Discrete optimization for down selection of positions in offshore wind farm layouts

Spyridon Giaroslav Acheimastos

DTU Wind-M-0979
August 2025

Author:
Spyridon Giaroslav Acheimastos

Title:
Discrete optimization for down selection of positions in offshore wind farm layouts

DTU Wind-M-0979

ECTS: 45

Education: Master of Science

Supervisor(s):

Pierre-Elouan Mikael Réthoré
Mikkel Friis-Møller
DTU Wind & Energy Systems

Anna Carcia-Teruel
Abhinav Kapila
RWE Renewables Danmark A/S

Michiel Zaayer
TU Delft

Remarks:
This report is submitted as partial fulfillment of the requirements for graduation in the above education at the Technical University of Denmark.

DTU Wind & Energy Systems is a department of the Technical University of Denmark with a unique integration of research, education, innovation and public/private sector consulting in the field of wind energy. Our activities develop new opportunities and technology for the global and Danish exploitation of wind energy. Research focuses on key technical-scientific fields, which are central for the development, innovation and use of wind energy and provides the basis for advanced education at the education.

Technical University of Denmark Department of Wind and Energy Systems Frederiksborgvej 399 4000 Roskilde
Denmark
www.wind.dtu.dk

Discrete Optimization for Down Selection of Positions in Offshore Wind Farm Layouts

by

Spyridon Giaroslav Acheimastos

to obtain the degrees of

Master of Science
in Wind Energy
at Technical University of Denmark

Master of Science
in Aerospace Engineering
at Delft University of Technology

To be defended on the 8th of September 2025

Supervisors:	Michiel Zaayer	TU Delft
	Pierre-Elouan Mikael Réthoré	DTU
	Mikkel Friis-Møller	DTU
	Anna Carcia-Teruel	RWE
	Abhinav Kapila	RWE
Defence committee:	Wei Yu	TU Delft
	Julian Antony Quick	DTU
Start date:	November 1st, 2024	
End date:	August 31st, 2025	
Student numbers:	TU Delft:	6053572
	DTU:	233800

An electronic version of this thesis is available at
`https://repository.tudelft.nl` and `https://findit.dtu.dk`

DTU

EWEM

TU Delft

Acknowledgements

I would first like to express my deepest gratitude to my supervisors, Michiel Zaayer, Pierre-Elouan Mikael Réthoré, Mikkel Friis-Møller, Anna Garcia-Teruel, and Abhinav Kapila, for their continuous guidance, support, and invaluable insights throughout this thesis. Although it was sometimes challenging to juggle the input of five brilliant minds, I would not have reached this point without the contributions of each of you.

I also thank the members of my defence committee, Wei Yu and Julian Antony Quick, for their time, effort, and constructive feedback. I am equally grateful to RWE Renewables for selecting me for this project and for providing the environment and resources that made this work possible.

This journey would not have been possible without the generous financial support of the Bodossaki Foundation and the Union of Greek Shipowners (UGS). Their scholarships truly gave me a golden ticket and enabled me to pursue this dream.

I am grateful to my fellow EWEM classmates for sharing this amazing journey with me across different countries, cultures, and experiences. I also cherish the countless memories with my friends from Lundto E0 during my time in Denmark, as well as the friendships, both old and new, that have accompanied me along the way.

Finally, I owe my deepest thanks to my family for their unwavering love, encouragement, and for instilling in me the values that carried me through this journey.

Spyros Acheimastos
Copenhagen, August 2025

Abstract

Offshore wind farm design increasingly faces the challenge of adapting layouts to larger turbines as technology advances. A promising approach is down-selection, where layouts designed for smaller turbines are adapted to higher-rated machines by selecting a subset of positions. This thesis investigates two central questions: which optimisation strategies are suitable for the down-selection problem, and what is the impact of down-selection on the final wind farm layout. The first question was addressed through a structured evaluation of candidate algorithms, including a literature-based ranking, parameter tuning, and comparative testing across multiple cases. The analysis showed that Gradient-Based methods and Greedy Heuristics were the most effective strategies, with complementary strengths: Gradient-Based approaches offered scalability and computational efficiency, while Greedy Heuristics achieved higher energy yields under different occupancy conditions. The second question was addressed by comparing down-selected farms with layouts directly optimised for larger turbines. Across all test cases, the down-selected layouts achieved annual energy productions within 0.15% of the optimised layouts, showing that down-selection can closely replicate optimal performance. The small residual differences were governed primarily by spacing constraints, with compatibility between initial and final requirements leading to near-lossless performance. Down-selection also influenced turbine distribution, with more retained along site perimeters, which may affect secondary design drivers such as cabling or support structures. Overall, the results demonstrate that down-selection is a viable and efficient design strategy, provided that algorithm selection and parameter tuning are carefully matched to the problem context, and that future spacing requirements are anticipated at the design stage. These insights underline the potential of down-selection to reduce computational and economic costs in wind farm development while maintaining energy yield, and point towards its integration within robust design frameworks for future turbine upgrades.

Contents

List of Figures

List of Tables

CHAPTER 1

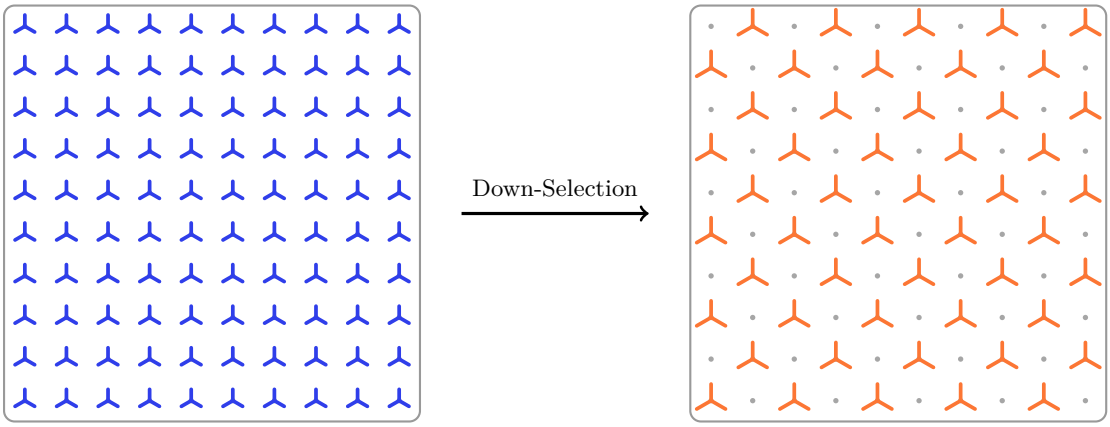
Introduction

1.1 Motivation and Context

Over the past few decades, the effects of global warming have become increasingly evident to all. To avoid reaching a point of no return and mitigate the risks and impacts of climate change, the Paris Agreement [45] was signed by 196 Parties during the UN Climate Change Conference (COP21). This landmark treaty aims to limit the global temperature increase to 1.5°C above pre-industrial levels. As highlighted by Gielen et al. [17], renewable energy and energy efficiency measures could achieve 94% of the emission reductions needed to meet these targets. Specifically, renewable energy is projected to supply 63% of the total primary energy by 2050, a significant increase from 15% in 2015, with wind energy alone contributing 24% of this share. In line with these ambitions, the Global Wind Energy Council (GWEC) [47] reports that over 20 governments have committed to installing 380 GW of offshore wind capacity by 2030 and 2000 GW by 2050. Notably, 2023 emerged as the second-best year on record, with the global offshore wind industry connecting 11 GW of power to the grid, representing a 24% year-on-year increase.

With the increasing demand for offshore wind farms (OWFs), optimizing the development of new wind farms has become more critical than ever. Designing efficient layouts for wind farms is essential to maximize energy production while keeping costs low. A key challenge lies in the high cost of conducting seabed surveys, which are crucial for assessing site conditions. For instance, a seabed survey for a 1 GW wind farm in the UK can cost approximately £8 million, according to BVG Associates [1]. To address this, companies often create optimized layouts for potential wind farms before performing detailed surveys, reducing the scope of these costly assessments to the proposed turbine positions. Furthermore, offshore wind projects require a lengthy development period, typically spanning 6–8 years from initial planning to commercial operation, according to GWEC [47]. Over this timeframe, advancements in turbine technology can lead to the availability of higher-rated turbines. This could enable developers to achieve the same wind farm capacity with fewer turbines, reducing costs and requiring only a subset of the originally surveyed layout positions.

Because layout optimizations and site surveys are performed early in the development process, the set of feasible turbine positions is fixed in advance. When higher-rated turbines become available, developers must therefore choose a subset of these predefined positions to deploy them. This process, referred to as *down-selection*, involves selecting the most suitable turbine locations from an existing layout originally optimized for lower-rated turbines. The challenge arises from the complex wake interactions between turbines, which strongly influence overall energy production. Since the candidate positions are restricted to a predefined set of coordinates, the task naturally takes the form of a discrete optimization problem.



Initial Layout
(100 smaller turbines)

Down-Selected Layout
(50 larger turbines)

Figure 1.1: Conceptual illustration of down-selection. An initial layout with many smaller turbines is reduced to a subset of positions, resulting in fewer larger turbines on the same site.

The scale of this problem is enormous: even for 100 candidate positions with 50 turbines to be placed, the number of possible layouts is so large that evaluating the performance of all individually would take longer than the age of the universe (see Section 2.1 for derivation). As a result, exhaustive search is infeasible, and smart optimization strategies are required instead. This project therefore investigates optimization strategies for down-selection and evaluates how this process affects the resulting wind farm layout.

1.2 Literature Review of Optimization Approaches for Down-Selection

This section reviews existing approaches to wind farm layout optimization in order to identify methods relevant to the down-selection problem. The aim is to establish what is known about turbine placement strategies, what methods have proven effective, and where knowledge gaps remain.

When a wind turbine converts the kinetic energy of the wind into electricity, it creates a wake of slower-moving, more turbulent air behind it (Manwell et al. [23]). In OWFs, where turbines are densely packed to optimize space and infrastructure costs, these wakes reduce the energy available to downstream turbines, leading to efficiency losses. Wind farm layout optimization therefore seeks to determine turbine placements that mitigate wake effects and improve overall farm performance. The optimization process can be driven by various objectives, such as maximizing annual energy production (AEP), minimizing the cost of energy (COE), or maximizing profitability (Herbert-Acero et al. [18]).

Since wind farm layout optimization involves complex, nonlinear wake interactions (e.g. superimposing wake deficits generated using the Jensen [20] wake model), gradient-free (GF) optimization methods are widely used (Azlan et al. [3]). A common approach is to discretize the available area into a grid of potential turbine locations and use optimization techniques to determine the optimal subset. Studies have shown success with genetic algorithms (Mosetti

et al. [28]), particle swarm optimization (Hou et al. [19]), greedy algorithms (Chen et al. [11]), extended pattern search (Pont and Cagan [33]), and random search (Feng and Shen [14]). While GF algorithms are easy to implement, they become computationally expensive as the number of design variables increases (Martins and Ning [24]).

An alternative to heuristic GF methods is is to use exact optimization formulations. Turner et al. [44] formulated the problem as a mixed-integer linear (MILP) and quadratic integer (QIP) optimization model to bypass the complexities of nonlinear wake interactions. Similarly, MirHassani and Yarahmadi [25] explored the impact of turbine hub height variation on layout optimization using a similar approach, supplemented by a heuristic iterative method to improve computational efficiency. However, as the problem size grows, exact approaches can become increasingly time-consuming, making them less practical for large-scale wind farms.

Recently, gradient-based (GB) methods have gained interest due to their lower computational cost and ability to handle large-scale optimization with many constraints, despite being prone to local optima (Herbert-Acero et al. [18]). Thomas and Ning [42] demonstrated that their GB approach achieved a 10% higher mean AEP than a genetic algorithm while requiring nearly three orders of magnitude fewer function evaluations for a 38-turbine case. Quick et al. [37] explored stochastic gradient descent (SGD), a machine learning technique, and found that it achieved larger AEP in substantially less time compared to its deterministic counterpart as the number of turbines increased. To address local minima issues, multiple optimizations from different initial conditions can be used, a strategy known as multi-start. Valotta et al. [46] demonstrated that the Smart-Start heuristic significantly reduces the number of multi-starts required for GB methods. However, deriving analytical gradients can be time-consuming, which is why Quaeghebeur et al. [36] proposed a heuristic pseudo-gradient approach, noting that their pseudo-gradient approach has also been applied in industry for large-scale offshore wind farm design.

Most studies, including many mentioned above, focus on optimizing a fixed number of wind turbines. However, when the number of turbines is also a variable, the problem becomes more complex due to the inclusion of discrete decision variables. Mosetti et al. [28] were the first to address wind farm layout optimization with a variable number of turbines, using a genetic algorithm to determine the optimal placement and number of turbines on a 100×100 grid. More recently, Stanley et al. [41] explored this problem using a greedy algorithm, a genetic algorithm, and a repeated sweep algorithm, finding that both the layout and the optimal number of turbines vary significantly depending on the chosen objective function (AEP, COE, or profit). Pollini [32] observed that variable turbine numbers have mostly been handled with gradient-free methods, and showed that a gradient-based approach can achieve optimal layouts much faster than a genetic algorithm. Although these studies treat the turbine count as a decision variable, the strategies they employ could also be adapted to cases where the number of turbines is predetermined but layout optimization remains challenging.

Comparing different optimization methods in complex and realistic wind farm scenarios is challenging due to variations in implementation, modeling assumptions, and problem formulations. To address this, Baker et al. [4] conducted a blind study within Task 37 of the International Energy Agency Wind Technology Collaboration Programme (hereafter referred to as IEA-37), evaluating various optimization methods and wake models for circular wind farms of 16, 36, and 64 turbines. More recently, Thomas et al. [43] compared eight optimization methods using

a standardized approach, ensuring that each method was applied by experts in that specific technique. Their study, based on the IEA-37 case study 4, modeled after the Borssele III and IV wind farms with 81 turbines, demonstrated that all methods improved AEP by clustering turbines along the outer boundaries while maintaining a more dispersed arrangement within the inner regions of the farm. However, their findings reinforce a common conclusion in optimization research: there is no single best algorithm. The effectiveness of an optimization method depends on the specific problem, its constraints, and the available computational resources. This observation is not unique to wind farm layout optimization, as similar conclusions have been drawn in other fields (Reddy and Kumar [38]). Nevertheless, incorporating problem-specific knowledge and heuristics can significantly enhance solution efficiency.

Offshore wind farm design extends beyond turbine layout, as multiple components, such as support structures and electrical systems, influence overall performance and cost. Perez-Moreno et al. [30] compared two approaches for preliminary offshore wind farm design: a traditional sequential method, where each component is optimized independently, and a multidisciplinary design analysis and optimization (MDAO) approach, which simultaneously optimizes all components with the levelized cost of energy (LCOE) as the global objective. Their study demonstrated that the integrated MDAO approach consistently leads to lower LCOE compared to the sequential method, highlighting the importance of considering interactions between different wind farm subsystems.

In summary, researchers have explored a wide range of wind-farm layout strategies, from GF heuristics (genetic algorithms, particle swarm, greedy, pattern and random searches), through exact optimization formulations (e.g., MILP, QIP), and more recently GB and SGD methods with multi-start and pseudo-gradient enhancements. Some of these methods have been extended to cases with variable turbine counts, and large comparison studies have confirmed that no single algorithm consistently outperforms across all farm sizes and objectives. Multidisciplinary frameworks further demonstrate the benefits of co-optimizing support structures, electrical systems, and turbine placement to minimize LCOE.

Taken together, the literature shows that many different approaches have been developed to handle the challenges of wind farm layout optimization, and that they can be applied under a variety of objectives and problem formulations. Strategies for selecting a subset of turbine positions have also been studied, including cases where the number of turbines is treated as a variable. What has not been explored is the case where an initial layout, already optimized for lower-rated turbines, exists in advance, and a subset of those positions must later be chosen when upgrading to higher-rated turbines in order to maintain the same farm capacity. This sequential, two-stage perspective distinguishes the down-selection problem considered here from the broader literature. Developing systematic approaches for this setting is therefore necessary to strengthen the business case for OWF upgrades. Looking ahead, such methods could also be integrated into robust optimization frameworks that account for turbine-rating uncertainty from the outset, but this lies beyond the scope of the present study and may be a valuable direction for future research.

1.3 Research Questions

The literature review revealed that while many optimization strategies for wind farm layouts exist, none directly address the down-selection problem. To investigate this gap, the following research questions are formulated, each supported by sub-questions that break the problem into more specific components.

RQ1. *Which optimization strategies are suitable for down-selection?*

1. Which strategies are most suitable for down-selection, based on defined requirements?
2. How do the top-ranked strategies compare across cases of increasing complexity?

RQ2. *What is the impact of down-selection on the final wind farm layout?*

1. How does the performance of the down-selected farm compare to an optimal layout designed directly for higher-rated turbines?
2. How does the performance of the down-selected farm change when the initial layout already satisfies the minimum spacing for higher-rated turbines?
3. How does the comparison between down-selected and directly optimized farms change when all layouts are generated using a larger minimum spacing constraint?
4. How does down-selection influence the spatial distribution of turbines within the layout?

1.4 Scope

The scope of this thesis is restricted to the technical aspects of down-selection in offshore wind farms. The analysis focuses on wake interactions and their effect on energy production when higher-rated turbines replace lower-rated ones in pre-optimized layouts. AEP is the sole performance metric considered, providing a first step that isolates the aerodynamic effects of down-selection before such work can be extended to more complex cost-driven analyses.

Several aspects of offshore wind farm design are not considered in this work, including:

- Economic modelling (turbine, installation, and operational costs)
- Site-specific constraints such as bathymetry and seabed conditions
- Foundation and structural design
- Electrical infrastructure and grid connection

1.5 Report Outline

The remainder of this thesis is structured as follows. Chapter 2 provides the necessary background on down-selection in offshore wind farm layout optimization, optimization strategies, and wake modeling. Chapter 3 introduces the general methodology, presenting the mathemati-

cal definition of the down-selection problem, the objective function, and constraints. Chapter 4 addresses RQ1 by evaluating and comparing different optimization strategies for down-selection. Chapter 5 investigates RQ2 by analyzing the impact of down-selection on the resulting layouts when higher-rated turbines are introduced. Chapter 6 summarizes the main findings and discusses directions for future work. Finally, supporting material is provided in the Appendices (Appendices A to C).

CHAPTER 2

Background

This chapter introduces the essential concepts required to understand the methodology and results of this thesis. Basic knowledge of wind turbines, aerodynamics, and general optimization principles is assumed. For further background, the reader is referred to standard references such as Manwell et al. [23] for wind energy, and Martins and Ning [24] and Audet and Hare [2] for optimization and algorithms.

The chapter is structured as follows. Section 2.1 introduces the down-selection problem within wind farm layout optimization and highlights its combinatorial complexity. Section 2.2 reviews optimization strategies that are commonly applied to such problems. Finally, Section 2.3 outlines the wake modeling approach adopted in this thesis, which underpins the evaluation of all candidate layouts.

2.1 Down-Selection in Wind Farm Layout Optimization

Wind farm layout optimization seeks to determine the placement of turbines within a site in order to maximize annual energy production (AEP) or minimize the levelized cost of energy (LCOE), subject to practical constraints such as site boundaries, bathymetry, and minimum spacing between turbines [18]. The challenge arises from the strong wake interactions between turbines, which introduce nonlinearities and multiple local optima into the problem [23, 20].

In many studies, the available area is discretized into a grid of potential positions, and optimization consists of selecting the best subset of these points [28]. In practice, the situation is often even more restrictive: once site surveys have been completed, the set of feasible turbine positions is fixed in advance. When higher-rated turbines later become available, developers must then choose a subset of these predefined locations to maintain capacity with fewer turbines. This sequential, two-stage setting defines the *down-selection problem*, which is the focus of this thesis.

Consider an OWF with N candidate positions and a requirement to install exactly k turbines. Each choice of k positions defines a unique layout, and since the ordering of turbines is irrelevant, the total number of possible configurations is given by the binomial coefficient:

$$C(N,k) = \binom{N}{k} = \frac{N!}{k!\,(N-k)!} \qquad k \leq N \tag{2.1}$$

The growth of this search space is explosive. For example, an OWF with $N = 100$ available positions and $k = 50$ turbines yields approximately $C(100, 50) \approx 10^{29}$ unique layouts. Even if the AEP of each layout could be computed in just one nanosecond, the total time required would exceed the age of the universe (13.8 billion years $\approx 4.35 \times 10^{26}$ nanoseconds). Exhaustive search is therefore infeasible, even with state-of-the-art hardware.

This combinatorial complexity motivates the use of dedicated optimization strategies, such as greedy heuristics or gradient-based approaches, which can efficiently search the solution space without evaluating every possible configuration.

2.2 Optimization Strategies

The combinatorial complexity of the down-selection problem makes exhaustive search infeasible, motivating the use of optimization strategies that can efficiently explore the design space and identify high-quality layouts within practical time limits. These strategies can be organized into broad categories, as outlined in Section 2.2.1, and specific representative methods are introduced in Section 2.2.2. This classification and selection provide the foundation for the comparative evaluation presented later in Section 4.2.

2.2.1 Classification of Strategies

The strategies considered in this thesis span several classes of optimization methods. *Gradient-free* (GF) approaches operate without derivative information, making them suitable for discrete or discontinuous problems. *Gradient-based* (GB) approaches, in contrast, exploit analytical or numerical derivatives to guide the search more efficiently but require continuous variables and smoothness, which are generally less applicable here.

Beyond these, methods can be broadly divided into three additional categories: *Heuristics*, which apply simple problem-specific rules; *Metaheuristics*, which are higher-level, problem-independent frameworks that iteratively improve solutions via systematic exploration; and *Exact* methods, which rely on mathematical programming formulations that can in principle guarantee global optimality.

A more detailed discussion of these categories and their theoretical properties can be found in Martins and Ning [24]. Here, only the aspects directly relevant to the down-selection problem are highlighted.

2.2.2 Candidate Strategies

Based on the problem characteristics discussed in Section 2.1 and an extensive review of the literature, ten strategies were identified as the most promising for solving the down-selection problem. Methods with overlapping principles or limited applicability were omitted to maintain focus and clarity. Each algorithm is briefly described below, while the detailed ranking is presented in Section 4.2.

2.2.2.1 Gradient-Based (GB)

This method relaxes the problem by using continuous variables (values between 0 and 1) and uses gradient-based solvers to adjust them, while using interpolation penalties to push solutions toward clear 0 or 1 selections. [32, 31]

2.2.2.2 Genetic Algorithm (GA)

Metaheuristic that represents candidate layouts as "individuals" in a population and evolves them over generations via crossover (recombining layouts) and mutation (random tweaks). [28, 40, 32]

2.2.2.3 Binary Particle Swarm Optimization (BPSO)

Metaheuristic that models each layout as a "particle" in a discrete search space, with particles nudged toward their own and group-wide best solutions using velocity-inspired updates. [27, 24]

2.2.2.4 Variable Neighborhood Search (VNS)

Metaheuristic that systematically explores increasingly distant neighborhoods of a current layout, moving outwards in search space until a better layout is found. [26, 10, 35]

2.2.2.5 Random Search (RS)

Gradient-free heuristic that begins from an initial feasible layout and iteratively applies random perturbations, moving a randomly selected turbine by a random direction and distance, and accepting layout updates when improvement occurs. [14]

2.2.2.6 Greedy Heuristic (GH)

Gradient-free heuristic that starts with an empty layout and iteratively adds turbines one-by-one, keeping the ones that maximize the objective. It is then followed by a second refinement stage. [12, 41, 46]

2.2.2.7 Repeated Sweep (RSW)

A gradient-free heuristic that starts from a random feasible layout and iteratively flips sites (0–1 or 1–0) whenever the objective improves. Two trade phases then follow, sequentially swapping sites with their right and upper neighbors if improvement is achieved, thereby emulating a gradient. [41]

2.2.2.8 Discrete Exploration-Based Optimization (DEBO)

Gradient-free, two-stage heuristic that first greedily allocates turbines to discrete grid locations, then refines the layout through a local search that iteratively moves one turbine at a time to improve the objective. It produced the best layout in terms of AEP among the eight tested algorithms. [43]

2.2.2.9 Discrete Perturbation Algorithm (DPA)

Gradient-free heuristic that discretizes the design space into a grid and optimizes turbine placement by applying both directed and random perturbations from existing positions. It produced the second best layout in terms of AEP among the eight tested algorithms. [43]

2.2.2.10 Mixed-Integer Linear Programming (MILP)

An exact method that formulates the layout problem with binary variables in a mixed-integer linear program. Wake effects are simplified and pre-computed in an interaction matrix, which is then used as a lookup table during optimization. Guarantees optimal or near optimal solutions. [44, 25, 34]

2.3 Wake Modeling in Layout Optimization

When a wind turbine extracts energy from the flow, it leaves behind a wake of reduced velocity and increased turbulence. In offshore wind farms, where turbines are spaced relatively close together, these wakes reduce the inflow available to downstream turbines and therefore have a strong influence on overall energy production. Accurately capturing wake interactions is thus essential for layout optimization, but it must also be done at a computational cost low enough to allow repeated evaluations during the optimization process.

Several simplified engineering models have been developed for this purpose. The classical Jensen or "top-hat" model [20] has been applied extensively in both research and industry due to its robustness and simplicity [10]. However, it represents the wake as a uniform velocity deficit across its cross-section, which can lead to unrealistic discontinuities in optimization studies. Improved models based on Gaussian-shaped deficits provide a smoother and more realistic description of wake recovery while remaining computationally efficient [8]. Comparative assessments of wake models and their uncertainties can be found in Barthelmie et al. [6] and Gao et al. [16].

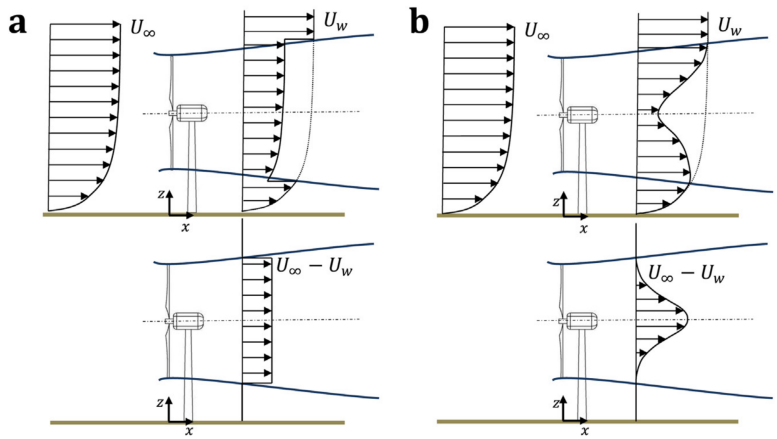


Figure 2.1: Schematic of vertical profiles of the mean velocity (top) and velocity deficit (bottom) downwind of a wind turbine assuming: (a) a top-hat distribution and (b) a Gaussian distribution. Adapted from Bastankhah and Porté-Agel [7].

In this thesis, the modified Gaussian model of Bastankhah and Porté-Agel [8], implemented following Thomas and Ning [42], is adopted. This formulation avoids the undefined near-wake region that complicates optimization and has been widely used in recent benchmark and comparison studies. A detailed description of the implementation, parameter choices, and superposition method is provided in Section 3.1.3.2.

CHAPTER 3

General Methodology

This chapter outlines the general methodology used to address the two research questions:

RQ1. *Which optimization strategies are suitable for down-selection?*

RQ2. *What is the impact of down-selection on the final wind farm layout?*

The overall approaches are summarized in Figs. 3.1 and 3.2, which provide a high-level overview of the steps taken for each research question.

Figure 3.1: Flowchart for answering RQ1.

For RQ1, the process consists of three stages. First, candidate strategies are ranked based on a set of requirements specific to down-selection. Second, the most promising strategies are finetuned on a simplified benchmark case. Finally, their performance is evaluated across a series of test cases of increasing complexity.

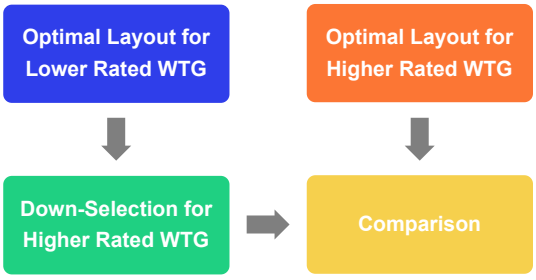


Figure 3.2: Flowchart for answering RQ2.

To address RQ2, a different process is employed. First, optimal layouts are generated for farms using lower-rated turbines. These layouts serve as the starting point for down-selection to fewer, higher-rated turbines. To assess the impact of this process, the results are compared against layouts optimized for the higher-rated turbines.

To ensure clarity and focus, each question is addressed separately in Chapters 4 and 5, respectively. This separation is also intentional, as the findings from RQ1 directly inform the down-selection strategy used to answer RQ2. The present chapter introduces the concepts, equations, and methods common to both research questions, including the mathematical formulation of the down-selection problem (Section 3.1) and the computational setup in terms of software and hardware resources (Section 3.2).

3.1 Down-Selection Problem Formulation

This section defines the optimization problem underlying the down-selection task. The problem is first presented in its general form (Section 3.1.1), highlighting the key elements of the formulation. The subsequent subsections then describe each of its individual components in detail: the optimization variables (Section 3.1.2), the objective function (Section 3.1.3), and the constraints (Section 3.1.4).

3.1.1 Mathematical Definition

The down-selection task can be formulated as a discrete optimization problem, where a fixed number k of turbine positions is chosen from a larger set of N candidates. The goal is to maximize the expected energy yield, subject to a fixed turbine count and a minimum separation distance between any selected turbines. The general formulation is:

$$\begin{aligned} \text{maximize} \quad & \mathrm{AEP}(\mathbf{b}) \\ \text{subject to} \quad & \sum_{i=1}^{N} b_i = k \\ & S_{ij} \geq S_{min} \qquad \forall\, i \neq j \\ & b_i \in \{0,1\} \qquad i = 1, \ldots, N \end{aligned} \tag{3.1}$$

Here, $\mathbf{b}$ is a binary vector with $b_i = 1$ if position i is selected and $b_i = 0$ otherwise. The first constraint enforces the number of turbines, while the second ensures that any two selected turbines respect the minimum separation distance S_{min}.

3.1.2 Optimization Variables

As introduced in Eq. (3.1), the problem involves selecting exactly k turbine locations out of N available candidates. To represent this choice, a binary decision variable b_i is associated with each position $i = 1, \ldots, N$, where $b_i = 1$ if the position is selected and $b_i = 0$ otherwise. The complete layout is then described by the binary vector $\mathbf{b} = [b_1, \ldots, b_N]$. This representation makes the problem combinatorial, with $C(N, k)$ possible configurations (see Eq. (2.1)).

3.1.3 Objective Function

As introduced in Eq. (3.1), the optimization problem is driven by the maximization of annual energy production (AEP). Its computation relies on two components: the calculation of AEP from wind statistics and turbine power output, and the modeling of wake losses that affect the effective wind speed at each turbine. These are described in the following subsections.

3.1.3.1 Annual Energy Production

Among the possible objective functions, this work adopts the maximization of total annual energy production (AEP), expressed as

$$\mathrm{AEP} = 8760 \sum_{p=1}^{N_p} \sum_{q=1}^{N_q} f_{p,q}\, P_{p,q}\,, \tag{3.2}$$

where N_p and N_q denote the numbers of wind-direction and wind-speed bins, $f_{p,q}$ is the joint probability of wind direction p and wind speed q, and $P_{p,q}$ is the farm power output under those conditions. The farm power output is obtained as

$$P_{p,q} = \sum_{i=1}^{N} b_i\, P_i(v_{\mathrm{eff},i}^{(p,q)}) \tag{3.3}$$

where $v_{\mathrm{eff},i}^{(p,q)}$ is the effective wind speed experienced by turbine i under wind direction p and wind speed q.

The turbine power $P_i(v)$ is evaluated from a tabulated power curve. Below cut-in and above cut-out wind speeds the power is set to zero, while values between the tabulated points are obtained by linear interpolation. The maximum of the curve corresponds to the rated power. $C_T(v)$ is handled in the same way, using its corresponding tabulated curve.

3.1.3.2 Wake-Loss Model

Wake deficits are modeled using the modified Gaussian formulation of Bastankhah and Porté-Agel [8], as implemented by Thomas and Ning [42]. This version avoids the undefined near-wake region that hinders optimization, by linearly interpolating the deficit from the rotor to the point where the analytical solution becomes valid (far wake). Compared to top-hat approaches such as the Jensen model [20], the Gaussian profile provides a smoother and more realistic description of wake recovery while remaining computationally efficient. For this reason, it has been widely adopted in recent layout optimization studies [4, 32, 40, 43].

The normalized deficit at downstream offset $(\Delta x, \Delta y)$ is given by

$$\frac{\Delta V}{V_\infty} = \left(1 - \sqrt{1 - \frac{C_T}{8(\sigma_y/D)^2}}\right) \exp\left(-\frac{1}{2}\left(\frac{\Delta y}{\sigma_y}\right)^2\right) \tag{3.4}$$

with $\sigma_y = k_y\,\Delta x + D/\sqrt{8}$, where C_T is the thrust coefficient, D the rotor diameter, and k_y the wake-expansion coefficient. In the present work, $k_y = 0.0324555$ is used for a turbulence intensity of 0.075, in line with values adopted in the aforementioned studies.

Wakes as superimposed using the root-sum-square method [22], which agrees better with experiments than linear addition, particularly when several wakes overlap [13]. The analysis assumes turbines operate aligned with the incoming wind, so wake deflection is neglected to keep the model simple. Finally, to reduce computational time, the inflow velocity at each turbine is evaluated at the hub center only, as in Thomas and Ning [42].

3.1.4 Constraints

As introduced in Eq. (3.1), two constraints define the feasibility of each layout. The first is the *selection constraint*, requiring exactly k turbines to be chosen. This guarantees that $\mathbf{b}$ contains exactly k entries equal to 1:

$$\sum_{i=1}^{N} b_i = k \tag{3.5}$$

The second is the *minimum spacing constraint*, which prevents turbines from being placed too close together, to limit wake effects and structural fatigue. If S_{ij} is the distance between positions i and j, and $S_{\min}$ the required minimum distance, then any two selected turbines must satisfy:

$$S_{ij} \geq S_{\min} \quad \forall\, i \neq j \tag{3.6}$$

3.2 Software & Hardware

All simulations were performed using Python 3.11.4, primarily with the open-source libraries `PyWake` 2.6.11 [29] and `TOPFARM` 2.5.1 [39], both developed and maintained by DTU Wind Energy. `PyWake` provides a collection of validated wake models and utility functions for wind farm flow simulations, while `TOPFARM` offers a modular framework for coupling such models with optimization strategies. These tools were selected due to their modularity, reproducibility, and broad adoption in both academic and industrial studies, which ensures that the results are transparent and comparable to existing literature.

The computational experiments were carried out on a laptop equipped with an Intel® i7-13850HX processor (20 cores, 28 threads) and 32 GB of RAM. For RQ1, all optimizations were executed on a single CPU core to ensure fair comparison between strategies. In contrast, simulations for RQ2 leveraged parallel processing using 16 cores to reduce computation time due to increased problem complexity. No GPU acceleration was used.

CHAPTER 4

Strategies for Down-Selection (RQ1)

This chapter addresses the first research question:

RQ1. *Which optimization strategies are suitable for down-selection?*

1. Which strategies are most suitable for down-selection, based on defined requirements?
2. How do the top-ranked strategies compare across cases of increasing complexity?

The chapter follows the three main steps of Fig. 3.1, expanded into five sections as shown in Fig. 4.1. First, the two most suitable strategies are introduced in Section 4.1, namely a Gradient-Based approach and a Greedy Heuristic. Presenting these strategies in the beginning helps direct attention to the strategies most relevant to answering the research question, and avoids redundancy in the later sections. The ranking that motivated their selection from the pool of candidate strategies is discussed in Section 4.2. Each chosen strategy is then finetuned in Section 4.3, compared across test cases in Section 4.4, and the overall findings are discussed in Section 4.5.

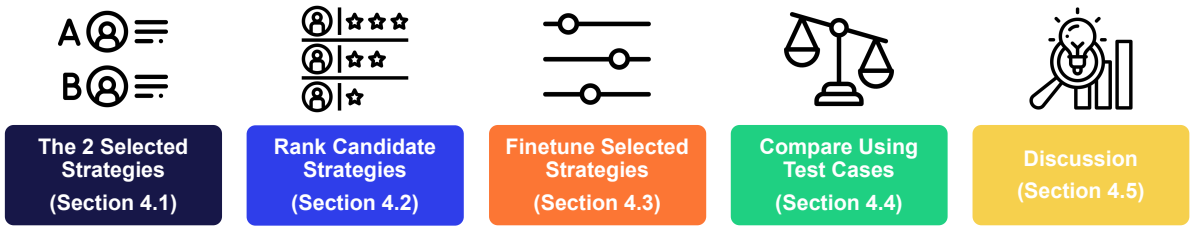


Figure 4.1: Roadmap for Chapter 4.

For the analysis of RQ1, the minimum turbine spacing constraint is assumed to be satisfied by the predefined available positions. This simplification removes the need to enforce explicit spacing constraints during optimization, thereby reducing computational cost and allowing the focus to remain on the comparative performance of the algorithms.

4.1 Description of Selected Strategies

This section presents the two optimization strategies that form the focus of this chapter, a *Gradient-Based* approach and a *Greedy Heuristic*. Their working principles and implementation details are provided in Section 4.1.1 and Section 4.1.2, serving as background for the subsequent ranking, finetuning, and comparative evaluation.

4.1.1 Gradient-Based Approach

As described in Section 2.2.2.1, the Gradient-Based (GB) approach by Pollini [32] relaxes the problem by using continuous variables. It then uses gradient-based solvers to adjust them, while using interpolation penalties to push solutions toward clear 0 or 1 selections.

Therefore, the optimization formulation described earlier in Eq. (3.1) must be relaxed with the use of a continuous variable $x_i \in [0, 1]$. The variable represents a fictitious existence of a wind turbine at an available position and is referred to as *"existence"* in the present work. Since all candidate positions already satisfy the spacing requirement (as mentioned above), the minimum spacing constraint does not appear here. The optimization problem then becomes

$$\begin{aligned} \text{maximize} \quad & \text{AEP}(\widetilde{\mathbf{x}}) \\ \text{subject to} \quad & \sum_{i=1}^{N} x_i = k \\ & 0 \leq x_i \leq 1 \qquad i = 1, \ldots, N \end{aligned} \tag{4.1}$$

where $\widetilde{\mathbf{x}} = [\widetilde{x}_1, \ldots, \widetilde{x}_N]$ is the penalized existence vector, and $\widetilde{x}_i$ is the penalized existence

$$\widetilde{x} = \frac{x}{1 + q(1 - x)} \tag{4.2}$$

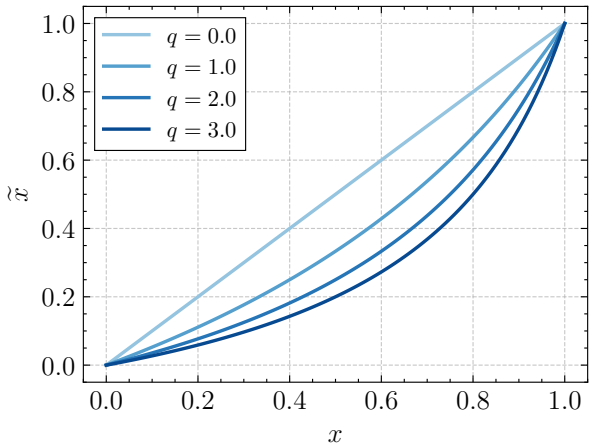


Figure 4.2: Example of interpolation with the RAMP functions for different values of the penalty parameters q.

The existence is penalized using the RAMP interpolation function, shown in Fig. 4.2. The interpolation leaves the endpoints unchanged ($\widetilde{x} = 0$ for $x = 0$ and $\widetilde{x} = 1$ for $x = 1$), while reducing the effectiveness of intermediate values ($\widetilde{x} < x$ for $0 < x < 1$). As a result, a turbine contributes less to the objective than what is counted in the constraint whenever x is fractional. This mismatch penalizes intermediate solutions and biases the optimization toward discrete 0–1 values. Increasing the penalty parameter q strengthens this effect.

In the original work, wind speed deficits were precomputed to accelerate simulations. This was feasible because the turbine used features a constant C_T, so deficits scale with the free-stream velocity and do not require evaluation at the wake-reduced inflow. In contrast, a turbine with a realistic C_T curve would couple the thrust coefficient to the local wind speed (see Eq. (3.4)), making precomputation impractical due to the large number of operating points required.

In the present work, the applicability of the method to turbines with realistic C_T curves is explored. Wake deficits are therefore evaluated directly rather than precomputed. Furthermore, the introduction of the existence variable x requires turbine characteristics to be expressed as functions of both wind speed and existence. This is achieved by scaling the original power and C_T curves with x, yielding `multidimensional curves` that interpolate smoothly between zero output ($x = 0$) and the original turbine characteristics ($x = 1$). For example, the IEA-37 3.35 MW power and C_T curves in Fig. 4.5 are transformed into the interpolated curves shown in Fig. 4.3. This functionality is implemented using PyWake's `PowerCtNDTabular` class.

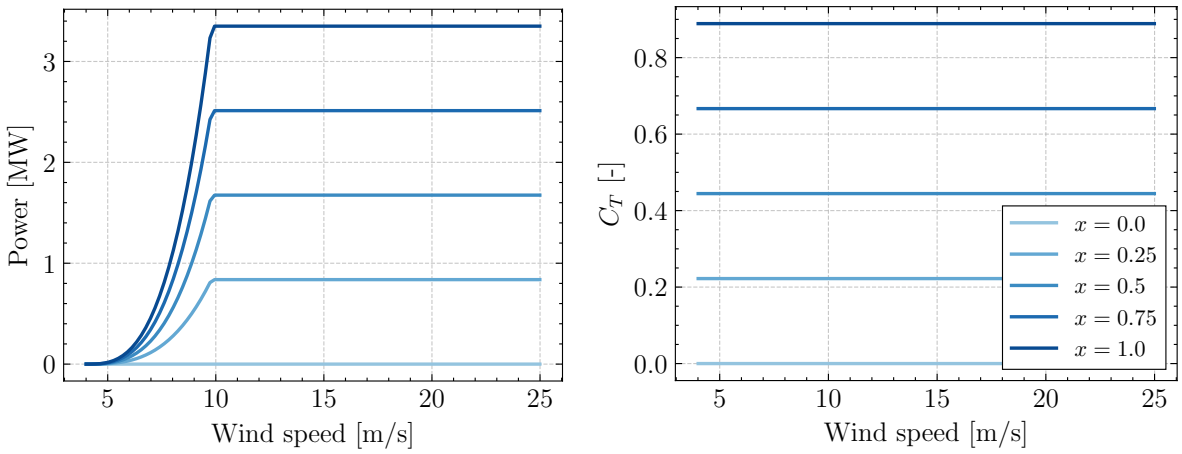


Figure 4.3: Power and C_T curves for sample existence values of the `ExistenceWindTurbine`, based on the IEA-37 3.35 MW reference turbine.

For initialization, the existence variables are assigned random values between 0 and 1 such that their sum equals the target number of turbines k. This ensures that the design space is explored fairly while satisfying the selection constraint.

The optimization is carried out using the gradient-based Sequential Least Squares Programming (SLSQP) algorithm. This solver is chosen for its robustness in handling constrained nonlinear problems, its open-source implementation, and its seamless integration with the TOPFARM framework. Unless otherwise specified, a convergence tolerance of 10^{-8} and a maximum of 500 iterations are used.

Upon convergence, the optimized existence variables typically take intermediate values rather than crisp 0–1. A post-processing step is therefore introduced: the k positions with the highest existence values are set to 1, and the remainder are set to 0. The AEP computed at the end of the optimization is denoted AEP_{drv}, while the final value after post-processing is referred to as AEP.

Since gradient-based methods are highly sensitive to initialization, a *multistart* strategy is adopted. Specifically, the optimization is repeated m times from independent random initializations, and the solution yielding the highest AEP is retained. Unless otherwise stated, $m = 100$ multistarts are used.

4.1.2 Greedy Heuristic

As described in Section 2.2.2.6, the Greedy Heuristic (GH) approach by Chen et al. [12] consists of two stages. In *Stage 1*, turbines are placed iteratively, one at a time, onto an initially empty layout. At each iteration, the next turbine is positioned at the location that maximizes the AEP, given the turbines already placed. In *Stage 2*, a refinement process is performed in which each turbine is revisited following the placement order and relocated to a different empty position that further increases the total AEP. This refinement is repeated until no further improvement is achievable.

In contrast to the Gradient-Based approach, this strategy is based directly on the original formulation in Eq. (3.1). As mentioned before, the minimum spacing constraint does not appear here, because all available positions are predefined to meet the requirement.

To ensure consistency with the Gradient-Based approach, all strategies use the same custom `ExistenceWindTurbine` incorporating multidimensional power and C_T curves (see Fig. 4.3). However, in the Greedy Heuristic only the two extreme states are evaluated ($b_i = 1$ or $b_i = 0$).

The outcome of the Greedy Heuristic is highly sensitive to the position selected for the first turbine placement (or removal). This sensitivity to initial condition can significantly affect the resulting layout and total AEP. To address this, a *multistart* strategy is adopted, involving multiple independent runs starting from a different available position. The layout yielding the highest AEP among all runs is selected as the final solution. Unless otherwise stated, $m = N$ multistarts are used.

One limitation of the baseline refinement procedure is that turbines are revisited in the exact order they were added. Early-placed turbines may undergo many relocation attempts, while later-placed turbines receive fewer or none before convergence. Consequently, the algorithm may become trapped in a local optimum that reflects the arbitrary placement order rather than the true global optimum.

To investigate possible improvements to this baseline, several alternative strategies are introduced in Table 4.5. These include adjustments to refinement order, early stopping, and removal-based placement. Their effectiveness will be tested in the tuning experiments of Section 4.3.3.

4.2 Ranking of Candidate Strategies

This section presents the literature-based screening and ranking of candidate strategies that motivated the selection of the two strategies described in Section 4.1. The screening methodology is outlined in Section 4.2.1, followed by a definition of the ranking criteria in Section 4.2.2, an overview of the considered strategies in Section 4.2.3, and the resulting ranking with discussion in Section 4.2.4.

4.2.1 Screening Methodology

The screening is carried out through a literature-based review of optimization strategies that could be applicable to the down-selection problem. A wide range of papers and textbooks are consulted to identify candidate approaches, since no comprehensive comparison of methods exists in the literature. Conducting such a comparison would be outside the scope of this thesis, both in terms of time and resources.

To enable a systematic assessment, a set of criteria considered important for down-selection is defined (see Section 4.2.2). Each candidate strategy is then scored qualitatively against these criteria. The scores are summed to obtain an overall ranking, from which the most promising strategies are identified for further analysis.

To keep the assessment simple and focused, hybrid strategies that combine multiple methods are not considered. Only stand-alone strategies are included, ensuring that the results reflect the inherent characteristics of each approach.

4.2.2 Ranking Criteria

The ranking is based on a set of requirements that reflect the characteristics of the down-selection problem. The aim is to identify strategies that are well suited for handling the specific characteristics of this task while remaining practical to apply in research and industry settings. The criteria are listed in Section 4.2.2.1, followed by the scoring system in Section 4.2.2.2.

4.2.2.1 Criteria

The following criteria are used to evaluate candidate strategies:

1. **Suitability for predefined available positions** The down-selection problem assumes a predefined set of available positions. Effective strategies must operate directly on this fixed choice set.
2. **Scalability to large combinatorial problems.** The number of possible layouts increases combinatorially with the number of available positions (see Eq. (2.1)). An effective strategy must therefore be able to handle problem sizes representative of offshore wind farms.
3. **Suitability for non-linear objectives.** Wake effects introduce non-linearities in the objective function. Strategies must be able to cope with such interactions without relying on restrictive assumptions.

4. **Number of tuning parameters.** Strategies with few tuning parameters are preferred, as this reduces the effort required for calibration and limits sensitivity to arbitrary choices.
5. **Ease of implementation.** Preference is given to methods that are straightforward to implement, as this reduces development effort within the time frame of this thesis and enables practical application.

4.2.2.2 Scoring System

Each candidate strategy is evaluated primarily qualitatively, and in some cases quantitatively, against the criteria in Section 4.2.2.1 using a four-level scoring scheme:

- **3 = Favourable:** strong suitability for down-selection.
- **2 = Moderate:** partial suitability or weaker performance than other strategies.
- **1 = Unfavourable:** poor suitability or clearly inferior performance.
- **0 = Insufficient evidence:** no clear information available.

No weights are assigned to the criteria, so all aspects are treated equally in the assessment. The individual scores are summed to produce an overall ranking, from which the most promising strategies are identified for further analysis.

4.2.3 Candidate Strategies

Following an extensive literature review, ten strategies were identified as the most promising for the down-selection problem and are carried forward to the ranking. Table 4.1 gives a concise overview of each strategy, including its category, a main strength and weakness, and representative references. More details about the algorithms can be found in Section 2.2.2.

Table 4.1: Overview of candidate strategies considered for ranking. Acronyms defined in Section 2.2.2.

Strategy	Category	Strength	Weakness	References
GB	Gradient-Based	Fast	Local minima	[32, 31]
GA	Metaheuristic	Widely used	Poor scalability	[28, 40, 32]
BPSO	Metaheuristic	Swarm exploration	Tuning	[27, 24]
VNS	Metaheuristic	Neighborhood jumps	Tuning	[26, 10, 35]
RS	Heuristic	Simple	Myopic	[14]
GH	Heuristic	Simple	Myopic	[12, 41, 46]
RSW	Heuristic	Gradient mimicry	Integration effort	[41]
DEBO	Heuristic	Top in reference	Integration effort	[43]
DPA	Heuristic	Top in reference	Integration effort	[43]
MILP	Exact	Provable optimality	Linear objective	[44, 25, 34]

4.2.4 Results and Discussion of Ranking

This section presents the outcomes of the ranking. Section 4.2.4.1 reports the aggregated scores and final ordering of the ten candidate strategies from Table 4.2. Section 4.2.4.2 interprets

the results with reference to the scoring criteria and highlights key trade-offs and limitations. Finally, Section 4.2.4.3 identifies the strategies shortlisted for subsequent evaluation.

4.2.4.1 Ranking Results

The ranking of the candidate strategies for down-selection is presented in Table 4.2.

Table 4.2: Ranking of candidate strategies for down-selection.

Strategy	Predefined Positions	Scalability	Non-Linear Objective	Tuning Parameters	Easy Implementation	Final Score
GB	3	3	3	2	2	13
GA	3	2	3	1	3	12
BPSO	3	2	3	1	1	10
VNS	3	2	3	2	1	11
RS	3	2	3	3	1	12
GH	3	2	3	3	2	13
RSW	3	2	3	3	1	12
DEBO	0	0	3	0	1	4
DPA	0	0	3	0	1	4
MILP	3	2	1	2	1	9

4.2.4.2 Discussion of Results

DEBO and DPA, despite strong results in their original study, are not formulated for selection from a predefined set of available positions. Applying them to down-selection would require substantial reformulation, so there is no direct evidence for their performance on the first criterion. For the same reason, they are assigned (0) on that criterion and score poorly across most of the criteria, placing them at the bottom for down-selection suitability. Their ranking is not discussed further below.

In terms of scalability, GB scores (3) because the relaxation of the problem with the use of continuous design variables enables the use of gradients, and gradient-based methods scale more favourably with the number of design variables than gradient-free alternatives [2, 41]. Empirical results further indicate markedly lower runtimes for GB relative to GA on comparable wind-farm problems [32]. The remaining strategies are scored (2), as they can be engineered to handle large candidate sets, yet their evaluation cost typically grows more steeply with dimensionality or depends on problem-specific proxies and tuning, consistent with reports that gradient-free techniques degrade as variable counts increase [24, 40].

In terms of non-linear objectives, all strategies except MILP score (3), as they can operate directly on AEP with non-linear wake interactions. MILP typically relies on the linearization of the objective to remain tractable, while also pre-computing an interaction matrix of the velocity deficits. Using the full non-linear AEP objective would require a computationally costly MINLP formulation outside this scope. Consequently, MILP is assigned (1) due to its dependence on linearizing assumptions that reduce fidelity.

In terms of having few tuning parameters, GH, RS, and RSW score (3) because they rely on simple settings such as basic stopping rules, sweep order, or acceptance checks, which rarely need fine calibration. GB, VNS, and MILP score (2) since they involve a moderate number of choices, for example choice of solver, tolerance, neighborhood size or gap tolerance. GA and BPSO score (1) because performance depends on several coupled hyperparameters, including population size, selection pressure, mutation rates, inertia or social weights, which increases the effort needed for finetuning.

In terms of ease of implementation, GA scores (3) because a ready to use implementation exists in TOPFARM. GB scores (2) since it can be realized in TOPFARM through a continuous relaxation with moderate additional setup. GH also scores (2) because a straightforward Python implementation appears feasible. The remaining strategies score (1) as their development would be more challenging and require some experience.

Overall, GB and GH come out on top with (13) points, followed by GA, RS and RSW with (12). Then come VNS, BPSO and MILP with (11), (10) and (9) points respectively, and in the last place with (4) points are DEBO and DPA.

Additional numerical results that support this ranking are provided in Appendix A.1.

4.2.4.3 Shortlisted Strategies

The criteria based ranking process revealed that two strategies are more suitable for the down-selection problem:

- **Gradient-Based (GB).** Scores highly on scalability and non-linear objective handling, with moderate tuning and implementation effort using a continuous relaxation in TOPFARM.
- **Greedy Heuristic (GH).** Scores highly on tuning simplicity and non-linear objective handling, with moderate scalability and implementation effort.

These strategies will form the focus of this chapter. Their operating principles and implementation details are presented in Section 4.1.1 and Section 4.1.2, serving as the background for the subsequent finetuning in Section 4.3.

4.3 Finetuning of Selected Strategies

The purpose of this section is to determine suitable parameter settings for the selected strategies prior to their comparative evaluation in Section 4.4. The finetuning is carried out on a single benchmark case, which provides a consistent environment for testing different parameter choices. In the process, the analysis of the Greedy Heuristic gives rise to an additional variant, so that three strategies are ultimately carried forward.

The benchmark setup is described in Section 4.3.1, followed by the tuning of the Gradient-Based approach in Section 4.3.2 and the tuning of the Greedy Heuristic in Section 4.3.3. The section concludes with a summary of the recommended settings in Section 4.3.4.

4.3.1 Benchmark Setup for Tuning

The benchmark case is based on the reference offshore wind site from the IEA's Wind Task 37 (IEA-37), as described by Baker et al. [4] with openly available data in [5]. This site features a circular domain populated with 64 predefined turbine positions, as shown in Fig. 4.4.

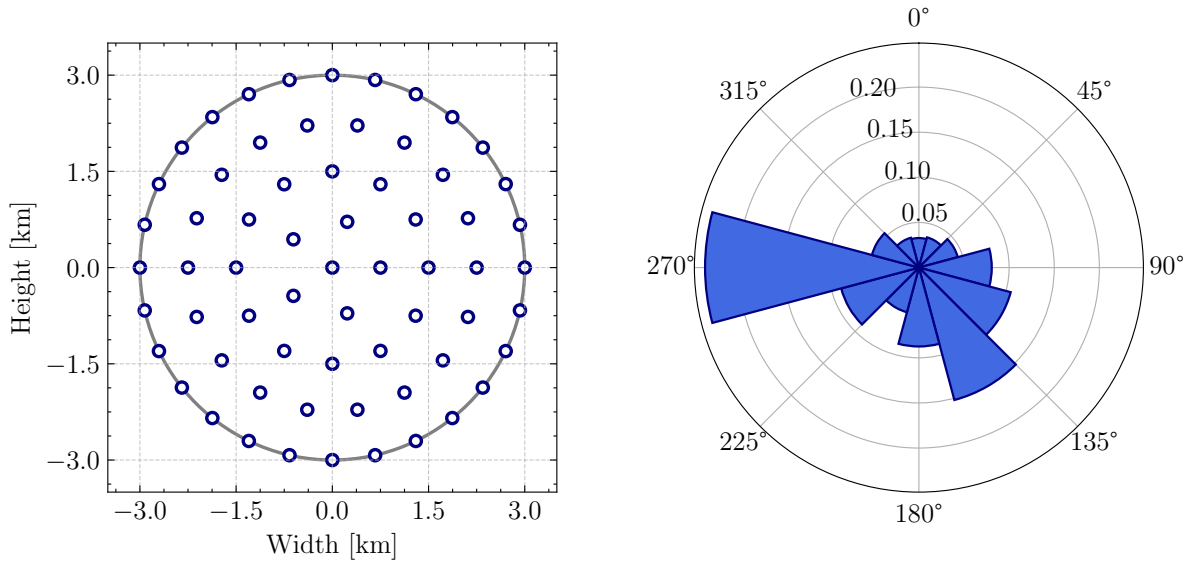


Figure 4.4: The 64 available turbine positions in the IEA-37 (left), and the wind rose corresponding to a uniform wind speed of 9.8 m/s (right).

The benchmark problem is formally defined as selecting 32 turbine locations out of the 64 predefined positions. This setting is both practically relevant and computationally challenging. Occupying 50% of the positions maximizes the number of possible combinations in a binary combinatorial problem (see Eq. (2.1)). Specifically, the number of distinct layouts is:

$$C(64, 32) \approx 1.8 \times 10^{18}$$

This high number of combinations ensures a low probability that all strategies converge to the global optimum, thereby providing a robust test environment.

All simulations use the idealized IEA-3.35 MW reference wind turbine described by Bortolotti et al. [9], for which this site was specifically designed. The turbine has a rotor diameter of $D = 130$ m. The wind speed of 9.8 m/s matches the rated speed of the turbine, increasing power variability across positions and leading to more local optima, which makes the problem more challenging for optimization. The power and thrust coefficient (C_T) curves are shown in Fig. 4.5.

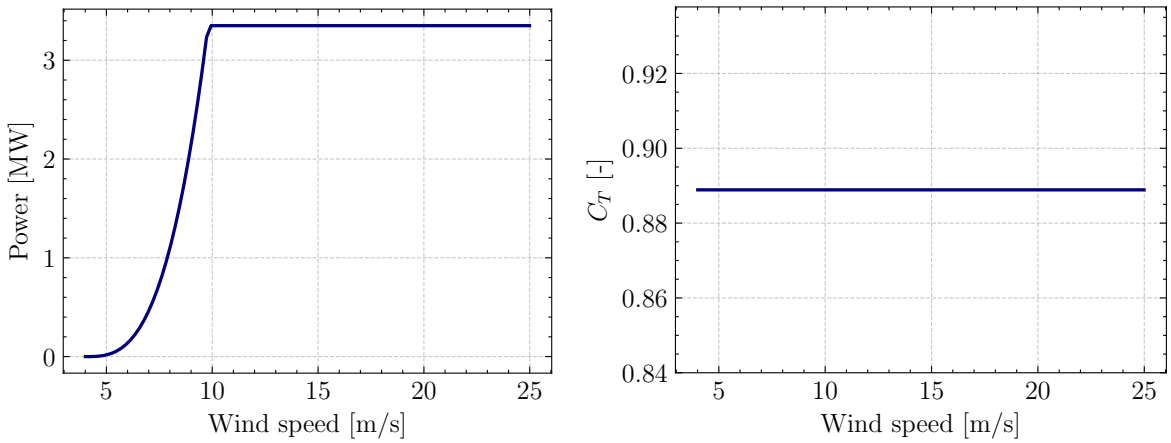


Figure 4.5: Power (left) and C_T (right) curves of the IEA-3.35 MW reference turbine.

The IEA-3.35 MW turbine features a constant C_T, so wake deficits scale with the free-stream velocity rather than the local wake-reduced inflow. This decouples C_T from the effective wind speed and enables non-iterative, single-pass wake evaluation. Thus, enabling the pre-computation of wake effects for given wind conditions, yielding substantial computational savings. The choice is a modeling convenience for layout-optimization benchmarks, not a reflection of actual turbine aerodynamics. However, in the present work, wakes are not pre-computed. Also, the selected wake model does not require a constant C_T, as realistic C_T curves have been used in the original formulation by Thomas and Ning [42].

An additional reason for selecting this site and turbine is that they were also employed by Pollini [32] in the development of the Gradient-Based approach. This ensures consistency with earlier work and could enable comparison of results in the future.

For the analysis of RQ1, the minimum turbine spacing constraint is assumed to be satisfied by the predefined positions of the IEA-37 site. The actual minimum pairwise spacing is $5.17D$, which is sufficient for the validity of the wake model. This simplification removes the need to enforce explicit spacing constraints during optimization, thereby reducing computational cost and allowing the focus to remain on the comparative performance of the algorithms.

4.3.2 Tuning of Gradient-Based

Here the Gradient-Based approach is tuned using the benchmark case of placing $k = 32$ turbines in the $N = 64$ available positions of the IEA-37 farm, as described in Section 4.3.1. Although a broader range of parameters was explored, these are documented in Appendix A.2 to avoid distracting from the main discussion. The most relevant findings are presented below.

4.3.2.1 RAMP vs SmoothStep

First, the choice of the interpolation function that can replace RAMP (Eq. (4.2)) is explored. After a lot of trial and error, the SmoothStep function is found to perform adequately:

$$\widetilde{x} = \frac{x^s}{x^s + (1-x)^s} \tag{4.3}$$

The function is controlled by the penalty parameter s, and it is a continuous monotonic function that passes through (0,0), (0.5,0.5), and (1,1), as can be seen from Fig. 4.6.

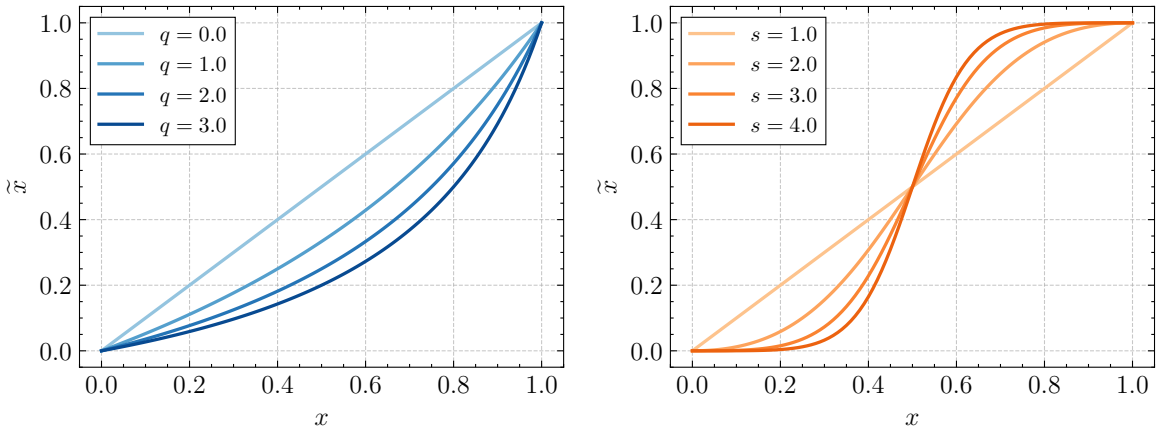


Figure 4.6: Example of interpolation with the RAMP (left) and the SmoothStep (right) functions for different values of the penalty parameters q and s respectively.

To draw meaningful conclusions about the behavior of each function, a sufficiently large sample size is required. Therefore, 100 different random seeds are used for the initial existence values. Then, 21 different values of q and s are tested while reusing the same 100 seeds. This results in a total of 2100 optimization runs for RAMP and another 2100 for SmoothStep. For RAMP, q values ranging from 0 to 1 are used, while for SmoothStep from 1 to 2. In both cases, the values start just after the linear region and increase in equal steps. The corresponding results are presented in Fig. 4.7.

It is immediately apparent that for both interpolation functions the min, median and max AEP decrease as their respective penalty factors increase. Also, at an initial reading it might seem that the SmoothStep performs better as it maintains a higher AEP statistics with smaller standard deviation. However, this would comparison would be unfair, as its function might be less or more aggressive for the same increase in the penalty parameter.

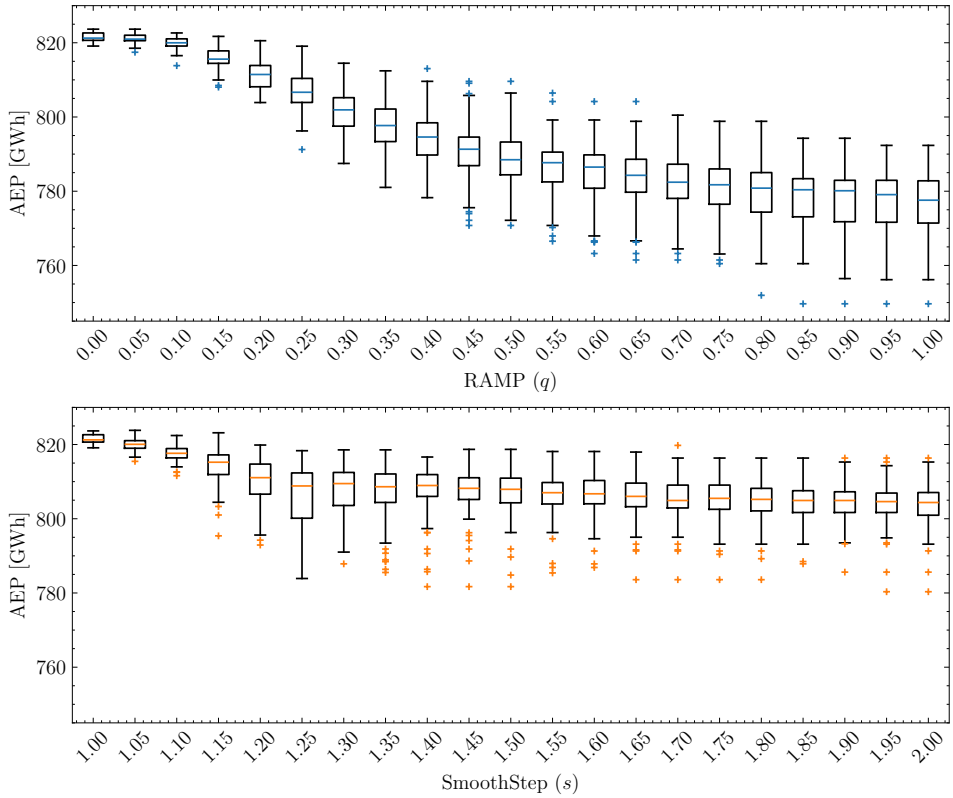


Figure 4.7: Boxplot of AEP using RAMP (top) and SmoothStep (bottom).

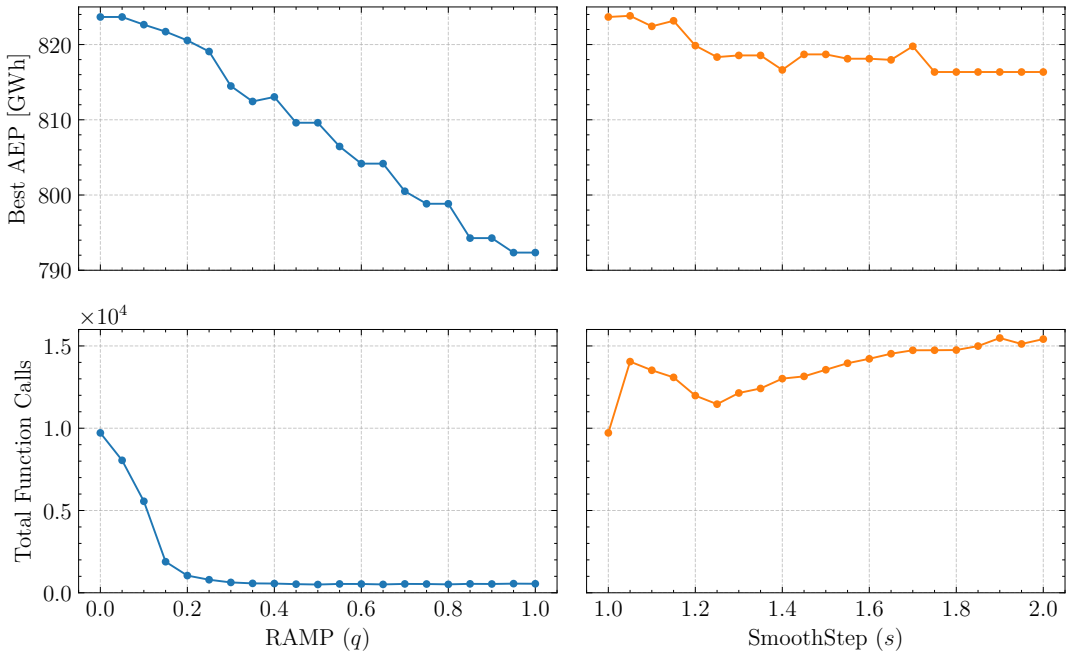


Figure 4.8: Best AEP (top row) and corresponding total function calls (bottom row), using RAMP (left column) and SmoothStep (right column).

Selecting the best out of the 100 runs for each penalty parameter value, leads to the multistart approach mentioned before. The best AEP and the corresponding total function calls for each penalty function are presented in Fig. 4.8. SmoothStep maintains a higher best AEP than RAMP as the penalty parameters increase. However, the total function calls of RAMP drop drastically and stay low, while for SmoothStep they increase steadily, apart from a sudden increase in the beginning.

As mentioned before, after each optimization, the $k = 32$ positions with the highest existence values are selected and assigned a final existence value of 1, with the remaining positions set to 0. This binary selection defines the final layout used to calculate the AEP. Fig. 4.9 depicts the existence of each turbine just before this top-32 selection.

It appears that as q increases, RAMP pushes the turbines closer to an existence of either 0 or 1. On the other hand, SmoothStep leaves a lot of turbines with intermediate values. Therefore, RAMP does a better job at splitting the turbines to ones that exist and ones that do not.

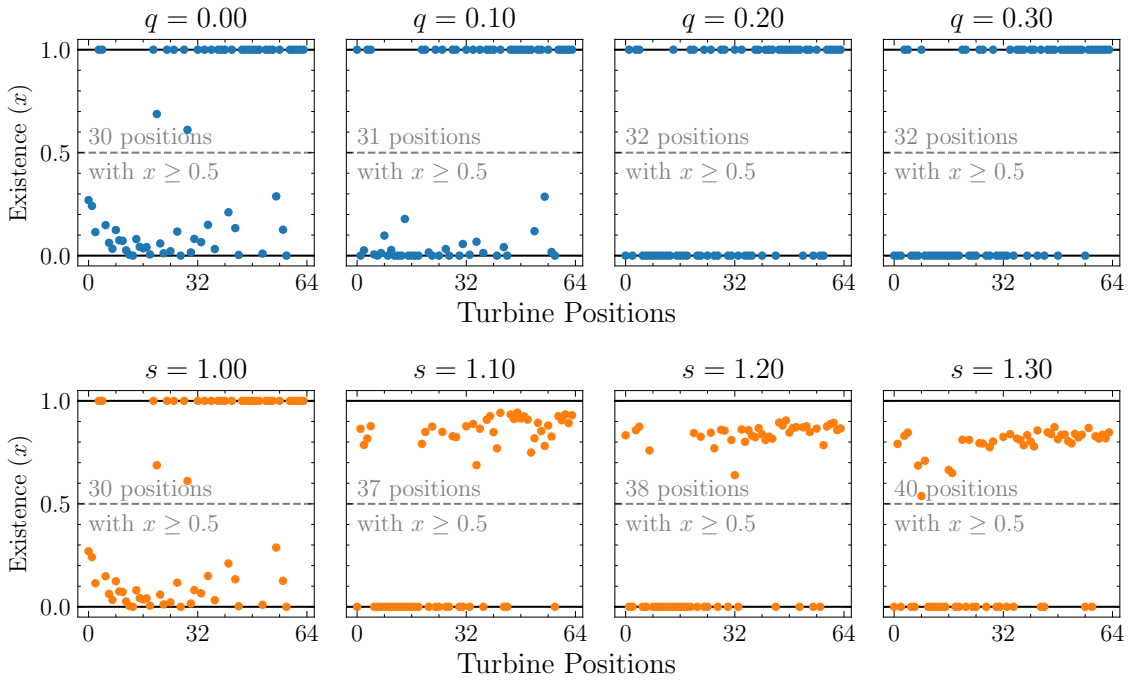


Figure 4.9: Existence of each of the 64 turbines at the end of the optimization using RAMP (top) and SmoothStep (bottom) for different q and s values. The optimization that yields the highest AEP is used for each q and s. Turbine numbering is not relevant but is consistent.

Despite all these observations, Fig. 4.8 suggests that the highest AEP is achieved for $q = 0$ (or $s = 1$), or when no interpolation is happening (*Linear* interpolation). Despite producing less crisp 0-1 values in Fig. 4.9, this approach is selected to be used in the later tuning tests due to its superior AEP performance.

4.3.2.2 Impact of Driver Tolerance

Relaxing the `SLSQP` driver tolerance from 10^{-8} could substantially reduce the number of function calls needed for convergence, while yielding nearly identical AEP results. To test this hypothesis, 100 penalty-free (Linear) optimizations are run using the same set of 100 random seeds for each driver tolerance, which range from 10^{-8} to 10^{-2}. By keeping the seeds constant, any performance differences are attributable solely to the tolerance setting. Fig. 4.10 summarizes the distributions of AEP and function calls for each tolerance value. The AEP boxplots remain virtually identical across tolerances. In contrast, the minimum, median, maximum, and standard deviation of the function calls all drop as the tolerance is relaxed.

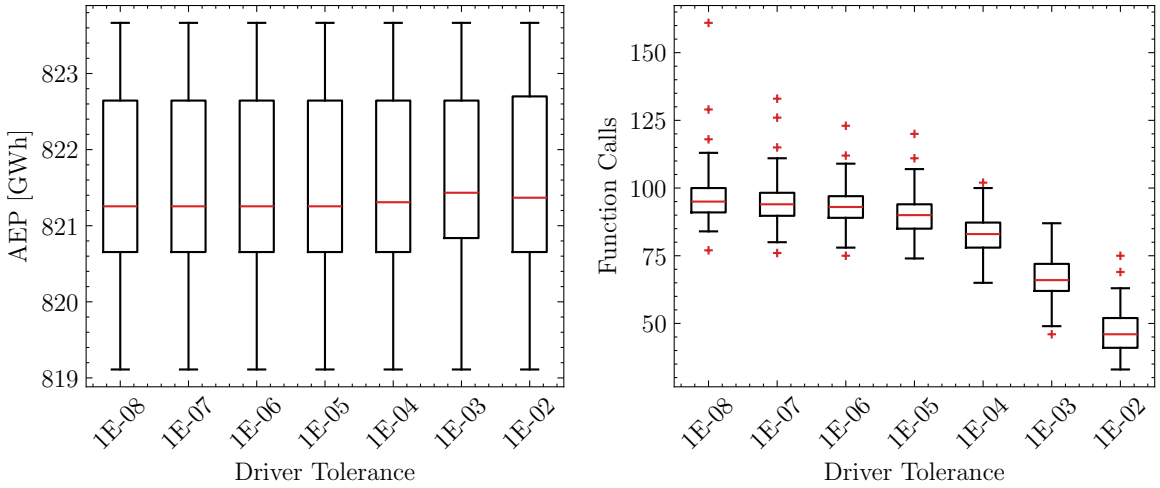


Figure 4.10: Boxplots of AEP obtained after selecting top-k (left) and function calls (right), for various driver tolerances.

However, since a multistart strategy is followed, which ultimately relies on the best AEP achieved and the total computational effort, Table 4.3 compares, for each tolerance, the highest AEP found and the total number of function calls. The results show that loosening the tolerance does not compromise the maximum AEP but dramatically reduces the total number of function calls, demonstrating that a higher tolerance can significantly accelerate convergence without sacrificing optimality.

Table 4.3: Max AEP and function calls for top-k selection under different driver tolerances.

Driver Tolerance	AEP_{max} [GWh]	Total Func. Calls
1E-08	823.7	9,717
1E-07	823.7	9,509
1E-06	823.7	9,328
1E-05	823.7	9,040
1E-04	823.7	8,302
1E-03	823.7	6,668
1E-02	823.7	4,696

Additionally, the seed that yielded the highest AEP under the strictest tolerance, is examined across all tolerance settings. The progress of the driver's AEP against the function calls is shown in Fig. 4.11. As expected, looser tolerances stop the optimizer sooner, but at 10^{-2} the driver stops before the AEP curve has fully flattened. Therefore, 10^{-3} is chosen as the new default tolerance, as it can produce the exact same maximum AEP, while requiring around half the number of total function calls.

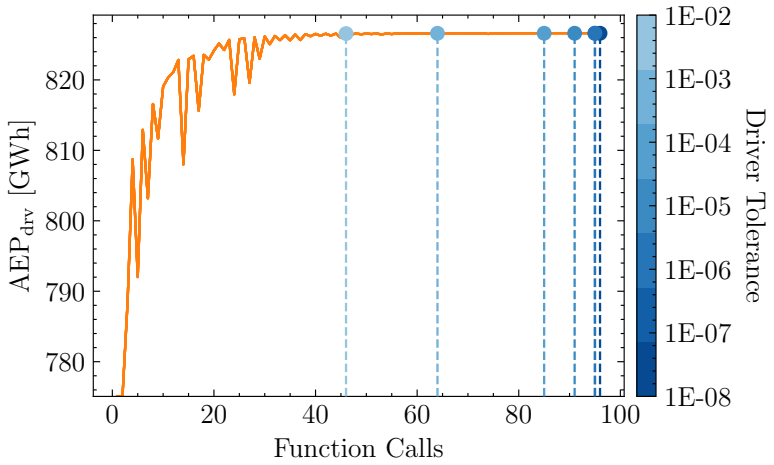


Figure 4.11: Progress of driver's AEP for various driver tolerances but the same seed.

4.3.2.3 Impact of Number of Multistarts

Thus far, every optimization has used $m = 100$ multistarts to maximize the chance of finding the true global optimum AEP. However, fewer starts might achieve equivalent AEP with fewer function calls, while more starts could further explore the design space and possibly find an even higher AEP.

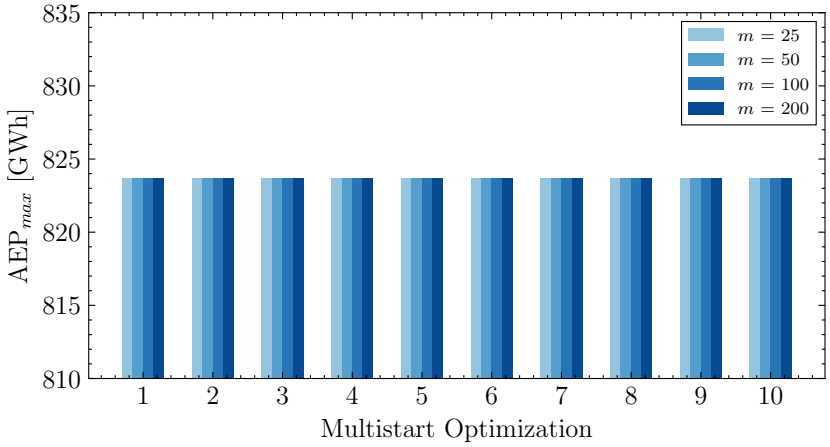


Figure 4.12: Best AEP values across 10 independent optimizations with 25, 50, 100, and 200 multistarts using random initial existence values.

To investigate this trade-off, 10 independent optimizations are run for each of four multistart counts (25, 50, 100, and 200) using random initial guesses without fixed seeds to ensure broad sampling. As shown in Fig. 4.12, the number of multistarts has no observable impact on AEP, since all runs converge to the same value.

Table 4.4 summarizes the average maximum AEP and the average total number of function calls for each multistart setting. As expected, the average peak AEP remains constant while the function-call counts scale almost linearly with the number of starts. Although 25 multistarts can reach the peak AEP in the fewest function calls, 100 multistarts will be used for the subsequent sections to ensure a better exploration of the design space.

Table 4.4: Average maximum AEP and average total function calls over 10 optimizations, for varying numbers of multistarts.

Number of Multistarts	Average AEP_{max} [GWh]	Average Total Func. Calls
25	823.7	1,690
50	823.7	3,398
100	823.7	6,750
200	823.7	13,580

4.3.3 Tuning of Greedy Heuristic

As outlined in Section 4.1.2, the baseline Greedy Heuristic suffers from certain limitations, such as sensitivity to placement order and the risk of becoming trapped in local optima. To investigate potential improvements and accelerate convergence, several alternative strategies are defined and tested (see Table 4.5):

- *Refinement Order*: Revisit turbines by ascending individual AEP contribution, so that the least effective turbines are refined first.
- *Early Stopping*: Terminate the relocation search for a turbine as soon as any improving move is found.
- *Removal-Based Greedy*: Start from a full layout and iteratively remove the turbine whose deletion causes the smallest decrease in AEP, followed by the same refinement schemes.

Table 4.5: Description of the various Greedy Heuristic strategies explored.

Strategy	Placement	Refinement	Refinement Based on	Early Stopping
A	Add 1-by-1	✗	–	–
B	Add 1-by-1	✓	Placement order	✗
C	Add 1-by-1	✓	Ascending AEP	✗
D	Add 1-by-1	✓	Ascending AEP	✓
E	Remove 1-by-1	✗	–	–
F	Remove 1-by-1	✓	Ascending AEP	✗
G	Remove 1-by-1	✓	Ascending AEP	✓

The tuning study is performed using the benchmark case of placing $k = 32$ turbines among $N = 64$ available positions in the IEA-37 wind farm, as described in Section 4.3.1. First, all seven strategies are directly compared in Section 4.3.3.1. Then the influence of the starting locations is examined in Section 4.3.3.2, after which the optimal number of multistarts is investigated in Section 4.3.3.3

4.3.3.1 Direct Comparison of Strategies

As previously discussed, the performance of the Greedy Heuristic is highly sensitive to the position selected for the first turbine placement (or removal). To address this, a multistart approach is employed in which each of the 64 available positions is used for the turbine placement and the layout yielding the highest AEP is retained. The results of applying all seven strategies to the benchmark case are summarized in Fig. 4.13, which presents the distribution of AEP values obtained across all 64 runs. The best AEP values and the total function calss are reported in Fig. 4.14.

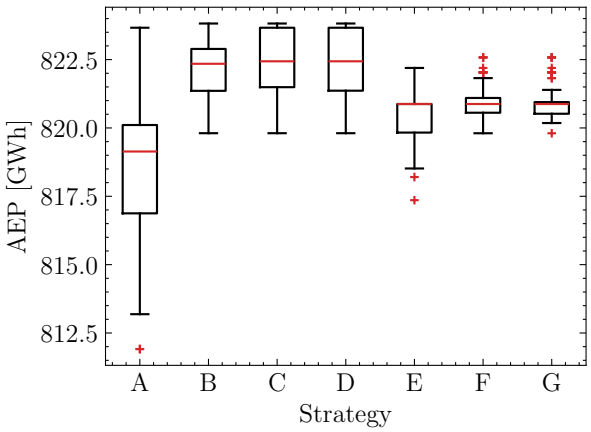


Figure 4.13: Boxplots of AEP for the various Greedy Heuristic strategies.

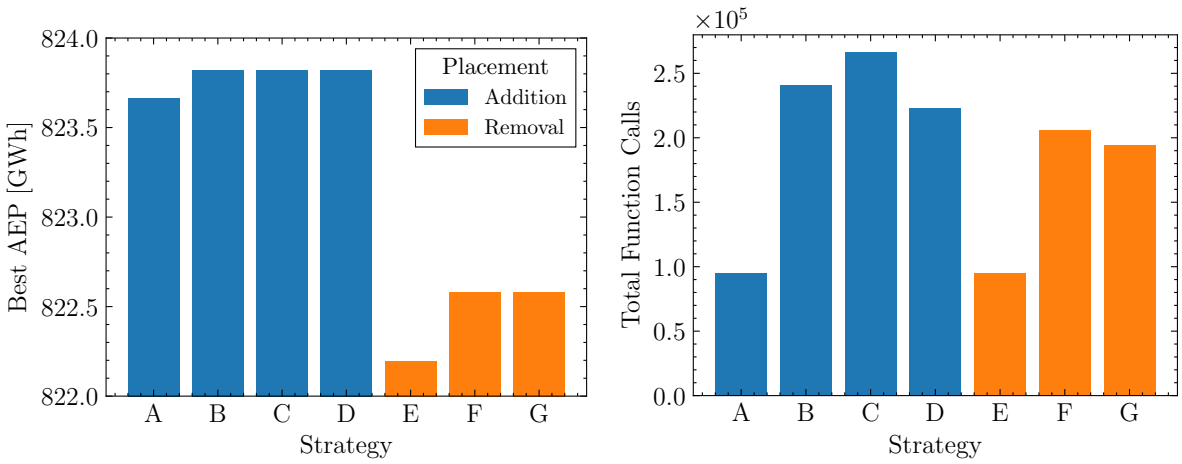


Figure 4.14: Bar chart of maximum AEP (left) and total function calls (right) for the various Greedy Heuristic strategies.

The results presented in Fig. 4.14 show that strategies A–D, which add turbines one-by-one to an initially empty layout, achieve higher AEP values compared to strategies E–G, which begin from a full layout and remove turbines sequentially. This outcome may be influenced by the target of $k = 32$ turbines out of $N = 64$ positions. It remains possible that for different values of k, the relative performance of removal-based strategies may differ.

As expected, all strategies that include a refinement stage (i.e. B–D and F–G) outperform their respective non-refined counterparts (A and E) in terms of AEP. Refinement enables repositioning of turbines after initial placement or removal, allowing the escape from local optima.

Among the addition-based strategies, B–D all achieve the same best AEP. but strategy D requires the least total function calls, followed by B and then C. This demonstrates the effectiveness of early stopping in reducing function calls without compromising on AEP. A similar trend is observed for the removal-based strategies F and G, which reach the same AEP, but strategy G requires fewer total function calls.

Even though strategies B–D achieve the same peak AEP, Fig. 4.13 shows that the AEP distributions of strategies C and D are shifted higher than that of B. This suggests that, under different conditions (e.g. number of multistarts), strategies that revisit turbines in ascending individual AEP contribution are more likely to reach a high-performing solution.

Based on these observations, strategies D and G are selected for further investigation. For simplicity in the following sections, they are referred to as shown in Table 4.6.

Strategy	Short Name
D	Greedy Addition
G	Greedy Removal

Table 4.6: Naming conventions for selected Greedy Heuristic strategies.

4.3.3.2 Influence of Starting Location

The previous results indicate that some starting locations for the Greedy Heuristic yield higher AEP than others. This observation raises the question of whether a spatial pattern exists that could guide the choice of starting points. To investigate this, Fig. 4.15 maps each of the 64 possible initial positions onto the site, colouring each one by the total AEP achieved when the Greedy Addition or Greedy Removal strategy is initialized at that point.

If a specific region of the layout consistently produced superior results, it would appear as a cluster of dark circles in one area. Instead, the darkest and lightest points are scattered throughout the entire farm. No single quadrant or ring of turbines shows a systematic advantage for either addition or removal. In other words, high-AEP and low-AEP starting points both occur in all directions and at all distances from the center.

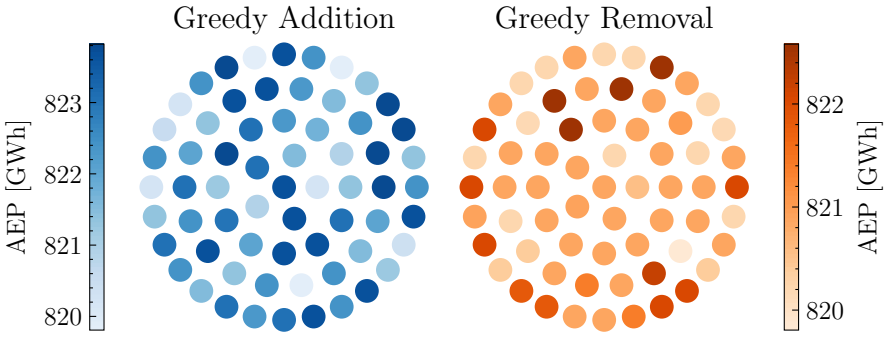


Figure 4.15: Starting points coloured according to total AEP found.

4.3.3.3 Number of Multistarts

Since there is no obvious region that consistently outperforms the rest, a resampling analysis is carried out to quantify how reducing the number of multistarts (m) impacts both the energy production and computational cost. For each value of m, 10,000 random subsets of m starting points are drawn from the full set of 64 starting points. The average ratio of AEP for m multistarts relative to all 64 is then calculated alongside the average ratio of total function calls for m multistarts versus all 64. The results are presented in Fig. 4.16.

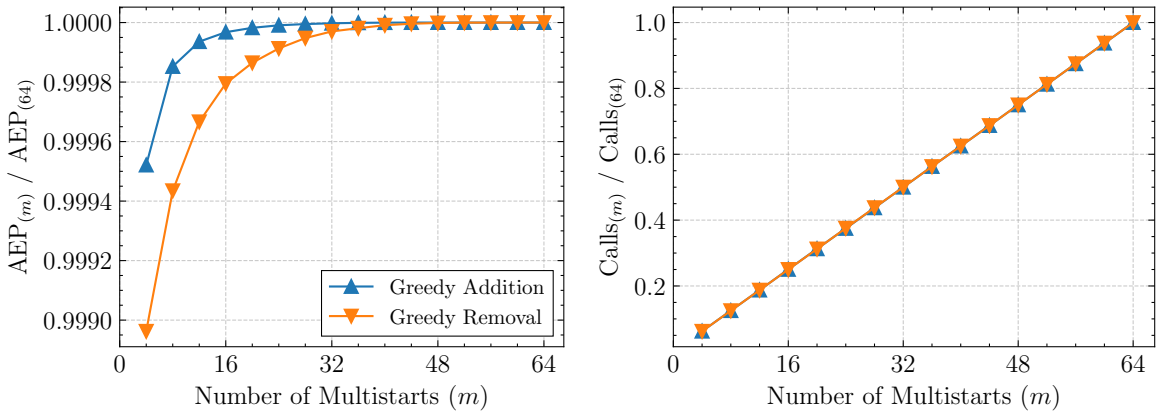


Figure 4.16: Average ratio of AEP (left) and total function calls (right) using m multistarts versus all 64. Derived from 10,000 random resamples for each m.

The results in Fig. 4.16 show that reducing the number of multistarts (m) leads to lower AEP compared to using all 64 starting points. This drop is more noticeable for Greedy Removal, where only a few starting positions give high AEP (see Fig. 4.15), making them easy to miss when m is small. In contrast, Greedy Addition has more good starting points, so its AEP recovers faster as m increases. The number of function calls increases almost linearly with m, showing a clear trade-off between computational cost and performance. Taken together, these trends show that $m \approx 32$ suffices to recover over 99.99% of the full AEP at about half the cost.

Despite this, all N multistarts are used in the rest of the study to ensure that the best starting points are always included, especially for the more sensitive Greedy Removal strategy.

4.3.4 Summary of Recommended Settings

Through the tuning process, three strategies emerge as the most promising candidates for down-selection, summarized in Table 4.7.

Table 4.7: Recommended strategies after tuning.

Name	Settings / Variant	Multistarts
Gradient-Based	no interpolation, tol=10^{-3}	100
Greedy Addition	Strategy D	N
Greedy Removal	Strategy G	N

For the Gradient-Based approach, Section 4.3.2 showed that no interpolation function is needed (RAMP can be replaced by Linear), and that lowering the driver tolerance to 10^{-3} achieves similar results with fewer function calls. With respect to multistarts, 25 runs were sufficient to reach the peak AEP in the fewest calls, but 100 multistarts are recommended to ensure more reliable exploration of the design space. The additional cost is acceptable because the number of function calls scales approximately linearly with the number of starts.

For the Greedy Heuristic, Section 4.3.3 identified the Addition (D) and Removal (G) variants as most effective. Since no spatial pattern was found to guide the choice of starting points, a multistart strategy is required. While using around 32 multistarts recovers over 99.99% of the maximum AEP at about half the cost, the number of function calls scales linearly with m, making the computational penalty acceptable. To ensure robust performance, especially for the more sensitive Removal variant, all N multistarts are retained in subsequent experiments. Functions calls also scale linearly, so the additional cost is deemed acceptable.

In summary, the Gradient-Based, Greedy Addition, and Greedy Removal strategies are carried forward with the tuned settings described above. Their relative performance is analyzed in detail in the following section (Section 4.4).

4.4 Comparison of Selected Strategies

This section compares the performance of the three strategies obtained in Section 4.3: *Gradient-Based*, *Greedy Addition*, and *Greedy Removal*. First, the methodology used to compare the strategies is described in Section 4.4.1. The benchmark case from Section 4.3.1 is then complemented by five additional test cases of increasing complexity, described in Section 4.4.2. Finally, the results are presented and discussed in Section 4.4.3.

4.4.1 Comparison Methodology

The comparison of the strategies is conducted on the benchmark and five additional test cases of increasing complexity introduced in Section 4.4.2. All strategies are evaluated under identical conditions, using the same turbine and wake model, and the recommended parameter settings identified in Section 4.3.4.

The primary performance metrics are the achieved AEP and computational cost, which is measured in terms of both the number of function evaluations and the wall-clock time. As mentioned earlier, a *multistart* approach is applied to all methods to reduce sensitivity to initialization, with the best-performing solution retained for each case.

This consistent protocol provides a fair basis for comparing solution quality and computational efficiency across the three strategies.

4.4.2 Test Case Setup

The benchmark introduced in Section 4.3.1 serves as the reference case for tuning and initial comparison. To assess the robustness of the strategies under different conditions, five additional test cases are defined as extensions of the benchmark. Each case introduces a variation in either the wind farm configuration or the turbine characteristics, thereby increasing the complexity of the optimization task towards more realistic scenarios.

To aid the reader, Table 4.8 provides an overview of the benchmark and the additional test cases, highlighting the key parameters that vary in each scenario. Test Cases A–E are described in Sections 4.4.2.1 to 4.4.2.5, respectively.

Table 4.8: Overview of benchmark and test cases used for evaluating optimization strategies.

Name	Effect of	Occupancy [%]	Available Positions	Variable C_T
Benchmark	$C(64, 32)$	50	64	✗
Test Case A	Occupancy	50 – 90	64	✗
Test Case B	Variable C_T	50 – 90	64	✓
Test Case C	Power Density	50 – 90	64	✓
Test Case D	Available Positions	50 – 90	36/64/94	✓
Test Case E	Scaled IEA-55 Layout	50 – 90	74	✓

4.4.2.1 Test Case A: Effect of Occupancy

Test Case A examines how the strategies perform under different levels of occupancy, defined as the ratio of the number of turbines to the number of available positions:

$$\text{Occupancy} = \frac{\#\text{ Wind Turbines}}{\#\text{ Available Positions}} \times 100 \tag{4.4}$$

Table 4.9: Occupancy levels and corresponding number of wind turbines for Test Case A.

Occupancy [%]	Wind Turbines
50	32
60	38
70	45
80	51
90	58

Occupancy levels between 50% and 90% are considered, with the closest corresponding integer number of turbines shown in Table 4.9. The lower bound of 50% corresponds to the benchmark case introduced in Section 4.3.1. The upper bound of 90% is motivated by ongoing industry trends toward larger offshore turbines. The largest commercially available models currently reach capacities of around 15 MW (e.g. Vestas V236-15.0 MW[1]), while prototypes under testing approach 21.5 MW (e.g. Siemens Gamesa SG 21-276 DD[2]). Such developments imply that future layouts may accommodate fewer but larger turbines, leading to higher occupancy levels in down-selection scenarios. The chosen range therefore reflects both the reference benchmark and realistic future design conditions.

4.4.2.2 Test Case B: Effect of Variable C_T

Test Case B extends Test Case A by introducing a modified turbine model with a variable thrust coefficient curve. This curve was generated using PyWake's `CubePowerSimpleCt` function and is shown in Fig. 4.17. Compared to the constant C_T used in the benchmark and Test Case A, this formulation provides a more realistic representation of turbine aerodynamics.

For consistency with the methodology described in previous sections, the corresponding multidimensional power and C_T curves are generated and used by all strategies. An example of these multidimensional curves is shown in Fig. A.3.

[1] `https://www.vestas.com/en/energy-solutions/offshore-wind-turbines/V236-15MW`

[2] `https://www.rinnovabili.net/business/energy/wind-turbine-sg-21-276-dd-hits-21-5-mw-in-denmark`

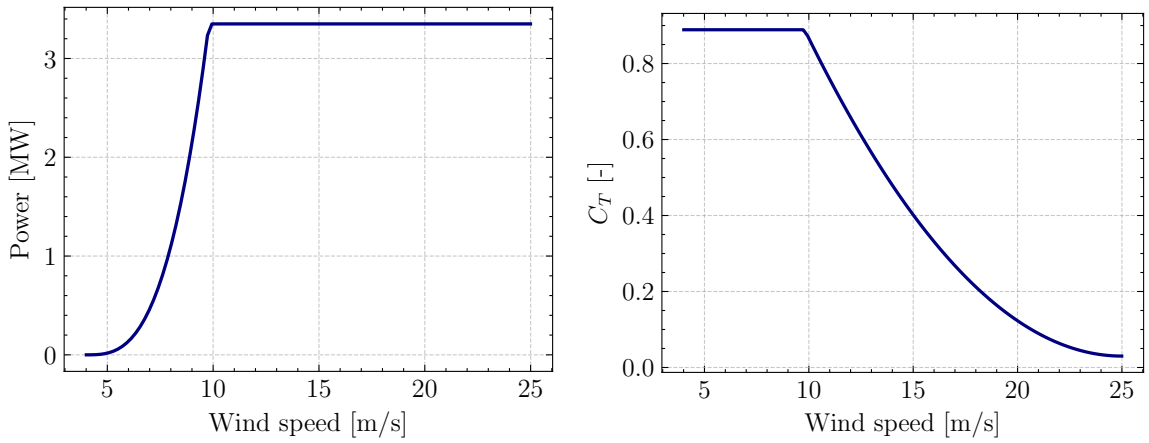


Figure 4.17: Power (left) and C_T (right) curves of the IEA-37 3.35 MW with variable C_T.

4.4.2.3 Test Case C: Effect of Power Density

Test Case C extends Test Case B by evaluating the impact of the power density of the farm, expressed in MW/km^2. Three configurations are considered, corresponding to a high-density (close-packed), original, and low-density (spaced-out) layout. The available positions for these configurations are shown in Fig. 4.18, with key parameters summarized in Table 4.10.

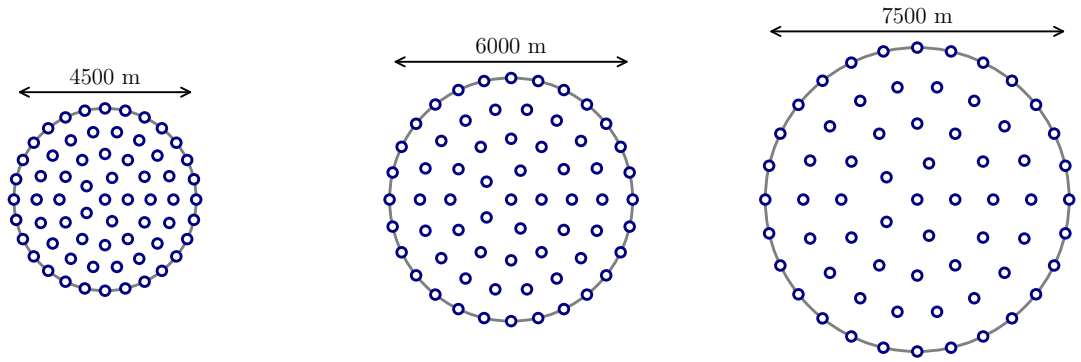


Figure 4.18: Available positions for the close-packed (left), original (middle), and spaced-out farms (right).

Table 4.10: Summary of farm parameters for Test Case C.

Description	Diameter [m]	Available Positions	Available Power Density [MW/km^2]	Minimum Distance Between WTs
Close-Packed	4500	64	13.48	$3.88D$
Original	6000	64	7.58	$5.17D$
Spaced-Out	7500	64	4.85	$6.46D$

4.4.2.4 Test Case D: Effect of Number of Available Positions

Test Case D extends Test Case B by varying the number of available turbine positions. Three layouts are considered, containing 36, 64, and 94 predefined positions, obtained by removing the outer ring, using the reference configuration, or adding an additional ring of turbines. For each layout, occupancy levels between 50% and 90% are tested, consistent with the approach in Test Case A. The layouts are shown in Fig. 4.19, with their parameters summarized in Table 4.11, and the corresponding number of wind turbines for each occupancy level is reported in Table 4.12.

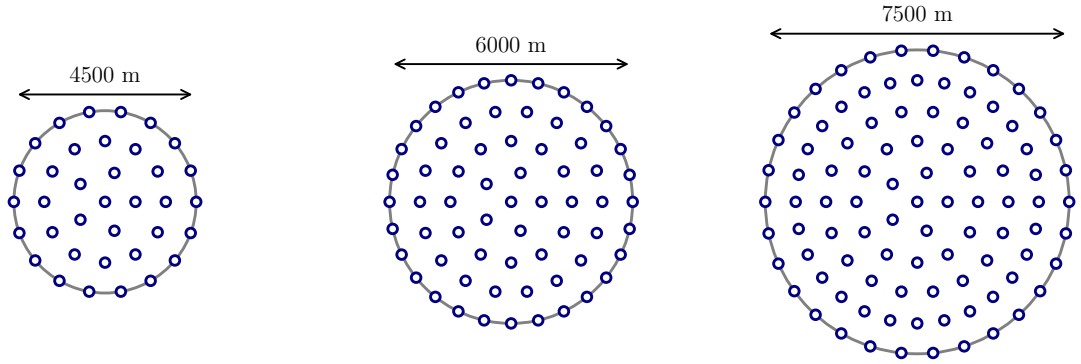


Figure 4.19: Available positions for compact (left), original (middle), and expanded farms (right).

Table 4.11: Summary of farm parameters for Test Case D.

Description	Diameter [m]	Available Positions	Available Power Density [MW/km^2]	Minimum Distance Between WTs
Compact	4500	36	7.58	$5.77D$
Original	6000	64	7.58	$5.17D$
Expanded	7500	94	7.13	$5.17D$

Table 4.12: Occupancy levels and corresponding number of wind turbines for Test Case D.

Occupancy [%]	Compact	Original	Expanded
50	18	32	47
60	22	38	56
70	25	45	66
80	29	51	75
90	32	58	85

4.4.2.5 Test Case E: Effect of Scaled IEA-55 Layout

The final Test Case E evaluates the strategies on a more realistic layout derived from the IEA-55 reference site of Kainz et al. [21]. This site features a regular grid arrangement of turbine positions, designed originally for larger turbines. To ensure comparability with the IEA-37 benchmark, the layout is uniformly scaled such that the minimum spacing between turbines matches that of the IEA-37 farm ($5.17D$). The original and scaled layouts are shown in Fig. 4.20.

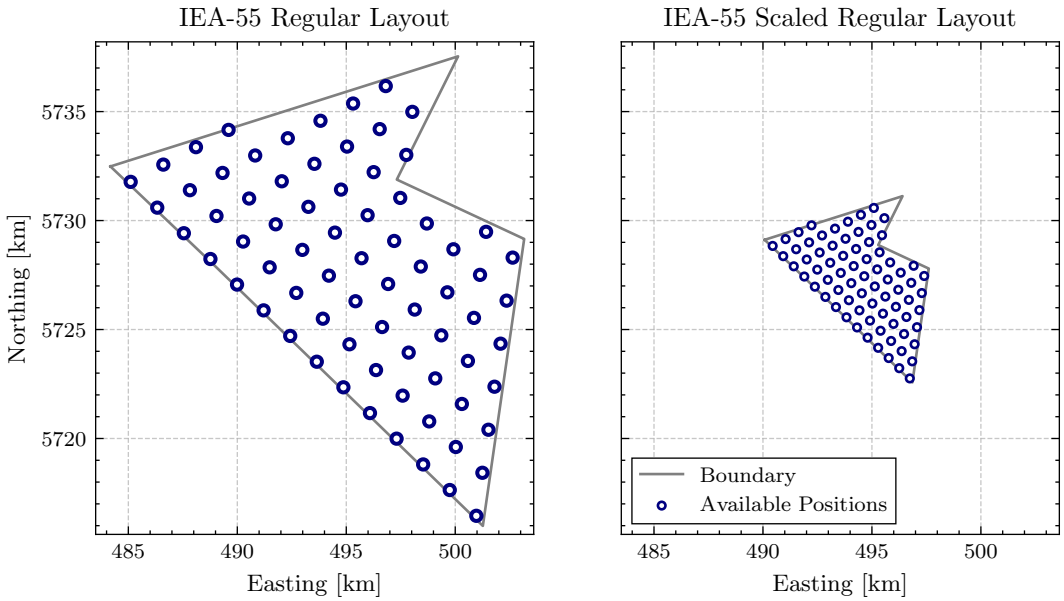


Figure 4.20: Available positions for the original IEA-55 layout (left) and the scaled version used in Test Case E (right). Turbine diameters not shown to scale.

Table 4.13: Occupancy levels and corresponding number of wind turbines for Test Case E.

Occupancy [%]	Wind Turbines
50	37
60	44
70	52
80	59
90	67

4.4.3 Results and Discussion

Based on the benchmark and the additional test cases introduced in Sections 4.3.1 and 4.4.2, the performance of the three strategies is now evaluated to highlight how they compare across increasingly complex scenarios. The results are presented and discussed in Sections 4.4.3.1 to 4.4.3.6.

4.4.3.1 Benchmark

Figure 4.21 compares the AEP, number of function calls, and computational time required by each optimization strategy at 50% occupancy. Greedy Addition achieves the highest AEP, followed closely by the Gradient-Based method, with Greedy Removal trailing behind. Although both greedy heuristics execute over an order of magnitude more PyWake AEP evaluations than the Gradient-Based approach, their actual times differ by only a factor of two. The Gradient-Based method is roughly 50% faster than Greedy Addition, while Greedy Removal takes about 50% longer.

The reason for the discrepancy between function calls and computational time is due to how the calls are counted, as only the PyWake evaluations for AEP are included, while other time-consuming operations are not. Additionally, although Greedy Addition requires more function calls than Greedy Removal, it takes less time overall. This is because, at the start of Greedy Addition, only a few turbines are in place, making the AEP evaluations faster. In contrast, Greedy Removal begins with a nearly full layout, so each PyWake call involves more complex wake interactions and takes longer to compute.

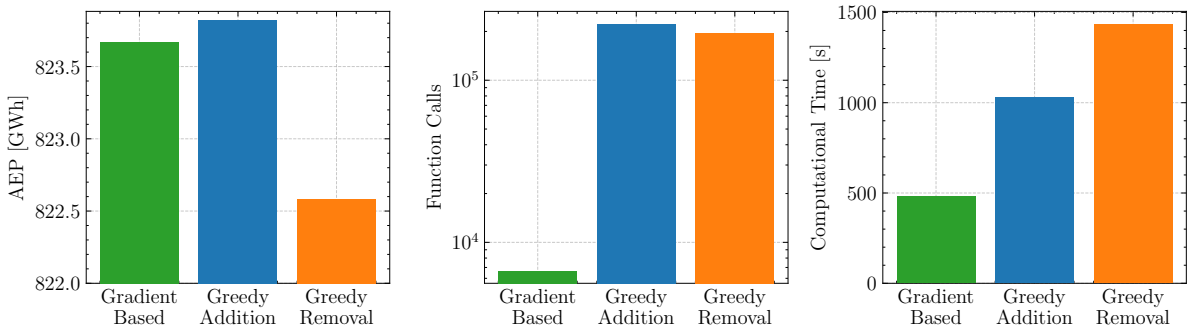


Figure 4.21: AEP (left), function calls (center), and computational time (right) for 50% occupancy using the three optimization strategies.

The AEP achieved by each strategy is directly linked to the layout it produces. Figure 4.22 shows the resulting layouts from the three optimization strategies. Overall, the layouts appear quite similar. All methods place three turbines near the center and distribute the rest around the edges of the farm. This pattern is reasonable, as turbines positioned near the boundaries are less affected by wakes from other turbines and can capture more wind. Interestingly, the Greedy Addition strategy is the only one that leaves the central position empty. This small difference might reduce wake effects in the core of the layout and could help explain why it achieves the highest AEP.

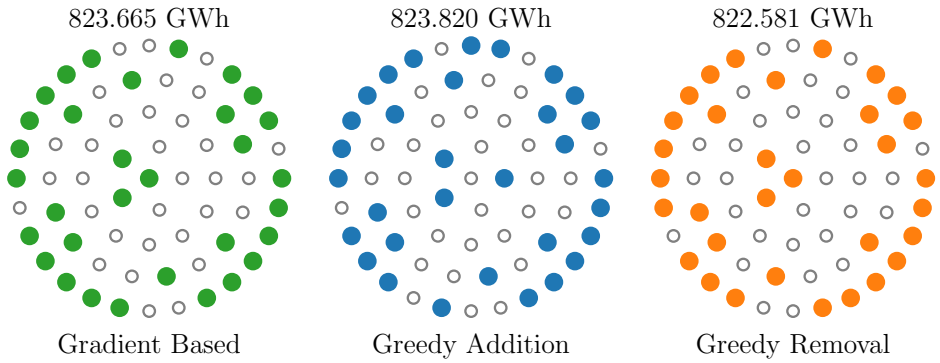


Figure 4.22: Resulting layouts for 50% occupancy using the three optimization strategies.

Given the complexity of wake interactions and the number of design variables, it is challenging to attribute performance differences to individual placement decisions. Therefore, subsequent sections will focus on AEP trends rather than detailed layout comparisons.

In summary, at 50% occupancy, Greedy Addition produces the highest AEP, the Gradient-Based method offers the best compromise between AEP and runtime, and Greedy Removal performs worst on both metrics.

4.4.3.2 Test Case A: Effect of Occupancy

In this test case, the number of turbines is increased from 50% to 90% occupancy by placing more turbines in the available positions, which leads to an increase in AEP. This overall rise is much larger than the differences between optimization strategies, making these differences difficult to distinguish when plotted on the same scale. To enable a clearer comparison, the AEP results are normalized by the best value achieved at each occupancy level, defined as:

$$\mathrm{AEP}_{\mathrm{best}} = \max(\mathrm{AEP}_{\mathrm{GB}},\ \mathrm{AEP}_{\mathrm{Add}},\ \mathrm{AEP}_{\mathrm{Rem}}) \tag{4.5}$$

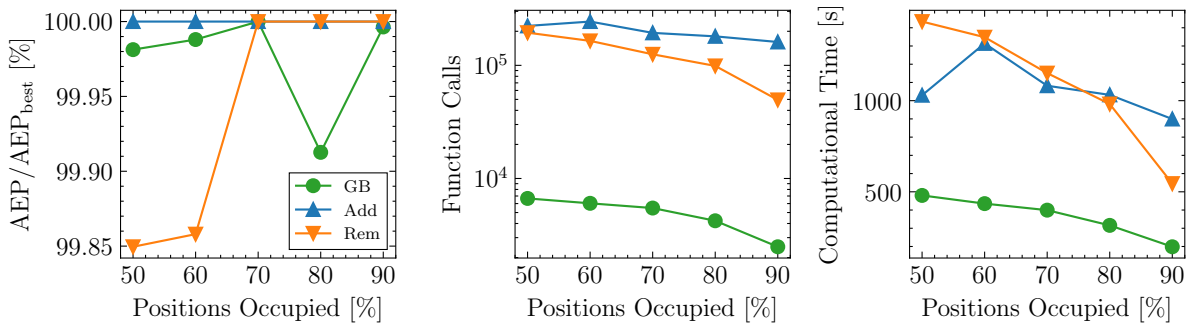


Figure 4.23: Normalized AEP (left), function calls (middle) and runtime (right) for the three strategies at various occupancy levels.

Figure 4.23 shows the normalized AEP, the number of function calls, and the computational time for each method across different occupancy levels. While it remains difficult to precisely explain why certain layouts outperform others, some trends are clear. Greedy Addition

consistently finds the highest AEP across all occupancy levels. The Gradient-Based method performs second-best for 50% and 60%, and matches the best result at 70%. Greedy Removal underperforms at lower occupancies but eventually matches the best AEP beyond 60%.

As expected, both Greedy strategies require roughly an order of magnitude more function calls than the Gradient-Based method. All three show a downward trend in function calls as occupancy increases, which can be explained by the reduced number of possible layouts. As illustrated in Fig. 4.24, the number of possible layouts drops sharply with higher occupancy, which effectively reduces the complexity of the optimization problem.

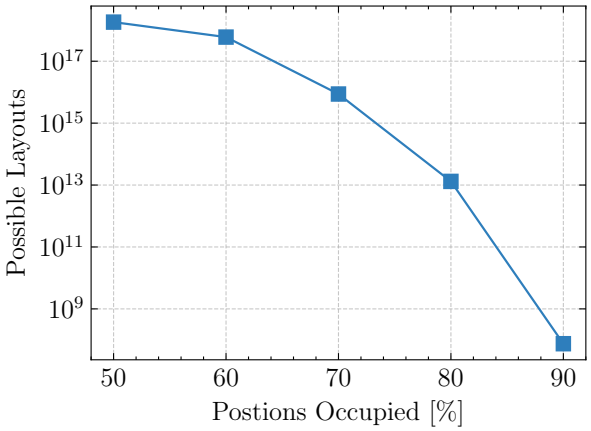


Figure 4.24: Number of possible layouts against occupancy.

Computational time shows a similar trend to AEP, decreasing as occupancy increases. Among the three strategies, the Gradient-Based approach is consistently the fastest. This is because it requires roughly an order of magnitude fewer PyWake calls than the Greedy methods. However, the number of function calls alone does not fully explain the runtime behaviour. As discussed earlier, only AEP evaluations via PyWake are counted as function calls, whereas other time-consuming operations are excluded from this count.

The two Greedy variants display opposite patterns depending on occupancy. Greedy Addition is faster at 50–70% occupancy because it begins with only a few turbines, making each early AEP evaluation very cheap. On the other hand, Greedy Removal becomes faster at 80–90% occupancy, as it needs to remove only a handful of turbines from an almost full layout, greatly reducing the total number of expensive wake calculations. Because computational time is the more relevant metric for strategy selection, plots of function calls are omitted in the following sections.

In summary, Greedy Addition produces the highest AEP across occupancy levels, the Gradient-Based method is the quickest, and the Greedy Removal match the results of Greedy Addition for high occupancies but for slightly lower runtimes.

4.4.3.3 Test Case B: Effect of Variable C_T

In this test case, the number of turbines again increases from 50% to 90% occupancy, but a variable C_T turbine is used. Figure 4.25 presents the normalized AEP and computational time for each method across occupancy levels. The results are nearly identical to those from the constant C_T turbine of Test Case A.

This similarity arises because the inflow is uniform across all directions, with a constant wind speed of 9.8 m/s corresponding to the rated speed of the IEA-3.35 MW turbine. Below this rated speed, the variable C_T curve aligns closely with the constant one, so that any wake-induced wind speed deficits remain within the same flat region of the thrust curve. As a result, wake losses and power production are nearly identical, and the resulting optimization surface of the objective function changes very little. Consequently, all three optimization strategies converge to almost identical results.

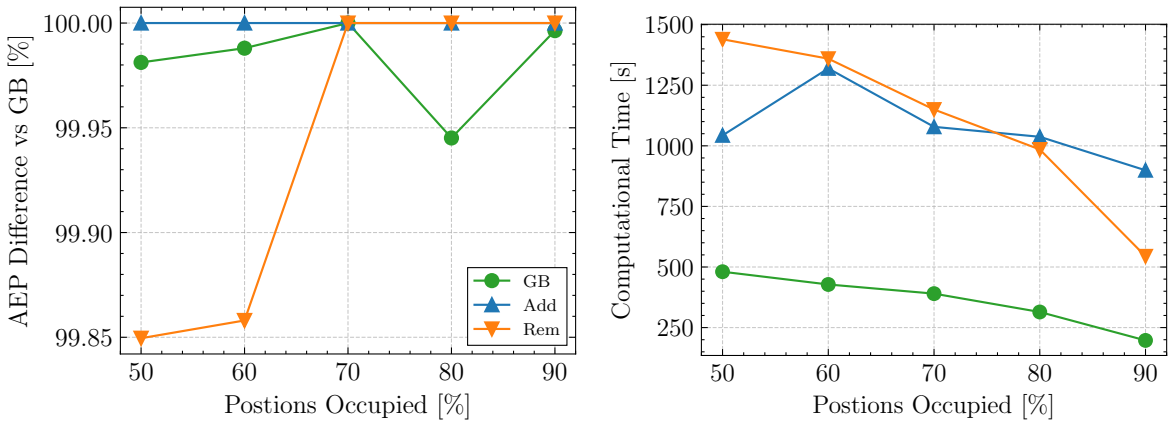


Figure 4.25: Normalized AEP (left) and computational time (right) for the three strategies at various occupancy levels, using a variable C_T wind turbine.

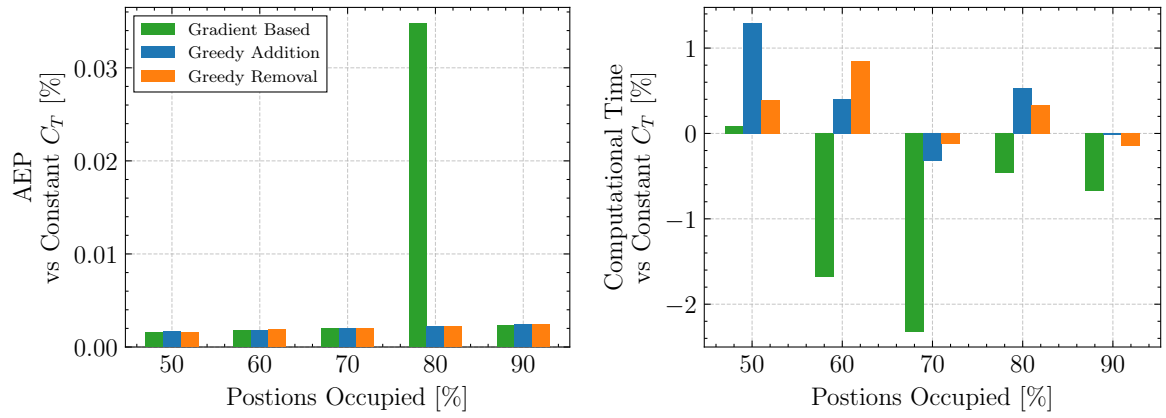


Figure 4.26: Difference in AEP (left) and computational time (right) between variable and constant C_T wind turbines at various occupancy levels, for the different approaches.

To discern the differences in AEP and computational time between the variable and constant

C_T cases, Fig. 4.26 is created. The computational time is pretty similar across strategies, with differences less than 2.5% and no clear pattern across occupancy levels. Therefore the computational times seem to be unaffected by the use of the variable C_T turbine and the slight differences can be attributed so small differences in computation conditions.

Tiny differences in AEP are noticeable since the C_T curve deviates slightly from the constant one near rated speed, altering wake strength and thus power production. The unusually larger AEP gap at 80% occupancy for the Gradient-Based method occurs because the variable C_T turbine leads the optimizer to a slightly different local optimum. The two layouts are slightly different, as illustrated in Fig. 4.27, which explains the larger performance discrepancy at that occupancy level.

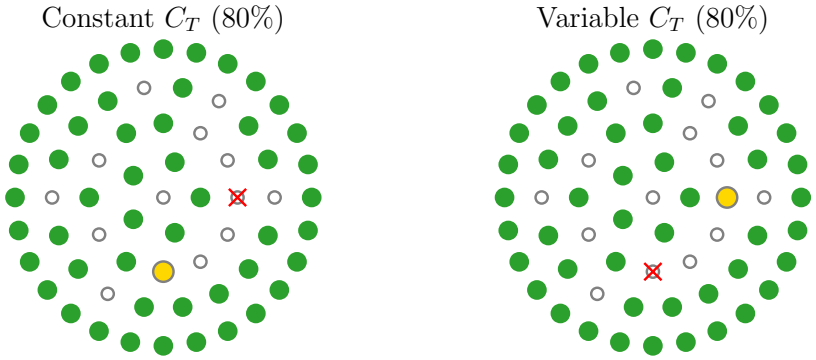


Figure 4.27: Resulting layouts using the Gradient Based approach for 80% occupancy, using a variable C_T (left) and constant C_T (right) wind turbine. Yellow dots mark selected positions and red crosses mark rejected positions.

In summary, using a variable C_T turbine yields results that are nearly identical to those of Test Case A, primarily because the inflow is uniform across directions and fixed at the rated wind speed of 9.8 m/s. Minor discrepancies, such as the larger AEP gap for the Gradient-Based method at 80% occupancy, highlight the higher sensitivity of gradient-based optimization to small differences in the C_T curve. In contrast, the Greedy strategies are less affected by such subtle aerodynamic changes, as their performance is driven more by the discrete placement logic than by continuous variations in the turbine model. If a broader wind speed distribution had been considered instead, the three strategies would likely have produced more distinct results compared to the constant C_T case. Nonetheless, the variable C_T turbine is used in all subsequent test cases to provide a more realistic aerodynamic representation.

4.4.3.4 Test Case C: Effect of Power Density

In this test case, turbine count again increases from 50% to 90% occupancy, but three farm power densities are compared. As expected, lower density reduces wake overlap and increases AEP. Since this effect is independent of the optimization strategy, the detailed results are provided in Fig. A.7, while Figure 4.28 focuses on the normalized AEP of each strategy across all occupancies and densities.

Greedy Addition appears to always find the best AEP regardless of density. This is because adding turbines one-by-one at the location of maximum AEP gain, makes this method adapt effectively to any density. A lower density simply increases each turbine's contribution, but the selection rule remains valid and yields the best incremental AEP gain.

Greedy Removal also appears to find the best AEP regardless of density, but primarily at higher occupancies (beyond 60%). This is for the same reason as Greedy Addition. The reason it struggles at low occupancy is because it must remove many turbines from a fully packed layout with strong, overlapping wakes, so each myopic "best removal" decision compounds errors and leads away from the best layout.

Gradient–Based comes closer to the best AEP as density decreases. This is because lower density produces a smoother, less rugged optimization surface, so SLSQP's gradient estimates and linear approximations become more accurate, allowing the continuous x_i variables to converge more reliably to the best 0–1 solution.

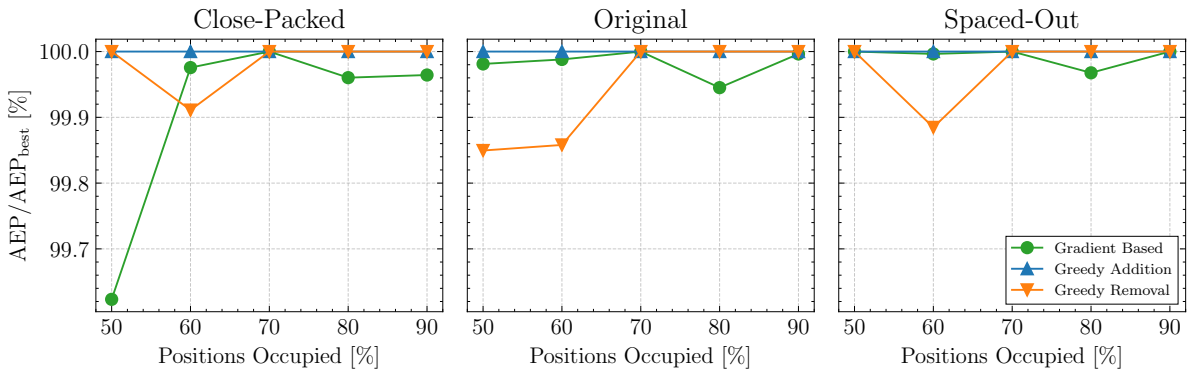


Figure 4.28: Normalized AEP versus occupancy for all strategies under three farm power density configurations.

Figure 4.29 shows that density has little direct effect on overall computational time. The occupancy-driven patterns observed in Test Case A are again visible: Gradient-Based remains the fastest overall, Greedy Addition is quicker at low occupancies, and Greedy Removal at high occupancies.

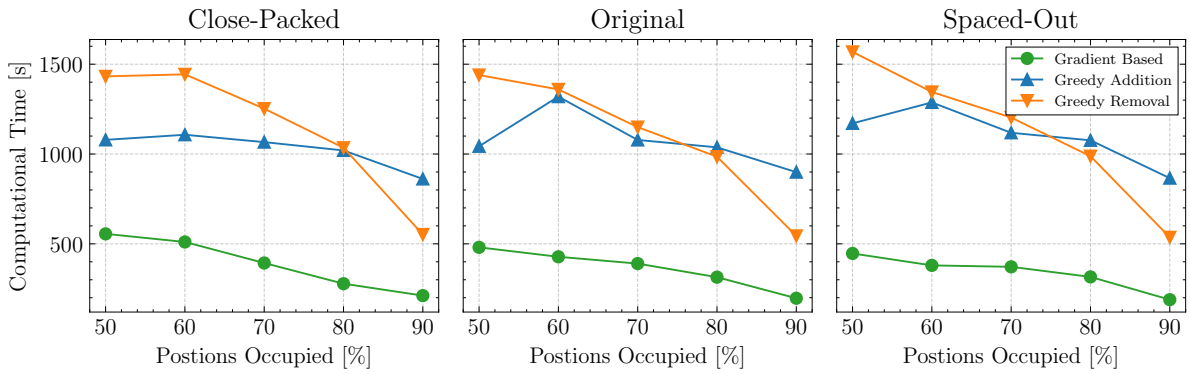


Figure 4.29: Computational time versus occupancy for all strategies under three farm power density configurations.

To help understand how power density affects each method separately, Fig. 4.30 is provided. For the Gradient–Based method, lower density reduces wake interactions and smooths the optimization surface, requiring fewer PyWake evaluation and making the gradient steps more reliable. This yields a clear speed-up at lower occupancies, although the benefit tapers off as more turbines have to be placed.

By contrast, both Greedy Addition and Greedy Removal are driven almost entirely by how many AEP evaluations they perform, so density has little direct effect on their runtimes. The noticeable spike in Greedy Addition at 60% occupancy is likely due to the specific interaction between the layout, distance between turbines, and windrose at that density. It reflects the nature of the problem rather than a consistent trend.

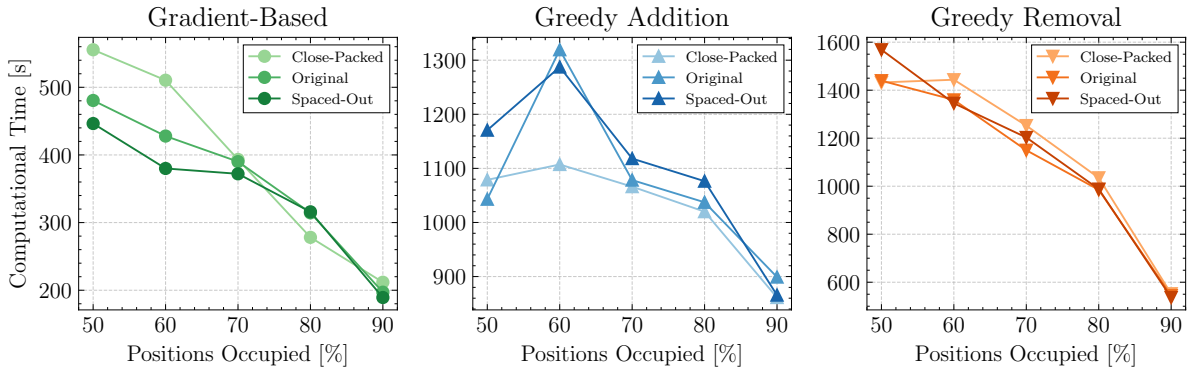


Figure 4.30: Computational time of each strategy separately.

In summary, Greedy Addition consistently delivers the highest AEP across all power densities and occupancy scenarios, thanks to its one-by-one placement strategy. Greedy Removal catches up at high occupancies, where only a few removals are needed. Gradient-Based remains the fastest overall and benefits most from decreased density, as reduced wake interactions yield a smoother optimization surface and quicker convergence.

4.4.3.5 Test Case D: Effect of Number of Available Positions

In this test case, turbine count again increases from 50% to 90% occupancy, but now the three farms have the same power density while having a different number of available positions: 36, 64 and 94. Figure 4.31 presents the normalized AEP of each strategy across all occupancies and available position sets. It should be noted that the Gradient-Based approach uses 100 multistarts for all three farm sizes, while the Greedy Heuristics use as many multistarts as the number of available positions (i.e. 36, 64 and 94).

Both Greedy strategies generally succeed in finding the best AEP across all three farm sizes. However, Greedy Removal performs noticeably worse at 50% and 60% occupancy in the original farm with 64 positions, while it performs well in the larger farm where wake interactions are even stronger. This suggests that its underperformance is not solely due to wake effects, but likely stems from a compounding of suboptimal "best removal" decisions that are sensitive to the specific layout and occupancy level.

In contrast, the Gradient-Based method consistently struggles to find the best AEP, especially in the largest farm with 94 available positions. One might argue that the increasing number of multistarts used by the Greedy methods (equal to the number of candidate positions) gives them an unfair advantage over the Gradient-Based approach, which uses a fixed number of 100. However, even in the smallest farm with 36 positions, the Gradient-Based method only finds the best AEP at 50% occupancy, while both Greedy strategies, with just 36 multistarts, succeed across all occupancies. This suggests that the Gradient-Based approach explores the design space less effectively and is more prone to becoming trapped in local optima compared to the Greedy Heuristics.

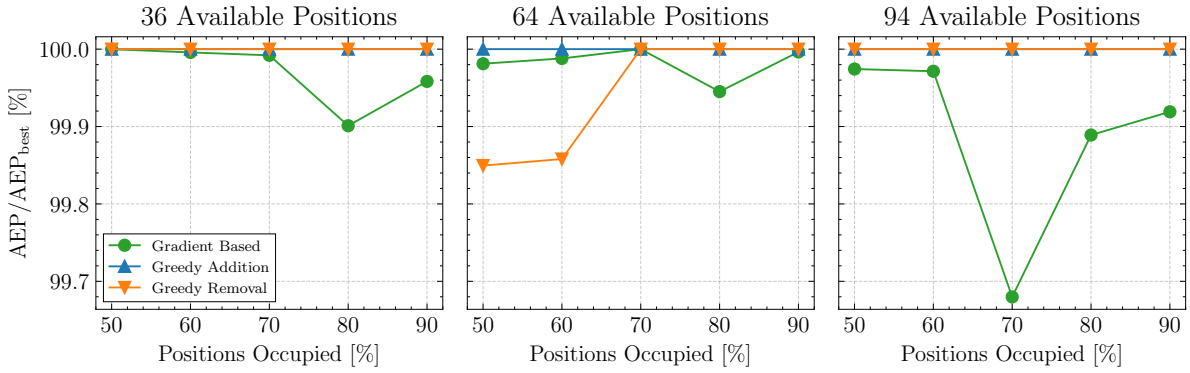


Figure 4.31: Normalized AEP versus occupancy for all strategies under three sets of available positions.

Figure 4.32 shows the computational time of each method across all occupancies and farm sizes. As expected, increasing the number of candidate positions increases runtime for every strategy, due to both the larger design space and the greater number of wake interactions per AEP evaluation. It appears that the computational time of the Greedy Heuristics grows faster with farm size compared to the Gradient-Based approach. This is because of two reasons: (i) the choice to keep the number of multistarts equal to the number of available positions for the former strategies, (ii) the Gradient-Based optimizer exploits derivative information and to converge in few iterations.

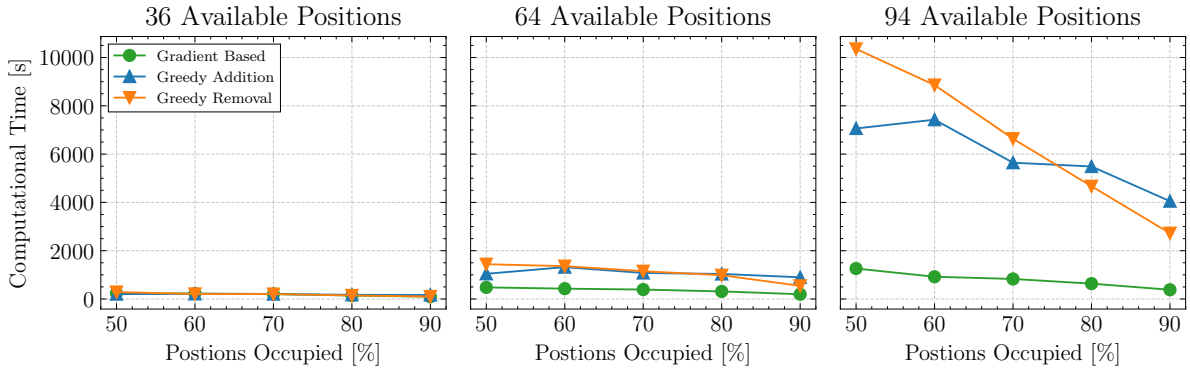


Figure 4.32: Computational time versus occupancy for all strategies under three sets of available positions.

Additionally, one can observe that the slope of the computational time versus occupancy level increases as the farm size increases. This makes sense as the number of possible layouts drops more sharply the greater the number of available positions (see Fig. 4.24), so runtime drops off faster as occupancy increases.

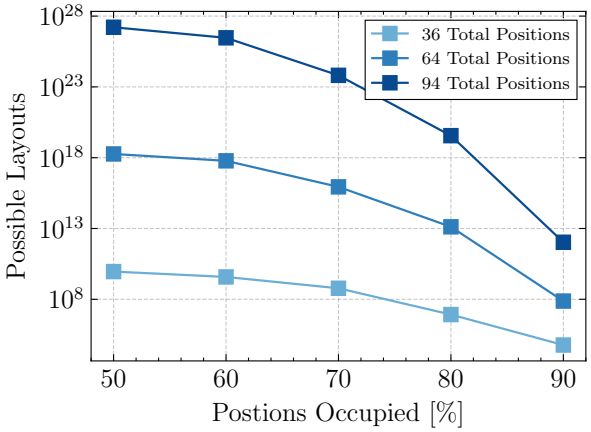


Figure 4.33: Number of layout combinations against occupancy per farm size.

In summary, both Greedy strategies consistently find the best AEP across farm sizes, while the Gradient-Based method struggles, especially in larger farms. This highlights the Greedy approaches' stronger ability to explore the design space effectively, though at the cost of higher computational time as farm size grows. However, Gradient-Based shows a clear advantage in terms of scalability.

4.4.3.6 Test Case E: Effect of Scaled IEA-55 Layout

In this test case, the number of turbines again increases from 50% to 90% occupancy, but this time a scaled-down version of the IEA-55 layout is used, containing 74 available positions. Figure 4.34 shows the normalized AEP and computational time of each method across all occupancy levels. The results are broadly consistent with the previous test cases.

Greedy Addition achieves the best AEP at all occupancy levels, while Greedy Removal performs slightly worse at 50% occupancy, likely due to a compounding of suboptimal removal decisions that depend on the initial layout and turbine density. The Gradient-Based method never finds the best AEP, reflecting its poorer exploration of the design space and greater tendency to get stuck in local optima. The dip at 70% occupancy likely stems from the specific characteristics of the IEA-55 layout at that turbine count.

As for computational time, the same trends that were discussed before are observed. Gradient-Based remains the fastest across all occupancies, while Greedy Addition is quicker at lower occupancies and Greedy Removal is faster at higher ones.

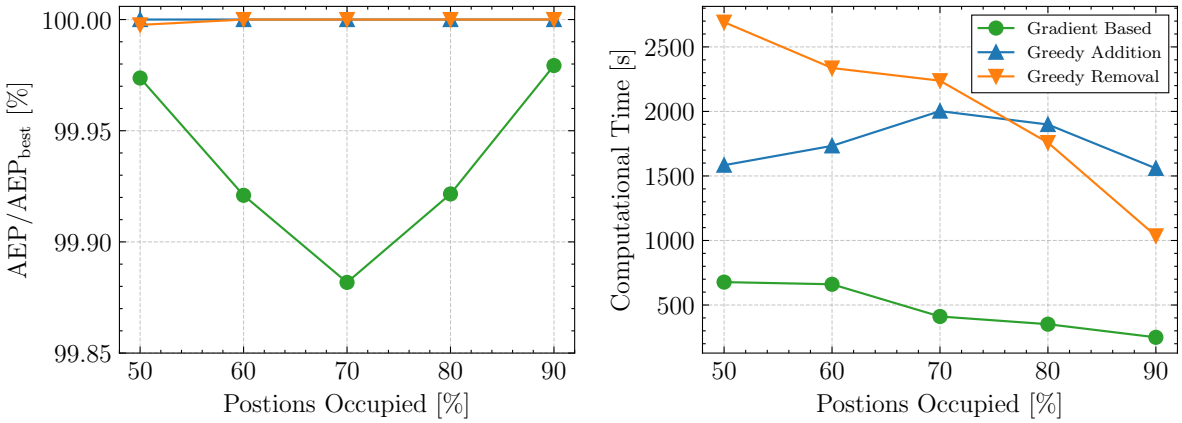


Figure 4.34: Normalized AEP (left) and computational time (right) for the three strategies at various occupancy levels, using the 74 available positions of the scaled-down IEA-55 farm.

In summary, Greedy Addition remains the most effective in terms of AEP, while Gradient-Based is consistently the fastest but least reliable. Greedy Removal performs well overall but is sometimes sensitive to layout and occupancy interactions.

4.5 Discussion

This chapter examined the down-selection problem in three stages: a broad screening and ranking of candidate algorithms, a detailed tuning of the shortlisted strategies, and a comparative evaluation across multiple test cases. While the preceding sections provided results at each stage, the present discussion synthesises these findings to identify general lessons and broader implications.

The literature-based ranking in Section 4.2 showed that Gradient-Based optimization and the Greedy Heuristic are the most promising strategies for down-selection. Their strengths are complementary: the Gradient-Based approach scales well to larger farms, while the Greedy Heuristic requires little parameter tuning and is straightforward to implement. Metaheuristics such as GA, BPSO, and VNS were penalised by their reliance on hyperparameter calibration and weaker scalability, and exact formulations like MILP proved less suitable in this setting because they require linearisation of the objective. These findings emphasise that the choice of strategy depends not only on theoretical performance but also on practicality and usability. At the same time, the wider literature suggests that different contexts can shift the balance. For example, robust optimization with Pareto-front exploration requires computational speed, making pre-computation of wakes and gradient-based approaches particularly attractive. Conversely, if linearisation is acceptable, exact methods such as MILP or QIP offer the guarantee of optimality, provided they are handled by experienced users. When ease of implementation is prioritised, genetic algorithms remain appealing, especially since ready-to-use implementations are available in frameworks like TOPFARM, though their effectiveness depends on careful tuning.

The tuning study in Section 4.3 highlighted important sensitivities in strategy performance. For the Gradient-Based approach, the observation that no interpolation outperformed the tested smoothing functions indicates that discretisation aids should be applied with caution, as they may reduce solution quality. Likewise, the limited influence of driver tolerance and multistarts on AEP demonstrates that default parameter settings can be unnecessarily conservative, resulting in wasted computational effort. For the Greedy Heuristic, tuning revealed that refining by ascending individual AEP contribution (Greedy Addition) can achieve similar results with fewer function calls than the baseline strategy, while removing turbines from a full layout (Greedy Removal) offers a promising alternative. The strong dependence of the final outcome on the starting solution confirmed that a multistart approach is essential to ensure better exploration. Although the number of multistarts was kept relatively high in this study, the results showed that comparable AEP levels could be achieved with substantially fewer runs, with the number of function calls scaling approximately linearly with the number of starts. Together, these results suggest that careful parameter selection is at least as critical as algorithm choice in shaping down-selection performance.

The comparative test cases in Section 4.4 confirmed and extended these insights. Table 4.14 summarises the relative performance of the three algorithms across all cases, illustrating how the trade-off between AEP and runtime persists under different problem conditions.

Table 4.14: Overview of relative performance of Gradient-Based (GB), Greedy Addition (GA) and Greedy Removal (GR) across the benchmark (BM) and all test cases.

Case	Effect of	AEP	Time	Notable Observations
BM	$C(64, 32)$	GA	GB	Confirms general trade-off
A	Occupancy	GA	GB	GR competitive at high occupancy
B	Variable C_T	GA	GB	GB highly affected by C_T
C	Power Density	GA	GB	High density slows down GB
D	Available Positions	GA/GR	GB	GB easily trapped but scalable
E	Scaled IEA-55 Layout	GA/GR	GB	GB never reached best AEP

The results across the benchmark and test cases highlight a consistent trade-off between accuracy and computational cost. The Gradient-Based approach was always the fastest and scaled well with problem size, but it was unable to reliably achieve the highest AEP and often became trapped in local optima. Greedy Addition reliably delivered the highest AEP, yet at high occupancies Greedy Removal achieved comparable solutions in less time by eliminating turbines from a full layout rather than constructing one sequentially. These patterns indicate that no single strategy dominates across all conditions. Instead, their suitability depends on the problem context: Gradient-Based methods are appealing when runtime is critical, Greedy Addition is most effective for maximising AEP in low-occupancy scenarios, while Greedy Removal is preferable in high-occupancy cases.

Taken together, the findings of ranking, tuning, and comparative evaluation suggest that there is no universally superior algorithm. Instead, the performance of each strategy depends strongly on problem characteristics such as occupancy level, farm size, and available computational budget. The broader implication is that algorithm selection in down-selection should not be treated as a one-time choice but as a context-dependent decision. Furthermore, the consistent influence of tuning parameters indicates that future applications should invest in systematic calibration to unlock the full potential of each method.

In conclusion, this chapter shows that while greedy heuristics provided the most effective means of maximising energy yield within the tested scenarios, their computational cost cannot be ignored. Gradient-based relaxations offer a scalable alternative when runtime is the dominant constraint, but their accuracy is limited. The interplay between algorithm choice, tuning, and problem conditions is therefore central to effective down-selection, and the insights from this chapter form the basis for the subsequent analysis of the broader impacts in Chapter 5

CHAPTER 5
Impact of Down-Selection (RQ2)

This chapter addresses the second research question (RQ2) of the thesis. It begins with an overview of the methodology in Section 5.1, followed by a description of the experimental setup in Section 5.2. The chapter concludes with a presentation and discussion of the results in Section 5.3. For reference, RQ2 is stated as follows:

RQ2. *What is the impact of down-selection on the final wind farm layout?*

1. How does the performance of the down-selected farm compare to an optimal layout designed directly for higher-rated turbines?
2. How does the performance of the down-selected farm change when the initial layout already satisfies the minimum spacing for higher-rated turbines?
3. How does the comparison between down-selected and directly optimized farms change when all layouts are generated using a larger minimum spacing constraint?
4. How does down-selection influence the spatial distribution of turbines within the layout?

5.1 Methodology

To investigate the effects of down-selection, optimal layouts must first be established for both the lower-rated and higher-rated turbines. The layout of the smaller-rated turbines defines the set of candidate positions for the down-selection procedure, while the layout of the larger-rated turbines serves as a benchmark for comparison.

Once again, the IEA-55 offshore site is selected for its publicly available data. While the literature provides both regular and irregular layouts for 74 IEA-10 MW turbines at this site (see Kainz et al. [21]), no equivalent layouts exist for turbines of different ratings with comparable installed capacity. Therefore, two new optimized layouts are created: one with $N_1 = 74$ of the IEA-10 MW turbines (740 MW) and one with $N_2 = 50$ of the IEA-15 MW turbines (750 MW). The small difference in installed capacity ($\approx$1.3%) is considered acceptable but should be kept in mind when comparing AEP results.

The general procedure for RQ2 is as follows. First, an optimal layout is generated for the 10 MW case. A down-selection procedure is then applied to obtain reduced layouts suitable for 15 MW turbines, subject to minimum spacing constraints ($S_{\text{min},1}$, $S_{\text{min},2}$). These down-selected layouts are compared against directly optimized 15 MW layouts in terms of energy production and turbine distribution within the layout. This procedure is repeated under three different spacing assumptions, which form the basis of the test cases described below.

The general procedure for RQ2 is illustrated in Fig. 5.1. For each turbine rating, a grid of candidate positions is first generated with the prescribed minimum spacing constraints ($S_{\text{min},1}$, $S_{\text{min},2}$). From this grid, multiple initial layouts are produced and refined through continuous optimization to obtain a set of candidate solutions. The best-performing layout from this set is retained as the *optimal layout* for that turbine size.

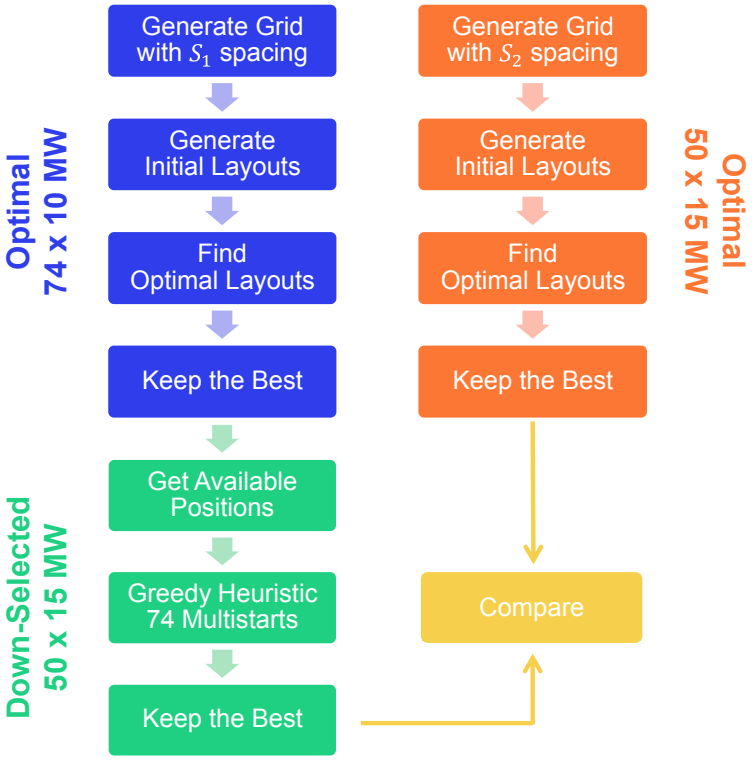


Figure 5.1: General procedure for evaluating the impact of down-selection (RQ2).

In the next stage, the optimal layout for the 74 lower-rated turbines serves as the pool of available positions for the down-selection procedure. The Greedy Heuristic is applied across multiple starts (as many as the available positions, i.e. 74) to generate 50-turbine layouts that satisfy the minimum spacing constraint, and the layout with the highest AEP is retained.

Finally, the resulting down-selected layout is compared against the directly optimized 15 MW layout. Each block of this process is described in more detail in the following subsections.

5.1.1 Minimum Spacing Constraint and Test Case Definition

The minimum spacing constraint not only governs layout feasibility but also provides the basis for defining the three test cases examined in RQ2. The constraint is expressed as a minimum separation distance S_{min} between turbines, relative to rotor diameters. Here, $D_1 = 198$ m and $D_2 = 240$ m denote the rotor diameters of the IEA-10 MW and IEA-15 MW turbines, respectively.

There is no universally accepted value for S_{min}. For example, the IEA-55 reference layout applies $S_{\text{min}} = 2D$ [21], whereas other studies use values exceeding $5D$ depending on objectives

and modeling assumptions [46]. In this work, the selected values balance the scope of this work and the limitations and the reliability of the engineering wake model.

Three test cases are defined in Table 5.1. Test Case A provides the baseline for RQ2.1 by comparing down-selected and directly optimized layouts under moderate spacing. Test Case B addresses RQ2.2 by ensuring the initial 10 MW layout already meets the 15 MW spacing requirement. Test Case C addresses RQ2.3 by examining the effect of wider separations. The effect of down-selection on the spatial distribution of turbines (RQ2.4) is examined within each of the three test cases.

Table 5.1: Minimum spacing constraints for each test case. $S_{\text{min},1}$ and $S_{\text{min},2}$ denote the spacing applied to the 10 MW and 15 MW turbines, respectively.

Test Case	$S_{\text{min},1}$	$S_{\text{min},2}$
A	$3.5D_1$	$3.5D_2$
B	$3.5D_2$	$3.5D_2$
C	$5.0D_1$	$5.0D_2$

With these spacing assumptions defined, the following subsections describe how optimal layouts are generated for each turbine rating and how the down-selection procedure is applied.

5.1.2 Optimal Layouts

To generate the optimal layouts used as benchmarks in RQ2, a two-stage procedure is followed: (i) an initial placement of turbines using the Smart-Start heuristic (explained below), and (ii) refinement through continuous optimization with SLSQP. An illustrative example of this process is shown in Fig. 5.2.

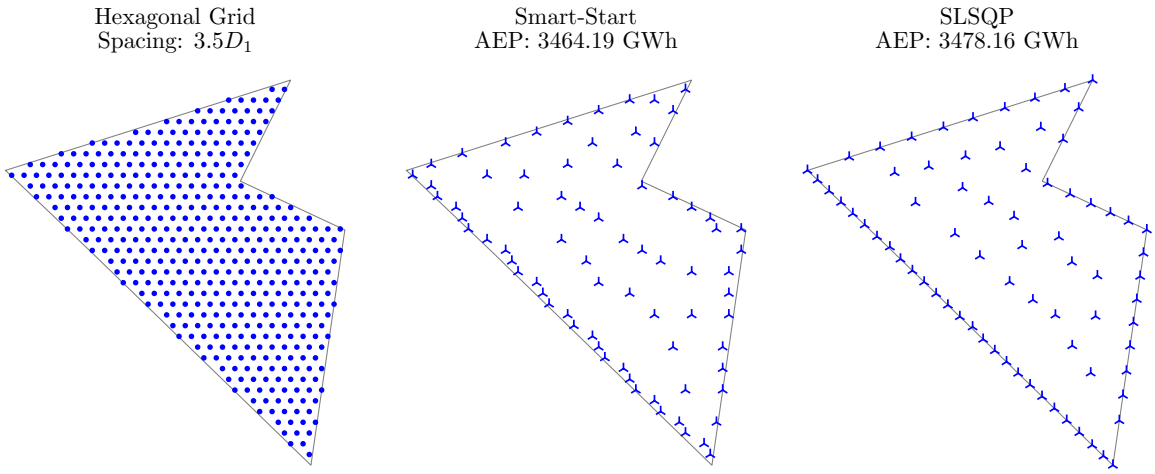


Figure 5.2: Example of the procedure used to generate one of the candidate layouts for the optimal 10 MW case under a $3.5D_1$ spacing constraint.

First, hexagonal grids of candidate points are generated within the farm boundaries using spacings of $S_{\text{min},1}$ and $S_{\text{min},2}$ for the IEA-10 MW and IEA-15 MW turbines, respectively. Hexagonal grids are chosen over square ones because they maintain more uniform spacing between adjacent turbines in all directions, reducing the likelihood of directional bias in wake interactions. The grid resolution is set equal to the minimum spacing constraint, as a finer resolution would substantial increase the computational cost.

From each grid, turbine positions are selected using the *Smart-Start* algorithm provided in TOPFARM. Smart-Start is a greedy heuristic that functions similarly to Greedy Addition without refinement (see Strategy A in Table 4.5): it sequentially places turbines in the locations that maximize AEP given the turbines already positioned. The algorithm includes a parameter controlling the probability of making random instead of greedy selections. In this work, that random selection percentage is set to zero, following Valotta Rodrigues et al. [46], to ensure reproducibility and to avoid introducing unnecessary stochastic variation.

Once the desired number of turbines has been placed with Smart-Start, the layout is refined through continuous optimization using the SLSQP algorithm. As in RQ1, SLSQP is chosen for simplicity and consistency with the gradient-based optimization approach. The optimizer is run for up to 1000 iterations, with a convergence tolerance of 10^{-3} and an expected cost of 10^{-6}. These settings were adopted from the IEA-55 benchmark GitHub repository, since they are not explicitly detailed in the corresponding publication.

Because SLSQP is highly sensitive to initial conditions, the Smart-Start + SLSQP procedure is repeated 50 times with different seeds. While Smart-Start is deterministic for a given seed when the random selection percentage is zero, changing the seed alters the position of the first turbine, resulting in different initial layouts. This approach allows a controlled exploration of multiple starting configurations while maintaining reproducibility. Among these 50 runs, the layout with the highest AEP is retained as the optimal solution for both the 74×10 MW and 50×15 MW configurations.

The choice of 50 repetitions represents a balance between exploration and computational effort. A smaller number of runs would risk missing promising configurations, since SLSQP can converge to different local optima depending on the starting layout. Conversely, increasing the number far beyond 50 yields diminishing returns while significantly extending runtime. Preliminary testing indicated that 50 runs provided sufficient diversity to reliably capture high-quality layouts while keeping the analysis computationally manageable.

5.1.3 Down-Selection

The 74 optimal turbine positions generated for the 10 MW case are considered as the candidate locations for the down-selection process. Since RQ1 demonstrated that the Greedy Heuristics consistently outperforms the Gradient-Based method in terms of AEP (see Table 4.14), the first approach is selected for RQ2.

For this case, 50 turbines must be placed among 74 available positions, resulting in an occupation rate of approximately 68%. As shown in Section 4.4.3.6, where the same site was used, Greedy Addition and Greedy Removal achieved very similar performance in terms of both AEP and computational time. Since neither approach consistently outperforms the other, both are retained here. This allows potential differences that may arise once the minimum distance

constraint is introduced to be observed, and provides continuity with the comparison carried out in RQ1.

The original implementation of the algorithms does not enforce a minimum distance between turbines, which would result in infeasible layouts in RQ2. To address this, a modification of the algorithms is necessary. While such a modification is more straightforward for Greedy Addition than for Greedy Removal, both are adapted in an equivalent way to avoid ruling out one prematurely and to ensure that any differences in performance can be attributed solely to the choice of Addition versus Removal.

Details on how the minimum distance constraint is incorporated are provided in Section 5.1.3.1, while the relative performance of Greedy Addition and Greedy Removal is presented in Section 5.1.3.2.

5.1.3.1 Incorporation of Minimum Distance Constraint

The modifications to the Greedy Heuristic are introduced to ensure feasibility of the resulting layouts. Stage 1 of the original algorithms, where turbines are either added (Addition) or removed (Removal) one by one, remains unchanged. A new intermediate step, referred to as Stage 1.5, is introduced to handle any violations of the minimum distance constraint before proceeding to the final refinement stage (Stage 2). By applying the same additional step to both algorithms, the comparison remains consistent: the only methodological difference between them lies in whether turbines are initially added or removed.

In Stage 1.5, any turbines that violate the minimum distance constraint are first removed from the layout. The algorithm then attempts to reinsert them one-by-one into available positions that comply with the constraint. For each turbine, the location yielding the highest increase in AEP is selected. This process continues until the desired number of turbines is placed. If no constraint violations are present in the layout generated by Stage 1, then Stage 1.5 is skipped entirely. However, if Stage 1.5 fails to restore a layout with the full number of turbines, the constraint-violating layout from Stage 1 is passed directly to Stage 2, in the hope that the refinement process will resolve the remaining violations.

Stage 2 (refinement) then attempts to improve the layout by relocating individual turbines. First, turbines are sorted in ascending order based on their individual AEP contributions. Each turbine is then considered for relocation to all empty candidate positions. The key modification introduced here is that only positions satisfying the minimum distance constraint are considered during this step. This additional check ensures that the refined layout either maintains validity or resolves any remaining violations from previous stages. For each turbine, the position that results in the highest increase in total AEP is selected. The procedure is repeated for all turbines and continues iteratively until no further improvements are observed.

5.1.3.2 Greedy Addition or Greedy Removal?

To evaluate the two variants of the Greedy Heuristic under spacing constraints, both the Greedy Addition and Greedy Removal algorithms were applied within RQ2 Test Case A, which uses a minimum spacing of $3.5D$. In this setting, the down-selection algorithm is tested only on the subset-selection step, i.e. generating a 50-turbine layout from the 74 available positions.

As shown in the distribution of AEP values across independent runs in Fig. 5.3, the performance of the two methods is generally comparable. However, Greedy Addition achieves a slightly higher maximum AEP, and is therefore chosen as the down-selection method for the remainder of the analysis.

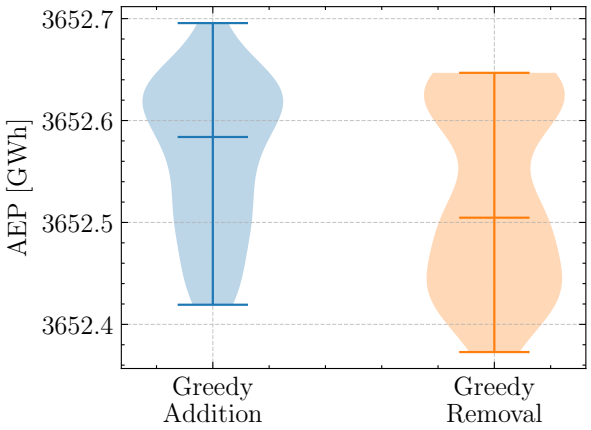


Figure 5.3: Comparison of AEP values from Greedy Addition and Removal across independent runs using the constraint-respecting algorithm.

The final layout is selected using a two-step criterion: (i) prioritize layouts with the fewest spacing violations, and (ii) among those, select the one with the highest AEP. This means that if feasible layouts exist (zero violations), the chosen solution is both high-performing and feasible. If no perfectly feasible layout is found, the algorithm still returns the best-performing solution among those with minimal violations.

A more detailed comparison between the constraint-respecting algorithm and the original unconstrained version is included in Appendix B.2.

5.1.4 Additional Considerations

As in the original work by Kainz et al. [21], wake effects are evaluated over a discrete grid of wind conditions, using increments of 1 m/s in wind speed and 1° in wind direction. This resolution provides a sufficiently detailed characterization of the inflow while keeping computational cost manageable, and is consistent with common practice in large-scale wind farm layout studies.

5.2 Setup

To answer RQ2, the IEA-55 reference offshore site is selected due to its publicly available wind resource data, as described by Kainz et al. [21]. The corresponding wind rose used for this study is shown in Fig. 5.4. While the sectoral Weibull distributions and corresponding frequency data are not reproduced here, they can be found in the original publication.

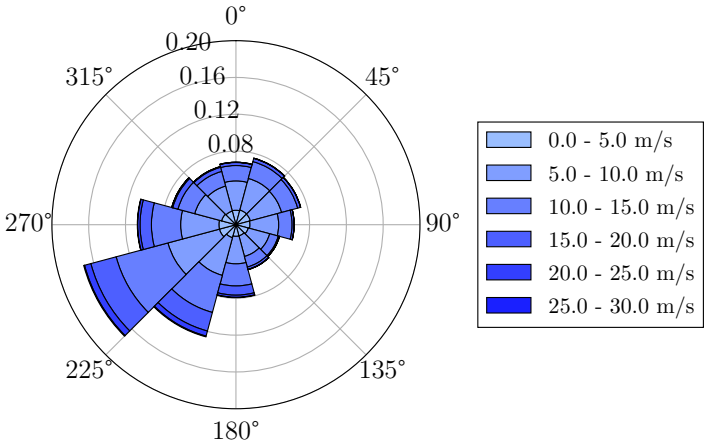


Figure 5.4: Wind rose from the IEA-55 reference site.

The turbines used in this study are the IEA-10 MW and IEA-15 MW reference models. Their power and thrust coefficient (C_T) curves are shown in Fig. 5.5, with data sourced from Bortolotti et al. [9] and Gaertner et al. [15], respectively. An overview of their rotor diameters and hub heights is provided in Table 5.2.

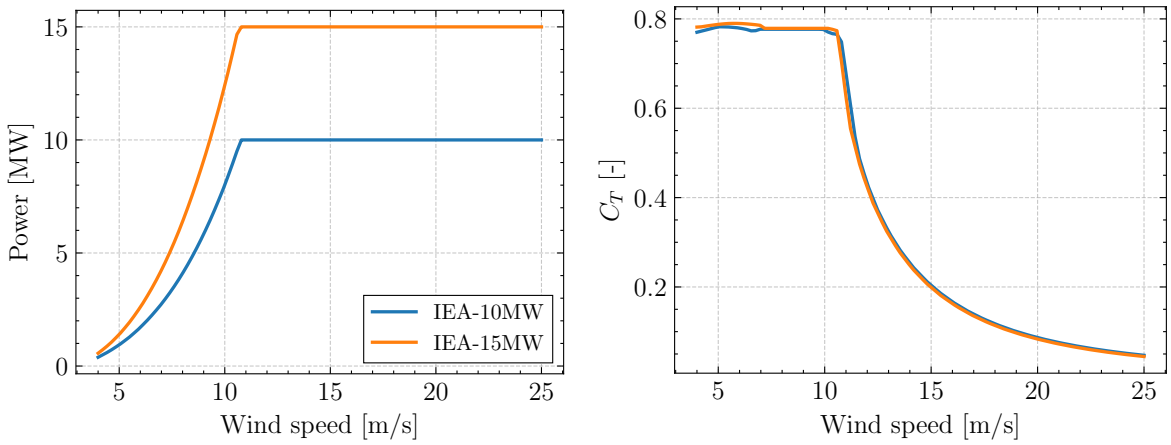


Figure 5.5: Power (left) and C_T (right) curves for the IEA-10 MW and IEA-15 MW reference turbines.

Table 5.2: Diameters and hub heights of the IEA-10 MW and IEA-15 MW reference turbines.

#	Turbine	Diameter (D_i) [m]	Hub Height ($z_{\text{ref},i}$) [m]
1	IEA-10 MW	198	119
2	IEA-15 MW	240	150

To account for differences in height between the wind resource data and the hub height of the IEA-15 MW, the wind speed distributions are adjusted using the power law:

$$U(z) = U_{\text{ref}} \left(\frac{z}{z_{\text{ref}}} \right)^{\alpha} \tag{5.1}$$

The reference height and shear exponent for the IEA-55 site are $z_{\text{ref}} = 119$ m and $\alpha = 0.08$, respectively. This adjustment is applied to the scale parameters of the Weibull distributions in each wind direction, using the `PowerShear` functionality in PyWake. The directional frequencies (i.e. the wind rose) remain unchanged, as they are assumed to be independent of height.

The optimal layouts used for Test Cases A, B, and C, generated following the procedure in Section 5.1.2, are shown in Figs. 5.6 to 5.8. These layouts define the starting point for the down-selection procedure and are therefore considered part of the setup, while the subsequent analysis focuses on the impact of down-selection itself.

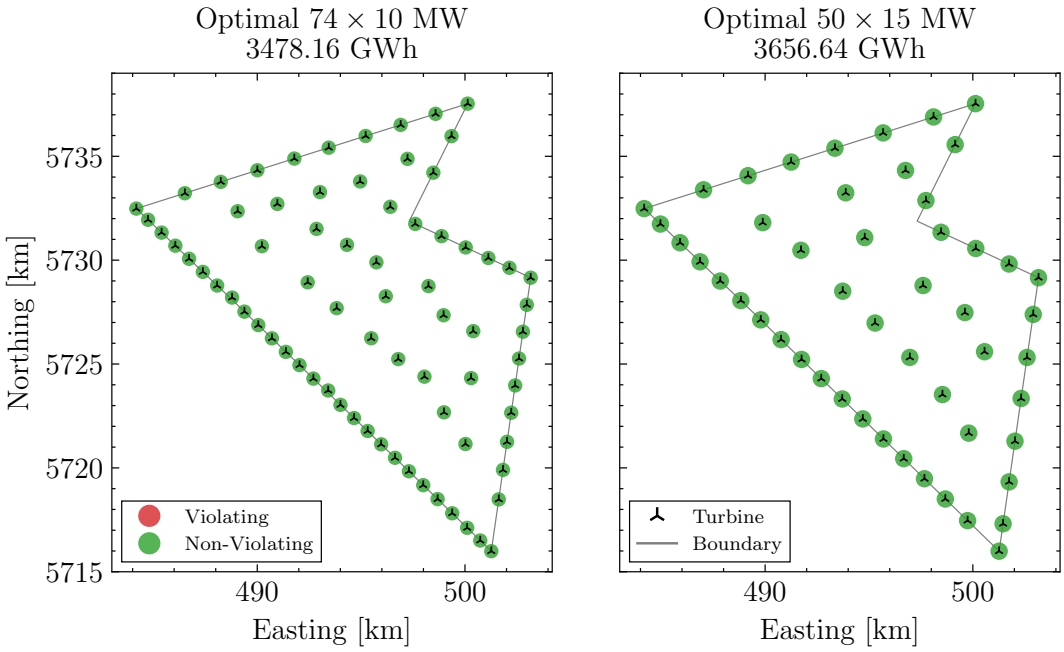


Figure 5.6: Optimal layouts for Test Case A. Constraint radii drawn to scale.

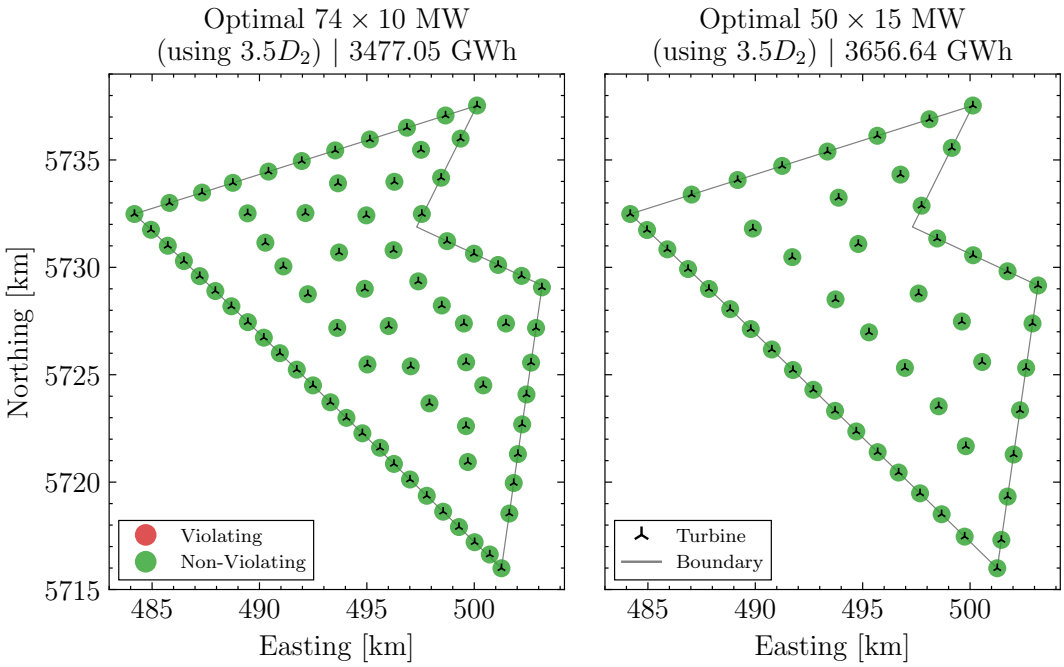


Figure 5.7: Optimal layouts for Test Case B. Constraint radii drawn to scale.

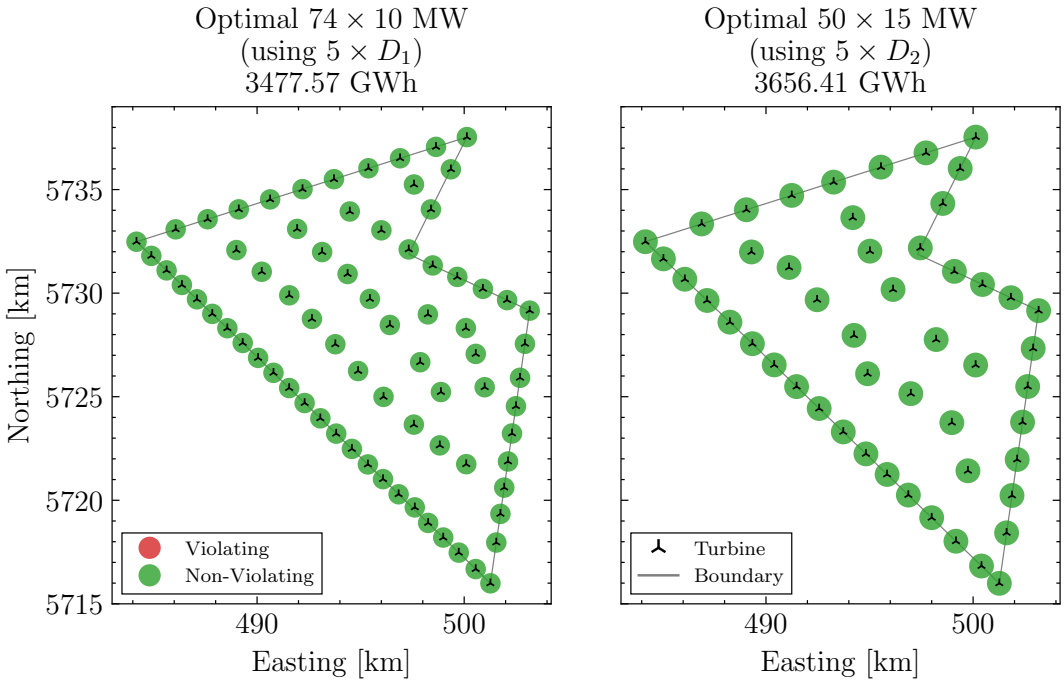


Figure 5.8: Optimal layouts for Test Case C. Constraint radii drawn to scale.

5.3 Results

The results of the three Test Cases for RQ2 are presented below. To enhance understanding and maintain coherence, the layouts are displayed in a format that mirrors the previously presented flowcharts. Additionally, the percentage of wind turbines located along the perimeter of each farm is shown in the bottom right corner of each figure. Also, the following term is used:

$$\text{Untapped Energy Potential} = \frac{\text{AEP}_{\text{Opt, 15 MW}} - \text{AEP}_{\text{Down, 15 MW}}}{\text{AEP}_{\text{Opt, 10 MW}}} \times 100 \tag{5.2}$$

5.3.1 Test Case A: Using a $3.5D$ Spacing

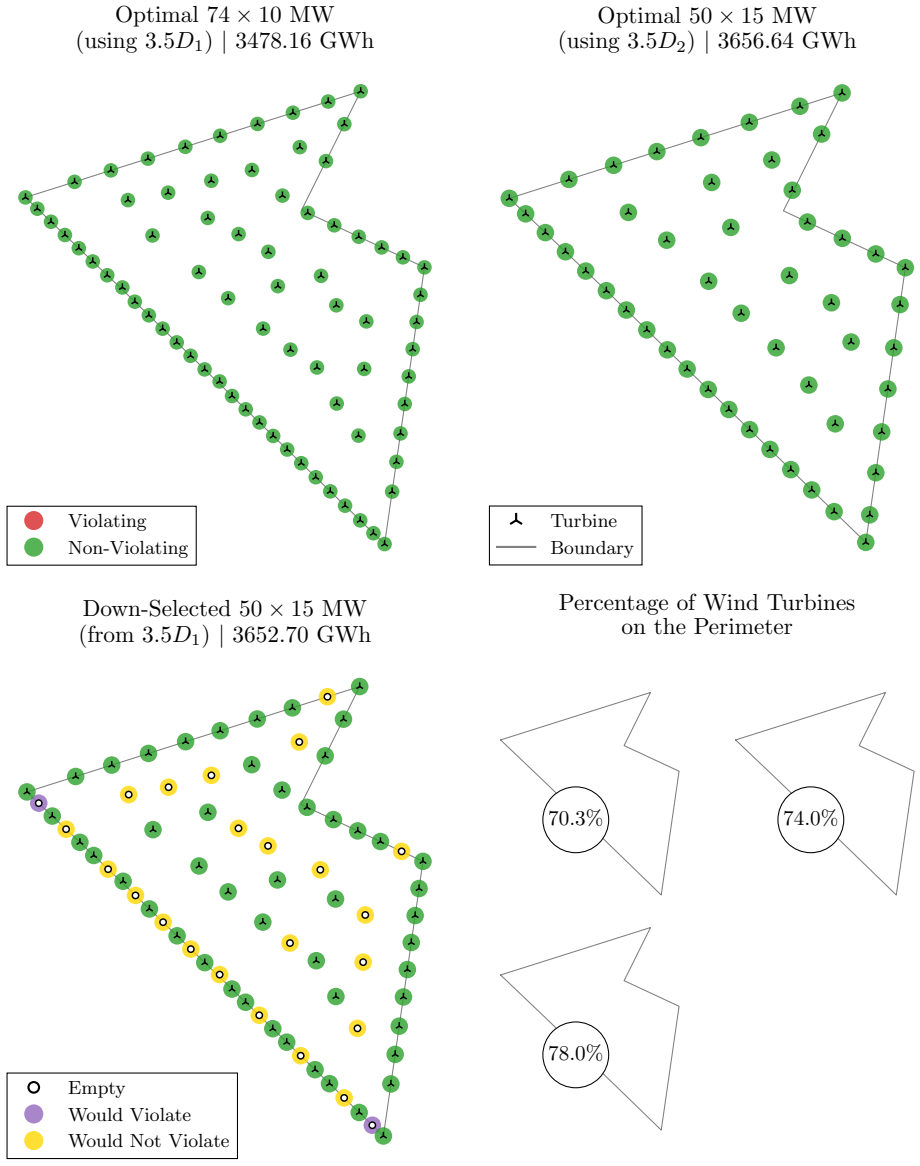


Figure 5.9: Optimal and down-selected layouts for Test Case A. Positions left empty are classified as potentially violating or not. Constraint radii drawn to scale.

The results from using a minimum spacing constraint of $3.5D$ can be found in Fig. 5.9. As expected, all three layouts place the majority of wind turbines along the perimeter of the farm to maximize exposure to clean inflow wind. Notably, the optimal layouts show the highest concentration of turbines in the bottom-left side of the boundary, which aligns with the predominant wind direction originating from that area.

One can observe a significant difference in AEP of almost 5% between the optimal configurations, as reported in Table 5.3. The slightly higher farm capacity of 750 MW compared to 740 MW explains part of the difference in AEP. However, other factors such as the higher hub height and the presence of fewer turbines leading to fewer wakes, also contribute to this significant difference, showcasing the advantage of choosing fewer higher-rated turbines.

Table 5.3: Reduction in AEP compared to optimal 15 MW layout. Values are reported with high numerical precision to ensure consistency with later tables where differences are smaller.

Layout	AEP [GWh]	Difference [%]
Optimal 50×15 MW	3656.642	–
Optimal 74×10 MW	3478.158	-4.881
Down-Selected 50×15 MW	3652.696	-0.108

Moving on to the more interesting comparison, one can see that down-selection yields around 0.1% lower AEP compared to the optimal configuration with 15 MW turbines. Depending on the cost of conducting another seabed survey for the positions of the optimal 15 MW layout that do not overlap with those already surveyed for the 10 MW layout, and other factors (i.e. cost of cabling etc.), one can determine whether this untapped energy potential is acceptable. The reason for the difference in AEP is that despite the down-selected configuration having a higher percentage of its turbines on the perimeter, the SLSQP optimizer used for the optimal layout could take advantage of the continuous design domain, thus placing more wind turbines more favorably. For example, it has manged to place more on the bottom-left side that aligns with the predominant wind direction. As a side note, one can also observe that the constrained Greedy algorithm has managed to place all turbines without any violations and has correctly identified and avoided the two positions (in violet) that would violate the minimum spacing constraint.

5.3.2 Test Case B: Using a $3.5D_2$ Spacing for the 10 MW Layout

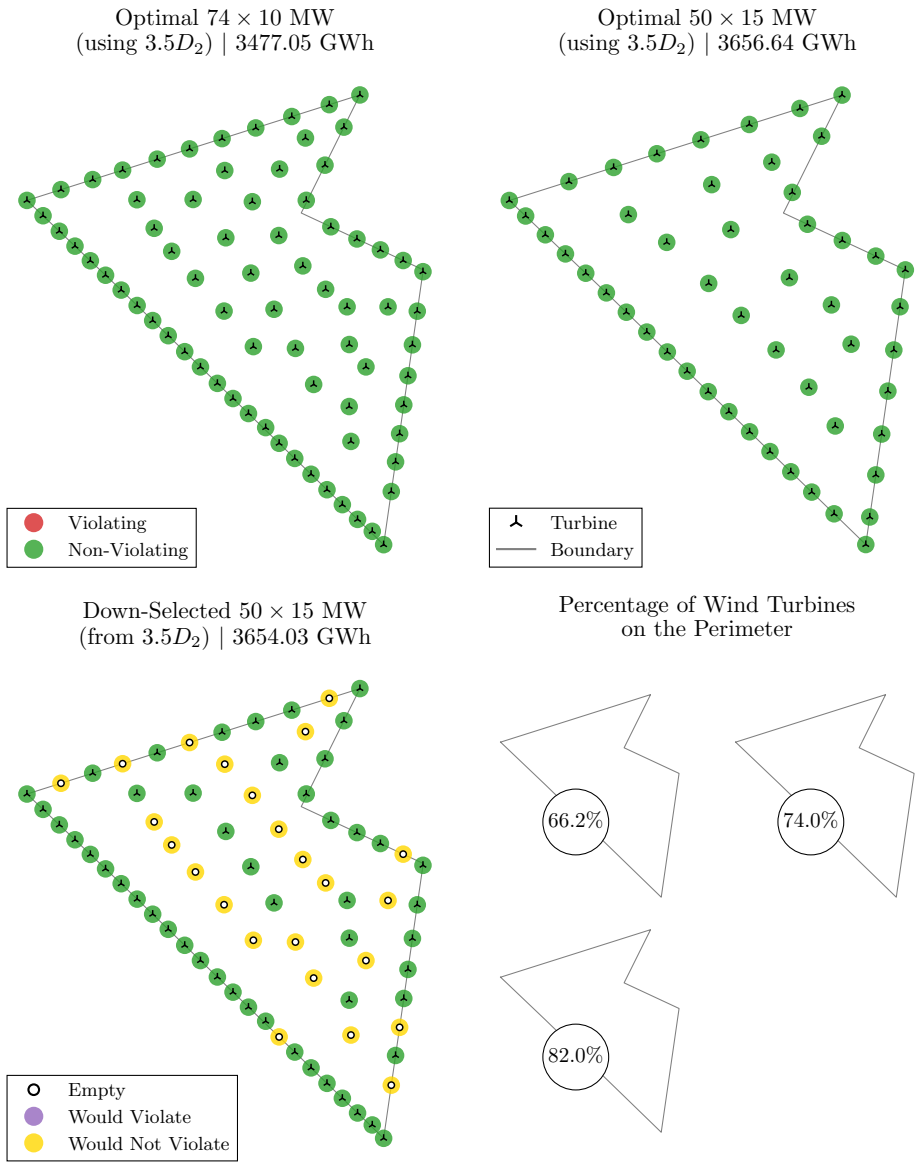


Figure 5.10: Optimal and down-selected layouts for Test Case B. Positions left empty are classified as potentially violating or not. Constraint radii drawn to scale.

The results for Test Case B, where a minimum spacing of $3.5D_2$ is applied to the 10 MW layout, are shown in Fig. 5.10. Similar turbine distribution patterns as in Test Case A can be observed. However, in this down-selected layout, the Greedy algorithm manages to place more turbines along the side aligned with the predominant wind direction, and no empty (violet) positions remain that would have violated the spacing constraint. This is expected, as the starting layout was already optimized under the stricter $3.5D_2$ constraint used for the higher-rated turbines.

The untapped energy potential of the down-selected layouts is summarized in Table 5.4. Notably, the percentage of perimeter turbines increases from 78% in Test Case A to 82% here,

and the down-selected layout experiences a small increase in AEP compared to its counterpart in Test Case A. As expected, this leads to a smaller untapped energy potential when compared to the optimal 15 MW configuration.

Table 5.4: Untapped energy potential of down-selected layouts for different spacing requirements of the 10 MW layouts.

Test Case	Optimal 15 MW AEP [GWh]	Down-Selected 15 MW AEP [GWh]	Difference [%]
A	3656.642	3652.696	-0.108
B	3656.642	3654.031	-0.071

The effect of the increased spacing on AEP of the 10 MW layouts is summarized in Table 5.5. The optimal 10 MW layout shows a slight drop in performance, which is reasonable as the tighter $3.5D_2$ constraint gives the optimizer less flexibility to adjust turbine positions, potentially leading to less efficient placements. For instance, the percentage of turbines placed on the perimeter drops from 70.3% in Test Case A to 66.2% here. In contrast, the down-selected layout benefits from starting with a layout that already respects its minimum spacing constraint, offering a more suitable starting point for the down-selection process.

Table 5.5: Changes in AEP resulting from increased spacing in the initial 10 MW layout.

Layout	Test Case A AEP [GWh]	Test Case B AEP [GWh]	Difference [%]
Optimal 74×10 MW	3478.158	3477.051	-0.032
Down-Selected 50×15 MW	3652.696	3654.031	0.037

Despite the relative differences being small in this test case, the above results lead to the conclusion that taking a small hit on the power production of the lower-rated layout by designing with the higher-rated spacing in mind, could lead to greater power production if a higher-rated turbine ends up being used. However, a more rigorous analysis of the costs associated with each layout should be done before further decisions can be made.

5.3.3 Test Case C: Using a $5D$ Spacing

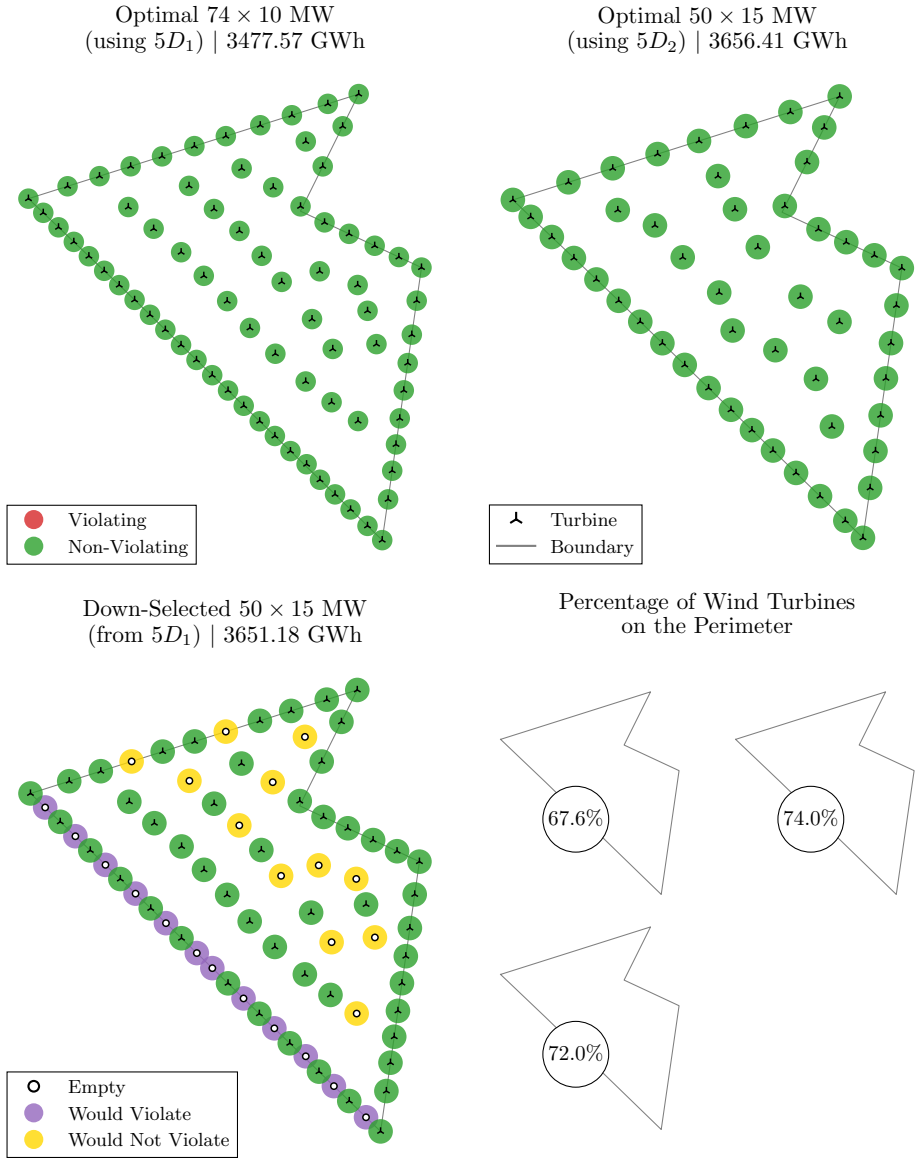


Figure 5.11: Optimal and down-selected layouts for Test Case C. Positions left empty are classified as potentially violating or not. Constraint radii drawn to scale.

The results from using a stricter minimum spacing constraint of $5D$ can be found in Fig. 5.11. Again, similar general turbine distribution patterns as in Test Case A can be observed. However, here it is immediately apparent that the down-selected layout has a greater number of potentially violating empty positions (in violet) compared to Test Case A. This is logical as, as the stricter $5D$ constraint has left less empty space around the wind turbines in the optimal 10 MW layout when compared to Test Case A. Therefore, when the down-selection happens, there is less empty space to accommodate the spacing constraint of the 15 MW. Additionally, the reason that all the potentially violating empty positions are located on the side aligning

with the predominant wind direction is that this is the region where the optimal 10 MW layout has the highest density of turbines.

The effect of the increased spacing on AEP is summarized in Table 5.6. The optimal 10 MW layout shows a slight drop in performance, which is reasonable as the tighter $5D$ constraint gives the optimizer less flexibility to adjust turbine positions, potentially leading to less efficient placements. For instance, the percentage of turbines placed on the perimeter drops from 70.3% in Test Case A to 67.6% here.

Table 5.6: Relative reduction in AEP due to increase in minimum spacing for each layout.

Layout	$3.5D$ AEP [GWh]	$5D$ AEP [GWh]	Difference [%]
Optimal 74×10 MW	3478.158	3477.567	-0.017
Optimal 50×15 MW	3656.642	3656.412	-0.006
Down-Selected 50×15 MW	3652.696	3651.184	-0.041

A similar trend is observed for the optimal 15 MW layout, but this time the percentage of turbines placed on the perimeter has remained constant at 74%. The drop in AEP could potentially be linked to the fact that here only 17 turbines are placed on the side aligning with the predominant wind direction, compared to 18 in Test Case A.

Moreover, it is evident that as the minimum spacing is increased, the optimal 10 MW layout exhibits a larger AEP reduction than the optimal 50 MW layout. This could partially be because of various reasons such as the coarser grid provided to Smart-Start, but both configurations should be affected equally. The main reason for the greater reduction is that the stricter $5D$ constraint gives the optimizer less flexibility to adjust turbine positions, and the larger number of turbines in the 10 MW configuration amplifies this effect, resulting in a more pronounced difference.

This in turn leads to worse initial conditions for the down-selection, leading to an even greater drop in AEP as the minimum spacing is increased. The worse initial conditions are highlighted by the increase in potentially violating empty positions mentioned before, and by the fact that the percentage of turbines placed on the perimeter has dropped from 78% in Test Case A to 72% here.

The greater loss in AEP of the down-selected layout compared to the optimal 10 MW layout, when comparing Test Case C with A, explains why the untapped energy potential of the down-selected layout is more pronounced at the larger $5D$ spacing, as can be seen from Table 5.7

Table 5.7: Untapped energy potential of down-selected layouts for different spacings.

Test Case	Optimal 15 MW AEP [GWh]	Down-Selected 15 MW AEP [GWh]	Difference [%]
A	3656.642	3652.696	-0.108
C	3656.412	3651.184	-0.143

Overall, the results lead to the conclusion that the untapped energy potential of the down-selected layout increases as the minimum spacing constraint increases.

5.4 Discussion

The results of RQ2 demonstrate that down-selection is a highly effective strategy for adapting layouts of lower-rated turbines to higher-rated machines. Across all three test cases, the AEP of the down-selected layouts was almost indistinguishable from that of directly optimized layouts, with untapped energy potentials consistently below 0.15%. From an aerodynamic perspective, this shows that re-optimization with larger turbines provides only marginal gains in energy yield compared to down-selection from an existing smaller-turbine layout.

The influence of spacing constraints emerged as the most important factor in explaining the small residual differences. When the initial 10 MW layout already satisfied the spacing requirement of the larger turbines (Test Case B), the untapped potential dropped to 0.07%, the smallest of all cases. Conversely, when stricter 5D spacing was applied to both layouts (Test Case C), the untapped potential increased to 0.14%. This pattern is illustrated in Fig. 5.12, which shows that the gap between optimal and down-selected layouts remains consistently small but grows with stricter spacing requirements. These results suggest that down-selection is most effective when compatibility of spacing constraints is ensured from the outset.

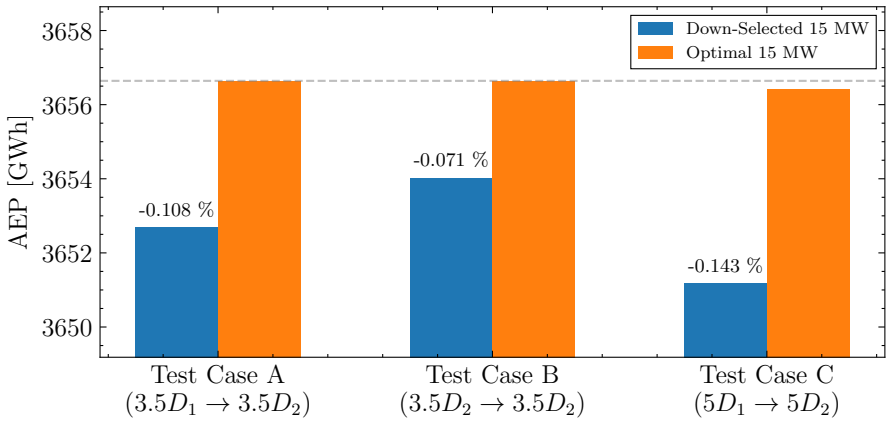


Figure 5.12: Comparison of optimal and down-selected layouts for the 15 MW turbine under different spacing constraints. The smallest gap occurs when the initial 10 MW layout already satisfied the spacing constraint of the larger turbine (Case B).

More broadly, the results imply that there may exist a threshold spacing beyond which down-selection would no longer be feasible, because the algorithm could not place all turbines without violating the constraint. The precise point of failure would depend on farm geometry, wind conditions, turbine rating, and the ratio of lower to higher-rated turbines. If such a threshold is encountered, several options exist: (1) design the lower-rated layout already using the spacing constraint of the larger turbines, (2) allow small positional adjustments to resolve spacing conflicts if financially acceptable, or (3) tolerate minor violations if they result only in acceptable increases in fatigue loading. These considerations show that down-selection should not be regarded as a rigid process but as a flexible design strategy that can be adapted depending on site-specific conditions and economic trade-offs.

In addition to spacing, the comparisons revealed differences in spatial distribution of turnines. Down-selected farms tended to retain more turbines along the perimeter of the site. While

this effect did not measurably influence AEP in the present study, it may have implications for secondary design aspects such as cabling length, support structure design, or visual footprint. These considerations highlight that the impact of down-selection should not be evaluated solely on aerodynamic performance but also in relation to other practical design drivers.

From a practical perspective, the advantages of down-selection extend beyond computational savings. Re-optimising a site for larger turbines may require new seabed surveys, updated bathymetric assessments, or additional environmental studies, all of which involve significant cost and time. The ability to adapt existing layouts with minimal loss in AEP therefore has direct economic value. At the same time, the method's effectiveness is conditional: anticipating spacing requirements during the initial design stage strongly improves the feasibility of later down-selection. This positions down-selection not as a reactive workaround but as part of a broader robust design strategy that balances current efficiency with adaptability to future turbine upgrades. Future work should also extend the analysis beyond spacing to include constraints such as bathymetry, cable routing, exclusion zones, and grid connection, in order to capture the full impact of down-selection.

CHAPTER 6

Conclusion and Outlook

6.1 Main Conclusions

This thesis addressed two research questions:

RQ1. Which optimization strategies are suitable for down-selection?
The ranking of candidate algorithms showed that Gradient-Based methods and Greedy Heuristics were the most suitable strategies for down-selection. Their strengths were complementary: Gradient-Based approaches scaled well to larger problems but risked convergence to local optima, while Greedy Heuristics required little parameter tuning and were straightforward to implement. Other approaches such as metaheuristics (e.g. GA, BPSO, VNS) and exact formulations (MILP, QIP) were less suitable in this context, either due to high sensitivity to parameter calibration, weaker scalability, or the need for linearisation of the objective function. These methods may still be valuable in different settings, but were not competitive under the criteria defined in this study. Comparative evaluation of the shortlisted strategies further confirmed that no single method dominates across all conditions. Gradient-Based optimisation consistently produced results with the shortest runtimes and scaled well with problem size, whereas Greedy Heuristics generally achieved the highest energy yields. Among them, Greedy Addition performed best in low-occupancy layouts, while Greedy Removal was preferable at high occupancy. These findings indicate that algorithm selection should be guided by the characteristics of the problem: Gradient-Based methods are most suitable when computational budget is the limiting factor, while Greedy Heuristics are preferable when maximising energy yield is the primary goal. Across all cases, careful parameter tuning was shown to be as critical as the algorithm itself, highlighting the importance of calibration before applying down-selection methods in practice.

RQ2. What is the impact of down-selection on the final wind farm layout?
The results demonstrated that down-selection can reproduce the performance of directly optimised layouts with only marginal aerodynamic losses. Across all test cases, the untapped energy potential of down-selected layouts was below 0.15%, confirming that down-selection is a practical approach for adapting designs to larger turbines. The performance gap was smallest when the initial layout already satisfied the spacing requirement of the larger turbines, and largest when both layouts were subject to a stricter spacing. This indicates that spacing constraints govern the effectiveness of down-selection: compatibility between initial and final constraints leads to nearly lossless transitions, whereas mismatched constraints reduce performance. In addition to energy yield, down-selection also influenced the spatial distribution of turbines, with more turbines retained along the perimeter of the site. While this had little effect on energy yield in the present study, it may have implications for secondary design drivers such as cabling layout, support structure design, or visual impact.

Taken together, the conclusions of this thesis show that down-selection is a viable and efficient strategy for handling turbine upgrades in offshore wind farm design. Its success, however,

depends on matching the strategy to the problem context and anticipating future spacing requirements during the initial design stage.

6.2 Future Work and Outlook

This thesis highlighted both the potential and the limitations of down-selection, and several directions for further work can be identified.

For optimisation strategies (RQ1), future research should explore a broader set of methods beyond those tested here. Exact approaches such as MILP or QIP could be valuable when linearisation of the objective function is acceptable, while metaheuristics like Genetic Algorithms may remain appealing when ease of implementation is a priority, especially since ready-to-use implementations are available in frameworks such as TOPFARM. Hybrid strategies that combine the scalability of Gradient-Based approaches with the solution quality of Heuristics may also be worth investigating. In addition, systematic studies of parameter tuning, including adaptive or problem-specific calibration, would help reduce unnecessary computational effort while preserving accuracy. Applying these strategies to larger and more realistic farm settings would further test their robustness.

For the down-selection process itself (RQ2), this thesis focused on spacing as the primary constraint. Future work should extend the analysis to include other practical drivers such as bathymetry, cable routing, exclusion zones, and grid connection requirements. These factors may interact with turbine distribution effects observed in down-selected layouts and could shift the trade-offs identified here. It would also be valuable to quantify the economic implications more explicitly, including potential savings from avoiding new seabed surveys or environmental assessments. Incorporating cost-of-energy metrics would provide a more holistic view of the benefits and limitations of down-selection.

Finally, the results emphasise that anticipating future turbine upgrades at the initial design stage is crucial for ensuring the feasibility of down-selection. Future work could therefore focus on embedding down-selection within a robust design framework that balances present-day efficiency with adaptability to evolving turbine technology. This would help developers design farms that remain both competitive and flexible over their lifetime.

APPENDIX A
Strategies for Down-Selection (RQ1)

A.1 Comparison of Candidate Strategies

Table A.1: Comparison of Mosetti Case III results (derived from Tables A.2 and A.3).

Research Paper	Approach	Improvement in Power [%]
Mosetti et al. [28]	GA	–
Turner et al. [44]	MILP	1.19
MirHassani and Yarahmadi [25]	MILP	2.95
Cazzaro and Pisinger [10]	VNS	0.99

Table A.2: Mosetti Case III with 15 WTs according to MirHassani and Yarahmadi [25]

Reference	Power [kW]
Mosetti et al. [28]	23,883
Turner et al. [44]	24,168
MirHassani and Yarahmadi [25]	24,587

Table A.3: Mosetti Case III with 15 WTs according to Cazzaro and Pisinger [10]

Reference	Power [MW]
Mosetti et al. [28]	5.05
Cazzaro and Pisinger [10]	5.10

A.2 Tuning of Gradient Based

A.2.1 Modification of Existence Bounds

When using the $[0, 1]$ bounds for the existence with the `SLSQP` driver, it is observed that some of the runs fail. The happens because the driver does not completely respect the boundaries when calculating the gradients close to the bounds. As such, it looks for existence values outside of $[0, 1]$ in the `PowerCtNDTabular`, thus failing.

A way to counter the problem is to decrease the bounds of the existence by a small ϵ, that is $[0 + \epsilon, 1 - \epsilon]$. After some trial and error, it is observed that a value of $\epsilon = 10^{-6}$ is sufficiently small for the problem to stop occurring.

Another approach to counter the problem would be to increase the values of `PowerCtNDTabular` by a similar amount, that is $[0-\epsilon, 1+\epsilon]$, while keeping the bounds given to `TopFarmProblem` as $[0, 1]$. This is approach does not seem to fail when the driver seeks for existence values outside of $[0, 1]$ in the `PowerCtNDTabular`. However, that implies that negative existence values can be used, and thus negative power and C_T. To avoid more unforeseen future problems, the first prevention method is selected.

A.2.2 Effect of Wind Speed Sampling on AEP Evaluation

A preliminary investigation was conducted to assess the influence of wind speed sampling on the annual energy production (AEP) estimates obtained with the `ExistenceWindTurbine` implementation in `TopFarm`. As shown in Table A.4, the AEP of a single turbine was compared using the standard `NormalWindTurbine` and the `ExistenceWindTurbine` for different numbers of wind speed samples (`wsp_samples`). The results show that `ExistenceWindTurbine` yields a lower AEP than the `NormalWindTurbine` when only a few samples are used, but the difference rapidly decreases as the number of samples increases. At 1000 or more samples, the discrepancy becomes negligible ($< 0.1\%$), making the results directly comparable to those obtained with the greedy heuristic.

Based on this observation, the optimization is performed with `ExistenceWindTurbine`, while the final AEP calculations are carried out using the `NormalWindTurbine` to ensure consistency across methods. The impact of the sampling resolution was further examined by testing `wsp_samples` values of 10, 100, 1000, and 10000 across four values of the ramping parameter q and ten multistarts. As illustrated in Fig. A.1, increasing the number of samples generally improves both the median and maximum AEP values, albeit at the cost of longer computational times. However, diminishing returns are observed beyond 1000 samples, which consistently achieve higher peak AEP than 100 samples while remaining computationally feasible. Therefore, 1000 wind speed samples are adopted as the default setting for this thesis.

Table A.4: Comparison of AEP values obtained with `Normal` and `Existence` wind turbines for different numbers of wind speed samples.

Wind Turbine	Wind Speed Samples	AEP [GWh]	Difference [%]
Normal	–	29.346	–
Existence	10	22.115	24.641
Existence	100	28.630	2.441
Existence	1000	29.321	0.085
Existence	10000	29.339	0.025

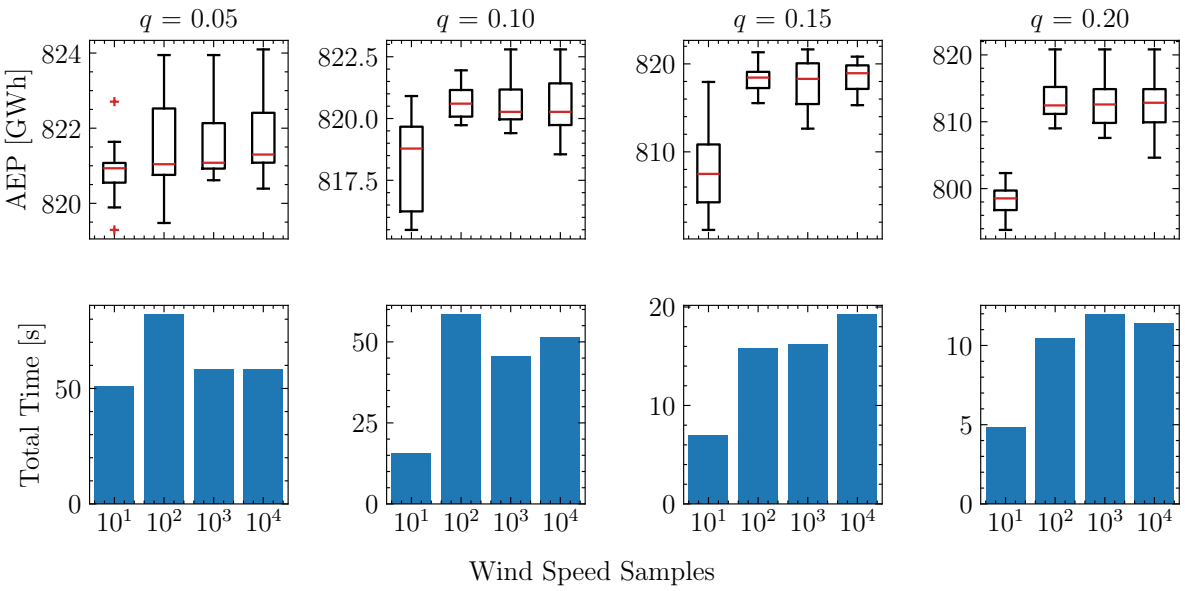


Figure A.1: Effect of `wsp_samples` on AEP and total computational time at different q values.

A.2.3 Finite Difference vs Automatic Differentiation

So far, finite differences were used to calculate all gradients used by the `SLSQP` solver. The performance could be improved by using analytical gradients and automatic differentiation (`autograd`) instead. For the sum constraint, the derivative with respect to the existence is just 1, while for the RAMP function the derivative is as seen in Eq. (A.1).

$$\frac{\mathrm{d}\widetilde{x}}{\mathrm{d}x} = \frac{1+q}{(1+q(1-x))^2} \tag{A.1}$$

An analytical expression for the derivative of the AEP function does not exist, thus the `autograd` can be used. To make the comparison between finite differencing and automatic differentiation fair, five different RAMP q factors are tested. For each of those, five optimizations are performed using random initial existences x_0, and the average AEP and optimization time are calculated. The five random seeds used are exactly the same between the two approaches. The results are shown in Fig. A.2, where it is clear that the average AEP is pretty similar between the two approaches, while the optimization time of the `autograd` is around five times lower. Additionally, when comparing individual runs, the difference in AEP in less than 0.5% for all 25 cases. Thus, the faster method will be used from this point forward.

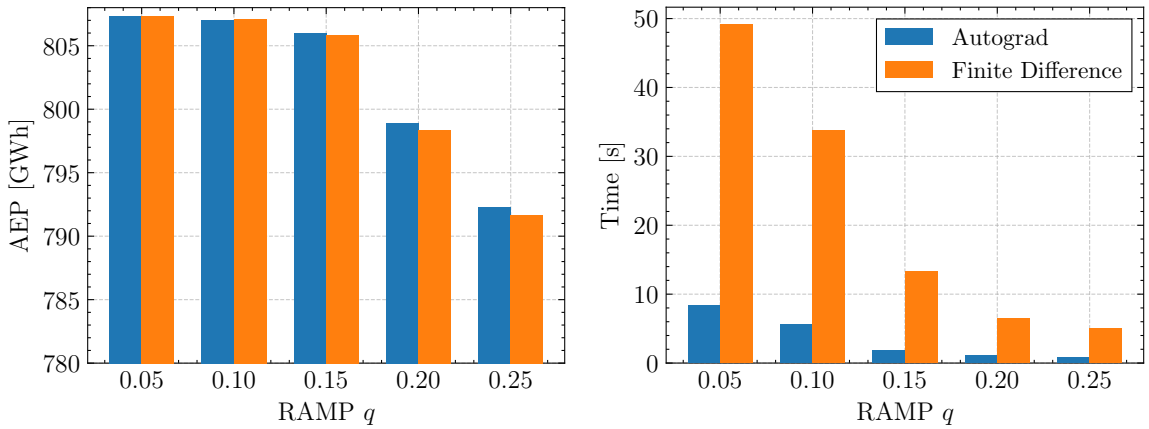


Figure A.2: Comparison of finite differencing with autograd in terms of AEP (left) and optimization time (right). Average of five runs per RAMP q factor.

A.3 Results

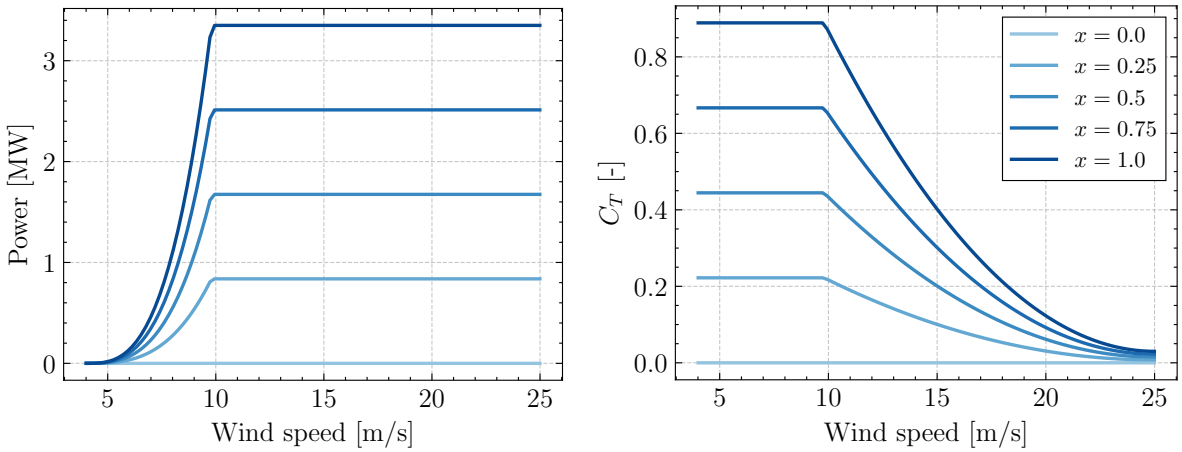


Figure A.3: Power and C_T curves for sample existence values of the `ExistenceWindTurbine` with variable C_T, based on the IEA-37 3.35 MW reference turbine.

To further explain the differing runtime trends between the Greedy Addition and Greedy Removal strategies, Fig. A.4 includes additional results from their respective variants that omit the refinement step (strategies A and E in Table 4.5).

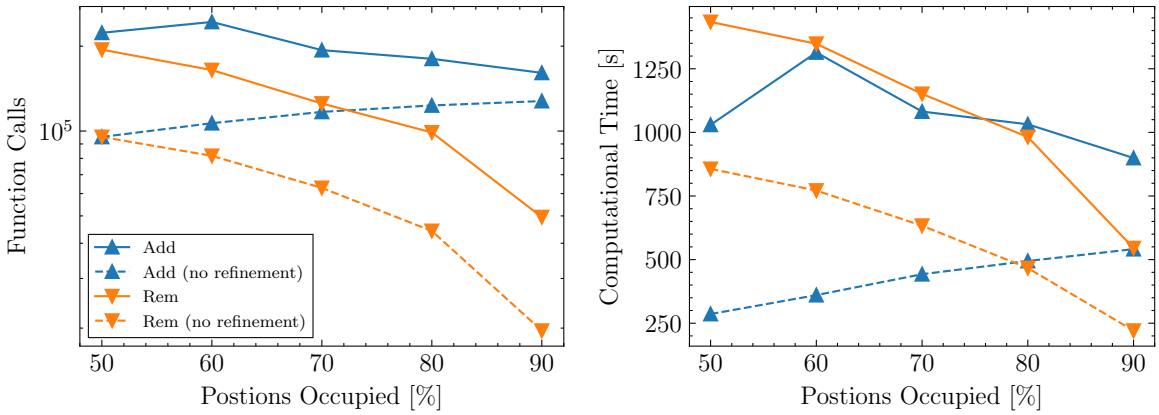


Figure A.4: Function calls (left) and computational time (right) with and without greedy refinement over varying occupancy levels, for Test Case A.

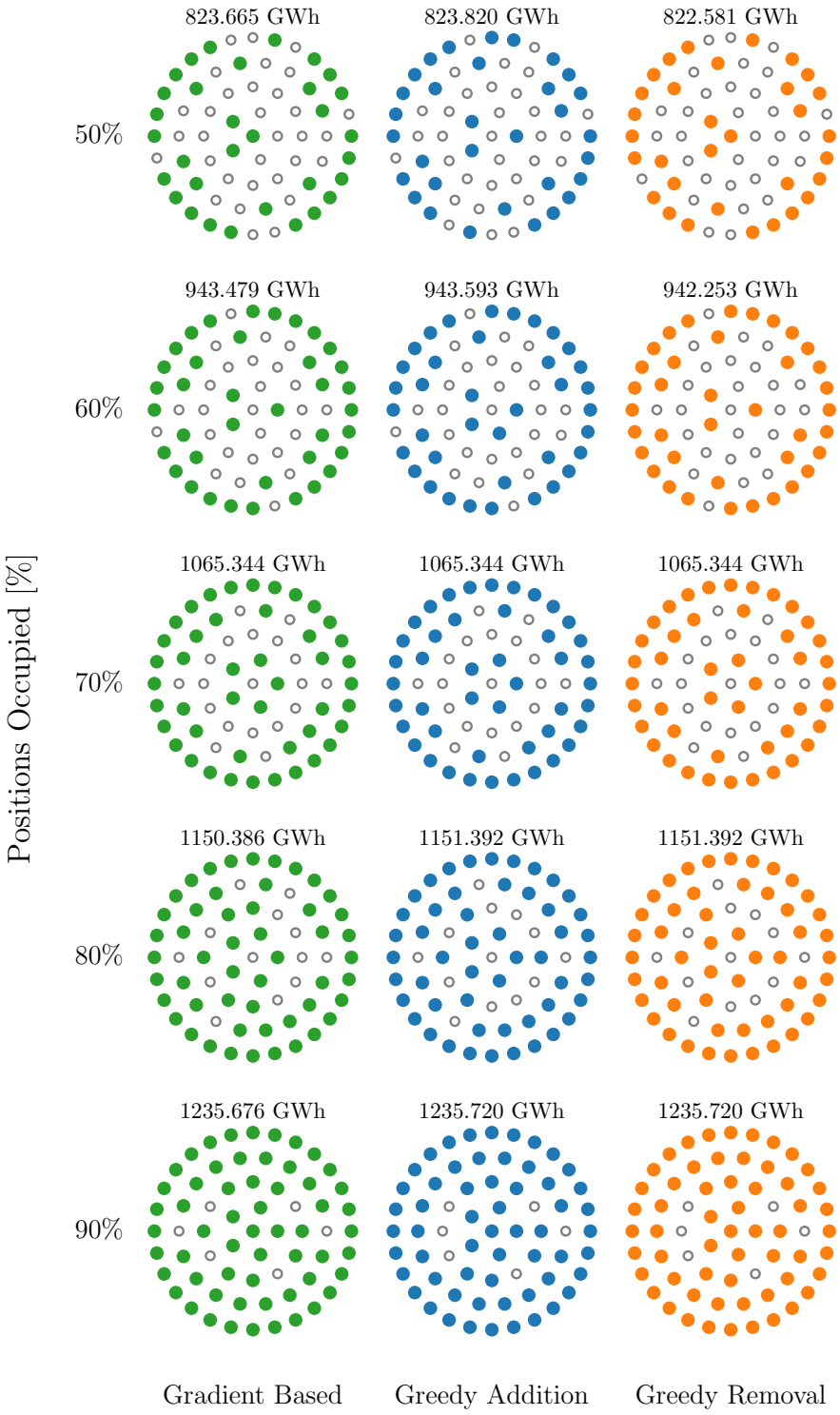


Figure A.5: Resulting layouts for Test Case A. Other cases are omitted for brevity.

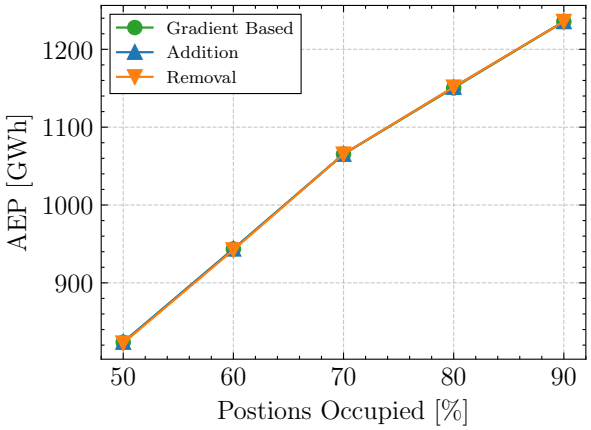


Figure A.6: Differences between methods become indiscernible as occupancy increases.

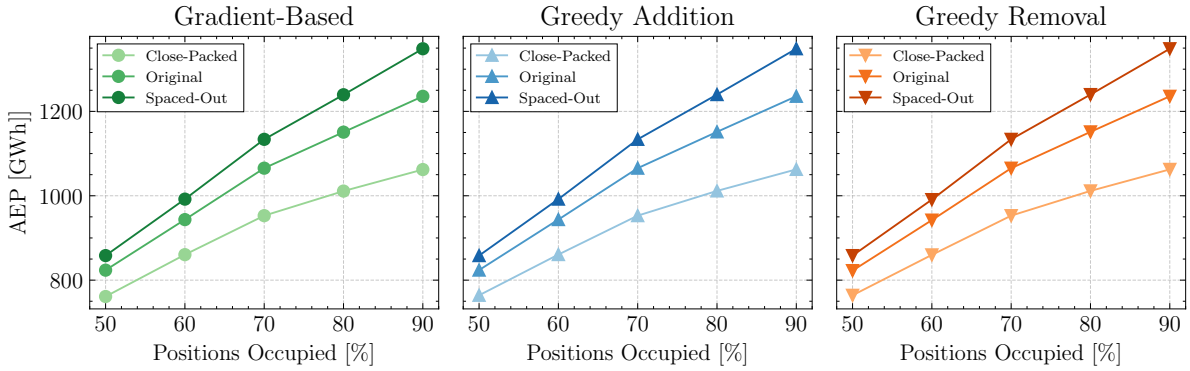


Figure A.7: AEP of each strategy separately. Greater spacing reduces wake overlap and increases AEP.

APPENDIX B

Impact of Down-Selection (RQ2)

B.1 Generation of Optimal Layouts

Table B.1: Summary of hexagonal grids for Test Case A. Other cases are omitted for brevity.

Hex Grid Target	Spacing [D]	Spacing [m]	Gridpoints	Turbines / Gridpoints [%]
74×10 MW	$3.5D_1$	693	439	16.9
50×15 MW	$3.5D_2$	840	297	16.8

B.2 Greedy Addition or Greedy Removal?

At first glance it might seem that unconstrained Greedy Addition is the best option. However, it managed to find only two layouts that respect the minimum distance constraint of $3.5D$, while both Addition and Removal that used the constraint respecting algorithm found a non-violating layout for each of the multistarts. Looking closer at the results of table, it is evident that Constrained Greedy Addition actually managed to find the highest non-violating AEP.

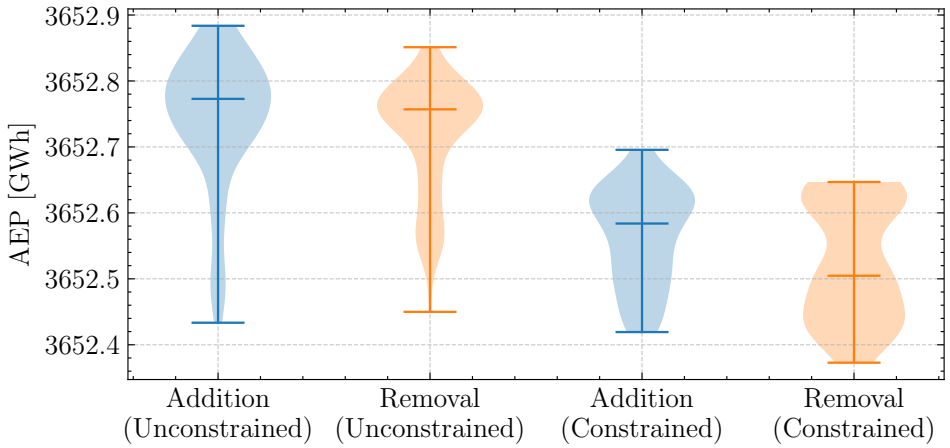


Figure B.1: AEP of unconstrained and constrained Greedy Addition and Removal.

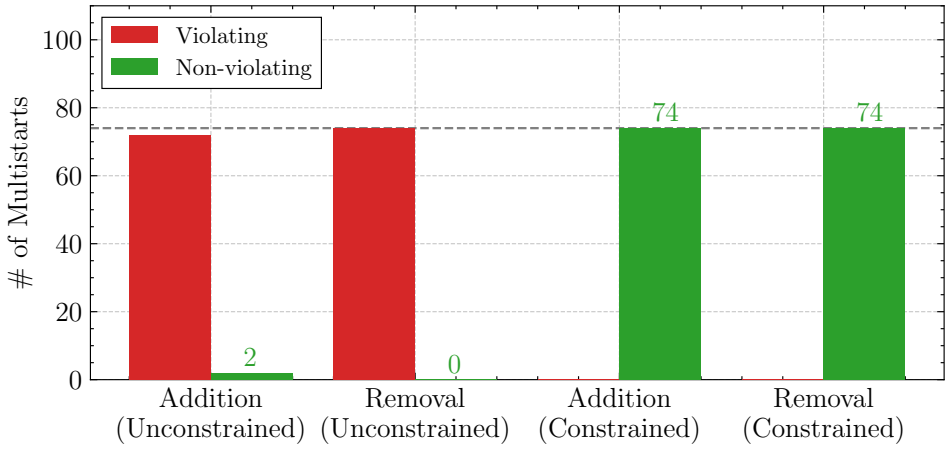


Figure B.2: Violations of unconstrained and constrained Greedy Addition and Removal.

Table B.2: Maximum AEP values and best feasible (non-violating) results for each algorithm.

Algorithm	Constrainted	Max AEP [GWh]	Max Feasible AEP [GWh]
Greedy Addition	✗	3652.884	3652.577
Greedy Removal	✗	3652.851	–
Greedy Addition	✓	3652.696	**3652.696**
Greedy Removal	✓	3652.647	3652.647

APPENDIX C

General

C.1 Use of Artificial Intelligence

Artificial intelligence tools were used throughout the thesis to support both writing and technical tasks. ChatGPT[1] assisted with refining scientific language, improving structure and clarity in written sections, and troubleshooting or generating Python code. AI-powered platforms such as Litmaps[2] and Connected Papers[3] were also used to explore relevant literature and map connections between studies. All AI-generated content was carefully reviewed, verified, and edited by the author to ensure accuracy, relevance, and academic integrity.

[1] https://chatgpt.com/
[2] https://app.litmaps.com/
[3] https://www.connectedpapers.com/

Bibliography

[1] BVG Associates. *Guide to an offshore wind farm*. en. Technical report. The Crown Estate, Offshore Renewable Energy Catapult, 2019. URL: https://guidetoanoffshorewindfarm.com/.

[2] Charles Audet and Warren Hare. *Derivative-Free and Blackbox Optimization*. en. Springer Series in Operations Research and Financial Engineering. Cham: Springer International Publishing, 2017. ISBN: 978-3-319-68912-8 978-3-319-68913-5. DOI: 10.1007/978-3-319-68913-5. URL: http://link.springer.com/10.1007/978-3-319-68913-5 (visited on February 16, 2025).

[3] F. Azlan et al. "Review on optimisation methods of wind farm array under three classical wind condition problems." en. In: *Renewable and Sustainable Energy Reviews* 135 (January 2021), page 110047. ISSN: 13640321. DOI: 10.1016/j.rser.2020.110047. URL: https://linkinghub.elsevier.com/retrieve/pii/S1364032120303385 (visited on December 30, 2024).

[4] Nicholas F. Baker et al. "Best Practices for Wake Model and Optimization Algorithm Selection in Wind Farm Layout Optimization." en. In: *AIAA Scitech 2019 Forum*. San Diego, California: American Institute of Aeronautics and Astronautics, January 2019. ISBN: 978-1-62410-578-4. DOI: 10.2514/6.2019-0540. URL: https://arc.aiaa.org/doi/10.2514/6.2019-0540 (visited on December 26, 2024).

[5] Nicholas F. Baker et al. *byuflowlab/iea37-wflo-casestudies: Initial Realize for Citation*. December 2021. DOI: 10.5281/zenodo.5809681. URL: https://zenodo.org/records/5809681 (visited on May 17, 2025).

[6] R. J. Barthelmie et al. "Comparison of Wake Model Simulations with Offshore Wind Turbine Wake Profiles Measured by Sodar." en. In: (July 2006). Section: Journal of Atmospheric and Oceanic Technology. DOI: 10.1175/JTECH1886.1. URL: https://journals-ametsoc-org.proxy.findit.cvt.dk/view/journals/atot/23/7/jtech1886_1.xml (visited on August 31, 2025).

[7] Majid Bastankhah and Fernando Porté-Agel. "A new analytical model for wind-turbine wakes." In: *Renewable Energy*. Special issue on aerodynamics of offshore wind energy systems and wakes 70 (October 2014), pages 116–123. ISSN: 0960-1481. DOI: 10.1016/j.renene.2014.01.002. URL: https://www.sciencedirect.com/science/article/pii/S0960148114000317 (visited on August 31, 2025).

[8] Majid Bastankhah and Fernando Porté-Agel. "Experimental and theoretical study of wind turbine wakes in yawed conditions." en. In: *Journal of Fluid Mechanics* 806 (November 2016). Publisher: Cambridge University Press (CUP), pages 506–541. ISSN: 0022-1120, 1469-7645. DOI: 10.1017/jfm.2016.595. URL: https://www.cambridge.org/core/product/identifier/S0022112016005954/type/journal_article (visited on May 17, 2025).

[9] Pietro Bortolotti et al. *IEA Wind TCP Task 37: Systems Engineering in Wind Energy - WP2.1 Reference Wind Turbines*. en. Technical report. NREL.

[10] Davide Cazzaro and David Pisinger. "Variable neighborhood search for large offshore wind farm layout optimization." en. In: *Computers & Operations Research* 138 (February 2022), page 105588. ISSN: 03050548. DOI: 10.1016/j.cor.2021.105588. URL: https://linkinghub.elsevier.com/retrieve/pii/S0305054821003130 (visited on January 21, 2025).

[11] K. Chen et al. "Wind turbine layout optimization with multiple hub height wind turbines using greedy algorithm." en. In: *Renewable Energy* 96 (October 2016), pages 676–686. ISSN: 09601481. DOI: 10.1016/j.renene.2016.05.018. URL: https://linkinghub.elsevier.com/retrieve/pii/S0960148116304293 (visited on January 29, 2025).

[12] K. Chen et al. "Wind turbine positioning optimization of wind farm using greedy algorithm." en. In: *Journal of Renewable and Sustainable Energy* 5.2 (March 2013), page 023128. ISSN: 1941-7012. DOI: 10.1063/1.4800194. URL: https://pubs.aip.org/jrse/article/5/2/023128/930101/Wind-turbine-positioning-optimization-of-wind-farm (visited on February 10, 2025).

[13] A. Crespo, J. Hernández, and S. Frandsen. "Survey of modelling methods for wind turbine wakes and wind farms." en. In: *Wind Energy* 2.1 (January 1999), pages 1–24. ISSN: 1095-4244, 1099-1824. DOI: 10.1002/(SICI)1099-1824(199901/03)2:1<1::AID-WE16>3.0.CO;2-7. URL: https://onlinelibrary.wiley.com/doi/10.1002/(SICI)1099-1824(199901/03)2:1<1::AID-WE16>3.0.CO;2-7 (visited on August 24, 2025).

[14] Ju Feng and Wen Zhong Shen. "Solving the wind farm layout optimization problem using random search algorithm." en. In: *Renewable Energy* 78 (June 2015), pages 182–192. ISSN: 09601481. DOI: 10.1016/j.renene.2015.01.005. URL: https://linkinghub.elsevier.com/retrieve/pii/S0960148115000129 (visited on December 24, 2024).

[15] Evan Gaertner et al. *Definition of the IEA 15-Megawatt Offshore Reference Wind*. en. Technical report. NREL, 2020. URL: https://www.nrel.gov/docs/fy20osti/75698.pdf.

[16] Xiaoxia Gao et al. "Comparisons of the accuracy of different wake models in wind farm layout optimization." en. In: *Energy Exploration & Exploitation* 38.5 (September 2020), pages 1725–1741. ISSN: 0144-5987, 2048-4054. DOI: 10.1177/0144598720942852. URL: https://journals.sagepub.com/doi/10.1177/0144598720942852 (visited on August 31, 2025).

[17] Dolf Gielen et al. "The role of renewable energy in the global energy transformation." en. In: *Energy Strategy Reviews* 24 (April 2019), pages 38–50. ISSN: 2211467X. DOI: 10.1016/j.esr.2019.01.006. URL: https://linkinghub.elsevier.com/retrieve/pii/S2211467X19300082 (visited on January 22, 2025).

[18] José F. Herbert-Acero et al. "A Review of Methodological Approaches for the Design and Optimization of Wind Farms." en. In: *Energies* 7.11 (November 2014). Number: 11 Publisher: Multidisciplinary Digital Publishing Institute, pages 6930–7016. ISSN: 1996-1073. DOI: 10.3390/en7116930. URL: https://www.mdpi.com/1996-1073/7/11/6930 (visited on December 24, 2024).

[19] Peng Hou et al. "Optimized Placement of Wind Turbines in Large-Scale Offshore Wind Farm Using Particle Swarm Optimization Algorithm." In: *IEEE Transactions on Sustainable Energy* 6.4 (October 2015). Conference Name: IEEE Transactions on Sustainable Energy, pages 1272–1282. ISSN: 1949-3037. DOI: 10.1109/TSTE.2015.2429912. URL: https://ieeexplore.ieee.org/document/7113903/?arnumber=7113903 (visited on January 28, 2025).

[20] N. O. Jensen. "A note on wind generator interaction." en. In: *Risø National Laboratory* Risø-M No. 2411 (1983).

[21] Samuel Kainz et al. *IEA-Wind 740-10MW Reference Offshore Wind Plants*. en. Technical report. NREL, 2024. URL: https://github.com/IEAWindTask37/IEA-Wind-740-10-ROWP.

[22] I Katic, J Højstrup, and N O Jensen. "A simple model for cluster efficiency." en. In: (1986).

[23] James F. Manwell, John G. McGowan, and Anthony L. Rogers. *Wind Energy Explained: Theory, Design and Application*. en. 2nd edition. John Wiley & Sons. ISBN: 978-0-470-01500-1.

[24] Joaquim R. R. A. Martins and Andrew Ning. *Engineering Design Optimization*. en. 1st edition. Cambridge University Press, November 2021. ISBN: 978-1-108-98064-7 978-1-108-83341-7. DOI: 10.1017/9781108980647. URL: https://www.cambridge.org/core/product/identifier/9781108980647/type/book (visited on December 28, 2024).

[25] S.A. MirHassani and A. Yarahmadi. "Wind farm layout optimization under uncertainty." en. In: *Renewable Energy* 107 (July 2017), pages 288–297. ISSN: 09601481. DOI: 10.1016/j.renene.2017.01.063. URL: https://linkinghub.elsevier.com/retrieve/pii/S0960148117300733 (visited on December 31, 2024).

[26] N. Mladenović and P. Hansen. "Variable neighborhood search." In: *Computers & Operations Research* 24.11 (November 1997), pages 1097–1100. ISSN: 0305-0548. DOI: 10.1016/S0305-0548(97)00031-2. URL: https://www.sciencedirect.com/science/article/pii/S0305054897000312 (visited on August 27, 2025).

[27] Mojtaba Ahmadieh Khanesar, Mohammad Teshnehlab, and Mahdi Aliyari Shoorehdeli. "A novel binary particle swarm optimization." en. In: *2007 Mediterranean Conference on Control & Automation*. Athens, Greece: IEEE, June 2007, pages 1–6. ISBN: 978-1-4244-1281-5 978-1-4244-1282-2. DOI: 10.1109/MED.2007.4433821. URL: http://ieeexplore.ieee.org/document/4433821/ (visited on August 27, 2025).

[28] G. Mosetti, C. Poloni, and B. Diviacco. "Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm." en. In: *Journal of Wind Engineering and Industrial Aerodynamics* 51.1 (January 1994), pages 105–116. ISSN: 01676105. DOI: 10.1016/0167-6105(94)90080-9. URL: https://linkinghub.elsevier.com/retrieve/pii/0167610594900809 (visited on December 24, 2024).

[29] Mads M. Pedersen et al. *PyWake 2.5.0: An open-source wind farm simulation tool*. DTU Wind, Technical University of Denmark. February 2023. URL: https://gitlab.windenergy.dtu.dk/TOPFARM/PyWake (visited on January 27, 2025).

[30] S Sanchez Perez-Moreno et al. "Multidisciplinary design analysis and optimisation of a reference offshore wind plant." en. In: *Journal of Physics: Conference Series* 1037 (June 2018), page 042004. ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1037/4/042004. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1037/4/042004 (visited on December 24, 2024).

[31] Nicolò Pollini. "A two-step topology and layout wind farm optimization approach." en. In: *Journal of Physics: Conference Series* 2767.9 (June 2024). Publisher: IOP Publishing, page 092003. ISSN: 1742-6596. DOI: 10.1088/1742-6596/2767/9/092003. URL: https://dx.doi.org/10.1088/1742-6596/2767/9/092003 (visited on December 24, 2024).

[32] Nicolò Pollini. "Topology optimization of wind farm layouts." en. In: *Renewable Energy* 195 (August 2022), pages 1015–1027. ISSN: 09601481. DOI: 10.1016/j.renene.2022.06.019. URL: https://linkinghub.elsevier.com/retrieve/pii/S0960148122008503 (visited on December 24, 2024).

[33] Bryony L Du Pont and Jonathan Cagan. "An Extended Pattern Search Approach to Wind Farm Layout Optimization." en. In: ().

[34] Juan-Andrés Pérez-Rúa and Nicolaos A. Cutululis. "Modelling the Wind Farm Layout Optimization Problem Using Integer Programming." In: *2023 IEEE Conference on Control Technology and Applications (CCTA)*. ISSN: 2768-0770. August 2023, pages 566–573. DOI: 10.1109/CCTA54093.2023.10252646. URL: https://ieeexplore.ieee.org/document/10252646/?arnumber=10252646 (visited on December 24, 2024).

[35] Juan-Andrés Pérez-Rúa, Mathias Stolpe, and Nicolaos Antonio Cutululis. "A neighborhood search integer programming approach for wind farm layout optimization." English. In: *Wind Energy Science* 8.9 (September 2023). Publisher: Copernicus GmbH, pages 1453–1473. ISSN: 2366-7443. DOI: 10.5194/wes-8-1453-2023. URL: https://wes.copernicus.org/articles/8/1453/2023/ (visited on February 13, 2025).

[36] Erik Quaeghebeur, René Bos, and Michiel B. Zaaijer. "Wind farm layout optimization using pseudo-gradients." English. In: *Wind Energy Science* 6.3 (June 2021). Publisher: Copernicus GmbH, pages 815–839. ISSN: 2366-7443. DOI: 10.5194/wes-6-815-2021. URL: https://wes.copernicus.org/articles/6/815/2021/ (visited on December 24, 2024).

[37] Julian Quick et al. "Stochastic gradient descent for wind farm optimization." English. In: *Wind Energy Science* 8.8 (August 2023). Publisher: Copernicus GmbH, pages 1235–1250. ISSN: 2366-7443. DOI: 10.5194/wes-8-1235-2023. URL: https://wes.copernicus.org/articles/8/1235/2023/ (visited on December 24, 2024).

[38] M. Janga Reddy and D. Nagesh Kumar. "Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review." en. In: *H2Open Journal* 3.1 (January 2020), pages 135–188. ISSN: 2616-6518. DOI: 10.2166/h2oj.2020.128. URL: https://iwaponline.com/h2open/article/3/1/135/74697/Evolutionary-algorithms-swarm-intelligence-methods (visited on January 23, 2025).

[39] Riccardo Riva et al. *Welcome to TOPFARM*. URL: https://topfarm.pages.windenergy.dtu.dk/TopFarm2/index.html (visited on January 27, 2025).

[40] Andrew P. J. Stanley and Andrew Ning. "Massive simplification of the wind farm layout optimization problem." English. In: *Wind Energy Science* 4.4 (December 2019). Publisher: Copernicus GmbH, pages 663–676. ISSN: 2366-7443. DOI: 10.5194/wes-4-663-2019. URL: https://wes.copernicus.org/articles/4/663/2019/ (visited on December 24, 2024).

[41] Andrew P. J. Stanley et al. "Objective and algorithm considerations when optimizing the number and placement of turbines in a wind power plant." English. In: *Wind Energy Science* 6.5 (September 2021). Publisher: Copernicus GmbH, pages 1143–1167. ISSN: 2366-7443. DOI: 10.5194/wes-6-1143-2021. URL: https://wes.copernicus.org/articles/6/1143/2021/ (visited on December 25, 2024).

[42] J J Thomas and A Ning. "A method for reducing multi-modality in the wind farm layout optimization problem." en. In: *Journal of Physics: Conference Series* 1037 (June 2018), page 042012. ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1037/4/042012. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1037/4/042012 (visited on December 24, 2024).

[43] Jared J. Thomas et al. "A comparison of eight optimization methods applied to a wind farm layout optimization problem." en. In: *Wind Energy Science* 8.5 (June 2023), pages 865–891. ISSN: 2366-7451. DOI: 10.5194/wes-8-865-2023. URL: https://wes.copernicus.org/articles/8/865/2023/ (visited on December 24, 2024).

[44] S.D.O. Turner et al. "A new mathematical programming approach to optimize wind farm layouts." en. In: *Renewable Energy* 63 (March 2014), pages 674–680. ISSN: 09601481. DOI: 10.1016/j.renene.2013.10.023. URL: https://linkinghub.elsevier.com/retrieve/pii/S0960148113005545 (visited on December 31, 2024).

[45] UNFCCC. *The Paris Agreement*. URL: https://unfccc.int/documents/184656 (visited on January 22, 2025).

[46] Rafael Valotta Rodrigues et al. "Speeding up large-wind-farm layout optimization using gradients, parallelization, and a heuristic algorithm for the initial layout." English. In: *Wind Energy Science* 9.2 (February 2024). Publisher: Copernicus GmbH, pages 321–341. ISSN: 2366-7443. DOI: 10.5194/wes-9-321-2024. URL: https://wes.copernicus.org/articles/9/321/2024/ (visited on December 24, 2024).

[47] Rebecca Williams and Feng Zhao. *Global Offshore Wind Report 2024*. en. Technical report. Global Wind Energy Council, 2024.