Application of selection hyper-heuristics to the simultaneous optimisation of turbines and cabling within an offshore windfarm

Butterwick, T.; Kheiri, A.; Lulli, G.; Gromicho, J.; Kreeft, J.

DOI
10.1016/j.renene.2023.03.075

Publication date
2023

Document Version
Final published version

Published in
Renewable Energy

Renewable Energy 208 (2023) 1–16

Contents lists available at ScienceDirect

Renewable Energy

journal homepage: www.elsevier.com/locate/renene

Application of selection hyper-heuristics to the simultaneous optimisation of turbines and cabling within an offshore windfarm

Thomas Butterwick [a], Ahmed Kheiri [a,*], Guglielmo Lulli [a], Joaquim Gromicho [b,d], Jasper Kreeft [c]

[a] Lancaster University, Department of Management Science, Lancaster, LA1 4YX, UK
[b] ORTEC, Houtsingel 5, 2719 EA Zoetermeer, The Netherlands
[c] Shell Global Solutions International BV, Carel van Bylandtlaan 30, 2596HR The Hague, The Netherlands
[d] Amsterdam Business School, University of Amsterdam, Plantage Muidergracht 12, 1018 TV Amsterdam, The Netherlands

ARTICLE INFO

Keywords:
Windfarm
Optimisation
Metaheuristics
Hyper-heuristic

ABSTRACT

Global warming has focused attention on how the world produces the energy required to power the planet. It has driven a major need to move away from using fossil fuels for energy production toward cleaner and more sustainable methods of producing renewable energy. The development of offshore windfarms, which harness the power of the wind, is seen as a viable approach to creating renewable energy but they can be difficult to design efficiently. The complexity of their design can benefit significantly from the use of computational optimisation. The windfarm optimisation problem typically consists of two smaller optimisation problems: turbine placement and cable routing, which are generally solved separately. This paper aims to utilise selection hyper-heuristics to optimise both turbine placement and cable routing simultaneously within one optimisation problem. This paper identifies and confirms the feasibility of using selection hyper-heuristics within windfarm optimisation to consider both cabling and turbine positioning within the same single optimisation problem. Key results could not identify a conclusive advantage to combining this into one optimisation problem as opposed to considering both as two sequential optimisation problems.

1. Introduction

Globally there is a need to move away from fossil fuel and carbon-producing energy sources toward cleaner and more sustainable methods of powering the world. This has led to the increased construction of large-scale offshore windfarms, which utilise the faster wind speeds found at sea, for greater and cleaner energy production. However, renewables are typically more expensive than their carbon dioxide emission generating counterparts, and this can create a barrier to investment within the industry. Increasing the energy produced whilst minimising the cost of production is key to reducing entry costs and attracting further investment into renewable energy. The creation of windfarms can present significant design challenges to ensure maximised production whilst minimising the cost of the farm. The development of a windfarm requires the consideration of several sub-problems and the need for them to be addressed. These include the interference impact of other turbines, turbine placement taking account of expected wind speeds, inter-turbine cabling and the connection to an external grid or substation. Each of these areas are of fundamental importance. Increased power output or better optimisation of cabling can result in a significant reduction to potential lifetime costs which, could have the ability to make renewable energy more competitive than traditional fossil fuel sources.

The majority of work to date in the area of offshore windfarm design has been divided into two steps:

(1) Turbine placements subject to maximisation of power production and consideration of other constraints such as interference between turbines and minimum separation distances.
(2) Once turbine positions are determined, the cabling layout between turbines is optimised with the goal of minimising costs and power loss subject to constraints such as cable capacity and layout constraints.

Development of these two steps has largely been covered using mixed-integer linear programs with the inclusion of heuristics in some areas such as a matheuristic [1] and hyper-heuristics [2]. Traditionally, the reduction in cabling cost is limited by the static positioning of the wind turbines from step (1). However, combining both steps (1) and (2) could yield lower cabling costs within the windfarm whilst also considering other objectives such as maximising power production.

* Corresponding author.
E-mail addresses: tombutterwick21@gmail.com (T. Butterwick), a.kheiri@lancaster.ac.uk (A. Kheiri), g.lulli@lancaster.ac.uk (G. Lulli), joaquim.gromicho@ortec.com (J. Gromicho), jasper.kreeft@shell.com (J. Kreeft).

https://doi.org/10.1016/j.renene.2023.03.075
Received 3 July 2022; Received in revised form 11 January 2023; Accepted 16 March 2023
Available online 17 March 2023

Cabling can account for around 4%–5% of the total capital expenditure for a typical windfarm construction [3]; and it can be as high as 18% for offshore windfarms [4]. Therefore, any potential reduction in this cost could be significant. This opportunity lends itself toward the use of a hyper-heuristic approach which would allow for a range of single low-level heuristics (LLH) to implement adjustments to a windfarm's turbine positions and cabling layout.

One of the main aims of this work is to investigate if the turbine placement and cable routing optimisation problem can be combined and whether such an approach is more beneficial than solving them sequentially (turbine placement followed by cable routing). To achieve this, several optimisation algorithms were developed using selection hyper-heuristics combined with various solution acceptance criteria (move acceptance) and applied to both a sequential model and a simultaneous model. In this paper, these models are referred to as either sequential (one after the other) or simultaneous (solving both at the same time).

The paper is structured as follows: Section 2 examines previous literature and research within the area of windfarm optimisation and selection hyper-heuristics. Section 3 defines the windfarm problem and gives a mathematical formulation alongside visual examples. Section 4 presents the methodology used. Section 5 presents the results when applied to real-world windfarm instances and provides computational results alongside discussion. Finally, Section 6 concludes against the overall aims and objectives of this study and provides recommendations for future work.

2. Related work

The considerable potential to increase output whilst reducing cost is reflected in the wide body of research addressing the optimisation of a windfarm's layout. Much of the research focuses on one aspect of a windfarm, either a turbine layout or inter-array cable routing; and very limited attention has been posed on the combination of these two aspects. Wu et al. [5] and Hou et al. [6] developed metaheuristic approaches to solve the combined problem. More in particular, Wu et al. [5] combined a genetic algorithm for the placement of wind turbines with an – inner – ant colony optimisation routine to assess the associated "optimal" cabling costs. Hou et al. [6] developed a particle swarm optimisation approach to solve the combined problem. Most of the research has considered both key components in sequence with turbine placement occurring first followed by cable routing [7]. Research differs in terms of the constraints considered (such as sound, wake or terrain) and objectives desired (cost, profit, power or efficiency). Some additional work has explored areas such as substation placement [8] and the use of machine learning to train a model for the faster computational examination of potential siting locations [1].

Mixed-integer linear programming (MILP) is a popular method for deriving an optimal turbine or cabling layout. Fagerfjäll [8] applied MILP to optimise an onshore windfarm. Two models were developed, a production model (for turbine placement) and an infrastructure model (for cabling) which would be implemented upon the production model's result. This linear programme aimed to maximise production and revenues from the windfarm. Within the production model, constraints on the MILP included minimum separation distances and consideration of the production loss between turbines due to the wake effect. These models were contrasted to commercial heuristics-based optimisation software and showed the potential to yield significantly higher production values (40% or so higher). The infrastructure model for inter-array cabling introduced Steiner nodes within the spanning trees, allowing for shorter cable pathways when multiple turbines were nearby. However, because the two models were not combined, there is limited scope for providing a true optimal windfarm layout. Furthermore, only two types of cables were considered within the inter-array cabling, whereas in real-world scenarios several types exist and are in use. Similarly, a MILP was implemented by Fischetti and Pisinger [9] to the cabling aspect in order to determine an optimal cable path between turbines. An initial solution was developed and applied using a commercial MILP solver and thereafter a matheuristic scheme was applied iteratively to develop the solution. Fischetti and Pisinger [9] found that using both these methods combined typically outperformed the use of a separate heuristic or MILP approach. But, the performance of the subsequent heuristics depended heavily on the initial MILP and what might work for larger projects was not always applicable to smaller windfarms with fewer turbines to place. A more unique MILP model was proposed by Donovan [10]; this required a minimum productivity requirement (MPR) for any potential turbine location. The MPR identified the required power production to justify investment into a turbine and ensure initial costs were paid back within the specified required payback period. Including this constraint within the model ensures that a windfarm can be profitable, however, overall production may be sacrificed in the pursuit of a minimum cost layout.

Saavedra-Moreno et al. [11] used an evolutionary algorithm to optimise the positioning of turbines based on factors such as orography, wind conditions, obstacles and cost of installation. Cazzaro et al. [3] also adopted a heuristic approach, but, they concentrated upon the cable routing problem. With a focus on fast heuristics that can scale well, various metaheuristics were used, including sweep multi-start, simulated annealing, tabu search, variable neighbourhood search, ant algorithm and genetic algorithm. These were applied to both test and training instances with tabu search and variable neighbourhood search reaching near-optimal values within the test set. Metaheuristics have also been applied to a floating offshore windfarm by Lerch et al. [12]. They adapted a particle swarm optimisation model to develop the inter-array cable layout subject to minimisation of the following costs: acquisition, installation and energy loss costs. Additional constraints included reliability assessments for electrical components insofar as floating windfarms have increased complexity with cables undergoing high mechanical load due to sea conditions. The model successfully avoided cable crossing and also produced shorter cable distances and costs compared to the reference model used. Bauer and Lysgaard [13] noted that the cabling routing decision is the same as a vehicle routing problem and thus built a heuristic algorithm for cable layouts based on the Clarke and Wright savings heuristic for vehicle routing. A planar open savings heuristic was developed which considered merging two routes into one and at each iteration chose to merge with the greatest saving subject to capacity constraints. This was compared to a hop-indexed integer programming formulation and found the heuristic approach was within 2% of the optimal layout. However, the research only focused on a maximum of two cable types which, whilst representative of the real-world sites used within the paper, may have limited wider application. The reader is directed to Wilson et al. [14] for more heuristic techniques applied to windfarm layout optimisation problems.

This paper focuses on utilising the latest developments in selection hyper-heuristics that focus on turbine positioning and cable layouts at the same time within one optimisation problem as opposed to one after the other. Cowling et al. [15] defined hyper-heuristics as 'heuristics to choose heuristics' and used a range of selection hyper-heuristics to schedule a sales summit. Selection hyper-heuristics consist of two key elements, selection method (SM) and move acceptance (MA) [16]. A move acceptance defines if a solution is accepted or not and these methods are either stochastic if there is a probability of accepting, or otherwise deterministic by nature. An example of a MA is 'improve or equal' whereby if a new solution's objective value is equal to or better than the current best, it is accepted and becomes the new best solution. Selection methods then aim to diversify the range of solutions searched by choosing the optimal low-level heuristic based on set criteria or methodology [17]. Li et al. [2] pursued a multi-objective approach utilising nine selection hyper-heuristics to control a set of low-level metaheuristics. These metaheuristics consisted of three multi-objective evolutionary algorithms. A variety of move acceptance methods were also considered including only-improve, great deluge and all-acceptance. Findings showed that selection hyper-heuristics could exploit the use of multi-objective metaheuristics and provide statistically significant performance compared to single objective use. Further work, however, would need to include a greater number of move acceptance methods and the application to other components of windfarm design such as inter-array cable routing.

The literature reviewed shows a significant and well-recognised gap in the optimisation of an optimal windfarm design. Separation of the main two stages (1) turbine placement and (2) cabling layout design can result in a missed opportunity to consider the potential cable costs alongside the turbine costs for a new position. Cabling between turbines (inter-array) and a substation or external grid can be a significant cost factor within any offshore windfarm; there may be benefits to it being considered alongside the placement of wind turbines as suggested by Cazzaro et al. [3]. The main methodology used within previous research is in the application of mixed-integer linear programming and heuristics with only a small amount of work considering the role of selection hyper-heuristics. This area is the focus of this paper methodology.

3. Problem description

Offshore windfarm design is a complex and challenging optimisation problem, with a large number of possible layouts and varying objectives. Several areas of design need to be addressed including turbine placement, cable routing and substation placement. This paper focuses on two areas of the design phase: turbine placement and cable routing. Previous research shows the process of optimising these two areas has typically been done sequentially, in the order of turbine placement and then cable routing second.

- **Turbine Placement Optimisation Problem:** The placement of turbines aims to determine a feasible selection of locations from which the power production of the windfarm is maximised subject to various constraints. A considerable impact upon potential production is the wake effect between turbines. As wind flows through a turbine, the kinetic energy of the wind is disrupted and results in a slower wind speed, reducing power production for any turbines downstream. Reduction of the wake effect is therefore of extreme importance and must be considered within any optimisation model. In addition to the consideration of the wake effect, there must exist a minimum separation distance to avoid turbine blades colliding with each other. Limits upon the number of turbines to locate should be specified in advance of any optimisation model.
- **Cable Routing Optimisation Problem:** Within an offshore windfarm, the power produced by each turbine must be transferred back to a substation located near to the farm; from which a high-capacity export cable transmits the power to the main electrical grid. Optimisation of this problem aims to find a feasible power routing between turbines and the substation. An example of how a typical offshore windfarm is connected is shown in Fig. 1. Cabling between turbines is called inter-array cabling and is typically low voltage cabling with some resistivity. These cables are connected to the base of each turbine (not the seabed) and then 'hang' down before laying on the seabed floor. Turbines can either be connected to each other or directly connected to the substation. Once arriving at the substation this power is exported to the grid. Key requirements of this optimisation problem involve the correct selection of cable type, minimisation of power losses due to resistivity in cables and minimisation of the cost of cabling. Offshore cabling can be an expensive component of a windfarm accounting for around 4%–5% of the total cost [3]. As power is transmitted through cabling, a certain amount is lost based upon the resistance of each cable; this varies dependent on the cable type with the tendency for more expensive cables to have lower resistance. Therefore, a trade-off can exist between choosing more efficient cabling (benefitting in the long-term) and reducing the cost of those cables.

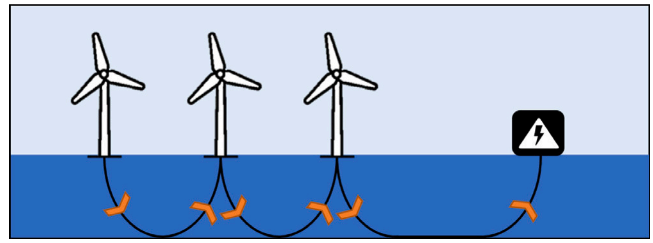


Fig. 1. Example offshore windfarm cable layout, orange arrows indicate the direction of power flow toward the substation.

Given the complex design challenges and a considerable number of factors involved in windfarm design, the problem is simplified within this paper. To reduce the potential turbine positions, a grid of pre-defined locations is used, also allowing for the minimum separation distance to be incorporated between each grid point. Further assumptions are made below:

- A1 Only one turbine may occupy any given position with the grid.
- A2 Cabling is only in straight lines and between turbine positions or the substation (in the real world cabling can curve but this adds a large amount of complexity).
- A3 Only one cable may traverse between each pair of turbines.
- A4 The substations' position is already known and cannot move.
- A5 Only one type of turbine is being placed with a rating of 9.5 MW.
- A6 Without any turbines, the expected production at each spot on the grid is the same, i.e., wind speed is equal everywhere.
- A7 There are no differences in foundation costs and therefore these are not considered.

These assumptions allow for easier development and evaluation of optimisation models whilst still considering major conditions such as wake effect and cabling factors.

The problem can be defined mathematically as follows: A vector, T of size n where n is the number of potential turbine positions represents whether or not a position on the grid has a turbine occupying it (1 = occupied, 0 = empty). Four matrices are defined to signify (1) cabling, (2) distances, (3) power loads, and (4) cabling costs.

(1) Cabling matrix represents the cabling between each subsequent siting option ($c_{i,j}$) and substation where: c is either 1 if a cable exists or 0 if no cable exists between position number i and j. n represents the number of sites + 1, with the additional site representing cabling to the substation.

$$Cabling = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \end{pmatrix}$$

An example potential grid of potential turbine locations and cabling is shown in Fig. 2. Shown is a grid of 16 potential positions with 5 selected and the substation shown in the bottom left, alongside the cabling layout with power flow indicated by the arrow direction. Within this example there is one power flow route to the substation defined as (9, 15, 8) → (6) → (1) → (sub). Where the cumulative net power is summed at each flow point (6), (1) and (sub).

(2) Distance matrix represents the distance between each potential position ($d_{i,j}$) with d representing the distance between site number i and j. n represents the number of sites + 1, with the additional site representing cabling to the substation.

$$Dist = \begin{pmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,n} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n,1} & d_{n,2} & \cdots & d_{n,n} \end{pmatrix}$$

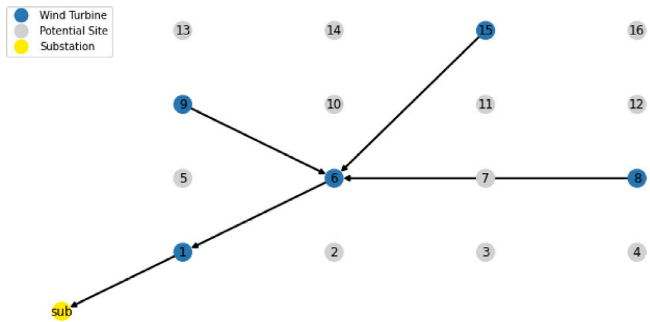


Fig. 2. Example grid showing potential turbine sites and substation position alongside power flow within the windfarm.

(3) Net Power matrix accounts for the net power sent between each position ($p_{i,j}$) with p representing the net power transferred from site number i and j. This is the net power after losses due to wake and cabling have been considered. n represents the number of sites + 1, with the additional site representing cabling to the substation. The sum of the nth column, therefore, shows the total net power flow into the substation from all turbines.

$$NP = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{pmatrix}$$

Net power flow between two points is defined as the initial power flow minus power losses due to resistivity in the cable. This varies dependent upon the cable cross-section and material used such as copper or aluminium. To calculate the power capacity, in MW for a given cable:

$$P = \frac{I \times V}{1000} \tag{1}$$

where: I is the rated cable current in Amps; V is the cable voltage in kV; and P is the power capacity, in MW for the cable. The expected power loss (in MW) for each cable over a set distance is therefore equal to:

$$PL_{i,j} = \frac{I^2 \times R \times D}{1 \times 10^9} \tag{2}$$

where: R is the resistance (ohm/km) within the cable; D is the distance travelled in km, from point i to j is equal to $Dist_{i,j}$; and $PL_{i,j}$ is the power loss between points i and j.

(4) The cost of cabling is represented below with cc indicating the individual costs from each siting position i and j ($cc_{i,j}$). The cabling cost between two points is defined by choosing an appropriate cable based upon the power load expected and identifying the cost per unit of distance and multiplying by the distance travelled.

$$CC = \begin{pmatrix} cc_{1,1} & cc_{1,2} & \cdots & cc_{1,n} \\ cc_{2,1} & cc_{2,2} & \cdots & cc_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ cc_{n,1} & cc_{n,2} & \cdots & cc_{n,n} \end{pmatrix}$$

3.1. Windfarm objective function

The key objectives of windfarm optimisation are to maximise power production within the farm whilst minimising the overall cost. As described previously, the power produced within an offshore windfarm is mainly impacted by the wake effect and power losses through the cabling layout. Minimisation of cost is highly dependent upon the number of wind turbines placed, the rated output of these turbines and the positioning and choice of cabling used between turbines and back to the substation. Saavedra-Moreno et al. [11] utilised similar objectives within their cabling optimisation by creating a cost function equal to $\frac{cabling\_costs}{net\_power}$. Marmidis et al. [18] optimised purely turbine layouts and proposed an equation in the same fashion to be $\frac{turbine\_costs}{net\_power}$. The objective function within this paper is therefore a combination of both, resulting in: $\frac{cabling\_costs+turbine\_costs}{net\_power}$. This equation gives a 'ratio' of the cost per unit of net power allowing for easier comparison between smaller and larger windfarm instances.

Defining this mathematically based upon the introduced matrices and previous equations gives:

$$obj = \frac{\sum_{i,j} CC_{i,j} + \sum_{i=1}^{n-1} S_i T_c}{\sum_{i,n} NP_{i,n}} + \alpha \tag{3}$$

where: $\sum_{i,j} CC_{i,j}$ equals total cabling costs; $\sum_{i=1}^{n-1} S_i T_c$ equals total turbine costs with T_c representing the cost per turbine and $n-1$ is the total number of grid positions; $\sum_{i,n} NP_{i,n}$ equals total net power with n representing the substation matrix column vector; α represents the feasibility of the windfarm and is a dummy variable (1 = feasible, inf = not feasible), these feasibility requirements are discussed in Section 3.2; and obj is the objective value to be minimised.

3.2. Constraints

In line with previous research, several commonly used constraints are defined. Firstly, there must exist a limit on the minimum and the maximum number of turbines to be placed within the windfarm and the number of turbines cannot exceed these. Secondly, a cable chosen to transfer power between two points must be capable of handling the power flowing through it, this includes all previous power flows. All turbines placed in the windfarm need to have a cable path directing the flow of power back to the substation; for each of these turbines, only one cable transferring power out of each turbine may exist (multiple inputs into one turbine is allowed). Finally, cabling must not cross over each other. Although this is possible in the real world it can result in significant costs and therefore is included as a constraint within the problem formulation.

- C1 A limit range on the number of turbines placed: $t_{lower\_limit} \leq t_{count} \leq t_{upper\_limit}$.
- C2 The cable between two points must be able to support the power load transferred.
- C3 All turbines must be connected back to the substation.
- C4 Turbines can only have one cable from which power flows out (no split power outputs).
- C5 Cabling cannot cross over.

Any violation of these constraints is considered a non-feasible solution. Examples of feasible and non-feasible layouts are shown in Fig. 3.

3.3. Problem instances

A variety of data is used within the optimisation problem, grid site positions, substation position, interference data and cabling data. Within the grid data, for each possible position, a 'Northing' and 'Easting' position is given which is used to represent the solution as the X position and Y position of each possible site. In addition, the substation's position is also given in the same way. The interference data consists of pre-computed wake values within a range of arrays. This is used to quickly determine the wake effect caused on each turbine by all the other turbines currently placed. This is then applied as a factor of reduction to the initially expected power (9.5 MW per turbine) to compute the 'expected' production of each turbine placed.

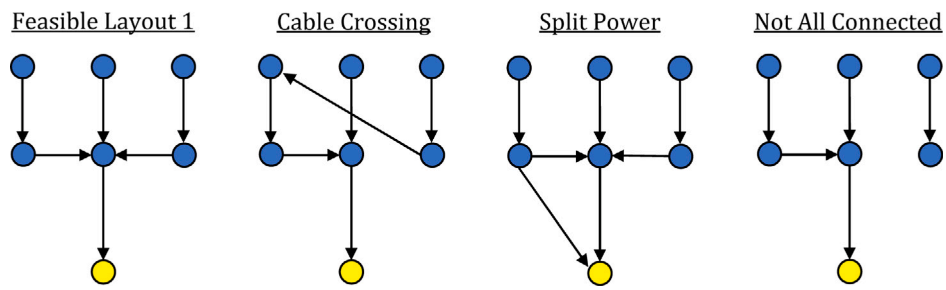


Fig. 3. Examples of feasible and non-feasible (cable crossing, split power output, disconnected turbine) layouts, blue points are turbines with arrows indicating cabling power flow and the yellow point indicates the substation (the power destination).

Table 1
Cabling data.

Cable number	Type	Material	Size [mm^2]	Cost [€/metre]	Current [Amps]	Resistance [Ohm/km]	Voltage [kV]
1	300AL	Aluminium	300	145	450	0.13	66
2	400AL	Aluminium	400	160	530	0.1	66
3	630AL	Aluminium	630	190	650	0.06	66
4	800AL	Aluminium	800	210	700	0.05	66
5	240Cu	Copper	240	190	540	0.1	66
6	630Cu	Copper	630	335	760	0.04	66
7	800Cu	Copper	800	390	810	0.03	66

Table 2
Windfarm problem Instances derived from Borssele 4 located within the Dutch part of the North Sea.

Instance	Name	Siting positions	Size (sq km)	Lower turbine limit	Upper turbine limit
4	Borssele 300	283	179.599192	20	40
3	Borssele 100	110	23.4259816	10	20
2	Borssele 100 (1)	55	10.41154639	5	10
1	Borssele 100 (2)	55	10.41154639	5	10

Data[1] for each cable available within the cabling layout is shown in Table 1. There are two key types of cabling, 'Aluminium' and 'Copper' both of which have varying subtypes with different sizes, cost, current and resistance.

Each turbine is capable of a maximum of 9.5 MW power output in perfect conditions. As no turbine cost data was provided, an estimate has been made based upon information available, for which the estimated cost of each turbine is €10,000,000.

For the data highlighted above, two problem instances are given of varying siting sizes. Both relate to a windfarm named 'Borssele 4' and one of the instances (Borssele 100) is a smaller sample of the larger windfarm (Borssele 300). Two additional instances have been created by splitting Borssele 100 in half, allowing for an increased sample to test algorithms on and verify results. In addition, for each instance, the lower and upper turbine placement limits have been defined based upon the size of the windfarm area, with an increase in maximum turbine placements for a larger area (see Table 2 and Fig. 4).

3.4. Windfarm wake model

For this optimisation study engineering wake models are considered to estimate the wake losses in the windfarm. Engineering wake models used in this study are based on 1D or 2D analytic descriptions of wind turbine wakes and a super-position to calculate the effect of merging wakes. Steady-state CFD type models could be used instead, but that reduces the reproducibility of this paper (see Figs. 5 and 6).

The two analytic wake models considered are the hat-shaped Jensen model described in [19,20] and the Gaussian-shaped model developed by Bastankhah and Porté-Agel [21] and Niayifar and Porté-Agel [22].

The Jensen model is one of the oldest analytic wake models and is based on three key assumptions. First it assumes that the far (turbulent) wake starts immediately after the rotor disk. Therefore, instead of using the rotor disk velocity at the start of the wake, it uses the near wake velocity, u_{nw}, obtained from 1D momentum theory:

$$\frac{u_{nw}}{u_\infty} = \sqrt{1 - C_T} \tag{4}$$

Here, $C_T = C_T(u_{in})$ is the wind turbine's thrust coefficient, which is a function of the incoming wind speed. The second key assumption is that there is only an axial velocity component and that the velocity deficit is constant across the wake. It is therefore sufficient to consider only the mass conservation equation:

$$D_d^2 u_{nw} + \left(D_{fw}^2 - D_d^2\right) u_\infty = D_{fw}^2 u_{fw} \tag{5}$$

where D_d is the rotor disk diameter, D_{fw} is the diameter of the wake, u_∞ is the free stream velocity and u_{fw} is the velocity in the wake. The resulting velocity deficit for the Jensen model becomes:

$$\frac{u_{\mathrm{def}}}{u_\infty} = \left(1 - \sqrt{1 - C_T}\right)\left(\frac{D_d}{D_{fw}}\right)^2 \tag{6}$$

The third key assumption in the Jensen model is that it considers a linear expansion of the wake diameter, with a uniform velocity deficit in radial direction (known as the "top-hat" profile). The wake expansion is given by:

$$\frac{D_{fw}}{D_d} = 1 + 2k_w \frac{x}{D_d} \tag{7}$$

where x is the downstream distance and the parameter k_w is the wake decaying constant, which represents how the wake breaks down due to turbulence by specifying the growth of the wake width. The value of the wake decay coefficient is typically chosen based on the site location, e.g., 0.04 for offshore and 0.075 for onshore. Alternatively, in [23] it is shown that the wake decay coefficient can be made a function of the incoming turbulence intensity or surface roughness length.

The assumptions show that the Jensen model is very limited in its behaviour. Therefore several other analytic wake models have been

[1] Data have been modified due to confidentiality requirements.

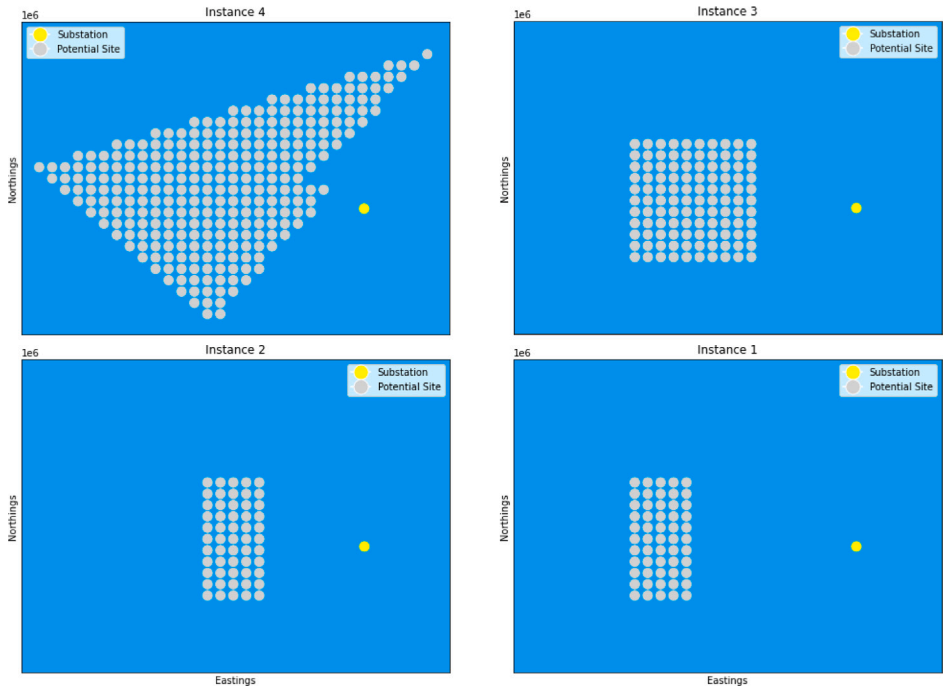


Fig. 4. Windfarm problem instances tested within this paper. Instances 1, 2 and 3 are extracts of the whole windfarm.

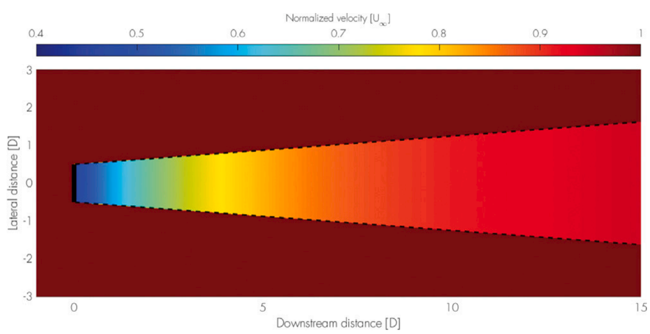


Fig. 5. Jensen wake.

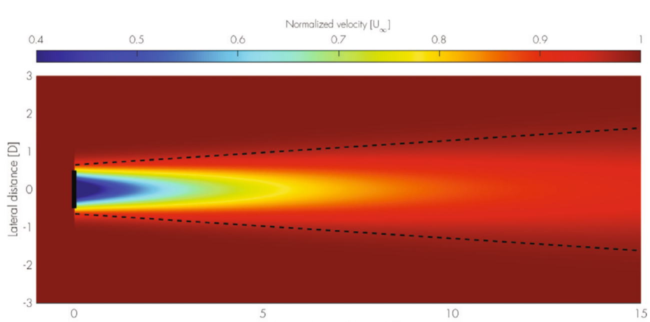


Fig. 6. Example of Gaussian wake visualised through its velocity field and wake diameter (dashed lines) for $C_T = 0.8$.

introduced over time. A more recent and fairly popular analytic wake model is the Gaussian wake model developed by Bastankhah and Porté-Agel [21]. Other than Jensen's model the Gaussian model is derived from the simplified momentum equation:

$$\int_{A_d} \rho u_{fw} \left(u_\infty - u_{fw}\right) \mathrm{d}A = T, \quad \text{with} \quad T = \frac{1}{2} C_T \rho A_d u_\infty^2 \tag{8}$$

where T is the thrust force of the rotor, A_d is the rotor swept area and ρ is the air density at hub height. The Gaussian wake model considers an axisymmetric Gaussian velocity deficit distribution in radial direction. As observed in wind tunnel tests and numerical simulations, especially the time-averaged far wake is well represented by the Gaussian shape. The Gaussian assumption leads to a self-similar solution for the far wake velocity in (8). As a result, the expression for normalised velocity deficit can be given in closed form:

$$\frac{u_{\text{def}}}{u_\infty} = \left(1 - \sqrt{1 - \frac{C_T}{2}\left(\frac{D_d}{2\sigma}\right)^2}\right) \exp\left(-\frac{1}{2}\left(\frac{r}{\sigma}\right)^2\right) \tag{9}$$

where the first term between brackets represents the maximum normalised velocity deficit in the wake at each downwind location, where r is the radial distance from the wake's centre, and σ is the standard deviation of the Gaussian-like velocity deficit profiles at each axial distance x.

Similar to the Jensen model, also the Gaussian wake model by Bastankhah & Porté-Agel assumes a linear expansion of the wake:

$$\frac{\sigma}{D_d} = k^* \frac{x}{D_d} + \varepsilon \tag{10}$$

where k^* is the wake growth rate ($\partial\sigma/\partial x$) (not directly comparable with k_w ($\propto \partial D_{fw}/\partial x$) of the Jensen model) and ε is equivalent to the value of σ/D_d as x approaches zero. Following Niayifar and Porté-Agel [22], the Gaussian model is closed by selecting the parameter ε based on mass conservation and the parameter k^* based on Large Eddy Simulations. The wake growth rate k^* is chosen to be a function of the incoming turbulence intensity, which for waked wind turbines deviate from the free-stream turbulence intensity. For this the same added turbulence intensity model by Crespo & Hernandez is used as was used in [22]. Merging wakes are modelled using a super-position model. There are a range of super-position models, all with their pro's and con's, and none fully representative for all cases, as shown in [24]. In this study we limit ourselves to the sum-of-squares approach, which is most commonly used in commercial codes:

$$\left(u_\infty - \bar{u}_j\right)^2 = \sum_{\forall i<j} \left(u_\infty - \bar{u}_{j,i}\right)^2 \tag{11}$$

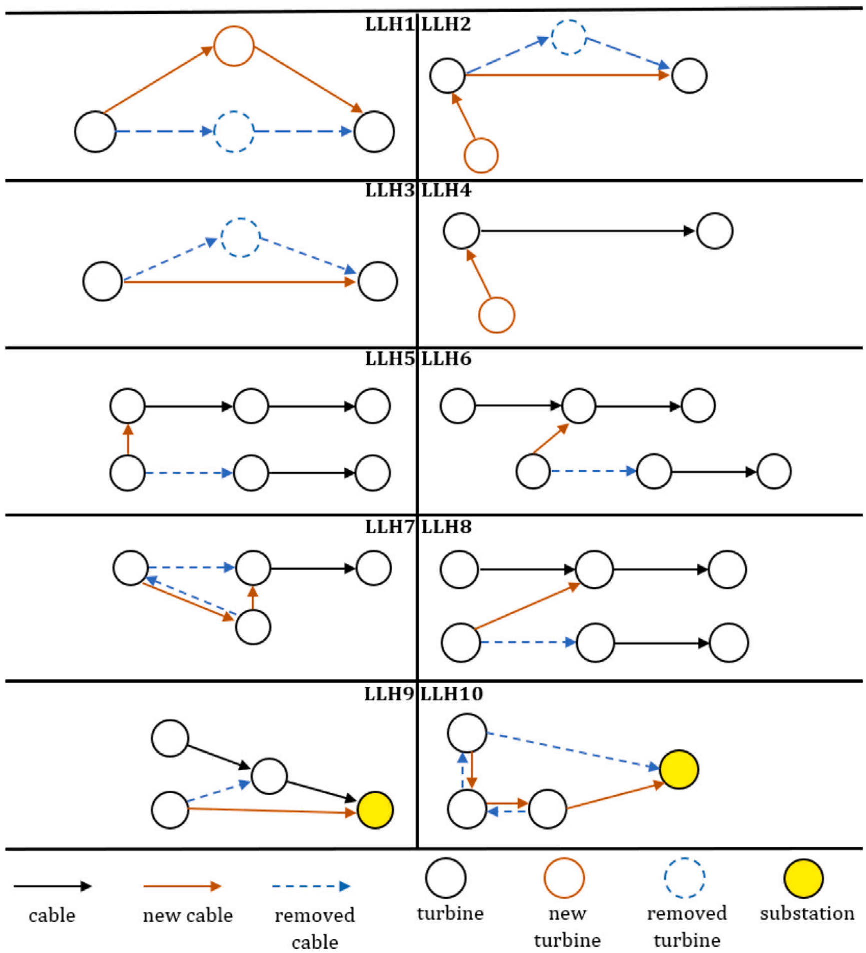


Fig. 7. Visualisation of low-level heuristics and the impact upon a section of a windfarm's layout.

Here, for each individual wake inside the windfarm, the kinetic energy deficit of multiple wakes is assumed to be equal to the sum of the energy deficits from the relevant upwind turbines.

4. Methodology

The primary aim of this paper is to investigate the application of combining optimisation of turbine positions and cable routing simultaneously whilst also determining if a simultaneous or sequential design of a windfarm is more optimal. To do this, selection hyper-heuristics are implemented across a range of selection methods (SM) and move acceptance criteria (MA). The selection hyper-heuristics control a group of pre-defined low-level heuristics (LLH) with the aim to minimise the objective function defined in Section 3.1 subject to the constraints within Section 3.2.

To evaluate if the simultaneous method of optimisation differs or outperforms the current widespread use of the sequential model, two models were developed and tested on each instance. The first model followed the sequential process and the second implemented the combined optimisation approach. The results from all instances, and combinations of MA and SM for both models, were then compared. These two models are referred to as 'sequential' and 'simultaneous'. The sequential model was developed using basic metaheuristics and some of the defined low-level heuristics, the reason past literature was not used was because of the considerable complexity found in replicating methods used. Therefore, the aim of this model was to provide some quantitative ability to compare.

4.1. Low-level heuristics

The selection hyper-heuristic is responsible for selecting which low-level heuristic to implement based upon its own set of criteria. Ten low-level heuristics were created which aim to allow for a wide range of moves and solutions. These are defined below and visualised in Fig. 7.

- **LLH1** Move a turbine within a set range and keep its current cabling path.
- **LLH2** Place a new turbine and connect it to the nearest turbine and remove one elsewhere and migrate its cabling.
- **LLH3** Remove one turbine and migrate its cabling.
- **LLH4** Place a new turbine and connect it to the nearest turbine.
- **LLH5** Connect an endpoint to the nearest endpoint.
- **LLH6** Connect an endpoint to the nearest point (any).
- **LLH7** Swap two end cables around.
- **LLH8** Connect a point (any) to another point (any).
- **LLH9** Identify a branch of turbines and connect one of the turbines direct to the substation instead.
- **LLH10** Identify a branch of turbines and swap the final cable (to the substation) to the closest point in the branch.

Within the described low-level heuristics, the first four (LLH1–LLH4), are primarily focused on the movement and changes to the turbine positions selected and excluded from the sequential model. Whilst the remaining six (LLH5–LLH10) purely re-arranged the current cable routing to find a more optimal layout. All heuristics are available to the simultaneous model.

Start Solution ->	43	44	32	33	17	13	
Swap for Another:	43	44	4	33	17	13	
Remove One:	43	44	32		17	13	
Add Another:	43	44	32	33	17	13	6

Fig. 8. Initialisation of turbine placement heuristics, orange indicates a change in the solution.

4.2. Initialisation methods

An initial windfarm was constructed so that it met all constraints laid out in Section 3.2. The construction process used the sequential stages widely used in previous literature. This process differed for each of the two model types explained below.

4.2.1. Sequential model initialisation

An initial model was constructed in two phases, firstly turbines were placed using a local search algorithm with three simple heuristics. One changes a chosen site for another, the second removes a turbine and the third adds another turbine. These are visualised in Fig. 8, showing an initial selection of six random turbine placements and how the three heuristics impacted them.

The limit on the number of turbines placed is subject to the turbine count limits. This heuristic algorithm was initialised by a random number of arbitrary turbines being chosen. The number of $total\_reps$ the algorithm is run for is equal to the number of potential turbine positions multiplied by one hundred (see Algorithm 1 for detail).

Algorithm 1: Sequential model initialisation algorithm (turbines)

Let $Site\_List[S_1, S_2, ..., S_N]$ be the list of available sites;
Let $Interference$ be an interference matrix;
Let $Power$ be a power matrix;
Let T_{Upper}, T_{Lower} be the upper and lower cap on turbines placed;
Let S be the initial randomly selected sites between T_{Lower}, T_{Upper};
Let $H = [h_1, h_2, h_3]$ be the list of heuristics;
$S_{Best} \leftarrow S$;
$obj_{Best} \leftarrow$ Obj(S_{Best}); /* Obj returns the total power minus total interference */
for $i \leftarrow 0$ **to** $total\_reps$ **do**
    $h \leftarrow$ Random(H);
    $S \leftarrow$ Apply(h, S_{Best});
    $obj \leftarrow$ Obj(S);
    if $obj > obj_{Best}$ **then**
        $obj_{Best} \leftarrow obj$;
        $S_{Best} \leftarrow S$;
    end
end
return S_{Best}

After the turbine positions were optimised, a simple feasible cabling structure was placed. All turbines were directly connected to the substation and no-inter array cabling occurred. For each cable, the correct type is selected based upon the expected power load.

4.2.2. Simultaneous model

Within the simultaneous model, two types of turbine initialisation were examined (cabling remains the same as in Section 4.2.1):

1. Optimised turbine placement as detailed in Section 4.2.1, with cabling direct to the substation.
2. Randomised initial turbine placement with cabling direct to the substation.

The aim of testing both of these initialisations was to determine if a randomised model, with potentially more freedom to optimise turbine placement, could develop a better solution or if a strong initial turbine placement benefits the selection hyper-heuristics later. The randomised turbine placement chose several turbines to place randomly, between the lower turbine limit and upper turbine limit. Once done, a simple random selection of turbine positions was conducted until the chosen number was placed.

4.3. Selection method

As mentioned previously, this paper focused on utilising and comparing a range of heuristic selection methods to determine the most applicable to the windfarm optimisation problem. These were as follows; simple random (SR), sequence-based selection (SS) and a range of selection heuristics labelled ‘best choice’ (BC). Simple random chooses an LLH based on pure randomness. Sequence-based selection is inspired by Kheiri [16], which identifies the next LLH based upon a probability matrix choosing the next LLH with the highest chance of improvement given the previous LLH used; this process is defined within Algorithm 2.

Four further selection methods named ‘best choice’ (BC1, BC2, BC3 and BC4) were developed to investigate different criteria for choosing an LLH. BC1 and BC2 used real-time information from all previous repetitions run to choose the LLH with the largest improvement rate and average improvement amount respectively. The improvement rate is defined as the number of times an LLH choice resulted in a better solution (less than the previous best) divided by the number of times that LLH has occurred in the run. For example, if LLH1 has occurred 50 times within the run and resulted in four better solutions, the improvement rate is $4/50 = 0.08$ or an 8% rate of finding an improvement on average. The average improvement amount follows the same methodology but is the sum of the total improvement amounts found by the respective LLH, divided by the number of times the LLH has occurred in the run. The two remaining selection heuristics, BC3 and BC4 utilise both average improvement rate and average improvement amount, but only kept the information for the most recent five iterations of each respective LLH. These methods aim to test if keeping more recent information provided a better selection of LLH and an overall better solution.

4.4. Move acceptance criteria

To evaluate the impact of each selection heuristic, each was tested using different move acceptance criteria. The move acceptance (MA) defines if a new solution is accepted as compared to the current best solution. Two categories of MA were used; deterministic (only improve, improve or equal and the great deluge) and stochastic (simulated annealing).

Only improve (OI) accepts solutions that are better (reduction in the objective value), improve or equal (IE) will accept solutions that are better or equal to the current best. The great deluge algorithm was first proposed by Dueck [25] and imposes a ‘tolerance value’ (water level)

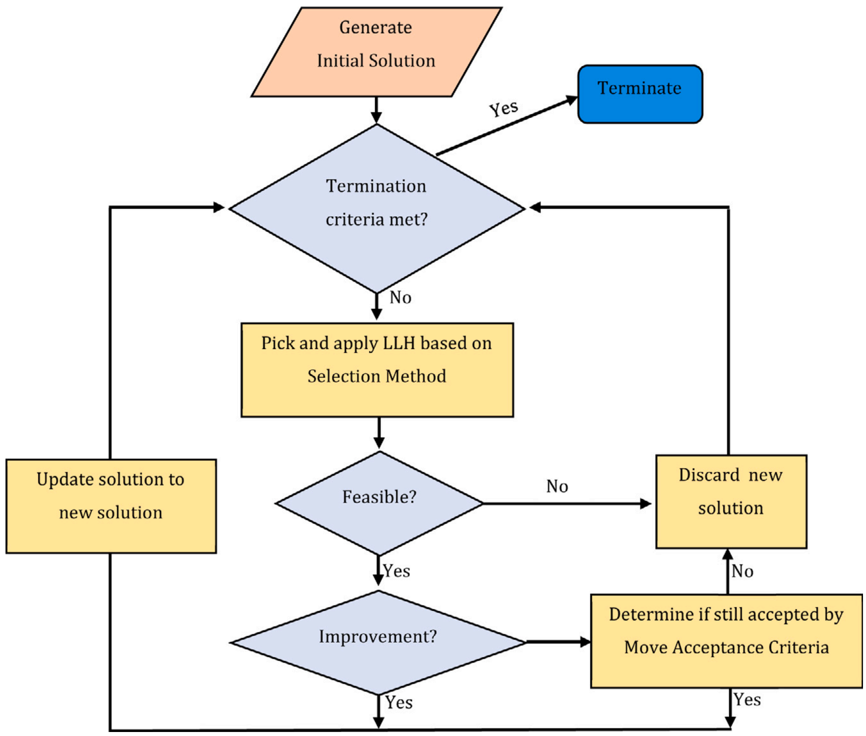


Fig. 9. Overall solution process flowchart.

Algorithm 2: Sequence-based selection algorithm

```
Let LLH be a list of possible low-level heuristics;
Let S_Initial be the initialised solution;
Let obj_Initial be the initialised solution's objective value;
Let Prob_M be the probability matrix initialised with 1's;
Let Rep_M be the repetition matrix initialised with 1's;
Let Improve_M be the improvement matrix initialised with 1's;
Let h, h_previous be the current LLH and previous LLH;
S_Best ← S_Initial;
obj_Best ← obj_Initial;
for i ← 0 to total_reps do
    if (h_previous = null) & (i ≠ 0) then
        h_previous ← h;
        h ← Random(LLH);
    end
    else if h_previous = null then
        h ← Random(LLH);
    end
    else
        h = Prob_M[h_previous].max();    /* Return h with the
         highest probability of improve */
    end
    S ← Apply(h, S_Best);
    obj ← Obj(S);
    if obj < obj_Best then
        obj_Best ← obj;
        S_Best ← S;
        Improve_M[h_previous, h] ← Improve_M[h_previous, h] + 1;
    end
    Rep_M[h_previous, h] ← Rep_M[h_previous, h] + 1;
    Prob_M ← Improve_M / Rep_M;
end
return S_Best
```

for which a solution may still be accepted if below. All improvements are accepted but some non-improvements may still be accepted if below the tolerance value. This changes over time based upon the initial solution value and expected end solution value, this tolerance level is determined as follows:

$$GD_{t,rep} = S_{end} + (S_{initial} - S_{end}) \times (1 - \frac{rep}{total\_reps}) \tag{12}$$

where: $GD_{t,rep}$ is the current tolerance (water level) at a specific rep; S_{end} is the expected best possible final solution; $S_{initial}$ is the initial solution value after the initialisation method; and rep, $total\_reps$ is the current rep and the total number of reps the algorithm is run for. For this study, an end value equal to 75% of the initial solution was used.

Simulated annealing (SA) was also implemented as the final move acceptance method. SA utilises a 'temperature' to try to move away from local optimums and to find the global optimum value. All improvements are accepted in the same manner as the great deluge, but the acceptance of non-improvements is now a stochastic process as opposed to deterministic. The method of acceptance is determined by a probability at a given repetition compared to a random number, whereby if the random float is less than the probability, a solution is accepted. The probability of acceptance can be found by:

$$probability = e^{-\frac{difference}{t}} \tag{13}$$

where: $difference$ is equal to the obj_{Best} minus the $obj_{Current}$; t is the temperature, calculated as the maximum of $\{\min(1, 1 - \frac{rep}{total\_reps}), 0.01\}$; and $probability$ is the chance of accepting a given solution.

4.5. Overall algorithm

The methodology for developing a final windfarm design is shown within the flow chart in Fig. 9. An initial solution was generated based upon the methods introduced in Section 4.2; from this, dependent upon the chosen selection hyper-heuristic, a low-level heuristic is chosen and applied to the initial solution. This was then evaluated for feasibility, and if feasible, it is accepted if it improves upon the initial objective value (reduction in value). If not, then the move acceptance criteria

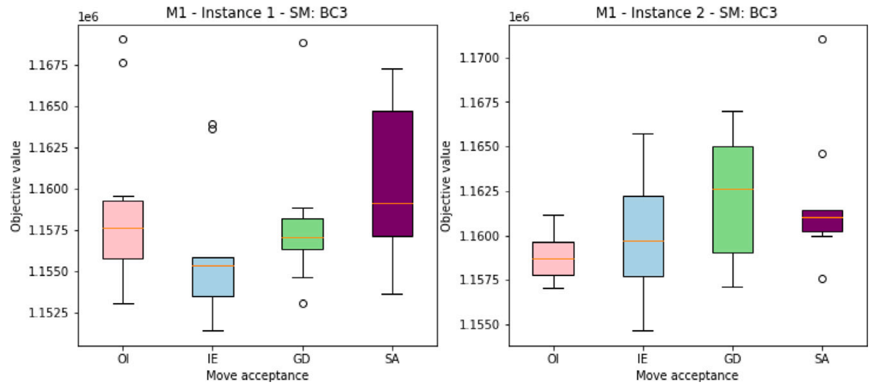


Fig. 10. Box plot from 10 repeats for selection method Best Choice 3 'BC3' for Model 1, Instance 1 and 2, combined with all four move acceptance criteria.

determine whether it is still accepted. The accepted solution then becomes the current solution and the process restarts. If not feasible, the new solution is discarded. This process repeats until the termination criteria (set number of iterations) is met.

In the sequential model, and from the initialised solution detailed in Section 4.2.1, the secondary stage, cable routing, is optimised. For this stage, the solution is iteratively developed following the process in Fig. 9. However, the pool of available low-level heuristics is restricted to just those that impact cabling, with no changes or movements of the current turbine positions. In contrast to the sequential model, the simultaneous models had access to the entire group of low-level heuristics that control both turbine and cable placements.

5. Experimental results

5.1. Expectations and hypothesis

The main investigative aim of this work is to evaluate if simultaneous optimisation of turbine placement and cable routing can provide any benefit over a sequential optimisation. This is on top of the objective to identify the best selection hyper-heuristic and move acceptance criteria for both of these optimisation types. To achieve these aims, a statistical evaluation was undertaken for the different models: sequential (M1), simultaneous (M2), and a variant of M2, referred to as simultaneous with an optimised start (M3). For these three models, the determination of the 'best' algorithm was as follows: For each of the first three problem instances (samples of Instance 4) run all combinations of selection hyper-heuristics and move acceptance criteria for a set number of repeats to ensure reliable results. The non-parametric Mann–Whitney U test was conducted between each pair of SM and MA at the 5% significance level. Where an algorithm is considered to have statistically significantly outperformed another if the average value of its repeats is less than another and the p-value from the non-parametric test is less than or equal to 5%. The algorithm with the best performance (statistically better than the greatest number of others) from each instance was then selected. From these algorithms, the best overall performer(s) were determined. Once the 'best' algorithm(s) from each model had been chosen, this was then applied to Instance 4 (the entire windfarm) for a longer number of iterations to allow for comparison between each model type.

The hypotheses set for this study are as follows:

- H_0 (Null Hypothesis): Sequential optimisation outperforms any method of simultaneous optimisation.
- H_1 (Alternate Hypothesis): Simultaneous optimisation outperforms traditional sequential methods.

5.2. Experiment setup

Each model (M1, M2 and M3) was applied to each of the smaller problem instances (1, 2 and 3, Fig. 4). This was run for every combination of selection method and move acceptance criteria. Each combination was run for ten repeats of 1000 iterations each time and the average, standard deviation and minimum values were measured over those repeats. Experiments were carried out on a computer with specifications: Intel Core i7 7700HQ (3.5 GHz) and 16 GB of 2400 MHz DDR4 memory. Each algorithm was compared against all other algorithms using the Mann–Whitney U test with a 5% significance level. This allowed for comparison to determine if, over the ten repeats, an algorithm is statistically different to another. Further identifying comparisons were made between each algorithm. Given algorithm A and algorithm B:

- A is statistically better than B (>)
- A is statistically worse than B (<)
- A is better than B but with no statistical significance (≥)
- A is worse than B but with no statistical significance (≤)

5.3. Model 1 (Sequential optimisation)

Table 3 shows the results from Model 1 concerning the objective function previously defined (Eq. (3)). Across Instance 1 and 2, selection method 'BC3' statistically outperformed all other algorithms when using GD or OI move acceptance criteria in Instance 1 and 2 respectively. The global minimum for Instances 1 and 2 also occurred when pairing BC3 with IE (Fig. 10). Within Instance 3, however, the best algorithm was using SR and SA, outperforming 18 of the other 23 selection hyper-heuristic combinations (Fig. 11). The min value found by SR:SA was also within 0.17% of the global minimum for Instance 3. This indicates that a wide equal usage, that a random selection brings, was most optimal in this instance size.

An interesting observation is an overall reduction in average objective value across all algorithms in Instance 3, a larger windfarm, in comparison to the smaller windfarm Instances 1 and 2. It indicates there could be a non-linear relationship (within Model 1) for the larger the windfarm the lower the ratio of cost to net power produced, potentially due to increased numbers of turbines allowing a wider range of cabling configurations.

Based upon the findings within Instances 1 and 2, selection method BC3 combined with move acceptance IE performs best. Due to the considerable difference in findings in Instance 3 compared to 1 and 2, a second algorithm SR combined with SA was also carried forward for application to Instance 4, the whole windfarm.

Table 3
Model 1 results for each selection hyper-heuristic and paired move acceptance criteria for Instances 1, 2 and 3. The best values for each Instance are shown in bold.

SM	MA	Instance 1							Instance 2							Instance 3						
		avg	std	Min	>	<	≥	≤	avg	std	min	>	<	≥	≤	avg	std	min	>	<	≥	≤
SR	OI	1 157 147	3435.644	1 152 042	3	0	19	1	1 160 548	2292.023	1 158 691	3	0	15	5	815 326.1	156 124.0	603 783.8	2	1	7	13
	IE	1 159 916	2651.326	1 154 904	1	6	7	9	1 160 951	2389.867	1 158 313	3	1	12	7	762 951.7	143 597.3	607 194.2	2	1	15	5
	GD	1 157 458	2561.657	1 154 449	4	0	16	3	1 161 135	3091.992	1 157 495	3	0	11	9	870 546.6	72 814.70	711 890.4	2	14	4	3
	SA	1 157 542	3151.316	1 153 103	6	0	13	4	1 162 967	5556.139	1 156 527	3	1	1	18	**590 977.1**	49 517.92	533 245.9	**18**	0	5	0
SS	OI	1 158 544	3769.635	1 153 103	2	0	12	9	1 160 181	3416.056	1 155 676	5	0	14	4	793 038.6	189 527.1	601 217.3	2	0	12	9
	IE	1 159 554	4729.139	1 153 103	1	0	9	13	1 159 360	2388.801	1 155 798	6	0	16	1	825 359.7	188 734.5	604 932.3	4	1	4	14
	GD	1 161 512	4092.74	1 156 020	0	5	4	14	1 162 240	4157.963	1 157 709	3	0	5	15	879 447.9	137 595.5	600 704.9	0	3	4	16
	SA	1 167 063	12 121.62	1 153 103	0	3	1	19	1 162 897	4632.429	1 156 527	3	1	2	17	785 002.0	108 259.8	651 941.5	1	1	14	7
BC1	OI	1 158 584	4744.257	1 153 709	1	0	12	10	1 161 465	3168.095	1 158 313	3	1	9	10	763 610.1	135 881.5	**532 339.0**	0	1	16	6
	IE	1 158 425	3403.101	1 153 103	1	0	14	8	1 161 769	3072.114	1 156 285	3	3	8	9	905 482.6	157 230.9	738 700.5	0	2	2	19
	GD	1 160 883	5530.535	1 153 709	0	1	5	17	1 163 123	3449.683	1 158 798	2	4	1	16	936 621.2	123 938.1	740 841.1	0	1	0	22
	SA	1 166 247	11 844.42	1 151 930	0	5	2	16	1 170 823	7599.826	1 158 717	0	21	2	0	743 543.3	92 197.74	602 547.2	3	1	18	1
BC2	OI	1 158 369	3008.349	1 153 709	1	0	15	7	1 159 677	2182.172	1 156 409	4	0	17	2	803 874.8	96 826.41	655 699.1	1	2	10	10
	IE	1 159 015	4012.57	1 153 313	1	0	10	12	1 160 581	2327.675	1 157 424	3	0	14	6	877 056.0	137 987.0	698 241.9	1	2	4	16
	GD	1 159 832	4465.864	1 154 638	1	0	8	14	1 162 846	3731.015	1 158 891	3	2	3	15	851 624.2	72 878.81	707 626.3	0	7	7	9
	SA	1 160 552	4247.512	1 154 904	1	2	5	15	1 162 452	4575.043	1 156 746	3	1	4	15	757 084.3	128 152.8	533 137.1	3	1	16	3
BC3	OI	1 158 975	5301.762	1 153 103	1	0	11	11	**1 158 792**	1321.925	1 157 030	**13**	0	10	0	716 934.9	97 845.25	602 067.7	3	1	19	0
	IE	**1 156 219**	4230.309	**1 151 403**	7	0	16	0	1 159 914	3245.064	**1 154 637**	3	0	17	3	758 643.5	162 779.5	602 261.1	4	1	14	4
	GD	1 157 887	4205.866	1 153 103	**8**	0	10	5	1 162 142	3673.511	1 157 100	2	0	7	14	898 410.2	114 366.0	695 809.9	1	13	2	7
	SA	1 160 442	4733.583	1 153 631	1	1	6	15	1 161 893	3647.636	1 157 583	3	1	7	12	755 134.9	128 507.1	532 637.5	2	1	18	2
BC4	OI	1 157 320	3062.818	1 153 103	6	0	15	2	1 161 219	3302.393	1 156 512	3	1	10	9	800 513.7	173 565.0	532 406.1	1	0	11	11
	IE	1 158 023	3043.352	1 153 103	3	0	14	6	1 160 885	3761.918	1 156 150	3	0	13	7	813 684.7	198 370.5	605 212.4	4	2	6	11
	GD	1 165 630	10 218.37	1 154 980	0	8	3	12	1 174 926	8181.377	1 159 795	0	21	0	2	922 768.1	164 492.4	606 769.9	0	0	1	22
	SA	1 167 981	9698.329	1 157 697	0	18	0	5	1 171 211	10 425.87	1 159 795	0	19	1	3	798 107.2	150 884.8	701 442.0	3	1	10	9

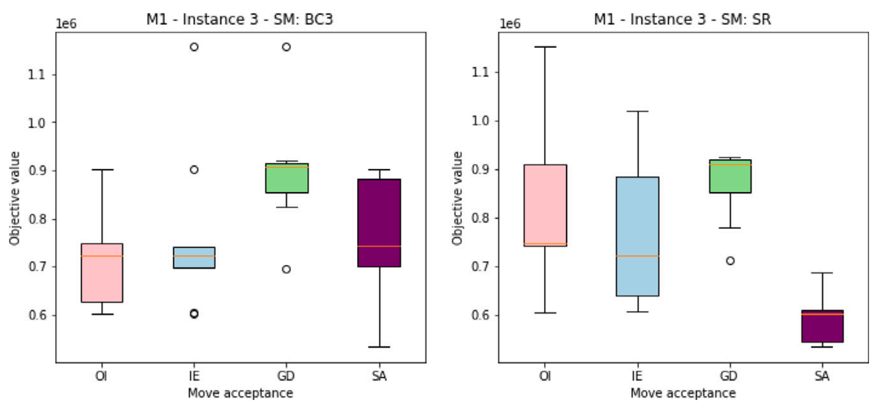


Fig. 11. Box plot for 10 repeats using selection method Best Choice 3 'BC3' (left) and Simple Random 'SR' (right) for Model 1, Instance 3, combined with all four move acceptance criteria.

5.4. Model 2 (Simultaneous optimisation with random turbine start)

Table 4 presents the results from a simultaneous optimisation; utilising all low-level heuristics to move both turbines and cabling at the same time with a randomised initial turbine layout. Results from the Mann–Whitney U pairwise comparison showed that across Instances 1 and 2, the SR selection method performed the overall best, with other notable results showing IE to contain both the global minimums for each Instance (1 and 2) (see Fig. 12). However, in the larger Instance 3, BC1 paired with OI performed the best in terms of average run value, minimum value and statistical outperformance of other algorithms. Part of this trend can also be found within Instances 1 and 2 where both the minimum values occurred within BC1 paired with IE (see Fig. 13). The pairing of BC1 and IE within Instances 1 and 2 also outperformed 13 other algorithms in each case compared to the best which outperformed 15. Based upon these findings, two methods were tested on Instance 4; SR:IE and BC1:OI.

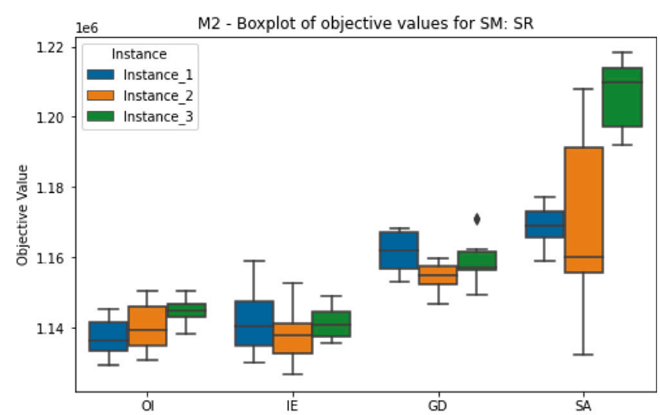


Fig. 12. Model 2 boxplot for selection method Simple Random for Instance 1, 2 and 3.

5.5. Model 3 (Simultaneous optimisation with optimised turbine start)

Table 5 summarises the results from Model 3, which initially generated an optimised turbine layout and then applied a variety of selection

Table 4
Model 2 results for each selection hyper-heuristic and paired move acceptance criteria for Instances 1, 2 and 3. The best values for each Instance are shown in bold.

SM	MA	Instance 1							Instance 2							Instance 3						
		avg	std	Min	>	<	≥	≤	avg	std	min	>	<	≥	≤	avg	std	min	>	<	≥	≤
SR	OI	**1 137 094**	5274.647	1 129 519	**15**	0	8	0	1 143 353	8455.036	1 136 165	11	0	6	6	1 144 694	3325.897	1 138 260	10	2	3	8
	IE	1 141 645	9105.192	1 130 164	12	0	8	3	1 137 104	4398.609	1 130 079	**15**	0	7	1	1 141 192	4491.485	1 135 720	13	0	5	5
	GD	1 161 871	5639.469	1 152 959	4	16	3	0	1 160 424	5638.503	1 152 926	2	15	5	1	1 159 380	7099.724	1 149 392	6	12	3	2
	SA	1 168 715	5586.059	1 159 107	2	17	3	1	1 165 146	5072.792	1 159 315	2	15	3	3	1 206 341	9509.750	1 191 948	0	22	1	0
SS	OI	1 142 912	7559.313	1 131 587	12	0	6	5	1 144 747	8129.454	1 134 522	11	1	4	7	1 144 203	6154.868	1 134 415	11	0	3	9
	IE	1 143 867	5936.956	1 136 765	12	0	3	8	1 145 177	5482.428	1 138 321	12	1	1	9	1 143 058	6870.426	1 134 786	11	0	5	7
	GD	1 151 677	5397.095	1 144 468	8	10	3	2	1 157 299	7514.475	1 146 985	3	11	5	4	1 162 679	10 030.10	1 150 983	6	12	1	4
	SA	1 181 271	21 014.83	1 150 557	0	16	2	5	1 173 381	14 432.26	1 158 058	1	17	1	4	1 211 797	14 068.94	1 186 287	0	22	0	1
BC1	OI	1 145 003	7011.122	1 132 672	11	1	2	9	1 140 280	7135.619	1 130 740	12	0	6	5	**1 137 577**	7927.860	**1 124 750**	**14**	0	9	0
	IE	1 140 058	9694.889	**1 125 464**	13	0	9	1	1 137 857	8644.342	**1 126 603**	13	0	8	2	1 141 053	12 165.15	1 127 858	12	0	7	4
	GD	1 152 460	6522.597	1 137 643	8	10	1	4	1 154 233	4186.612	1 146 746	7	11	3	2	1 152 781	11 452.58	1 136 162	6	10	5	2
	SA	1 171 466	19 586.23	1 150 801	0	16	4	3	1 167 752	27 105.47	1 132 153	1	10	2	10	1 168 052	29 983.09	1 131 951	2	8	4	9
BC2	OI	1 143 662	8224.775	1 132 851	12	0	4	7	**1 136 018**	8909.451	1 127 806	14	0	9	0	1 147 679	3935.976	1 143 337	10	5	2	6
	IE	1 141 415	6144.944	1 132 928	14	0	7	2	1 143 802	8802.568	1 131 050	11	0	5	7	1 139 571	6744.264	1 130 350	13	0	8	2
	GD	1 151 720	4564.318	1 146 413	8	11	2	2	1 152 490	7075.229	1 139 223	7	8	4	4	1 155 754	8808.224	1 136 107	6	12	4	1
	SA	1 166 219	9417.751	1 153 151	2	16	4	1	1 160 877	4306.847	1 153 496	3	15	3	2	1 186 419	17 215.57	1 153 099	2	17	1	3
BC3	OI	1 144 991	6460.86	1 131 628	11	2	3	7	1 149 363	10 055.42	1 132 140	7	4	5	7	1 142 236	6438.867	1 134 518	12	0	5	6
	IE	1 142 337	8726.38	1 131 226	12	0	7	4	1 139 752	5664.844	1 130 242	12	0	7	4	1 143 222	6181.775	1 130 522	12	0	3	8
	GD	1 152 847	3835.706	1 147 853	8	11	0	4	1 154 606	6991.417	1 140 784	7	11	2	3	1 162 030	6482.014	1 148 101	6	12	2	3
	SA	1 175 828	11 708.25	1 159 401	2	17	1	3	1 166 911	13 311.68	1 152 442	2	15	2	4	1 183 856	13 860.28	1 158 541	2	17	2	2
BC4	OI	1 147 269	9258.166	1 130 099	8	3	4	8	1 144 921	7604.365	1 135 387	11	1	3	8	1 140 355	6641.533	1 129 398	13	0	7	3
	IE	1 143 253	7422.153	1 134 978	12	0	5	6	1 138 915	6106.091	1 129 794	13	0	7	3	1 139 536	4391.514	1 134 853	**14**	0	8	1
	GD	1 191 835	11 563.21	1 172 261	0	20	0	3	1 203 561	16 167.52	1 179 092	0	23	0	0	1 190 437	7625.244	1 179 310	2	17	0	4
	SA	1 188 838	14 278.06	1 168 044	0	20	1	2	1 186 272	25 425.27	1 155 092	1	20	0	2	1 183 847	10 661.10	1 168 580	2	17	3	1

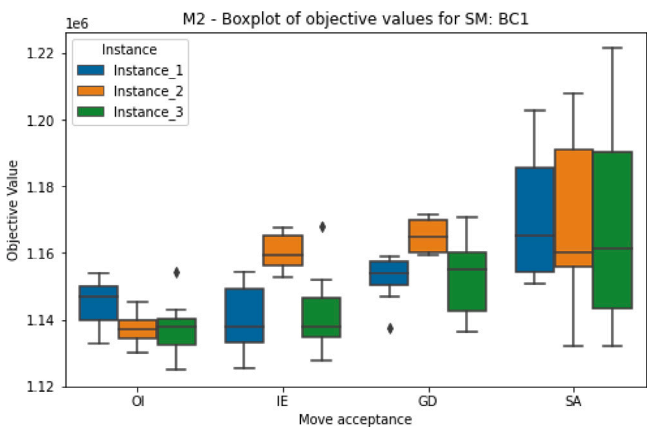


Fig. 13. Model 2 boxplot for selection method Best Choice 1 for Instance 1, 2 and 3.

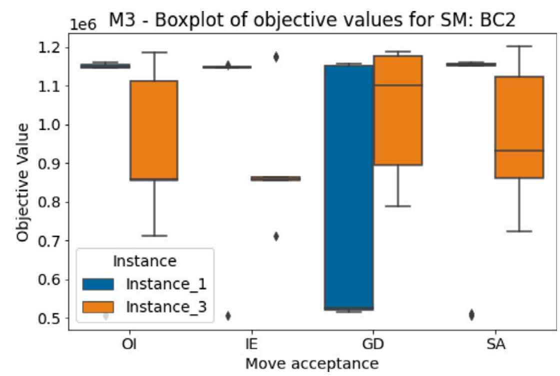


Fig. 14. Model 3 boxplot for selection method Best Choice 2 for Instance 1 and 3.

methods and move acceptance criteria to both turbine positions and the cabling layout. From the statistical pair-wise tests, there is no clear overall best algorithm. Instances 1 and 3 show the strongest performance from selection method BC2 paired with move acceptance GD and SA respectively (see Fig. 14). However, the best minimum run values from these two Instances, appear when using the SS selection method. Contrasting this, Instance 2 showed more consistency in run results with the best average, minimum and overall performance present within BC4, where move acceptance OI is the best performer. It was not clear from Instance 1 and 3 which combination is most successful so BC2 paired with both GD and SA was included within the algorithms tested upon Instance 4 (entire windfarm). Therefore, BC4:OI, BC2:GD and BC2:SA were tested further on Instance 4.

5.6. Further experiments

Table 6 summarises the results employing the best algorithms found from each of the three model types. These are applied for a longer length of iterations to the complete windfarm instance described as Instance 4 to identify and allow for a comparison of the overall performance between model types. Both deterministic and stochastic methods are included with all move acceptance criteria appearing. Selection Sequence (SS) is the only selection method not carried forward, this may be due to the difficulty in identifying appropriate sequences with the high randomness present and a great number of potential cabling layouts.

The table shows that in the long-run results for the two best-performing algorithms within Model 1, both methods can improve upon the initially generated solution, improving both cable costs and overall net power. Selection method 'BC3' combined with IE performed slightly better overall compared to SR:SA and was able to reduce cabling costs by a further 7,000,000 euros. Fig. 15 demonstrates the difference of using simulated annealing against improve or equal, with simulated annealing identifying significantly more local optimums, but this still underperformed compared to the combination of BC3 and IE. The usage of each low-level heuristic is shown in Fig. 16, there are only slight differences between utilisation rates, suggesting that within Model 1, a wide even selection of LLH is most effective.

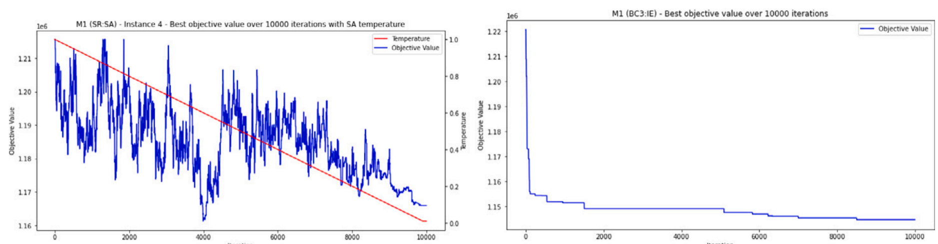
In Model 2, Table 6 indicates that selection method SR paired with move acceptance IE outperformed the pairing of BC1:OI by an objective

Table 5
Model 3 results for each selection hyper-heuristic and paired move acceptance criteria for Instances 1, 2 and 3. The best values for each Instance are shown in bold.

SM	MA	Instance 1							Instance 2							Instance 3						
		avg	std	Min	>	<	≥	≤	avg	std	min	>	<	≥	≤	avg	std	min	>	<	≥	≤
SR	OI	1 020 144	270 539.8	505 491.9	1	1	8	13	1 159 360	5237.905	1 152 037	10	0	4	9	943 104.7	163 697.6	718 231.9	0	0	20	3
	IE	1 018 611	270 823.3	504 351.1	4	3	7	9	1 159 251	4849.177	1 153 846	9	0	6	8	1 052 326	156 560.3	850 908.6	0	5	5	13
	GD	963 904.5	299 130	522 362.6	0	0	16	7	1 165 863	5352.275	1 159 989	6	12	1	4	1 037 098	149 695.9	888 752.0	**3**	0	5	15
	SA	897 410.6	335 558	507 441.3	7	1	13	2	1 176 523	6642.833	1 166 064	3	17	1	2	915 595.3	174 542.0	724 988.7	0	0	21	2
SS	OI	954 296.7	309 717.2	503 683.7	0	1	18	4	1 159 944	4625.848	1 154 170	9	0	4	10	966 962	185 178.4	**709 716.7**	0	0	14	9
	IE	890 450	331 359.5	503 577	0	1	21	1	1 157 407	2908.019	1 153 313	12	0	10	1	909 562.2	191 564.5	715 197.7	0	0	22	1
	GD	901 304.4	318 336.8	512 227.5	0	0	19	4	1 163 653	5947.454	1 153 877	7	5	4	7	1 051 630	158 466.0	796 619.7	0	0	6	17
	SA	1 033 356	279 217.1	**503 179.9**	2	5	3	13	1 193 165	12 102.5	1 172 053	0	20	1	2	1 006 618	139 652.5	865 298.6	2	0	11	10
BC1	OI	1 017 607	269 931.9	504 240.6	4	3	10	6	1 157 912	4268.111	1 152 016	12	0	8	3	961 814.5	146 688.9	847 534.9	0	0	15	8
	IE	954 623.2	310 068	503 778	0	2	17	4	1 157 845	2361.792	1 153 846	12	0	9	2	948 691.5	202 485.5	715 964.6	0	0	18	5
	GD	1 019 792	262 406.7	520 265.8	1	1	9	12	1 165 804	4712.667	1 159 465	6	10	2	5	1 015 068	145 941.5	863 000.8	0	0	11	12
	SA	1 094 167	200 550.3	524 828.7	0	1	2	20	1 175 229	9798.443	1 160 472	3	15	2	3	1 149 290	108 926.1	875 269.0	0	0	1	22
BC2	OI	1 022 245	271 446.7	506 149.7	3	2	5	13	1 158 108	3064.212	1 154 801	12	0	6	5	946 126.7	170 570.6	712 130.1	2	0	17	4
	IE	1 083 655	203 111.5	505 640.4	1	6	2	14	1 160 262	4545.656	1 154 801	8	1	4	10	**907 068.4**	148 560.6	711 692.3	2	0	21	0
	GD	**772 454**	326 280.5	513 747	**14**	0	9	0	1 163 942	5104.055	1 159 329	7	8	2	6	1 038 822	158 323.7	787 218.7	0	0	7	16
	SA	1 025 201	272 507.9	505 311	3	4	3	13	1 174 240	7891.829	1 162 200	3	17	3	0	955 156.5	179 669.8	723 532.5	**3**	0	14	6
BC3	OI	1 018 111	270 402.6	504 178	4	3	9	7	1 159 053	2995.128	1 153 450	10	0	7	6	960 748.8	190 268.4	717 989.2	0	0	16	7
	IE	1 018 309	269 746.9	505 562.6	5	3	7	8	1 158 061	4283.742	1 153 450	10	0	9	4	1 022 398	163 137.5	852 852.7	0	0	10	13
	GD	1 022 388	267 220.8	514 220.8	1	1	6	15	1 163 776	6067.37	1 153 096	7	5	3	8	1 007 909	142 033.1	878 759.1	0	0	12	11
	SA	964 666.4	313 115.4	508 082.8	7	0	8	8	1 184 109	13 971	1 163 501	0	17	3	3	1 036 176	170 838.1	784 252.6	**3**	0	6	14
BC4	OI	889 710.7	331 107.8	503 325.3	1	1	21	0	**1 156 214**	2653.784	1 151 664	**13**	0	10	0	1 059 117	147 778.2	853 916.5	0	3	4	16
	IE	1 148 912	992.0097	1 147 053	1	7	0	15	1 159 171	5824.655	**1 151 332**	8	0	8	7	1 110 503	134 021.6	853 970.7	2	6	1	14
	GD	1 170 877	20 013.35	1 148 896	0	7	0	16	1 188 886	13 064.33	1 170 392	0	20	2	1	1 206 921	11 676.23	1 189 541.0	0	2	0	21
	SA	1 034 823	276 569.6	504 363.9	3	9	1	10	1 196 367	18 069.89	1 173 335	0	20	0	3	1 132 735	126 262.9	887 238.7	0	1	2	20

Table 6
Long-run experiment results on Instance 4 over 10 000–20 000 iterations using algorithms identified as top performers from initial experiments. Costs are in euros and power is in MW. M1 is run for fewer iterations as it only utilises half the available low-level heuristics. Best values and algorithm highlighted in bold.

Model 1		Obj value	Turbine costs	Cabling costs	Initial power	Net power	Iterations
SR:SA	Initial	1 215 651.941	400 000 000	39 553 772.98	380	361.579	10 000
	Final	1 165 973.054	**400 000 000**	25 072 631.06	380	364.565	
BC3:IE	Initial	1 220 731.388	400 000 000	40 456 068.52	380	360.813	10 000
	Final	**1 144 670.286**	**400 000 000**	**18 202 570.88**	380	**365.348**	
Model 2		**Obj value**	**Turbine costs**	**Cabling costs**	**Initial power**	**Net power**	**Iterations**
SR:IE	Initial	1 246 198.183	230 000 000	20 775 423.47	218.5	201.2323778	20 000
	Final	**1 134 551.794**	**290 000 000**	**11 517 439.13**	275.5	265.7590783	
BC1:OI	Initial	1 291 983.711	360 000 000	37 433 156.93	342	307.6146808	20 000
	Final	1 149 372.179	400 000 000	16 306 408.95	380	**362.2033112**	
Model 3		**Obj value**	**Turbine costs**	**Cabling costs**	**Initial power**	**Net power**	**Iterations**
BC2:GD	Initial	1 228 532.417	400 000 000	39 910 328	380	358.0779163	20 000
	Final	1 151 705.025	**400 000 000**	**15 901 043.5**	380	361.1176772	
BC2:SA	Initial	1 214 633.283	400 000 000	39 704 217.89	380	362.005738	20 000
	Final	1 172 362.878	**400 000 000**	19 082 779.61	380	357.4684829	
BC4:OI	Initial	1 216 328.864	400 000 000	40 684 725.73	380	362.307217	20 000
	Final	**1 142 579.182**	**400 000 000**	18 057 543.06	380	**365.8893402**	



(a) (b)

Fig. 15. Model 1 with (a) Simple Random: Simulated Annealing objective value over 10 000 iterations, combined with the simulated annealing temperature, and (b) Best Choice 3: Improve or Equal best objective value over 10 000 iterations.

value of nearly 15,000. However, both algorithms had different initialised numbers of turbines due to the random start element of Model 2, therefore the difference may not be significant with BC1:OI having a larger number of initial turbines, possibly adding complexity to the ability to solve the problem efficiently. This additional complexity can be seen within the final layout for both algorithms in Fig. 17. Fig. 18

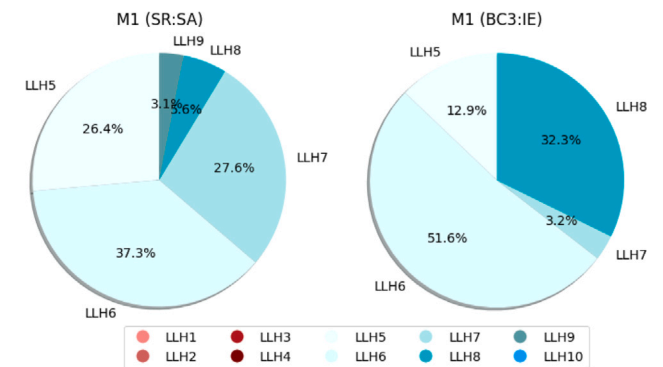


Fig. 16. Model 1 algorithm heuristic utilisation rates Simple Random: Simulated Annealing (left) and Best Choice 3: Improve or Equal (right).

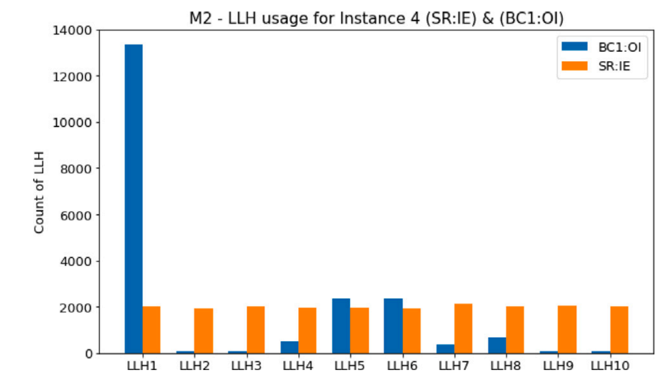


Fig. 18. Low-level heuristic usage for each algorithm within Model 2 applied to Instance 4 for 20 000 iterations.

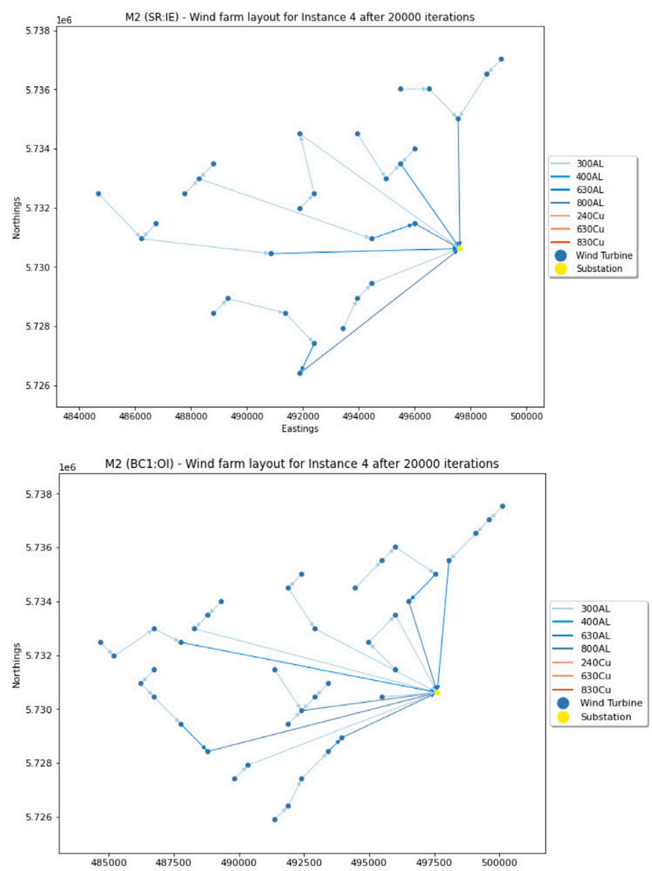


Fig. 17. Windfarm layout for Model 2 algorithms Simple Random: Simulated Annealing (top) and Best Choice 1: Only Improve (bottom) after 20 000 iterations on Instance 4.

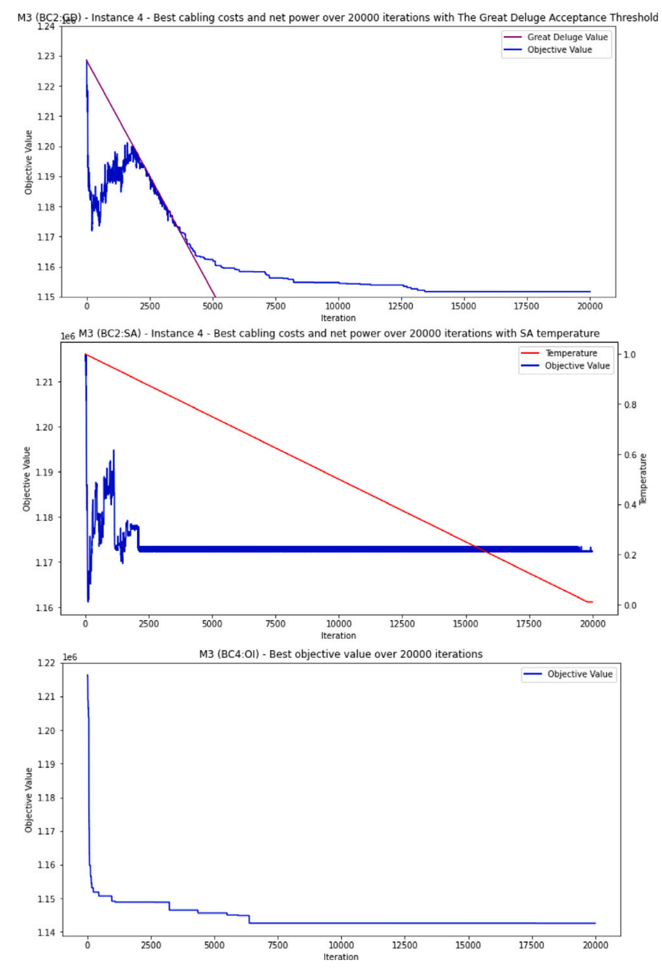


Fig. 19. Objective over 20 000 iterations for Model 3 with algorithms Best Choice 2: Great Deluge (top), Best Choice 2: Simulated Annealing (middle) and Best Choice 4: Only Improve (bottom).

shows, as expected, LLH utilisation rates are even when using simple random (SR) however when using best choice 1 (BC1) there was a clear preference toward LLH1, which moved turbines within a nearby space.

For Model 3, Table 6 shows that none of the three algorithms tested added or removed any turbines from the initial starting number of 40. Within the final objective value, it can be seen that BC4:OI outperformed the other two algorithms, even with BC2:SA having a slight advantage with a lower initial objective value. However, BC2:GD was able to find significantly cheaper cabling costs, but this was at the expense of net power with an increase in losses due to the wake effect as turbines got closer together (minimising cable distance). Fig. 19 highlights that BC2:SA got stuck within two local optima, consistently going back and forth between them. Whilst BC2:GD successfully use the GD acceptance threshold to move away from a local optimum. From these results, the overall best algorithm was BC4:OI with the lowest objective value.

Best pairings of selection method and move acceptance criteria for Model 1, 2 and 3 from the further experiments conducted upon Instance 4 are: BC3:IE, SR:IE and BC4:OI. Fig. 20 shows the utilisation rates for each of the three selection hyper-heuristics where the LLH chosen resulted in an improvement (reduction in solution objective). Model 1 was restricted to just LLHs that impact cabling and within those,

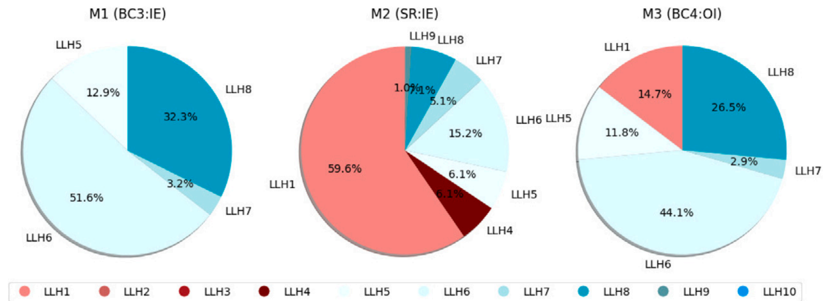


Fig. 20. Low-level heuristic utilisation that resulted in improvements for Model 1 (Best Choice 3: Improve or Equal), Model 2 (Simple Random: Improve or Equal) and Model 3 (Best Choice 4: Only Improve). Shades of red indicate heuristics that aim to impact turbines, shades of blue represent those that impact cabling.

LLH6 (connect endpoint to nearest turbine) proved most successful in finding improvements. With LLH8 (connect any point to nearest turbine) second. Within Model 2, which had a randomised turbine start, the movement of turbines, LLH1, provided the most improvements as expected because this heuristic allows for minimisation of the wake effect to occur. Model 3 benefited most from LLH6 in the same fashion as Model 1, LLH6 was also the second most successful within Model 2. These results indicate that the simpler the low-level heuristic the more accessible it was to bring about an improvement in the solution objective value.

The results show that the customised selection methods were unable, in this instance and run, to beat a simple random selection (SR) method paired with improve or equal (IE) (layout shown in Fig. 17). With an overall objective value of 0.88% and 0.7% better than BC3:IE, BC4:OI respectively. In terms of model type (sequential against simultaneous), the results do not give a clear decisive answer. With a very small variation recorded between each final objective, it cannot conclusively determine if either outperforms the other. In addition, the randomness of each low-level heuristic must be considered and that some selection hyper-heuristics may have been ‘lucky’ with the improvements found by each LLH.

Li et al. [2] implemented a range of hyper-heuristics to control state-of-the art low-level metaheuristics to solve each aspect of the windfarm layout. Their findings showed that random choice (referred to as simple random on this paper) performed better than the implemented choice function (similar to BC1,2,3 and 4 in this paper). These findings are consistent with findings when applied to Instance 4 from the long-run experiments, where simple random prevailed as the most effective in finding an optimal solution. The Borselle 4 problem instances have also been solved by Fischetti [1]. This was solved in stages and used a range of methodologies including heuristics and MILP. However, it is not possible to directly compare results due to the varying constraints, power data, turbine costs considered and complexity of the windfarm optimisation problem, alongside the difficulty in replicating the cable routing optimisation undertaken within the paper.

6. Conclusion

This paper investigated the application of selection hyper-heuristics to solving the windfarm optimisation problem; specifically, the proposal to combine both the turbine optimisation problem and cable routing problem simultaneously, rather than sequentially. Objectives included the maximisation of the expected net power, minimisation of both turbine costs and cabling costs. The aim was to solve this complex problem computationally using selection hyper-heuristics that combined a selection method with a move acceptance criteria. Several previously documented selection methods and move acceptance were used alongside the development of customised selection methods. These selection hyper-heuristics were applied to three different models: M1 — sequential optimisation, M2 — simultaneous optimisation with random start and M3 — simultaneous optimisation with an optimised start. Further experiments run on the best selection hyper-heuristic combinations found within the initial experiments, identified the following three algorithms as the best for each model type BC3:IE (M1), SR:IE (M2) and BC4:OI (M3). Empirical results indicated that there was no clear best model with all three solutions less than 1% apart. M2 performed the best using a combination of simple random as the selection method and improve or equal as the move acceptance. The custom selection methods, BC3 and BC4, performed almost as well. To summarise, the findings did not meet the expectations laid out in Section 5.1, with no clear difference between each model type.

To conclude, it was found selection hyper-heuristics can effectively find feasible windfarm layouts with the combined optimisation shown to be a potential method for future windfarm design. However, it is not conclusive in determining whether sequential optimisation or simultaneous optimisation was better overall; further experiments are required to arrive at a decisive outcome. Therefore, one cannot reject or accept the null hypothesis defined in Section 5.1.

6.1. Study limitations

Whilst selection hyper-heuristics are relatively easy to implement, there were some limitations due to the scope of this paper. Firstly, the cabling data was modified for confidentiality reasons and subsequently is not reflective of the true cost. Additionally, turbine costs were estimated at 10 million however, these may differ in the real-world scenario. The initial decision to reduce the overall complexity of the model involved removing consideration of flexible cabling (non-straight lines), cable hang, ocean floor conditions or varying foundation costs at each site. Whilst reducing complexity for the purpose of this study, it also reduced accurate representation of the true situation.

The objective function used further limited the scope of the study insofar as it included the initial costs of the layout but did not take into account the long-term benefits of producing power, which could be sold. Introducing this factor would enable a more accurate reflection of the long-term costs and rewards of constructing the windfarm.

The range of low-level heuristics available was restrictive. The complexity of an electrical cabling layout, with many inputs, meant it was difficult to develop low-level heuristics to successfully manipulate some layouts of cabling running the risk of a worse optimisation overall.

Reflecting upon the work undertaken, the following areas are recommended as of potential research interest: (i) Consideration of more factors within the optimisation (foundation costs, flexible cabling, obstacles and various turbine capacities); (ii) Introduction of additional low-level heuristics that are capable of better modifying the cabling layout; and (iii) Potentially fix the number of turbines and modify the objective function to have a minimum expected power production, with the inclusion of a required power threshold for each site to be placed.

CRediT authorship contribution statement

Thomas Butterwick: Conceptualization, Methodology, Software, Writing – original draft. **Ahmed Kheiri:** Supervision, Conceptualization, Writing – review & editing, Methodology, Software. **Guglielmo**

Lulli: Supervision, Conceptualization, Writing – review & editing. **Joaquim Gromicho:** Supervision, Conceptualization, Software, Data curation. **Jasper Kreeft:** Supervision, Data curation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Fischetti, Mathematical programming models and algorithms for oshore wind park design, 2017.

[2] W. Li, E. Özcan, R. John, Multi-objective evolutionary algorithms and hyper-heuristics for wind farm layout optimisation, Renew. Energy 105 (2017) 473–482.

[3] D. Cazzaro, M. Fischetti, M. Fischetti, Heuristic algorithms for the Wind Farm Cable Routing problem, Appl. Energy 278 (2020) 115617.

[4] P. Fuglsang, K. Thomsen, Cost Optimization of Wind Turbines for Large Scale Off-Shore Wind Farms, Technical Report, Risoe National Laboratory, 1998.

[5] Y.-K. Wu, C.-Y. Lee, C.-R. Chen, K.-W. Hsu, H.-T. Tseng, Optimization of the wind turbine layout and transmission system planning for a large-scale offshore wind farm by AI technology, IEEE Trans. Ind. Appl. 50 (2014).

[6] P. Hou, W. Hu, M. Soltani, C. Chen, Z. Chen, Combined optimization for offshore wind turbine micro siting, Appl. Energy 189 (2017) 271–282.

[7] T. Marge, S. Lumbreras, A. Ramos, B.F. Hobbs, Integrated offshore wind farm design: Optimizing micro siting and cable layout simultaneously, Wind Energy 22 (2019) 1684–1698.

[8] P. Fagerfjäll, Optimizing Wind Farm Layout: More Bang for the Buck Using Mixed Integer Linear Programming, Chalmers University of Technology and Gothenburg University, 2010, p. 111.

[9] M. Fischetti, D. Pisinger, Mixed Integer Linear Programming for new trends in wind farm cable routing, Electron. Notes Discrete Math. 64 (2018) 115–124.

[10] S. Donovan, An improved mixed integer programming model for wind farm layout optimisation, in: Proceedings of the 41st Annual Conference of the Operational Research Society of New Zealand, Christchurch, New Zealand, 2006, pp. 143–152.

[11] B. Saavedra-Moreno, S. Salcedo-Sanz, A. Paniagua-Tineo, L. Prieto, A. Portilla-Figueras, Seeding evolutionary algorithms with heuristics for optimal wind turbines positioning in wind farms, Renew. Energy 36 (11) (2011) 2838–2844.

[12] M. Lerch, M. De-Prada-Gil, C. Molins, A metaheuristic optimization model for the inter-array layout planning of floating offshore wind farms, Int. J. Electr. Power Energy Syst. 131 (2021) 107128.

[13] J. Bauer, J. Lysgaard, The offshore wind farm array cable layout problem: a planar open vehicle routing problem, J. Oper. Res. Soc. 66 (3) (2015) 360–368.

[14] D. Wilson, S. Rodrigues, C. Segura, I. Loshchilov, F. Hutter, G.L. Buenfil, A. Kheiri, E. Keedwell, M. Ocampo-Pineda, E. Özcan, S.I.V. Pena, B. Goldman, S.B. Rionda, A. Hernandez-Aguirre, K. Veeramachaneni, S. Cussat-Blanc, Evolutionary computation for wind farm layout optimization, Renew. Energy 126 (2018) 681–691.

[15] P. Cowling, G. Kendall, E. Soubeiga, A hyperheuristic approach to scheduling a sales summit, in: International Conference on the Practice and Theory of Automated Timetabling, Springer, 2000, pp. 176–190.

[16] A. Kheiri, Heuristic sequence selection for inventory routing problem, Transp. Sci. 54 (2) (2020) 302–312.

[17] J.H. Drake, A. Kheiri, E. Özcan, E.K. Burke, Recent advances in selection hyper-heuristics, European J. Oper. Res. 285 (2) (2020) 405–428.

[18] G. Marmidis, S. Lazarou, E. Pyrgioti, Optimal placement of wind turbines in a wind park using Monte Carlo simulation, Renew. Energy 33 (7) (2008) 1455–1460.

[19] N. Jensen, A Note on Wind Generator Interaction, 1983, Risø-M, No.2411.

[20] I. Katic, Højstrup, N. Jensen, A simple model for cluster efficiency, in: European Wind Energy Association Conference and Exhibition, Rome - Italy, 1986.

[21] M. Bastankhah, F. Porté-Agel, A new analytical model for wind-turbine wakes, Renew. Energy 70 (2014) 116–123.

[22] A. Niayifar, F. Porté-Agel, Analytical modeling of wind farms: A new approach for power prediction, Energies 9 (741) (2016).

[23] A. Peña, P.-E. Réthoré, M. Van der Laan, On the application of the Jensen wake model using a turbulence-dependent wake decay coefficient: the Sexbierum case, Wind Energy 19 (2016).

[24] M. Bastankhah, B. Welch, L. Martínez-Tossas, J. King, P. Fleming, Analytical solution for the cumulative wake of wind turbines in wind farms, J. Fluid Mech. 911 (2021).

[25] G. Dueck, New optimization heuristics: The great deluge algorithm and the record-to-record travel, J. Comput. Phys. 104 (1) (1993) 86–92.