University of Strathclyde Glasgow

Pillai, Ajit C. and Chick, John and Johanning, Lars and Khorasanchi, Mahdi and Barbouchi, Sami (2016) Comparison of offshore wind farm layout optimization using a genetic algorithm and a particle swarm optimizer. In: 35th International Conference on Ocean, Offshore and Arctic Engineering, 2016-06-19 - 2016-06-24. ,

This version is available at https://strathprints.strath.ac.uk/57640/

Proceedings of the ASME 2016 35th International Conference on Ocean, Offshore and Arctic Engineering
OMAE2016
June 19-24, 2016, Busan, South Korea

OMAE2016-54145

COMPARISON OF OFFSHORE WIND FARM LAYOUT OPTIMIZATION USING A GENETIC ALGORITHM AND A PARTICLE SWARM OPTIMIZER

Ajit C. Pillai[*]
Industrial Doctorate Centre for
Offshore Renewable Energy
The University of Edinburgh
Edinburgh, United Kingdom
Email: a.pillai@ed.ac.uk

Dr. John Chick
Institute for Energy Systems
The University of Edinburgh
Edinburgh, United Kingdom

Prof. Lars Johanning
College of Engineering,
Mathematics, and Physical Sciences
University of Exeter
Penryn, United Kingdom

Dr. Mahdi Khorasanchi
Department of Naval Architecture,
Ocean and Marine Engineering
University of Strathclyde
Glasgow, United Kingdom

Dr. Sami Barbouchi
EDF Energy R&D UK Centre
London, United Kingdom

ABSTRACT

This article explores the application of a binary genetic algorithm and a binary particle swarm optimizer to the optimization of an offshore wind farm layout. The framework developed as part of this work makes use of a modular design to include a detailed assessment of a wind farm's layout including validated analytic wake modeling, cost assessment, and the design of the necessary electrical infrastructure considering constraints. This study has found that both algorithms are capable of optimizing wind farm layouts with respect to levelized cost of energy when using a detailed, complex evaluation function. Both are also capable of identifying layouts with lower levelized costs of energy than similar studies that have been published in the past and are therefore both applicable to this problem. The performance of both algorithms has highlighted that both should be further tuned and benchmarked in order to better characterize their performance.

INTRODUCTION

With the development of large offshore wind farms it has become increasingly important to ensure that wind farms are designed such that they use the available space as efficiently as possible.

Wind farm layout optimization tools have grown significantly in recent years from the original tools such as those developed by Mosetti et al. [1] or Grady et al. [2] to include not only the impact the turbine positions have on the energy extracted from the wind, but also to include the impact on the project costs as a result of changes in the layout [3–6]. In recent years, many studies have explored the performance and applicability of different optimization strategies to the wind farm layout optimization problem [7–16]. With the aim of advancing this field further, a layout optimization framework has been developed, including a more detailed approach for assessing wind farm layouts and including as many real world constraints as possible, enabling this framework to be applied to real sites by a project developer.

The levelized cost of energy (LCOE) acts as a single metric which encompasses the annual energy production (AEP) of the wind farm over its lifetime as well as the lifetime project costs.

[*] Address all correspondence to this author.

By using such a metric to evaluate the layouts, it takes into account both the lifetime energy generated by the wind farm and the lifetime costs, allowing a project developer to easily compare the layouts on an economic basis which relates both the energy outputs of the project and the cost inputs. This optimization framework therefore minimizes the LCOE of the wind farm by adjusting the turbine positions, substation positions, and cable routes, ensuring that the effect this has on the AEP and project costs are accurately accounted for.

The LCOE expressed in £/MWh is mathematically given by:

$$LCOE = \frac{\sum_{t=1}^{n} \frac{C_t}{(1+r)^t}}{\sum_{t=1}^{n} \frac{AEP_t}{(1+r)^t}} \tag{1}$$

where C_t is the total costs incurred in year t, n is the project lifetime, AEP_t is the annual energy production in year t, and r is the discount rate of the project.

The present framework has been developed with future UK wind farm sites in mind and therefore includes the consideration of constraints and costs that a future UK offshore wind farm will face. Initial results of this framework previously presented by the authors have shown that it is capable of satisfying real world constraints while at the same time including a validated evaluation function in a manner in which existing work does not [17].

This article deploys this modular framework using two separate optimization algorithms in order to both simultaneously benchmark the framework against existing work, and to identify the differences in performance between the genetic algorithm (GA) and the particle swarm optimization (PSO) algorithm. By deploying these two different optimization algorithms using the same framework, the results can be directly compared advising future work in this field.

APPROACH

The framework deployed for this study is made up of separate modules for the the design of the electrical infrastructure, assessment of the AEP, estimation of the project costs, and for the overall optimization. This approach has allowed alternate wake models, cost functions, and optimization algorithms to be implemented and tested as part of the development. As part of the development, each individual module has been validated independently prior to integrating them into the larger optimization tool. This work, looks specifically at the comparison of two optimization modules for the same case study keeping all other modules in the framework constant. In order to compute the LCOE and thereby get an assessment of each layout, it is necessary for the approach to design the necessary electrical infrastructure, assess the AEP, and estimate the cost in sequence as shown in fig. 1.

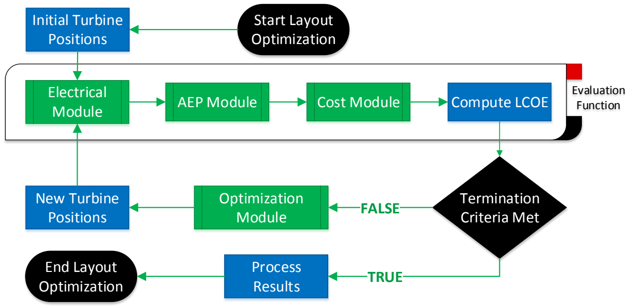

FIGURE 1: MODULAR APPROACH TO WIND FARM LAYOUT OPTIMIZATION

In the case of both optimization algorithms, the assessed LCOE for each layout is an important contributor to how new candidate solutions are generated in subsequent iterations. The LCOE is therefore needed for each layout in order for the optimization algorithms to successfully navigate the search space.

Evaluation of LCOE

As indicated in fig. 1, the assessment of the LCOE is subdivided into three distinct steps. In the first, the electrical infrastructure required for a given turbine layout is determined, then the energy production of the wind farm is assessed, and finally this information is used to estimate the project costs and establish an LCOE for the given layout. The overall approach for the evaluation function is described in greater detail in previous work by the authors [17, 18].

Electrical Infrastructure Optimization. Existing offshore wind farm layout optimization tools generally do not consider any impact on the project cost as a result of changes in the substation positions or intra-array cables [1–5, 7–16]. By including this in the present framework, the accuracy of the cost function is increased and it is easier to differentiate accurately between different layouts. Given that the cost of cables can exceed £500,000 per kilometer installed, it is important that this length be computed accurately [18].

The developed electrical infrastructure optimization tool is unique in its ability to consider not only the electrical constraints of the turbines, substations, and cables, but also the bathymetry, seabed features which define constraint regions, and the physical constraints of the turbines. This therefore, allows the electrical infrastructure optimization tool to propose realistic layouts which satisfy the real constraints of an offshore wind farm developer. It should be noted that as a heuristic approach is used to define the possible cable paths, the electrical infrastructure is therefore not guaranteed to reach proven optimality, but will find a good feasible solution.

The overall approach further divides the optimization of this infrastructure into three separate stages:

1. Determination of substation positions
2. Determination of possible intra-array cable paths
3. Selection of intra-array cable paths to use

The substation positions and the assignment of turbines to a specific substation are found based on applying a modified version of the *kmeans++* clustering algorithm [19]. It has previously been shown that by placing the substation as close as possible to the center of a wind farm, the intra-array cable costs will be reduced [18,20]. Using the cluster center as the substation position therefore minimizes the distance between the substation and all the turbines assigned to that substation. In this tool, the standard *kmeans++* algorithm is modified to account for the capacity constraints on substations, and the fact that within the wind farm area there may be regions where substations cannot be placed [18].

Once the substation positions are determined and the turbines have been assigned to a specific substation, a pathfinding algorithm is used to identify the possible cable paths and the accurate distance that a cable must cover in order to connect any two turbines, or any of the turbines and the substation. The use of the pathfinding algorithm accounts for the fact that there are regions where cables cannot be placed and must therefore navigate around. From this, a capacitated minimum spanning tree (CMST) is constructed based on the cable costs identified through the use of the pathfinding algorithm. In this case, the CMST represents the optimal intra-array cable network given the possible paths under consideration. The CMST is solved using a standard mixed-integer linear programming (MILP) formulation using the commercial solver Gurobi [21]. As cables in offshore wind farms cannot cross one another, a check is done after solving the MILP problem. If any crossings are found, these individual constraints are introduced to the MILP problem and the problem is resolved. This process has been found to solve the MILP problem more quickly than including all the crossing constraints from the beginning [18].

The electrical infrastructure optimization is run first as part of the evaluation function in order to account for the electrical cable losses in the AEP calculation, and the cable costs in the project cost module.

AEP Estimation. The assessment of the AEP includes considering the local wind conditions, modeling the wakes that develop within the wind farm, as well as modeling any other sources of energy loss that are affected by the wind farm layout.

Any device extracting energy from a natural flux such as winds, is known to directly impact that flux. In the case of wind turbines, the region directly behind an operating wind turbine, known as the wake, is characterized by reduced wind speeds and higher levels of turbulence [22–25]. Wakes of multiple turbines are also known to interact with one another, such that when estimating the AEP for an entire wind farm it is important to account for the impact that the wakes have on one another [26,27]. Though a number of kinematic wake models have been implemented into the framework, the present study uses the G.C. Larsen wake model [28, 29]. This model was selected as previous studies have shown this to be a good balance between accuracy and computational intensity [30, 31].

The AEP is assessed by stepping through each wind speed and direction combination and modeling, using a kinematic wake model, the impact that each turbine has on the free wind speed. Each turbine, therefore, experiences conditions based on how the wakes of the upstream turbines impact the free wind speed. The turbine power curve is then used to assess the energy production from each individual wind turbine using the respective wind speed that they experience. From this, the electrical cable losses for that specific set of conditions is then assessed given the intra-array cable layout previously designed. The total generation for this free wind condition is then scaled by the number of hours during the year that this condition would be expected, and the sum of each of these scaled outputs for all the wind conditions under consideration gives the AEP [17].

$$AEP = 8766 \times \sum_{d_i} \sum_{v_i} P(d_i, v_i) \times [E(d_i, v_i) - L(d_i, v_i)] \quad (2)$$

where d_i is the wind direction; v_i is the wind speed; $P(d_i, v_i)$ is the probability of the combination of d_i and v_i; $E(d_i, v_i)$ is the energy production for the wind farm for that combination of free wind speed and direction; and $L(d_i, v_i)$ is the electrical losses associated with that wind speed and direction.

Cost Assessment. The final step in the assessment of a layout is the determination of the costs incurred by the wind farm. For an offshore wind farm, eight principle cost elements have been identified which all have varying degrees of sensitivity to the layout (table 1). Each of the cost elements outlined in table 1 are estimated using a validated cost model which considers not only the positions of the turbines, but also the water depth. The implemented cost model has been validated where possible using available published data and data supplied by active projects currently under development [17]. By including costs which are relative to the turbines' absolute position and their relative position to one another, more accurate project costs can be computed compared to existing optimization frameworks [7–16].

Though some costs such as the foundation costs, and the cable installation costs would be expected to be impacted by the

TABLE 1: COST ELEMENT CONTRIBUTION TO CAPEX

Cost Element	CAPEX	OPEX	Sensitivity to Layout
Turbine Supply	✓	-	Low
Turbine Installation	✓	-	Medium
Foundation Supply	✓	-	Medium
Foundation Installation	✓	-	Medium
Intra-Array Cables	✓	-	High
Decommissioning	✓	-	Medium
Operations and Maintenance	-	✓	Medium
Offshore Transmission Assets	✓	✓	Low

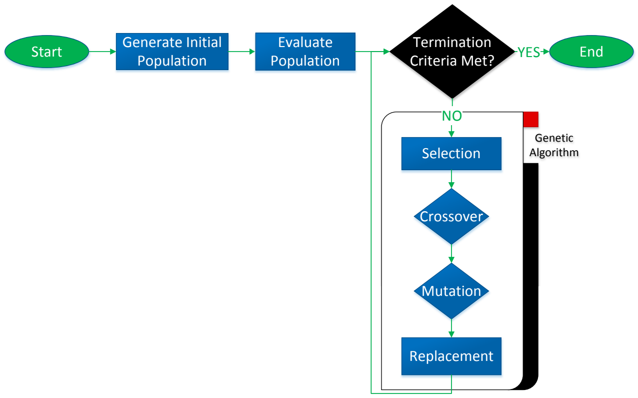

FIGURE 2: FLOWCHART OF THE GENETIC ALGORITHM

soil conditions at the site, previous work has found that even if very detailed geotechnical data is available, a bottom-up cost model tends to validate poorly [3]. Considering this and the fact that the tool would likely be applied at an early stage when geotechnical data would not be available, the present cost relations ignore the geotechnical and soil conditions. Having said that, the modular approach developed would make it straight forward for the optimization process to consider this if the data was available and the cost relationships established.

Genetic Algorithm

The final step of the procedure is to use an optimization algorithm to alter the wind turbine layouts given the LCOE values of already assessed layouts. GAs represent a family of bio-inspired population based heuristic optimization algorithms that borrow ideas from natural evolution as observed in biological systems [32]. GAs are commonly deployed as they represent a family of generic algorithms which can be applied to a wide range of problems of varying degrees of complexity [33]. As such, GAs have commonly been applied to the offshore wind farm layout optimization problem with good quality solutions being found [1–3, 5, 34, 35].

In a GA, the candidate solutions within the population are formulated such that the encoding can be considered a *genome* which defines the individual solutions. The evaluation function is used to determine the *fitness* of each solution. In this case, the fitness of each layout is the LCOE, with small LCOE values considered to be superior in fitness. Under these terms, the GA then tries to use solutions with favorable fitness values to generate new candidate solutions. Solutions with higher fitness values (in this case, layouts with lower LCOE values) have a higher probability of contributing genetic material towards new candidate solutions. The flowchart in fig. 2 shows the principle steps of a GA. After selecting pairs of individuals among the population to reproduce (i.e. to generate new candidate solutions), the pair undergoes what is referred to as *crossover*. During crossover, the two *parent* solutions are combined in such a way that two new solutions are generated, each with 50% of their genome being defined by each parent. In this way, the two candidate child solutions represent a combination of the two parents, hopefully exploiting the good elements of the two parents to create a solution with a superior fitness value. In order to ensure that the GA does not get stuck at a local solution, a *mutation* operator is used to randomly alter the child solutions. This process is repeated until the solutions converge, or there is insufficient diversity within the remaining population for the process to continue effectively. An *elitism factor* is used to define what proportion of the population must be replaced with new solutions in each generation.

In this case, as the wind farm region has been discretized, the problem can be solved using a binary genetic algorithm. A binary genetic algorithm is one in which the genome is represented as a binary string. In this case, each bit of the genome represents the presence of a turbine in a specific cell of the discretized wind farm area. Given the binary GA approach, crossover is implemented using a uniform crossover mask. This is a method in which if crossover occurs, a second binary string the same length as the genome is generated. This string, however, represents which parent the children should inherit each individual bit from (i.e. each bit in the crossover mask represents which parent contributes to that specific bit in the child solution). To generate two complementary children, the crossover mask has every bit flipped to generate a second child. The mutation operator also works on a bitwise basis, cycling through each bit in the child solutions with a low probability that each bit gets flipped.

The key parameters that define a GA are therefore the size of the population; the probabilities associated with mutation and crossover; and the elitism factor. In the present implementation, adaptive parameters are used for the mutation and crossover rates as this has been shown to improve convergence and foster diversity within the population [36]. The crossover and mutation probabilities are therefore a function of the solution's fitness value (f) compared to both the population's mean fitness $\left(\bar{f}\right)$, and the population's best fitness (f_{max}).

$$p_c = \frac{k_1\left(f_{max} - f'\right)}{f_{max} - \bar{f}} \qquad \text{for} \qquad f' \geq \bar{f} \qquad (3)$$

$$p_c = k_3 \qquad \text{for} \qquad f' < \bar{f} \qquad (4)$$

$$p_m = \frac{k_2\left(f_{max} - f\right)}{f_{max} - \bar{f}} \qquad \text{for} \qquad f \geq \bar{f} \qquad (5)$$

$$p_m = k_4 \qquad \text{for} \qquad f < \bar{f} \qquad (6)$$

where p_c and p_m are respectively the probabilities of crossover and mutation. The population size was kept at 50 individuals, and an elitism factor of 25% was used.

Particle Swarm Optimization

The GA is often thought of as a competitive population based optimization algorithm, as a solution's ability to contribute to the improvement among its peers is based on its own fitness. The PSO on the other hand is considered to be a cooperative population based optimization algorithm in which the candidate solutions (now thought of as particles) explore the search space while aware of their neighbors [33]. Like the GA, this algorithm is also analogous to a biological system, though unlike the GA rather than based on an evolutionary process, it is based on how birds flock or fish school [37]. The general approach is shown in fig. 3.

In a PSO, the particles are randomly seeded in a manner similar to that of a GA, however, from here the two algorithms differ quite significantly. A PSO treats the candidate solutions and particles exploring the search space. In this analogy, the change from generation to generation is encapsulated in what is thought of as the particle's velocity through the search space. This velocity is partially random to avoid local minima, partially based on the particle's historical best position within the search space, and partially on the population's best position. In this way, by including the particle's previous best position, and the global best positions, the particle tries to exploit the knowledge of the swarm, while the random element helps the particle explore the search space. A major difference between the PSO and the GA is that the PSO allows particles to decline in fitness from generation to generation, recognizing that it may lead to better future positions. In the PSO, each particle's position at any given iteration is related to its past position by:

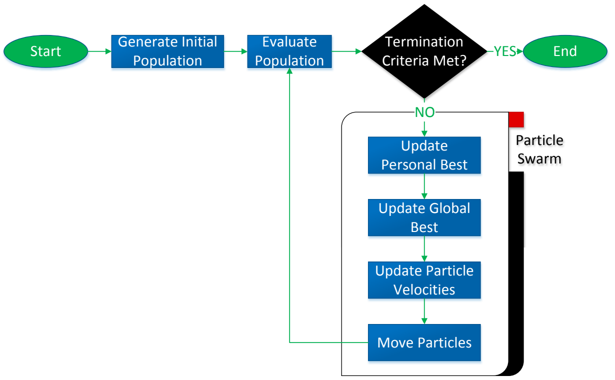

FIGURE 3: FLOWCHART OF THE PARTICLE SWARM OPTIMIZATION ALGORITHM

$$x_i = x_{i-1} + v_i; \qquad (7)$$

where the velocity v_i is given by:

$$v_i = C_1 v_{i+1} + C_2(p - x_{i-1}) + C_3(g - xi - 1) + C_4 \times rand \qquad (8)$$

where C_1, C_2, C_3, and C_4 are coefficients representing the weighting of the different contributors determined by tuning the PSO to the problem at hand; p is the best historical position of the particle in question, g is the best historical position of the swarm, and *rand* is a random number between 0 and 1.

Like the GA, the PSO was implemented with a binary encoding. This complicates matters slightly because the position for each bit must be either 0 or 1. The continuous velocity, must therefore be adjusted such that it corresponds to the bit in question changing to either a 0 or a 1. To solve this, a sigmoid transfer function is commonly used to convert the velocity for a given bit to a probability of the bit being a 1 [38–40].

Using this standard transfer function, however, introduces a challenge in satisfying the number of turbines constraint. In order to easily check and satisfy this constraint, the transfer function was redefined such that it represented the probability that a bit is flipped. This then allowed the algorithm to ensure that equal numbers of 1's and 0's were flipped thereby preserving the number of turbines within the wind farm. This, however, required a change in the transfer function as both highly negative and highly positive velocities should correspond to a high probability of the bit flipping. This was done by replacing the s-shaped sigmoid function with a v-shaped function [40]. Figure 4 shows typical s-shaped (sigmoid) and v-shaped transfer functions.

In general, PSO has been found to be suitable for solving similar problems as the GA. However, the PSO tends to require a smaller population to reach similar quality solutions, and as

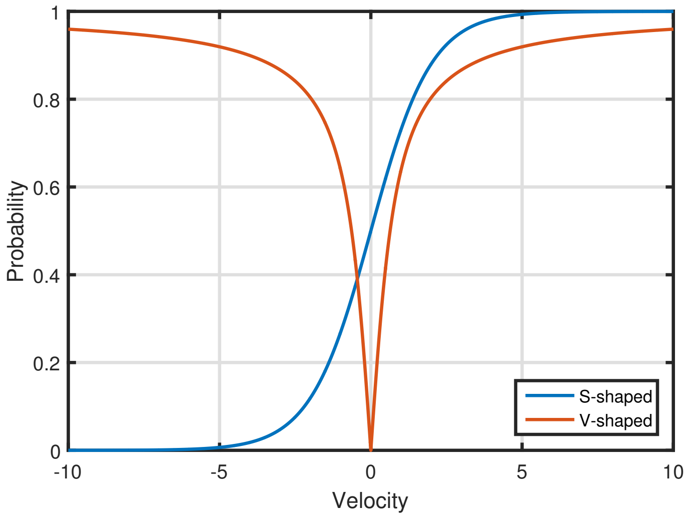

FIGURE 4: TRANSFER FUNCTIONS FOR BINARY PARTICLE SWARM OPTIMIZER

the number of function evaluation calls is related to the size of the population, a smaller population will result in a quicker execution time. PSOs, however, can suffer from premature convergence and sensitivity to local solutions when the population is insufficiently large. For these simulations a population size of 30 was used.

RESULTS

Case Definition

For this comparison of the GA and the PSO it was desirable to use a case which had previously been addressed in other layout optimization projects in order to simultaneously highlight the advantages of including a comprehensive layout evaluation function as well as allowing the optimization algorithms to be benchmarked.

In one of the first works to explore the optimization of wind farm layouts, Mosetti et al. [1] laid out case studies which have been commonly used since. Each of these cases considers a square shaped wind farm area (2 km by 2 km) discretized into 100 possible turbine positions. Given the discretization of the wind farm area, the optimization problem can be implemented as a binary decision problem.

One of these cases, which is under consideration in this study, considers a "case of multiple wind direction with constant intensity" [1]. This wind regime is defined as having a constant wind speed of 12 $\mathrm{m\,s^{-1}}$ with an equal probability that the wind will blow from any direction. For the computation, the wind direction is defined as being discretized into 36 sectors each of 10° width.

The original definition of the cases omitted the water depth, the location of ports to be used relative to the wind farm, or any regions that must be avoided as these were not seen as impacting the layouts. However, all are used by by the present evaluation function in the determination of a layout's LCOE. In order to keep the case as close as possible to the original definition while using the more detailed evaluation function developed here, a constant water depth was assumed, the port was assumed to be very far away relative to the size of the wind farm, and it was assumed that no constraint regions existed within the wind farm area.

As the case study does not define the number of turbines under consideration, the case study was executed for two different wind farm sizes (19 and 39 turbines) corresponding to the results shown in two layout optimization studies using these cases [1,2]. In order to compare fairly, the published optimal layouts for this case study have been re-evaluated using our evaluation function in order to ensure that a direct layout-to-layout comparison can be done for both wind farm sizes.

Though more recent work has explored the same case study, these have on the whole explored the application of more advanced optimization algorithms than the original, making use of the same evaluation function. These studies have also either not used the same number of turbines or the same discretized grid making it difficult to make a fair comparison [7–16]. The work by Mosetti et al. [1] and Grady et al. [2] remain the reference cases which new work is compared against. The present work has focused on the improvement of the evaluation function by adding the detail necessary for the tool to be applied to real sites by a project developer. The results presented here are meant to highlight that even with the increased detail in the evaluation function, these optimization techniques are of interest and can highlight improvements over the original work in the field [1,2].

TABLE 2: ASSESSMENT OF LAYOUTS

	Number of Turbines	AEP [GWh]	Cost [£]	LCOE [£/MWh]
Mosetti et al. [1]	19	81.71	3.770×10^{8}	540.25
GA-19	19	81.77	3.771×10^{8}	539.88
PSO-19	19	82.11	3.769×10^{8}	537.49
Grady et al. [2]	39	156.99	5.620×10^{8}	419.00
GA-39	39	159.23	5.613×10^{8}	412.60
PSO-39	39	159.00	5.616×10^{8}	413.50

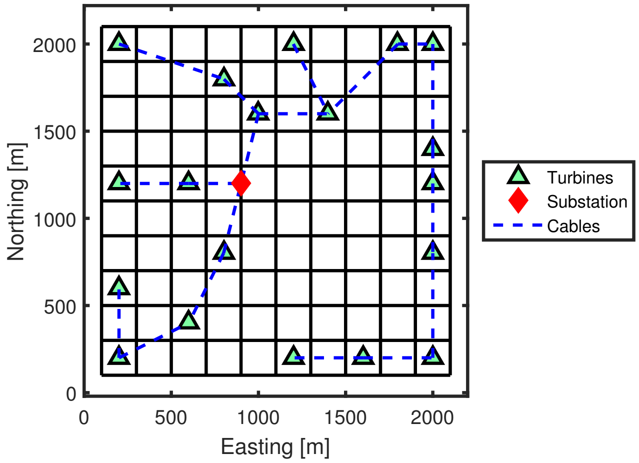

FIGURE 5: LAYOUT PRODUCED BY GA WITH 19 TURBINES

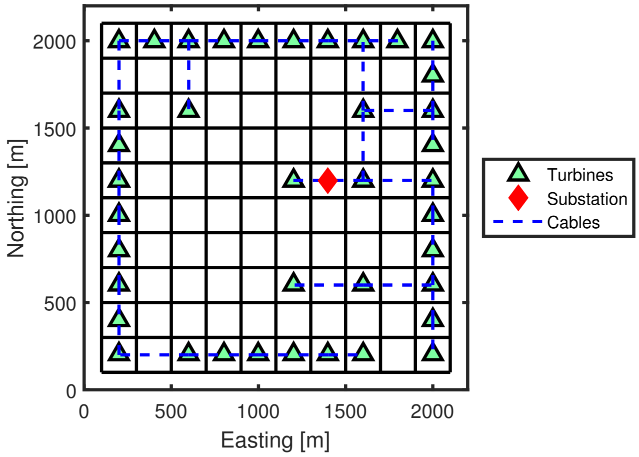

FIGURE 6: LAYOUT PRODUCED BY GA WITH 39 TURBINES

Genetic Algorithm

Figures 5 and 6 show the layouts produced by the GA. As can be seen, the proposed layouts differ significantly from one another as a result of the additional 20 turbines in the larger wind farm. The plots shown in figs. 7 and 8 show that in both cases the solution converged and the diversity within the population fell below the required threshold terminating the optimization run. Table 2 shows the results attained in this study compared to the layouts proposed by the benchmark studies [1, 2].

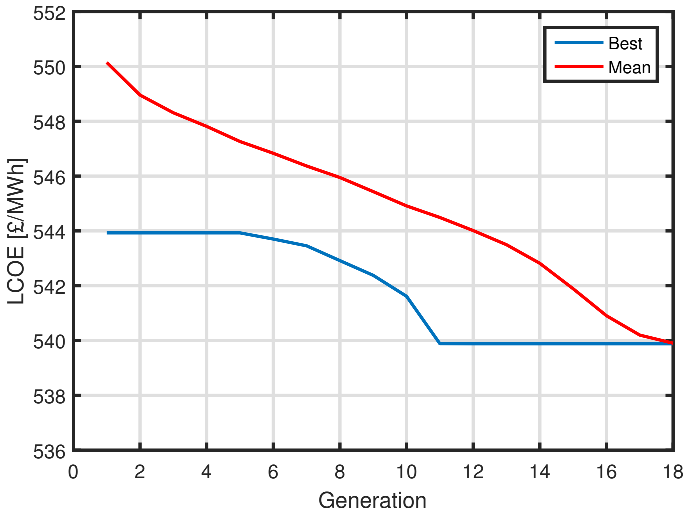

FIGURE 7: CONVERGENCE PLOT FOR GA WITH 19 TURBINES

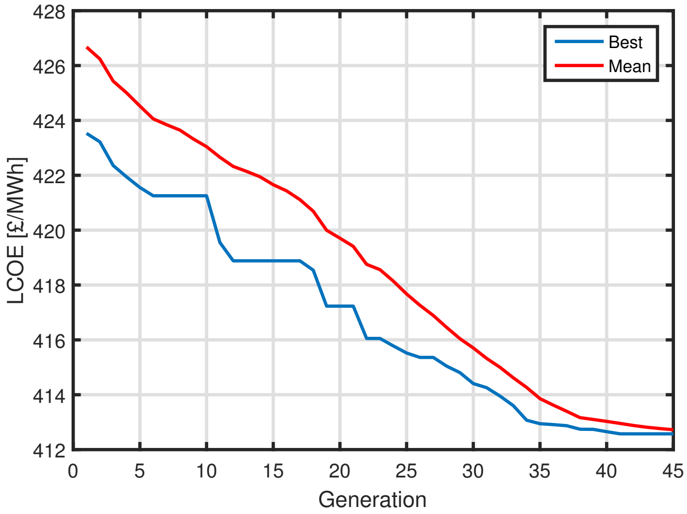

FIGURE 8: CONVERGENCE PLOT FOR GA WITH 39 TURBINES

Particle Swarm Optimization

Similar to the results of the GA, the PSO was run for both wind farm sizes in order to compare the layouts to both those generated by the adaptive GA and those produced by the previous studies [1, 2]. The layouts produced by the PSO are shown in figs. 9 and 10.

Like the GA, the results in table 2 indicates that the PSO produces layouts for both wind farm sizes that have lower LCOE values than the past studies [1,2]. Interestingly, the PSO does not create the same solutions as the GA.

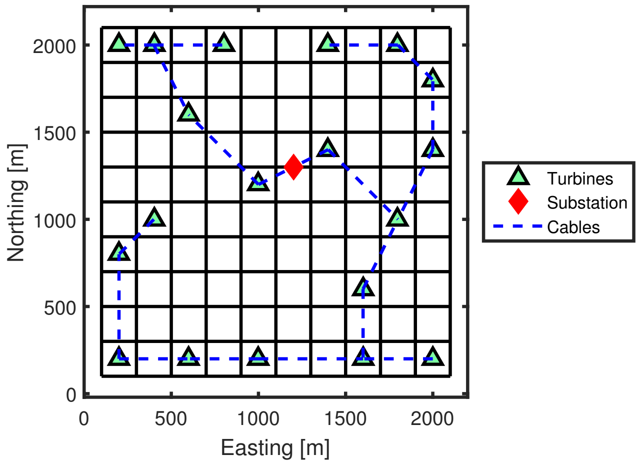

FIGURE 9: LAYOUT PRODUCED BY PSO WITH 19 TURBINES

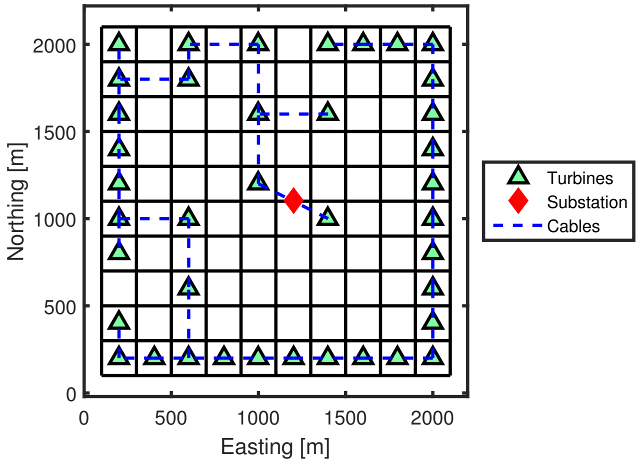

FIGURE 10: LAYOUT PRODUCED BY PSO WITH 39 TURBINES

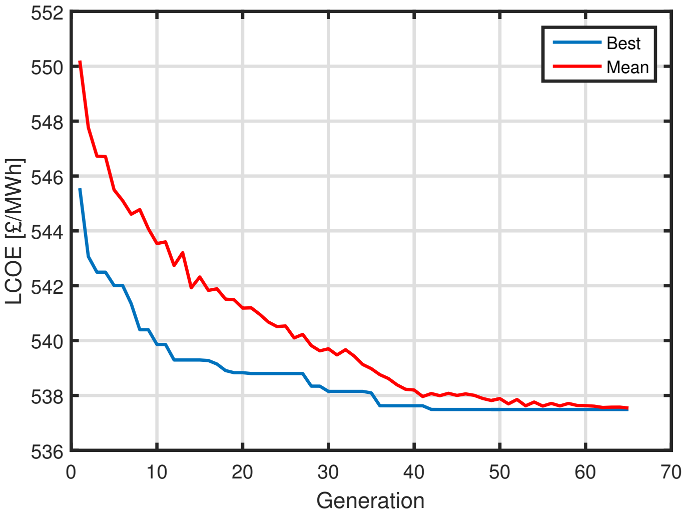

FIGURE 11: CONVERGENCE PLOT FOR PSO WITH 19 TURBINES

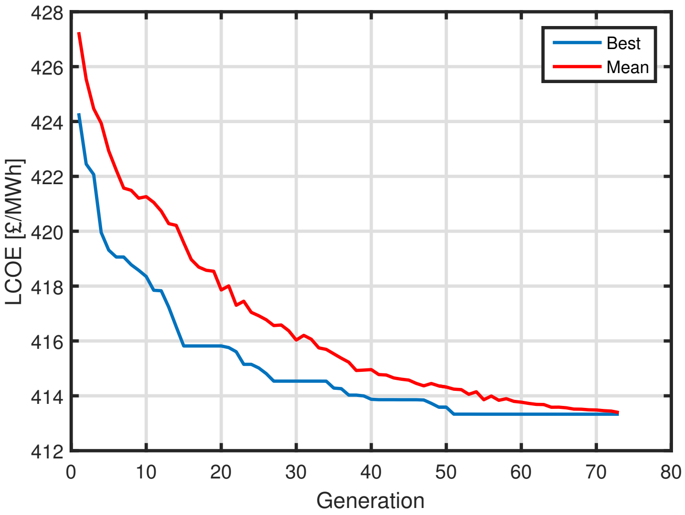

FIGURE 12: CONVERGENCE PLOT FOR PSO WITH 39 TURBINES

DISCUSSION AND CONCLUSIONS

From the four solutions presented here, it can be seen that the new framework shows that given a more accurate evaluation function, both the GA and the PSO are capable of finding better solutions than those found by previous studies [1, 2]. This is an important result as it indicates that even given the increased complexity of the evaluation function, these optimization algorithms are relevant choices.

In general as can be seen in the convergence plots (figs. 7, 8, 11 and 12), all four solutions represent the best solutions found by the solvers prior to convergence. The performance of the GA with the small wind farm, however, showed very quick convergence indicating that the population may have prematurely converged. This suggests that though the solution found is good and in fact better than that found by the literature for the same sized wind farm, it could be further improved by further tuning or executing multiple runs. In fact, comparing it to the PSO results for the same conditions, one can see that the PSO finds a much better solution than both the implemented GA and the results of past studies used as a benchmark [1, 2]. In fact, as heuristic algorithms are deployed, there is no guarantee that proven optimality has been reached and both optimizers could be further tuned to

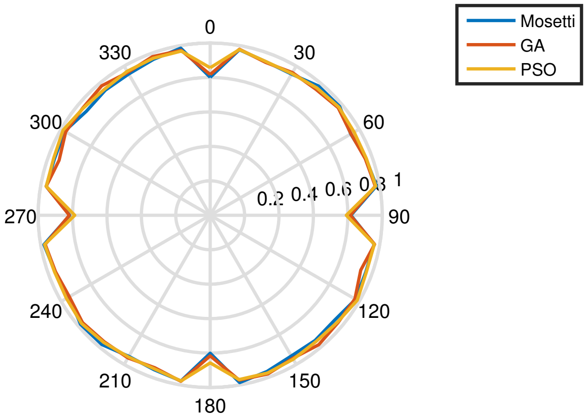

FIGURE 13: WAKE EFFICIENCY BY WIND DIRECTION FOR 19 TURBINE WIND FARM

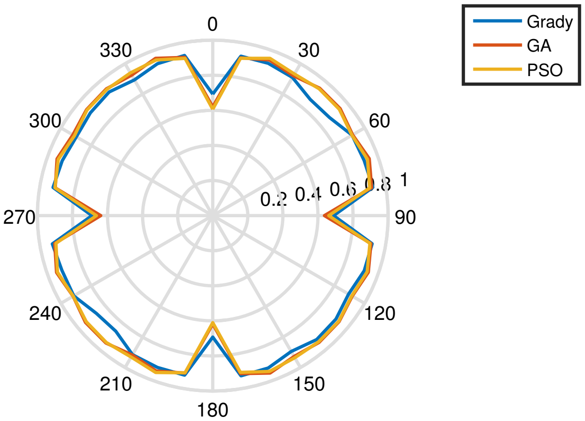

FIGURE 14: WAKE EFFICIENCY BY WIND DIRECTION FOR 39 TURBINE WIND FARM

ensure that they have not prematurely converged.

Both the layouts proposed by this framework for the 39 turbine wind farm (figs. 6 and 10) appear to exploit the symmetry of the wind resource by the majority of the turbines along the edge of the wind farm with the GA slightly outperforming the PSO in this case. For both the proposed layouts, this leads to significant wakes along the four wind direction sectors that are aligned with these edges, however, it also leads to relatively high wake efficiency for the remaining 32 directions (figs. 13 and 14). Taking this idea to the extreme, a case was executed using the GA where three turbines were to be placed in the 64 central cells, and the remaining 36 turbines were locked to the 36 cells along the edges (results shown in fig. 15). For this restricted case, the GA was, however, unable to find a solution that was superior to the layout presented in fig. 6 with the best solution in this restricted case having an LCOE of £412.77/MWh. Though this is very similar to the GA result shown in fig. 6 (LCOE of £412.60/MWh), it is marginally higher, indicating that the optimal solution is likely not symmetrical. It is important to note that though the resource may be symmetrical, the cost functions are not, and we would not therefore expect the optimal layout to be symmetrical.

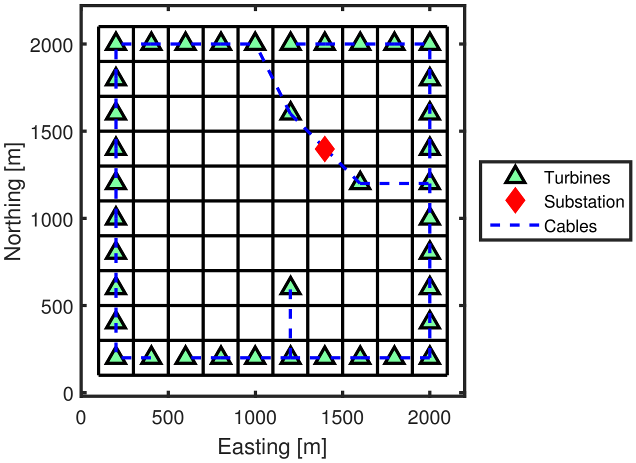

FIGURE 15: LAYOUT PRODUCED BY GA WITH 36 TURBINES FIXED TO EDGE

The wake efficiency plots (figs. 13 and 14) indicate the relative efficiency of the different wind sectors across the different layouts. As can be seen, there is a variation in performance across all sectors, including the inline directions which result in significant reductions in AEP across all the layouts. Interestingly, for the small wind farm it appears to be the slight increase in wake efficiency along these inline wind directions which results in the marginal increase in AEP. For the larger wind farm, however, there are significant increases along the 40°, 50°, 220°, and 230° directions in addition to the North and South inline cases. For both the GA and PSO, however, the East and West wind directions are less efficient than the reference layout.

Given the simplicity of the case at hand, the inclusion of the electrical infrastructure optimization does not significantly affect the layouts produced as it has a very minimal impact on both the AEP and cost. Given the small size of the turbines considered in this case study (659 kW), all the turbines considered could be connected on a single string reducing the sensitivity of the cable cost to the layout. Furthermore, as the site was assumed to be a constant water depth, the cost variation across this site does not represent a realistic case and the very slight improvements in cost observed come principally from reductions in the installation processes. A real wind farm will be expected to observe a more significant reduction in cost as a result of improvements made to the layout. Given this, the present case is dominated by the AEP term with the electrical infrastructure and cost modules yielding small impacts to the LCOE. However, it can be expected that for large offshore wind farms with real constraints impacting where substations, turbines, and cables can be placed the inclusion of these modules will be necessary in order to ensure that the layouts produced are feasible and to ensure that the LCOE is accurately estimated for the layout.

As the PSO does not require particles to improve in fitness from generation to generation, each iteration of the PSO requires the same number of evaluation calls. The GA, however, looks to replace a specific proportion of the population each generation with superior individuals. The number of layouts that need to be generated and therefore the total number of evaluation calls varies from generation to generation. For the same size population, the PSO would therefore be expected to be faster, as in general fewer evaluation calls will be needed, especially after the results begin to converge. Having said that, PSOs are generally run with a smaller population than their equivalent GA further reducing the total number of evaluation calls required and therefore the execution time.

The initial results shown here have indicated that both the GA and PSO implemented here are capable of finding superior layouts to those that have been identified in previous published studies [1, 2]. At the same time, both the GA and PSO have found solutions of similar quality and as neither optimizer outperforms the other consistently, it has indicated that both are applicable to this problem, though for the reasons stated earlier, the PSO may offer significant time savings when compared to the GA. Future work can explore the application of this framework to additional test cases in order to further benchmark the framework as well as aid in the tuning of both optimization algorithms. A principle output of this work is that though the objective function has increased complexity due to the inclusion of a more detailed cost function and the optimization of the electrical infrastructure, these optimization algorithms are still effective for addressing this problem. While previous studies [7–16] have addressed the problem using a simple evaluation function, this study has advanced the field by including instead a detailed evaluation function representative of what a project developer would use to assess future projects. This demonstrates that the present framework would be of use to a wind farm developer. Future work should also explore the importance of including the electrical infrastructure optimization through the application of the framework to large real offshore wind farms. An important point to note is that the results presented represent only single runs of the optimization algorithm. Given the stochastic nature of the optimization algorithms, future work should explore performing multiple runs and looking at the average results of these ensembles of runs.

ACKNOWLEDGMENT

This work is funded in part by the Energy Technologies Institute (ETI) and RCUK energy program for IDCORE (EP/J500847/1).

REFERENCES

[1] Mosetti, G., Poloni, C., and Diviacco, B., 1994. "Optimization of wind turbine positioning in large wind-farms by means of a genetic algorithm". *Journal of Wind Engineering and Industrial Aerodynamics,* ***51***(1), pp. 105–116.

[2] Grady, S., Hussaini, M., and Abdullah, M., 2005. "Placement of wind turbines using genetic algorithms". *Renewable Energy,* ***30***(2), feb, pp. 259–270.

[3] Elkinton, C. N., 2007. "Offshore Wind Farm Layout Optimization". Doctor of philosophy dissertation, University of Massachussetts Amherst.

[4] Elkinton, C. N., Manwell, J. F., and McGowan, J. G., 2008. "Algorithms for offshore wind farm layout optimization". *Wind Engineering*, pp. 67–83.

[5] Réthoré, P.-E., Fuglsang, P., Larsen, T. J., Buhl, T., and Larsen, G. C., 2011. *TOPFARM wind farm optimization tool.* Riso DTU National Laboratory for Sustainable Energy.

[6] Chowdhury, S., Zhang, J., Messac, A., and Castillo, L., 2013. "Optimizing the arrangement and the selection of turbines for wind farms subject to varying wind conditions". *Renewable Energy,* ***52***(315), apr, pp. 273–282.

[7] Marmidis, G., Lazarou, S., and Pyrgioti, E., 2008. "Optimal placement of wind turbines in a wind park using Monte Carlo simulation". *Renewable Energy,* ***33***(7), jul, pp. 1455–1460.

[8] Huang, H.-S., 2009. "Efficient hybrid distributed genetic algorithms for wind turbine positioning in large wind farms". *IEEE International Symposium on Industrial Electronics*(ISIE), pp. 2196–2201.

[9] Ituarte-Villarreal, C. M., and Espiritu, J. F., 2011. "Optimization of wind turbine placement using a viral based optimization algorithm". *Procedia Computer Science,* ***6***, jan, pp. 469–474.

[10] Du Pont, B. L., and Cagan, J., 2012. "An Extended Pattern Search Approach to Wind Farm Layout Optimization". *Journal of Mechanical Design,* ***134***(8), p. 081002.

[11] Couto, T. G., Farias, B., Diniz, A. C. G. C., and Morais, M. V. G. D., 2013. "Optimization of Wind Farm Layout Using Genetic Algorithm". *10th World Congress on Structural and Multidisciplinary Optimization*, pp. 1–10.

[12] Zhang, P. Y., 2013. "Topics in Wind Farm Layout Optimisation: Analytical Wake Models, Noise Propagation, and

Energy Production". Master of applied science dissertation, University of Toronto.

[13] Geem, Z. W., and Hong, J., 2013. "Improved Formulation for the Optimization of Wind Turbine Placement in a Wind Farm". *Mathematical Problems in Engineering,* ***2013***(1), pp. 1–5.

[14] Chen, Y., Li, H., Jin, K., and Song, Q., 2013. "Wind farm layout optimization using genetic algorithm with different hub height wind turbines". *Energy Conversion and Management,* ***70***, jun, pp. 56–65.

[15] Zhang, P. Y., Romero, D. a., Beck, J. C., and Amon, C. H., 2014. "Solving wind farm layout optimization with mixed integer programs and constraint programs". *EURO Journal on Computational Optimization,* ***2***(3), jul, pp. 195–219.

[16] Shakoor, R., Yusri, M., Raheem, A., and Rasheed, N., 2016. "Wind farm layout optimization using area dimensions and definite point selection techniques". *Renewable Energy,* ***88***, pp. 154–163.

[17] Pillai, A. C., Chick, J., Johanning, L., Khorasanchi, M., and Pelissier, S., 2015. "Optimisation of Offshore Wind Farms Using a Genetic Algorithm". In Proceedings of the Twenty-fifth (2015) International Ocean and Polar Engineering Conference, pp. 644–652.

[18] Pillai, A. C., Chick, J., Johanning, L., Khorasanchi, M., and de Laleu, V., 2015. "Offshore wind farm electrical cable layout optimization". *Engineering Optimization,* ***47***(12), pp. 1689–1708.

[19] Arthur, D., and Vassilvitskii, S., 2006. "k-means ++ : The Advantages of Careful Seeding". *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete Algorithms,* ***8***, pp. 1–11.

[20] Scutariu, M., Yi, X., and Transmission, P., 2015. "Optimisation of offshore wind farm collection systems". *Proceedings of EWEA Offshore 2015*, pp. 1–8.

[21] Gurobi Optimization Inc., 2015. Gurobi Optimizer Reference Manual.

[22] Barthelmie, R. J., Folkerts, L., Larsen, G. C., Frandsen, S. T., Rados, K., Pryor, S. C., Lange, B., and Schepers, G., 2006. "Comparison of Wake Model Simulations with Offshore Wind Turbine Wake Profiles Measured by Sodar". *Journal of Atmospheric and Oceanic Technology,* ***23***(7), jul, pp. 888–901.

[23] Barthelmie, R. J., Hansen, K., Frandsen, S. T., Rathmann, O., Schepers, J. G., Schlez, W., Phillips, J., Rados, K., Zervos, A., Politis, E. S., and Chaviaropoulos, P. K., 2009. "Modelling and Measuring Flow and Wind Turbine Wakes in Large Wind Farms Offshore". *Wind Energy,* ***12***(June), pp. 431–444.

[24] Renkema, D. J., 2007. "Validation of wind turbine wake models". Master of science dissertation, TU Delft.

[25] Makridis, A., and Chick, J., 2013. "Journal of Wind Engineering Validation of a CFD model of wind turbine wakes with terrain effects". *Jnl. of Wind Engineering and Industrial Aerodynamics,* ***123***, pp. 12–29.

[26] Katic, I., Højstrup, J., and Jensen, N., 1986. "A Simple Model for Cluster Efficiency". *European Wind Energy Conference and Exhibition 1986*(October), pp. 407–410.

[27] Schlez, W., and Neubert, A., 2009. "New developments in large wind farm modelling". *Proceedings of EWEC 2009*.

[28] Larsen, G. C., 1988. A Simple Wake Calculation Procedure. Tech. rep., Risø National Laboratory.

[29] Göçmen, T., Laan, P. V. D., Réthoré, P.-e., Diaz, A. P., Larsen, G. C., and Ott, S., 2016. "Wind turbine wake models developed at the technical university of Denmark: A review". *Renewable and Sustainable Energy Reviews,* ***60***, pp. 752–769.

[30] Pillai, A. C., Chick, J., and de Laleu, V., 2014. "Modelling Wind Turbine Wakes at Middelgrunden Wind Farm". In Proceedings of EWEA 2014.

[31] Gaumond, M., Rethore, P., and Bechmann, A., 2012. "Benchmarking of Wind Turbine Wake Models in Large Offshore Windfarms". *Proceedings of the Science of Making Torque from Wind Conference*.

[32] Holland, J. H., 1992. "Genetic algorithms". *Scientific American,* ***July***.

[33] Burke, E. K., and Kendall, G., 2013. *Search Methodologies*, second ed. Springer US, Boston, MA.

[34] Herbert-Acero, J., Probst, O., Réthoré, P.-E., Larsen, G., and Castillo-Villar, K., 2014. "A Review of Methodological Approaches for the Design and Optimization of Wind Farms". *Energies,* ***7***(11), oct, pp. 6930–7016.

[35] Tesauro, A., Réthoré, P., and Larsen, G., 2012. "State of the art of wind farm optimization". *Proceedings of EWEA 2012*, pp. 1–11.

[36] Srinivas, M., and Patnaik, L. M., 1994. "Adaptive probabilities of crossover and mutation in genetic algorithms". *IEEE Transactions on Systems, Man, and Cybernetics,* ***24***(4), pp. 656–667.

[37] Eberhart, R., and Kennedy, J., 1995. "A new optimizer using particle swarm theory". *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43.

[38] Kennedy, J., and Eberhart, R., 1997. "A discrete binary version of the particle swarm algorithm". *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation,* ***5***, pp. 4–8.

[39] García-Gonzalo, E., and Fernández-Martínez, J. L., 2012. "A Brief Historical Review of Particle Swarm Optimization (PSO)". *Journal of Bioinformatics and Intelligent Control,* ***1***(1), pp. 3–16.

[40] Mirjalili, S., and Lewis, A., 2013. "S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization". *Swarm and Evolutionary Computation,* ***9***, pp. 1–14.