# The Effect of Acceleration Coefficients in Particle Swarm Optimization Algorithm with Application to Wind Farm Layout Design

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

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

**Luai M. Alhems**
Center Director and Professor
Center for Engineering Research
Research Institute
King Fahd University of Petroleum & Minerals, Dhahran 31261
Saudi Arabia

*Wind energy has become a strong alternative to traditional sources of energy. One important decision for an efficient wind farm is the optimal layout design. This layout governs the placement of turbines in a wind farm. The inherent complexity involved in this process results in the wind farm layout design problem to be a complex optimization problem. Particle Swarm Optimization (PSO) algorithm has been effectively used in many studies to solve the wind farm layout design problem. However, the impact of an important set of PSO parameters, namely, the acceleration coefficients, has not received due attention. Considering the importance of these parameters, this paper presents a preliminary analysis of PSO acceleration coefficients using the conventional and a modified variant of PSO when applied to wind farm layout design. Empirical results show that the acceleration coefficients do have an impact on the quality of final layout, resulting in better overall energy output.*

***Keywords:*** *Wind farm layout design, Wind enegy, Optimization, Particle Swarm Optimization, Artificial Intelligence, Nature-inspired algorithms*

## 1. INTRODUCTION

Wind power has evolved as a promising energy source for sustainable development and a cost-effective alternative to fossil fuels for power generation. In addition, the perception of green and clean energy further promotes the utilization of wind energy at a commercial scale. These attractive factors have resulted in serious attention being given to wind energy in recent years. Consequently, a significant level of research and development has been observed in various sub-domains of wind energy. These sub-domains include sensors and instrumentation, design and characterization of wind turbines, assessment of wind energy potential, and the development of wind farms [1,2]. This paper deals with efficient design of wind farms. This efficient design demands optimal siting of wind turbines in a wind farm with considerations of several design objectives and constraints.

Despite the availability of a number of commercial software tools for wind farm layout design, serious attention has been paid by researchers to artificial intelligence techniques for the purpose. One limitation of such commercial software is that, despite their sophistication and ease of use, these software programs rely on human interactions in the form of an experienced and intelligent designer. Consequently, this could lead to layout designs that are not fully efficient [3]. Nature-inspired algorithms (NIAs),which stem from the domain of artificial intelligence, have turned out to be effective approaches in solving a huge variety of complex optimization problems. It is due to the fact that NIAs require least level of human interaction and produce efficient solutions through their built-in intelligence.

During the past recent years, various NIAs have been engineered to design wind farms in an optimal way [3-11]. These algorithms have proven efficient in generating the optimalor near-optimal wind farm layouts. The research has utilized various NIAs. In this regard, genetic algorithm (GA) has been the most utilized algorithm[12]. The genetic algorithm also gets the credit for being the first algorithm that was used in initial research works on wind farm design [12]. Until today, GA is being utilized in wind farm layout design problems [4,8,10,11,13,16,17]. Apart from GA, several other intelligent algorithms, such as simulated annealing [6,14,15], cuckoo search [3,7], imperialist competitive algorithm [9], differential evolution [16,17], and many others,have been employed for efficient layout design of a wind farm. Among these, particle swarm optimization (PSO) [18] has also found some interest by researchers working in the area of wind farm design [5,19,20-23], although the interest has been limited. This indicates that there is a potential in the PSO algorithm for more efficient wind farm layout designs, which could lead to improved energy outputs. The motivation for applying PSO to wind farm design is further strengthened due to the following facts.

- PSO has been extensively and successfully applied to a number of complex optimization problems in a variety of disciplines [24].
- PSO has fewer parameters to adjust, making the implementation of PSO relatively easier [25].
- PSO has an effective memory component since in PSO, each particle remembers its own previous best value as well as the neighborhood best [25].

Received: July 2020, Accepted: September 2020
Correspondence to: Prof. Salman A. Khan
College of Computing & Information Sciences,
Karachi Institute of Economics & Technology
E-mail: sakhan@pafkiet.edu.pk
**doi:10.5937/fme2004922R**

- PSO has higher efficiency in maintaining swarm diversity (i.e. diversity among solutions) [26].
- An important point to consider in any NIA is the algorithmic parameters. These parameters have a strong impact on the search capabilities of an NIA, leading to efficient solutions. Therefore, it is necessary to tune these parameters. In the context of PSO, acceleration coefficients, which are associated with the cognitive and social components of the algorithm, play an important role in efficient search [27].
- While the application of PSO in this study is on square shaped wind farm layout design, the algorithm can be applied to other shapes such as circular, rectangular, or even irregular. This can be done by making necessary changes to the problem model which can effectively be incorporated within the PSO algorithm. This makes PSO a robust algorithm.
- The versatility of the PSO algorithm is further amplified by the fact that the algorithm can be easily adapted to handle various layout design approaches. For example, the current study considers fixed number of turbines to be placed within a given layout. However, the algorithm can be effectively used to decide the optimal number of turbines provided that an appropriate problem model is used.

Keeping in view the aforementioned reasons, this paper has two major contributions. The first is a preliminary analysis of the acceleration coefficients of the PSO algorithm, and its impact on the quality of solutions produced. The second contribution is a comparative analysis of the conventional PSO and modified PSO (MPSO) algorithm proposed by Rehman and Ali [23] in the context of wind farm layout design problem. The main idea behind MPSO is to utilize seed solutions generated by heuristics to improve the algorithm's performance which would eventually result in better overall energy output.

The rest of the paper is organized as follows. Section 2 describes the wake and cost models used in this study. This is followed by a discussion on particle swarm optimization algorithm in Section 3. Section 4 provides the results and discussion. The paper concludes with Section 5.

## 2. WAKE AND COST MODELING

A variant of Jensen model has been used. It is motivated by the fact that the Jensen model has been employed in several old and recent studies for wake modeling [28-32]. The grid is divided into a 10 × 10 spacing, resulting in 100 possible turbine locations, or cells as shown in Figure 1. A turbine is placed at the center of each cell, where a cell has an area of 5D × 5D, with D representing the rotor diameter. In this study, homogeneous turbine types with rotor diameter of 40 m are assumed. This defines the cell size to be 200m × 200m. A hub directly facing the wind direction is not affected by any wake, which makes the wind speed unaffected. To calculate the wake, generated power, and optimization objectives, equations have been adopted from Mosetti et al. [28]. These equations (Eqs. (2.1) to (2.12) are presented below for the sake of clarity and comprehensiveness of the paper. For more details on the wake and power efficiency model, readers can refer to Mosetti et al. [28].

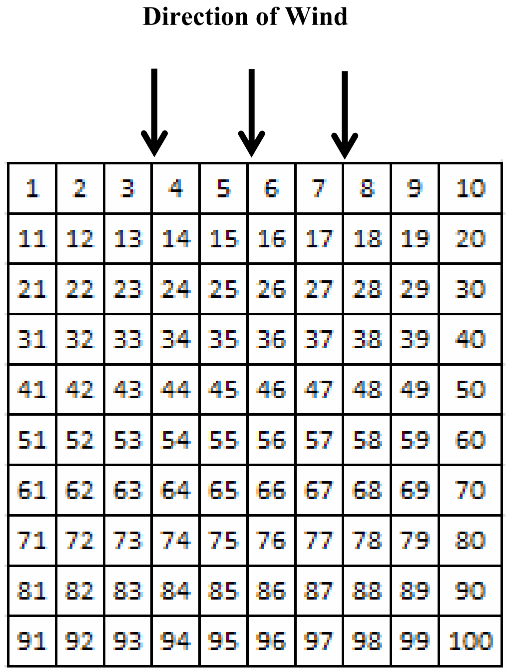


**Figure 1. Wind farm layout divided into 10 x 10 grid with 100 cells. The cell number is mentioned in each cell location.**

According to model of Mosetti et al. [12], we have:

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

Subjecting the turbine to only one wake affects the wind speed as follows:

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

However, subjecting a turbine to multiple wakes determines the wind speed as follows:

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

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

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

The following equation is used to calculate the radius $r_{d1}$ of the wake at a distance $x_{ij}$ downstream of any wind turbine

$$r_{d1} = \alpha\, x_{ij} + r_{d0} \tag{5}$$

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

$$C_t = 4a(1-a) \tag{6}$$

The thrust coefficient is normally known for the system. Therefore, the axial induction factor $a$ can be calculated instead of $C_t$. The solution of Eq. (6) gives two values of $a$. The value which gives a real number for $r_{d0}$ in Eq. (4) is selected.

Finally, the entrainment factor $\alpha$ is found out using:

$$\alpha = \frac{0.5}{\ln\left( z/z_0 \right)} \tag{7}$$

If $N$ turbines are placed in the grid, the cost is calculated using the following equation [28]

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

Total power generated by $N$ turbines under multiple wakes is calculated as follows

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

Total power generated by $N$ turbines without any wake is calculated as follows

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

The efficiency of the wind power generation is calculated as follows

$$Efficiency = P_{axtual}/P_{ideal} \tag{11}$$

Given the above, the wind farm layout design problem can be considered as 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 given in Eq. (2.12) below:

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

## 3. PARTICLE SWARM OPTIMIZATION ALGORITHMS FOR WIND FARM LAYOUT DESIGN

This section presents the PSO algorithms for designing the wind farm layout. The first approach is adaptation of the basic PSO algorithm with random initial placement of the given number of turbines in the 10×10 grid positions, while the second approach was proposed by Rehman and Ali [23] that incorporates heuristic based initial placement of the given number of turbines in the 10×10 grid positions. The search space for PSO is a grid of 10×10. Each location in the grid represents a possible position for placement of a wind turbine (i.e., initially either randomly or using heuristic to place the turbine which later move to different places as guided by the algorithms). Two important parameters in the PSO algorithm are the acceleration coefficients which are used for exploration and exploitation to control the overall search process. The PSO algorithm performs an evolutionary search to minimize the objective function by placing wind turbines at different positions of the grid as guided by the algorithm for a particular set of operating parameters.

### 3.1 Basic Particle Swarm Optimization Algorithm

Particle swarm optimization is a swarm intelligence based algorithm used to solve optimization problems. The algorithm was proposed by Kennedy and Eberhart [18] and uses the physical movements of individuals (called particles) in the swarm. These movements are governed by a mechanism so as to control and enhance the global and local exploration abilities. The strength of PSO lies in its simple design since the algorithm does not require mathematical computations like derivatives or complex encodings. The algorithm maintains historic best position (i.e. the best solution) of each particle. In addition, the global best solution of the population is also maintained. Due to these features, the algorithm is less sensitive to getting trapped in local minima compared to many other optimization algorithms.

PSO operates on a set of particles which are randomly initialized in the solution space. Each particle in its current position represents a solution. The performance of a particle is evaluated by an objective (fitness) function which is problem specific. The velocity, $v_j$, of particle $j$ corresponds to change in the position of the particle. The direction of movement of each particle is governed by its individual flying history as well as the overall swarm experiences. Each particle defines its movement towards a new solution on the basis of its own previous best position and previous best position of the whole swarm, represented by $p_j$ and $p_g$, respectively [13]. The velocities and positions of particles are updated according to the following equations

$$v_j(t+1) = v_j(t) + c_j rand_j \left( p_j - s_j(t) \right) + \\ + c_g rand_g \left( p_g - s_j(t) \right) \tag{13}$$

$$s_j(t+1) = s_j(t) + v_j(t+1) \tag{14}$$

where $t$ represents the previous iteration and $t$+1 refers to the current iteration, respectively. $c_j$ and $c_g$ are the acceleration coefficients associated with the particle's own best position and the best positions of any particle in the whole swarm, respectively. The purpose of $c_j$ and $c_g$ is to allow the particle to cover the maximum distance in a single iteration. $rand_j$ and $rand_g$ refer to two random numbers between 0 and 1, both inclusive, associated with the best solution of a particular particle and the best solution of the whole swarm. The value of the objective function is computed using particles placed in new positions at iteration $t$+1. Eqs. (3.1) and (3.2) are repeatedly used to calculate the new position and new velocity until the stopping condition is met. The best solution found by the whole swarm is recorded when the pre-defined stopping criterion is reached.

### 3.2 Impact of Acceleration Coefficients

Since the focus of this study is on carrying out a preliminary investigation on the impact of the acceleration coefficients, some discussion on this aspect is deemed necessary. The acceleration coefficients play an important role in governing the particle's search in the solution space and the convergence ability of PSO. The coefficients$c_j$ and $c_g$ are associated with the cognitive and social components respectively.

If $c_j = c_g$, particles are attracted towards the average of $p_j$ and $p_g$. Most applications use $c_j = c_g$, but the ratio between these constants is problem dependent [29]. With $c_j$>>$c_g$, each particle is much more attracted to its own personal best position which results in excessive wandering in the search space [29]. However, if $c_g$>>$c_j$ particles aremore strongly attracted to the global best position, which results in premature convergence to optima [29]. Furthermore, low values for $c_j$ and $c_g$ result in smooth particle trajectorieswherein particles roam far from good regions to explore before being pulled back towards good regions [29]. In contrast, high values of acceleration coefficients result in more acceleration, with sudden movement towards or past good regions [29].

### 3.3 Solution Structure

A particle in PSO represents a potential solution (a layout). This solution is represented as a binary matrix with 100 possible positions (representing a 10 × 10 grid as shown in Figure 1). In terms of programming implementation, this grid is treated as a one-dimensional array, where each element in the array corresponds to a cell in the grid. A '1' in a specific position shows presence of a turbine while a '0' indicates absence of a turbine. Different configurations of this matrix represent different solutions. An example of this solution structure is shown in Figure 2. In this figure, turbines are present in cells 1, 4, and 100 (among other turbines present at other locations not shown); while turbines are absent at locations 2, 3, and 5, (and many others not depicted in the figure).

| Cell # | 1 | 2 | 3 | 4 | 5 | .... | 100 |
|---|---|---|---|---|---|---|---|
| Turbine | 1 | 0 | 0 | 1 | 0 | .... | 1 |

**Figure 2. Example of solution structure**

### 3.4 Initialization for Basic Particle Swarm Optimization Algorithm

Since PSO is a population-based algorithm, a number of candidate solutions (seed solutions) are generated randomly in the initialization phase. During this phase, problem specific constraints are checked to ensure that only feasible solutions are generated. An example of a constratint could be the number of turbines defined by the designer. For example, if the designer defines that 20 turbines should be present in any configuration, then in the initialization phase, all configurations (particles) which result in less than or more than 20 tubrines would be rejected. The process of generating feasible solutions continues until the number of solutions reaches the population size defined by the designer. The fitness of each solution is evaluated using Eq. (12)

### 3.5 Constraint Handling

During initialization as well as position update phases of the PSO algorithm, newly generated/ modified solutions are checked for constraint satisfaction. More specifically, for the test scenarios considered in this study, the number of turbines are defined and fixed for each test case. If a new (or modified) configuration results in more or less number of turbines than defined, the modifed solution is not accepted and the immediate previous solution is restored.

### 3.6 Solution Perturbations

During a single iteration, each solution is perturbed through the velocity and position update using Eq. (13) and Eq. (14) respectively. A perturbation operation interchanges '1's with '0's and vice versa in various positions. The positions which require perturbations are selected through Eq. (13). These perturbations could be done anywhere in the layout, while ensuring that the constraint is not violated. Once these perturbations are done, a new solution (i.e., a new layout configuration) is formed according to Eq. (14). Each solution is then evaluated based on the fitness function given in Eq. (12).

### 3.7 Modified Particle Swarm Optimization Algorithm

The MPSO algorithm [23] evolved from the basic PSO algorithm. Unlike the basic PSO algorithm, which may start with a set of random initial solutions, the MPSO algorithm uses seed solutions which allow the algorithm to converge faster to an optimal solution. Seed solutions are pre-defined initial feasible solutions that are used by an optimization algorithm. Their purpose is to assist the algorithm in reaching the optimal solution in less amount of time as compared to random initial solution. Seed solutions are problem specific and are carefully constructed.

In the context of wind farm layout design problem, the MPSO algorithm uses two types of seed solutions. In the first type, turbines are placed in specific configuration. This seed solution is effective for situations if the prevailing wind is at $0^o$ with the turbine. The second type of seed solution is a modification of the first seed solution and is obtained by random shuffle of rows and columns in the first seed solution. These two types of seed solutions are used alongwith random initial solutions. That is, in the initial population, some solutions are generated randomly while others are generated using the two types of seed solutions. Specific details of the two seed solutions as well as the modified PSO can be found in [23].

## 4. RESULTS AND DISCUSSIONS

Simulations were performed and empirical results were generated for the basic and modified PSO algorithms. A swarm size of 10 was assumed, which means that the algorithms maintain 10 solutions in each iteration. Both PSO and MPSO algorithms were run for 30 minutes for Scenarios A and B, and for 50 minutes for Scenario C (see details below). The reason for using runtime is that the PSO and MPSO algorithms have a different structure, and therefore it is not fair to compare the two algorithms in terms of number of iterations. The comparison using runtime has also been advocated in similar studies [24,33]. These runtimes were set after experimentation with different timing values. Note that our intention is not to study the optimal convergence of both basic and modified PSO algorithms, but to investigate their mutual relative performance. Therefore, at the end of allocated runtimes for the different scenarios, the quality of solution obtained by both PSO variants were mutually compared.

In accordance with the standard practice for analyzing results of iterative heuristics (such as PSO), 30 independent runs were made for each algorithm setup, and average of these 30 runs were reported. Two sets of experiments were done. The first set of experiments investigated the effect of acceleration coefficients on basic and modified PSOs to find out whether the search is influenced by particle's own positions, swarm's positions, or both. In the second set, a preliminary comparison between basic PSO and modified PSO was done. The results measured three aspects: fitness of solution (calculated using Eq. (12)), yearly power output, and efficiency of the wind farm with the obtained configuration. Three test scenarios were assumed which have been used in many previous studies [28,30,35,36]. For the sake of completeness, these scenarios are summarized below. Furthermore, only one wind turbine type has been considered with hub height of 60 m, turbine diameter of 40 m, and a thrust coefficient equal to $C_t$=0.88 which was kept constant for the wind speeds considered. Roughness is $Z_o$ = 0.3 m.These parameters have been used in previous studies [28,30,37]. Furthermore, the power curve was adopted from the study of Mosetti et al. [28] which assumed variable power values at different speeds below 12 m/s, and became constant for wind speeds over 12 m/s.

*Scenario A*

In this scenario, a turbine is placed at the center of the cell. The cell is assumed to have dimensions of 5D×5D in the grid. Wind is assumed to be unidirectional with fixed speed of 12 m/s. Due to cell width of 5D with wind prevailing at an angle of $0^o$, there is no wake effect between grids in different columns. However, if turbines are places in the same column, then a turbine gets affected by the wake created by a turbine ahead of it in the same column. For evaluating this scenario, the number of turbines used were 26 and 30 turbines. These numbers were adopted from previous studies [3,23,28,30].

*Scenario B*

In this scenario, the wind is assumed to be coming from all the directions with equal probability, with mean wind speed of 12 m/s. For simplified calculations, wind directions are divided into 36 equal intervals with difference of 10 degrees (i.e., $0^o$, $10^o$, $20^o$, …, $350^o$). It is implicitly assumed that each turbine is capable of rotating in the direction of the prevailing wind. The turbines facing wakes from preceding turbines receive downstream wind speeds according to Eqs. (2.2) and (2.3) for single and multiple wakes, respectively. It should also be noted that since the wind may be approaching from all directions, it is essential to determine the wake effects geometrically on the turbines downstream. For testing, the number of turbines considered in this scenario were 19 and 39 turbines, same as used in some previous studies [3,23,28,30].

*Scenario C*

This scenario assumes that wind is coming from all directions with equal probability but with varying mean wind speeds of 8, 12, and 17 m/s. All other assumptions are exactly the same as in scenario B. The main difference between this scenario and scenario B is the varying wind speed. The complexity of scenario C is higher than scenario B since the probability of having wind direction may be different for different mean wind speeds. The number of turbines used in this scenario for testing is 15 and 39 turbines, which were adopted from some previous studies [3,23,28,30].

### 4.1 Effect of acceleration coefficients on basic PSO and modified PSO

As mentioned earlier in Section 3.2, the acceleration coefficients govern the search of the PSO towards the particle's own previous best position as well as the best position found by any particle in the whole swarm. The impact of unequal and equal values of the acceleration coefficients was also discussed earlier. Keeping that discussion in view, a sensitivity analysis of acceleration coefficients was performed with four different combinations of$c_j$ and $c_g$. The values ranged between 2 and 4 for both $c_j$and $c_g$.These combinations were $c_j$ = 4and $c_g$ = 2, $c_j$ = 2and $c_g$ = 4, $c_j = c_g = 2$, and $c_j = c_g = 4$.

Tables 1 and 2 show the results for scenario A considering 26 and 30 turbines respectively. Note that the results for both PSO and MPSO were the same, since scenario A is a very simple scenario. Accordingly, the results obtained for PSO and MPSO were same, since both were able to reach the same quality of solutions. It is observed from Table 1 that with 26 turbines, the best results were obtained while both acceleration coefficients were having a high and same value, i.e. 4. On the other hand, the situation changed in Table 2 when the turbines were increased to 30, in which case the best results were obtained when the acceleration coefficient associated with the swarm behavior was stronger than the acceleration coefficient of individual behavior.

Table 3 shows the results for different acceleration coefficients for PSO and MPSO while considering scenario B with 19 turbines. It is observed from the table that basic PSO obtained best results when both acceleration coefficients are having equal and high values. For MPSO, the best results were obtained when the coefficient for individual behavior is stronger than the coefficient associated with swarm behavior. However, results are quite different when the number of turbines is changed to 39 for the same test scenario, as depicted in Table 4. In this table, it is observed that as far as PSO is concerned, the best results were obtained when the swarm behavior was dominant over individual behavior, as displayed by the values of $c_j$ = 2and $c_g$ = 4. On the other hand, when MPSO was evaluated for the same test scenario and number of turbines, the best results were obtained when the two acceleration coefficients were the same, and high, as shown with the values of $c_j = c_g = 4$.

**Table 1. Results for different acceleration coefficients for basic PSO/MPSO with 26 Turbines and scenario A. Best results are shown in boldface.**

| $c_j$ | $c_g$ | Fitness value | Total kw/year | Efficiency (%) |
|---|---|---|---|---|
| 4 | 2 | 0.001704 | 11743.87 | 87.131 |
| 2 | 4 | 0.001704 | 11743.1 | 87.125 |
| 2 | 2 | 0.001713 | 11680.11 | 86.658 |
| 4 | 4 | **0.001685** | **11872.87** | **88.088** |

**Table 2. Results for different acceleration coefficients for basic PSO/MPSO with 30 Turbines and scenario A. Best results are shown in boldface.**

| $c_j$ | $c_g$ | Fitness value | Total kw/year | Efficiency (%) |
|---|---|---|---|---|
| 4 | 2 | 0.001689 | 13078.63 | 84.096 |
| 2 | 4 | **0.001655** | **13346.14** | **85.816** |
| 2 | 2 | 0.001672 | 13209.67 | 84.939 |
| 4 | 4 | 0.001679 | 13153.35 | 84.577 |

As far as scenario C is concerned, Tables 5 and 6 shows the results for basic PSO and MPSO while considering 15 and 30 turbines, respectively. As shown in Table 5, PSO was able to find the best solutions when $c_j = 2$and $c_g = 4$, whereas for MPSO, best results were obtained when $c_j = 4$and $c_g = 2$. As for 39 turbines, Table 6 shows that the situation was the same as far as PSO is concerned, since PSO was able to find the best results again with values of $c_j = 2$and $c_g = 4$. However, the situation for MPSO changed with regard to the values of acceleration coefficients. MPSO obtained best results when $c_j = c_g = 4$.

**Table 3. Results for different acceleration coefficients for basic PSO and MPSO with 19 Turbines and scenario B. Best results are shown in boldface. AC = Acceleration Coefficients. Eff = Efficiency.**

| A C | | PSO | | | MPSO | | |
|---|---|---|---|---|---|---|---|
| $c_j$ | $c_g$ | Fitness Value | Total kw/ year | Eff. (%) | Fitness Value | Total kw/ year | Eff. (%) |
| 4 | 2 | 0.001725 | 9303.94 | 94.46 | **0.00172** | **9334.1** | **94.766** |
| 2 | 4 | 0.001725 | 9301.68 | 94.437 | 0.00172 | 9315.82 | 94.581 |
| 2 | 2 | 0.00172 | 9329.93 | 94.724 | 0.00172 | 9313.36 | 94.556 |
| 4 | 4 | **0.001718** | **9338.03** | **94.806** | 0.00173 | 9299.37 | 94.414 |

**Table 4. Results for different acceleration coefficients for basic PSO and MPSO with 39 Turbines and scenario B. Best results are shown in boldface. AC = Acceleration Coefficients, Eff = Efficiency.**

| AC | | PSO | | | MPSO | | |
|---|---|---|---|---|---|---|---|
| $c_j$ | $c_g$ | Fitness Value | Total kw/ year | Eff. (%) | Fitness Value | Total kw/ year | Eff. (%) |
| 4 | 2 | 0.001526 | 17641.1 | 87.256 | 0.00152 | 17697.8 | 87.536 |
| 2 | 4 | **0.001522** | **17688.2** | **87.489** | 0.00152 | 17708.8 | 87.591 |
| 2 | 2 | 0.001529 | 17612.3 | 87.114 | 0.00152 | 17731.5 | 87.703 |
| 4 | 4 | 0.001535 | 17543 | 86.771 | **0.00152** | **17737** | **87.73** |

From the results in Tables 1 to 6, certain interesting observations can be made about basic PSO and MPSO. As far as basic PSO is concerned, results in Tables 1 to 6 indicated that basic PSO was able to find the best results when the acceleration coefficient was always equal to its maximum value, that is, when $c_g = 4$. Another pattern for basic PSO was that a value of $c_j$= 2was associated with the majority of cases, with two exceptions. These exceptions were scenario A with 26 turbines, and scenario B with 19 turbines, where $c_j$= 4 was also associated with best results. These observations, when put together, indicate that the better performance of basic PSO was more influenced by the positions of the best particle in the whole swarm, rather than the individual positions of a particle.

**Table 5. Results for different acceleration coefficients for basic PSO and MPSO with 15 Turbines and scenario C. Best results are shown in boldface. AC = Acceleration Coefficients, Eff = Efficiency.**

| AC | | PSO | | | MPSO | | |
|---|---|---|---|---|---|---|---|
| $c_j$ | $c_g$ | Fitness Value | Total kw/ year | Eff. (%) | Fitness Value | Total kw/ year | Eff. (%) |
| 4 | 2 | 0.000916 | 14605.2 | 96.53 | **0.00091** | **14700.1** | **97.155** |
| 2 | 4 | **0.000906** | **14774** | **97.644** | 0.00091 | 14673.2 | 96.978 |
| 2 | 2 | 0.000916 | 14611.3 | 96.569 | 0.00092 | 14629.9 | 96.691 |
| 4 | 4 | 0.000914 | 14636.3 | 96.734 | 0.00091 | 14648.4 | 96.814 |

**Table 6. Results for different acceleration coefficients for basic PSO and MPSO with 39 Turbines and scenario C. Best results are shown in boldface. AC = Acceleration Coefficients, Eff = Efficiency.**

| AC | | PSO | | | MPSO | | |
|---|---|---|---|---|---|---|---|
| $c_j$ | $c_g$ | Fitness Value | Total kw/ year | Eff. (%) | Fitness Value | Total kw/ year | Eff. (%) |
| 4 | 2 | 0.000783 | 34378.11 | 87.389 | 0.000776 | 34714.76 | 88.244 |
| 2 | 4 | **0.000779** | **34578.70** | **87.899** | 0.000777 | 34664.56 | 88.117 |
| 2 | 2 | 0.000788 | 34151.90 | 86.814 | 0.000783 | 34403.72 | 87.454 |
| 4 | 4 | 0.000784 | 34337.90 | 87.285 | **0.000773** | **34826.53** | **88.529** |

On the other hand, the results for MPSO indicate quite different patterns as compared to basic PSO. For majority of cases, MPSO was able to find the best results when $c_j = 4$, with the exception of one case (scenario A, 30 turbines) where $c_j = 2$was associated with the best results. Moreover, with most of the cases, values of $c_g = 4$ were also associated with the best results, with the exception of two cases (scenario B with 19 turbines and scenario C with 15 turbines) where $c_g = 2$ was associated with the best results. Therefore, it can be fairly claimed that the results of MPSO were influenced by both the individual positions of particles as well as the positions of the best particle in the whole swarm, although the results are more inclined towards the individual behavior.

### 4.2 Comparison of basic PSO and modified PSO

A preliminary comparison of basic and modified PSO was also performed. The focus was on search pattern of the two algorithms with respect to the different scenarios. As mentioned in the previous section, both PSO and MPSO produced the same results for scenario A. Therefore, search pattern of this scenario was not analyzed. Figure 3 shows the typical search patterns for scenarios B and C with different number of turbines. For scenarios B, both PSO and MPSO were run for the same amount of time (30 minutes) and search patterns were recorded. In Figures 3(a) and 3(b), the search patterns of PSO and MPSO are shown for 19 and 30 turbines respectively, while considering scenario B.

Figure 3(a) indicates that both PSO and MPSO were able to achieve the same fitness value, while in Figure 3(b), MPSO showed a much lower fitness value (note that the objective is to minimize the fitness value).

As far as scenario C is concerned, Figures 3(c) and 3(d) depict the search patterns for 15 and 39 turbines, respectively, for both PSO and MPSO. Figure 3(c) indicates that PSO was able to achieve better (lower) fitness values than MPSO. However, for 39 turbines, the graphs in Figure 3(d) suggest that MPSO performed a more efficient search, resulting in lower fitness values.

From the above discussion, the overall trend appears to be in favor of MPSO. In two out of four cases, MPSO was better than PSO, and equal to PSO in one case. There was one case in which PSO was able to achieve better performance. Thus, it can be fairly claimed that MPSO showed a relatively better performance than PSO. However, as mentioned earlier in this section, the results are preliminary and further investigation is required in this regard.

## 5. CONCLUSIONS

Wind farm layout design has been classified as a complex optimization problem. The problem involves designing an optimal layout for a given wind farm considering design objectives and technical constraints. Due to the complexity of the problem, algorithms of linear or polynomial complexity cannot guarantee optimal or even feasible solutions. This motivates the use of nature-inspired iterative heuristics since these algorithms have proven to very effective in solving complex optimization problems. To solve the wind farm layout optimization problem, this paper presented the application of basic and modified particle swarm optimization algorithms, with specific emphasis on the effect of an important algorithmic parameter, namely, acceleration coefficients. It was observed that, in general, the values of acceleration coefficients have an impact on the quality of solutions produced by both basic PSO and modified PSO. Furthermore, preliminary comparison between basic PSO and modified PSO suggests that MPSO produced slightly better results.

Our future research will be focused on an in-depth study of acceleration coefficients, in addition to the study of other PSO parameters. We also intend to propose more variants of the PSO algorithm, and to compare with other algorithms, in the context of wind farm layout design problem.

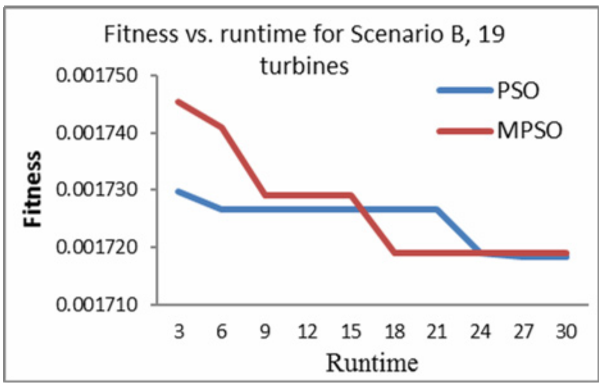


**Figure 3(a). Progression of fitness versus runtime (in minutes) for PSO and MPSO with Scenario B, 19 turbines**

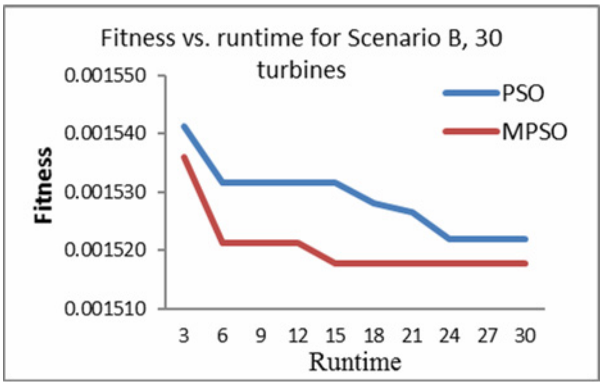


**Figure 3(b). Progression of fitness versus runtime (in minutes) for PSO and MPSO with Scenario B, 30 turbines**

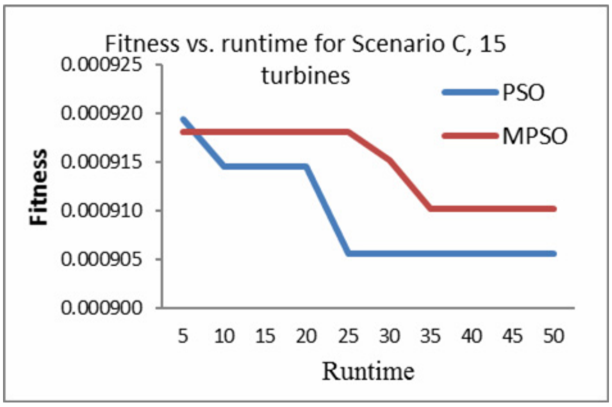


**Figure 3(c). Progression of fitness versus runtime (in minutes) for PSO and MPSO with Scenario C, 15 turbines**

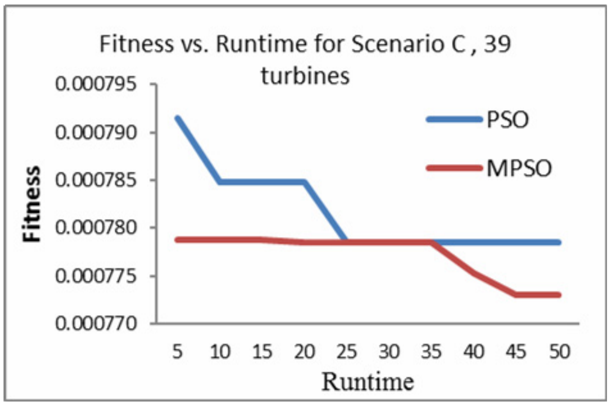


**Figure 3(d). Progression of fitness versus runtime (in minutes) for PSO and MPSO with Scenario C, 39 turbines**

## ACKNOWLEDGMENT

The authors thank King Fahd University of Petroleum & Minerals, for supporting this research under project grant # SB181015. The authors would also like to thank Syed S. Ali for his support.

## REFERENCES

[1] Ettoumi, F., Adane, A., Benzaoui, M., Bouzergui, N.: Comparative simulation of wind park design and siting in Algeria. Renew. Energ. Vol. 33, No. 10, pp. 2333–2338, 2008.

[2] Mustakerov, I., Borissova, D.: Wind turbines type and number choice using combinatorial optimization. Renew. Energ., Vol. 35, No. 9, pp. 1887–1894, 2010.

[3] Rehman, S., Ali, S.S. and Khan, S.A.: Wind farm layout design using cuckoo search algorithms. Appl. Artif. Intell., Vol. 30, No. 10, pp.899-922, 2016.

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

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

[6] Yang, K., Cho, K.: Simulated annealing algorithm for wind farm layout optimization: A benchmark study. Energies, Vol. 12, No. 23, p.4403. 2019.

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

[8] Wu, Y., Zhang, S., Wang, R., Wang, Y. and Feng, X.: A design methodology for wind farm layout considering cable routing and economic benefit based on genetic algorithm and GeoSteiner. Renew. Energ., Vol. 146, pp.687-698, 2020.

[9] Kiamehr, K. and Hannani, S.K.: Wind farm layout optimization using imperialist competitive algorithm. J. Renew. Sustain. Energ, Vol. 6, No. 4, p.043109, 2014.

[10] Gao, X., Li, Y., Zhao, F. and Sun, H.: Comparisons of the accuracy of different wake models in wind farm layout optimization. Energ. Explor. Exploit., p.0144598720942852., 2020.

[11] Wang, L.: Comparative study of wind turbine placement methods for flat wind farm layout optimization with irregular boundary. Appl Sci., Vol. 9, No. 4, p.639, 2019.

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

[13] Yang, J., Zhang, R., Sun, Q., Zhang, H.: Optimal Wind Turbines Micrositing in Onshore Wind Farms Using Fuzzy Genetic Algorithm, Math. Prob. Eng., Vol. 2015, Article ID 324203, 9 pages, 2015.

[14] Bilbao, M. and Alba, E.: CHC and SA applied to wind energy optimization using real data. In Proceedings of IEEE Conf. Evol. Comput., pp. 1–8, 2010.

[15] Herbert-Acero, J., Franco-Acevedo, J., Valenzuela-Rendon, M., Probst-Oleszewski, O.: Linear wind farm layout optimization through computational intelligence. in: *Proceedings Mexican Int. Conf. Artific. Intell., Lecture Notes in Artificial Intelligence*, pp. 692–703. 2009.

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

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

[18] Kennedy, J. and Eberhart, R.: Particle swarm optimization. in: *Proceedings of IEEE Int. Conf. Neural Netw.*, pp. 1942–1948, 1995

[19] Chowdhury, S., and Zhang, J.: Exploring key factors influencing optimal farm design using mixed-discrete particle swarm optimization. in: *Proceedings of 13th AIAA/ISSMO Multidisciplinary Analys. Optim. Conf.*, pp. 1–16, 2010

[20] Chowdhury, S., Zhang, J., Messac, A., and Castillo, L.: Unrestricted wind farm layout optimization UWFLO: investigating key factors influencing the maximum power generation. Renew. Energ., Vol 38. No. 1, pp. 16–30, 2012.

[21] Rahmani, R., Khairuddin, A., Cherati, S., and Pesaran, H.: A novel method for optimal placing wind turbines in a wind farm using particle swarm optimization (PSO). in: *Proceedings IEEE Int. Conf. Power Eng.*, pp. 134–139, 2010

[22] Wan, C., Wang, J., Yang, G., and Zhang, X.: Optimal micro-siting of wind farms by particle swarm optimization. In: *Proceedings of Int Conf. Swarm Intell., LNCS*, pp. 198–205, 2010.

[23] Rehman, S., Ali, S.S.: March. Wind farm layout design using modified particle swarm optimization algorithm. In: *Proceedings of IREC2015 The Sixth International Renewable Energy Congress*, pp. 1-6, 2015.

[24] Khan, S. A., Engelbrecht, A. P.: A fuzzy particle swarm optimization algorithm for computer communication network topology design, Appl. Intell., Vol. 36, No. 1, pp. 161-177, 2012.

[25] Boeringer, D. W., and Werner, D. H.: Particle swarm optimization versus genetic algorithms for phased array synthesis, IEEE Trans. Anten. Propagat., Vol. 52, No. 3, pp. 771-779, 2004.

[26] Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J. C., and Harely, R. G.: Particle swarm optimization: basic concepts, variants and applications in power systems, IEEE Trans. Evol. Comput., Vol 12, No. 2, pp. 171-195, 2008.

[27] Engelbrecht, A.P.: *Computational intelligence: an introduction*. John Wiley & Sons, 2007

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

[29] Engelbrecht, A.P.: *Fundamentals of computational swarm intelligence*. Wiley, New York, 2005

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

[31] Mora. J., Baron, J., Santos, J., Payan, M.: An evolutive algorithm for wind farm optimal design. Neurocomput., Vol. 70, pp. 2651–2658, 2007.

[32] Kwong, W. et al.: Wind farm layout optimization considering energy generation and noise propagation. in: *Proceedings of the ASME 2012 international engineering technology conference & computing and information engineering conference*, pp. 1–10. 2012

[33] Gualtieri, G.: A novel method for wind farm layout optimization based on wind turbine selection. Energ. Conver. Mgmt., Vol. 193, pp.106-123, 2019.

[34] Mohiuddin, M., Khan, S. A., Engelbrecht, A. P.: Fuzzy particle swarm optimization algorithms for the open shortest path first weight setting problem. Appl.Intell. Vol. 45, No. 3, pp.598-621, 2016.

[35] Huang, H.: Distributed genetic algorithm for optimization of wind farm annual profits. in: *Proceedings of the IEEE international conference intelligence system applied to power systems*, pp. 1–6, 2007.

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

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

## NOMENCLATURE

| | |
|---|---|
| $a$ | Axial induction factor |
| $z_0$ | Surface roughness |
| $u_0$ | Mean wind speed |
| $Z$ | 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 |
| $r_{dl}$ | Wake radius at distance $x$ downstream of the wind turbine |
| $D$ | Rotor diameter |
| $P_{actual}$ | Total power generated by turbines |
| $P_{ideal}$ | Ideal power generated by turbines |
| $v_j(t+1)$ | Updated velocity of $j^{th}$ particle |
| $v_j(t)$ | Updated velocity of $j^{th}$ particle |
| $c_j$ | Acceleration coefficient for $j^{th}$particle best position |
| $c_g$ | Acceleration coefficient for best position of any particle in swarm |
| $rand_j$ | Random number |
| $rand_g$ | Random number |
| $S_j(t)$ | Current position of particle $j$ at time $t$ |
| $p_j$ | Previous best position of $j^{th}$ particle |
| $p_g$ | Previous best position of the swarm |

### *Greek symbols*

| | |
|---|---|
| $\alpha$ | Entrainment factor |

### *Abbreviations and Acronyms*

| | |
|---|---|
| NIA | Nature-inspired algorithm |
| GA | Genetic algorithm |
| PSO | Particle swarm optimization |
| MPSO | Modified particle swarm optimization |

## ЕФЕКАТ КОЕФИЦИЈЕНАТА УБРЗАЊА КОД АЛГОРИТМА ОПТИМИЗАЦИЈЕ РОЈА ЧЕСТИЦА КОРИШЋЕНОГ У ПРОЈЕКТОВАЊУ РАСПОРЕДА ВЕТРОГЕНЕРАТОРА

**Ш. Рехман, А.А. Кан, Л.М. Алхемс**

Енергија ветра је постала атернатива класичним изворима енергије. Ефикасност ветропарка се базира на доношењу једне важне одлуке, а то је израда оптималног распореда ветрогенератора. Распоред одређује локацију турбине у ветропарку. Сложеност процеса намеће проблем пројектовања распореда ветротурбина, што представља сложен проблем оптимизације. ПСО алгоритам је коришћен у бројним студијама за решавање проблема распореда ветрогенератора. Међутим, није посвећена адекватна пажња групи ПСО параметара, тј. коефицијентима убрзања. С озиром на значај ових коефицијената у раду је извршена прелиминарна анализа коефицијената убрзања коришћењем конвенционалне и модификоване варијанте ПСО алгоритма у примени код пројектовања распореда ветрогенератора. Емпиријски резултати показују да коефицијенти убрзања имају утицаја на квалитет финалног распореда, чиме се постиже већа укупна излазна енергија.