Poulsen, and M. Greiner. 2015. Wind farm power optimization including flow variability. *Renewable Energy* 81:173–81. doi:10.1016/j.renene.2015.03.034.

Huang, H. 2007. Distributed genetic algorithm for optimization of wind farm annual profits. IEEE international conference intelligence system applied to power systems, 1–6.

Huang, H. 2009. Efficient hybrid distributed genetic algorithms for wind turbine positioning in large wind farms, Proceedings of the IEEE international symposium industrial electronics, 2196–2201.

Kennedy, J., and R. Eberhart. 1995. Particle swarm optimization. *IEEE International Conference on Neural Networks* 1942–48.

Khan, S. A., and S. Rehman. 2013. Iterative non-deterministic algorithms in on-shore wind farm design: A brief survey. *Renewable and Sustainable Energy Reviews* 19 (3):370–84. doi:10.1016/j.rser.2012.11.040.

Kumar, A., and S. Chakarverty. 2011. *Design optimization using Genetic Algorithm and Cuckoo Search*, 1–5. IEEE International Conference on Electro/Information Technology. doi:10.1109/EIT.2011.5978616.

Kusiak, A., and Z. Song. 2010. Design of wind farm layout for maximum wind energy capture. *Renewable Energy* 35:685–94. doi:10.1016/j.renene.2009.08.019.

Mittal, A., Optimization of the layout of large wind farms using a genetic algorithm. MS thesis, Case Western Reserve University; 2010.

Mora, J., J. Baron, J. Santos, and M. Payan. 2007. An evolutive algorithm for wind farm optimal design. *Neurocomputing* 70:2651–58. doi:10.1016/j.neucom.2006.05.017.

Mosetti, G., C. Poloni, and B. Diviacco. 1994. Optimization of wind turbine positioning in large wind farms by means of a genetic algorithm. *Journal of Wind Engineering and Industrial Aerodynamics* 51:105–16. doi:10.1016/0167-6105(94)90080-9.

Pookpunt, S., and W. Ongsakul. 2013. Optimal placement of wind turbines within wind farm using binary particle swarm optimization with time-varying acceleration coefficients. *Renewable Energy* 55:266–76. doi:10.1016/j.renene.2012.12.005.

Rahmani, R., A. Khairuddin, S. M. Cherati, and H. A. Pesaran. 2010. A novel method for optimal placing wind turbines in a wind farm using particle swarm optimization (PSO). Proceedings of IEEE international conference on power engineering, 134–39.

Rasuo, B., and A. Bengin. 2010a. Optimization of wind farm layout. *FME Transformers* 38:107–14.

Rasuo, B., and A. Bengin. 2010b. On aerodynamic optimization of wind farm layout. *Proceedings of the Applied Mathematics and Mechanics* 10 (1):539–40. doi:10.1002/pamm.201010262.

Rehman, S., and S. Ali. 2015. Wind farm layout design using modified particle swarm optimization algorithm. IEEE 6th International Congress on Renewable Energy Congress, March, Tunisia, pp. 1–6.

Saavedra-Moreno, B., S. Salcedo-Sanz, A. Paniagua-Tineo, L. Prieto, and A. Portilla- Figueras. 2011. Seeding evolutionary algorithms with heuristics for optimal wind turbines positioning in wind farms. *Renewable Energy* 36:2838–44. doi:10.1016/j.renene.2011.04.018.

Salcedo-Sanz, S., D. Gallo-Marazuela, A. Pastor-Sánchez, L. Carro-Calvo, A. Portilla-Figueras, and L. Prieto. 2014. Offshore wind farm design with the Coral Reefs Optimization algorithm. *Renewable Energy* 63:109–15. doi:10.1016/j.renene.2013.09.004.

Sisbot, S., O. Turgut, M. Tunc, and U. Camdali. 2009. Optimal positioning of wind turbines on Gokceada using multi-objective genetic algorithm. *Wind Energy* 13 (4):297–306. doi:10.1002/we.339.

Song, M., K. Chen, X. Zhang, and J. Wang. 2016. Optimization of wind turbine micro-siting for reducing the sensitivity of power generation to wind direction. *Renewable Energy* 85:57–65. doi:10.1016/j.renene.2015.06.033.

Song, Z., Z. Zhang, and X. Chen. 2016. The decision model of 3-dimensional wind farm layout design. *Renewable Energy* 85:248–58. doi:10.1016/j.renene.2015.06.036.

Sorkhabi, S. Y. D., D. A. Romero, G. K. Yan, M. D. Gu, J. Moran, M. Morgenroth, and C. H. Amon. 2016. The impact of land use constraints in multi-objective energy-noise wind farm layout optimization. *Renewable Energy* 85:359–70. doi:10.1016/j.renene.2015.06.026.

Wan, C., J. Wang, G. Yang, X. Li, and X. Zhang. 2009. Optimal micro-siting of wind turbines by genetic algorithms based on improved wind and turbine models. Proceedings of the 48th IEEE conference decision control, 5092–96.

Wan, C., J. Wang, G. Yang, and X. Zhang. 2010. Optimal micro-siting of wind farms by particle swarm optimization, Proceedings of International conference on swarm intelligence, LNCS, 198–205.

Wang, F., X. He, Y. Wang, and S. Yang. 2012. Markov model and convergence analysis based on cuckoo search algorithm. *Computer Engineering* 38 (11):180–85.

Wang, F., D. Liu, and L. Zeng. 2009a. Modeling and simulation of optimal wind turbine configurations in wind farms, *Proceedings of the IEEE world non-grid connected wind power energy conference*, 1–5.

Wang, F., D. Liu, and L. Zeng. 2009b. Study on computational grids in placement of wind turbines using genetic algorithm. IEEE world non-grid connected wind power energy conference, 1–4.

Wang, L., A. C. C. Tan, Y. Gu, and J. Yuan. 2015. A new constraint handling method for wind farm layout optimization with lands owned by different owners. *Renewable Energy* 83:151–61. doi:10.1016/j.renene.2015.04.029.

Yang, J., R. Zhang, Q. Sun, and H. Zhang. 2015. Optimal wind turbines micrositing in onshore wind farms using fuzzy genetic algorithm. *Mathematical Problems in Engineering* 2015:1–9. Article ID 324203, doi:10.1155/2015/324203.

Yang, X., and S. Deb. 2009. Cuckoo search via levy fights. In *Proceedings of IEEE Conference on Nature & Biologically Inspired Computing*, 210–14.

Yang, X., and S. Deb. 2014. Cuckoo search: recent advances and applications. *Neural Computing and Applications* 24 (1):169–74. doi:10.1007/s00521-013-1367-1.

Yang, X. S., and S. Deb. 2010. Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation* 1 (4):330–43. doi:10.1504/IJMMNO.2010.035430.