Renewable Energy 165 (2021) 162–173

Contents lists available at ScienceDirect

Renewable Energy

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

An efficient method for modeling terrain and complex terrain boundaries in constrained wind farm layout optimization[★]

Sohail R. Reddy [*]

Department of Applied Mathematics, Naval Postgraduate School, Monterey, CA, United States

A R T I C L E I N F O

Article history:
Received 15 March 2020
Received in revised form
15 October 2020
Accepted 17 October 2020
Available online 5 November 2020

Keywords:
Wind farm layout optimization
Wind energy
Land-based constraints
Support vector machine
Spatial interpolation

A B S T R A C T

A novel and efficient approach for modeling irregular domain boundaries and enforcing complex land-based constraints in wind farm layout optimization is presented. The presented method can be used to enforce constraints and represent complex domains for which an analytical expression is not available. The developed approach uses support vector domain description to obtain an analytical expression of the domain from data points sampled from the various regions for wind turbine placements. Such an analytical, continuous, differentiable description can then be used for wind farm layout optimization using gradient based and non-gradient based optimization algorithms. The applicability of the method was demonstrated on a wind farm layout optimization problem where several regions of the terrain were restricted. The method was shown to accurately and efficiently model regular, irregular, convex, non-convex and discontinuous domains/regions using only scattered data sampled from the domains. An efficient method for terrain modeling and interpolation is also presented. The proposed interpolation method uses quadtree decomposition to decompose the large interpolation problem into several smaller interpolation problems thereby reducing the complexity of the interpolation. The framework developed (libSVDD) is also made publicly available.

1. Introduction

With the environmental impacts of fossil fuels becoming more evident, a significant amount of resources have been allocated to move towards energy independence. In the last two decades, a strong push towards clean, renewable energy has resulted in significant advances in the area of solar, wind and nuclear energies. Wind energy contributed 19% to the 11% of total renewal energy produced in 2015. With an annual growth rate of 34%, wind power is the second fastest growing renewable energy [1]. It also leads to the least water consumption, lowest greenhouse gas emission, and most favorable social impact than geothermal, hydro-power and solar energy [2]. There are over 57,000 wind turbines and over 200 wind farms in the United States alone.

Wind farm design and layout optimization has been a growing area of research for several year. It is a challenging area due to the computationally expensive model, large number of design variables and constraints, and extreme multimodality of the function space. Much of the published work in wind farm layout optimization has focused on maximizing power generation with less consideration given to the land usage. Ryberg et al. [3] evaluated the eligibility of land over Europe for wind farm development. They investigated the different factors that influences the eligibility of land for wind farm development. Although the land eligible and available for wind farm development can be identified, an efficient and accurate approach is still needed to incorporate such constraints within an optimization framework. Restrictions placed on land available for wind farm design often lead to a complex-shaped eligible/feasible regions. Shapes of such complex regions are difficult to define analytically and, therefore, need special treatment to be incorporated into the optimization framework. In addition to modeling and incorporating land-based constraints, an accurate method for modeling the terrain as a continuous function is also required. The terrain profile is often defined using the elevation measurements at a large set of scattered points. These scattered points are used as the support points for interpolation to obtain an analytical definition of the terrain profile. These interpolation methods are often plagued with scaling issues due to the large number of support points. Nonetheless, much effort has been dedicated to tackle the wind

[★] libSVDD can be downloaded from https://github.com/sohailrreddy/libSVDD.

[*] Dept. of Mechanical and Materials Eng, Florida Int'l Univ., Miami, FL, United States.

E-mail address: sredd001@fiu.edu.

https://doi.org/10.1016/j.renene.2020.10.076
0960-1481/© 2020 Elsevier Ltd. All rights reserved.

farm optimization problem while modeling and incorporating terrain constraints.

Zhang et al. [4] solved the wind farm optimization problem using the mixed interger linear programming (MIP) and constraint programming (CP) approach. Their work considered the case where different areas of land are owned by different owners. Wang et al. [5] developed a constraint handling method particularly for enforcing constraints for lands owned by different owners. Their approach uses penalty functions to transform infeasible designs into feasible designs.

Wind farm layout optimization over complex terrains has been a topic of great interest. Table 1 presents some efforts in the recent years to model irregular domain boundaries. Feng and Shen [6] optimized the farm layout over on a Gaussian hill. Their work considered a rectangular farm where all the land is assumed to be feasible for wind farm development. Brogna et al. [7] performed layout optimization of wind farm in a complex terrain. They modeled the terrain profile using linear interpolation of nearby sampled points. Their work considered the farm layout to be rectangular where no constraints are placed on the land available for wind turbine placement. Feng et al. [8] also performed wind farm layout optimization over a complex terrain for maximizing annual energy production (AEP). They also modeled the complex terrain as a linear function of near by sampled points. Their work incorporated constraints on the feasible land available for turbine placement. The feasible or available land for wind turbine placement was modeled as a polygon where the placement of the turbines was only allowed within the polygon. This approach works well for very simple convex shaped boundaries but cannot model irregular, non-convex boundaries. Mittal and Mitra [9] performed layout optimization over a flat surface where the placement of turbines within certain areas was restricted. Their method represented the wind turbines as points and the restricted region as a polygon. A single boundary ray intersection method is then used to evaluate the presence or absence of wind turbines inside the restricted region. As stated before, the polygon representation approach can work for convex and concave shapes but not non-convex boundaries. Chen et al. [10] performed multi-objective optimization of wind farm layout using a genetic algorithm. They too used a polygon to represent the eligible land region where the boundary of the region was divided into smaller straight-line segments. Sorkhabi et al. [11,12] optimized a wind farm layout for maximum AEP and minimum noise. They too incorporated land feasibility constraints using polygon representation. Rather than representing the feasible domain, the authors represented the infeasible region using a collection of smaller polygons. The non-convex regions were represented using smaller convex polygons. Since polygons are straight-line segments, a large number of polygon might be needed to accurately model a non-convex, curved boundary.

Past and recent efforts indicate that a robust method for modeling such irregular, non-convex, disconnected domains is needed. This work addresses these needs. The unique contributions of this work are as follows:

- It develops a method for simplifying the complex domain boundaries and land-based constraints arising in wind farm optimization while preserving accuracy. It uses the support vector domain description (SVDD) to map the region to a feature space where the complex domain boundaries can be represented simply as a spherical boundary. A wind farm layout optimization problem with several terrain constraints can be converted to a single constraint using the SVDD approach.
- It presents an efficient approach for modeling the terrain profile where the single interpolation of a large dataset is decomposed into to several smaller subsets. The method uses quadtree decomposition to hierarchically partition the domain where only the neighboring information is used to perform the interpolation.
- The library for SVDD is made open source and released for public use [13].

The manuscript is organized as follows. Section 2 presents the framework for support vector data description where the mathematical formulation and its use in wind farm design is presented in Section 2.1 and Section 2.2, respectively. Section 3 presents an efficient framework for modeling terrain using interpolation based on radial basis function and inverse distance weighting. Finally, Section 4 demonstrates the use of SVDD and the efficient interpolation technique in constrained wind farm layout optimization. The optimization considers a real wind turbine (V90-3 MW turbine) on the terrain profile of a real wind farm (AWEC wind farm) subjected to the actual wind conditions experienced by this wind farm in the year 2019.

Table 1
Methods used by other authors to model irregular domains.

Authors	Year	Method
Wang et al. [14]	2017	Discrete grid parameterization
Gonzalez et al. [15]	2017	Discrete grid parameterization
Perez-Moreno et al. [16]	2018	Discrete grid parameterization
Feng et al. [8]	2018	Polygon representation
Sorkhabi et al. [12]	2018	Polygon representation
Mittal and Mitra [9]	2019	Polygon representation
Stanley and Ning [17]	2019	Boundary-grid parameterization

2. Support vector domain description

The support vector domain description [18], also sometimes called support vector data description, is a technique inspired by the support vector machines (SVM) of Vapnik [19] for defining an optimal description of the objects. Whereas SVM identifies a separating hyperplane that classifies objects, the SVDD approach attempts to perform the classification using hyperspheres. This method has been successfully used to define the boundary of the data set and can be used to model complex, irregular, non-convex and disconnected boundaries.

2.1. Mathematical background

The SVDD method identifies a sphere of radius R with center $\overrightarrow{a}$ that encompasses the data set such that the radius is minimized

$$F(R, \overrightarrow{a}, \xi_i) = R^2 + C\sum_i \xi_i \tag{1}$$

where C represents the trade-off between the simplicity and the number of objects rejected (outliers). The radius is minimized under the following constraint

$$\left|\left|\overrightarrow{x}_i - \overrightarrow{a}\right|\right|^2 \leq R^2 + \xi_i \quad \forall_i, \xi_i \geq 0 \tag{2}$$

where $\overrightarrow{x}_i$ are the objects (points) whose description is being constructed. Using Eq. (1) and Eq. (2), the Lagrangian can be constructed as

$$L(R,\vec{a},\alpha_i,\xi_i)=R^2+C\sum_i\xi_i-\sum_i\alpha_i\left(R^2+\xi_i-\left\|\vec{x}_i-\vec{a}\right\|^2\right)-\sum_i\gamma_i\xi_i \tag{3}$$

where the Lagrange multipliers satisfy $\alpha_i \geq 0$ and $\gamma_i \geq 0$. Differentiating Eq. (3) yields a new set of constraints

$$\begin{aligned}&\sum_i\alpha_i=1 \quad \forall_i\\ &\vec{a}=\frac{\sum_i\alpha_i\vec{x}_i}{\sum_i\alpha_i}=\sum_i\alpha_i\vec{x}_i \quad \forall_i\\ &C-\alpha_i-\gamma_i=0 \quad \forall_i\end{aligned}\tag{4}$$

It can be seen from the second constraint in Eq. (4) that the center of the sphere can be represented as a linear combination of weights α_i and the objects $\vec{x}_i$. Since both $\alpha_i \geq 0$ and $\gamma_i \geq 0$, the third constraint in Eq. (4) can be rewritten as $0 \leq \alpha_i \leq C$. Substituting Eq. (4) into Eq. (3) yields a function to maximize given by

$$L=\sum_i\alpha_i\left(\vec{x}_i\cdot\vec{x}_i\right)-\sum_{i,j}\alpha_i\alpha_j\left(\vec{x}_i\cdot\vec{x}_j\right) \tag{5}$$

with constraints $0 \leq \alpha_i \leq C$ and $\sum_i \alpha_i = 1$. It can be seen from the second constraint in Eq. (4) that only those objects $\vec{x}_i$ with $\alpha_i > 0$ are needed to define the hypersphere. These objects are referred to as support vectors. The radius of the sphere (R) can be obtained by taking the distance between the center ($\vec{a}$) and any support vector. This procedure of identifying the support vectors, α and the radius R by minimizing Eq. (5) is referred to as the training phase and is similar to that used in SVM.

It is well known that an object $\vec{z}$ is on or within a sphere with the center at $\vec{a}$ and radius R if $||\vec{z}-\vec{a}||^2 \leq R^2$. If the center is expressed using the support vectors then the object z is on or within the sphere if

$$\left(\vec{z}\cdot\vec{z}\right)-2\sum_i\alpha_i\left(\vec{z}\cdot\vec{x}_i\right)+\sum_{i,j}\alpha_i\alpha_j\left(\vec{x}_i\cdot\vec{x}_j\right)\leq R^2 \tag{6}$$

An object is “accepted” if it satisfies this acceptance criteria (Eq. (6)) and has the same classification as the training set. Fig. 1 shows the standard SVDD approach. It can be seen that any point inside the hypersphere is considered to be inside the region formed by the training points.

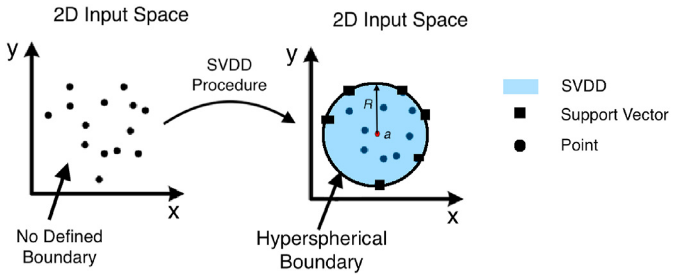


Fig. 1. Visualization of the SVDD procedure without using kernel mapping for a two-dimensional case.

In real-world applications, the data sets are not spherically distributed and, therefore, classification using hyperspheres will not result in a tight fit (description) around the data set. This issue can be addressed by mapping the data set (objects) into a feature space where a sphere can be a better approximation. This is done using a “kernel-trick”, where a kernel $K(\vec{x}_i \cdot \vec{x}_j)$ that satisfies Mercer’s theorem [20] is used to map the data set. This SVDD procedure using kernel mapping is shown in Fig. 2. It can be seen that the approach constructs an irregular boundary in input parameter space that conforms to the training points and represents this irregular boundary as a hyperspherical boundary in an $n-$ dimensional feature space. If a point is within this hyperspherical boundary, then it also lies within the irregular boundary.

Replacing all inner products $(\vec{x}_i \cdot \vec{x}_j)$ by the kernel representation $K(\vec{x}_i \cdot \vec{x}_j)$, the domain description problem is now given by

$$L=\sum_i\alpha_iK\left(\vec{x}_i\cdot\vec{x}_i\right)-\sum_{i,j}\alpha_i\alpha_jK\left(\vec{x}_i\cdot\vec{x}_j\right) \tag{7}$$

where constraints $0 \leq \alpha_i \leq C$ and $\sum_i\alpha_i = 1$ are enforced. Then, the acceptance criteria is given as

$$K\left(\vec{z}\cdot\vec{z}\right)-2\sum_i\alpha_iK\left(\vec{z}\cdot\vec{x}_i\right)+\sum_{i,j}\alpha_i\alpha_jK\left(\vec{x}_i\cdot\vec{x}_j\right)\leq R^2 \tag{8}$$

One of the more commonly used kernel is the Gaussian kernel $K_G(\vec{x}_i \cdot \vec{x}_j) = \exp(-(\vec{x}_i - \vec{x}_j)^2/\sigma^2)$. Using the Gaussian kernel, the domain description problem can be written as

$$L=1-\sum_i\alpha_i^2-\sum_{i\neq j}\alpha_i\alpha_jK_G\left(\vec{x}_i\cdot\vec{x}_j\right) \tag{9}$$

where the constraints are $0 \leq \alpha_i \leq C$ and $\sum_i\alpha_i = 1$. The acceptance rule then becomes

$$-2\sum_i\alpha_iK_G\left(\vec{z}\cdot\vec{x}_i\right)\leq R^2-C_X-1 \tag{10}$$

where C_X is the third term on the left-hand side of Eq. (8). The standard deviation of the Gaussian kernel (σ) determines the number of support vectors and the tightness of the fit, where a smaller σ results in a tighter fit and more support vectors. It should be mentioned that once the SVDD model is trained, the classification of new points in their respective regions is exceptionally fast. This is because of the acceptance criteria which simply compares the distance of the point from the center with the radius of the mapped hypersphere.

2.2. Use of SVDD in complex land-based constraints for wind farm optimization

There are several factors that can restrict the placement of wind turbines in a particular region. These restrictions can be due to elevation constraints, proximity to features such as cities, soil conditions and land availability to name a few. Often, an analytical expression to define the feasible region or the boundary of the feasible region for wind turbine placement is not available. Therefore, the feasible domain has to be represented by scattered data sets of the sampled from the feasible and infeasible regions. Here, the infeasible region is analogous to the restricted region. The scattered data set (objects) can be used with the SVDD approach to create a description of the feasible areas for wind turbine placement. Once the SVDD is trained using the sampled set, it can be

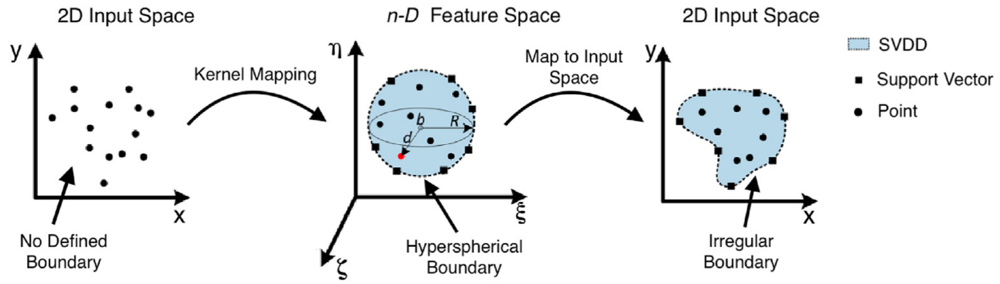


Fig. 2. Visualization of the SVDD procedure using kernel mapping for a two-dimensional case.

used to determine whether a turbine with location $\overrightarrow{z}$ lies within the feasible region or outside of it (infeasible region) by means of Eq. (10).

Consider the terrain defined by the Himmelblau function (Eq. (11)) where Fig. 3 shows the Himmelblau function over $x, y \in [-6, 6]$. It can be seen that function defines a terrain with large variations in function value (elevation). This functions was chosen because slicing the function at different function values yields boundaries of various shapes and allows for rigorous validation of the SVDD approach.

$$z(x,y) = \left(x^2 + y - 11\right)^2 + \left(x + y^2 - 7\right)^2 \tag{11}$$

The SVDD is trained for classification using the scatter data set sampled from the Himmelblau function. A training set of 1000 samples uniformly distributed over $x, y \in [-6, 6]$ was generated using SOBOL's quasi-random sequence. Out of these uniformly sampled points, only those that fit the criteria for feasibility were retained and used to train the SVDD models. Four different cases were considered each with a different feasibility criteria. The feasible regions for the four cases were all areas where $z < 200$, $z < 150$, $z < 100$ and $z < 50$, respectively. Fig. 3 shows the boundaries of the feasible regions as slices at different heights. A testing set of 10,000 samples was also generated to test the SVDD model's ability to classify objects into the feasible and infeasible regions. The result will be compared to the benchmark convex hull algorithm.

Fig. 4 shows the testing sample set classified using the trained SVDD model for each of the four cases. The figure shows the exact boundary of the feasible region (black), the boundary of the convex hull (red), the points accepted by the trained SVDD model (blue) and the points rejected by the trained SVDD model (green). The accepted points are those in the testing sample set that satisfy Eq. (10) and, therefore, lie within the feasible region. It can be seen that the convex hull is not able to conform to the boundary of the feasible region. It is especially true for cases where the feasible domain is non-convex (Case 1 and Case 2) or disconnected (Case 3 and Case 4). The SVDD approach is able to accurately determine whether a point lies within the feasible or infeasible region, and therefore, the trained SVDD can also be used to determine whether a turbine is within the feasible region.

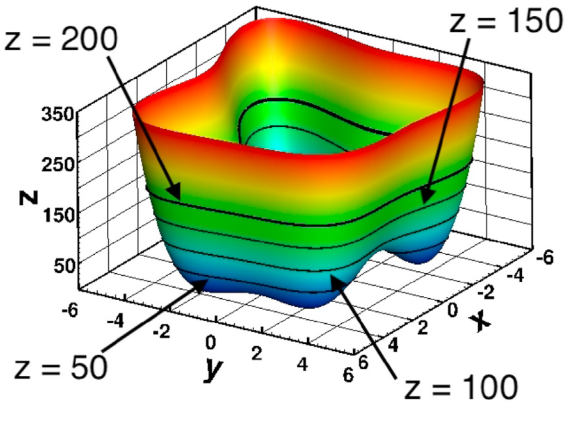


Fig. 3. Himmelblau function showing the boundaries of the four feasible regions.

It is important to note that the SVDD approach can identify whether a point lies within or outside the region of samples used for training. That is, if the SVDD model was trained using samples from the feasible domain, it can be used to determine whether a new point lies within the feasible domain or outside the feasible domain (i.e. infeasible domain). Conversely, the SVDD model can also be trained using samples from the infeasible domain, in which case the constraints would have to be appropriately modified.

3. Spatial interpolation for terrain modeling

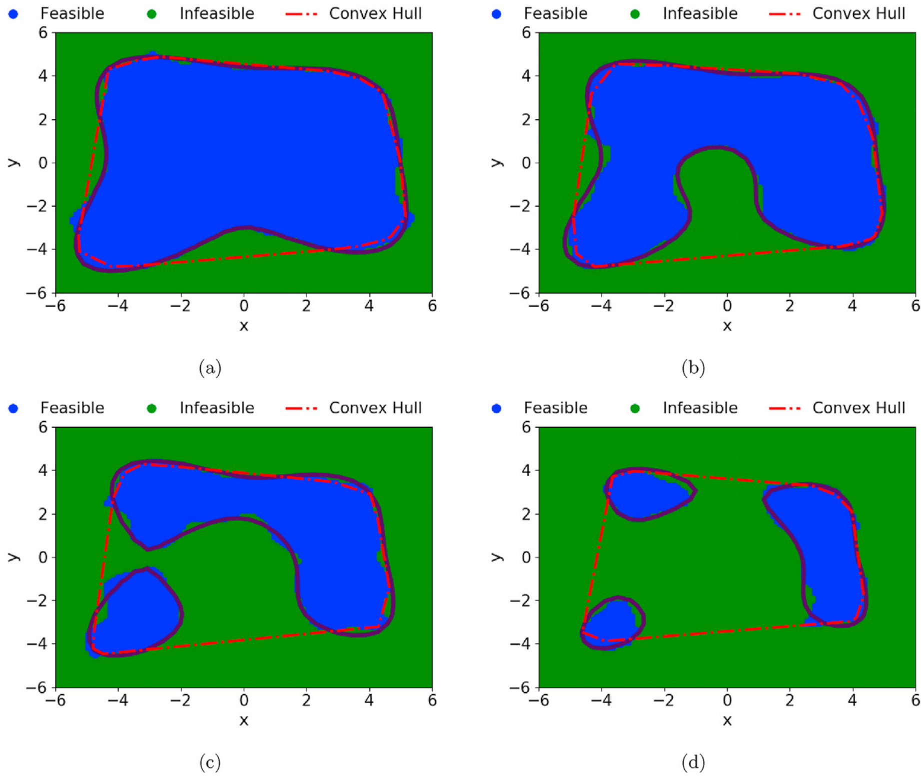
Wind farm layout optimization for non-flat regions require the use of accurate models for defining terrain elevation. There exist several databases for extracting the terrain elevation/profile from satellite images [21]. These databases however, only provide elevation at scattered points and not an analytical function defining the terrain profile. Such cases require the use of spatial interpolation techniques. Two such approaches for interpolation are inverse distance weighting and radial basis function interpolation [22]. These two approaches are presented and incorporated into the WindFLO framework.

The Inverse Distance Weighting (IDW) approach models a function as a linear combination of weights and function values of nearby known points. It assumes each point has local influence that diminishes with increasing distance. The IDW formulation, given by Eq. (12), is an exact interpolator and goes through each sample point. The maximum and minimum values in the interpolated surface can only occur at sample points.

$$u(\overrightarrow{x}) = \begin{cases} \dfrac{\sum_{i=1}^{N} \omega_i(\overrightarrow{x}) u_i}{\sum_{i=1}^{N} \omega_i(\overrightarrow{x})}, & \text{if } d(\overrightarrow{x}, \overrightarrow{x}_i) \neq 0 \text{ for some } i \\ u_i, & \text{if } d(\overrightarrow{x}, \overrightarrow{x}_i) = 0 \text{ for some } i \end{cases} \tag{12}$$

where $\omega_i(\overrightarrow{x}) = 1/d(\overrightarrow{x}, \overrightarrow{x}_i)^p$, $\overrightarrow{x}_i$ are the samples used for interpolation and N are the total number of samples. The power term p dictates the rate at which the influence of the nearby points decreases. A higher value of p leads to a more localized interpolation.

The Radial Basis Function (RBF) approach models the function as a linear combination of weights and RBF kernels functions [23]. The RBF interpolation is also an exact interpolator but, unlike IDW, the local extrema of the interpolation are not limited to the sample points. The RBF interpolation is given as



(a)

(b)

(c)

(d)

Fig. 4. Regions classified using SVDD where the feasible region was taken to be: a) $z < 200$, b) $z < 150$, c) $z < 100$ and d) $z < 50$.

$$f_i = \sum_{j=1}^{N} \omega_j \varphi \left|\left| \left(\vec{x}_i, \vec{x}_j \right) \right|\right| \tag{13}$$

where $\varphi||(\vec{x}_i, \vec{x}_j)||$ is the kernel using the i^{th} and j^{th} point, ω_j are the weights obtained by solving the linear system using the training set and c is the shape factor. The shape factor scales the kernel and controls the accuracy of the interpolation. Various different RBF kernel have been developed and are given in Table 2. It was seen that the Multiquadrics kernel was the most accurate for terrain interpolation [24].

When performing IDW interpolation, a total of k nearest points are typically used for the interpolation. Therefore, computational complexity of identifying the k nearest points using standard sorting algorithms is typically $O(N\log N)$. For a total of N interpolation points, the computational complexity for the nearest neighbor search is $O(dN)$, where d is the dimensionality of the problem. For the RBF interpolation, the construction of the Φ matrix ($\Phi_{ij} = \varphi(\vec{x}_i, \vec{x}_j)$) requires $O(N^2)$ operations. Solving the resulting linear systems takes approximately $O(N^3)$ operations. Therefore it can be seen that both types of interpolation are inefficient when a large number of sample points N is used. For this reason, efficient space partitioning techniques must be used to accelerate the search for the nearest neighbor search and the construction of the interpolation. This work uses quadtree decomposition to partition the sample space.

Table 2
Expressions for various radial basis function kernels.

	RBF Kernels ($r = \|\vec{x}_i - \vec{x}_j\|$)
Multiquadrics	$\varphi\|\|(\vec{x}_i, \vec{x}_j)\|\| = \sqrt{r^2 + c^2}$
Inverse Multiquadrics	$\varphi\|\|(\vec{x}_i, \vec{x}_j)\|\| = \frac{1}{\sqrt{r^2 + c^2}}$
Gaussian	$\varphi(\vec{x}_i, \vec{x}_j) = e^{-(rc)^2}$

The sample space is first partitioned using an iterative hierarchical domain decomposition method (Octree in 3D and Quadtree in 2D). For a 2D example, at the first iteration a simple square encompassing all points in the 2D domain is created. All sample points are placed within this first square. The square is then decomposed into 4 smaller squares of equal sizes. This decomposition is continued until the number of points within a square is less than some specified number. It should be noted that only those squares that contain more than the allowable number of points are decomposed. This decomposition creates a hierarchical tree structure where the childless boxes are called leaves and all other boxes are called parent boxes. The neighbors of each leaf cell are then computed at the leaf level. That is, the neighbors of a leaf cell can only be other leaf cells and not parent cells. At each decomposition step, the sample points in the parent boxes are placed into the appropriate child boxes that contain these points. Fig. 5a the decomposition for a 2D problem.

Once the quadtree is constructed, the computational cost of the interpolation can be drastically reduced. To obtain the function value (elevation) at point $\vec{y}$, the appropriate leaf cell that encompasses the point $\vec{y}$ is identified. The maximum computational cost for the search of this leaf cell in a quadtree structure with a maximum depth of d is $O(4d)$, where d is typically between 10 and 20. The IDW interpolation is then applied using the points in the identified leaf cell and it's neighboring leaf cells. This alleviates the need to perform the nearest neighbor search.

In the case of RBF interpolation, since the weights ω_j in the interpolation only depend on the samples used for the construction of the interpolation and not the point being evaluated, they can be determined beforehand and reused. This is done by constructing an

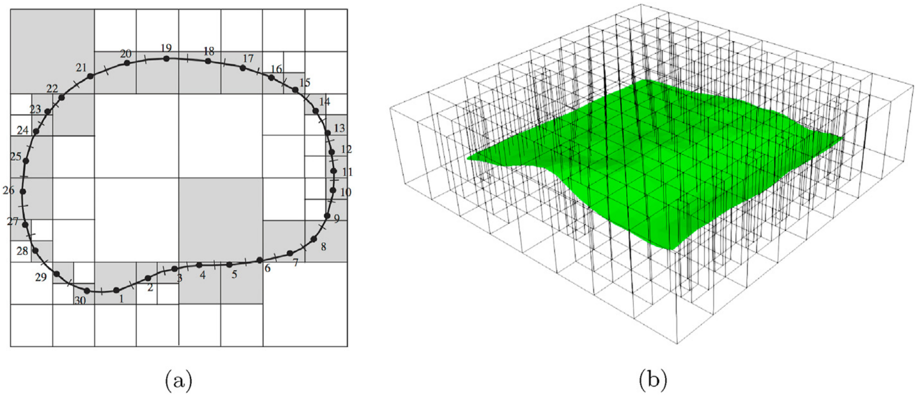


Fig. 5. a) Sample quadtree decomposition in 2D and b) Quadtree decomposition of the AWEC terrain in 3D.

RBF interpolation for each leaf cell using the points in the leaf and its neighbors. As before, the appropriate leaf cell that encompasses the point $\overrightarrow{y}$ is identified. The RBF interpolation constructed for it is then used to compute the function value (elevation).

The proposed interpolation methods are investigated using the terrain data for the AWEC wind farm in Mojave CA, USA. Fig. 5b shows the quadtree decomposition of the AWEC terrain. The IDW interpolation was performed with $p = 3$. The RBF interpolation used the Multiquadrics kernel with $c = 5$. A maximum depth of 10 divisions was allowed for the Quadtree decomposition with each cell containing a maximum of 10 points. Fig. 6 shows the exact terrain elevation obtained using TouchTerrain, the IDW interpolated elevation and the RBF interpolated elevation. For the analysis of flow over a complex terrain, a detailed and accurate terrain description is needed. For the case of wind farm modeling where lower fidelity analytical wake models are used for analysis, such approximations of terrain are sufficient. It can be seen that both methods model the elevation reasonably well but the RBF approximation yields more accurate results.

4. Constrained wind farm layout optimization

The developed classification and interpolation methods are investigated for their use in wind farm layout optimization. This section presents the wind farm analysis framework, problem formulation, optimization algorithm and results of the wind farm layout optimization.

4.1. Wind farm analysis framework: WindFLO

The wind farm analysis in this work was performed using the WindFLO software [25,26]. WindFLO is a publicly available software that can be used to analyze the power generation, land-usage and cost of wind farms. It uses analytical wake models, wake merge

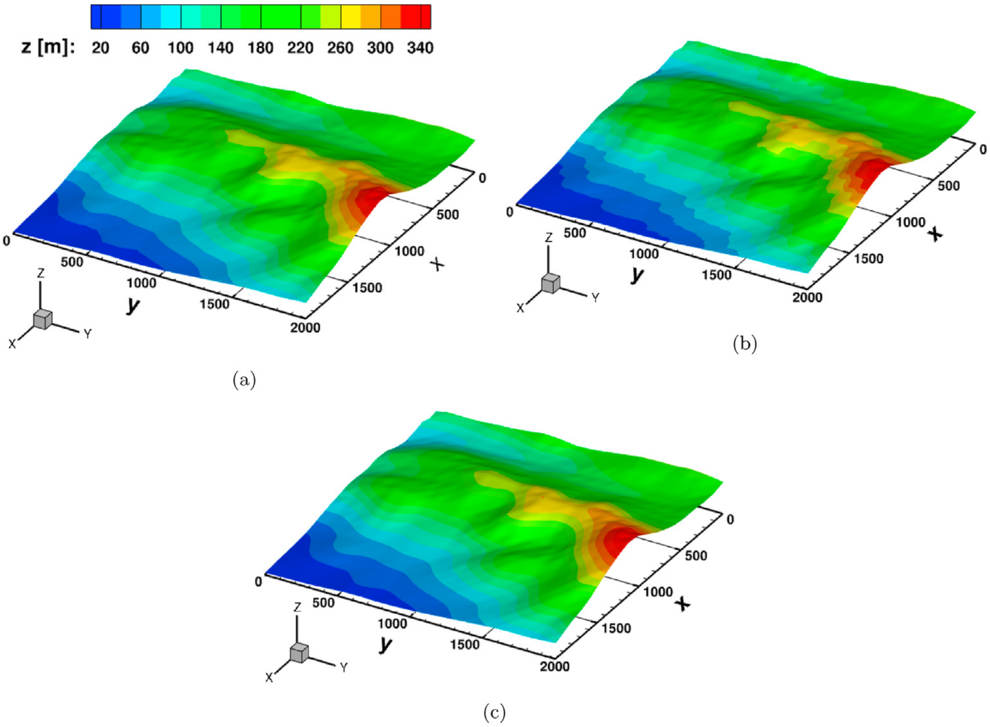


Fig. 6. Elevation of the AWEC terrain obtained using: a) TouchTerrain [21], b) quadtree-based inverse distance weighting and c) quadtree-based radial basis function interpolation.

schemes and ambient wind models to compute the wind velocity experienced by wind turbines in a farm. The framework features six analytical wake models and four wake merge/superposition models. It also accounts for the terrain profile/elevation and wake-terrain interactions. The schemes in WindFLO were validated using experimental data from a scaled wind farm to within 1% relative error. It has been used for wind farm layout analysis and optimization. The WindFLO framework also features one-dimensional and two-dimensional, linear and non-linear cost models and uses the convex hull approach to quantify the area of land used by the wind farm layout. For brevity, the specific numerical methods present in WindFLO are not discussed in this manuscript. For a detailed implemented, readers are referred to Ref. [26]. The previously presented quadtree-based IDW and RBF interpolation techniques were incorporated into the WindFLO framework.

4.2. Problem definition

The wind farm considered in this optimization study consisted of 16, three-blade Vestas V90-3 MW turbines [27]. Fig. 7 shows such a 3-blade design. The V90-3 MW turbine has a rotor radius $R = 45$m and tower height $H = 105$m. Fig. 7b shows the wind turbine location relative to the terrain, where the reference elevation Z_{REF} is taken to be at sea-level (i.e. 0 m) and the terrain elevation Z_E is obtained using the quadtree-based interpolation techniques presented earlier.

The power curve of the V90-3 MW turbine is shown in Fig. 8a. To obtain the thrust curve, the induction factor a was first computed using the relation $C_p = 4a(1-a)^2$, where C_p is the coefficient of power and is related to the turbine power by

$$P = k_g k_b C_P \left(\frac{1}{2} \rho \pi \frac{D^2}{4} U^3 \right) \tag{14}$$

where k_g and k_b are the generator efficiency and the gearbox efficiency, respectively, ρ is the air density, U is the wind speed, D is the rotor diameter, and C_P is the coefficient of power. The coefficient of thrust can then be computed as $C_T = 4a(1-a)$. Fig. 8 shows the power curve, and the computed turbine coefficients. The coefficient of thrust is needed by all wake models in WindFLO.

The velocity in the wake emanating from the turbines is modeled using the Frandsen's wake model. The interaction between multiple wakes is modeled using the quadratic superposition scheme. It was shown that the Frandsen's wake model with the quadratic wake merge scheme yielded the most accurate results. All expressions used to compute the turbine and farm power and efficiency are given in Refs. [26].

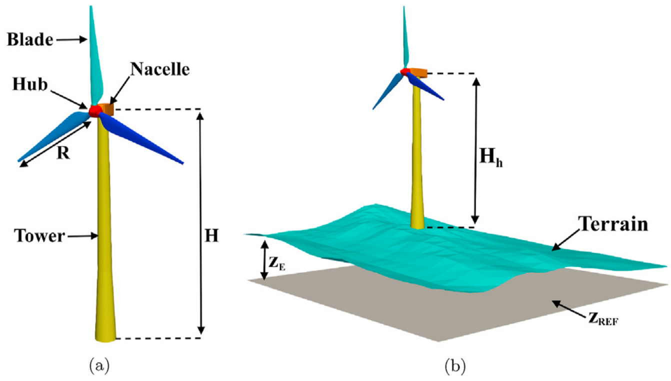


Fig. 7. Parameters in: a) a stand alone wind turbine and b) wind turbine on a terrain.

The optimization study in this work considers a realistic wind farm terrain and wind farm conditions. The terrain profile was again taken to be that of the AWEC farm in Mojave CA, USA and was obtained using the TouchTerrain software [21]. For comparison, the optimization is performed using both IDW and RBF quadtree-based interpolation. The wind conditions experienced by the AWEC wind farm are shown in Fig. 8c and were obtained using the cli-MATE tool [28]. Here, the wind rose diagram (Fig. 8c) shows the wind conditions at the farm from January 1, 2019 to December 31, 2019.

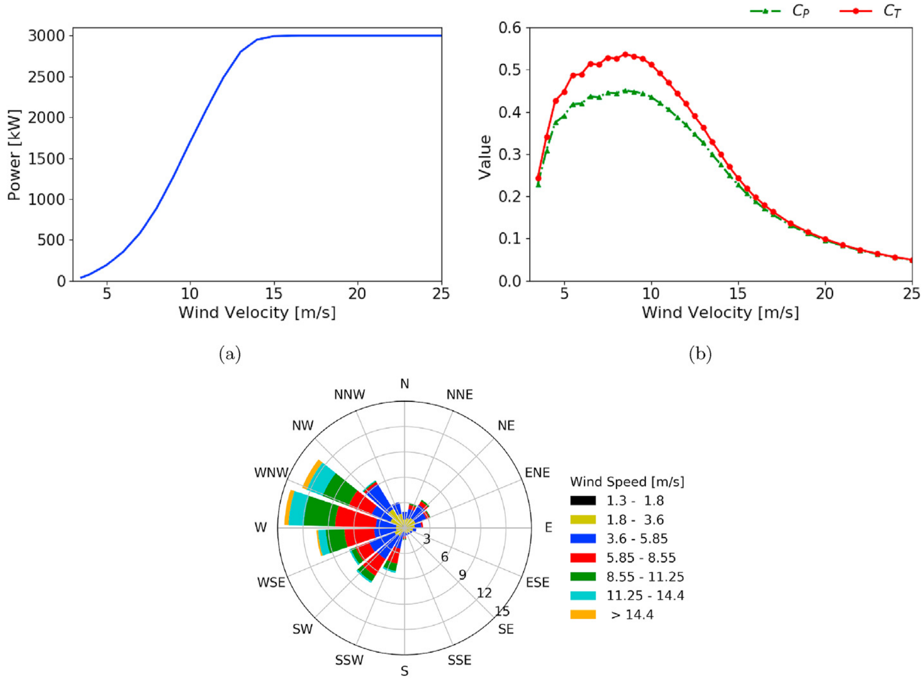
The SVDD-based constraint approach is investigated using three test cases, each with a different feasible region. It should be stated that an infeasible region is the same as a restricted region. The feasible and infeasible regions for the three cases are shown in Fig. 9. The first case placed a circular infeasible region of radius 600 m centered at $x = 1000$ m, $y = 1000$ m. The trained SVDD model for Case 1 used approximately 400 points scattered in the **infeasible** region as support vectors. This case was used to demonstrate the applicability of the SVDD approach for well-defined, regular, convex domains. The second case considered all regions with elevation greater than 180 m ($z > 180$) to be infeasible. As shown in Fig. 9b, such a constraint results in an irregular domain with a non-convex boundary. The trained SVDD model for Case 2 used approximately 530 points scattered in the **feasible** region as support vectors. The third case considered all regions where 130m $< z <$ 210m to be infeasible. This separates the feasible region into four disconnected regions. The infeasible region can be assumed to be body of water such as a river. The trained SVDD model for Case 3 used approximately 810 points scattered in the **feasible** region as support vectors. That is the Case 3 SVDD model was trained using samples from disconnected regions. It should be mentioned that a single SVDD model was trained for each of these cases. Although Case 3 features several disconnected regions, they can all be efficiently modeled using a single SVDD model.

4.3. Optimization algorithm

The typical multi-variate, single objective, constrained optimization can be written as

$$\begin{aligned} & min \quad f(\overrightarrow{x}) \\ & \quad \overrightarrow{x} = \{x_1, \ldots, x_m\} \\ & \quad subject\ to : x \in [a_i, b_i], \quad i = 1, \ldots, m \\ & \quad h_j(\overrightarrow{x}) = 0, \quad j = 1, \ldots, n \\ & \quad g_k(\overrightarrow{x}) \leq 0, \quad k = 1, \ldots, o \end{aligned} \tag{15}$$

where $\overrightarrow{x}$ is a vector of design variables, h_j is the j^{th} equality constraint and g_k is the k^{th} inequality constraint. The constraints are enforced using the constraint violation approach [29]. Rodrigues et al. [30] performed a detailed review on techniques for wind farm optimization. They show that no particular constraint handling technique is superior when performing constraint optimization. The wind farm layout optimization in this work is performed using the Single-Objective Hybrid Optimizer (SOHO) [31,32]. The No-Free-Lunch theorem states [33] that no single algorithm is superior over another for an entire problem set. This means that the superiority of one algorithm over another for a problem set is paid for by the loss of its superiority over another problem set. Therefore, coupling several optimization algorithms and actively switching between them can increase their robustness over a larger problem set. The SOHO algorithm features the single objective variants of three individual algorithms: NSGA-III [34], NSDE-R [35] and MOEA-DD [36]. Each algorithm then operates until convergence or stagnation is reached. If stagnation is detected, an alternative algorithm is randomly selected from the remaining two. A

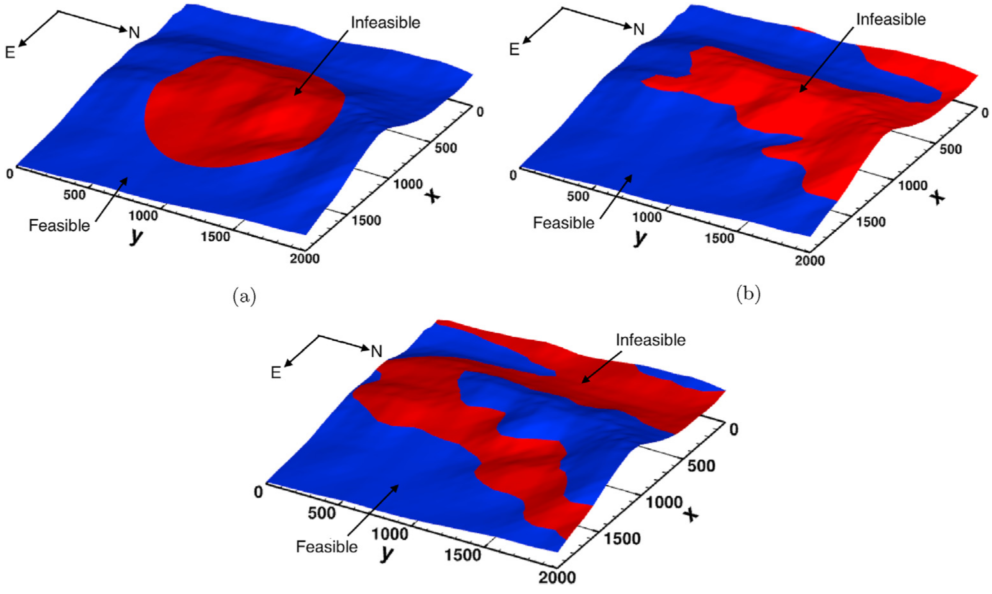


(a)

(b)

(c)

Fig. 8. a) Experimental power curve of the V90-3 MW turbines, b) computed turbine coefficients and c) windrose diagram for the AWEC wind farm for the year 2019.



(a)

(b)

(c)

Fig. 9. Feasible and infeasible regions of the AWEC terrain for: a) Case 1, b) Case 2 and c) Case 3.

detailed description of SOHO is given in Ref. [37].

The optimization study attempts to find the farm layout that minimizes the cost-of-energy (CoE) in $/MWh, where AEP is the annual energy production (Eq. (16)). The cost of the wind farm is obtained using the non-linear, bivariate cost model in WindFLO. The design variables were the x and y coordinates of the wind turbines. This resulted in a total of 32 design variables (two for each of the 16 turbines).

$$AEP_{\mathrm{Farm}} = (365 \times 24) \int_{0^o}^{360^o} \int_{0}^{U_{\max}} P_{\mathrm{Farm}}(U,\theta) p(U,\theta) \mathrm{d}U \mathrm{d}\theta \qquad (16)$$

The AEP of the wind farm is also normalized by the AEP calculated if all turbines were operating year around at their rated wind speed (420,480 MWh). Two constraints were also included for each case. The first constraint required that the clearance around each turbine be greater than its diameter.

$$g_1(\vec{x}) = \max_{j=1,\ldots,N}\left(D - \vec{x} - \vec{x}_{j_2}\right) \tag{17}$$

The second constraint required that each turbine be within the feasible region (Eq. (10)). The SVDD model trained for each of the three cases was used to determine whether a turbine was in the feasible or infeasible region. Using the SVDD approach, the second constraint can be written as

$$g_2(\vec{x}) = C_X + 1 - 2\sum_i \alpha_i K_G\left(\vec{x} \cdot \vec{\chi}_i\right) - R^2 \tag{18}$$

where $\vec{\chi}_i$ are the support vectors and $\vec{x}$ is the location of the turbines. In Case 1, where the SVDD model was trained using samples from the infeasible region, for a turbine to be in the feasible region, Eq. (18) must be greater than 0. In Case 2 and Case 3, where the SVDD model was trained using samples from the feasible region, for a turbine to be in the feasible region, Eq. (18) must be less than 0.

4.4. Optimization results

For comparison purposes, the reference farm layout was obtained by optimizing the farm layout while neglecting the SVDD based constraint (Eq. (18)). That is, the optimization problem was first solved with only one constraint (Eq. (17)) while neglecting the SVDD-based constraint (Eq. (18)).

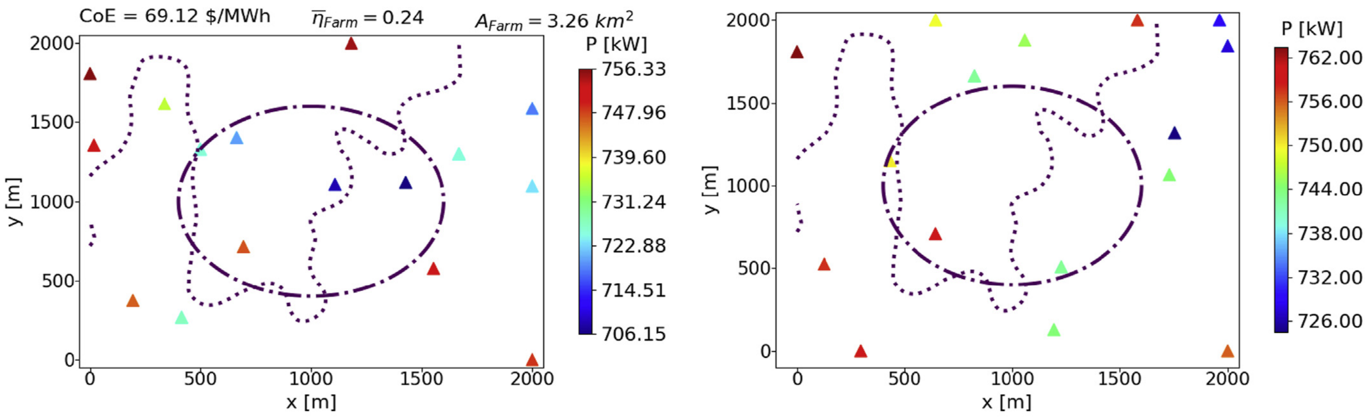
Fig. 10 shows the optimized reference layouts and the boundaries of the feasible regions of two of the three cases. It can be seen that the optimized layouts feature five (when IDW interpolation is used) to three (when RBF interpolation is used) turbines within the Case 1 restricted region. There were also a total of six to seven turbines within the Case 2 restricted region. It will be shown that the incorporating the SVDD-based constraint restricts the placement of wind turbines within these infeasible regions.

Table 3 shows the performance characteristics of the reference farm and the wind farms optimized with SVDD-based constraints. It can be seen that the SVDD-based constraint in Case 1 resulted in a slight decrease (less than 1%) in farm performance. The performance decrease is expected since the constraint restricts the placements of turbines at optimal locations and decreases the effective land available. The optimized Case 2 layout results in an approximately 2% decrease in performance relative to the reference, unrestricted wind farm. This is because the infeasible region in Case 2 is much greater than in Case 1 and, therefore, further restricts the placement of turbines at optimal locations. The presence of infeasible region in Case 1 and Case 2 decreases the available land area by 28% and 35% respectively. It can be seen that constricting the wind turbine layout leads to degraded performance. It should be mentioned that the configurations obtained using RBF based interpolation were also analyzed using IDW based interpolation. The results of the two methods of interpolation varied by less than 0.02%.

Fig. 11 shows the turbine and power distribution in the two optimized wind farm layouts under the regular boundary constraint for Case 1. Immediately, it can be seen that the optimized layouts no longer show any turbines within the infeasible regions. This shows that the SVDD approach can be used to efficiently enforce constraints where the feasible domain is a regular convex shape. Table 3 and Fig. 11 also show the multi-modal nature of wind farm layout optimization where multiple arrangements of wind turbines can result in similar performances. The performance decreases relative to the unrestricted farm is due to the displacement of turbines from their optimum location.

Fig. 12 shows the turbine and power distribution in the two optimized Case 2 wind farm layouts. The layouts optimized under SVDD-based constraints result in configuration where no turbines are placed in the infeasible region. Fig. 12a shows that there are five turbines placed near the boundary of the feasible domain. This shows that the optimizer is attempting to place the turbines as close to their most optimum location (in the infeasible region as seen in the reference farm) as possible without violating the constraint. It can be seen that the feasible boundary is highly irregular and non-convex. This shows that the SVDD approach can also be used to efficiently enforce constraints where the feasible domain is irregular and non-convex. This drastically simplifies the constraints since an analytical description of the feasible region is not needed. The two Case 2 layouts again show that multiple arrangements of wind turbines can result in similar performances.

Table 4 shows the Case 3 optimized wind farm performance. It can be seen that the decrease in land availability due to the infeasible region in Case 3 further degrades the wind farm performance. Comparing Case 2 and Case 3, it can be seen that the Case 3 infeasibility definitions displace a greater number of wind turbines (six to seven) from their optimum locations. This suggests that the area occupied by the wind farm should also be a design objective.



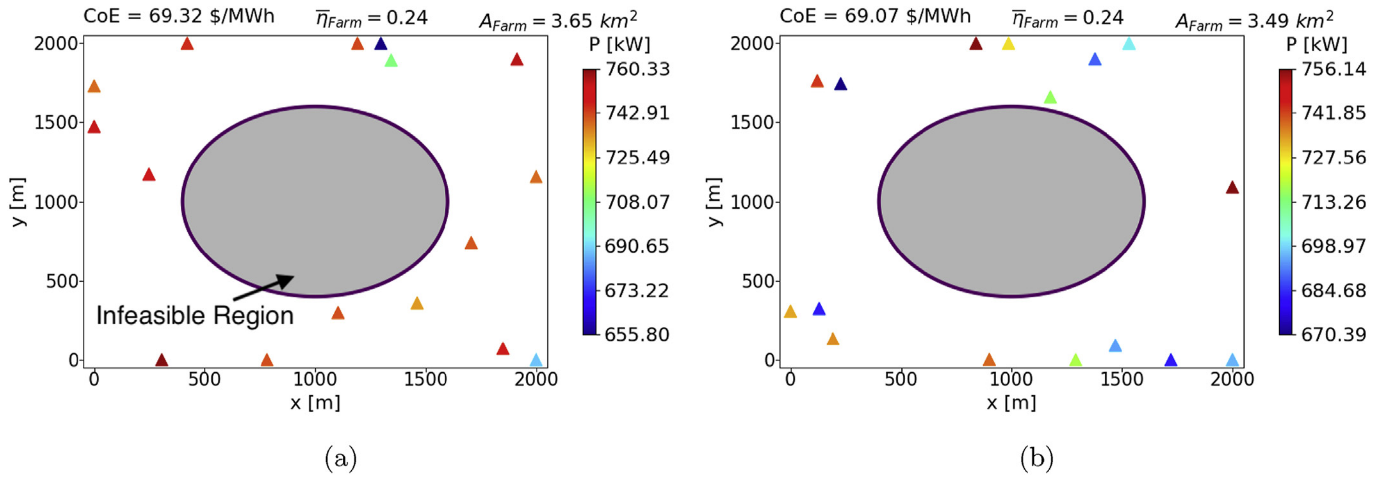
(a)

(b)

Fig. 10. Optimized reference farm layout using: a) IDW interpolation and b) RBF interpolation.

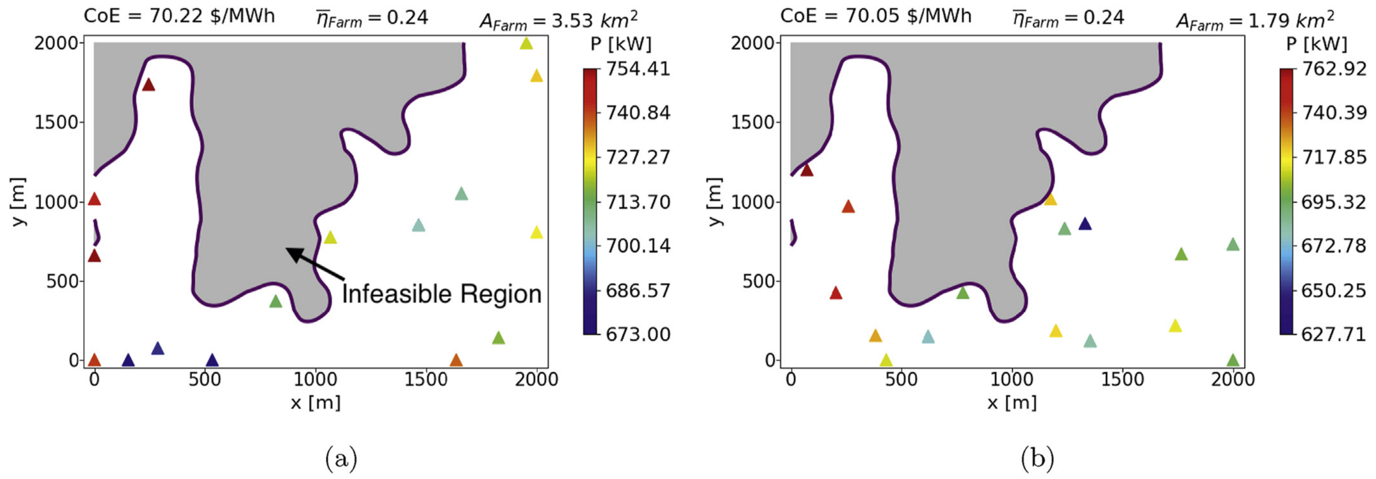
Table 3
Performance of optimized wind farms in Case 1 and Case 2 and percent changes from reference farm (in brackets).

Land Available (km_2)	Ref. Farm		Case 1		Case 2	
	4.1		2.9 (−28)		2.6 (−37)	
	RBF	IDW	RBF	IDW	RBF	IDW
Normalized AEP	1.37	1.37	1.37 (−0.2)	1.36 (−0.3)	1.35 (−1.5)	1.35 (−1.5)
Avg. Farm Power (MW)	11.75	11.75	11.44 (−2.6)	11.75 (0.0)	11.32 (−3.6)	11.52 (−2.0)
Farm efficiency	0.245	0.245	0.238 (−2.6)	0.245 (0.0)	0.236 (−3.6)	0.240 (-2.0)
COE (in $/ MWh)	68.90	69.12	69.07 (0.2)	69.32 (0.3)	70.05 (1.6)	70.22 (1.5)



(a) (b)

Fig. 11. Optimized farm layout for Case 1 using: a) IDW interpolation and b) RBF interpolation.



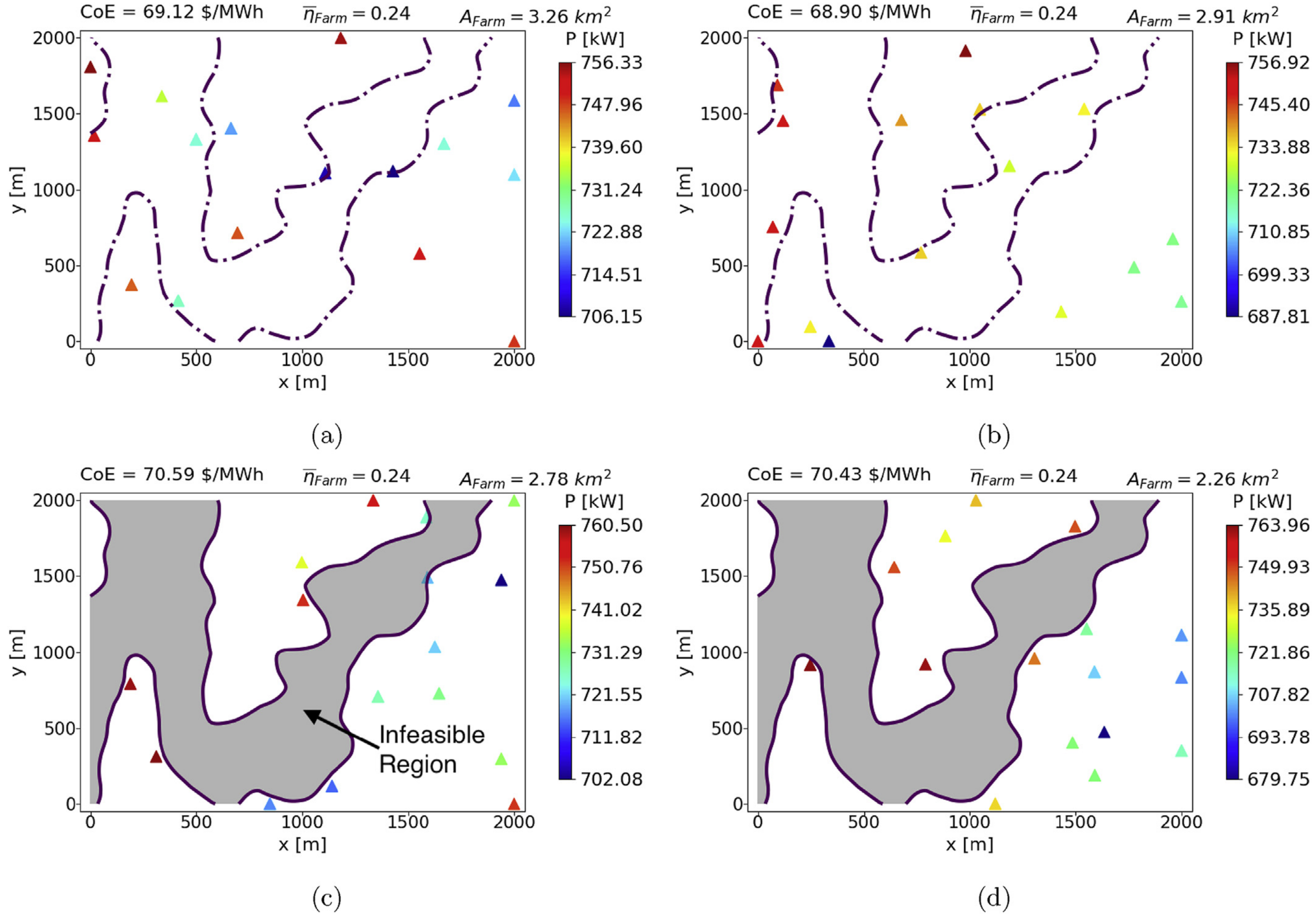
(a) (b)

Fig. 12. Optimized farm layout for Case 2 using: a) IDW interpolation and b) RBF interpolation.

Table 4
Performance of optimized wind farms in Case 3 and percent changes from reference farm (in brackets).

Land Available (km_2)	Ref. Farm		Case 3	
	4.1		2.4 (−41)	
	RBF	IDW	RBF	IDW
Normalized AEP	1.37	1.37	1.34 (−2.2)	1.34 (−2.1)
Avg. Farm Power (MW)	11.75	11.65	11.44 (−0.8)	11.74 (0.0)
Farm efficiency	0.245	0.245	0.242 (−0.8)	0.245 (0.0)
COE (in $/ MWh)	68.90	69.12	70.43 (2.2)	70.59 (2.1)

Fig. 13 shows the turbine and power distribution in the two reference farms and the two optimized Case 3 wind farm layouts. Whereas the unconstrained optimization results in six (when IDW interpolation is used) to seven (when RBF interpolation is used) turbines being placed in the infeasible regions, the layouts optimized under SVDD-based constraints result in configuration where no turbines are placed in the infeasible region. It can also be seen that several wind turbines are placed on the boundary of the infeasible regions without violating any constraints. This shows that the SVDD approach is able to accurately model not only convex and non-convex domains but also disconnected regions.



(a)

(b)

(c)

(d)

Fig. 13. Reference farm layout, with superimposed Case 3 infeasible region, obtained using: a) IDW interpolation and b) RBF interpolation and optimized farm layout for Case 3 obtained using: c) IDW interpolation and d) RBF interpolation.

5. Conclusion

The land available for wind farm construction is often not defined by rectangular or convex boundaries but by irregular, non-convex and complex boundaries. Analytical function definition for such feasible boundaries are often not possible. This work presented an efficient approach for enforcing complex land-based constraints in wind farm layout optimization. The support vector domain description (SVDD) technique was used to simplifying the complex constraint into a simpler one by mapping the region to a feature space where the complex boundaries can be represented simply as a spherical boundary. It does not require an analytical description of the domain boundary but only requires scattered points sampled from the feasible region. The SVDD model was shown to accurately model convex, non-convex, irregular, regular and disconnected domains using only data sampled from the domains. It was shown that the SVDD approach can help design wind farm layouts that are restricted to an area defined by a complex terrain boundary shape (feasible region). The approach was demonstrated by performing wind farm layout optimization using the terrain and wind conditions of the Alta Wind Energy Center in Mojave, CA, USA. All optimized layouts resulted in the placement of all turbines within the feasible region.

This work also presented an efficient method for modeling the terrain profile using spatial interpolation. The method utilizes hierarchical domain decomposition method based on quadtree to accelerate the interpolation. The quadtree-based approach was demonstrated using inverse distance weighting interpolation and radial basis function interpolation but can be extended to other forms of interpolation. It was shown that the quadtree-based interpolation resulted in an accurate and more efficient interpolation of the terrain. The presented interpolation methods were incorporated into the publicly available software WindFLO. The libSVDD library used to train the SVDD was also made publicly available.

CRediT authorship contribution statement

Sohail R. Reddy: Conceptualization, Methodology, Software, Data curation, Visualization, Investigation, Supervision, Validation, Writing - original draft, Writing - review & editing.

Declaration of competing interest

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

Acknowledgments

This research was performed while the author held an NRC Research Associateship award at the Naval Postgraduate School.

References

[1] A.A. Bazmi, G. Zahedi, "Sustainable energy systems: role of optimization modeling techniques in power generation and supply—a review, Renew. Sustain. Energy Rev. 15 (8) (2011) 3480–3500.

[2] A. Evans, V. Strezov, T.J. Evans, Assessment of sustainability indicators for renewable energy technologies, Renew. Sustain. Energy Rev. 13 (5) (2009) 1082–1088.

[3] D.S. Ryberg, M. Robinius, D. Stolten, Evaluating land eligibility constraints of renewable energy sources in Europe, Energies 11 (May 2018) 1–19.

[4] P.Y. Zhang, D.A. Romero, J.C. Beck, C.H. Amon, Solving wind farm layout optimization with mixed integer programs and constraints programs, EURO J. Comput. Optim. 2 (2014) 195–219.

[5] L. Wang, A.C. Tan, Y. Gu, J. Yuan, A new constraint handling method for wind farm layout optimization with lands owned by different owners, Renew. Energy 83 (2015) 151–161.

[6] J. Feng, W.Z. Shen, Wind farm layout optimization in complex terrain: a preliminary study on a Gaussian Hill, J. Phys. Conf. 524 (2014), 012146.

[7] R. Brogna, J. Feng, J.N. Sorensen, W.Z. Shen, F. Porte-Agel, A new wake model and comparison of eight algorithms for layout optimization of wind farms in complex terrain, Appl. Energy 259 (February 2020).

[8] J. Feng, W.Z. Shen, Y. Li, An optimization framework for wind farm design in complex terrain, Appl. Sci. 8 (October 2018) 1–18.

[9] P. Mittal, K. Mitra, Determination of optimal layout of wind turbines inside a wind farm in presence of practical constraints, in: *2019 Fifth Indian Control Conference (ICC)*, (New Delhi, India), 2019, pp. 353–358.

[10] Y. Chen, H. Li, B. He, P. Wang, K. Jin, Multi-objective genetic algorithm based innovative wind farm layout optimization method, Energy Convers. Manag. 105 (September 2015) 1318–1327.

[11] S.Y.D. Sorkhabi, D.A. Romero, G.K. Yan, M.D. Gu, J. Moran, M. Morgenroth, C.H. Amon, The impact of land use constraints in multi-objective energy-noise wind farm layout optimization, Renew. Energy 85 (2016) 359–370.

[12] S.Y.D. Sorkhabi, D.A. Romero, C. Beck, C.H. Amon, Constrained multi-objective wind farm layout optimization: novel constraint handling approach based on constraint programming, Renew. Energy 126 (2018) 341–353.

[13] S.R. Reddy, libSVDD: A Library for Support Vector Domain Description, 2020, https://doi.org/10.5281/zenodo.3712443. https://github.com/sohailrreddy/libSVDD.

[14] L. Wang, A.C. Tan, M.E. Cholette, Y. Gu, Optimization of wind farm layout with complex land divisions, Renew. Energy 105 (2017) 30–40.

[15] J.S. Gonzalez, A.L.T. Garcia, M.B. Payan, J.R. Santos, A.G.G. Rodriguez, Optimal wind-turbine micro-siting of offshore wind farms: a grid-like layout approach, Appl. Energy 200 (2017) 28–38.

[16] S.S. Perez-Moreno, K. Dykes, K. Merz, M. Zaaijer, Multidisciplinary design analysis and optimization of a reference offshore wind plant, J. Phys. Conf. 1037 (2018) 1–15.

[17] A.P.J. Stanley, A. Ning, Massive simplification of the wind farm layout optimization problem, Wind Energy Sci. 4 (2019) 663–676.

[18] D.M. Tax, R.P. Duin, Support vector domain description, Pattern Recogn. Lett. 20 (1999) 1191–1199.

[19] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.

[20] N. Aronszajn, Theory of reproducing kernels, Trans. Am. Math. Soc. 68 (May 1950) 337–404.

[21] F.J. Hasiuk, C. Harding, A.R. Renner, E. Winer, TouchTerrain: a simple web-tool for creating 3D-printable topographic models, Comput. Geosci. 109 (2017) 25–31.

[22] R. Gradka, A. Kwinta, A short review of interpolation method used for terrain modeling, Geomatics Landmanag. Landsc. 4 (2018) 29–47.

[23] Z. You-Long, H. Fa-Long, Z. Can-Can, L. Chao-Liu, D. Keh-Jim, Analysis of radial basis function interpolation approach, Appl. Geophys. 10 (December 2013) 397–410.

[24] R.L. Hardy, Multiquadric equations of topography and other irregular surfaces, J. Geophys. Res. 76 (March 1971) 1905–1915.

[25] S.R. Reddy, Wind Farm Layout Optimization (WindFLO): A Framework for Fast Wind Farm Layout Optimization, 2019, https://doi.org/10.5281/zenodo.3694675. https://github.com/sohailrreddy/WindFLO.

[26] S. R. Reddy, "Wind farm layout optimization (WindFLO) : an advanced framework for fast wind farm analyis and optimization," Appl. Energy, vol. 269, pp. 115090–115091–115090–13, May 2020.

[27] Vestas, V90-3.0MW: an Efficient Way to More Power, 2008.

[28] Midwestern Regional Climate Center, Cli-MATE,, 2000 date retrieved December, 2019, https://mrcc.illinois.edu/CLIMATE/.

[29] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (Apr. 2002) 182–197.

[30] S. Rodrigues, P. Bauer, P.A. Bosman, "Multi-Objective optimization of wind farm layouts – complexity, constraint handling and scalability, Renew. Sustain. Energy Rev. 65 (2016) 587–609.

[31] S.R. Reddy, M.K. Scharrer, F. Pichler, D. Watzenig, G.S. Dulikravich, Accelerating parameter estimation in Doyle-Fuller-Newman model for lithium-ion batteries, COMPEL: Int. J. Comput. Math. Electr. Electron. Eng. 38 (5) (2019) 1533–1544.

[32] S.R. Reddy, G.S. Dulikravich, MOHO - A Many-Objective Hybrid Optimizer, in: Swarm and Evolutionary Computation, Press, 2020.

[33] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (Apr. 1997) 67–82.

[34] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (Aug. 2014) 577–601.

[35] S.R. Reddy, G.S. Dulikravich, Many-objective differential evolution optimization based on reference points: NSDE-R, Struct. Multidiscip. Optim. 60 (2019) 1455–1473.

[36] K. Li, K. Deb, Q. Zhang, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Trans. Evol. Comput. 19 (Oct. 2015) 694–716.

[37] S.R. Reddy, Many-objective Hybrid Optimization under Uncertainty with Applications, PhD thesis, Florida International University, Miami, FL, USA, 2019.