

Fast and Effective Multi-Objective Optimisation of Wind Turbine Placement

Raymond Tran

Junhua Wu

Christopher Denison

Thomas Ackling

Markus Wagner

Frank Neumann

Evolutionary
Computation Group
The University of Adelaide
Adelaide, SA 5005, Australia

ABSTRACT

The single-objective yield optimisation of wind turbine placements on a given area of land is already a challenging optimization problem. In this article, we tackle the multi-objective variant of this problem: we are taking into account the wake effects that are produced by the different turbines on the wind farm, while optimising the energy yield, the necessary area, and the cable length needed to connect all turbines.

One key step contribution in order to make the optimisation computationally feasible is that we employ problem-specific variation operators. Furthermore, we use a recently presented caching-technique to speed-up the computation time needed to assess a given wind farm layout. The resulting approach allows the multi-objective optimisation of large real-world scenarios within a single night on a standard computer.

Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search—*Heuristic Methods*

General Terms

Algorithms, Performance

Keywords

Wind Power, Wind Farm Layout, Multi-Objective Optimisation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'13, July 6–10, 2013, Amsterdam, The Netherlands.
Copyright 2013 ACM 978-1-4503-1963-8/13/07 ...\$15.00.

1. INTRODUCTION

Wind power is an increasingly important source of growth within the global renewable energy market. Global cumulative production capacity is forecast to increase to almost 500 Gigawatts by 2016. This is more than double the figure recorded in 2011 [6]. *Wind turbine placement optimisation* is the process of determining the placement of wind turbines in Wind farms. This enables wind farm installations to maximise their cost-effectiveness, thereby increasing their competitiveness in the renewable energy market. In the single-objective yield optimisation of wind turbine placements, the wind turbines are placed within a specified land or offshore area, such that the output power is maximised.

In this paper, we demonstrate the multi-objective variant of the Wind Turbine Placement problem. Three different objective combinations are considered: output power and the area of the convex hull, output power and the total distance of the minimum spanning tree, and a combination of output power, convex hull area and minimum spanning tree. An improved version of the implementation of the wind turbine placement model [10] presented in [15] is used to evaluate the power output of the wind farm layouts. We use the multi-objective evolutionary algorithms NSGA-II [3], SPEA2 [20] and IBEA [19] to produce a number of different wind turbine layouts. These algorithms are combined with the problem-specific variation operators that are presented in the paper.

1.1 Related work

The optimal siting of wind turbines on a given area of land is a complex optimisation problem that is hard to solve by exact methods. The decision space is non-linear with respect to how sited turbines interact, when considering wake loss and energy capture. Several bio-inspired computation techniques such as evolutionary algorithms [5, 7] and particle swarm optimisation [9] have been used for the optimisation.

The different approaches for the single-objective wind farm layout problem are summarized in [12]. For example, Wan et al. [16–18] use cell-based approaches and compare different bio-inspired algorithms, each applied to the same set of wind farm models and parameters. An alternative to cell placement was explored in [10]: each turbine's location is a decision variable pair of real-valued, spatial (x,y) coordinates. In that paper a simple evolution strategy (ES) is used to optimise very small wind farms. In general, an ES is effective because it is easily parallelised and it self-adapts the extent to which it perturbs decision variables when generating

a new potential solution. A more powerful ES called CMA-ES has been used for the same problem in [14]. Despite being computationally expensive, it allows for the effective optimisation of huge layouts for up to 1000 turbines. Recently, a random local search was presented that combines a problem-specific operator with an asymptotic speed-up of the computation time of the wake effects [15]. The resulting algorithm can effectively optimise huge layouts overnight, instead of requiring weeks on specialised computing servers.

Difficulties such as the expensive wake effects computation carry over from the single-objective problem to the multi-objective problem, which we see as one reason why the latter came up just recently (see, e.g. [2, 11, 13]). Common investigated objectives are the maximization of energy yield, the minimization of land cost, and the minimization of environmental impact. However, very often the results leave room for improvements, either because simplified wake models (to make the optimisation computationally feasible), or because the optimisation algorithms are not adjusted to the real-world scenarios. In [13] a first step was taken to combine a general purpose multi-objective algorithm with problem-specific repair strategies, while using the Park wake model for the energy yield computations. In contrast to their work, we employ problem-specific variation operators right from the beginning, investigate different optimisation algorithms and more scenarios, we use a faster wake calculation, and the cable length costs are added as an additional objective.

We proceed as follows. In Section 2, we formally introduce the multi-objective wind turbine placement problem and the different objectives that are subject to our investigations. New variation operators such as mutation and crossover for this highly constrained optimisation problem are described in Section 3. We report on our experimental investigations with popular evolutionary multi-objective optimisation algorithms in Section 4 and finish with some concluding remarks.

2. PRELIMINARIES

In the following, we outline the different objectives and constraints that we considered for the wind farm optimisation.

Let $X = \{x_1, \dots, x_n\}$ and $Y = \{y_1, \dots, y_n\}$ be a set of x and y coordinates of n wind turbines in the plane. The goal is to find a set of coordinates such that the energy output of the whole wind farm is maximised. At the same time, the total length of the cable necessary to interconnect the turbines, as well as the area necessary for the wind farm, should be minimised. Furthermore, the layout has to respect certain constraints. Each of the aforementioned attributes is set to the solution as a separate objective.

2.1 Energy Yield Prediction

2.1.1 Energy Output

The overall energy output of the wind farm varies depending on the chosen coordinates, because we have to take the wake effects into account. We consider the Park wake model, as it is a trade-off between highly simplifying wake models and computationally extremely expensive simulations based on fluid-dynamics. In this model, the wake effects on a turbine i change the wind resources available to it along different directions by reducing the *scale* parameter c of the Weibull distribution estimated for the entire farm, which is also called the free-stream wind resource. This wind resource is dependent on its location and the location of the rest of the turbines. Hence, we have a scale parameter c_i for each turbine i : its computation is complex and involves wind velocity deficits $Vdef_j$ that the turbine i experiences due to the influence of other

Algorithm 1: Procedure for evaluation of wake effects due to the Park model [10]

```

1 Given  $\{X, Y\}$  as turbine locations, turbine thrust coefficient  $C_T$ ,
  rotor diameter  $R$ , landscape-specific wake spreading factor  $\kappa$ ;
2  $a = 1 - \sqrt{1 - C_T}$ ,  $b = \kappa/R$ ,  $u \leftarrow$  unit step function,
   $o = (x_i - x_j)\cos\theta + (y_i - y_j)\sin\theta$ ;
3  $d_{i,j} = \|o\|$ ,  $\alpha = \tan^{-1}\kappa$ ;
4 for  $i = 1$  to number of turbines do
5   for  $\theta = 0^0$  to  $360^0$  do
6     for  $j = 1$  to  $n-1$  and  $j \neq i$  do
7        $\delta_{i,j} =$ 
8          $\cos^{-1}\left(\frac{o+R/\kappa}{\sqrt{(x_i-x_j+\frac{R}{\kappa}\cos\theta)^2+(y_i-y_j+\frac{R}{\kappa}\sin\theta)^2}}\right)$ ;
9        $Vdef_{(i,j)} = u(\delta_{i,j} - \alpha) \frac{a}{(1+bd_{i,j})^2}$ ;
10       $Vdef_i^\theta = \sqrt{\sum_j (Vdef_{(i,j)}^\theta)^2}$ ;
11       $c_i(\theta) = c_i(\theta) \times (1 - Vdef_i)$ ;

```

turbines j , $j \neq i$ (see Algorithm 1). We refer the reader to [10] for a detailed presentation on the computation of this parameter when considering wake effects in the Park wake model. In short, the expected energy output η of the whole wind farm is given by

$$E^{farm}[\eta] = \sum_i \int_{\theta} P(\theta) \int_v p(v(\theta), c_i(\theta, X, Y), k(\theta)) \beta^i(v).$$

Here, v is the wind speed, and the function $\beta^i(v)$ defines the power curve for turbine i . Wind speed v however is a random variable with a Weibull distribution, $p(v(\theta), c_i(\theta, X, Y), k(\theta))$, which is estimated from wind resource data and considers the wake effect using X and Y . This distribution is also a function of the wind direction, θ which varies from $0^0 - 360^0$. Note that the *shape parameter* of the Weibull distribution is not influenced by the Wake effects here. In addition, wind flows from a certain direction with some probability $P(\theta)$.

2.1.2 Constraints and Assumptions

Following recent literature (see [10, 13–15]), we have the following constraints placed on our optimization function. The first one enforces an upper bound on the area of the farm. This constraint ensures that we can only place a turbine i within a certain area, which is a realistic constraint for most layout problems. For a rectangular farm with length l and width w this constraint is satisfied iff

$$0 \leq x_i \leq l \text{ and } 0 \leq y_i \leq w, 1 \leq i \leq n.$$

Because turbines can be damaged through turbulences when they are located too close to each other, the second constraint regulates the spatial proximity. It is satisfied iff

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq M \cdot R, 1 \leq i \neq j \leq n$$

where R is the rotor radius and M is a proximity factor usually decided ahead of the optimization based on the make and model of the turbines used. We use $M = 8$ based on the industry standard.

In addition to the above constraints, we assume that all turbines have the same power curves (approximated as piecewise linear functions) and that the same wind resource spans the entire farm.¹ Note that the assumptions can be revised in a very straight forward manner to generate more realistic scenarios.

¹To increase accuracy, these resources can be estimated for different parts in the farm.

2.2 Euclidean Minimum Spanning Tree

The Euclidean Minimum Spanning Tree is used to calculate the minimum length of cable required to connect all wind turbines in a specific wind farm configuration. It is computed by first constructing the complete graph on the set of points that represent the wind turbines and edge costs given by the Euclidean distance between any pair of turbines. The minimum spanning tree for this graph is computed and used as an objective representing the costs of the cable length. Figure 1 displays a wind turbine layout, as well as the minimum spanning tree, represented by the lines joining each wind turbine. All turbines are encompassed by a rectangular land area, represented by the dashed line.

2.3 (Cost of the) Convex Hull

For the purposes of our experiment, the cost of the convex hull is defined as the area contained by the set of points forming the convex hull. This value is the minimum land area that is required for a wind farm layout. We compute the convex hull using Graham's scan algorithm [8]. Figure 1 displays a wind turbine layout, as well as the area (cost) of the convex hull, shaded in grey. Provided that this shape is a non-intersecting polygon with n vertices, the area can be calculated as follows [1]:

$$A = \frac{1}{2} \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i).$$

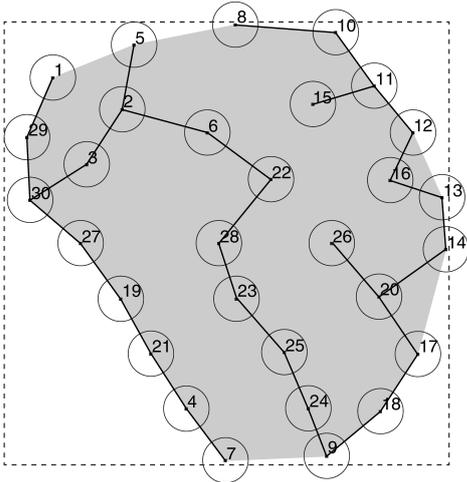


Figure 1: An example wind turbine layout. The shaded area represents the area of the convex hull. The lines signify the edges of the Euclidean Minimum Spanning Tree, or cables. The circles visualise the minimum safety distance imposed by the proximity constraint.

3. VARIATION OPERATORS FOR TURBINE PLACEMENT

In this section, we outline our framework and the problem-specific variation operators for the multi-objective wind turbine placement problem. As the problem is highly constrained due to a large number of turbines and the given safety margin around each turbine, our operators have to ensure that feasible placements are produced.

3.1 Movement Mutation

Mutation operators should do local changes to the current solution, i.e. the placement of the turbines on the given area of

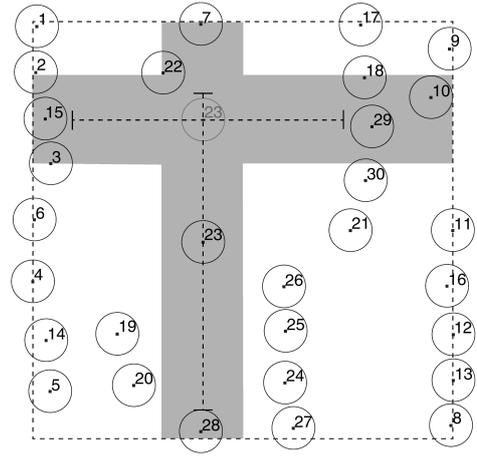


Figure 2: An instance of a Movement Mutation operation. The old and new positions of turbine 23 are shown, as well as the investigated area.

land. Our mutation operator called *Movement Mutation* (see Algorithm 2) moves a randomly selected specified proportion of the turbines in a given solution.

The distance and direction of the movement is determined randomly, subject to constraints.

In order to ensure that the solution remains feasible, the maximum distance that the wind turbine can move in each direction is computed. This is done by inspecting a designated area surrounding the x - and y -axis of the movement. In the inspection area, wind turbines whose rotor safety margins are found to overlap, restrict the constraints of movement for the selected turbine. The designated area is determined as two times the 38.5 metre margin of safety, emanating in each direction perpendicular from the axis of travel. In order to achieve a feasible placement after the movement of the turbines, the movement ranges M_v and M_h are computed based on the neighbouring turbines in the placement. By doing this it is guaranteed that a *Movement Mutation* operation does not produce an infeasible child individual.

Figure 2 depicts a single instance of a Movement Mutation operation. In this instance, a wind turbine (23) is randomly selected and moved directly south. The dotted lines represent the constraints of movement for the selected wind turbine. The shaded region represents the area that is inspected for overlapping wind turbines. The location of the turbine prior to the mutation is shown in grey. Note that the movements of turbine 23 are restricted by neighbouring turbines whose safety margins encroach upon the inspection area. The location of the turbine following the mutation is shown in black.

Provided that the algorithm of Movement Mutation (Algorithm 2) is based on a single loop of simple calculation, the time complexity of it can be determined as $O(n)$, in which n is the number of turbines in a solution.

3.2 Block Swap Crossover

Block Swap Crossover is designed to implant a randomly selected *block* from each of two parents to produce two children, each with a varying degree of information from each parent. A block is a rectangular area on the wind farm that is encompassed by the boundary of the entire wind farm. The first child uses the first parent as a basis. A block is randomly selected from the second parent for copying to the first child. Subsequently, the bounds of this block are expanded beyond the proximity constraint of every turbine by the value of twice the rotor radius (38.5 metres). This is to maintain

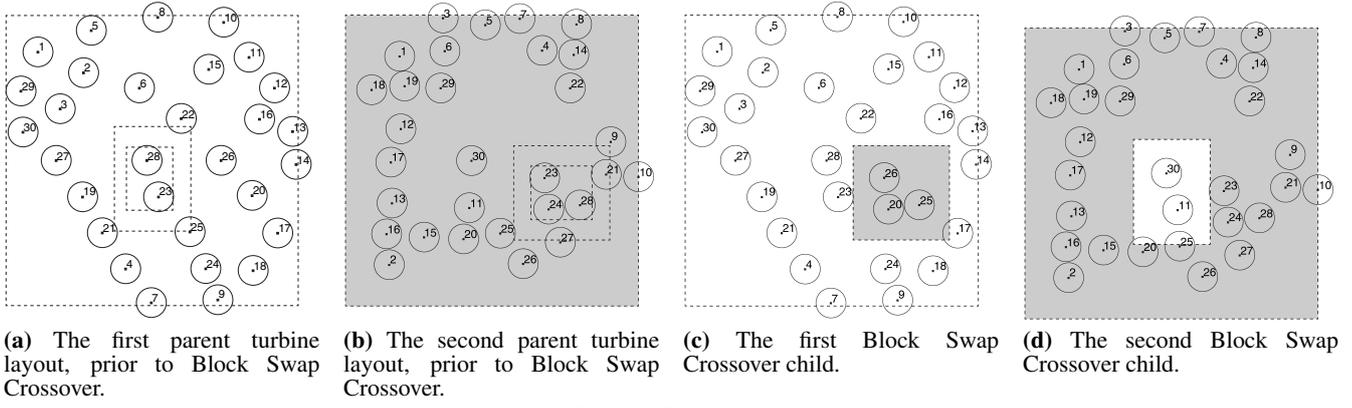


Figure 3: Example: Block Swap Crossover.

Algorithm 2: Movement Mutation

- 1 Given P as selected parent with turbine locations $\{X, Y\}$;
 - 2 Assuming turbine $t = \{x, y\}$ is selected for movement, $t \in P$, $x \in X$ and $y \in Y$;
 - 3 Let S = safety distance;
 - 4 Set M_h as horizontal movement range of t , M_v as vertical movement range of t ;
 - 5 **foreach** $t' = \{x', y'\} : (y - S < y' < y + S)$ **do**
 - 6 **if** x' is nearest to x on positive direction;
 - 7 **then**
 - 8 Save x' to x_p ;
 - 9 **if** x' is nearest to x on negative direction;
 - 10 **then**
 - 11 Save x' to x_n ;
 - 12 M_h ranges from $x_p - S$ to $x_n + S$;
 - 13 Redo the lines 5–12 to get of M_v ;
 - 14 **return** $t^n = \{x^n, y\} : (x^n \in M_h)$ or $t^n = \{x, y^n\} : (y^n \in M_v)$;
-

a margin of safety between the turbines in the copied block and the turbines outside these bounds. This boundary is then used in the child to determine the area that is replaced by the turbine information from the second parent. The above process is repeated for the second child, which uses the second parent as a basis, and copies information from the first parent.

The turbine position copying process is straightforward, provided that the destination area to be replaced in the child contains a lesser or equal number of turbines compared to the parent. In the event that this is not the case, a new boundary is randomly selected until this condition is satisfied. If in the event that a suitable block is not found for copying to create a child, its respective parent is returned as a result of the operation. This indicates that the crossover operation has failed. The turbines within the destination area are rearranged to the same configuration as the source area in the parent. If the number of turbines in the destination area matches the source, the operation ends. Otherwise, turbines are randomly selected from outside the child’s destination area and moved into the destination area as substitutes. This ensures that the number of turbines is equivalent in both the source and destination.

Figures 3a and 3b show the configuration of the first and second parents, as well as the source area selected for copying. In Figures 3c and 3d, each child is shown, following the copying operation. The shaded areas represent information that is obtained from the second parent. White areas represent information from the first parent.

Algorithm 3: Block Swap Crossover

- 1 Given P_1 and P_2 are selected parents with turbine locations;
 - 2 **for** $i = 0$ to T **do**
 - 3 Randomly generate a rectangular area A_1 in wind farm area;
 - 4 The turbines in P_1 are divided into two subsets, I_1 for the turbines inside A_1 and O_1 for the ones outside of A_1 ;
 - 5 Modify A_1 to A'_1 in which each edge keeps a safety distance to nearest turbine in I_1 ;
 - 6 Apply A'_1 on P_2 to get I_2 and O_2 ;
 - 7 **if** $|I_2| \geq |I_1|$ **then**
 - 8 Move $n = |I_2| - |I_1|$ turbines from O_1 into I_1 ;
 - 9 Set turbine locations in I_1 as in I_2 ;
 - 10 **Return** P_1 as child number 1;
 - 11 **else**
 - 12 continue;
 - 13 Apply the above on P_2 to get child number 2;
-

The time complexity of the crossover operator is similar to the mutation operator. As it is also using a single loop (Algorithm 3), the complexity can be determined as $O(n)$.

4. EXPERIMENTAL INVESTIGATIONS

In this section, we describe our experimental setup and report on the results of different popular evolutionary multi-objective algorithms using our new variation operators for the multi-objective wind turbine placement problem.

4.1 Experimental Setup

We are solving the multi-objective wind turbine placement problems using popular multi-objective evolutionary algorithms together with the variation operator described in the previous section. For the basis of our experiments, the evolutionary multi-objective algorithms IBEA [19], NSGA-II [3], and SPEA2 [20] and their implementation in the jMetal framework [4] are utilised. The *BinaryTournament2* selection operator of NSGA-II is used for all experiments. In contrast to a pure binary tournament, this selection operator checks whether one solution dominates the other, before comparing the corresponding fitness values.

To evaluate our algorithm’s performance we used wind *Scenario 2* defined in [10]. *Scenario 2* is rather complex, and the prevailing wind direction covers a broad sector of about 105° . The wind intensity per direction is given by Weibull distributions in both cases, which results in non-zero probabilities for wind coming from any direction. Therefore, one has to optimize the layout to work with minimum wake loss along all the wind directions.

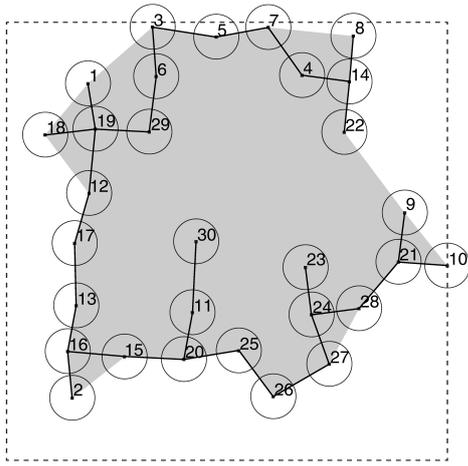


Figure 4: A *packed* 30 turbine configuration from an NSGA-II triple-objective experiment. It yields a total power output of 2.03×10^5 kW, with a convex hull area of 5.47 km^2 and a minimum spanning tree length of 10.7 km .

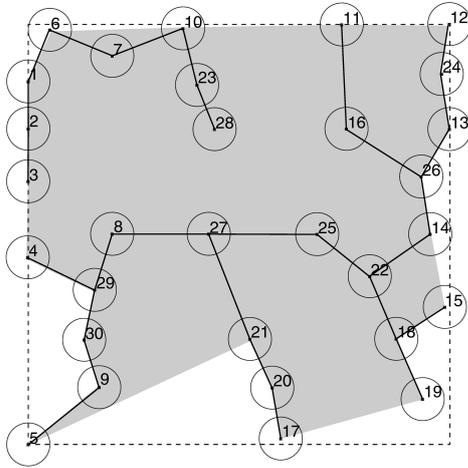


Figure 5: A *distributed* 30 turbine configuration from an NSGA-II triple-objective experiment. It yields a total power output of 2.31×10^5 kW, with a convex hull area of 8.55 km^2 and minimum spanning tree length of 14.1 km .

Lastly, we use the turbine rotor radius $R = 38.5 \text{ m}$ for the proximity constraints.

Based on the wind scenario, we can compute the expected yield that we then use as one objective in our optimisation. For the other objectives, the Euclidean minimum spanning tree and the convex hull area, more details are available in Sections 2.2 and 2.3 respectively.

In the experiments, we applied the multi-objective optimisation algorithms NSGA-II, SPEA2 and IBEA using our operators. In each simulation instance, a population size of 50 is utilised, over 20,000 generations, and its runtime is less than 15 minutes. The Movement mutation operator is applied with a probability of $p_m = 0.7$ in each iteration and it moves in each mutation step 5% of the turbines. The Block Swap crossover operator is applied with a probability of $p_c = 0.3$.² These parameter values are used for all

²Both values were chosen based on initial testing.

NSGAI - 30 turbines, Scenario 2

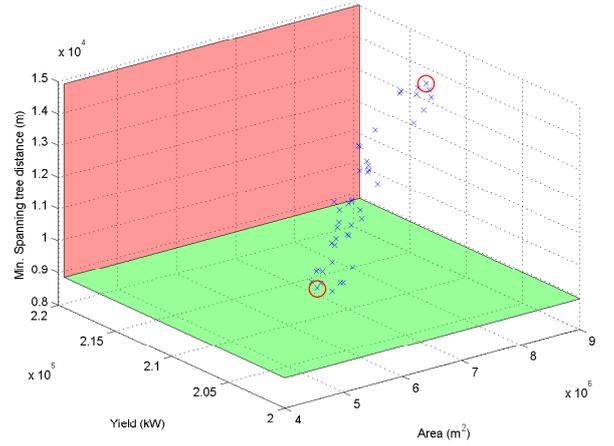


Figure 6: The locations of the packed and distributed turbine configurations on the Pareto frontier.

three multi-objective algorithms, and for the following objective combinations:

1. maximising the power output (yield) and minimising the area of the convex hull,
2. maximising the power output (yield) and minimising the total distance of the minimum spanning tree, and
3. a combination of power, convex hull area and minimum spanning tree.

Each variation on the algorithm and objective combinations is repeated 10 times in order to build a point cloud. The land area is restricted to $3 \times 3 \text{ km}^2$ in the experiments involving 30, 50, and 70 turbines. In the case of the 200 turbines, the land is restricted to $10 \times 10 \text{ km}^2$. As a consequence of this fixed area, the set-ups have different qualities. For example, the algorithms can pack the 30 turbines tightly as well as distribute them over a large area, whereas even the tightest 70 turbine layouts will cover a significant portion of the available land area already. The scenario involving 200 turbines is computationally expensive due to the increasing costs of the wake computations. It allows us to compare our approach with the single-objective algorithms listed in [15] and the multi-objective approach outlined in [18] respectively.

4.2 Experimental Results

Figure 6 displays a 3D point cloud that represents the Pareto frontier over 10 runs of an experiment using the NSGA-II algorithm. The total footprint of the wind farm is shown in the x-axis. Yield, or power output of the wind farm is shown in the y-axis. The z-axis represents the total distance of the Euclidean minimum spanning tree formed by all wind turbines. The surfaces shown in the plot represent the feasible bounds of the experiment. The red surface represents the maximum feasible yield of a 30 turbine wind farm, without wake loss. This theoretical maximum is calculated using formula 22 in [10]. The green surface represents the minimum feasible value of the minimum spanning tree, taking into account the turbine rotor radius proximity constraints. This value is calculated as follows:

$$L = (n - 1) \times 8.0 \times 38.5$$

where n is the number of turbines in the wind farm.

The two extremes of the Pareto frontier are highlighted in Figure 6. The lower left focuses on minimising the footprint in the packed configuration. The upper right is an optimisation for maximum power.

Figures 4 and 5 draw comparisons between a “packed” turbine layout and a “distributed” turbine layout. In the former layout, we try to deliver a layout that yields the most power, while at the same time providing a compact footprint and reducing the length of cabling required to connect each wind turbine. In Figure 5, an attempt is made to make best use of the 3×3 km land area available for use. In this instance, we are simply concerned with optimising the yield without regard to cable length or footprint.

We summarise all experiments on the 200 turbine scenario in Figure 7. It is observed that IBEA has a tendency to select solutions with a minimal spanning tree distance and convex hull area. This is in contrast with NSGA-II and SPEA2 that both achieve an apparently greater spread. This is likely due to IBEA’s lack of diversity preservation techniques [19]. The greater spread of NSGA-II and SPEA2 may be considered computationally wasteful, as a wind farm builder may generally prefer smaller EMSTs and smaller convex hulls. This is covered adequately by IBEA, as clearly shown in the Yield versus Convex hull area and Yield versus Minimum spanning tree distance plots. In addition, IBEA’s layouts for small EMSTs and smaller convex hulls have a significantly higher energy yield. Therefore, due to its bias, IBEA seems to be the most adequate algorithm for the multi-objective optimisation of wind farm layout, given our scenarios.

It is interesting to see how our approach compares to existing ones. Firstly, we achieve virtually the same maximum energy yield as the best-performing single-objective optimisation algorithm listed in [15]. Secondly, we achieve maximum energy yields that are about 10% better than the results achieved by the approach outlined in [18]. We see both as strong indicators that our approach with the problem-specific variation operators enable the fast and efficient multi-objective optimisation of wind farm layouts.

To compare the performances of the different multi-objective algorithms, we use the hypervolume indicator. Before we compute the hypervolumes, we normalize objective values to the following ranges:

1. 30 turbines: area $5\text{km}^2 - 9\text{km}^2$, yield $200,000\text{kW} - 215,000\text{kW}$, spanning tree length $8,5\text{km} - 16\text{km}$
2. 50 turbines: area $7\text{km}^2 - 9\text{km}^2$, yield $320,000\text{kW} - 340,000\text{kW}$, spanning tree length $15\text{km} - 21\text{km}$
3. 70 turbines: area $8\text{km}^2 - 9\text{km}^2$, yield $430,000\text{kW} - 460,000\text{kW}$, spanning tree length $21\text{km} - 25\text{km}$
4. 200 turbines: area $10\text{km}^2 - 100\text{km}^2$, yield $1,000,000\text{kW} - 1,440,000\text{kW}$, spanning tree length $60\text{km} - 150\text{km}$

The results are listed in Table 1. Interestingly, IBEA’s performance varies significantly for 30 and 50 turbines. However, it clearly outperforms the other two algorithms for 70 turbines. On 200 turbines, IBEA achieves the highest hypervolume values, and with very little performance variation.

5. CONCLUSIONS

Wind energy plays an increasing role in the energy supply worldwide. We have investigated the problem of placing wind turbines on a given area of land using different conflicting objective functions. We designed new variation operators for the wind turbine

n Turbines	IBEA	NSGA-II	SPEA2
30	0.1382 _{0.126}	0.2230 _{0.018}	0.2654 _{0.024}
50	0.1179 _{0.116}	0.2079 _{0.018}	0.2358 _{0.018}
70	0.3421 _{0.075}	0.2119 _{0.023}	0.2403 _{0.021}
200	0.5380 _{0.001}	0.4060 _{0.012}	0.3589 _{0.020}

Table 1: Comparison of mean 3D hypervolume indicator values. The corresponding standard deviations are listed in subscript, and the best performing algorithm (according to the hypervolume value) is highlighted in boldface.

placement problem and have shown that state-of-the-art evolutionary multi-objective algorithms are very effective for the multi-objective wind turbine placement problem when using these operators. The quality of our results is confirmed by the lower bounds given for the cost of a minimum spanning tree and the maximum achievable energy which is very close to the one obtained in the single-objective approach of [15].

It is interesting to see how far the optimiser can improve upon the initial grid-based layouts, as the energy yield increases by about 6% and the EMST length reduces by about 40%. These improvements can translate into significant capital expenditure savings when setting up the wind farm, and into millions of additional dollars of income per year during the lifetime of the wind farm.

References

- [1] P. Bourke. Calculating the area and centroid of a polygon, 1988.
- [2] S. Şişbot, O. Turgut, M. Tunç, and U. Çamdali. Optimal positioning of wind turbines on Gökçeada using multi-objective genetic algorithm. *Wind Energy*, 13(4):297–306, 2010.
- [3] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolutionary Computation*, 6(2):182–197, 2002.
- [4] J. J. Durillo, A. J. Nebro, and E. Alba. The jMetal framework for multi-objective optimization: Design and architecture. In *Proc. Congress on Evolutionary Computation (CEC ’10)*, pages 4138–4325. IEEE Press, 2010.
- [5] A. Eiben and J. Smith. *Introduction to Evolutionary Computing*. Springer, 2007.
- [6] Global Wind Energy Council. Global Statistics – Market Forecast 2012–2016, 2012. <http://tinyurl.com/gwec12>.
- [7] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. MIT Press, Cambridge, MA, USA, 1989.
- [8] R. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information processing letters*, 1(4):132–133, 1972.
- [9] J. Kennedy, R. C. Eberhart, and Y. Shi. *Swarm Intelligence*. Morgan Kaufman, 2001.
- [10] A. Kusiak and Z. Song. Design of wind farm layout for maximum wind energy capture. *Renewable Energy*, 35(3):685–694, 2010.
- [11] W. Y. Kwong, P. Y. Zhang, D. Romero, J. Moran, M. Morgenroth, and C. Amon. Wind farm layout optimisation considering energy generation and noise propagation. In *Proc. International Design Engineering Technical Conference & Computers and Information in Engineering Conference (IDETC/CIE ’12)*, pages 1–10, 2012.
- [12] P.-E. Rethore. State of the art in wind farm layout optimization. *Wind Energy Research*, page 179, 2010.
- [13] K. Veeramachaneni, M. Wagner, U.-M. O’Reilly, and F. Neumann. Optimizing energy output and layout costs for large

- wind farms using particle swarm optimization. In *Proc. Congress on Evolutionary Computation (CEC '12)*, pages 1–7, 2012.
- [14] M. Wagner, K. Veeramachaneni, F. Neumann, and U.-M. O'Reilly. Optimizing the layout of 1000 wind turbines. In *European Wind Energy Association Annual Event*, 2011.
- [15] M. Wagner, J. Day, and F. Neumann. A fast and effective local search algorithm for optimizing the placement of wind turbines. *Renewable Energy*, 51(0):64–70, 2013.
- [16] C. Wan, J. Wang, G. Yang, X. Li, and X. Zhang. Optimal micro-siting of wind turbines by genetic algorithms based on improved wind and turbine models. In *Conference on Decision and Control*, pages 5092–5096, 2009.
- [17] C. Wan, J. Wang, G. Yang, X. Li, and X. Zhang. Optimal siting of wind turbines using real coded genetic algorithms. *European Wind Energy Association Conference and Exhibition*, 2009.
- [18] C. Wan, J. Wang, G. Yang, and X. Zhang. Optimal micro-siting of wind farms by particle swarm optimization. In *Advances in Swarm Intelligence*, volume 6145 of *Lecture Notes in Computer Science*, pages 198–205. Springer, 2010.
- [19] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *Proc. 8th International Conference Parallel Problem Solving from Nature (PPSN VIII)*, volume 3242 of *LNCS*, pages 832–842. Springer, 2004.
- [20] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *Proc. Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN '01)*, pages 95–100, 2002.

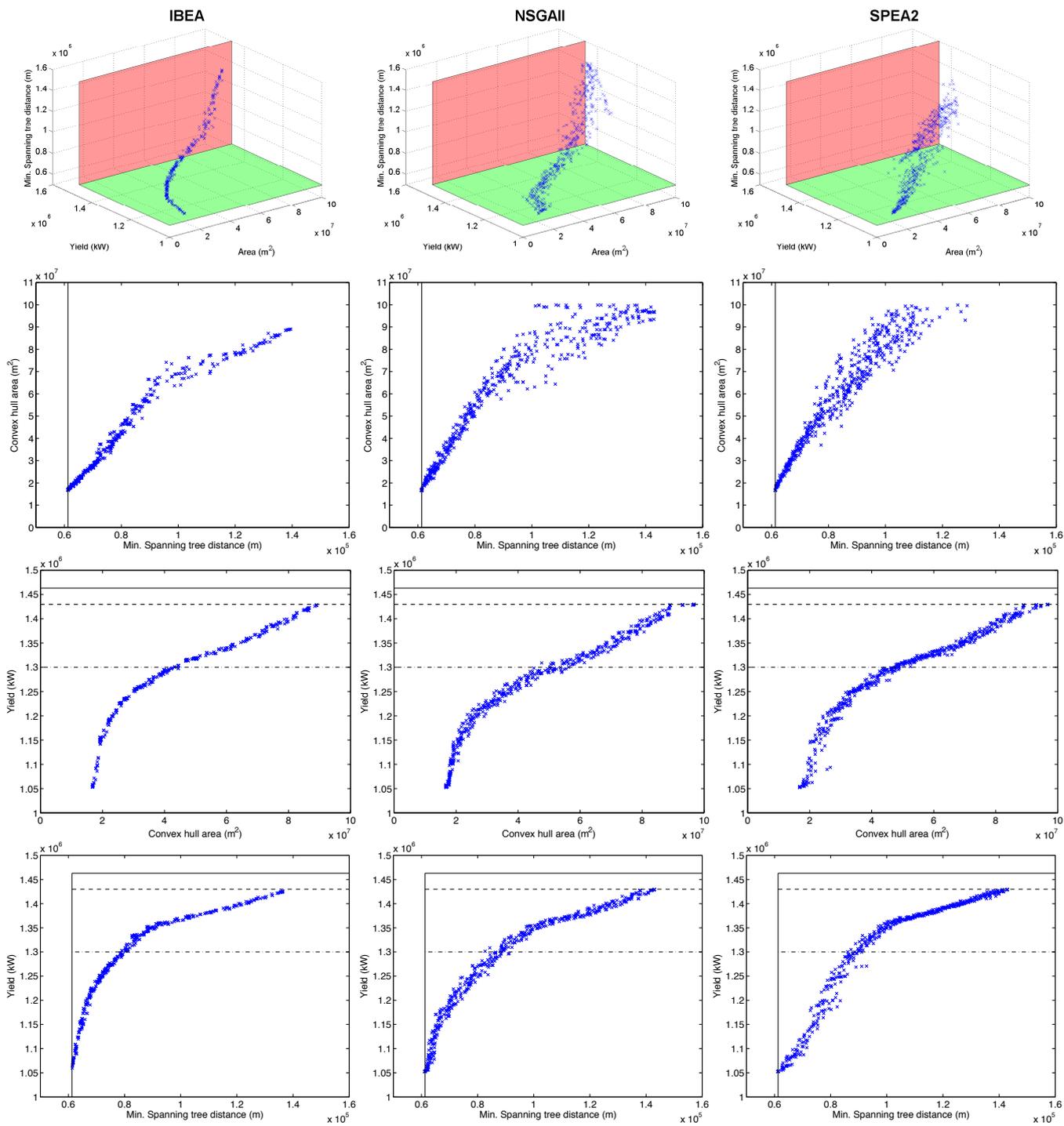


Figure 7: Point clouds depicting the Pareto frontiers for 10 runs - Scenario 2, 200 turbines. Where applicable, we show the theoretical limits (red=maximum yield, green=minimum spanning tree length) as coloured surfaces or as solid lines. The dashed lines show a yield of $1.43 \cdot 10^6$ kW and $1.30 \cdot 10^6$ kW. These are the averages achieved by the best-performing single-objective optimisation algorithm listed in [15] and the result achieved by the approach outlined in [18] respectively.