Aesthetic local search of wind farm layouts

Michael Mayo Corresp., [1] , **Maisa Daoud** [1]

[1] Department of Computer Science, University of Waikato, Hamilton, New Zealand

Corresponding Author: Michael Mayo
Email address: michael.mayo@waikato.ac.nz

The visual impact of wind farm layouts has seen little consideration in the literature on the wind farm layout optimisation problem to date. Most existing algorithms focus on optimising layouts for power or cost of energy alone. In this paper, we consider the geometry of wind farm layouts and whether it is possible to bi-optimise a layout for both energy efficiency and the degree of visual impact that the layout exhibits. We develop a novel optimisation approach for solving the problem, with our approach towards measuring mathematically the degree of visual impact drawing inspiration from the field of architecture. To evaluate our ideas, we demonstrate them on three benchmark problems for the wind farm layout optimisation problem in conjunction with two recently published stochastic local search algorithms. Optimal patterned layouts are shown to be very close in terms of energy efficiency to optimal non-patterned layouts.

Aesthetic Local Search of Wind Farm Layouts

Michael Mayo[1] **and Maisa Daoud**[2]

[1,2]**Dept. of Computer Science, University of Waikato, Hamilton, New Zealand**

Corresponding author:
Michael Mayo[1]

Email address: michael.mayo@waikato.ac.nz; Tel.: +64-7-8384403

ABSTRACT

The visual impact of wind farm layouts has seen little consideration in the literature on the wind farm layout optimisation problem to date. Most existing algorithms focus on optimising layouts for power or cost of energy alone. In this paper, we consider the geometry of wind farm layouts and whether it is possible to bi-optimise a layout for both energy efficiency and the degree of visual impact that the layout exhibits. We develop a novel optimisation approach for solving the problem, with our approach towards measuring mathematically the degree of visual impact drawing inspiration from the field of architecture. To evaluate our ideas, we demonstrate them on three benchmark problems for the wind farm layout optimisation problem in conjunction with two recently published stochastic local search algorithms. Optimal patterned layouts are shown to be very close in terms of energy efficiency to optimal non-patterned layouts.

1 INTRODUCTION

Worldwide, renewable energy production via wind is becoming increasingly important. In particular, it has been forecast that by 2030 approximately 18% of the planet's total energy production will be sourced from wind farms (Global Wind Energy Council, 2014). The rapid growth in wind energy production is well-illustrated by examining some of the current and planned wind farms installations around the world: the London Array [1], for example, generates 630MW of power (enough for 490,000 households); similarly, the ongoing Gansu project in China (Watts, 2012) is planned to generate 20GW by 2020 (equating to power for approximately 15 million households). Concerns about the environmental impact of wind energy have also been increasing alongside growth in its production. In particular, it has been noted for some time that wind turbines have considerable impact on local wildlife populations such as birds, bats, and for offshore farms, various marine wildlife (Dai et al., 2015). So serious is the problem that there have been recent calls for entirely new research programs to be developed solely to study the effects of wind generation on wildlife and how to mitigate them (Piorkowski et al., 2012). The human impact of wind energy production is also considerable: wind farms tend to generate significant noise, and they have a major visual impact on the landscape due to the size of the turbines. On clear days turbines can be seen up to 30 kilometers away depending on the turbine height and terrain conditions (Dai et al., 2015). Moreover, the human impact can range in severity from a "mere" belief (strong or otherwise) that the wind turbines detract from the visual value of the landscape, to shadow flicker, a phenomenon caused by the interaction of a wind turbine's blades with direct sunlight. Such a phenomenon is known to cause severe headaches when nearby residents are exposed to it for a long period (Tabassum-Abbasi et al., 2014).

This paper concerns itself with one aspect of the environmental impact of wind farm design, specifically, the arrangement of wind turbines into geometrical patterns, and the relationship of these geometric patterns with the overall energy efficiency of the farm.

To partially mitigate the negative visual impacts of wind farms, it has been noted that farms with a regular layout of turbines tend to be perceived as blending into the visual landscape in a better way than farms with an irregular layout (Dai et al., 2015). More precisely, research by (Tsoutsos et al., 2006) discusses the aesthetic principles of wind farm layout design: farms have a higher aesthetic appeal either

[1] http://www.londonarray.com/wp-content/uploads/London-Array-Brochure.pdf

when turbines are arranged clearly into rows, or when they are arranged into uniform density small clusters of 2-8 turbines which are separated by obvious landmarks. The latter arrangement is particularly preferred when the wind farm must be integrated with agriculture.

One significant issue arising when considering the layout of turbines on a wind farm is the loss of wind energy due to the interaction between nearby turbines, a phenomenon known as the "wake effect". Therefore, all configurations of turbines in a wind farm are not equal, and often a computationally-expensive simulation (or an approximation thereof) is required to assess the wake effect so that it can be mitigated as much as possible. Sometimes other objectives may be also be considered (e.g. construction costs) but the commonality amongst many papers in the literature is a focus on wake effect minimisation. Examples include the seminal work in the field by Mosetti et al. Mosetti et al. (1994) as well more recent works such as that by Wagner et al. (2013), Rodrigues et al. (2013), Guirguis et al. (2016), Mayo and Zhen (2016); and Mayo and Daoud (2016).

In general there is a trade-off between the geometric constraints required to minimise the negative visual impact of a wind farm, and the energy output of the farm itself. A simple example would be the arrangement of turbines equidistantly around the perimeter of a circle: although a circle is visually interesting shape, there will always be a sizeable portion of turbines (on opposite sides of the circle) that lie in each other's wakes regardless of the predominant wind direction.

Beyond forcing turbines to be arranged into simple geometric shapes such as circles and grids, it is not immediately clear how to define "visually appealing" arrangements in a more general way. Most of the past and current works on wind farm layout optimisation (WFLO), therefore, tend to ignore geometric appearances and consequently highly optimised layouts may taken on a "random scattering" appearance. This is shown in some of the figures later in this paper.

Very few authors have considered the visual aspects of layouts as part of the optimisation process. Two works exist as far as the authors are aware: an excellent paper by Neubert et al. (2010) in which turbines are constrained to a skewed grid, and the orientation and skew of the grid is optimised; and an approach by Al-Yahyai et al. (2015), in which turbines are also assigned positions on a grid but in this case only the grid's orientation is optimised.

In both cases, the geometric constraints are extreme and therefore the optimisation problem can be solved by varying only two or one variables respectively. Despite the limitations of these approaches, Neubert et al. show that the geometrically constrained layouts are almost (within a few percentage points) as efficient as completely unconstrained layouts that are optimised purely for energy efficiency.

In this paper, we take a different tack. Rather than strongly constraining layouts in order to force a geometric pattern on them (as the previous authors have done), we instead add a pattern-based metric to the optimiser that assesses the quality of the pattern that the turbines form. Thus, each individual turbine's position on the layout is still a degree of freedom, but at the same time, a poorly-arranged layout with the same energy efficiency as a well-arranged layout will score an overall worse objective value. Thus the optimiser should focus its search towards layouts with either minimal visual impact (or, alternatively, a strong aesthetic appeal).

To evaluate our novel approach, we use two stochastic local search algorithms for the wind farm layout optimisation problem that have recently appeared in the literature. Both approaches are combined with our novel objective function, thus producing two new approaches. The first existing approach we utilise is called the Turbine Displacement Algorithm (TDA) (Wagner et al., 2013). The second approach is our own recently published approach known as BlockCopy (Mayo and Zhen, 2016; Mayo et al., 2016).

In order to assess the geometric/pattern quality of layouts, we have utilised a pattern measure originally proposed by Salingaros (1997) and Klinger and Salingaros (2000). This metric has its origin in the architectural evaluation of building facades, but has been generalised for the evaluation of any kind of symbolic pattern. Therefore it is ideally suited for our purposes.

The pattern metric we use as well as all other relevant technical details are described in the next section. Following that, we describe how we modify the objective function of the TDA and BlockCopy algorithms to optimise for geometric qualities. Sections 4 and 5 describe a comprehensive evaluation of both algorithms that was performed, and finally Section 6 concludes the paper.

We acknowledge at this point that the work undertaken here is largely focussed on the aesthetics of two dimensional layouts. Thus, it would be applicable to situations where the wind farm is located off-shore or on a plain, but not in a situation where the farm is located on a three dimensional terrain (e.g. along a ridge). Furthermore, adjustments would also have to be made to the proposed method if

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017

the importance of aesthetics varies across the layout. For example, an area of the farm close to a tourist attraction is likely to have much more visual impact than the part of the farm furtherest from the attraction. We address these concerns in the conclusion.

2 TECHNICAL BACKGROUND

2.1 Jensen Wake Model

The Jensen far wake model, originally proposed in the mid-1980s (Jensen, 1983; Katic et al., 1986), is the approach we use in this research to assess the wake interactions between turbines in a wind farm layout. Although dated, the Jensen is still used widely in the community. To illustrate, Samorani (2013) describes it precisely in a recent 2013 introductory survey to the Wind Farm Layout Optimisation (WFLO) problem, and in a 2016 comparison of three kinematic far wake models and two field-based far wake models, Shakoor et al. (2016) concluded that "...Jensen's far wake model is a good choice to solve the wind farm layout optimisation problem due to its simplicity and relatively high degree of accuracy." We therefore adopt the Jensen model for our initial investigations reported here, while acknowledging that more sophisticated models do exist that we will explore in future work.

In this section we therefore describe briefly the Jensen far wake model. As we are using, more or less, the same notation as Samorani (2013), the interested reader is referred to that publication for more specific details.

The first element required in wind farm modeling is a power curve, which describes the relationship between incoming wind speed and the power generated by a single wind turbine. This is generally dependent on the type of wind turbine being modelled and therefore will vary depending on manufacturer and model. In general, however, the relationship can be modelled as a cubic function from wind speed to power between two bounding wind speeds: (i) the cut in speed, which is the wind speed at which the turbine begins generating power; and (ii) the nominal speed, which is the wind speed at which maximum power production is reached. A final element of the power curve is the cut out speed. This is the maximum allowable wind speed that the turbine can tolerate before shutting down to avoid damage.

Due to the variability and manufacturer-dependence of different wind turbine models, we adopt in this paper the power curve used by Mosetti et al. (1994) and also described by Samorani (2013):

$$power(u) = \begin{cases} 0\text{kw} & \text{where } u < 2\text{m/s} \\ 0.3u^3\text{kw} & \text{where } 2\text{m/s} \leq u < 12.8\text{m/s} \\ 629.1\text{kw} & \text{where } 12.8\text{m/s} \leq u < 18\text{m/s} \\ 0\text{kw} & \text{where } u > 18\text{m/s} \end{cases} \quad (1)$$

In this power curve, wind speed u is measured in metres per second (m/s) and power in kilowatts (kw). We acknowledge here that this turbine model is somewhat dated (for example, modern turbines may produce 8-10MW of power), but it is a model that is frequently used in the literature and therefore we adopt it in this paper for the purposes of reproducibility ease.

The next part of the Jensen far wake model is the modelling of the velocity deficit, i.e. the reduction in wind speed as wind passes through the blades of a turbine. This is best introduced schematically, and here we reproduce a diagram from Samorani (2013) in Figure 1.

The key elements of Figure 1 are an illustration of the initial wind speed u_0; the reduced wind speed u_j at a distance x metres from the wind turbine; and the notion that the wake spreads out with linearly increasing radius as it gets further away from the turbine.

In fact, the radius of the spreading wake is modelled by the following equation:

$$r_1 = \alpha x + r_r \quad (2)$$

where r_r is the turbine's radius and r_1 is the radius of the wake. This equation shows that the rate of spreading is determined by a constant α which in turn depends on two further factors: first, the height of the turbine z; and second, the surface roughness, z_0. The function to calculate α is:

$$\alpha = \frac{0.5}{\ln \frac{z}{z_0}} \quad (3)$$

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017

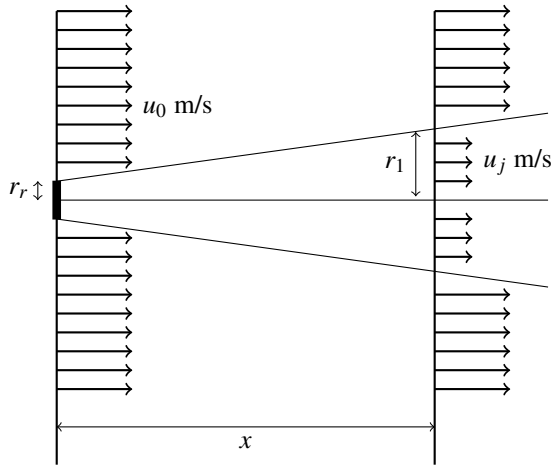


Figure 1. Depiction of the wake effect (reproduced from Samorani (2013)).

Once the size of the spreading wake can be determined as a function of distance x, the degree of wind speed velocity deficit needs to be next computed. The relationship between wind speed and velocity deficit is modelled as

$$u_j = u_0(1 - vd_{ij}) \tag{4}$$

where u_j is the wind speed at a position j which is inside the wake of a turbine at position i. The term vd_{ij} represents the velocity deficit between positions i and j due to the turbine, and is calculated (according to Samorani (2013)) thus:

$$vd_{ij} = \frac{2a}{1 + \alpha(\frac{x_{ij}}{r_d})^2} \tag{5}$$

where x_{ij} is the distance between the two points and a, the "axial induction factor" is defined as

$$a = 0.5(1 - \sqrt{1 - C_T}) \tag{6}$$

while r_d, the "downstream wake radius" is calculated as

$$r_d = r_r(\sqrt{\frac{1-a}{1-2a}}) \tag{7}$$

This value r_d is used as the input to the velocity deficit calculation (Equation 5 above) as per the model.

A key corollary of these equations is that while power increases with the cubic of wind speed, velocity deficit decreases at a rate proportional to the square of the distance from a turbine. Therefore it follows that simply finding a "windier" site should increase power production regardless of whether wake effects are minimised or not.

Next, the model also accounts for the fact that a turbine may be in the wake of not one, but many other wind turbines, at the same time. In this case, the velocity deficit calculation is more complex because the different velocity deficits must all be aggregated and subtracted from the incoming wind speed together. This is achieved in the Jensen model by calculating the square root of the sum of the squared velocity deficits:

$$v^s_{def}(j) = \sqrt{\sum_{i \in W^s(j)} vd_{ij}^2} \tag{8}$$

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017

where $v^s_{def}(j)$ is the total velocity deficit and $W^s(j)$ is the set of turbines affecting the turbine at position j. Turbines are represented as points for the purposes of this set membership calculation and therefore they are either completely inside or outside the wake; we acknowledge that other wake models allow partial wake overlaps of the rotors, but we have followed the Jensen model rigorously in this work which treats turbines as points. As pointed out earlier by Shakoor et al. (2016), this should be sufficiently accurate.

The index s denotes the *wind scenario* which specifies both the wind direction and the initial wind speed: both of these factors determine the wakes in which a turbine lies and therefore what the total velocity deficit will be.

Two things must be noted about the calculation of the set $W^s(j)$. Firstly, computing the wakes within which a turbine j lies for any arbitrary wind direction requires some non-trivial 2D geometric calculations. This is because the wind may blow in any direction, and therefore wakes may expand in any direction. However, this calculation is readily computable with some standard trigonometry.

Secondly, and much more significantly, a routine to calculate the velocity deficit for every turbine in a layout for a single wind direction is a function with quadratic complexity: this is because every turbine j in the layout must be compared to every other turbine in order to determine $W^s(j)$. Thus, layout evaluators can face scalability issues as the size of the layout increases.

Finally, to complete our presentation of the Jensen model, there are several constants required. Specifically, these are r_r, the turbine radius, which we set to 20 meters; z, the hub height, which is initialised to 60 meters; z_0, the surface roughness constant, which is 0.3 meters; and C_T, which is 0.88. A minimum allowable distance between turbines must also be specified, because the Jensen model is not accurate at close distances. We set this constant to three times the diameter of the rotors, namely 120 meters. These constants are all as-used by Samorani (2013). We expect that other situations will require different unique values for the above parameters since they depend on the model of wind turbine being used as well as characteristics of the site. However, these values are good defaults for the purposes of reproducibility, and we don't expect the behavior of the approaches that we present later to be significantly dependent on particular choices of values.

2.2 Objective Function

Once the Jensen model is completely specified, the next step is to precisely specify the objective function for the WFLO problem. There are generally many different ways of doing this depending on what optimisation is required. One approach is to simply calculate the total expected power generated by a wind farm, which must be maximised (e.g. Song et al. (2016)). Another approach is to calculate the expected cost of energy: take the total expected power, convert it into units of currency that would be obtained if the power were sold at market, and divide that revenue by the cost of building and maintaining the wind farm (e.g. Mayo and Daoud (2016)). This is an objective that must be minimised.

For this preliminary assessment of our new approach, we use a simple objective function that divides the total expected power generated by the farm with wake interference by the total *hypothetical* expected power that would be generated by the farm without wakes. Clearly, this ratio should result in a value between 0 and 1, with higher values being more desirable.

If $\mathbf{l}$ is a wind farm layout, the objective function therefore is:

$$F(\mathbf{l}) = \sum_{s \in S} r_s \frac{\sum_{j \in \mathbf{l}} power\left(u_s(1 - v^s_{def}(j))\right)}{\sum_{j \in \mathbf{l}} power(u_s)} \tag{9}$$

where j is a turbine's position in the layout, and $v^s_{def}(j)$ is the total velocity deficit at j. S is a set of wind scenarios, r_s is the probability of scenario $s \in S$ and u_s is the wind speed under scenario s. It should be evident that $\sum_{s \in S} r_s = 1.0$ in order compute proper expected power values.

2.3 Turbine Displacement Algorithm

The Turbine Displacement Algorithm (TDA) is a highly effective stochastic local search algorithm for the wind farm layout optimisation problem first introduced by Wagner et al. (2013). The algorithm shifts a single turbine at a time, and then re-evaluates the layout to determine if the turbine move should be accepted or not.

Although initially designed to be used in conjunction with a specific wake model in order to reduce the computational complexity of layout evaluation, the algorithm is in fact competitive with many other

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017

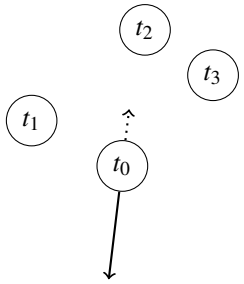


Figure 2. Illustration of the TDA operator. In this example, $K = 3$ and the displacement vector for t_0 (and its potential inverted displacement vector after rescaling to prevent collisions) is shown.

approaches that use different wake models. A recent evaluation by Wilson et al. (2014) showed that TDA outperformed several other metaheuristic algorithms including genetic algorithms and particle swarm optimisation. We therefore use TDA as one of the algorithms in our evaluation.

The basic behaviour of one iteration of TDA is shown in Figure 2. Essentially, a neighbourhood size K must be specified initially by the user. A random turbine is then picked, and it is moved either away from the K neighbours or, with reduced probability, towards the K neighbours. The direction that the turbine moves is called its displacement vector.

In the original paper on TDA (Wagner et al., 2013) a study on the best value of K for different layout sizes was performed. It was found that for very small layout, a small K (e.g. $K = 1$) lead to an efficiency gain of just over 1%; however, as the layout size increased, the difference in efficiency caused by varying K approached a negligible value. $K = 8$ was the highest value tested in that paper.

Each displacement vector has a specific size, and one feature of the algorithm is that the size of the displacement vectors is not constant. Instead, it varies on a per-turbine basis: if a turbine's moves are frequently accepted (i.e. lead to improvements in objective value) then the size of the displacement vectors is gradually increased; conversely if a turbine's moves are not accepted, the size decreases.

A complete specification of TDA can be found in Wagner et al. (2013), and it suffices to state the parameters that we used: the best neighbourhood size we found in initial experiments was $K = 8$; the initial displacement vector size was set to 120 meters; the scaling factor for reducing displacement vector sizes was 0.9, and conversely the factor for increasing sizes was $\frac{1}{0.9}$; and finally the amount of "distance noise" added to the displacement vectors was set to 40 meters. All other parameters and properties of TDA are the same as reported in Wagner et al. (2013).

2.4 BlockCopy Local Search Algorithm

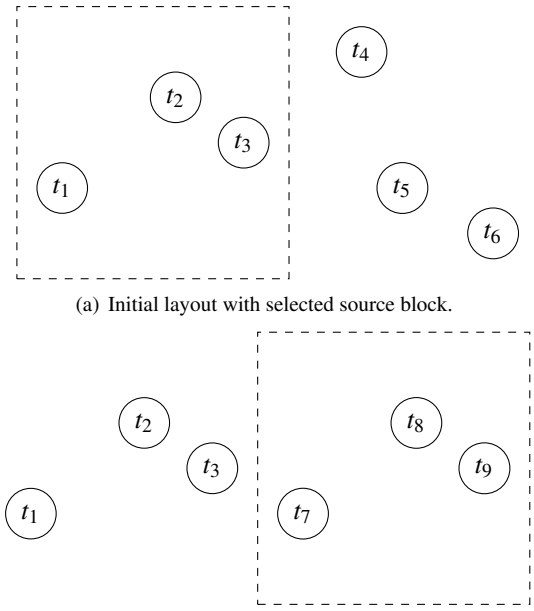
In comparison to TDA which moves one turbine at a time, the BlockCopy local search algorithm described in Mayo and Zhen (2016) and further extended in Mayo et al. (2016) operates by copying entire groups of turbines at a time. This operation is illustrated in Figure 3. The basic idea to replace one random square region (a "block") of a wind farm layout with copy of another square region. Any turbines in the destination region before the copy occurs are deleted. After the copy, if the total number of turbines in the layout has either increased or decreased (because of differences in the number of turbines per block), then turbines are either randomly added or randomly purged in order to keep the total number of turbines in the layout fixed.

An advantage of this approach is that the relative configuration of turbines is maintained whenever a block is copied, and therefore if a particularly good configuration of turbines (for the conditions) is present in the layout, then this configuration will quickly replicate itself across the layout via successive BlockCopy operations.

In the initial evaluation of the algorithm that was recently published (Mayo and Zhen, 2016), the algorithm was shown to outperform TDA on a set of benchmark problems using a cost-based objective function and a different far wake model than Jensen. However, the number of iterations of each algorithm in that paper was only 2,000. In this paper, we give both algorithms ten times as many iterations, which should make TDA more competitive.

BlockCopy is the second layout optimiser used in this paper, and the single main parameter for the algorithm (i.e. the block size) is set in this research to 250 meters $\times$ 250 meters. The block sizes and

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017



(a) Initial layout with selected source block.

(b) Final layout showing new target block.

Figure 3. Illustration of the BlockCopy operator. In this example, the left block of a small layout is duplicated to the right hand side of the layout.

positions are fixed, non-overlapping, and completely exhaustive across the layout.

2.5 Harmony Pattern Metric

In order to assess the visual elegance of wind farm layouts, we selected an aesthetic pattern metric called harmony first proposed by Salingaros (1997) as a method of assessing the aesthetics of building designs. Subsequently the approach was generalised so that it could be applied to any type of pattern, as long as the pattern could be represented by an array of discrete symbols (Klinger and Salingaros, 2000).

One motivation for selecting this metric over some of the more recent methods from the field of computational aesthetics (den Heijer and Eiben, 2010; Galanter, 2012) is that harmony does not expect patterns to be derived from images. Other metrics frequently assume they are being used for image assessment and therefore rely on the calculation of quantities such as compression ratios, or statistics related to image properties such as colour, which make them difficult to apply to non-image patterns.

The harmony metric can be described as follows in the remainder of this section. We use a slightly more succinct notation than that presented in the original paper, mainly for improved clarity.

Firstly, the pattern must be represented as a rectangular (preferably square) array of symbols, where each symbol corresponds to one basic constituent of the pattern For example, in architecture, one element might correspond to a curved corner and another to a window. In our approach, we make use of only two symbols (1 or 0) which correspond to the presence or absence of a turbine in a particular small region in the layout. Examples of some small symbol arrays, two of which are from Klinger and Salingaros (2000) are shown in Figure 4. Note that in the figure, the entries $\{0,1,2,3\}$ denote symbols and H denotes the harmony metric which is defined next.

Once the symbol array is available, the harmony metric is computed by first of all evaluating a number of functions on the symbol array. Each function concerns one particular class of symmetry, either reflective, rotational, or in relation to another pattern. The functions (nine of them) are listed in Table 1, and each returns either 1 or 0 depending on whether the pattern has the particular class of symmetry that the function is concerned with.

Before showing how the overall harmony of a pattern is computed, we must first define the value $h(\mathbf{a},B)$. This quantity, where $\mathbf{a}$ is a pattern and B is a set of different patterns of the same size, is defined as follows:

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017

2	2	2	2	2	2
2	2	2	2	2	2
2	2	2	2	2	2
2	2	2	2	2	2
2	2	2	2	2	2
2	2	2	2	2	2

(a) $H = 8$

2	2	2	2	2	2
2	0	0	0	0	2
2	0	1	3	0	2
2	0	3	1	0	2
2	0	0	0	0	2
2	2	2	2	2	2

(b) $H = 2.67$

0	0	0	0	1	0
1	1	1	0	1	0
0	0	0	0	1	0
0	0	0	0	1	0
1	1	1	0	1	0
0	0	0	0	1	0

(c) $H = 3.89$

Figure 4. Examples of three 6×6 patterns and their harmonies, computed using levels $N = \{6, 3, 2\}$.

Harmony	Description
h_1	Symmetry about the x axis.
h_2	Symmetry about the y axis.
h_3	Symmetry about the $y = x$ diagonal.
h_4	Symmetry about the $y = -x$ diagonal.
h_5	$\pm 90^\circ$ rotational symmetry.
h_6	180° rotational symmetry.
h_7	Translational symmetry with another pattern.
h_8	Translation plus reflectional symmetry with another pattern.
h_9	Translation plus rotational ($\pm 90^\circ$ or 180°) symmetry with another pattern.

Table 1. The six possible internal symmetries and the three additional hierarchical symmetries required to compute the harmony metric. Each h value is either 1 or 0.

$$h(\mathbf{a}, B) = \sum_{i=1}^{6} h_i(\mathbf{a}) + \sum_{i=7}^{9} h_i(\mathbf{a}, B) \tag{10}$$

The functions h_1–h_6 measure top-level properties of the pattern. The functions h_7–h_9 measure properties of the pattern in relation to all of the patterns in B. For these three latter functions, if $\mathbf{a}$ matches any of the elements in B, then 1 is returned; otherwise (or if B is empty) 0 is returned. Therefore $h(\mathbf{a}, B)$ must return a value either between 0 and 9 if B is non-empty, or 0 and 6 if B is empty.

We now define $B_{n\times n}(\mathbf{a})$ to be the set of all non-overlapping $n \times n$ sub-patterns of $\mathbf{a}$ that can be obtained by dividing $\mathbf{a}$ into $n \times n$-sized subarrays. It can be seen that if the size of $\mathbf{a}$ is 6×6 then $B_{3\times 3}$ will have four distinct elements, $B_{2\times 2}$ will have nine elements, and $B_{6\times 6}$ will have one element. If the size of $\mathbf{a}$ is 36×36, on the other hand, then $|B_{6\times 6}| = 36$.

The final harmony for a pattern $\mathbf{a}$ can thus be defined as:

$$H(\mathbf{a}) = \frac{1}{|N|(\mathbf{a})|} \sum_{n \in N} \left[\frac{1}{|B_{n\times n}(\mathbf{a})|} \sum_{\mathbf{b} \in B_{n\times n}(\mathbf{a})} h(\mathbf{b}, B_{n\times n}(\mathbf{a}) \setminus \mathbf{b}) \right] \tag{11}$$

where $H(\mathbf{a})$, being an average across values computed by the h function, is also in the range 0 to 9 inclusive. In the definition of H, the set N consists of positive integers which are not greater than (and preferably divide evenly into) the smallest dimensionality of $\mathbf{a}$. The elements of N define the sizes of the sub-patterns to consider. In the original paper (Klinger and Salingaros, 2000), $N = \{6, 3, 2\}$ and we use the same values, although we do increase the sizes of the patterns being considered from 6×6 to 36×36.

A brief consideration of Equation 11 should make clear fact that what is being computed is the average of the harmonies of each sub-pattern at the various different scales defined by N. This is represented by the inner summation, with each individual $n \times n$ sub-pattern being represented by $\mathbf{b}$ in the equation.

Then the average harmony across scales, represented by the outer summation and division by $|N|$, is calculated. The resulting quantity is the final metric. Thus, a higher harmony indicates that the pattern contains more symmetries at the various scales, while a lower harmony indicates fewer multi-scale symmetries.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017

Problem	Direction(s)	Expected Speed(s)	#Wind Scenarios
A	$\{0°\}$	{12m/s}	1
B	$\{0°, 10°, \ldots, 350°\}$	{12m/s}	36
C	$\{0°, 10°, \ldots, 350°\}$	{8m/s,12m/s,17m/s}	108

Table 2. Problems from Samorani Samorani (2013).

3 OPTIMISING WIND FARM LAYOUTS FOR BOTH ENERGY EFFICIENCY AND HARMONY

We define in this section two new approaches that extend both TDA and the BlockCopy local search method with the harmony metric explained in the previous section. We dub these new approaches TDA* and BC*.

The basis is fairly straightforward. Rather than optimising directly for energy efficiency (i.e. maximising F only, which is defined by Equation 9), we instead replace F with a composite objective function F' obtained by adding F and H. This new objective function is defined as follows:

$$F'(\mathbf{l}) = F(\mathbf{l}) + \lambda H(pattern(\mathbf{l})) \tag{12}$$

Two key parts of the new approach are (i) a function $pattern()$ that converts a layout (consisting of real-valued double coordinates) into a symbolic pattern array, and (ii) a parameter λ that specifies how much influence harmony will have in objective calculations.

For the $pattern()$ function, we simply divide the layout (which is square in our experiments) into 36×36 "cells". Each cell maps to a symbol in a 36×36 symbol array representing the layout, and corresponds to the number of turbines in that cell. As it turns out, on our test layout, the cells were relatively small and so they only ever contain either one turbine or no turbines due to the minimum turbine distance constraint.

The choice of value for λ was a more difficult decision, however, and we therefore decided to test four different values: 0, corresponding to harmony having no influence on the optimisation process; 0.001 and 0.01, corresponding to harmony having small to medium effects on the objective function; and 0.1, corresponding to harmony being nearly equally weighted with energy efficiency. The choice of these values was made because the range of the H parameter (0..9) is nine times the range of the F parameter (0..1) and thus small values of λ make sense. Any larger values of λ would result in harmony overwhelming the combined objective function.

The case of $\lambda = 0$ is thus our baseline because it essentially reverts the TDA* and BC* approaches back into their original versions.

4 EXPERIMENTS

To evaluate our new modified objective functions, we implemented Jensen's far wake model and used it to simulate a wind farm layout of size 1.5 kilometers $\times$1.5 kilometers, with 64 turbines to be sited. The turbine power curve and the other settings for the wake model are described in Section 2.1. The optimisers used are the algorithms described in the previous sections. Each algorithm was initialised with a starting random layout created by iteratively adding turbines at random locations within the layout bounds, subject to the constraint of not placing any two turbines too closely together, until all 64 turbines were placed.

Samorani (2013) describes three different problems of increasing complexity for testing wind farm layout optimization algorithms, and we adopted these three benchmark problems for our experiments. The problems, referred to as A, B and C, are described by Table 2.

Essentially, Problem A is the simplest benchmark, and comprises a single wind scenario in which wind blows with a single expected speed and in a single constant direction. The set of scenarios S therefore (used in Equation 9 for calculating F) consists of only a single element. Problem B, on the other hand, consists of 36 different wind scenarios, each differing only in the wind direction. Unlike Problem A, there is no dominant wind direction: instead , the expected wind speed is the same for all directions. Although this is an unrealistic setting, it is useful for testing purposes.

Problem C, on the other hand, is the most interesting and challenging benchmark. In Problem C, as in Problem B, there are 36 possible wind directions. In Problem C's case, however, for each different

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017

Direction	$u_s = 8$m/s	$u_s = 12$m/s	$u_s = 17$m/s
0°-260°	0.00404	0.00865	0.0115
270°	0.00404	0.0107	0.0127
280°	0.00404	0.0121	0.0156
290°	0.00404	0.0141	0.0185
300°	0.00404	0.0138	0.0300
310°	0.00404	0.0190	0.0352
320°	0.00404	0.0138	0.0300
330°	0.00404	0.0141	0.0185
340°	0.00404	0.0121	0.0156
350°	0.00404	0.0107	0.0127

Table 3. Probabilities used for the 108 wind scenarios under Problem C (rounded to three significant figures) derived from a chart in Samorani (2013). Please note that the first row the table, labelled "0°-260°", represents 27 rows, each of which have the same values. We have written these values once only to prevent unnecessary duplication in the table.

direction, there are also three different expected wind speeds which evidently correspond to three different meteorological conditions. Furthermore, there is a clear dominant wind direction: the probability of the greatest wind speed (and therefore the greatest power production) is maximised at 310°. In total, this problem comprises 108 different wind scenarios.

Since Samorani (2013) only describes this problem benchmark graphically by means of a histogram of wind speeds vs. directions, in order to implement this benchmark, we reverse-engineered the probabilities from his published chart. The probabilities we used for Problem C are given in Table 3.

To summarise, our experiments consist of algorithms with a new modified objective function (TDA* and BC*) with four different λ values on three different benchmark problems. This amounts to 24 different configurations.

Next, for each configuration, we ran 30 repetitions. Each repetition consisted of one run of a local search algorithm for 20,000 iterations. We note that the number of iterations is significantly higher than the number of iterations (2,000) in a previous comparison of TDA and BlockCopy (Mayo and Zhen, 2016).

5 RESULTS

The summary results are depicted in Figures 5–7. The figures show the mean and maximum F and H values achieved by algorithm over all thirty runs. We note in this section that our use of the term "performance" refers to the best objective values achieved by the various algorithms and not, as is the common interpretation, to computational efficiency.

Broadly speaking, then, the figures show that BlockCopy local search "outperforms" TDA in terms of energy efficiency when λ is small. The difference between the algorithms for Problem A is approximately up to 10%; for Problems B and C the difference is less pronounced.

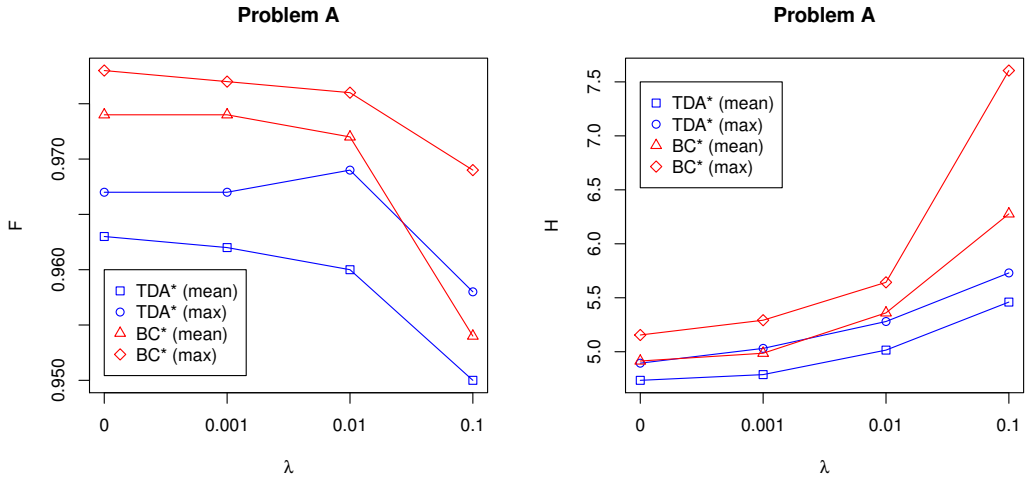
In terms of the harmony metric, however, the BlockCopy approach generally scores a much higher value than TDA, especially as λ increases. This is most likely due to the approximate preservation of translational symmetry, a property of the BlockCopy operator discussed earlier.

Overall, under both algorithms, the best layouts degrade in terms of energy efficiency as λ increases. This energy loss is approximately 6% on average for Problem C. Analysing the final best layouts produced by every 30-run configuration, we found that there is a significant negative correlation between final F and H values. This confirms that the two objectives, energy efficiency and harmony, tend to trade off. Testing more advanced multi-objective algorithms may therefore be worthwhile in the future.

Next, we were also interested in the statistically significant differences in performances for a more quantitative comparison.

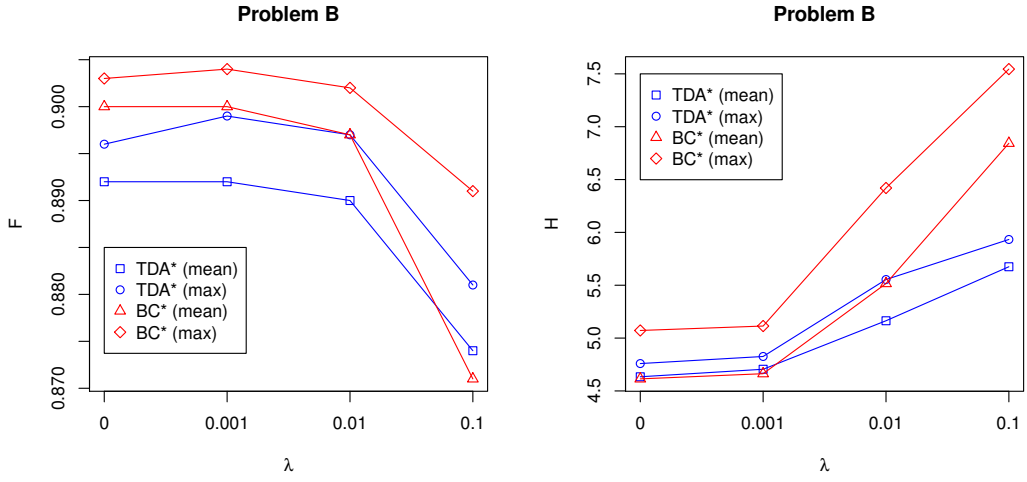
To this end, we performed a post-hoc Tukey Honest Significant Difference test comparing F value mean between all eight techniques on each of the three different benchmark problems. The test was performed at 95% significance. The results indicate that for Problems A and B, there is no significant difference in mean performance between algorithms in the following sets:

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017



(a) (b)

Figure 5. Optimisation performance over 30 runs with different λ values on Problem A



(a) (b)

Figure 6. Optimisation performance over 30 runs with different λ values on Problem B

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017

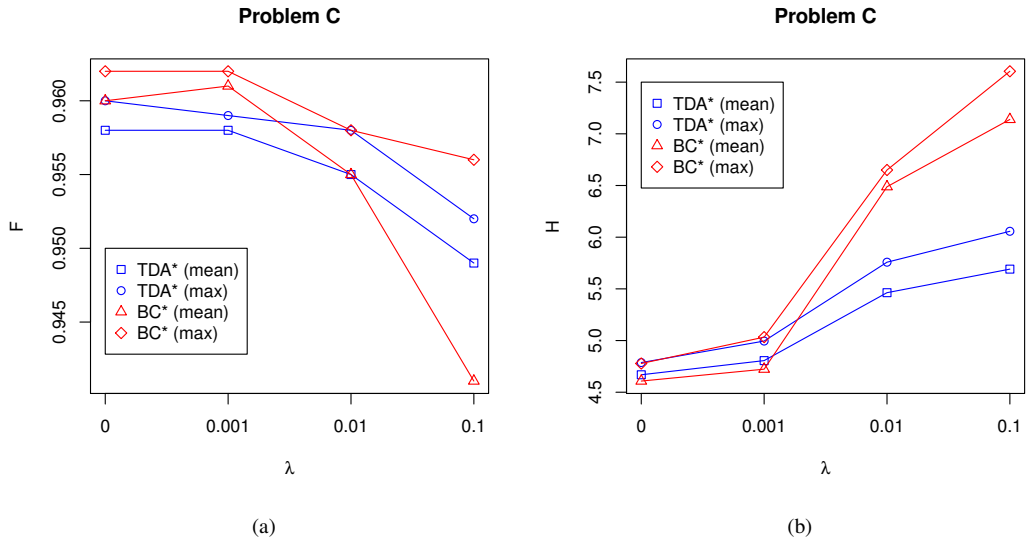


Figure 7. Optimisation performance over 30 runs with different λ values on Problem C

- $\{(\text{TDA}^*, \lambda = 0), (\text{TDA}^*, \lambda = 0.001), (\text{TDA}^*, \lambda = 0.01)\}$
- $\{(\text{BC}^*, \lambda = 0), (\text{BC}^*, \lambda = 0.001), (\text{BC}^*, \lambda = 0.01)\}$
- $\{(\text{TDA}^*, \lambda = 0.1), (\text{BC}^*, \lambda = 0.1)\}$ (Problem B only)

The result of these tests show that small to moderate values for λ do not significantly impact on optimisation performance. The figures quantify the actual difference in mean performances between algorithms.

For Problem C, the situation is a slightly more complex. The sets of algorithms with no statistically significant difference in mean performance are:

- $\{(\text{BC}^*, \lambda = 0.001), (\text{BC}^*, \lambda = 0), (\text{TDA}^*, \lambda = 0.001)\}$
- $\{(\text{TDA}^*, \lambda = 0.001), (\text{TDA}^*, \lambda = 0), (\text{BC}^*, \lambda = 0.01)\}$
- $\{(\text{TDA}^*, \lambda = 0), (\text{BC}^*, \lambda = 0.01), (\text{TDA}^*, \lambda = 0.01)\}$
- $\{(\text{TDA}^*, \lambda = 0.1)\}$
- $\{(\text{BC}^*, \lambda = 0.1)\}$

These results clearly are more difficult to interpret, which suggests that no conclusion can be drawn without more repetitions of the algorithms.

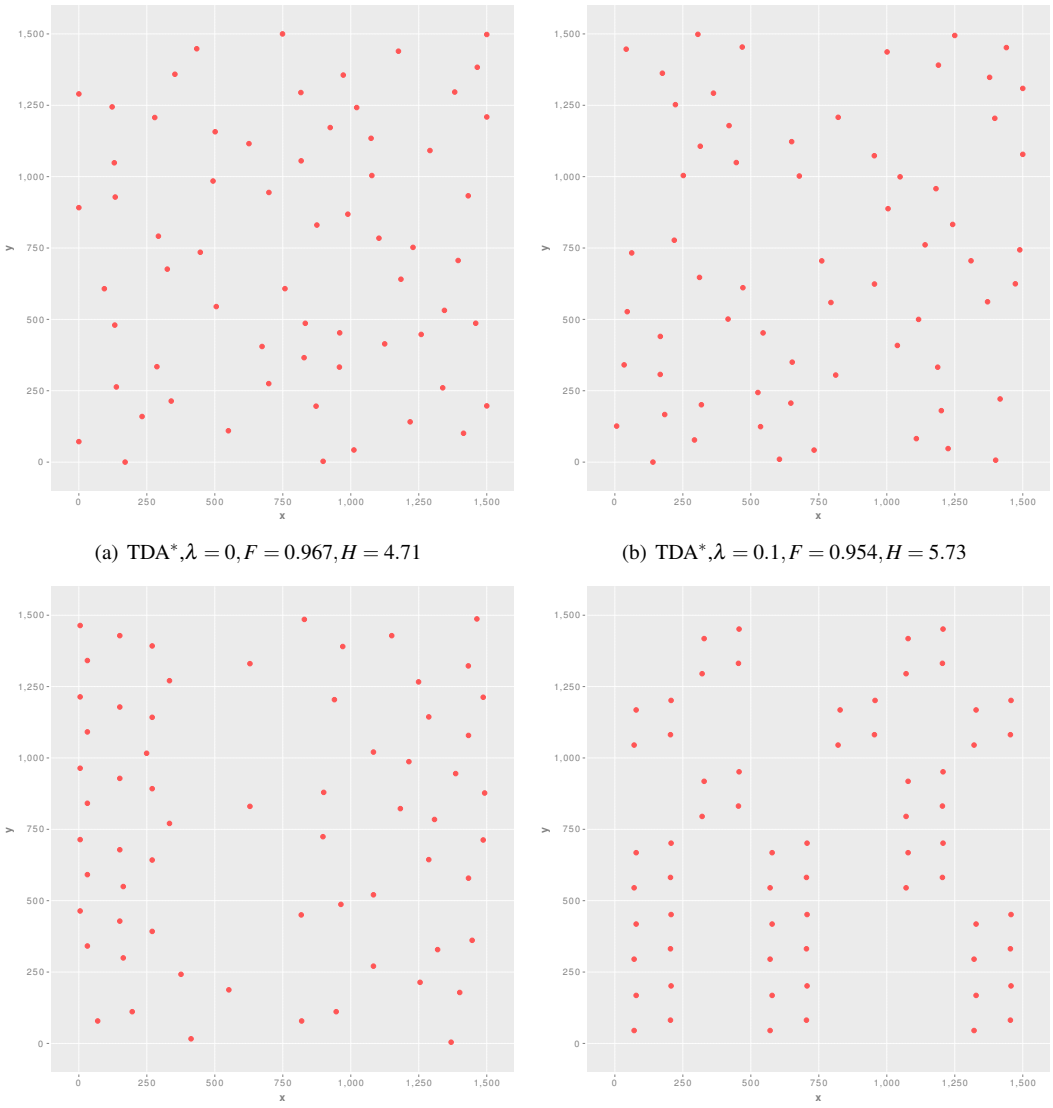
We note at this point that tests for statistical significance only determine the likelihood of average algorithm performance variations. The results, however, show that best-of-run performance is often quite different from the mean. This is illustrated most clearly in Figure 7(a) for the algorithm BC* with $\lambda = 0.1$: the best layout identified by the algorithm is comparable to average layouts found by the other algorithms with $\lambda < 0.1$, even though the mean F values performance of this algorithm is quite low.

Finally, we examined visually some of the optimal layouts found after different runs of the various techniques. Examples of different optimised layouts are shown in Figures 8–10.

Focussing firstly on Figure 8, which depicts some layout solutions to Problem A, we can see clearly the difference between the TDA and BC. For the TDA-based layouts (Figure 8(a) and (b)), the arrangement of turbines has a clear random character. This is even the case where the λ value is at its highest in Figure 8(b) – in this case, the H objective is not much different than it is in (a) where it is not optimised at all.

In contrast, the BC* algorithms tend to produce more patterned layouts. For example, Figure 8(c) does not make use of the harmony metric at all but it still produces a degree of translational symmetry

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017



(a) TDA*,$\lambda = 0, F = 0.967, H = 4.71$

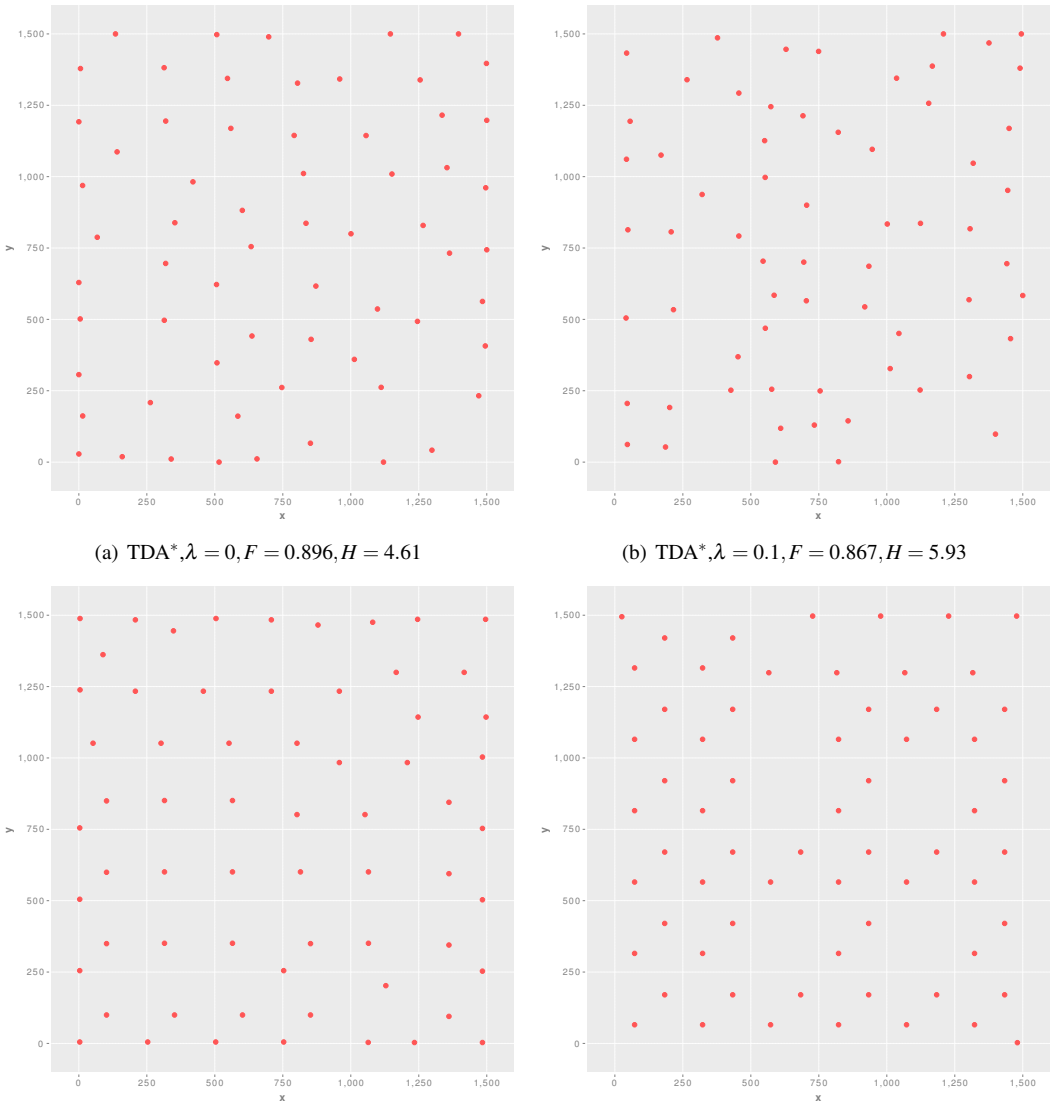
(b) TDA*,$\lambda = 0.1, F = 0.954, H = 5.73$

(c) BC*,$\lambda = 0, F = 0.978, H = 4.81$

(d) BC*,$\lambda = 0.1, F = 0.955, H = 7.60$

Figure 8. Example layouts produce by TDA* (a,b) and BC* (c,d) on Problem A.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017



(a) TDA*,$\lambda = 0, F = 0.896, H = 4.61$

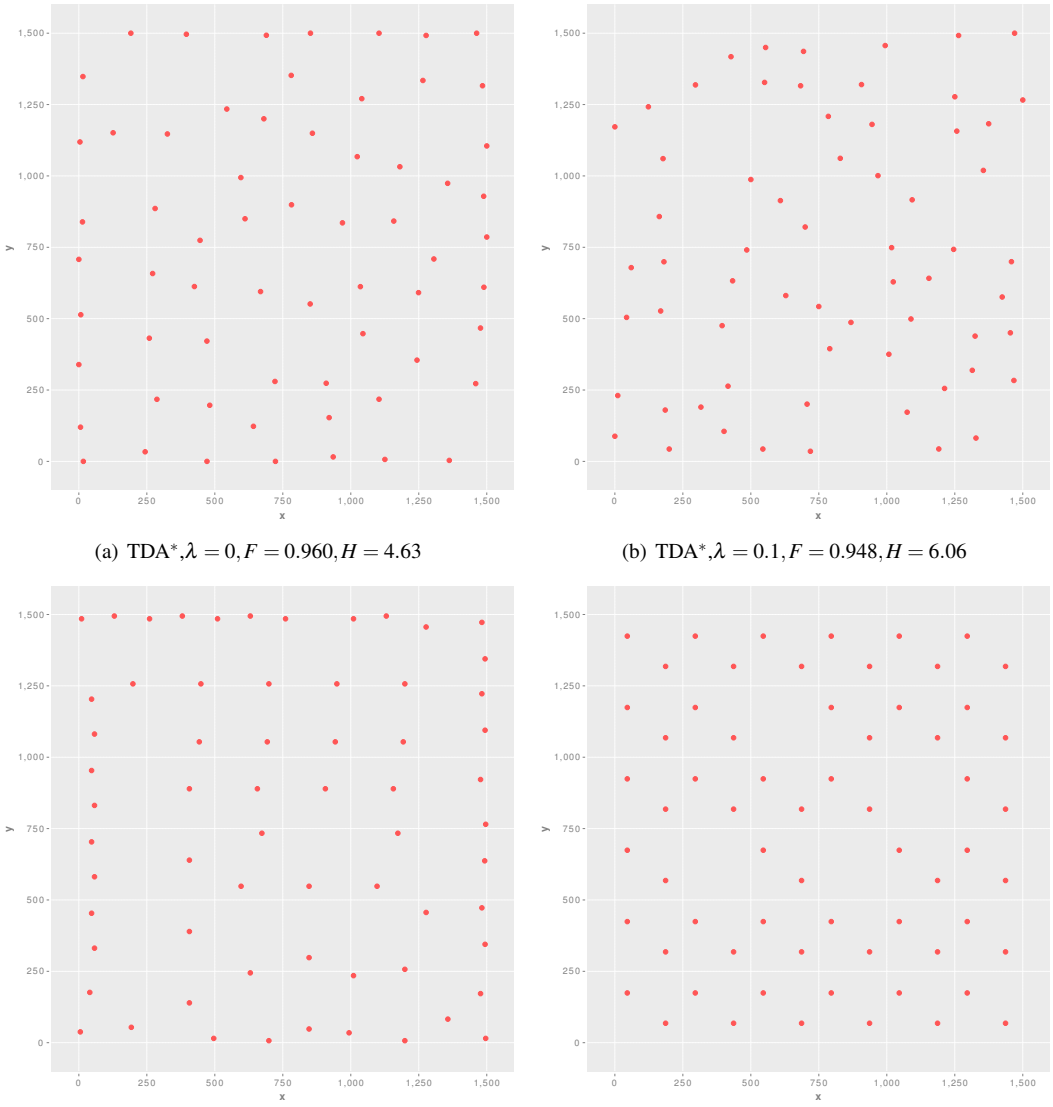
(b) TDA*,$\lambda = 0.1, F = 0.867, H = 5.93$

(c) BC*,$\lambda = 0, F = 0.903, H = 4.76$

(d) BC*,$\lambda = 0.01, F = 0.891, H = 6.42$

Figure 9. Example layouts produce by TDA* (a,b) and BC* (c,d) on Problem B.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017



(a) TDA*,$\lambda = 0, F = 0.960, H = 4.63$

(b) TDA*,$\lambda = 0.1, F = 0.948, H = 6.06$

(c) BC*,$\lambda = 0, F = 0.962, H = 4.78$

(d) BC*,$\lambda = 0.1, F = 0.956, H = 6.65$

Figure 10. Example layouts produce by TDA* (a,b) and BC* (c,d) on Problem C.

across the layout. When λ is high however, as in Figure 8(d), the translational symmetry is increased dramatically to produce a considerably more regular arrangements of turbines.

A somewhat different story can be told for the examples shown in Figure 9 which depicts example solutions to Problem B. This benchmark problem has no dominant wind direction, which becomes evident when the layouts are examined. To illustrate, both algorithms optimising solely for efficiency (see Figures 9(a) and (c)) tend to push turbines out to the furtherest possible edge of the layouts, thus maximising the space between turbines in all directions. The BC algorithm interestingly produces a layout that is somewhat grid-like in structure in this case (see Figure 9(c)).

Finally, Figure 10 depicts selected solutions to Problem C. Similar trends can be observed in the solutions to this problem as were observed for the other problems. For example, BC* produces more regular layouts than TDA regardless of λ value on this problem. Also of considerable interest with regards to Problem is the difference in efficiencies achieved: the BC* algorithm with $\lambda = 0.1$ produces an optimal layout in the Figure with an efficiency of only about half a percent less than TDA*'s best layout with $\lambda = 0$.

As a final comment, it will be pointed out that the harmony metric is maximised for patterns that are

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017

completely uniform. For example, see Figure 4(a). The practical effect of this bias is that the search is more likely to be focussed on sparse layouts – i.e. layouts with several empty blocks. This is because empty blocks correspond to uniform patterns which maximise H.

To illustrate, see Figure 8(d) for an extreme example, and Figure 10(d) for a less extreme example. The practical consequences of this are considerable, since less land can be used for the same or approximately the same efficiency. This may positively impact on both the cost of land (e.g. see Chen and MacDonald (2012)) and the effect on wildlife (e.g. see the survey on environmental implications of wind energy by Tabassum-Abbasi et al. (2014)) and warrants further investigation as well. This finding also suggests that purely regular approaches (e.g. Neubert et al. (2010)) may not be ideal solutions to this problem because such approaches distribute turbines uniformly across the layout without any chance of free space areas such as those in figures appearing.

6 CONCLUSION

To conclude, we have investigated a metahuristic optimisation approach to solving the wind farm layout optimisation problem, in which one objectives concerns maximising layout aesthetics while the other concerns maximising energy efficiency. The aesthetics of layouts is an important consideration which most other literature in this field to date has not considered. Our experiments reported here were successful and encourage future research to further refine this initial approach. In particular, while the Jensen model is adequate for this initial work, future work should explore more realistic wind farm simulations that better account for both wind (e.g. Feng and Shen (2015a)) and partial wakes (e.g. Feng and Shen (2015b)).

As mentioned in the Introduction section, there are two main limitations of the work presented here. Firstly, our approach is largely constrained to two dimensional layouts, as would typically be encountered offshore or on sites that are plains. Clearly therefore, Salingaros' approach must be generalised to three dimensions before the same ideas can be applied to other types of site. Our current thinking is that there are two possible approaches to this generalisation: (i) topography could be included in the definition of the symbols, which would complicate the definition of what a symbol is somewhat (e.g. a turbine on the top of a knoll would result in a different symbol compared to a turbine at the same relative position but in the middle of plain); or alternatively (ii) the method could be generalised to include transformations in the z dimension – this approach would require expanding the set of symmetries listed in Table 1 to include all 3D symmetries as well.

The second main limitation of this work is that it focusses on the overhead view of the layout only and assumes by default that aesthetics is uniformly important across the entire layout. This is in fact a false assumption for farms located near places that people frequent such as nearby towns, tourist attractions and highways. In such situations, the aesthetics of the portion of the farm in clear view of the people will be far more important than the parts of the farm that are obscured. Therefore vantage point is important in this situation. We therefore feel that our "global" approach to aesthetics could be complemented by a corresponding "local" approach that takes into account viewpoint. The local approach could take an image aesthetics-based approach and render the farm and its surrounding terrain, skyline and other features as an eye-level scene, and then assess its aesthetics using machine learning. Marchesotti et al. (2015) is one example of such an approach that could be gainfully employed here.

REFERENCES

Al-Yahyai, S., Charabi, Y., and Gastli, A. (2015). Geometrical approach for wind farm symmetrical layout design optimization. In *GCC Conference and Exhibition (GCCCE), 2015 IEEE 8th*, pages 1–6.

Chen, L. and MacDonald, E. (2012). Considering landowner participation in wind farm layout optimization. *Journal of Mechanical Design*, 134(8):084506–084506.

Dai, K., Bergot, A., Liang, C., Xiang, W.-N., and Huang, Z. (2015). Environmental issues associated with wind energy – a review. *Renewable Energy*, 75:911 – 921.

den Heijer, E. and Eiben, A. (2010). Comparing aesthetic measures for evolutionary art. In *Applications of Evolutionary Computation*, volume 6025 of *Lecture Notes in Computer Science*, pages 311–320.

Feng, J. and Shen, W. (2015a). Modelling wind for wind farm layout optimisation using joint distribution of wind speed and wind direction. *Energies*, 8:3075–3092.

Feng, J. and Shen, W. Z. (2015b). Solving the wind farm layout optimization problem using random search algorithm. *Renewable Energy*, 78(0):182 – 192.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017

Galanter, P. (2012). Computational aesthetic evaluation: Past and future. In McCormack, J. and d'Inverno, M., editors, *Computers and Creativity*, chapter 11, pages 255–293. Springer Berlin Heidelberg.

Global Wind Energy Council (2014). *Global Wind Energy Outlook 2014*.

Guirguis, D., Romero, D. A., and Amon, C. H. (2016). Toward efficient optimization of wind farm layouts: Utilizing exact gradient information. *Applied Energy*, 179:110 – 123.

Jensen, N. (1983). A note on wind generator interaction. Technical report, Risø DTU National Laboratory for Sustainable Energy.

Katic, I., Høstrup, J., and Jensen, N. (1986). A simple model for cluster efficiency. In *Proc. Europe and Wind Energy Association Conference and Exhibition*.

Klinger, A. and Salingaros, N. (2000). A pattern measure. *Environment and Planning B: Planning and Design*, 27:537–547.

Marchesotti, L., Murray, N., and Perronnin, F. (2015). Discovering beautiful image attributes for aesthetic image analysis. *International Journal of Computer Vision*, 113:246–266.

Mayo, M. and Daoud, M. (2016). Informed mutation of wind farm layouts to maximise energy harvest. *Renewable Energy*, 89:437–448.

Mayo, M., Daoud, M., and Zheng, C. (2016). Randomising block sizes for blockcopy-based wind farm layout optimisation. In *Proc 20th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, pages 277–289.

Mayo, M. and Zhen, C. (2016). Blockcopy-based operators for evolving efficient wind farm layouts. In *Proc 2016 IEEE Congress on Evolutionary Computation*, pages 1085–1092.

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

Neubert, A., Shah, A., and Schlez, W. (2010). Maximum yield from symmetrical wind farm layouts. In *Proc. 10th German Wind Energy Conference, DEWEK*.

Piorkowski, M. D., Farnsworth, A. J., Fry, M., Rohrbaugh, R. W., Fitzpatrick, J. W., and Rosenberg, K. V. (2012). Research priorities for wind energy and migratory wildlife. *The Journal of Wildlife Management*, 76(3):451–456.

Rodrigues, S. M. F., Bauer, P., and Pierik, J. (2013). Modular approach for the optimal wind turbine micro siting problem through CMA-ES algorithm. In *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '13 Companion, pages 1561–1568. ACM.

Salingaros, N. (1997). Life and complexity in architecture from a thermodynamic analogy. *Physics Essays*, 10(1).

Samorani, M. (2013). The wind farm layout optimization problem. In Pardolas, P., editor, *Handbook of Wind Power Systems*, pages 21–38. Springer-Verlag.

Shakoor, R., Hassan, M., Raheem, A., and Wu, Y. (2016). Wake effect modelling: A review of wind farm layout optimization using Jensen's model. *Renewable and Sustainable Energy Reviews*, 58:1048–1059.

Song, Z., Zhang, Z., and Chen, X. (2016). The decision model of 3-dimensional wind farm layout design. *Renewable Energy*, 85:248 – 258.

Tabassum-Abbasi, Premalatha, M., Abbasi, T., and Abbasi, S. (2014). Wind energy: Increasing deployment, rising environmental concerns. *Renewable and Sustainable Energy Reviews*, 31:270 – 288.

Tsoutsos, T., Gouskos, Z., Karterakis, S., and Peroulaki, E. (2006). Aesthetic impact from wind parks. Technical report, Technical University of Crete, Chania, Greece.

Wagner, M., Day, J., and Neumann, F. (2013). A fast and effective local search algorithm for optimizing the placement of wind turbines. *Renewable Energy*, 51:64–70.

Watts, J. (2012). Winds of change blow through china as spending on renewable energy soars, `http://www.theguardian.com/world/2012/mar/19/china-windfarms-renewable-energy`. *The Guardian*.

Wilson, D., Cussat-Blanc, S., Veeramachaneni, K., O'Reilly, U., and Luga, H. (2014). A continuous development model for wind farm layout optimization. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 745–752, New York, NY, USA. ACM.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.2864v1 | CC BY 4.0 Open Access | rec: 10 Mar 2017, publ: 10 Mar 2017