Solving Wind Farm Layout Optimization with Mixed Integer Programming and Constraint Programming

Peter Y. Zhang, David A. Romero, J. Christopher Beck, Cristina H. Amon

Department of Mechanical and Industrial Engineering
University of Toronto
peteryun.zhang@mail.utoronto.ca, d.romero@utoronto.ca,
jcb@mie.utoronto.ca, cristina.amon@utoronto.ca

Abstract. The wind farm layout optimization problem is concerned with the optimal location of turbines within a fixed geographical area to maximize energy capture under stochastic wind conditions. Previously it has been modelled as a maximum diversity (or p-dispersion-sum) problem, but such a formulation cannot capture the nonlinearity of aerodynamic interactions among multiple wind turbines. We present the first constraint programming (CP) and mixed integer linear programming (MIP) models that incorporate such nonlinearity. Our empirical results indicate that the relative performance between these two models reverses when the wind scenario changes from a simple to a more complex one. We also propose an improvement to the previous maximum diversity model and demonstrate that the improved model solves more problem instances.

1 Introduction

Wind farm layout optimization problems deal with the optimal placement of turbines in a given wind farm field. Currently this problem appears only in the engineering research literature [1–6], where much effort has been spent on developing metaheuristics [1–3, 7] for variations of this problem. Some existing heuristic methods [1, 2] and mixed integer models [4, 5] have explored this problem with discretization: land is decomposed into a set of small cells, where each accommodates one turbine. Compared with a continuous approach [6], the discrete approach is less sensitive to discontinuity in the wind farm land. Such discontinuity is common in practice due to existing infrastructure and geographic constraints [8]. In the current work, without loss of generality, our problem instances are square wind farms with equal-size square cells.

An interesting feature of this problem that sets it apart from standard location problems is the aerodynamic interaction among multiple turbines. In a basic scenario where only two turbines are present, the turbine downstream is said to be in the wake region of the upstream turbine, and it experiences a loss in energy production due to the reduction in wind speed and increase in turbulence intensity [9]. In practice, a turbine that is downstream of multiple turbines

is affected by all upstream turbines simultaneously, and the overall effect is a nonlinear function of individual wakes. There are different analytical equations to describe the superposition of multiple wakes, some being closer to the physical reality than the others [10]. It is difficult to incorporate the more accurate wake equations into a mathematical programming model due to their nonlinearity: currently only heuristics [1–3, 6] include the most accurate wake models. Our goals are to computationally improve existing mixed integer programming (MIP) models and incorporate more accurate wake models into constraint programming (CP) and MIP models.

The contributions of this paper are: the proposal of two novel mathematical programming models (CP and MIP) that can describe the physics of the problem more accurately than the previous MIP models; the extension of a previous MIP model so that the solution quality and time are improved; the comparison of four models on twelve problem instances, with varying wind scenario complexity, turbine numbers, and wind farm grid resolution; and the elicitation of insights from the experiments to suggest future research directions.

2 Problem Definition and the Physics of Wake Modelling

2.1 Description of the Problem

Wind farm site selection, or wind farm siting, is based on, among other factors, meteorological conditions, topological features of the site, and accessibility for construction and grid transmission [8]. After siting, wind farm developers optimize the layout of the turbines according to prescribed objectives and constraints in a process called *micro-siting*. In a typical case, design engineers try to maximize the expected profit and minimize hazardous side-effects during wind farm construction and operation [8]. This is a challenging task because there are many objectives and constraints, and every site is different. To limit our scope, we consider the maximization of energy capture of a wind farm as our only objective, as it is closely related to the long term profit of the wind farms and it is well accepted in the wind farm optimization literature [1–3]. We further assume that the wind farm land is flat, and all turbines are of the same type.

We use the same problem setup that Mosetti et al. [1] proposed in their seminal paper. The objective is to maximize the wind farm's overall power generation capability. There are three types of constraints:

1. Proximity: turbines must be placed five diameters apart to avoid structural damage induced by strong aerodynamic interactions;
2. Boundary: Turbines must be placed within the wind farm boundaries;
3. Turbine number: The number of turbines is fixed.

The reason that the total number of turbines is fixed – instead of bounded by a maximum number of turbines – is due to practical considerations. During wind farm development, the total number of turbines is determined prior to the design process, by government regulations and the local electricity grid interconnection capacity among other factors. However, to explore the design space more fully,

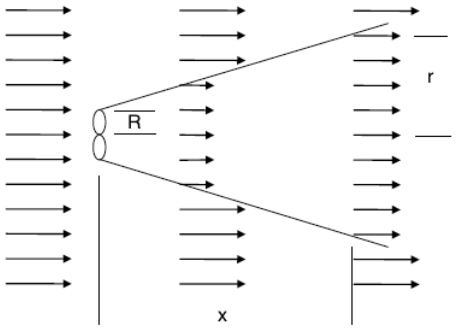

Fig. 1. Jensen [9] wake model.

a given model can always be solved multiple times with different numbers of turbines.

As mentioned in the previous section, we use a discrete representation of wind farm: land is decomposed into a set of cells, where each cell can only accommodate one turbine. This approach is common in the literature [1, 2, 4].

2.2 The Physics: Wake and Energy Models

While some constraints of this problem are similar to vertex packing [11], undesirable facility location [12], and circle packing problems [13], the objective function is unique to wind farm layout optimization. In particular, the energy capture at each turbine is proportional to the cube of wind speed at that location. In turn the wind speed at a turbine is a nonlinear function of the distances to its upstream turbines. Note that "upstream" is relative to the wind direction, which varies over time.

Although wind changes speed and direction frequently, we assume that the turbine can re-orient its rotor towards the upcoming wind direction. We further assume that there is no power loss during the transient states. Overall, the yearly wind frequency data at each direction fits well into a Weibull distribution [14]. In the literature, it is a common practice to discretize the yearly wind frequency data into multiple directions and multiple speeds [1–3], so that the total energy production is the weighted sum of energy produced at each wind state (speed and direction). Then the expected power is only different from the expected energy by a scalar (the number of seconds per year). Therefore we only deal with the expected power in this work to simplify calculations.

Single Wake The downstream region of a wind turbine, with increased level of turbulence and decreased energy, is called the wake region (Fig. 1). Equation (1), first proposed by Jensen [9], describes the propagation of a single wake.

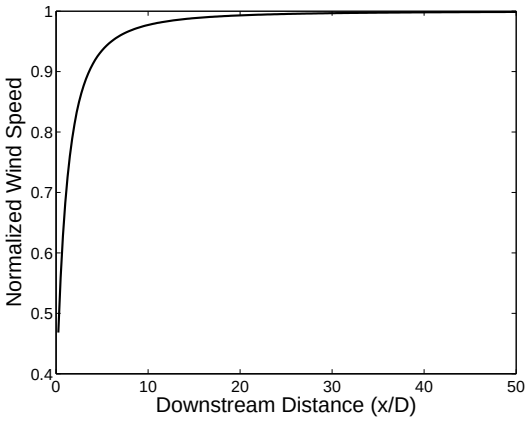

Fig. 2. Wind speed recovery after a turbine. D is the turbine diameter.

Parallel arrows in Fig. 1 represent wind direction and speed. The region with lower wind speed (shorter arrows) is the wake region. The two ellipses represent a turbine. The key idea of Jensen's wake model is momentum conservation within the wake region. In addition, wind speed is assumed to be uniform and non-turbulent across the circular wake cross-section.

Here R is the wake radius immediately after rotor; r is the downstream wake radius; r_0 is the rotor radius; u_∞ is the free stream wind speed; u_r is the wind speed immediately behind the rotor; u is the speed of wind at downstream distance x; α is the wake decay constant; z_0 is the roughness of terrain; z is the turbine height; and a is the axial induction factor (the percentage reduction in wind speed between the free stream and the turbine rotor) [15]:

$$\pi R^2 u_r + \pi \left(r^2 - R^2\right) u_\infty = \pi r^2 u \tag{1}$$

where $u_r = (1-a)u_\infty$, $r = R + \alpha x$, $R = r_0\sqrt{\frac{1-a}{1-2a}}$ and $\alpha = 0.5/\ln(\frac{z}{z_0})$.

Figure 2 describes the wind speed recovery after an upstream turbine, based on the previous equations.

Multiple Wakes: Sum-of-Squares Following Renkema [10], we write the effective wind speed in the wakes of multiple turbines as:

$$u_{id} = u_{id,\infty}\left[1 - \sqrt{\sum_{j \in \mathcal{U}_{id}} \left(1 - \frac{u_{ijd}}{u_{id,\infty}}\right)^2}\right] \tag{2}$$

u_{id} and $u_{id,\infty}$ are the wind speeds at turbine i at wind state (speed and direction) d with and without wake interactions respectively, where $d \in \mathcal{D}$, the set of all

possible wind states; $\mathcal{U}_{id}$ is the set of upstream turbines for turbine i at wind state d; u_{ijd} is the wind speed at turbine i due to a single wake from upstream turbine at j, which can be obtained by (1). Currently, this is the most accurate analytical expression accounting for multiple wakes [10].

Based on this model, the expected power production of the wind farm can be calculated as:

$$\text{expected power} = \sum_{i=1}^{m} \sum_{d \in \mathcal{D}} \frac{1}{3} u_{id}^3 p_d \tag{3}$$

where m is the total number of turbines and p_d is the probability of wind state d occurring, subject to $\sum_{d \in \mathcal{D}} p_d = 1$.

Note that we will be using average power (watts) instead of total energy (kilo-watt hours) in the objective function because they are equivalent for our purpose, and the former is easier to represent.

Multiple Wakes: Linear Superposition Another way to account for multiple wakes in the energy production calculation is to use a direct linear superposition of power deficits. This is known to be less accurate than (2). However, it is more easily representable in the mathematical programming models [4, 5], because we can pre-calculate the pairwise interactions between two locations, then "activate" the interactions with binary variables indicating the existence of turbines at those locations, and sum up the interactions linearly:

$$\text{expected power} = \sum_{i=1}^{n} \sum_{d \in \mathcal{D}} \left(\frac{1}{3} u_{id,\infty}^3 - \sum_{j \in \mathcal{U}_{id}} \frac{1}{3} \left(u_{id,\infty}^3 - u_{ijd}^3 \right) \right) p_d \tag{4}$$

and again u_{ijd} can be obtained from (1).

With the physics introduced, we want to make a note on the representation of these equations in our optimization models. Although the power calculation equations (2) and (4) appear to be nonlinear in wind speeds, we can actually remove some of the nonlinearity due to the choice of discrete optimization models. As illustrated by Donovan [4], the linear superposition model (4) is completely linear, because all the wind speed terms can be calculated prior the optimization, since the candidate turbine locations (i, j) and wind states $\mathcal{D}$ are known in the discrete representation.

However, linearizing the wake model given by (2) is a non-trivial task, even though the wind speed terms can be pre-calculated. The next section will first introduce a CP model that directly represents the nonlinearity in its objective function, and then describe our novel approach that can incorporate the physics of (2) into a mixed integer linear program.

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{n} \sum_{d \in \mathcal{D}} \frac{1}{3} x_i \left(u_{id,\infty} \left[1 - \sqrt{\sum_{j \in \mathcal{U}_{id}} x_j \left(1 - \frac{u_{ijd}}{u_{id,\infty}} \right)^2} \right] \right)^3 p_d \\
\text{subject to} \quad & \sum_{i=1}^{n} x_i = k \\
& x_i + x_j \leq 1 \quad \forall (i,j) \in \mathcal{E} \\
& x_i \in \{0,1\} \quad \forall i = 1, \ldots, n.
\end{aligned}
$$

Fig. 3. SOM1: a constraint programming model.

3 Optimization Models

3.1 Sum-of-Squares Optimization Models (SOM)

The following three optimization models are based on the more accurate way of accounting for multiple wakes (2).

CP and MIP Models. Figure 3 presents the SOM1 CP model. The binary decision variable x_i represents whether there is a turbine at location i; n is the total number of grid points; k is the total number of turbines; and $\mathcal{E} = \{(i,j) |$ grid i and grid j cannot both have turbines due to proximity constraint$\}$. This set is determined by the proximity constraint and the grid resolution. We choose equality for the constraint $\sum_{i=1}^{n} x_i = k$ for practical reasons – the total number of turbines is usually determined prior to the optimization based on project financing and government regulations. For the problem instances used in this work, the optimal energy production is an increasing function of k [16].

In Figure 4, we present SOM2, a MIP sum-of-squares model where the non-linearity is dealt with via a potentially exponential number of constraints. The auxiliary variable z_i represents the average power production at each location i. The key of this model are the constraints indicated by $(*)$. M is a sufficiently large constant. In general, $w_{i,\mathcal{S}_i}$ is the maximum amount of power convertible when all cells with indices in $\mathcal{S}_i$ have turbines and all cells with indices in $\mathcal{I} \setminus \mathcal{S}_i$ do not; $\mathcal{I}$ is the set of all turbine location indices $\{1, \ldots, n\}$; and $\mathcal{S}_i$ is a set of turbine locations not including i. $w_{i,\mathcal{S}_i}$ is calculated according to (2) and (3).

A Decomposition Model. It is not hard to see that the number of constraints $(*)$ is exponential in n due to the requirement $(\forall \mathcal{S}_i \subset \mathcal{I} \setminus i)$. Therefore, rather than experimenting with SOM2, we propose a third model, SOM3, which can be understood as a decomposition of SOM2.

In Fig. 5, a MIP master problem is formulated that includes all constraints of SOM2 except for those indicated with $(*)$. After solving the master problem, a sub-problem calculates the actual power according to (2) and (3) as follows:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{n} z_i \\
\text{subject to} \quad & \sum_{i=1}^{n} x_i = k \\
& x_i + x_j \leq 1 \quad \forall (i,j) \in \mathcal{E} \\
& z_i \leq \sum_{i=1}^{n} \sum_{d \in \mathcal{D}} \frac{1}{3} u_{id,\infty}^3 p_d x_i \quad \forall i = 1, \ldots, n. \\
& z_i \leq M \left(|\mathcal{S}_i| - \sum_{j \in \mathcal{S}_i} x_j \right) + w_{i,\mathcal{S}_i} \quad \forall \mathcal{S}_i \subset \mathcal{I} \setminus i \quad (*) \\
& x_i \in \{0,1\} \quad \forall i = 1, \ldots, n.
\end{aligned}
$$

Fig. 4. SOM2: a mixed integer programming model.

1. Evaluate the turbine layout power considering full wake effects based on $\mathbf{x}^t$ (the solution from the master problem at iteration t) by substituting it into
$$\sum_{i=1}^{n} \sum_{d \in \mathcal{D}} \tfrac{1}{3} x_i^t \left(u_{id,\infty} \left[1 - \sqrt{\sum_{j \in \mathcal{U}_{id}} x_j^t \left(1 - \tfrac{u_{ijd}}{u_{id,\infty}} \right)^2} \right] \right)^3 p_d$$
2. If it is evaluated to the same as the objective value from the master problem or the maximum solution time is reached, terminate; otherwise:
3. Generate cuts in the form of $z_i \leq g_i(\mathbf{x}^t)$, where $g_i(\mathbf{x}^t)$ is defined by (5) and (6); return to the master problem.

The master problem is then re-solved with the new cuts. In the first iteration, the master problem assumes that there is no wake interaction at all. In each subsequent iteration, the cuts refine the modeling of turbine interactions. The master problem does not represent the interaction of a specific group of turbines unless the related cuts are added. Therefore, the master problem always overestimates the true objective value.

Instead of solving the master problem to optimality, we run it with a time limit of T seconds. In our experiment, we choose $T_0 = 30$ seconds for the first iteration. T is increased by 5 seconds each time the current best master solution is the same as the previous iteration. In other words, if the master problem produces the same solution as the previous iteration and it does not converge to the subproblem value, the algorithm will keep running with no new cuts generated, therefore getting stuck in a loop. This will happen if the master problem is unable to make any new progress in a new iteration (compared with the previous iteration) within the prescribed time limit.

Cuts We propose two types of cut: a *no-good cut* and a *3-cut*. The former is presented in Equation (5). M is a large constant; x_j^t is the jth component of $\mathbf{x}^t$;

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{n} z_i \\
\text{subject to} \quad & \sum_{i=1}^{n} x_i = k \\
& x_i + x_j \leq 1 \quad \forall (i,j) \in \mathcal{E} \\
& z_i \leq \sum_{i=1}^{n} \sum_{d \in \mathcal{D}} \frac{1}{3} u^3_{id,\infty} p_d x_i \quad \forall i = 1, \ldots, n. \\
& \text{(cuts)} \\
& x_i \in \{0,1\} \quad \forall i = 1, \ldots, n.
\end{aligned}
$$

Fig. 5. SOM3: A mixed integer programming model of the master problem.

$w_{i,\mathcal{A}}$ is the maximum amount of power convertible when all cells with indices in $\mathcal{A}$ have turbines; $\mathcal{S}_i = \{j \mid x^t_j = 1\}$; w_i, $w_{i,j}$ and $w_{i,jk}$ are short forms for $w_{i,\emptyset}$, $w_{i,\{j\}}$ and $w_{i,\{jk\}}$, following the definition of $w_{i,\mathcal{A}}$.

$$
g_i(\mathbf{x}^t) = M \left(|\mathcal{S}_i| - \sum_{j \in \mathcal{S}_i} x_j \right) + w_{i,\mathcal{S}_i} \tag{5}
$$

In practice, the no-good cuts alone are inefficient in large problem instances, because an exponential number of them are required to correctly shape the feasible region and the information of each cut is minimal when there are many wind states and location cells. Therefore, we propose another type of cut to increase the speed of refinement of the representation of turbine interactions. Equation (6) presents the 3-cuts.

$$
g_i(\mathbf{x}^t) = w_i + (w_{i,j} - w_i)x_j + (w_{i,jk} - w_{i,j})x_k \quad \forall j,k \in \mathcal{S}_i \ . \tag{6}
$$

For each downstream turbine i, there are $\binom{|\mathcal{S}_i|}{2}$ cuts generated. The power of 3-cuts lie in their accurate description of the interaction between a group of three turbines (thus the name 3-cut). In practice, the closest few upstream turbines have the most significant influence on a downstream turbine (see Fig. 2).

The following proposition states that the three-turbine interaction accurately describes the feasible region:

Proposition 1. *The cut $z_i \leq w_i + (w_{i,j} - w_i)x_j + (w_{i,jk} - w_{i,j})x_k$ is tight (cutting off all infeasible values for z_i assuming no turbines are "on" except for j,k) at $(x_i, x_j, x_k) = (1,0,0)$, $(1,1,0)$, and $(1,1,1)$.*

Proof. When $(x_i, x_j, x_k) = (1,0,0)$, $(1,1,0)$, and $(1,1,1)$, the cut reduces to $z_i \leq w_i$, $z_i \leq w_{i,j}$, and $z_i \leq w_{i,jk}$ respectively. These values are tight by definition

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{n} \sum_{d \in \mathcal{D}} \left(\frac{1}{3} u^3_{id,\infty} x_i - \sum_{j=1}^{n} \frac{1}{3} \left(u^3_{id,\infty} - u^3_{ijd} \right) y_{ij} \right) p_d \\
\text{subject to} \quad & \sum_{i=1}^{n} x_i = k \\
& x_i + x_j \leq 1 \quad \forall (i,j) \in \mathcal{E} \\
& x_i + x_j - 1 \leq y_{ij} \quad \forall i,j = 1,\ldots,n. \\
& y_{ij} \geq 0 \quad \forall i,j = 1,\ldots,n. \\
& x_i \in \{0,1\} \quad \forall i = 1,\ldots,n.
\end{aligned}
$$

Fig. 6. LSOM1 [4]

of w_i, $w_{i,j}$, and $w_{i,jk}$. When $(x_i, x_j, x_k) = (1, 0, 1)$, the cut reduces to $z_i \leq w_i + w_{i,jk} - w_{i,j}$. Since $z_i \leq w_{i,k}$ by the definition of $w_{i,k}$, and the combination of power deficits is sub-linear (2 and 3), $w_i + w_{i,jk} - w_{i,j} \geq w_{i,k}$. Therefore $z_i \leq w_i + w_{i,jk} - w_{i,j}$ is not tight and does not cut off any feasible region. □

Overall, the no-good cuts ensure that the problem can eventually reach the true optimality while 3-cuts increase the communication between subproblem and master to speed up convergence.

3.2 Linear Superposition Optimization Models (LSOM)

Previous MIP models use a simpler (and less accurate) [10] calculation of energy: the power deficits from individual wakes are combined linearly to account for the total power loss. The following two MIP models are based on such linear superposition technique. The first model (LSOM1) was originally proposed by Donovan [4], while the second one (LSOM2) is our extension of LSOM1.

Figure 6 presents the LSOM1 model where $\frac{1}{3}\left(u^3_{id,\infty} - u^3_{ijd}\right)$ is the power reduction at wind state d at turbine i due to the presence of upstream turbine j. These values can be calculated prior to running the optimization (4). Variable y_{ij} indicates whether there are turbines at both positions i and j, and so y_{ij} is 1 if both x_i and x_j are 1, and 0 otherwise. $\mathcal{E}$ is the set of cell pairs (i, j) that are too close to both host turbines. Due to the use of the simpler linear superposition model of upstream turbines, the model over-estimates the energy deficit [10].

Other location problems in the literature such as the maximum diversity problem (MDP) [17] and the p-dispersion-sum (pDS) problem [18] are similar to Donovan's model. However, we have not seen any application of the state-of-art MDP/pDS solution algorithms to this wind farm layout optimization model.

In LSOM1, the y_{ij} variables are always equal to the product of x_i and x_j, indicating if there are turbines at both places. Since $i, j \in \mathcal{I}$, there are in total $|\mathcal{I}|^2$ y_{ij} variables. As our experiments below demonstrate, a high resolution of the

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{n} z_i \\
\text{subject to} \quad & \sum_{i=1}^{n} x_i = k \\
& x_i + x_j \leq 1 \quad \forall (i,j) \in \mathcal{E} \\
& z_i \leq \sum_{i=1}^{n} \sum_{d \in \mathcal{D}} \frac{1}{3} u_{id,\infty}^3 p_d x_i \quad \forall i = 1, \ldots, n. \quad (\dagger) \\
& z_i \leq \sum_{i=1}^{n} \sum_{d \in \mathcal{D}} \left(\frac{1}{3} u_{id,\infty}^3 - \sum_{j=1}^{n} \frac{1}{3} \left(u_{id,\infty}^3 - u_{ijd}^3 \right) x_j \right) p_d \quad \forall i = 1, \ldots, n. \quad (\dagger\dagger) \\
& x_i \in \{0,1\} \quad \forall i = 1, \ldots, n.
\end{aligned}
$$

Fig. 7. LSOM2

wind farm grid with a complicated wind regime results in too many y_{ij} variables for reasonable performance. To address this weakness we propose LSOM2.

Figure 7 presents the LSOM2 model. It does not have y_{ij} variables. Instead, we use z_i to represent the power production at location i. If there is no turbine at location i, then the right hand side of constraint (†) is zero, and in most cases it is tighter than (††). So, if no turbine appears at i, then there will be no power production from location i. If there is a turbine at i, then in general constraint (††) is tighter due to the extra negative terms (deduction of power due to upstream turbines). The value that z_i is, therefore, calculated by the total available power subtracting the linear combination of power losses due to wakes.

However, LSOM2 is not equivalent to LSOM1. When $x_i = 0$ and constraint (†) is $z_i \leq 0$, constraint (††) may become $z_i \leq -c$, where $-c$ is a negative value. This is because the linear superposition model over-estimates power losses, making it possible for the right hand side of (††) to be negative. Fortunately, this case does not arise during our experiments due to the proximity constraint such that turbines cannot be less than 5 rotor-diameters apart. This value is an industry standard for wind farm design.

4 Experiment Setup

All models were implemented with Microsoft Visual C++ Express 2010 and IBM ILOG CPLEX 12.3. Twelve benchmark instances [1–3, 6, 16] (referred to as WRq-n-k) were used to test the performance of models. WR1 refers to the wind regime of 1 directional wind (from west to east) and WR36 refers to the wind regime with wind coming from 36 directions at different speeds (Fig. 8).

Experiments were run on a Dell Vostro 460 with Core i5-2500 CPU (3.30GHz) and 64-bit Windows 7 OS. Since CPLEX solvers are deterministic by default,

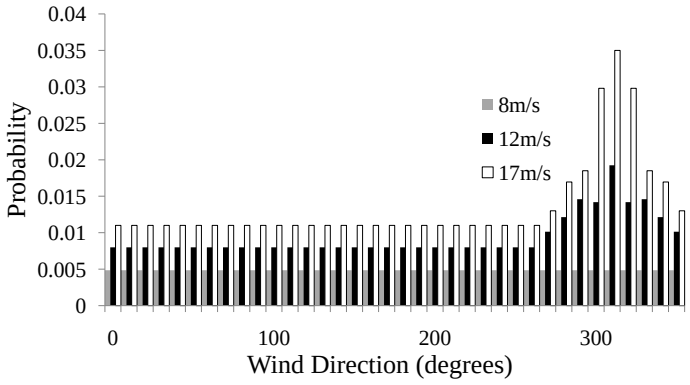

Fig. 8. Cumulative wind probability distribution for problem instance WR36.

only one run of each instance was performed. Common parameters are: $z = 60$m, $z_0 = 0.3$m $R = 20$m, and wind farm is 2km by 2km.

Since the SOM models evaluate the power production by sum-of-squares (2) and the LSOM evaluation is based on linear superposition (4), the four models are not directly comparable. We therefore compare the solution quality in Table 1 by a posteriori re-evaluating the LSOM solutions based on the (more accurate) sum-of-squares method. The power production values in brackets indicate the objective function value of the LSOM solutions.

5 Results

Table 1 summarizes the performance of the four models by comparing the expected power and solution times. The MIP optimality gaps are included where applicable. Columns n and k represent the total numbers of cells and turbines. For LSOM1 and LSOM2, there are two power values: the power calculated with the sum-of-squares wake model and, in parentheses, the objective function value based on linear superposition model. Overall, LSOM2 outperforms the other models in most cases in terms of solution quality. SOM3 and LSOM2 can solve problem instances with high grid resolution and high wind data resolution (WR36-400), while the other two models cannot even initialize these instances within an hour due to the size of the model.

SOM1 vs. SOM3 Table 1 shows that SOM3 solves more instances than SOM1. In the higher resolution case (WR36-400), the SOM1 objective function expression must account for many wind directions and turbine pairs leading to memory saturation during the model creation phase and the inability to *start* search within an hour. SOM3 also performs better in WR1 cases. We believe that the

simpler turbine interactions for WR1 instances are accurately captured by no-good and 3-cuts.

For a more detailed examination of these results, Figs. 9-11 present the evolution of solution quality over time for SOM1 and SOM3 for selected problem instances. In the single-direction scenario (WR1), SOM3 consistently outperforms SOM1. For the WR36 instances where SOM1 was able to run, SOM3 performs much worse than SOM1 (Fig. 11). To understand these results, recall that SOM3 is a decomposed model where the improvement from iteration to iteration is based on cuts representing information of turbine interactions. In the WR1 cases, every cell has about 10 (20) upstream cells, because the wind farm resolution is 10 by 10 (20 by 20). During the optimization, the better layouts often have turbines spaced out in the wind direction, thus the 3-cut, although only describing the interactions between a few turbines, already contains enough information for the master problem to make good decisions. However, in the WR36 cases, every turbine has $k-1$ upstream turbines and the 3-cut only expresses the impact of the most significant two upstream turbines. Therefore, SOM3's search for better objective value in WR36 instances is often stalled due to the lack of effective cuts.

Wind Regime	n	k	SOM1		SOM3		LSOM1			LSOM2		
			Power (W)	Sol. Time (s)	Power (W)	Sol. Time (s)	Power (W)	Sol. Time (s)	Opt. Gap (%)	Power (W)	Sol. Time (s)	Opt. Gap (%)
WR1	100	20	10253.4	3600	**10256.0**	3600	**10256.0** (10256.0)	3.5	0	**10256.0** (10256.0)	8.0	0
	100	30	14732.5	3600	**14800.9**	3600	14795.1 (14702.2)	2.5	0	14798.6 (14702.2)	3600	1.2
	100	40	18459.5	3600	**18674.5**	3600	**18674.5** (18186.3)	1.0	0	**18674.5** (18186.3)	98.8	0
	400	20	**10368.0**	3600	**10368.0**	3600	**10368.0** (10368.0)	2.5	0	**10368.0** (10368.0)	0.4	0
	400	30	15359.6	3600	15410.9	3600	15407.6 (15409.3)	3600	0.9	**15414.2** (15414.2)	3600	0.9
	400	40	20270.0	3600	20334.7	3600	**20353.6** (20118.4)	3600	3.1	20341.2 (20172.4)	3600	2.8
WR36	100	20	16675.1	3600	16631.0	3600	16705.2 (16356.6)	3600	3.0	**16706.7** (16365.2)	3600	3.2
	100	30	24574.0	3600	24391.0	3600	24597.3 (23442.3)	3600	7.3	**24625.1** (23520.5)	3600	7.4
	100	40	32204.6	3600	31616.0	3600	32197.3 (29651.9)	3600	9.9	**32310.7** (29904.0)	3600	11.9
	400	20	∘	∘	16551.0	3600	∘	∘	∘	**16762.3** (16470.1)	3600	3.1
	400	30	∘	∘	24426.0	3600	∘	∘	∘	**24870.4** (24027.1)	3600	5.9
	400	40	∘	∘	31977.0	3600	∘	∘	∘	**32715.4** (30844.9)	3600	10.0

Table 1. Comparison of solutions based on sum-of-squares power calculation (∘: experiments took more than one hour to setup; boldface: better objective value). Numbers in brackets are the original obj. values (based on linear superposition method) of LSOM solutions.

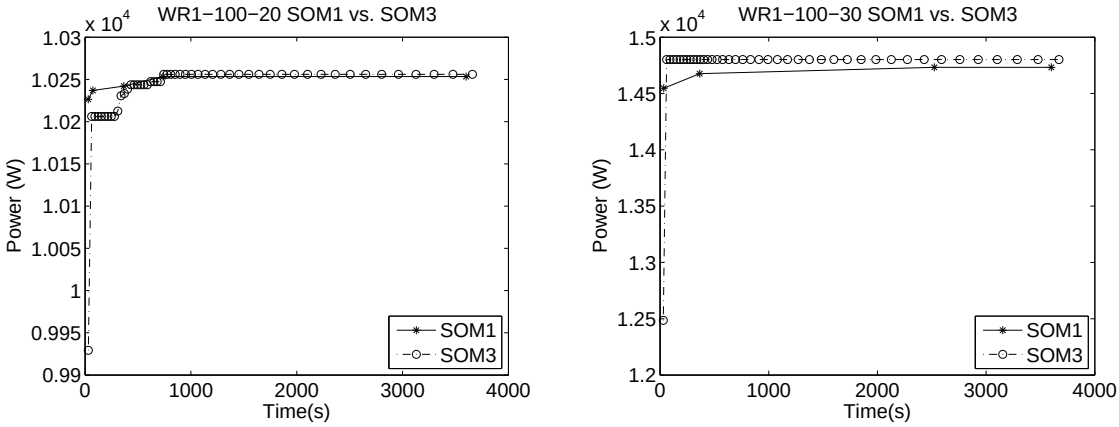

Fig. 9. Single wind direction, 100 cells.

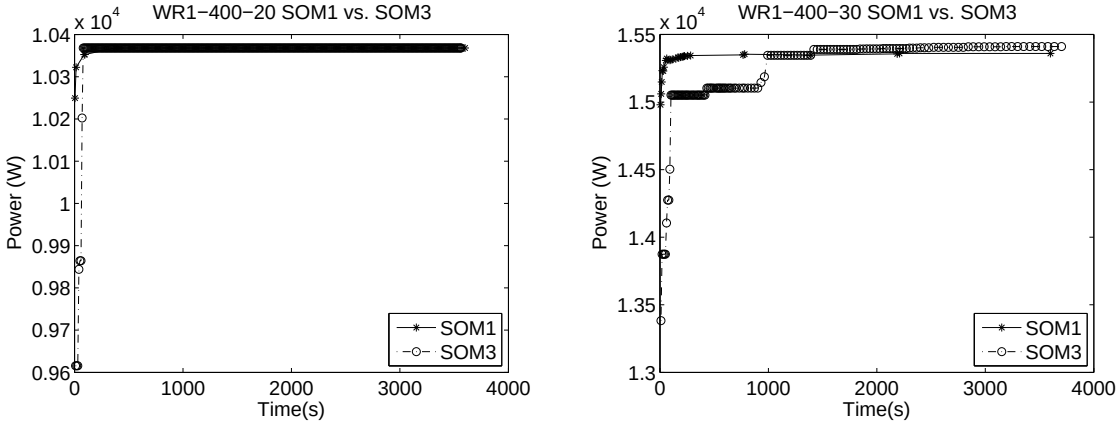

Fig. 10. Single wind direction, 400 cells.

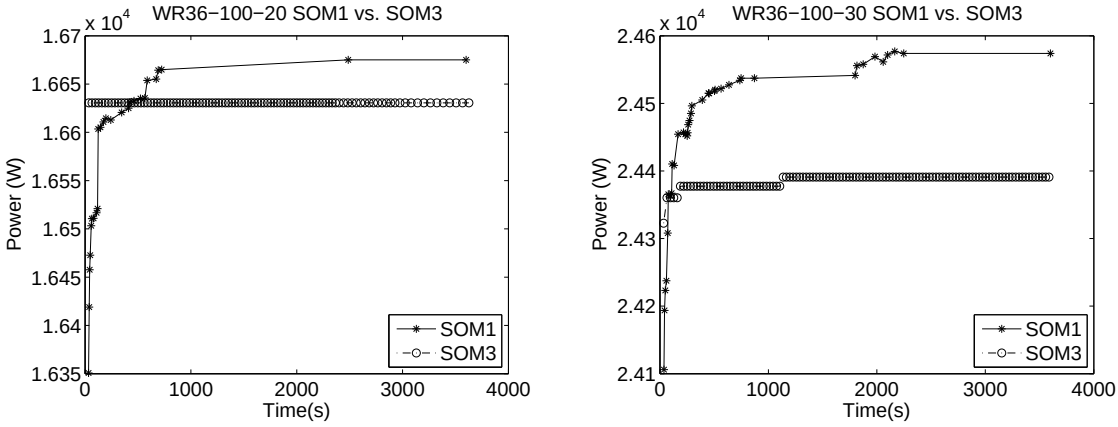

Fig. 11. Thirty-six wind directions, 100 cells.

SOM3 also improves more slowly in the WR1-400 instances (Fig. 10) than in the WR1-100 instances (Fig. 9). There are more combinations of three turbines in the former instances and therefore more 3-cuts must be generated for the master problem to improve its objective value. Eventually SOM3 catches up with SOM1 because the 3-cuts describe the interactions of all turbines reasonably accurately.

LSOM1 vs. LSOM2 Table 1 shows that the power production calculated by linear superposition method (in brackets) is always lower than the sum-of-squares calculation. For some values of n and k, the problem cannot be solved to optimality within an hour, while some other instances are solved in less than 10 seconds. This observation confirms with other work on MDP and p-dispersion-sum problems [19]: increase in n with fixed k will often lead to longer solution time, and increase in k while n is fixed often leads to longer solution time too (except for when k is close to 0 or n). Current state-of-art MDP algorithms are benchmarked on problem instances with similar dimensions (in n and k) as our instances [19].

Table 1 clearly shows that LSOM2 outperforms LSOM1 in solution quality in all but one instance (WR1-400-40). In the WR1-400-40 case, LSOM1 outperforms LSOM2 in terms of the revised power calculation, however, LSOM2 is "misled" by the objective function (in brackets). Thus if we compare the true optimization objective for LSOM1 and LSOM2, LSOM2 strictly dominates LSOM1 in solution quality. In terms of computation time, LSOM2 performs similarly to LSOM1 except the WR1-100-30 and WR1-100-40 instances. A closer look at the CPLEX solution log reveals that LSOM2 arrived at the actual optimal solution (proven by LSOM1) within a few seconds for both cases, but was unable to quickly tighten the dual bound.

As with the SOM1 model, the bigger cases (WR36-400) cannot be solved by LSOM1 within 1 hour because the CPLEX solver either took a few hours to initialize or could not start at all.

6 Discussion

LSOM1 represents the state-of-art solution model for this wind farm layout optimization problem. Our LSOM2, an extension of LSOM1, outperforms it and the other two models we proposed on most instances. Since the SOMs represent the first time that the most accurate analytical wake equations [10] are modelled with constraint programming and mixed integer programming, there is much to learn about the performance and potential opportunities for the SOMs. We describe several promising research directions.

For SOM1, nonlinearity appears only in the objective function, thus we could apply nonlinear solvers that are based on linear solvers (e.g., SCIP [20]). The SOM3 cuts capture information in two ways: no-good cuts capture interaction between all k turbines in very specific layouts, while the 3-cuts capture information from a wider range of layouts, but limited to the interaction among three turbines. We can potentially apply the same idea of 3-cut and generate

constraints that inform the master problem more effectively, without having to generate too many of them (e.g., 4,5-cut).

A straightforward hybridization would be the sequential application of SOM1 and SOM3, where we start the problem with low resolution (coarse grid) and progressively increase it. In this case, we could solve SOM1 in the initial stages, utilizing the constraint propagation of CP solvers for the proximity constraints, and then solve the problem with SOM3 in the later stages while fine-tuning the turbine positions, utilizing the fact that this fine-tuning focuses on clusters of closely located turbines and that such information can be effectively captured by 3-cut (or 4,5-cut).

Finally, it is interesting to observe that although the LSOMs employ less accurate power models (i.e., their optimal solutions are not the same as the optimal solutions of SOM), LSOMs can still produce good solutions even when benchmarked by the more accurate power calculations. We plan to explore this in more detail once the SOMs are improved.

7 Conclusion

We have presented the wind farm design layout problem and proposed two models to incorporate the nonlinearity in the problem. The first model (SOM1) is a direct formulation of the problem in constraint programming. While having promising performance under complex wind scenarios, the major drawback of this approach is the "curse of dimensionality" – the growth of the numbers of variables and terms quickly exceeds reasonable computational capacity. A second decomposed MIP model (SOM3) performs well in the simple wind regimes, because no-good and 3-cuts can accurately describe the turbine interactions. However with more complicated wind regimes, SOM3 is unable to improve its early feasible solutions due to the weakness of the current cuts. We also presented a novel extension of an existing LSOM model. The LSOM models are based on a less accurate model of power productions, thus having different objective functions than the SOMs. However, the models can be solved more quickly and achieve high quality solutions when a posteriori evaluated with the more accurate sum-of-squares power calculation.

In summary, we have presented two new models for the wind farm layout optimization problem. These CP and MIP models are the first mathematical programming models that capture the wind turbine interactions by modelling the sum-of-squares equations – most accurate analytical multi-wake modelling in the literature [10]. We also presented an extension (LSOM2) to a previous MIP model (LSOM1) and demonstrated improved solution quality and time. Based on the experimental study, we think that the most promising directions for future work include the strengthening of cuts for SOM3 and hybridization of SOM1 and SOM3.

References

1. Mosetti, G., Poloni, C., Diviacco, B.: Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm. Journal of Wind Engineering and Industrial Aerodynamics **51**(1) (January 1994) 105–116
2. Grady, S.: Placement of wind turbines using genetic algorithms. Renewable Energy **30**(2) (February 2005) 259–270
3. Chowdhury, S., Messac, A., Zhang, J., Castillo, L., Lebron, J.: Optimizing the Unrestricted Placement of Turbines of Differing Rotor Diameters in a Wind Farm for Maximum Power Generation. In: Proceedings of the ASME 2010 International Design Engineering Technical Conference & Computers and Information in Engineering Conference IDETC/CIE 2010, Montreal, Quebec, Canada (2010) 1–16
4. Donovan, S.: Wind Farm Optimization. In: 40th Annual Conference, Operational Research Society of New Zealand, Wellington, New Zealand (2005)
5. Fagerfjäll, P.: Optimizing wind farm layout: more bang for the buck using mixed integer linear programming. Master's thesis, Chalmers University of Technology and Gothenburg University (2010)
6. Kwong, W.Y., Zhang, P.Y., Romero, D., Moran, J., Morgenroth, M., Amon, C.: Multi-objective Optimization of Wind Farm Layouts under Energy Generation and Noise Propagation. In: Proceedings of the ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2012, Chicago (2012)
7. Dilkina, B., Kalagnanam, J., Novakovskaia, E.: Method for designing the layout of turbines in a windfarm (2011)
8. Manwell, J.F., McGowan, J.G., Rogers, A.L.: Chapter 9 - Wind Turbine Siting, System Design, and Integration. In: Wind Energy Explained. (2010)
9. Jensen, N.: A note on wind generator interaction. Technical report, Risoe National Laboratory (1983)
10. Renkema, D.J.: Validation of wind turbine wake models. Master of science thesis, Delft University of Technology (2007)
11. Atamtürk, A., Nemhauser, L.G., Savelsbergh, M.W.P.: The Mixed Vertex Packing Problem
12. Owen, S.H.: Scenario Planning Approaches to Facility Location: Models and Solution Methods. Phd thesis, Northwestern University (1998)
13. Stephenson, K.: Introduction to Circle Packing: The Theory of Discrete Analytic Functions. Cambridge University Press, Cambridge (2005)
14. Manwell, J.F., McGowan, J.G., Rogers, A.L.: Chapter 2 - Wind Characteristics and Resources. In: Wind Energy Explained. (2009)
15. Manwell, J.F., McGowan, J.G., Rogers, A.L.: Chapter 3 - Aerodynamics of Wind Turbines. In: Wind Energy Explained. (2009)
16. Du Pont, B.L., Cagan, J.: An Extended Pattern Search Approach to Wind Farm Layout Optimization. ASME Conference Proceedings **2010**(44090) (2010) 677–686
17. Kuo, C.C., Glover, F., Dhir, K.S.: Analyzing and Modeling the Maximum Diversity Problem by Zero-One Programming. Decision Sciences **24**(6) (November 1993) 1171–1185
18. Pisinger, D.: Exact solution of p-dispersion problems. (December) (1999) 1–16
19. Martí, R., Gallego, M., Duarte, A.: A branch and bound algorithm for the maximum diversity problem. European Journal of Operational Research **200**(1) (January 2010) 36–44
20. Achterberg, T.: SCIP: solving constraint integer programs. Mathematical Programming Computation **1**(1) (January 2009) 1–41