Taylor & Francis
Taylor & Francis Group

Engineering Optimization

ISSN: 0305-215X (Print) 1029-0273 (Online) Journal homepage: https://www.tandfonline.com/loi/geno20

A tight upper bound for quadratic knapsack problems in grid-based wind farm layout optimization

Ning Quan & Harrison M. Kim

To cite this article: Ning Quan & Harrison M. Kim (2018) A tight upper bound for quadratic knapsack problems in grid-based wind farm layout optimization, Engineering Optimization, 50:3, 367-381, DOI: 10.1080/0305215X.2017.1316844

To link to this article: https://doi.org/10.1080/0305215X.2017.1316844

Published online: 08 May 2017.

ENGINEERING OPTIMIZATION, 2018
VOL. 50, NO. 3, 367–381
https://doi.org/10.1080/0305215X.2017.1316844

Taylor & Francis
Taylor & Francis Group

A tight upper bound for quadratic knapsack problems in grid-based wind farm layout optimization

Ning Quan and Harrison M. Kim

University of Illinois at Urbana-Champaign, Department of Industrial and Enterprise Systems Engineering, Urbana, IL, USA

ABSTRACT

The 0-1 quadratic knapsack problem (QKP) in wind farm layout optimization models possible turbine locations as nodes, and power loss due to wake effects between pairs of turbines as edges in a complete graph. The goal is to select up to a certain number of turbine locations such that the sum of selected node and edge coefficients is maximized. Finding the optimal solution to the QKP is difficult in general, but it is possible to obtain a tight upper bound on the QKP's optimal value which facilitates the use of heuristics to solve QKPs by giving a good estimate of the optimality gap of any feasible solution. This article applies an upper bound method that is especially well-suited to QKPs in wind farm layout optimization due to certain features of the formulation that reduce the computational complexity of calculating the upper bound. The usefulness of the upper bound was demonstrated by assessing the performance of the greedy algorithm for solving QKPs in wind farm layout optimization. The results show that the greedy algorithm produces good solutions within 4% of the optimal value for small to medium sized problems considered in this article.

ARTICLE HISTORY
Received 15 July 2016
Accepted 28 March 2017

KEYWORDS
Wind farm; layout optimization; mixed integer programming; quadratic knapsack problem; upper bound; greedy algorithm

1. Introduction

In recent years, wind energy has become one of the most promising alternatives to traditional fossil fuels in power generation. According to the European Wind Energy Association (Pineda *et al.* 2016), wind power's share of total installed power capacity increased from 2.4% in 2000 to 15.6% in 2015, overtaking hydropower as the third largest power generation capacity. The United States Department of Energy (2015) predicts that the prevalence of wind power would increase from 5% of current power generation to 20% by 2030. Research into wind farm design generally seeks to determine the location, number, and layout of turbines in the wind farm for optimizing some performance measure, such as power generation or cost per unit of energy. The reader can refer to González *et al.* (2012) and Herbert-Acero *et al.* (2014) for a review of past works in wind farm design optimization.

Problems in wind farm layout optimization can be classified as continuous or grid-based depending on the feasible space for turbine placement. Continuous approaches allow for unrestricted placement of turbines, while grid-based approaches allow placement of turbines over a regularly spaced grid of possible locations. These two approaches are not mutually exclusive; previous works

CONTACT Harrison M. Kim ✉ hmkim@illinois.edu

by Wagner *et al.* (2013); Réthoré *et al.* (2014); Park and Law. (2015); Long and Zhang. (2015) have used grid-based approaches to find an initial layout, which is then refined using continuous search methods. This article focuses on the grid-based approach, with power output maximization as the objective.

The grid-based layout optimization problem can be modelled as an optimization problem over an undirected complete graph. The nodes in the graph represent possible turbine locations, and the edge between any two nodes represents wake effect interactions between turbines placed at those two locations. When wind passes through a turbine's disc, a wake cone of slower moving air is created behind the turbine. Therefore, the problem of finding a power output maximizing layout is closely related to the problem of finding a layout where wake effect interactions are minimized.

The optimization problem is complicated by the fact that the incoming wind speed to any turbine is dependent on the relative positions of the other turbines in the farm. This means that the edge coefficient that represents the mutual power loss caused by the turbines placed at the two nodes is not a constant, but a function of the entire layout. Furthermore, the total power loss experienced by a turbine placed at a particular node is a non-linear function of the connecting edge coefficients. For example, the commonly used Jensen wake model (Katic *et al.* 1986) uses a sum-of-squares approach to calculate the total wind speed deficit experienced by a turbine. All of this creates a non-linear, non-convex discrete optimization problem that is very difficult to solve to global optimality. This is why such heuristics as the genetic algorithm (Mosetti *et al.* 1994; Grady *et al.* 2005; Chen and MacDonald. 2014; Dobrić and Duriić. 2014; Long and Zhang. 2015) or greedy algorithm (Zhang *et al.* 2011; Saavedra-Moreno *et al.* 2011; Chen *et al.* 2013; Song *et al.* 2015) are commonly used to find good, but not necessarily optimal, solutions.

The complexity of the grid-based layout optimization problem can be greatly reduced by ignoring the influence of the entire layout when calculating node and edge coefficients. The node coefficient becomes the power generated by a turbine placed at that location, without considering power loss due to wake overlaps, and the edge coefficient becomes a negative constant representing power loss caused by the two turbines' wakes on each other, without considering the wakes of other turbines in the wind farm. The objective is also changed from power generation to a quadratic function of binary variables indicating turbine placements in the grid. The squared and cross terms in the quadratic objective function are paired with the corresponding node and edge coefficients. The overall goal is thus to select up to a certain number of nodes, such that the sum of node and edge coefficients belonging to the selected nodes is maximized. This produces a 0-1 quadratic knapsack problem (QKP) that has a simple, straightforward formulation, but may not necessarily be easier to solve. Small instances of the QKP can be transformed into an equivalent mixed integer linear program (MILP) and solved to global optimality via branch and bound (Zhang *et al.* 2014; Turner *et al.* 2014), but the QKP is NP-hard, as shown by Caprara *et al.* (1999), so heuristics must be used for problems problems with denser grids.

This article puts forward the argument that the main advantage of using the QKP formulation in wind farm layout optimization is not that it is solvable for small problem instances, but that there are numerous ways of finding good upper bounds on the optimal values of QKPs. One can refer to Pisinger (2007) for an in-depth review and comparison of methods for general QKPs. Obtaining a globally optimal solution may be beyond reach for large QKPs in layout optimization, but getting a tight upper bound on the optimal value can provide a good estimate of the optimality gap of any feasible solution, and give the designer a measure of confidence in the solution generated by the chosen heuristic.

An important question to consider is whether the lower-fidelity QKP formulation will produce sub-par layouts compared with traditional formulations with power generation objective functions. This article does not provide a conclusive answer for all possible problems, but comparisons for one-dimensional problems are possible. Table 1 shows the percentage differences in power generation of two optimal layouts produced using the Jensen wake model and the simpler approach used in the

Table 1. Percentage difference in power generation of optimal layouts.

	Ambient wind speed (m/s)		
Turbine count	14	9	4
4	0	0	0
5	0	0.01	0.02
6	0.09	0.1	0.2
7	0.3	0.3	0.6

QKP formulation. The total power generation of both layouts was calculated based on the higher-fidelity Jensen wake model. The feasible space consists of 20 m spaced points along a 1600 m line, and the turbine model is the same as the one used in the numerical experiments in this article. The example may be one-dimensional, but it reflects how power output in layout optimization is commonly assessed along individual wind directions.

The results show that the power generation of the layout produced by the QKP formulation for a variety of wind speeds and turbine counts is very close to the layout produced by the higher-fidelity Jensen wake model. The same trend was observed for larger grid spacings of 40 m and 80 m. It seems that the simplifications made in the QKP formulation do not result in a significant decrease in solution quality for the example considered in this article. In return, these simplifications produce a formulation that has certain properties that allow a tight upper bound on its optimal value to be found.

The contribution of this work is to adapt an upper bound method developed by Billionnet *et al.* (1999) (denoted the BFS bound) for general QKPs to QKPs in grid-based layout optimization. The BFS bound was chosen for its promising performance, as reported by Billionnet *et al.* (1999) for general QKPs, and because the QKP formulation proposed in this article has certain properties that improve the quality of the BFS bound and decrease its computational complexity.

The usefulness of the BFS bound is demonstrated; it is used to assess the performance of the greedy algorithm for solving QKPs in grid-based layout optimization. The QKP formulation in grid-based layout optimization is an example of a submodular maximization problem (Krause and Golovin. 2014). A well-known result from Nemhauser and Wolsey. (1978) states that the solution value generated by the greedy algorithm is at least 63% of the optimal value for submodular maximization problems with non-decreasing objective functions. In practice, the greedy algorithm is usually able to perform much better than the worst-case bound, and has comparable performance to the commonly used genetic algorithm, as demonstrated by Zhang *et al.* (2011). The BFS bound is used in this article to get much tighter estimates of the optimality gaps of greedy solutions to a range of QKP instances in grid-based layout optimization.

In the rest of the article, Section 2 introduces the QKP formulation in grid-based layout optimization and elaborates on its submodular nature. Section 3 then describes in detail how the BFS bound can be applied to QKPs in grid-based layout optimization, as well as the computational advantages of the proposed QKP formulation. The numerical performance of the greedy algorithm is assessed in Section 4, before conclusions are drawn in Section 5.

2. QKP formulation in grid-based layout optimization

2.1. QKP in grid-based layout optimization

The QKP formulation for grid-based layout optimization is given in Equation (1). A binary variable x_i is used to indicate whether a turbine is placed at a particular point i in the grid, and the set $X = \{x_1, \ldots, x_n\}$ contains n binary variables corresponding to n points in the grid. The parameter T in the knapsack constraint denotes the maximum number of turbines to be placed on the wind farm.

$$\begin{aligned} \max_{x_{i=1,\ldots,n}} f(X) = \sum_{i=1}^{n} c_{ii}x_i + \sum_{i=1}^{n}\sum_{\substack{j=1\\ i\neq j}}^{n} c_{ij}x_ix_j \\ \text{subject to :} \\ \sum_{i=1}^{n} x_i \leq T \\ x_i \in \{0,1\}\ \forall i \in \{1,\ldots,n\} \end{aligned} \tag{1}$$

The coefficient c_{ii} represents the stand-alone expected power generated by a turbine placed at point i, and the coefficient c_{ij} is defined in Equation (2). The term l_{ij} in Equation (2) represents the expected power loss experienced by a turbine at point j caused by a turbine at point i (without considering any other turbines in the wind farm). The definition places no restrictions on how l_{ij} is calculated, other than the requirement that it must be non-negative. The wake models used in this article to calculate l_{ij} are covered in the following subsection.

$$c_{ij} = \begin{cases} -P & \text{if } ij \text{ violates minimum inter-turbine distance} \\ -l_{ij} & \text{otherwise} \end{cases} \tag{2}$$

There are no constraints in the QKP formulation enforcing minimum inter-turbine distance requirements. A penalty approach is used instead by assigning a penalty P to the coefficient c_{ij} of any pair ij that violates the minimum inter-turbine distance requirement. Suppose an optimal solution has a pair mn with distance less than the minimum inter-turbine distance; then a better solution can be found by setting either x_m or x_n to 0. The resulting solution still satisfies the knapsack inequality in the QKP formulation, and the minimum increase in objective function value from getting rid of c_{mn} is $2P$. The maximum possible decrease in objective function value from setting x_m or x_n to 0 is given by the right-hand side of Equation (3). Therefore, if the problem admits feasible solutions, no pairs violating the minimum inter-turbine distance requirement can be present in the optimal solution if the penalty is set large enough to satisfy Equation (3).

$$2P > \max_{i} c_{ii} \tag{3}$$

2.2. Wake and wind speed models

It is assumed in this article that the continuous range of wind directions is discretized. Let W represent the set of wind directions, each with probability $p_{w\in W}$ of occurring. It is also assumed that each direction $w \in W$ has an associated wind speed probability density function f_w. The power loss term l_{ij} used in this article is defined in Equation (4).

$$l_{ij} = \sum_{w\in W} p_w \int_{0}^{+\infty} \big(G(v_w) - G(v_{w,ij})\big) f_w(v_w)\mathrm{d}v_w \tag{4}$$

The expression $G(v)$ represents the power generated by a turbine with incoming wind speed of v, and $v_{w,ij}$ is the incoming wind speed along direction w for a turbine at point j. Note that $v_{w,ij}$ is calculated assuming that there are only two turbines placed at points i and j in the entire wind farm. The wake model used to calculate $v_{w,ij}$ is taken from Lackner and Elkinton. (2007), and is shown in

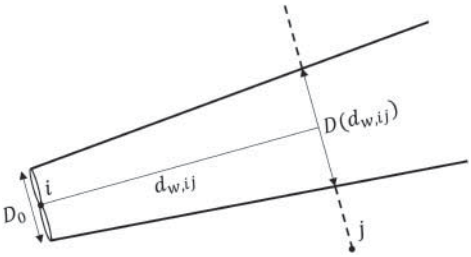

Figure 1. Downstream distance and wake cone diameter.

Equation (5).

$$v_{w,ij} = \left[1 - \left(\frac{4A_{w,ij}}{\pi D_0^2}\right)\left(\frac{1 - \sqrt{1 - C_t(v_w)}D_0^2}{D(d_{w,ij})^2}\right)\right] v_w \tag{5}$$

D_0 in Equation (5) represents the turbine's disc diameter, and $C_t(v)$ is the thrust coefficient of the turbine with an incoming wind speed of v. The term $A_{w,ij}$ is the area of a turbine's disc located at point j that lies inside a wake cone originating from point i along direction w. The diameter $D(d_{w,ij})$ of the wake cone originating from point i along direction w at a downstream location j is given in Equation (6), where κ is the wake expansion coefficient, and $d_{w,ij}$ is the downstream distance along direction w between points i and j. The definitions of $d_{w,ij}$ and $D(d_{w,ij})$ are made clear in Figure 1.

$$D(d_{w,ij}) = D_0 + 2\kappa d_{w,ij} \tag{6}$$

The QKP formulation allows l_{ij} to be defined with any level of modelling fidelity. However, the net effect of all pairwise interactions between turbines can only be linearly additive, which produces a submodular maximization problem, as shown in the following subsection.

2.3. Submodularity of the objective function

Song *et al.* (2015) showed that the submodularity of the power generation objective function in grid-based layout optimization holds under certain conditions when non-linear wake effect supposition models are used. When the total wake effect is linearly additive, as in the case of the QKP formulation considered in this article, the objective function's submodular property holds in general.

Let $F(S_k)$ be the objective value when k turbines are placed on the grid points in set S_k. Then for any two sets S_{k_1} and S_{k_2} where $S_{k_1} \subset S_{k_2}$, and any grid point $i \notin S_{k_2}$, F must satisfy the condition shown in Equation (7) in order to be submodular.

$$F\left(S_{k_1} \cup \{i\}\right) - F\left(S_{k_1}\right) \geq F\left(S_{k_2} \cup \{i\}\right) - F\left(S_{k_2}\right) \tag{7}$$

The submodularity condition in Equation (7) can be expressed in terms of the QKP's objective function coefficients, as shown in Equation (8). The interaction terms c_{ij} are non-positive, so the submodularity condition holds for the QKP formulation considered in this article. The submodularity of QKPs in grid-based layout optimization is illustrated in Figure 2, which shows how adding a new turbine location i to solution S_{k_2} will produce more new wake interactions than adding i to S_{k_1}.

$$c_{ii} + \sum_{j \in S_{k_1}} \left(c_{ij} + c_{ji}\right) \geq c_{ii} + \sum_{j \in S_{k_1}} \left(c_{ij} + c_{ji}\right) + \sum_{j \in S_{k_2} \setminus S_{k_1}} \left(c_{ij} + c_{ji}\right) \tag{8}$$

If the greedy algorithm can find a feasible solution for a given QKP, then the large penalty term in c_{ij} for infeasible location pairs can be replaced with l_{ij}, without changing the greedy solution value. The

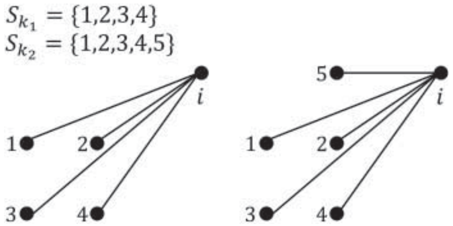

Figure 2. Illustration of submodularity.

magnitude of the power loss term l_{ij} is usually much smaller than the magnitude of the stand-alone power generation term c_{ii}, so the objective function is non-decreasing for realistic problem instances, and the increase in objective function value by adding i to S_{k_2} is smaller than adding i to any subset of S_{k_2}.

Nemhauser and Wolsey. (1978) provided a worst-case bound on the performance of the greedy algorithm for solving submodular maximization problems with non-decreasing set objective functions. In practice, the greedy algorithm is capable of performing much better than the worst case. This article references previous work done by Billionnet *et al.* (1999) to obtain a much tighter upper bound on the optimal value of the QKP formulation in grid-based layout optimization.

3. The BFS bound

This section gives a detailed overview of the dual decomposition approach developed by Billionnet *et al.* (1999) to calculate the BFS bound, and why it can be well-suited to QKPs in grid-based layout optimization. The first step is to split the original grid into disjoint sets. Let $\{X_1, \dots, X_p\}$ be a partition of X into p disjoint sets. The notation used in the rest of the article is:

- $Y_k = X \setminus X_k$.
- I_k (or J_k) is the index set of variables in X_k (or Y_k).
- $cl(i)$ is the index of the set that contains variable x_i.
- $\boldsymbol{x}_{I_k}$ is the vector of variables x_i, $i \in I_k$.
- $\boldsymbol{x}_{J_k}$ is the vector of variables x_j, $j \in J_k$.

The original formulation can now be described in terms of this notation. First, let $f_k(\boldsymbol{x}_{I_k}, \boldsymbol{x}_{J_k})$ be a function defined as shown in Equation (9); $f_k(\boldsymbol{x}_{I_k}, \boldsymbol{x}_{J_k})$ represents the original objective function broken down according to the grid partitions. The first term in Equation (9) is the total stand-alone power generated by turbines in partition k, and the second and third terms represent the intra-partition and inter-partition pairwise power losses, respectively. The original objective $f(X)$ in Equation (1) can now be expressed as a sum, $f(X) = \sum_{k=1}^{p} f_k(\boldsymbol{x}_{I_k})$.

$$f_k(\boldsymbol{x}_{I_k}) = \sum_{i \in I_k} c_{ii} x_i + \sum_{i \in I_k} \sum_{\substack{i' \in I_k \\ i \neq i'}} c_{ii'} x_i x_{i'} + \sum_{i \in I_k} \sum_{j \in J_k} c_{ij} x_i x_j \tag{9}$$

In the next step, copies of the variables in $\boldsymbol{x}_{J_k}$ are created for every disjoint set k. These copies (denoted $y^k_{j \in J_k}$) collectively form the vector $\boldsymbol{y}_{J_k}$, and constraints shown in Equation (10) are added to the original formulation to maintain consistency between the original variables and their copies.

$$\begin{aligned} x_j &= y^k_j \quad \forall j \in J_k,\ \forall k \in \{1, \dots, p\} \\ x_i y_j^{cl(i)} &= x_j y_i^{cl(j)} \quad \forall i \in \{1, \dots, n-1\},\ \forall j \in \{i+1, \dots, n\},\ cl(j) \neq cl(i) \end{aligned} \tag{10}$$

These changes are made to the original QKP formulation, yielding the equivalent formulation shown in Equation (11), where $\boldsymbol{x}$ (or $\boldsymbol{y}$) is the vector of x_i (or y_j^k) variables.

$$\max_{\boldsymbol{x},\boldsymbol{y}} \bar{f}(\boldsymbol{x},\boldsymbol{y}) = \sum_{k=1}^{p}\left(\sum_{i\in I_k} c_{ii}x_i + \sum_{i\in I_k}\sum_{\substack{i'\in I_k\\ i\neq i'}} c_{ii'}x_i x_{i'} + \sum_{i\in I_k}\sum_{j\in J_k} c_{ij}x_i y_j^k\right)$$

subject to :

$$x_j = y_j^k \;\forall j\in J_k,\; \forall k\in\{1,\dots,p\} \tag{11a}$$

$$x_i y_j^{cl(i)} = x_j y_i^{cl(j)} \;\forall i\in\{1,\dots,n-1\},\; \forall j\in\{i+1,\dots,n\},\; cl(j)\neq cl(i) \tag{11b}$$

$$\sum_{i\in I_k} x_i + \sum_{j\in J_k} y_j^k \leq T \;\forall k\in\{1,\dots,p\} \tag{11c}$$

$$x_i\in\{0,1\} \;\forall i\in I_k,\; \forall k\in\{1,\dots,p\} \tag{11d}$$

$$y_j^k\in\{0,1\} \;\forall j\in J_k,\; \forall k\in\{1,\dots,p\} \tag{11e}$$

Constraints (11a) and (11b) can be brought into the objective function via dual relaxation. The Lagrangian multipliers

$$\boldsymbol{\lambda} = \left(\lambda_j^k\right)_{1\leq k\leq p,\, j\in J_k}$$

and

$$\boldsymbol{\mu} = \left(\mu_{ij}\right)_{1\leq i<j\leq n,\, cl(i)\neq cl(j)}$$

are introduced for constraints (11a) and (11b), respectively. Relaxing constraints (11a) and (11b) makes the formulation in Equation (11) completely separable, with each subproblem $k\in\{1,\dots,p\}$ having decision variable vectors $\boldsymbol{x}_{I_k}$ and $\boldsymbol{y}_{J_k}$. The objective function of the relaxed formulation, also known as the Lagrangian function L, can be expressed as shown in Equation (12).

$$\begin{aligned} L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\mu}) = \sum_{k=1}^{p}\Bigg(&\sum_{i\in I_k}\Big(c_{ii} + \sum_{h\neq k}\lambda_i^h\Big)x_i - \sum_{j\in J_k}\lambda_j^k y_j^k \\ &+ \sum_{i\in I_k}\sum_{\substack{i'\in I_k\\ i\neq i'}} c_{ii'}x_i x_{i'} + \sum_{i\in I_k}\sum_{\substack{j\in J_k\\ i<j}}\left(c_{ij}+\mu_{ij}\right)x_i y_j^k + \sum_{i\in I_k}\sum_{\substack{j\in J_k\\ j<i}}\left(c_{ij}-\mu_{ji}\right)x_i y_j^k\Bigg) \end{aligned} \tag{12}$$

The dual function $w(\boldsymbol{\lambda},\boldsymbol{\mu})$ of the formulation in Equation (11) is the maximum of $L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\mu})$ with respect to $\boldsymbol{x},\boldsymbol{y}$, and subject to the constraints (11c), (11d), and (11e):

$$w(\boldsymbol{\lambda},\boldsymbol{\mu}) = \max_{\substack{\boldsymbol{x},\boldsymbol{y}\\ \text{s.t.}(11c),(11d),(11e)}} L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\mu}) \tag{13}$$

The dual function $w(\boldsymbol{\lambda},\boldsymbol{\mu})$ is convex (Bertsekas 2008), and provides an upper bound on the optimal solution of Equation (11). The BFS bound is obtained by minimizing $w(\boldsymbol{\lambda},\boldsymbol{\mu})$ with respect to the

Lagrangian multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ using the subgradient method described in Held *et al.* (1974), the details of which will be provided in Section 4.

Evaluation of the dual function for fixed $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ can be performed in parallel since the Lagrangian maximization problem in Equation (13) can be separated into p independent Lagrangian subproblems, as shown in Equation (15), where L_k (Equation (14)) is the part of L belonging to subproblem k.

$$L_k(\boldsymbol{x}_{I_k}, \boldsymbol{y}_{J_k}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{i \in I_k} \left(c_{ii} + \sum_{h \neq k} \lambda_i^h \right) x_i - \sum_{j \in J_k} \lambda_j^k y_j^k + \sum_{i \in I_k} \sum_{\substack{i' \in I_k \\ i \neq i'}} c_{ii'} x_i x_{i'} + \sum_{i \in I_k} \sum_{\substack{j \in J_k \\ i<j}} (c_{ij} + \mu_{ij}) x_i y_j^k + \sum_{i \in I_k} \sum_{\substack{j \in J_k \\ j<i}} (c_{ij} - \mu_{ji}) x_i y_j^k \tag{14}$$

$$\max_{\boldsymbol{x}_{I_k}, \boldsymbol{y}_{J_k}} L_k(\boldsymbol{x}_{I_k}, \boldsymbol{y}_{J_k}, \boldsymbol{\lambda}, \boldsymbol{\mu})$$

$$\text{subject to :}$$

$$x_i \in \{0, 1\} \ \forall i \in I_k \tag{15a}$$

$$y_j^k \in \{0, 1\} \ \forall j \in J_k \tag{15b}$$

$$\sum_{i \in I_k} x_i + \sum_{j \in J_k} y_j^k \leq T \tag{15c}$$

3.1. Computational advantages of grid-based layout optimization

The Lagrangian subproblem in Equation (15) reduces to a linear knapsack problem when $\boldsymbol{x}_{I_k}$ is fixed. Therefore, finding the optimal solution to the Lagrangian subproblem involves solving a linear knapsack problem with respect to $\boldsymbol{y}_{J_k}$ for every combination of values in $\boldsymbol{x}_{I_k}$ (Billionnet *et al.* 1999). Note that the coefficients of the knapsack inequality in Equation (15c) are not ones in general QKPs, so using the standard greedy algorithm to solve the linear knapsack problem will only give an upper bound on the optimal solution. This is obviously not the case for grid-based layout optimization, where the greedy algorithm will give the integer optimal solution to the Lagrangian subproblem, leading to a better BFS bound.

Intuitively, the performance of the BFS bound improves as the number of partitions decreases (the BFS bound and optimal value are equal when there is only one partition). However, the computational complexity of solving the Lagrangian subproblem using the enumerative approach mentioned is $O(2^{|X_k|})$ for general QKPs, which prevents each partition X_k from becoming too large. The QKPs in grid-based layout optimization are not restricted as much by this issue since infeasible grid point pairs are penalized. If the penalty is large enough that the optimal solution to the Lagrangian subproblem in Equation (12) cannot contain infeasible pairs from the same partition, then only feasible combinations of points in $\boldsymbol{x}_{I_k}$ need to be considered in the enumerative process of finding the optimal solution to the Lagrangian subproblem.

Infeasible grid point pairs from the same partition can be ignored if the penalty term P for any infeasible pair ij from partition k satisfies the criterion shown in Equation (16). The explanation for Equation (16) is similar to the explanation for Equation (3). Suppose the optimal solution to the kth Lagrangian subproblem contains an infeasible pair of grid points from X_k; Equation (16) shows that it is possible to obtain a better solution by removing one of the points from the infeasible pair, leading to a contradiction.

$$2P > \max_{i\in I_k}\left\{c_{ii} + \sum_{h\neq k}\lambda_i^h\right\} + (T-2)\max_{\substack{i\in I_k, i'\in I_k, j\in J_k, j'\in J_k\\ j>i, j'<i}}\left\{0, c_{ii'}, c_{ij}+\mu_{ij}, c_{ij'}-\mu_{ij'}\right\} \tag{16}$$

The criterion in Equation (16) depends on the value of the Lagrangian multipliers in every iteration of the BFS algorithm so P needs to be first set to some value that is at least as large as the criterion shown in Equation (3). The criterion in Equation (16) is then checked after the BFS algorithm to determine whether P was set large enough. The Lagrangian subproblems are solved using the greedy algorithm so P can be set to some arbitrarily large constant that also satisfies Equation (3) (feasibility condition) without any numerical issues.

The elimination of infeasible pairs from the evaluation of the Lagrangian subproblem allows for larger partitions and consequently better bounds without significantly increasing the computational complexity of the BFS bound. This property, together with the fact that the greedy algorithm gives integer optimal solutions to the Lagrangian subproblem for fixed $\boldsymbol{x}_{I_k}$, is why the BFS bound is particularly well-suited for QKP problems in grid-based layout optimization.

4. Numerical experiments

4.1. BFS algorithm

The detailed steps of calculating the BFS bound are described in Algorithm 1. Let $\boldsymbol{\lambda}(t)$ (or $\boldsymbol{\mu}(t)$) be the vector of $\boldsymbol{\lambda}$ (or $\boldsymbol{\mu}$) Lagrangian multipliers at iteration t. The rest of the notation used in the algorithm's description follows from Section 3.

The initial upper bound UB_0 is set to some large multiple β of the best lower bound LB. The best lower bound is obtained by starting the greedy algorithm at every grid point, and then selecting the best solution. The BFS algorithm (Algorithm 1) runs until the iteration limit t_{lim} is reached.

The multiplier update step in the BFS algorithm is based on the work done by Held *et al.* (1974). Ideally, the numerator in the subgradient coefficient should be $UB_t - f^*$, where f^* is the optimal solution to the primal problem. The implementation of the multiplier update step in this article replaces f^* with γLB, where LB is the objective function value of the best greedy algorithm solution, and γ is a number slightly larger than one. The step size parameter α, and the practice of decreasing α whenever the algorithm fails to improve the upper bound for $consec_{\text{lim}}$ consecutive iterations was proposed by Caprara *et al.* (1999).

4.2. Greedy algorithm performance

The BFS bound was used to assess the performance of the greedy algorithm for a number of experiments, as shown in Table 2. Every wind farm size and turbine count combination was paired with two wind profiles and two grid configurations, giving a total of 24 experiments.

Wind profiles 1 and 2 each have 360 directions with equal probability. According to Feng and Shen. (2015), using a large number of wind directions gives a more realistic assessment of wind farm

Table 2. Experiment settings.

Wind farm dimensions	Turbines	Wind profiles	Grid configurations
1120 × 1200 m (small)	9, 14	1, 2	Sparse, dense
1760 × 1760 m (medium)	25, 32		
2400 × 2400 m (large)	49, 56		

Algorithm 1 BFS algorithm.

```
Let $LB$ be the best lower bound, and $UB_t$ be the BFS bound at iteration $t$
$t = 0;\ UB_0 = \beta LB;\ consec = 0;\ \alpha = 2$
$\lambda_j^k(0) = 0,\ \forall j \in J_k,\ \forall k \in \{1, \ldots, p\}$
$\mu_{ij}(0) = 0,\ \forall i \in \{1, \ldots, n-1\},\ \forall j \in \{i+1, \ldots, n\}$
while $t < t_{\text{lim}}$ do
    Increment $t$ by 1
    for each $k \in \{1, \ldots, p\}$ do
        $(\boldsymbol{x}_{I_k}*, \boldsymbol{y}_{J_k}*) = \arg\min_{\substack{\boldsymbol{x}_{I_k}, \boldsymbol{y}_{J_k} \\ \text{s.t.}(11c),(11d),(11e)}} L_k\left(\boldsymbol{x}_{I_k}, \boldsymbol{y}_{J_k}, \boldsymbol{\lambda}(t), \boldsymbol{\mu}(t)\right)$
    end for

    $UB_t = \sum_{k=1}^{p} L_k(\boldsymbol{x}_{I_k}^*, \boldsymbol{y}_{J_k}^*, \boldsymbol{\lambda}, \boldsymbol{\mu})$
    if $UB_t >= UB_{t-1}$ then increment $consec$ by 1
    else $consec = 0$
    end if

    Subgradient calculation:
    $\phi_j^k = x_j^* - y_j^{k*}$;
    $\delta_{ij} = x_i^* y_j^{cl(i)*} - x_j^* y_i^{cl(j)*}$

    if $consec = consec_{\text{lim}}$ then
        Decrease $\alpha$ by half; Set $consec = 0$
    end if

    Update $\boldsymbol{\lambda}$ multipliers for each $k \in \{1, \ldots, p\}$:
        $\lambda_j^k(t) = \lambda_j^k(t-1) - \alpha \frac{UB_t - \gamma LB}{\sum_{k=1}^{p} \sum_{j \in J_k} \left(\phi_j^k\right)^2 + \sum_{1 \le i < j \le n} \left(\delta_{ij}\right)^2} \phi_j^k$
    Update $\boldsymbol{\mu}$ multipliers:
        $\mu_{ij}(t) = \mu_{ij}(t-1) - \alpha \frac{UB_t - \gamma LB}{\sum_{k=1}^{p} \sum_{j \in J_k} \left(\phi_j^k\right)^2 + \sum_{1 \le i < j \le n} \left(\delta_{ij}\right)^2} \delta_{ij}$
end while
```

power generation. Wind speeds along all wind directions in both profiles are assumed to have Weibull distributions with shape parameter 2, and the average wind speeds shown in Figure 3. Both sparse and dense grid configurations have regularly spaced grid points, with the point-to-point distance being 160 m in the sparse configuration, and 80 m in the dense configuration.

The turbine model used for all experiments has a disc diameter D_0 of 80 m, and a rated power of 1.5 MW. A minimum inter-turbine distance of 320 m was enforced for all experiments. The power curve and thrust coefficient curve of the turbine used in the experiments are shown in Figure 4. The power curve is based on a reference turbine model developed by Jonkman *et al.* (2009), and the thrust coefficient curve is based on the curve used by Pérez *et al.* (2013).

The BFS algorithm was run for 1500 iterations for all of the 24 experiments. The average convergence history of the BFS bound for small, medium, and large wind farm experiments is shown in Figure 5. Convergence of the BFS bound was slower for larger wind farms, but 1500 iterations was sufficient for the BFS bound to reach a reasonable degree of convergence for all wind farm sizes.

The final optimality gaps, number of partitions, and the values of the γ parameter used in the BFS algorithm for the dense and sparse grid configurations are shown in Tables 3 and 4, respectively.

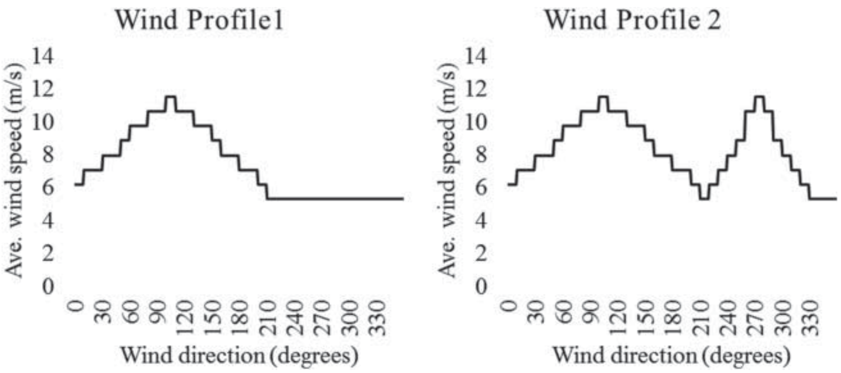

Figure 3. Average wind speeds in wind profiles 1 and 2.

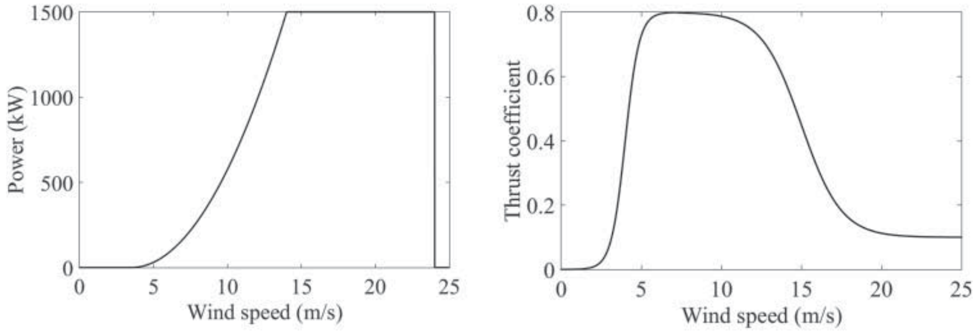

Figure 4. Turbine thrust coefficient curve, and power curves for different values of P_0.

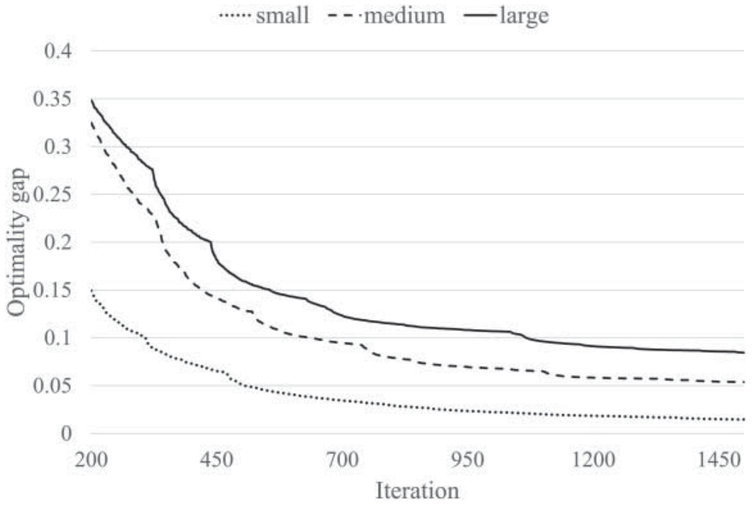

Figure 5. Average BFS bound convergence for small, medium, and large wind farms.

Note that the number of partitions used in the experiments is larger than what is possible for general QKPs, owing to the modifications proposed in Section 3.1 that allow increased partition sizes without significantly increasing the computational complexity of calculating the BFS bound.

Table 3. Dense grid experiment results.

					Optimality gap		
Wind farm	Turbines	Grid points	Partitions	γ	Wind profile 1	Wind profile 2	LP
1120 × 1120 m	9	225	8	1.00	0.022	0.021	0.051
1120 × 1120 m	14	225	8	1.00	0.032	0.035	0.115
1760 × 1760 m	25	529	18	1.01	0.079	0.072	0.136
1760 × 1760 m	32	529	18	1.01	0.109	0.100	0.211
2400 × 2400 m	49	961	30	1.02	0.127	0.116	0.230
2400 × 2400 m	56	961	30	1.02	0.142	0.131	0.296

Table 4. Sparse grid experiment results.

					Optimality gap		
Wind farm	Turbines	Grid points	Partitions	γ	Wind profile 1	Wind profile 2	LP
1120 × 1120 m	9	64	8	1.00	0.002	0.004	0.052
1120 × 1120 m	14	64	8	1.00	0.001	0.002	0.111
1760 × 1760 m	25	144	12	1.01	0.035	0.027	0.137
1760 × 1760 m	32	144	12	1.01	0.050	0.043	0.214
2400 × 2400 m	49	256	16	1.02	0.081	0.076	0.226
2400 × 2400 m	56	256	16	1.02	0.098	0.079	0.286

Let F_{UB} and F_{LB} denote the BFS bound and best greedy solution value, respectively. The optimality gap is then defined as

$$\frac{F_{UB} - F_{LB}}{F_{LB}}$$

The values under the 'LP' columns in Tables 3 and 4 are the averaged optimality gaps across both wind profiles obtained by replacing F_{UB} with the optimal value of the linear programming relaxation of the QKP formulation (Equation (1)). The QKP formulation can be converted to an equivalent MILP, as shown by Turner *et al.* (2014). The integer requirements in the MILP can then be relaxed to obtain an upper bound on the QKP's optimal value. The LP optimality gaps are provided to serve as a baseline for evaluating the performance of the BFS bound.

The results show that the optimality gaps for wind profile 2 are usually smaller than the optimality gaps for wind profile 1, especially as problem size increases. One possible explanation is that the square wind farms used in the experiments are better suited to wind profile 2, with two orthogonal primary wind directions, than to wind profile 1, with only one primary wind direction, which would be better suited to a rectangular wind farm. As demonstrated by Chowdhury *et al.* (2014), the relation between wind farm shape and wind profile can have a significant effect on power output.

The optimality gaps in Tables 3 and 4 were averaged over the two wind profiles, and the results are shown in Figure 6. The greedy algorithm was able to produce near-optimal solutions for the smallest problems, with 64 grid points. As the number of grid points or turbines increased, a clear increase in optimality gaps was observed. When the number of grid points was fixed and turbine count increased, the optimality gap increased by 1.3% on average. Conversely, a much larger increase, of 4.1% on average, was observed when the turbine count was fixed and the number of grid points was increased.

The actual gap between the greedy solution value and the optimal value is likely to be less than the observed gap since the observed gap includes the BFS bound's overestimation of the optimal value. It is also likely that the accuracy of the BFS bound will degrade with increasing problem size. It is difficult to pinpoint the exact contribution of the BFS bound's overestimation toward the observed optimality gap, since optimal values of large problems are very difficult to obtain, if at all possible. However, for small to medium-sized wind farms with fewer than 300 grid points and 30 turbines or fewer, the observed gaps were small enough to suggest that the greedy algorithm is a capable alternative to optimal solution methods. Observed optimality gaps for experiments of that size were 1.8% on

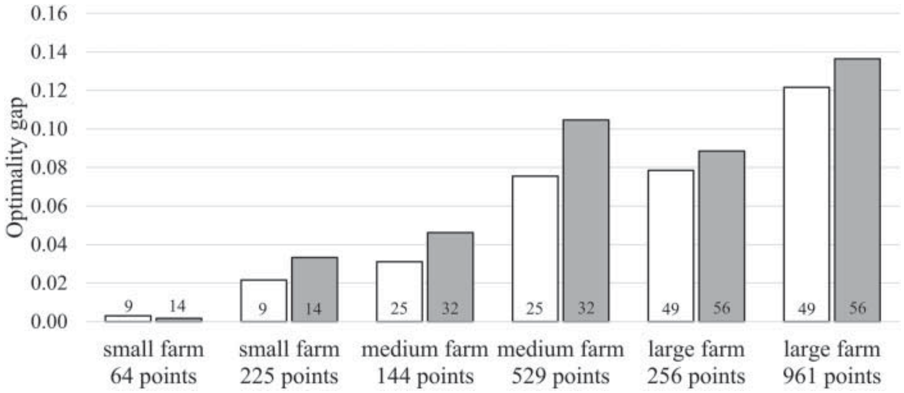

Figure 6. Average optimality gaps for various wind farm sizes, grid configurations, and turbine counts.

average, and less than 3.3% for all cases. If the designer is willing to accept a larger optimality gap of 8%, the greedy algorithm can be a suitable solution method for medium to large wind farms with up to 500 grid points or 50 turbines. When the number of grid points and turbines increases past that mark, the optimality gaps became too large to make any meaningful judgements on greedy algorithm performance.

5. Conclusions

The QKP formulation offers a simplified, graph-based approach toward grid-based wind farm layout optimization. The QKP formulation can be reduced to a mixed integer linear program that, theoretically, can be solved to optimality using branch and bound solvers. In practice, branch and bound solvers may struggle to solve even small QKP instances, so a heuristic like the greedy algorithm is a more reliable solution method, especially given the submodular nature of the QKP's objective function.

A theoretical lower bound on the optimality of the greedy solution exists for generic submodular maximization problems but, as this article has demonstrated, it is possible to obtain a much tighter and more useful optimality gap by adapting the BFS bound developed by Billionnet *et al.* (1999) to QKPs in grid-based layout optimization. This article demonstrates the use of the BFS bound to evaluate the optimality of greedy solutions for a range of experiments. One of the primary conclusions drawn from the experiments is that the greedy algorithm is a good alternative to optimal solution methods for small to medium-sized QKPs considered in this article, but the observed optimality gaps became too large for any meaningful judgements on greedy solution quality when problem sizes increased further.

The accuracy of the BFS bound could be improved by reducing the number of partitions even further, but this would drastically increase the computational complexity of calculating the bound as the number of linear knapsack problems that need to be solved for each evaluation of the dual function increases in a near-exponential manner with increasing partition size. This issue can be partly mitigated by the fact that evaluation of the dual function can be easily parallelized, so extra computational resources can be effectively used to cut computational times for large problem instances. The shape of the partitions could also be a factor that affects BFS bound accuracy and computational complexity. It was observed that narrower, rectangular partitions produced better bounds than square partitions of the same size. Future work could explore the effects of using more complicated partitioning schemes, such as non-regular partitions or disjointed partitions on the trade-off between BFS bound accuracy and computational complexity.

The BFS bound could also be used to reduce problem size by eliminating grid points that belong to far-from-optimal solutions. This concept has been applied successfully to general QKPs by Pisinger *et al.* (2007), and could be a promising method to reduce problem sizes of large QKPs in grid-based layout optimization to something more tractable for the greedy algorithm, or even branch and bound solvers, which can produce optimal solutions if the problem is small enough.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Bertsekas, D. P. 2008. *Nonlinear Programming*. Belmont, MA: Athena Scientific.

Billionnet, A, A. Faye, and E. Soutif. 1999. "A New Upper Bound for the 0-1 Quadratic Knapsack Problem." *European Journal of Operational Research* 112 (3): 664–672.

Caprara, A., D. Pisinger, and P. Toth. 1999. "Exact Solution of the Quadratic Knapsack Problem." *INFORMS Journal on Computing* 11: 125–137.

Chen, K., M. X. Song, Z. Y. He, and X. Zhang. 2013. "Wind Turbine Positioning Optimization of Wind Farm using Greedy Algorithm." *Journal of Renewable and Sustainable Energy* 5: 023128.

Chen, L., and E. MacDonald. 2014. "A System-level Cost-of-energy Wind Farm Layout Optimization with Landowner Modeling." *Energy Conversion and Management* 77: 484–494.

Chowdhury, S., J. Zhang, W. Tong, and A. Messac. 2014. "Modeling the Influence of Land-Shape on the Energy Production Potential of a Wind Farm Site." *Journal of Energy Resources Technology* 136 (1): 011203.

Dobrić, G., and Z. Duriić. 2014. "Double-stage Genetic Algorithm for Wind Farm Layout Optimization on Complex Terrains." *Journal of Renewable and Sustainable Energy* 6: 033127.

Feng, J., and W. Z. Shen. 2015. "Solving the Wind Farm Layout Optimization Problem using Random Search Algorithm." *Renewable Energy* 78: 182–192.

González, J. S., M. B. Payán, and J. M. Riquelme-Santos. 2012. "Optimization of Wind Farm Turbine Layout Including Decision Making Under Risk." *IEEE Systems Journal* 6 (1): 94–102.

Grady, S. A., M. Y. Hussaini, and M. M. Abdullah. 2005. "Placement of Wind Turbines using Genetic Algorithms." *Renewable Energy* 30 (2): 259–270.

Held, M., P. Wolfe, and H. P. Crowder. 1974. "Validation of Subgradient Optimization." *Mathematical Programming* 6: 62–88.

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

Jonkman, J., S. Butterfield, W. Musial, and G. Scott. 2009. *Definition of a 5-MW Reference Wind Turbine for Offshore System Development*. Technical report, no. NREL/TP-500-38060. Golden, CO: National Renewable Energy Laboratory.

Katic, I., J. Høstrup, and N. O. Jensen. 1986. "A Simple Model for Cluster Efficiency." In *Proceedings of the European Wind Energy Association Conference and Exhibition,* Vol. 1, edited by W. Palz and E. Sesto, 407–410. Rome: A. Raguzzi.

Krause, A., and D. Golovin. 2014. "Submodular Function Maximization." In *Tractability,* edited by L. Bordeaux, Y. Hamadi, and P. Kohli, 71–104. Cambridge, UK: Cambridge University Press.

Lackner, M. A., and C. N. Elkinton. 2007. "An Analytical Framework for Offshore Wind Farm Layout Optimization." *Wind Engineering* 31 (1): 17–31.

Long, H., and Z. Zhang. 2015. "A Two-Echelon Wind Farm Layout Planning Model." *IEEE Transactions on Sustainable Energy* 6 (3): 863–871.

Mosetti, G., C. Poloni, and B. Diviacco. 1994. "Optimization of Wind Turbine Positioning in Large Windfarms by Means of a Genetic Algorithm." *Journal of Wind Engineering and Industrial Aerodynamics* 51 (1): 105–116.

Nemhauser, G. L., and L. A. Wolsey. 1978. "An Analysis of Approximations for Maximizing Submodular Set Functions—I." *Mathematical Programming* 14: 265–294.

Park, J., and K. H. Law. 2015. "Layout Optimization for Maximizing Wind Farm Power Production using Sequential Convex Programming." *Applied Energy* 151: 320–334.

Pérez, B., R. Mínguez, and R. Guanche. 2013. "Offshore Wind Farm Layout Optimization Using Mathematical Programming Techniques." *Renewable Energy* 53: 389–399.

Pineda, I., G. Corbetta, A. Mbistrova, A. Ho, and K. Ruby. 2016. *Wind in Power, 2015 European Statistics*. Technical report. Brussels: The European Wind Energy Association.

Pisinger, D. 2007. "The Quadratic Knapsack Problem—A Survey." *Discrete Applied Mathematics* 155 (5): 623–648.

Pisinger, D. W., A. B. Rasmussen, and R. Sandvik. 2007. "Solution of Large Quadratic Knapsack Problems Through Aggressive Reduction." *INFORMS Journal on Computing* 19 (2): 280–290.

Réthoré, P. E., P. Fuglsang, G. C. Larsen, T. Buhl, T. J. Larsen, and H. A. Madsen. 2014. “TOPFARM: Multi-fidelity Optimization of Wind Farms.” *Wind Energy* 17 (12): 1797–1816.

Saavedra-Moreno, B., S. Salcedo-Sanz, A. Paniagua-Tineo, L. Prieto, and A. Portilla-Figueras. 2011. “Seeding Evolutionary Algorithms with Heuristics for Optimal Wind Turbines Positioning in Wind Farms.” *Renewable Energy* 36 (11): 2838–2844.

Song, M., K. Chen, X. Zhang, and J. Wang. 2015. “The Lazy Greedy Algorithm for Power Optimization of Wind Turbine Positioning on Complex Terrain.” *Energy* 80: 567–574.

Turner, S. D. O., D. A. Romero, P. Y. Zhang, C. H. Amon, and T. C. Y. Chan. 2014. “A New Mathematical Programming Approach to Optimize Wind Farm Layouts.” *Renewable Energy* 63: 674–680.

US Department of Energy. 2015. *Wind Vision: A New Era for Wind Power in the United States*. Oak Ridge, TN: US Department of Energy.

Wagner, M., J. Day, and F. Neumann. 2013. “A Fast and Effective Local Search Algorithm for Optimizing the Placement of Wind Turbines.” *Renewable Energy* 51: 64–70.

Zhang, C., G. Hou, and J. Wang. 2011. “A Fast Algorithm Based on the Submodular Property for Optimization of Wind Turbine Positioning.” *Renewable Energy* 36 (11): 2951–2958.

Zhang, P. Y., D. A. Romero, J. C. Beck, and C. H. Amon. 2014. “Solving Wind Farm Layout Optimization with Mixed Integer Programs and Constraint Programming.” *EURO Journal of Computational Optimization* 2 (3): 195–219.