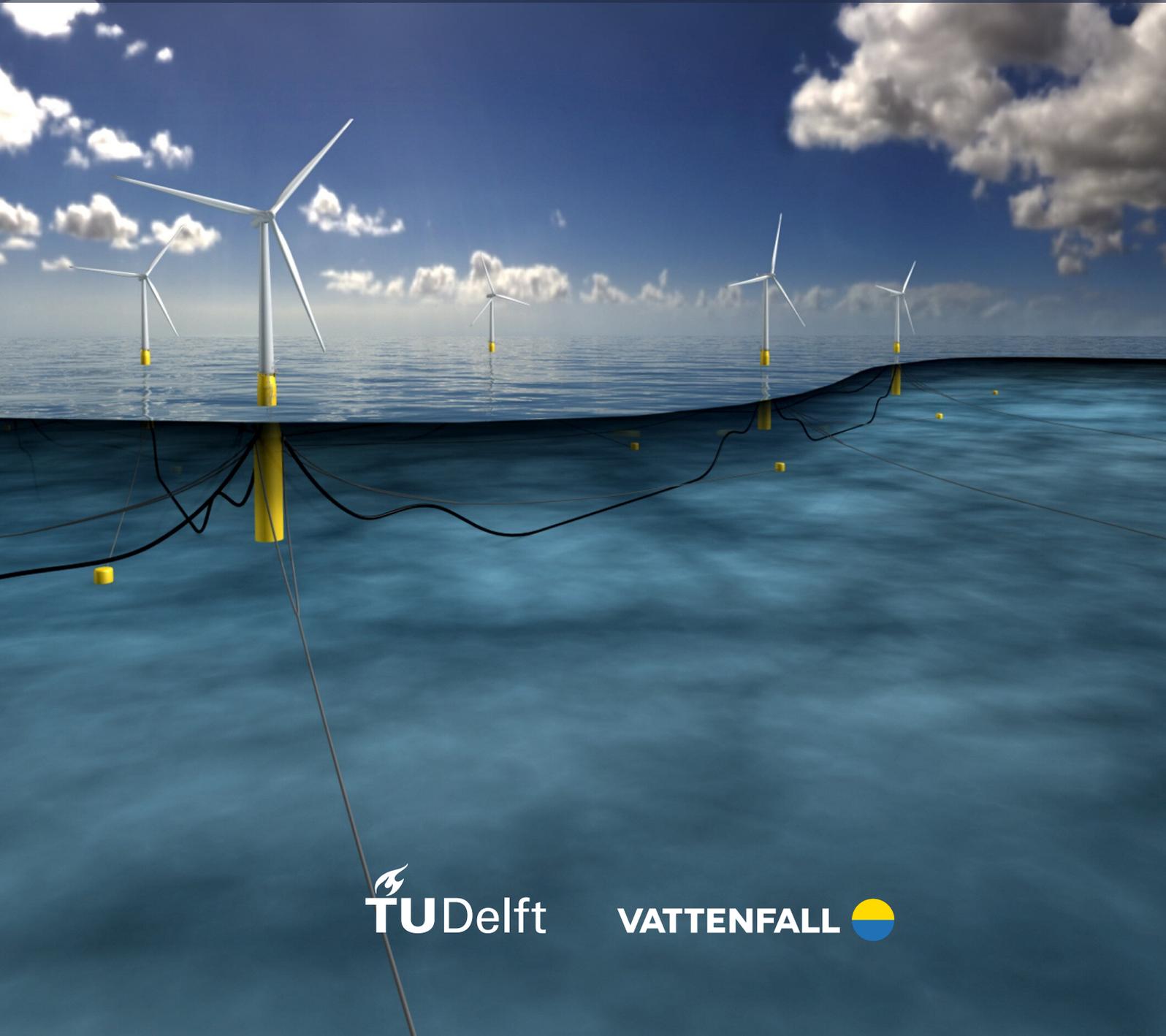


Wind Farm Cable Routing Optimization for Floating Offshore Wind Farms

Finding feasible solutions for loop topologies with mooring system constraints

V.H.H. (Veerle) Witkop



Wind Farm Cable Routing Optimization for Floating Offshore Wind Farms

Finding feasible solutions for loop topologies with
mooring system constraints

by

V.H.H. (Veerle) Witkop

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on March 13th, 2025.

Student number	4690699
Thesis committee	Dr. ir. M.B. Zaayer Dr. M.P. Kidd Dr. J. Iori Dr. G. Lavidas
Project duration	April, 2024 - March, 2025



Preface

This thesis marks the completion of my academic journey with a Master's degree in Sustainable Energy Technology at Delft University of Technology. It was conducted in collaboration with Vattenfall, a leading energy company committed to accelerating the transition to a sustainable future free from fossil fuels. Through this collaboration, I had the opportunity to work on the optimization of cable routing for floating offshore wind farms, a topic that aligns with my passion for wind energy and my ambition to improve my programming skills.

The global energy transition is progressing rapidly, with wind energy expected to play a crucial role in the shift towards a more sustainable energy mix. While space for onshore wind is limited and offshore wind is expanding, deep-water locations pose new technical and economic challenges. Floating offshore wind turbines provide a promising solution, but their electrical infrastructure, particularly the routing of cables, remains a complex optimization problem due to the need for a mooring system. This thesis aims to contribute to this evolving field by developing a framework to optimize cable routing while considering the spatial and technical constraints of floating turbines.

When I started this thesis, my knowledge of optimization and programming was minimal, and I often questioned whether I had taken on too great of a challenge by wanting to push myself and learn something new. It felt like standing at the base of a steep hill, unsure if I had the right gear, or any gear at all, to make the climb. However, as with any difficult ascent, the journey became easier with the right support.

First and foremost, I would like to express my deepest gratitude to my supervising professor and chair of my thesis committee, Michiel Zaaijer, for his invaluable guidance, patience, and encouragement. His expertise and constructive feedback provided a clear path forward, keeping me from straying too far off course. More than just guiding me through the technical aspects, he was always willing to lend a listening ear when I struggled with the mental challenges of writing a thesis. What once felt like an intimidating task turned into an experience I genuinely enjoyed, thanks to him, and I often wished it was not coming to an end.

I am also deeply grateful to my company supervisor, Martin Kidd, for giving me the opportunity to conduct my research at Vattenfall and for believing in my potential, even when I admitted my limited experience with programming and optimization. I am incredibly grateful for his time, insightful discussions, and technical expertise.

Furthermore, I sincerely appreciate Jenna Iori and George Lavidas for taking the time to review my thesis and for being part of my thesis committee.

Lastly, to my family and friends, thank you for being my foundation when I felt lost, for listening to my endless complaints about optimization models, and for reminding me that every uphill struggle leads to an incredible view at the top. I could not have done this without your encouragement and support. Especially Pauline, thank you for spending hours and hours, days and evenings with me in the library, motivating me and being my wall to talk to and share all my thoughts with, even though you often had no idea what I was talking about. I wish everyone a friend like you, and I am very confident you will excel in your thesis.

This thesis is not just the cherry on top of my academic journey; it is the summit of a challenging climb. Although the path was steep at times, I would not have reached the top without the incredible people who walked beside me.

*V.H.H. (Veerle) Witkop
Amsterdam, March 2025*

Summary

Offshore wind energy is set to become a key contributor to the global energy transition, with projections indicating that it could supply up to 25% of the total energy mix by 2050 [13]. While bottom-fixed offshore wind turbines have been the dominant technology, their deployment is limited to shallow waters, restricting expansion in regions with deep coastal waters such as Japan, the United States, and parts of Europe [20]. Floating offshore wind turbines offer a solution to this challenge, enabling the development of wind farms in deeper waters, often further from shore, where wind conditions are more stable and favorable. Additionally, floating wind farms reduce issues related to land scarcity, noise, and visual pollution. However, they introduce additional technical and economic complexities, particularly in cable routing, which is crucial for cost efficiency and reliability.

Cable routing in floating wind farms differs significantly from that of bottom-fixed wind farms due to the presence of mooring systems and the need for dynamic cables [56]. Mooring lines create spatial constraints that cables must avoid, while dynamic cables require additional length to accommodate the movement of floating platforms [50], [29]. These factors inherently result in longer and more complex cable layouts, increasing capital expenditures [33], [11]. While extensive research exists on cable routing optimization for bottom-fixed wind farms, studies on floating wind farms remain limited. Most existing research focuses on cost modeling and branch or radial topologies but does not fully integrate mooring constraints or explore the challenges posed by loop topologies, which are the preferred cable topology for floating offshore wind farms [28], [29], [48], [39].

This thesis addresses this gap by developing an optimization framework that minimizes inter-array cable length while ensuring compliance with mooring system constraints and loop topology requirements. By incorporating these constraints, the research contributes to improving the feasibility and economic viability of floating wind farms. Conducted in collaboration with Vattenfall, the study ensures the findings are applicable to real-world offshore wind projects.

An exact optimization approach using a Mixed-Integer Linear Programming model is chosen due to its ability to rigorously enforce technical and spatial constraints while optimizing cable layout. The methodology is structured into three phases, preprocessing, optimization, and postprocessing, to improve computational efficiency while maintaining feasibility under realistic conditions. The preprocessing phase defines feasible direct connections between turbines and the offshore substation while ensuring that cables do not cross mooring lines. To manage computational complexity, the problem is decomposed into smaller subproblems using a sweep-based clustering algorithm. Cluster boundaries are established using a concave hull approach, ensuring that shortest paths remain within defined areas and eliminating interdependencies between clusters. The optimization phase then determines the optimal cable layout, enforcing loop topology, preventing crossings, and balancing electrical loads. The postprocessing phase evaluates the optimized layout against industry standards, particularly the minimum clearance distance between inter-array cables and mooring systems. If violations occur, adjustments such as modifying turbine positions or reclusterings are applied to improve feasibility.

The results include a case study, an analysis of minimum distance violations, and an evaluation of the model's limitations. The case study applies the framework to the London Array wind farm, showing that the optimized layout closely aligns with the original design while reducing total cable length by approximately 1.1% for the differently routed clusters. However, when adapting the layout for a floating wind farm, several clearance violations occur due to mooring system constraints. Scaling the wind farm layout by a factor of 1.25 resolves most issues, but one remaining violation is addressed through a minor turbine position adjustment. These findings confirm that the optimization framework functions as intended while also demonstrating how it can be applied in real-world scenarios. Furthermore, they illustrate that small geometric modifications can significantly improve feasibility while preserving optimal routing. An analysis of minimum distance violations indicates that increasing the number of loops and reducing loop size leads to more routing conflicts, particularly in inter-cluster passages where cables from different clusters converge. The study also reveals that wind farms with centrally positioned offshore substations experience significantly fewer violations compared to those with substations placed at the field boundary. This suggests that careful substation placement can

mitigate congestion and improve routing feasibility. Computational efficiency is influenced by clustering decisions and the inclusion of subtour elimination constraints. As cluster size increases, the runtime grows factorially, making smaller clusters preferable for iterative model testing. However, larger clusters offer more routing flexibility and reduce total cable length. The study highlights that clustering feasibility is highly dependent on the starting index, particularly for wind farms with substations positioned at the boundary, where only one specific starting index ensures compact and well-distributed clusters. In contrast, centrally positioned substations allow for more flexible clustering configurations. The formation of cluster boundaries is another key factor affecting routing feasibility. The threshold length parameter defining concave hull boundaries must be carefully chosen to balance routing space with constraint enforcement. A threshold length of 870 meters is recommended based on empirical testing, as it minimizes infeasibility while preserving routing flexibility.

Future research should focus on developing a heuristic to optimize the selection of the starting index, eliminating the need for manual intervention. Alternative clustering methods that do not rely on radial sweep algorithms could be explored, particularly for wind farms with substations positioned at the boundary. Further improvements could be achieved by refining boundary generation through a metaheuristic approach that dynamically adjusts boundaries based on turbine distribution while minimizing inter-cluster space and reducing infeasibility risks. Additionally, the study identifies that passage width definitions could be refined to include anchor-to-mooring distances rather than only anchor-to-anchor spacing, improving the accuracy of clearance assessments. The optimization of mooring system orientations may also enhance available routing space, with minor variations in mooring line placement potentially reducing clearance violations without major layout modifications. Furthermore, computational efficiency could be improved by exploring alternative formulations for subtour elimination constraints. Metaheuristic approaches, which combine exact optimization with metaheuristic techniques, could provide scalable solutions for large-scale wind farms. A two-stage optimization process, where a heuristic generates an initial feasible solution followed by refinement using an exact model, may offer a balance between computational efficiency and solution quality.

By addressing these challenges, future research can further enhance the applicability of this optimization framework to large-scale floating wind farms. This study contributes to the field by presenting a methodology that integrates mooring constraints and loop topology into cable routing optimization, providing insights into the spatial constraints that must be considered in floating wind farm design. The proposed framework offers a structured and computationally feasible approach to inter-array cable routing, advancing both the economic and technical feasibility of floating offshore wind technology.

Table of Contents

Preface	i
Summary	ii
1 Introduction	1
1.1 The Rise of Offshore Wind Energy	1
1.2 Floating Offshore Wind	2
1.3 The Wind Farm Layout Optimization Problem	3
1.4 The Wind Farm Cable Routing Problem	3
1.5 Cable Topologies	3
1.6 Research Problem and Academic Knowledge Gap	6
1.7 Research Objective and Research Questions	8
1.8 Thesis Outline	9
2 Methodology	10
2.1 Problem Formulation	10
2.2 Solution Approach	10
2.2.1 Preprocessing	13
2.2.2 Optimization	13
2.2.3 Postprocessing	13
2.3 Preprocessing	13
2.3.1 Direct Connections	14
2.3.2 Shortest Path Formation	15
2.3.3 Turbine Clustering	18
2.3.4 Cluster Boundary Creation	20
2.3.5 Cluster-Specific Shortest Path Formation	23
2.3.6 Shortest Path Crossing Detection	24
2.4 Optimization	26
2.4.1 Mixed Integer Linear Programming Model	26
2.4.2 Miller-Tucker-Zemlin Subtour Elimination Constraint	27
2.5 Postprocessing	28
3 Results and Discussion	32
3.1 Site Description	32
3.2 Case Study	34
3.2.1 Optimization of London Array Offshore Wind Farm	34
3.2.2 London Array Offshore Wind Farm as a Floating Offshore Wind Farm	35
3.2.3 Modification and Refinement of London Array Offshore Wind Farm as Floating Offshore Wind Farm	36
3.3 Industry-Standard Violations	38
3.4 Model Limitations	42
3.4.1 Turbine Clustering	43
3.4.2 Cluster Boundary Creation	47
4 Conclusions and Recommendations	49
4.1 Conclusions	49
4.2 Recommendations for Future Research	51
References	53

List of Figures

1.1	(a) Shares of energy sources in the total global energy supply mix by 2050 (based on [14]). (b) Mix of electricity output by energy source in the Netherlands in 2050, averaged from six 2050 scenarios (based on [47]).	1
1.2	Technical potential for offshore wind in shallow and deep waters in (a) Japan; (b) United States; (c) Europe (modified from [20]).	2
1.3	Schematic of FOWT with floating platform, mooring lines, anchors and dynamic IACs highlighted (modified from [42]).	2
1.4	(a) Branch topology; (b) Radial topology; (c) Ring or loop topology.	4
1.5	Ring topology with one OSS supporting eight turbines. The warmer the color, the more power is carried by the cable and hence, a higher cable capacity is needed. (a) Regular situation where the cable between WTG 4 and WTG 5 can be considered redundant. (b) Situation where disconnection appears between the OSS and WTG 1, illustrating the redundancy of the system as all power is still transferred to the OSS.	5
1.6	Optimization approaches for the WFCR problem for BFOWFs used in the published literature since 2020.	6
2.1	Flow chart of the complete solution approach to optimize the WFCR for FOWFs as defined in Section 2.1.	12
2.2	Visibility graph for random wind farm layout with $N_{farm} = 2$	14
2.3	Visualization of the k shortest paths calculated using Yen's algorithm on the visibility graph for a random layout with $N_{farm} = 2$. Each color corresponds to a specific source-target node pair, maintaining consistency across the different shortest path orders. (a) First shortest paths ($k = 1$). (b) Second shortest paths ($k = 2$). (c) Third shortest paths ($k = 3$).	17
2.4	(a) Start of sweep algorithm with the OSS as initial sorting point and sweep line pointing north. (b) The sorting point is changed to the anchor node of WTG 2, with the sweep line extending back along the common edge between OSS and anchor node.	19
2.5	(a) Clustering configuration using the first-sorted turbine as starting index. (b) Alternative clustering configuration obtained by adjusting the starting index.	20
2.6	(a) Wind farm layout with $N_{farm} = 4$. (b) Delaunay triangulation based on OSS and anchor nodes of turbines in the layout of (a)	21
2.7	(a) Delaunay triangulation with identified boundary edges. (b) Boundary edges marked for removal when the threshold length is set to $L_{threshold} = 1000$ m.	22
2.8	Final concave hull obtained after iterative boundary edge removal when (a) $L_{threshold} = 1000$ m, and (b) $L_{threshold} = 2000$ m.	22
2.9	Transformation of concave hulls into final cluster boundaries. (a) Initial concave hulls generated independently for two clusters. (b) Final boundaries after replacing concave hull edges with shortest paths.	23
2.10	(a) First ($k = 1$) shortest paths for random wind farm layout with $N_{farm} = 3$ and the corresponding node labels to match the crossing matrix row and column headers. (b) The corresponding crossing matrix based on the first shortest paths from the layout in (a)	24
2.11	Examples of two shortest paths sharing a common node. (a) Paths do not cross as their path edges are adjacent. (b) Paths cross as their edges alternate.	25
2.12	Optimized cable routing for random wind farm layout with $N_{farm} = 8$ and $N_{loop} = 4$ in case (a) no Miller-Tucker-Zemlin (MTZ) subtour elimination constraints are made, and (b) with MTZ subtour elimination constraints.	28
2.13	Passage crossings checked for special case where a shortest path uses an intermediate node. (a) The shortest path crosses the passage between anchor nodes of WTG 1 and WTG 3; (b) The shortest path does not cross the passage between anchor nodes of WTG 1 and WTG 4.	30
2.14	Shortest path uses entry node that belongs to passage.	31

3.1	Substation placement scenarios with their corresponding allowed turbine areas A_{farm} : (a) Centrally-located substation; (b) Substation located near the field boundary.	33
3.2	Schematic representation of neighboring turbines with key geometric parameters.	34
3.3	(a) Original farm and cable layout for the London Array BFOWF; (b) Optimized cable layout for the London Array BFOWF.	35
3.4	(a) Passage violations in the optimized cable routing in case the London Array wind farm would become a FOWF; (b) Close-up of most violated passage, passage 2, with its seven cables that are routed through the passage.	36
3.5	Comparison of passage violations before and after turbine position adjustment. (a) Optimized cable layout for the London Array FOWF scaled by a factor of 1.25, highlighting passage violations; (b) Passage violation resolved after shifting the turbine associated with the violated passage, ensuring sufficient clearance for the suggested cable routing.	37
3.6	Comparison of (a) total number of unique violated passages and (b) total number of violations for different OSS placements across five random layouts.	39
3.7	Spatial distribution of passage violations for the highest loop-to-turbine density scenario, comparing (a) OSS positioned at the edge and (b) OSS positioned at the center of the wind farm. The size of each marker represent the frequency of violations at that passage, with larger circles indicating a higher number of violations.	40
3.8	Ratio of unique violated passages to total amount of violations for seven loop-to-turbine densities for two OSS placement scenarios.	41
3.9	Proportion of inter- and intra-cluster violated passages across different loop-to-turbine densities for both OSS placement scenarios.	42
3.10	Spatial distribution of unique violated passages aggregated across five random wind farm layouts for three different turbine densities: (a) $N_{farm} = 40$, (b) $N_{farm} = 60$, and (c) $N_{farm} = 80$. Larger markers indicated passages with a higher frequency of violations.	42
3.11	Computational time for sequentially solving clusters of $N_{cluster} = 10$ with two loops, increasing from one to ten clusters.	44
3.12	(a) Average computational time for solving solving a single cluster with $N_{cluster}$ ranging from 8 to 12 and loop sizes N_{loop} varying from 1 to 6; (b) Average computation time for solving a single cluster with $N_{cluster}$ ranging from 12 to 15 and loop sizes N_{loop} varying from 1 to 7. In both plots, $N = N_{cluster}$	45
3.13	Effect of the starting index on cluster formation, where (a) the default start index leads to an intuitive less efficient clustering where turbines that are geographically distant are grouped together, and (b) adjusted starting index leads to a more intuitive clustering.	46
3.14	Optimized cable routing for (a) the suboptimal, default clustering, and (b) the adjusted clustering configuration.	46
3.15	Sensitivity of the valid length threshold $L_{threshold}$ range for different turbine amounts, where (a) $N_{farm} = 40$, (b) $N_{farm} = 60$, and (c) $N_{farm} = 80$	47
3.16	Delaunay triangulation as input for the concave hull construction and its removed edges due to $L_{threshold}$ for (a) $L_{threshold} = 870$ m, and (b) $L_{threshold} = 860$ m.	48
3.17	Concave hull boundaries for (a) $L_{threshold} = 870$ m and (b) $L_{threshold} = 860$ m.	48

List of Tables

1.1	Overview of optimization methods and non-FOWF-specific design choices in published research regarding WFCR of FOWFs.	7
1.2	Overview of floating-specific design choices in published research regarding WFCR of FOWFs.	8
3.1	Overview of wind farm parameters for commercially commissioned Floating Offshore Wind Farms (FOWFs).	33
3.2	Comparison of original and optimized cable lengths for cluster 1 and 2.	35
3.3	Violated passages in the optimized cable routing of the London Array wind farm as a FOWF.	36
3.4	Violated passages in the non-scaled and scaled version of the London Array FOWF layout.	37
3.5	Cluster and loop configurations for randomly generated wind farms with $N_{farm} = 60$	38
3.6	Comparison of allowed turbine area A_{farm} and packing efficiency η_p for $N_{farm} = 60$ for different OSS placement scenarios.	39
3.7	Number of loops L_N inside a wind farm or single cluster of size $N_{farm} = N_{cluster} = 8, 9, 10, 12, 14, 15$ for different loop sizes N_{loop}	44
3.8	Wind farm packing efficiencies η_p for different N_{farm}	47

Nomenclature

List of Abbreviations

BFOWF	Bottom-Fixed Offshore Wind Farm
BFOWT	Bottom-Fixed Offshore Wind Turbine
CAPEX	Capital Expenditures
CVRP	Capacitated Vehicle Routing Problem
FOWF	Floating Offshore Wind Farm
FOWT	Floating Offshore Wind Turbine
IAC	Inter-Array Cable
IMCTS	Improved Monte Carlo Tree Search
MILP	Mixed-Integer Linear Programming
MINLP	Mixed-Integer Non-Linear Programming
MIQP	Mixed-Integer Quadratic Programming
MTZ	Miller-Tucker-Zemlin
OSS	Offshore Substation
PSO	Particle Swarm Optimization
VNS	Variable Neighborhood Search
WFCR	Wind Farm Cable Routing
WFLO	Wind Farm Layout Optimization
WTG	Wind Turbine Generator

List of Symbols

A_{farm}	Allowed turbine area	S	Scope
$A_{mooring}$	Footprint of mooring system	w	Width of farm or field
a	Orientation parameter	η_p	Packing efficiency
b	Orientation parameter	ϕ	Angle between two mooring lines
$C_{max,allowed}$	Allowed amount of cables that can be routed through a passage	φ	Theoretical angle between x-as and position of the closest turbine at the left or rightmost position
D	Horizontal distance between two turbines	θ_{cone}	Cone angle in which all cables must enter and exit the OSS
D_{min}	Minimum horizontal distance between two turbines	θ_m	Angle between reference and mooring line edge
D_{rotor}	Rotor diameter	θ_p	Angle between reference and passage edge
d_{min}	Industry standard for minimum clearance distance between two cables, a cable and mooring line and anchors		
d_{OSS}	Minimum distance between OSS and any anchor		
H	Vertical distance between two turbines		
H_{min}	Minimum vertical distance between two turbines		
h	Water depth		
k	Shortest path order		
L_{anchor}	Distance between two anchor points of same turbine		
L_{lower}	Lower bound of threshold length parameter		
L_N	Amount of loops		
L_p	Passage length		
$L_{threshold}$	Boundary threshold length parameter for concave hull		
L_{upper}	Upper bound of threshold length parameter		
$l_{mooring}$	Actual mooring line length		
l_p	Length of passage		
N	Amount of turbines		
$N_{cluster}$	Amount of turbines in cluster		
N_{farm}	Amount of turbines in wind farm		
N_{loop}	Amount of turbines in a loop		
R_{anchor}	Horizontal projection of mooring line		

List of Optimization Variables and Sets

A	List of k -shortest paths
B	List of potential shortest paths
\mathcal{B}	Root-branches connected to substation
C	Set of cable pairs that cross each other
d_{ij}	Euclidean distance of shortest path between nodes i and j
E	Set of edges in graph G
f_{ij}	Continuous variable that represents the flow in the cable
G	Graph with nodes V and edges E
M	Constant for subtour elimination
\mathcal{N}	Set of turbines
\mathcal{N}_s	Set of turbines and substation
n_s	Source node
n_t	Target node
s	Substation
t_j^k	Binary decision variable if turbine j is included in loop k
u_j^k	Continuous decision variable that resembles the position of node j in the sequence of loop k
V	Set of nodes in graph G
x_{ij}^k	Binary decision variable for cable connection between nodes i and j in loop k

1.1. The Rise of Offshore Wind Energy

Wind energy, both onshore and offshore, has the potential to contribute to 25% of the total global energy supply mix by 2050, see Figure 1.1a [14]. In the Netherlands, it is expected that up to 58% of the electricity output mix will come from wind energy, see Figure 1.1b [47]. However, the expansion of onshore wind is limited by land use restrictions, as well as noise and visual pollution concerns. Therefore, realizing these projections necessitates exploiting the potential of offshore wind energy. According to the International Energy Agency, annual offshore wind installations need to more than quadruple to align with the Sustainable Development Scenario for 2030 and the Paris Agreement targets. Hence, the European Union aims to invest nearly 20 billion euros annually in the wind energy sector by 2030, with 60% of this investment directed towards the offshore wind market [21], [51]. Next to the fact that issues like land scarcity, noise, and visual pollution are less of a concern offshore, the wind flow is also generally steadier and faster, which enhances the capacity factors of the turbines, allowing them to generate electricity more consistently and abundantly [4]. Innovations in turbine technology, such as larger blades and robust foundations capable to withstand harsher environmental conditions and support heavier turbines, have further driven growth in the offshore wind sector. These advantages, technical advancements, and political and financial incentives make it feasible to deploy massive multi-megawatt turbines and expansive wind farms offshore, significantly pushing the growth of offshore wind energy [27].

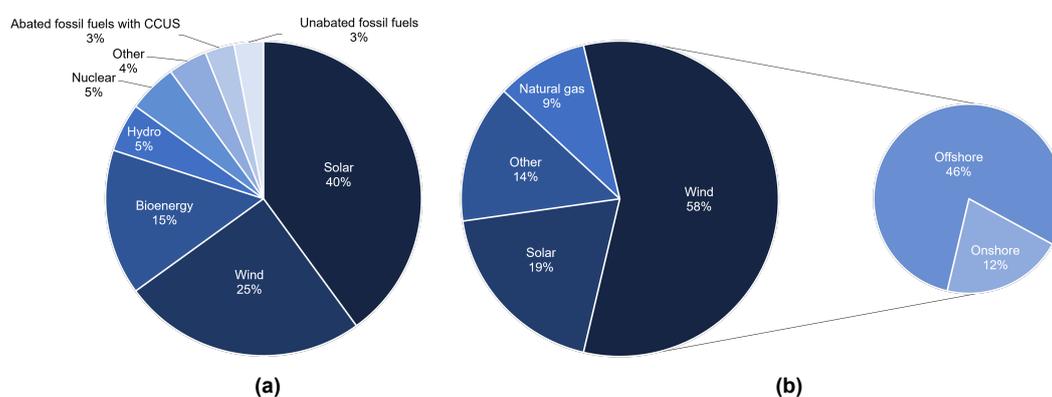


Figure 1.1: (a) Shares of energy sources in the total global energy supply mix by 2050 (based on [14]). (b) Mix of electricity output by energy source in the Netherlands in 2050, averaged from six 2050 scenarios (based on [47]).

Most existing offshore wind turbines are built with bottom-fixed foundations in water depths under 60 meters. However, countries like Japan, the United States, and some in Western Europe, such as France, Portugal, and Norway, often have limited coastal areas with such shallow waters [20], see Figure 1.2. As a result, over the past decade, there has been a significant interest in Floating Offshore Wind Turbines (FOWTs) [54], which currently can reach depths of 300 meters [30]. These deep water locations are typically remote from human habitats, which reduces the impact on human activities and lessens potential conflicts related to nearshore development [56]. It is estimated that about 80% of Europe's offshore wind potential is located in waters deeper than 60 meters, highlighting the increasing importance of FOWTs [27].

Shallow water (10-60 m): ■ Near shore (< 60 km) ■ Far shore (60-300 km)
 Deep water (60-200 m): ■ Near shore (< 60 km) ■ Far shore (60-300 km)

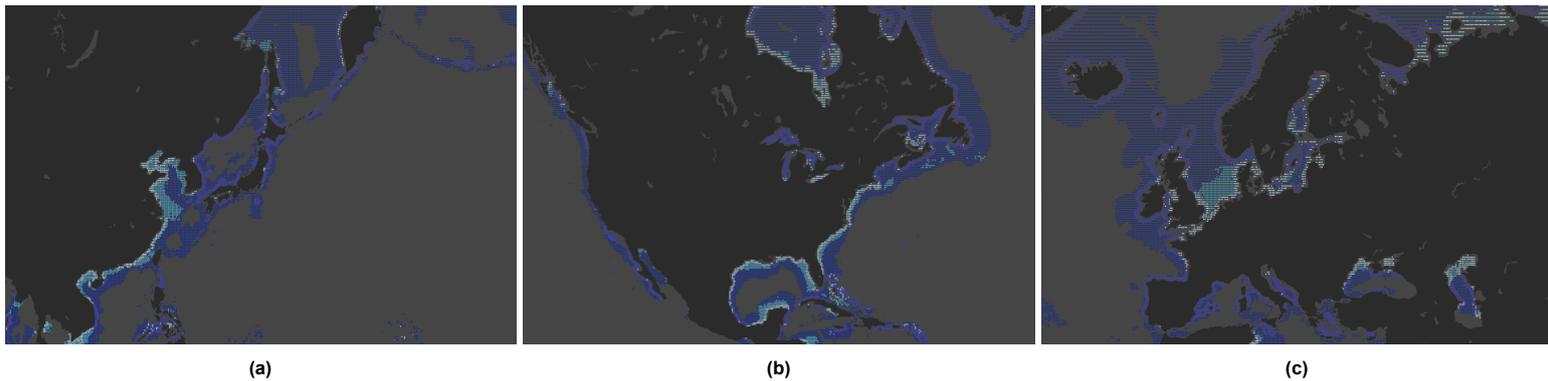


Figure 1.2: Technical potential for offshore wind in shallow and deep waters in (a) Japan; (b) United States; (c) Europe (modified from [20]).

1.2. Floating Offshore Wind

The primary distinction in design between shallow and deep water offshore wind turbines lies in their support structure. Besides the Wind Turbine Generator (WTG), which is similar to that of Bottom-Fixed Offshore Wind Turbines (BFOWTs), FOWTs include three main components: a floating platform, mooring lines, and anchors to secure the structure to the seabed [56], as illustrated in Figure 1.3. The mooring systems currently employed for FOWTs are adapted from those traditionally used in the offshore oil and gas industry. These systems typically consist of at least three mooring lines and a corresponding number of anchors per wind turbine [9]. Inter-Array Cables (IACs) are needed to connect the turbines and transmit the generated electricity to an Offshore Substation (OSS). Typically, an OSS houses one or more transformers that step up the voltage. This higher-voltage electricity is then transmitted via export cables to an onshore substation, where the voltage is further increased to meet the specifications required for distribution by the network operator [26]. Most future floating wind projects are expected to require one or more OSSs, similar to existing large-scale offshore projects that already utilize multiple OSSs [29].

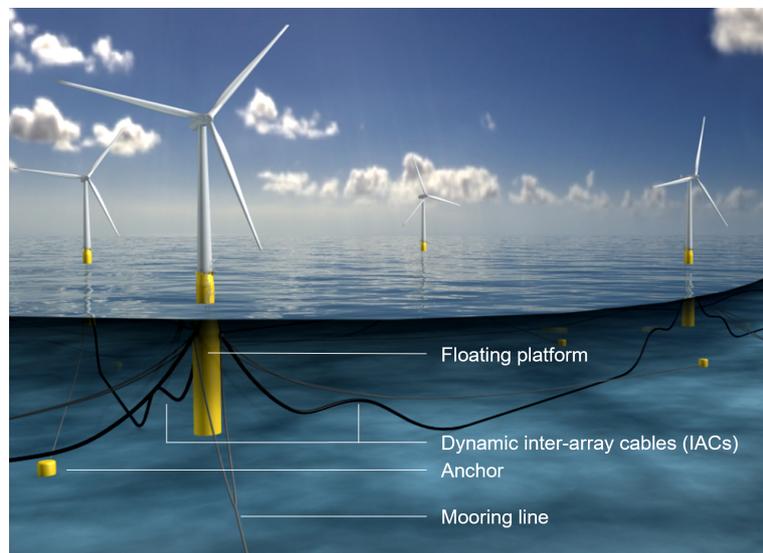


Figure 1.3: Schematic of FOWT with floating platform, mooring lines, anchors and dynamic IACs highlighted (modified from [42]).

Although the first FOWFs have been commissioned, the primary challenge in achieving commercial viability of floating wind power is the substantial initial capital investment required, which significantly exceeds that of BFOWTs [11], [33]. A study by Maienza et al. [31] estimated that the Capital Expenditures (CAPEX) for a FOWF is twice as high as that of a Bottom-Fixed Offshore Wind Farm (BFOWF). In addition to the increased cost of floating platforms compared to bottom-fixed foundations, the higher CAPEX of FOWFs primarily arises

from several factors intrinsic to their floating nature. Firstly, unlike BFOWTs that use static IACs designed to remain stationary once installed, dynamic IACs are suggested for FOWTs. These dynamic cables are engineered to endure frequent bending, flexing, and movement, making them well-suited to the dynamic marine environment where turbines face significant motion from waves, currents, and wind forces [29], [50]. However, the flexibility and durability of dynamic cables come at a higher cost compared to static cables, as their design often incorporates additional joints or floaters. Possible configurations for dynamic cables are adapted from designs commonly used in the oil and gas industry. These configurations include the lazy wave, steep wave, and lazy S, among others. Such configurations inherently result in longer cable lengths than static cables, as they follow an S-shaped trajectory, as illustrated in Figure 1.3. Another option is a hybrid design, which would incorporate a static section in areas where frequent movement is minimal. Although static cables are less expensive than dynamic ones, this hybrid design requires costly joints to connect the dynamic and static sections [29]. Secondly, FOWFs require longer export cables, as the turbines are often located in deeper waters farther from shore. Thirdly, cable routing in FOWFs is complicated by the presence of mooring lines that anchor the floating platforms. These mooring lines are uncrossable, as any mechanical clash between IACs and mooring lines can result in damage, leading to downtime, energy losses, and expensive repairs or replacements. Consequently, cable routing for a FOWF is invariably longer and more intricate than for a BFOWF with an equivalent layout.

1.3. The Wind Farm Layout Optimization Problem

To enhance the cost-competitiveness of FOWFs compared to BFOWFs, it is essential to maximize the annual energy production and thereby reducing the levelized cost of energy [2]. Achieving this objective involves optimal positioning of turbines within the given boundaries of the wind farm while adhering to a minimum distance constraint. This optimization problem is known as the Wind Farm Layout Optimization (WFLO) problem. The minimum distance constraint defines a radius around each turbine within which no other turbines may be placed, often ranging from two to four times the rotor's diameter [60]. This spacing not only prevents collusion between turbine blades but also ensures that mechanical loads and stresses, arising from wakes and turbulence, do not exceed the manufacturer's design thresholds [7]. Besides these safety reasons, the WFLO problem also seeks to reduce these wake effects to increase the turbine's power generation capacity and efficiency, as turbines positioned in the wake of others experience reduced wind speeds [60]. Limitations on the use of space are typically captured in governmental tender regulations, which specify a minimum wind farm capacity for a given area. For FOWFs, the WFLO is further complicated by the need to determine the optimal placement and number of mooring lines and anchors, ensuring stability while maintaining economic viability [16]. Although increasing the spacing between turbines can reduce wake effects and simplify mooring system decisions, it simultaneously leads to an increased use of space and longer IACs.

1.4. The Wind Farm Cable Routing Problem

The trade-off between turbine spacing and cable length is particularly pronounced in FOWFs, as the factors contributing to the increased costs of FOWFs are closely tied to the electrical infrastructure of the wind farm. In BFOWFs, the electrical system typically accounts for 15-30% of the CAPEX. For FOWFs, the challenges of dynamic cables, extended routing, and mooring line constraints likely increase the proportion of electrical infrastructure costs even further [2], [28], [35]. To mitigate these costs while satisfying technical constraints, optimizing the cable routing of a wind farm becomes imperative. This optimization problem is called the Wind Farm Cable Routing (WFCR) problem. As wind farms increase in size and are developed on more complex sites, the optimization of the cable routing is further complicated by obstacles such as variable seabed conditions, existing infrastructure (e.g. pipelines and shipwrecks), and regulatory zones like navigational and fishing corridors, where cable placement is restricted [7]. These cost-increasing factors make the WFCR problem a critical area of focus for reducing capital costs and improving the commercial viability of FOWFs.

1.5. Cable Topologies

The WFCR is influenced by the selected topology, as it has a significant impact on cable length, electrical losses, and overall system reliability. Typically, three topologies are widely considered: the branch, radial, and ring or loop topology, as illustrated in Figure 1.4.

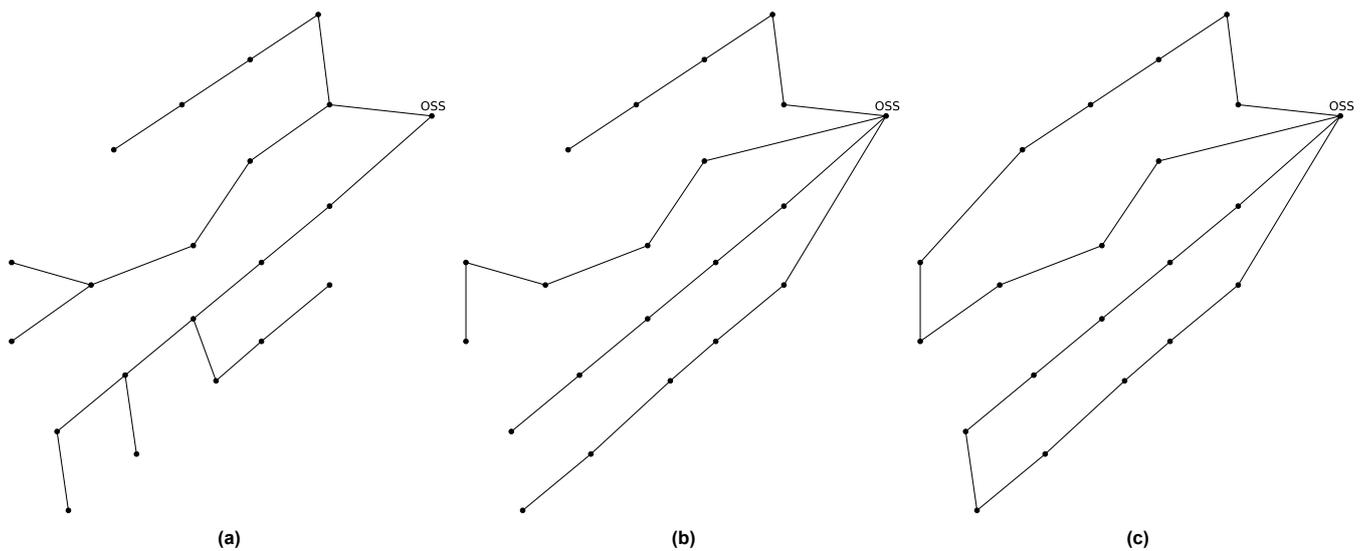


Figure 1.4: (a) Branch topology; (b) Radial topology; (c) Ring or loop topology.

In the branch topology, turbines are connected to the OSS through multiple branches. Each turbine in the branch can receive power from several incoming cables, but only one outgoing cable is used to transmit the power towards the subsequent receiver, which can be either another turbine or an OSS. The cables exiting each turbine need to be sized to handle both the power generated by that turbine and the upstream turbines connected through smaller branches. This leaves room for optimization by ensuring that cables are properly dimensioned, avoiding unnecessary over-sizing. While smaller branches may fail without significantly impacting the overall system, a failure in a main branch - carrying power from multiple smaller branches - can cause a widespread loss of power transmission. This cable topology can be modeled using a Capacitated Minimum Spanning Tree, ensuring that the largest available cable can handle the required power capacity [34].

The radial topology, a subset of the branch configuration, organizes turbines into strings, where at most one cable enters and exactly one cable exits each turbine. Since the radial topology is a more restricted version of the branch topology, its cable routing can never be shorter than that of the branched. However, while it can not result in a shorter cable routing, it can ease the computational complexity by considering engineering constraints from the beginning [7]. The radial topology is widely used in commercial-scale BFOWFs due to its simplicity, flexibility in control, and relatively low initial investment cost. However, like the branch topology, the radial topology lacks redundancy, as a cable failure or disconnection in any main branch will disconnect several upstream turbines, compromising overall reliability. The branch topology could address this issue by allowing for the isolation or shutdown of upstream turbines in smaller branches when cable failures or replacements occur. Yet, this potential advantage comes at the expense of higher costs and added complexity compared to the radial topology, as additional electrical components are required. Consequently, the branch topology is often dismissed early in the optimization process, despite the theoretical operational benefits it may offer [34].

As mentioned, both radial and branched topologies lack redundancy, meaning that a disconnection in main branch will cause all upstream turbines to shut down, as power can not be transferred to a receiver [49]. Nevertheless, the frequent use of these topologies is often defended by the fact that the buried cables have relatively low failure rates - ranging from 0.08% to 1.5% per kilometer per year [12] - compared to other components. Despite these low failure rates, Young [59] reports that cable failures contribute up to 77% of the global cost associated with offshore wind farm losses, and 95% of offshore wind projects have encountered at least one cable-related insurance claim. In addition, the justification of buried cables is less relevant for FOWFs, where a significant part of the cables can not be buried due to the need for dynamic cables. These cables are inherently more vulnerable to damage, potentially resulting in higher failure rates and consequently increasing the global cost associated with offshore wind farm losses even more. Furthermore, for large-scale FOWFs located far from shore, the inaccessibility and long repair times of these cables can exacerbate the situation, leading to outages that may last two to three months, causing significant revenue losses [53].

To improve the redundancy of the electrical infrastructure, a so-called double-sided loop or ring topology is proposed, as it enables bidirectional power flow to the OSS [64]. The loop topology closely resembles the radial topology, but with the key distinction that turbine strings form closed loops. In the event of cable failure or maintenance, the power can still be rerouted to the OSS, enhancing system robustness. However, this increased redundancy comes at a higher cost, as the loop topology involves longer cable routes, the need for cables with higher ratings, and additional electrical components like switches. Moreover, the system demands more complex control mechanisms. To accommodate bidirectional flow, cable capacities must be designed to withstand worst-case scenarios, as shown in Figure 1.5. Under normal operating conditions, each cable carries approximately half of the total power generated within the loop at its highest load point, which occurs near the OSS, highlighted by the orange lines in Figure 1.5a. However, in a worst-case disconnection where the loop is interrupted, resulting in a single string, or radial configuration, the maximum cable capacity must be able to support the full electricity output of all turbines in the loop. This capacity is needed at the connection between the most downstream turbine and the OSS, as indicated by the dark red line in Figure 1.5b. The loop topology can be modeled as a Capacitated Vehicle Routing Problem (CVRP), where the OSS, turbines, and cable capacities are analogous to the depot, customers, and vehicle capacities, respectively. Although the initial investment cost for this topology is higher, it is argued that the loop topology could result in long-term operational savings, ultimately making it the most cost-effective solution over the lifespan of the wind farm [10].

In all three topologies, cables departing from the OSS are part of a root-branch. Balanced cable routing is achieved when each root-branch supports a nearly equal number of turbines, with variations of at most one turbine per branch if even distribution is not possible [7]. A balanced configuration is widely used in the offshore wind farm industry [1]. Receiving the same electrical loads from each root-branch allows the transformers in the substation to share the same design, which presents the following advantages [7]:

- The design and parameter testing phase of the transformers can be standardized, thus reducing installation costs and labor time.
- A single type transformer is needed as spare in case of failures. If the spare transformer is already installed in the substation (online spare), it can substitute any of the other transformers in case of failure, since they share the same specifications. Due to spatial imitations, it is frequently impossible to accommodate more than one spare transformer in the substation.
- It balances the wear of the components, thus it is preferred for the lifetime of the transformers.
- The protection schemes are easier to design for a set of identical transformers.

Furthermore, balanced cable routing can help reduce the variety of cable types needed, as similar load conditions across branches enable the use of standardized cable designs, minimizing the overall quantity of different cable types required [1]. While balanced cable routing can simplify the optimization process by reducing the search space and improving computational tractability, it is important to note that imposing such constraints does never lead to a better global optimum.

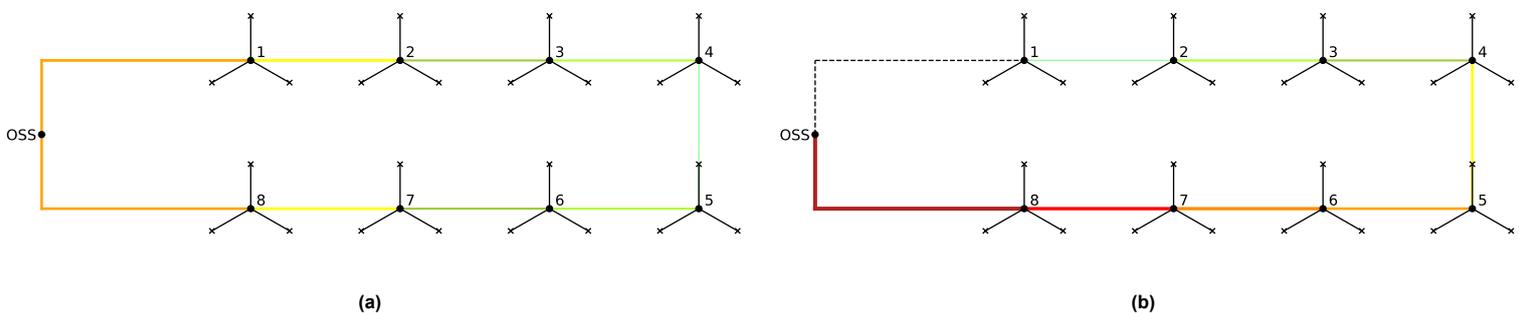


Figure 1.5: Ring topology with one OSS supporting eight turbines. The warmer the color, the more power is carried by the cable and hence, a higher cable capacity is needed. **(a)** Regular situation where the cable between WTG 4 and WTG 5 can be considered redundant. **(b)** Situation where disconnection appears between the OSS and WTG 1, illustrating the redundancy of the system as all power is still transferred to the OSS.

1.6. Research Problem and Academic Knowledge Gap

The cable routing of FOWFs is inherently longer compared to that of BFOWFs. This is primarily due to several unique factors discussed in Sections 1.4 and 1.5: the use of dynamic cables, which require additional length due to their S-shaped trajectory and to accommodate the movement of floating platforms; the necessity of routing around mooring systems to avoid interference; and the proposed adoption of loop topologies, which enhances system redundancy. Since cable length is directly proportional to the cost of the wind farm's electrical infrastructure, optimizing the WFCR is critical for reducing CAPEX and ensuring the commercial viability of FOWFs.

There is a large body of research concerning the optimization of WFCR for BFOWFs using deterministic ([8], [25], [36], [37], [38], [45]), heuristics ([18], [62], [63], [65]), and metaheuristics ([7], [19], [29], [41], [43], [46], [53], [55], [58], [64]), as shown in Figure 1.6. Among these, exact optimization techniques such as branch-and-cut algorithms are widely used for WFCR for BFOWFs [37]. For large-scale wind farms involving hundreds of WTGs, matheuristic approaches have gained prominence over the past five years [15], [39], [52]. Matheuristics combine heuristics with exact optimization methods to enhance computational efficiency and improve the scalability of solutions.

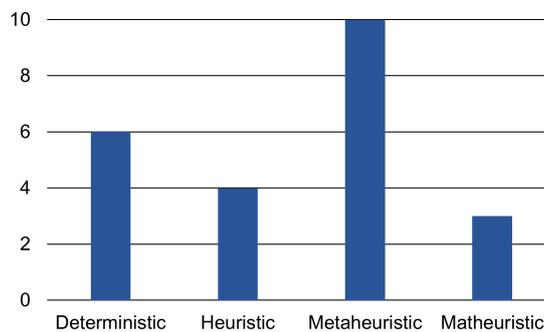


Figure 1.6: Optimization approaches for the WFCR problem for BFOWFs used in the published literature since 2020.

Optimizing the WFCR for FOWFs presents additional challenges due to the integration of mooring systems into the optimization. Mooring systems create restricted zones where cables cannot be routed to avoid crossing or interference with mooring lines. These restricted zones significantly reduce the available space for routing between turbines and/or its OSS, complicating the overall cable routing process. Additionally, the adoption of a loop topology further increases complexity, as it requires cable lengths that are at least twice as long as in a radial topology. In a loop topology, cables must not only circumvent mooring lines but also avoid crossing or overlapping with other cables in the routing, which places further constraints on the available routing possibilities. The combination of reduced routing space due to mooring systems and the increased cable length demands of loop topologies makes identifying a feasible routing solution challenging, let alone achieving an optimal one.

The design and analysis of dynamic cables and cable routing for FOWFs is a relatively new research area. Most research to date regarding floating wind has focused on the dynamic analysis, simulation, and testing of floating platforms under environmental conditions such as wave excitation, currents, and wind loading. Optimization of the design of a single dynamic power cable and its components has been subject of investigation by Rentschler et al. [44]. However, research addressing the optimization of the WFCR of FOWFs remains limited. To the best of the author's knowledge, only the works of Lerch et al. [28], [29] and Song et al. [48] have been published prior to this thesis.

Both studies by Lerch et al. [28], [29] employ metaheuristic approaches using Particle Swarm Optimization (PSO). Similarly, Song et al. [48] uses a metaheuristic approach, employing Binary PSO in its upper layer of its two-layer optimization framework. The lower layer incorporates a heuristic method, namely Improved Monte Carlo Tree Search (IMCTS). All three studies apply their models to layouts of existing wind farms, each featuring a single OSS and a maximum of up to 50 turbines in a branch topology. A fourth study addressing the WFCR problem for FOWFs was published recently, while this thesis was in progress. This study by Pérez-Rúa et al. [39] adopts a matheuristic approach, combining three sequential Mixed-Integer Linear Programming (MILP) models for global optimization with a Variable Neighborhood Search (VNS) heuristic for

solution refinement. The optimization framework is applied to two self-designed FOWF layouts of different sizes, each featuring a single OSS and branch topology. Table 1.1 provides an overview of the optimization methods and non-floating-specific design characteristics used in these state-of-the-art studies addressing WFCR for FOWFs.

Article	Approach	Algorithm	Topology	OSS	WTGs	Existing layout
Lerch et al. [28]	Metaheuristic	PSO	Branch	1	50	Gulf of Fos Wind Farm
Lerch et al. [29]	Metaheuristic	PSO	Branch	1	30, 50	Barrow, Gulf of Fos Wind Farm
Song et al. [48]	(Meta)heuristic	Binary PSO, IMCTS	Branch	1	28	Guishan Wind Farm
Pérez-Rúa et al. [39]	Matheuristic	MILP, VNS	Branch	1	10, 34	No

Table 1.1: Overview of optimization methods and non-FOWF-specific design choices in published research regarding WFCR of FOWFs.

The four articles vary in their choice of cable types and how extended cable lengths resulting from the presence of dynamic cables in FOWFs are addressed. Lerch et al. [28], [29] both consider hybrid cables, which combine a dynamic segment with a static segment buried in the seabed. In the 2019 study, the dynamic cable length is estimated using a simplified empirical factor, but this factor is not tied to any specific cable configuration, limiting its accuracy and applicability. The 2021 study improves upon this by explicitly incorporating the dynamic cable's lazy wave configuration. It also makes a distinction between static and dynamic segments in their cable length calculation, providing a more detailed and realistic calculation of its length. In addition, a cost comparison is made between hybrid and fully dynamic cables for both layouts. Song et al. [48] also adopts hybrid cables and builds upon earlier research by Rentschler et al. [44] to account for the extended length of dynamic cables. Unlike earlier studies, Song et al. [48] incorporates the effects of platform motion and environmental factors such as wind-wave coupling in the cable length calculation. Pérez-Rúa et al. [39] also assumes hybrid cables and applies the definition of dynamic cable lengths as established by Rentschler et al. [44].

The articles also differ in how additional cable routing caused by mooring systems is addressed and whether the mooring systems are explicitly modeled in their optimization frameworks. Lerch et al. [28] does not account for additional routing caused by the mooring system. It neither models the interaction between cables and mooring lines, nor introduces constraints to avoid interference. The concept of a mooring or station-keeping system is entirely absent, ignoring its influence on cable routing. In their updated study, Lerch et al. [29] introduces a more refined approach by adding a 5% increase in static cable length to account for routing around mooring lines. While this adjustment improves upon the 2019 methodology, the increase is a simplified approximation without any geometric or spatial justification. The mooring system remains absent from the layout, and its spatial impact is only addressed indirectly through this adjustment, leaving significant room for further refinement. Song et al. [48] takes a more advanced approach by including a routing constraint to ensure cables avoid mooring lines. The optimization framework states "cable lines should not be arranged in the mooring system area". However, the study does not define the extent or shape of this mooring system area or analyzes how this constraint impacts cable routing feasibility or costs. The mooring system is also not explicitly modeled or visualized in the wind farm layout, leaving its integration incomplete. Pérez-Rúa et al. [39] provides the most detailed representation of mooring systems among the four studies. This study explicitly models the mooring system by introducing safety zones around mooring lines. These zones are defined based on the worst-case two-dimensional horizontal displacements of the lines, determined through dynamic simulations of the floating platform under environmental conditions. These safety zones, represented as triangular bounding boxes, ensure that cables are routed outside these areas, avoiding potential clashes with mooring components. This explicit modeling of mooring systems and their integration into the cable layout optimization represents a significant advancement in the state-of-the-art knowledge, enabling the framework to address the spatial implications of the mooring systems more comprehensively. An overview of the FOWF-specific design considerations in the state-of-the-art research can be found in Table 1.2.

Article	Cable type	Extended cable length	Extended routing	Mooring modelled
Lerch et al. [28]	Hybrid	Yes	No	No
Lerch et al. [29]	Hybrid, dynamic	Yes	Yes	No
Song et al. [48]	Hybrid	Yes	No	No
Pérez-Rúa et al. [39]	Hybrid	Yes	Yes	Yes

Table 1.2: Overview of floating-specific design choices in published research regarding WFCR of FOWFs.

In summary, the existing research on WFCR optimization for FOWFs has primarily focused on the cost modeling of cable routing and the integration of dynamic cables into the design framework. While these studies have made efforts in addressing challenges specific to FOWFs, only one study explicitly models the spatial impact of mooring systems within the optimization framework. Pérez-Rúa et al. [39] introduces safety zones based on mooring line displacements, representing a significant step forward in incorporating mooring constraints. However, this study assumes a branch topology, which limits its ability to explore the complexities introduced by loop topologies. The academic knowledge gap lies in the lack of a comprehensive optimization framework that integrates mooring system constraints and explores the implications of loop topologies. Based on this academic knowledge gap and the limited state-of-the-art research concerning WFCR optimization for FOWFs, the following research problem is identified:

Existing research on WFCR optimization for FOWFs predominantly addresses cable types, cost modeling of the electrical infrastructure, and approximations for routing adjustments around mooring systems, typically assuming a branched cable topology. However, there is a significant gap in understanding how to effectively integrate the mooring system and its constraints into the WFCR optimization framework while addressing the complexities and spatial demands of loop topologies.

1.7. Research Objective and Research Questions

The objective of this thesis is to provide insights into how the WFCR for FOWFs can be optimized when considering a loop topology. The research focuses on understanding the integration of mooring systems into the wind farm layout and how the additional constraints and routing adjustments imposed by the mooring system and loop topology impact the optimization process. To ensure clarity and feasibility within the scope of this project, the turbine layout is assumed to be fixed, thus excluding the WFLO problem from the scope of this thesis. While the interaction between turbine placement and cable routing is a relevant topic, this thesis specifically focuses on developing feasible solutions for fixed FOWF layouts. This approach allows the study to concentrate on the complexities introduced by mooring systems and loop topologies, which already present significant challenges in the optimization of FOWFs. The framework will aim to minimize the IAC length while ensuring compliance with spatial and technical constraints. The insights gained from this study will contribute to filling the identified knowledge gap in WFCR optimization for FOWFs. This research objective leads to the following main research question:

How can the WFCR for a FOWF with a fixed turbine layout be optimized to minimize total IAC lengths while addressing the technical and spatial constraints imposed by the mooring system and loop topology?

To answer this main research question, the thesis is structured around the following subquestions:

- SQ1. How can feasibility of the solution be enforced or stimulated?
- SQ2. How can we deal with any remaining infeasibility after optimization, and can this infeasibility give insights for design of the wind farm layout?
- SQ3. Can the optimization of cable routing for a FOWF be divided into subproblems of cable routing for clusters of turbines and how does this influence the optimization process?
- SQ4. How does the location of the OSS impact the feasibility of the optimization?

This thesis is conducted in collaboration with Vattenfall, leveraging real-world data and industry insights to ensure that the optimization tool is both practical and applicable to ongoing and future floating wind farm projects. The expected outcome of the research includes the development of a robust optimization tool for

the WFCR of FOWFs. By minimizing cable length while maintaining feasibility, the tool aims to support the technical and economic optimization of floating wind farms, ultimately contributing to their commercial viability and scalability in offshore renewable energy development.

1.8. Thesis Outline

This thesis begins with a detailed explanation of the methodology used to create the optimization tool in Chapter 2 *Methodology*. The *Methodology* chapter is followed by Chapter 3 *Results and Discussion*, where the results of the optimization framework are presented and discussed. Finally, Chapter 4 *Conclusions and Future Recommendations* concludes the research and provides recommendations for future research.

2

Methodology

This chapter outlines the methodological framework used to address the WFCR problem in the context of FOWFs. Section 2.1 describes the problem, outlining the objective function and relevant constraints. The solution approach is then discussed and visualized using a flow chart in Section 2.2. The solution approach starts with several preprocessing steps, presented in Section 2.3, necessary to generate key inputs for the MILP model, which is given in Section 2.4. Any necessary postprocessing steps for further optimization are discussed in Section 2.5.

2.1. Problem Formulation

This thesis focuses on the optimization of the electrical infrastructure for FOWFs, i.e. the Wind Farm Cable Routing (WFCR) problem. This section defines the objective and relevant constraints to find a solution to this optimization problem. The objective of the WFCR problem, as used in this thesis, is:

Given a predefined cable type, the goal is to optimally route cables between turbines, and between turbines and the substation, so that the total cable length is minimized.

In addition to this objective, the optimization is subject to a set of constraints, which are:

- C1.** All produced electricity should be transferred to the OSS, and therefore every loop must contain the OSS. To ensure that all electricity can still be transferred in the event of loop topology disruption, the cable capacity of each interconnection must be sufficient to carry the electricity generated by its own turbine plus all upstream turbines [24].
- C2.** The WFCR will have a loop topology to enhance the robustness of the electrical infrastructure. This ensures that in case of turbine maintenance or cable failure, the affected turbine can be removed while maintaining connectivity of all other turbines [10], [24]. Each turbine must have exactly one incoming and one exiting cable, forming closed loops of connections.
- C3.** The cable routing needs to be balanced to ensure that the OSS receives the same electrical loads from all loops, allowing the transformers in the OSS to share the same design [7]. This means that all loops should support an equal amount of turbines, or can vary at most by one turbine if the total number is not divisible evenly among the closed loops.
- C4.** Mooring lines cannot be crossed by the IACs from a planar point of view, as mechanical interactions between mooring lines and IACs can cause damage, resulting in downtime, energy losses, and expensive repair or replacement [22].
- C5.** According to industry experts, IACs cannot cross each other from a planar point of view and should maintain a minimum clearance distance from both each other and the mooring system to ensure safety during installation and maintenance [22].

2.2. Solution Approach

The WFCR optimization problem, as defined in Section 2.1, is a routing challenge similar to the well-known CVRP, but with the added complexity of planarity constraints [3]. In the CVRP, goods stored at a central depot

must be delivered to multiple customers while ensuring that vehicle capacity constraints are not exceeded. In the WFCR problem, turbines represent customers, and their generated electricity must be transmitted to the OSS, which functions as the depot. Each cable routing loop corresponds to a vehicle route and is constrained by the maximum cable capacity in the same way that a vehicle route is limited by a maximum distance or travel time.

Several approaches can be used to solve the WFCR problem, as discussed in Section 2.1, including exact methods, heuristics, metaheuristics, and matheuristics. Exact optimization methods, such as MILP, provide structured solutions where all variables, constraints, and objectives are explicitly modeled. Compared to heuristics and metaheuristics, exact methods offer several advantages. They guarantee feasibility by rigorously enforcing all constraints, ensuring that no infeasible cable layouts are produced. They provide optimality by identifying the best possible solution within the defined search space, minimizing total cable length while maintaining all constraints. Furthermore, exact methods enhance interpretability by representing routing decisions as binary variables, making it easier to analyze and validate results. While MILP can become computationally demanding for large wind farms, problem decomposition techniques, such as clustering turbines before optimization, can improve efficiency without compromising feasibility.

Heuristic approaches, such as the Minimum Spanning Tree or shortest-path algorithms like Dijkstra's algorithm, can generate initial cable layouts quickly but lack mechanisms to dynamically ensure planarity constraints, maintain loop topology adherence, and balance cable capacities. These limitations make pure heuristics unsuitable as a standalone approach for WFCR. Metaheuristic techniques, such as Genetic Algorithms and Ant Colony Optimization, can efficiently explore large solution spaces but rely on randomized search mechanisms that do not inherently enforce feasibility constraints during optimization. Instead, constraint violations must be handled after optimization, often requiring additional manual adjustments or constraint penalty methods, which do not guarantee a fully feasible solution. Additionally, metaheuristics introduce significant computational uncertainty, as their convergence to a high-quality solution depends on hyperparameter tuning and multiple iterations, making them less predictable than MILP solvers.

Given the limitations of heuristics and metaheuristics, one possible alternative could have been a matheuristic approach, which integrates heuristic techniques within an exact optimization framework. However, this option was not selected for several reasons. First, matheuristics still rely on heuristics to guide the optimization, which means they do not inherently guarantee constraint satisfaction. While they can reduce computational effort, they introduce additional complexity in designing heuristic-guided MILP formulations, making the problem less structured. Since feasibility is a strict requirement in WFCR, using heuristics to influence the MILP solver could lead to suboptimal constraint enforcement. For this reason, an exact approach was chosen in this thesis to optimize the problem as formulated in Section 2.1.

Several exact mathematical optimization techniques can be considered to solve the WFCR problem, including MILP, Mixed-Integer Non-Linear Programming (MINLP), and Mixed-Integer Quadratic Programming (MIQP). MINLP allows nonlinear relationships between variables and constraints, which could theoretically model cable crossings dynamically. However, nonlinear constraints significantly increase computational complexity and make it difficult to achieve feasible solutions within a reasonable time for large-scale wind farms. MIQP is well suited for problems with quadratic cost functions or constraints, such as energy losses in power systems, but is unnecessary for this problem, where the objective function is based on minimizing total cable length, which is linear. MILP is chosen because it enables a structured approach where routing decisions are represented as binary variables, determining whether a cable segment is selected or not, while additional constraints, such as cable capacity limits and loop topology enforcement, are efficiently handled within a linear optimization framework. MILP is also preferred due to its well-established solvers that systematically search for feasible solutions while enforcing strict planarity constraints.

Based on these considerations, this thesis adopts an exact optimization approach using MILP, as it offers the most rigorous and reliable method to ensure feasibility while minimizing total cable length.

The process begins by defining a wind farm layout with a specified number of turbines, N_{farm} , which can either be randomly generated or predefined. While MILP provides a structured optimization framework, its computational complexity increases rapidly with the number of turbines and potential cable connections. Therefore, preprocessing steps are introduced to reduce the solution space before optimization, ensuring that the problem remains computationally manageable. The solution approach is divided into three phases: preprocessing, optimization, and postprocessing, as can be seen in Figure 2.1. Each of these phases plays a critical role in

ensuring that the final solution respects all physical constraints while remaining computationally efficient.

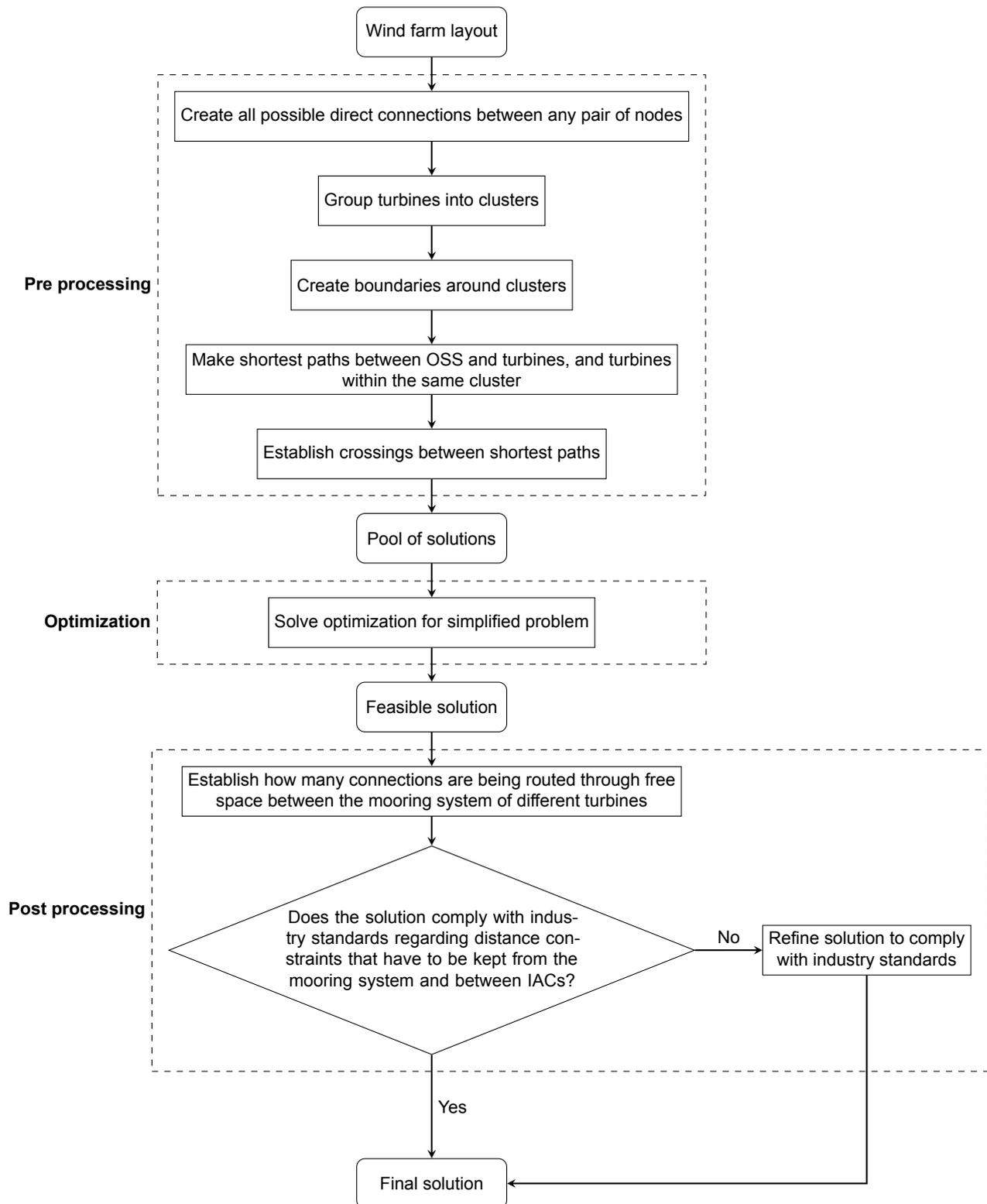


Figure 2.1: Flow chart of the complete solution approach to optimize the WFCR for FOWFs as defined in Section 2.1.

2.2.1. Preprocessing

The preprocessing phase is essential to reduce the number of possible solutions that the exact solver must evaluate. Instead of allowing the solver to explore every potential cable layout, preprocessing limits the search space by defining a set of feasible cable routing paths between turbines and the OSS, and turbine pairs. The first step involves generating all possible direct connections between turbine pairs, between turbines and the OSS, between anchors, turbines and anchors, and anchors and the OSS. These connections are necessary as cable routing paths may have to be routed around anchors.

As the number of possible connections between turbine pairs, and between turbines and the OSS increases factorially with the number of turbines, solving the entire wind farm as a single optimization problem becomes intractable. To address this issue, turbines are grouped into smaller clusters. Clustering reduces the number of possible connections that must be evaluated simultaneously, effectively breaking the problem into smaller, more manageable subproblems. However, it is important to note that due to clustering the turbines, the exact approach turns into an heuristic approach, as optimality is sacrificed to gain computational efficiency.

A key challenge in enforcing planarity constraints is that the MILP model cannot dynamically check for cable crossings, as doing so would introduce nonlinear constraints, converting the problem into a MINLP, which is much harder to solve efficiently. To ensure that crossings between cables do not occur, defined in Constraint C5, potential crossings are identified before optimization begins, and constraints are introduced that prevent the solver from selecting conflicting paths. However, solving clusters sequentially means that the routing decision of one cluster affects the choices available for subsequent clusters, meaning that the constraints and input for each cluster must be dynamically updated, creating dependencies between clusters. This interdependency complicates the optimization process and increases computational overhead.

To eliminate these dependencies and allow clusters to be optimized independently, predefined boundaries are established around each cluster. By restricting the potential cable paths to cross other cluster's boundaries, crossings between clusters are inherently avoided, as the shortest paths for each cluster do not intersect with those from other clusters. This ensures that each cluster can be optimized independently, eliminating the need for dynamically adjusting constraints after each cluster is solved, simplifying the problem structure and eliminating the need to check for inter-cluster crossings. As a result, the shortest paths and potential crossings are determined within each cluster immediately after defining its boundaries, ensuring that all constraints are met while maintaining computational efficiency.

2.2.2. Optimization

Once preprocessing is complete, the MILP solver is used to determine the optimal routing configuration for each cluster sequentially. The solver selects paths from the predefined set, ensuring that all constraints are met, including the loop topology constraint, cable capacity limits, and the restriction against crossings within the cluster. Since inter-cluster crossings have already been eliminated during preprocessing, the MILP solver does not need to check for them dynamically. This significantly reduces the number of constraints and decision variables, improving computational efficiency.

2.2.3. Postprocessing

The postprocessing phase addresses any remaining constraint violations. One of the key challenges is ensuring that the final routing solution does not violate industry standards regarding safe distances between cables and between cables and the mooring system, as defined in Constraint C5. Since this constraint depends on the final cable layout, they cannot always easily be enforced with a preprocessing step or within the MILP model itself, which is further elaborated on in Section 2.5. Instead, the final cable configuration is checked against these requirements in the postprocessing phase, and any violations can be corrected through an iterative manual refinement process.

2.3. Preprocessing

The preprocessing phase is crucial to generate structured inputs for the MILP model, which will be introduced in Section 2.4. The key preprocessing steps include creating direct connections, clustering the turbines, defining cluster boundaries, generating shortest paths, and identifying crossings between potential cable paths. Each of these steps play a vital role in shaping the optimization problem in a way that aligns with the MILP formulation and its computational limitations. Although the sequence of preprocessing steps follows a structured flow,

the concept of shortest paths is fundamental to multiple steps in the preprocessing phase, including turbine clustering and cluster boundary formation. Therefore, this section first details the process of establishing direct connections, followed by an explanation of how the shortest paths are determined. After that, the remaining preprocessing steps are presented in chronological order, following the flowchart in Figure 2.1.

2.3.1. Direct Connections

A fundamental requirement for the WFCR problem is ensuring that all generated electricity is transferred to the OSS, as defined in Constraint C1 in Section 2.1. To satisfy this requirement, every turbine must be directly or indirectly connected to the OSS. Establishing connections between turbines and the OSS, as well as between turbine pairs, is therefore essential. This can be done by representing the wind farm layout as a graph $G(V, E)$, where V represents the set of nodes, or vertices, between which connections are desired, and E represents the set of edges, the direct connections between two nodes. Such a graph can be either weighted or unweighted and directed or undirected. A weighted graph is a graph where a number, or weight, is assigned to each edge. This number can represent for example cable costs, lengths or capacities, depending on the problem formulation. Because cable length is the only objective in this problem formulation, each edge in this graph is assigned a weight corresponding to the Euclidean distance between two nodes. Another result of the cable length being the sole objective, is that the graph is undirected, as the routing direction does not impact the objective.

To define the set of possible edges, a so-called visibility graph is constructed, similar to the approach in [7], [40] and [58]. Figure 2.2 shows the visibility graph for a random wind farm layout with two turbines, hence $N_{farm} = 2$.

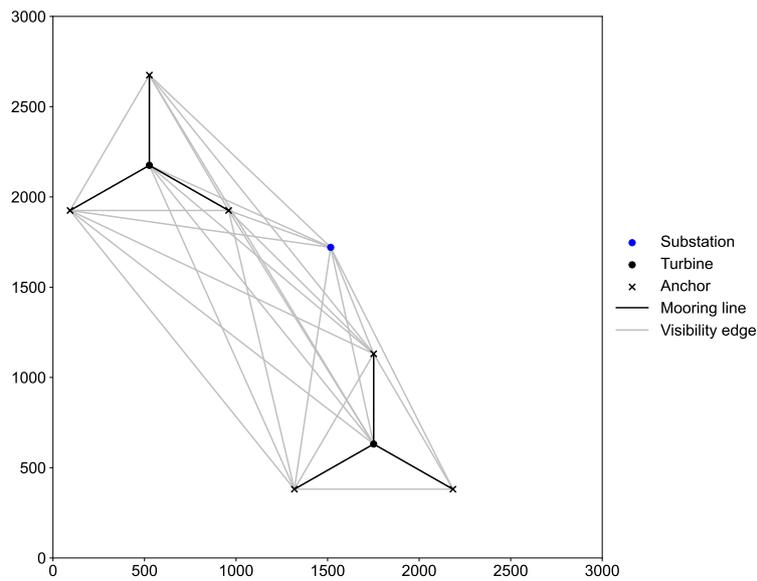


Figure 2.2: Visibility graph for random wind farm layout with $N_{farm} = 2$.

In a visibility graph, two nodes are connected by an edge if the edges are "visible" to each other, meaning that a straight-line connection between them does not intersect any obstacle. From Constraint C4 defined in Section 2.1, it is known that IACs are not allowed to cross mooring lines, and therefore it is desired to treat mooring lines as obstacles. Therefore, any direct connection that intersects a mooring line is excluded from the visibility graph. To ensure feasible cable routing while avoiding such crossings, alternative paths must be introduced in cases where direct connections are infeasible. Since the shortest alternative paths will naturally follow straight-line connections that circumvent mooring lines, anchors are incorporated as additional nodes in the visibility graph. These anchor nodes serve two purposes: they can replace direct connections that would otherwise violate the mooring constraint, and they enable the generation of alternative paths, such as second-shortest and third-shortest paths, when additional constraints are violated. As a result, the set of nodes used in the visibility graph includes not only turbines and the OSS, but also all anchors. The process for constructing the visibility graph is detailed in pseudocode in Algorithm 1. No other obstacles are considered in the used wind farm layouts in this thesis.

Algorithm 1: Visibility graph

Data: Set of nodes V with coordinates, set of mooring lines
Result: Visibility graph $G(V, E)$ with edges only where no mooring line is crossed

```

1 Initialize an empty graph  $G(V, E)$ ;
2 foreach node  $i \in V$  do
3   foreach node  $j \in V$  do
4     if  $j \leq i$  then
5       |                                     ▷ Avoid self-loops and redundant checks
6     end
7      $edge\_excluded \leftarrow \text{False}$ ;
8     foreach mooring line  $(p_1, p_2)$  do
9       if edge  $(i, j)$  intersects mooring line  $(p_1, p_2)$  then
10      |    $edge\_excluded \leftarrow \text{True}$ ;
11      |                                     ▷ Skip adding this edge
12      end
13    end
14    if not  $edge\_excluded$  then
15      |   Compute Euclidean distance  $d$  between nodes  $i$  and  $j$ ;
16      |   Add edge  $(i, j)$  to  $E$  with weight  $d$ ;
17    end
18  end
19 end
20 return  $G(V, E)$                                      ▷ Return the constructed visibility graph

```

2.3.2. Shortest Path Formation

After the visibility graph is created between all nodes, Dijkstra's algorithm is used to find the shortest simple path from a source node to a target node by minimizing the total weight of the path. A simple path is a path without repeated nodes. However, as a loop topology has been identified as a constraint to this problem, turbines can have exactly one incoming and one exiting cable. To enforce this constraint, the shortest path from a source node to a target node must not pass through any other turbine or OSS nodes. These nodes, collectively referred to as center nodes, are excluded from the path unless they are the source or target. This restriction is implemented by constructing a filtered subgraph of the visibility graph, where all center nodes, except the source and target, are removed.

Dijkstra's algorithm begins by initializing all nodes with an infinite distance, except for the starting node, which is set to zero. The algorithm maintains a set of unvisited nodes and selects the node with the smallest known distance, or weight, at each step. For this node, it updates the distances to its neighbors by comparing the sum of the current node's distance and the edge weight to the neighbor's existing distance. If the new distance is smaller, it replaces the previous value. Once all neighbors are considered, the node is marked as visited and removed from further consideration. This process repeats until all nodes are visited or until the shortest path to a specific target node is found. The final result includes the shortest path distances and a record of predecessors that can be used to reconstruct the path from the starting node to any other node [23]. The pseudocode of Dijkstra's algorithm used in this thesis is shown in Algorithm 2.

Multiple restricted versions in terms of excluding certain nodes or edges from the visibility graph before creating the paths of this pathfinding algorithm are used in this optimization framework based on their application. However, whenever this is the case, certain restrictions in terms of excluding edges or nodes will be mentioned in their respective section.

Algorithm 2: Dijkstra's algorithm

Data: Visibility graph $G(V, E)$, source node n_s , target node n_t
Result: Shortest simple path from n_s to n_t , avoiding center nodes

- 1 Construct subgraph G' of G excluding all center nodes except n_s and n_t ;
- 2 **foreach** node v in G' **do**
- 3 $dist[v] \leftarrow \infty$ ▷ Initialize distances to infinity
- 4 $prev[v] \leftarrow \text{None}$ ▷ No known predecessor
- 5 **end**
- 6 $dist[n_s] \leftarrow 0$ ▷ Distance to source node is zero
- 7 Initialize the set of unvisited nodes $U \leftarrow V$;
- 8 **while** U is not empty **do**
- 9 Select node $u \in U$ with the smallest $dist[u]$;
- 10 **if** $u = n_t$ or $dist[u] = \infty$ **then**
- 11 ▷ Terminate if target node is reached or no more reachable nodes exist
- 12 **end**
- 13 **foreach** neighbor v of u in G' **do**
- 14 **if** $v \in U$ and edge (u, v) exists **then**
- 15 $alt \leftarrow dist[u] + \text{weight}(u, v)$ ▷ Compute alternative distance
- 16 **if** $alt < dist[v]$ **then**
- 17 $dist[v] \leftarrow alt$;
- 18 $prev[v] \leftarrow u$;
- 19 **end**
- 20 **end**
- 21 **end**
- 22 Remove u from U ;
- 23 ▷ Mark node as visited
- 24 **end**
- 25 Initialize empty list $path$;
- 26 Set $current \leftarrow n_t$;
- 27 **while** $current \neq \text{None}$ **do**
- 28 Prepend $current$ to $path$;
- 29 $current \leftarrow prev[current]$;
- 30 **end**
- 31 **return** $path$ ▷ Return shortest path from n_s to n_t

Furthermore, sometimes it might be the case that two shortest paths (read: IACs) are crossing each other, which is not allowed as stated in Constraint C5 in Section 2.1. Therefore, it would be desired to generate also the second-shortest path, third-shortest path, up until the k -th shortest path, where k represent the order in the weighted list of paths that are generated. Figure 2.3 shows up until the third-shortest paths between all nodes. These k -shortest paths are generated using Yen's algorithm [57].

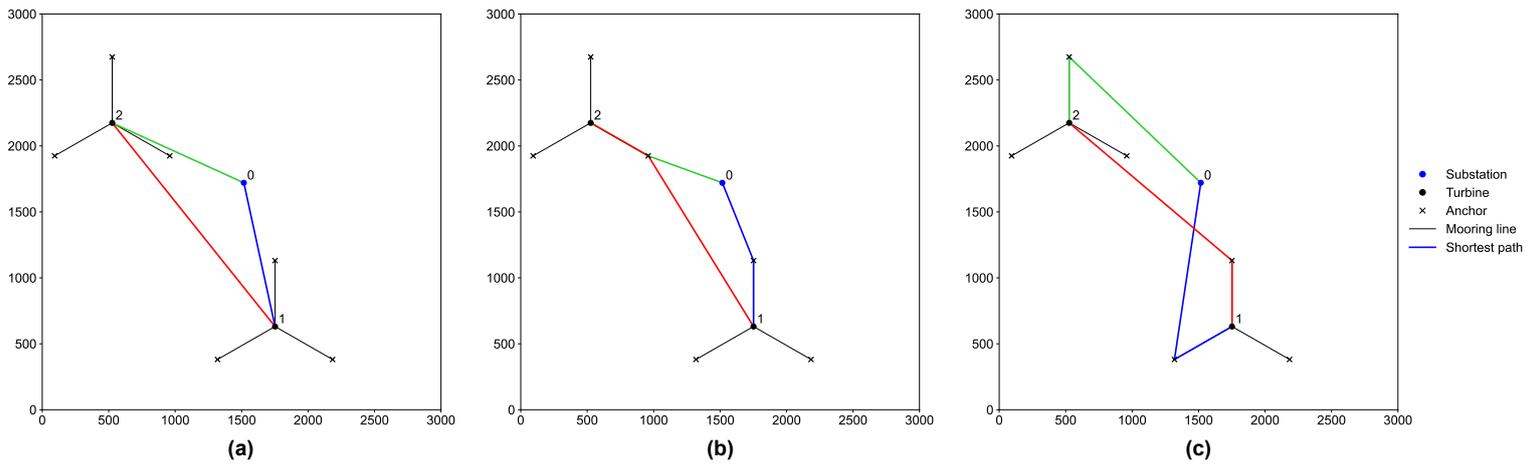


Figure 2.3: Visualization of the k shortest paths calculated using Yen's algorithm on the visibility graph for a random layout with $N_{farm} = 2$. Each color corresponds to a specific source-target node pair, maintaining consistency across the different shortest path orders. **(a)** First shortest paths ($k = 1$). **(b)** Second shortest paths ($k = 2$). **(c)** Third shortest paths ($k = 3$).

This algorithm can be broken down into two parts: determining the first k -shortest path between the source and target node, A^1 , hence $k = 1$, and then determining all other k -shortest paths. To determine A^1 , any efficient shortest path algorithm can be used, but in this case Dijkstra's algorithm is used as mentioned before. This first shortest path is then stored in list A , which keeps track of the k -shortest paths found so far. To find the second shortest path, third shortest path and so on, we need to introduce small variations to the previously found paths. This can be divided into a two-step approach: root path selection and spur path selection. We start by following the first-shortest path as if we were using it again. This common part of the path is called the root path. Then, at some node along the way, we force a deviation by selecting a different path to the target node from that node onward. This new part of the path is called the spur path. After combining the root path and spur path, we get a new, slightly different path, which is added as a candidate to be the next-best path. Each time we create a new candidate path, we add it to a container B , which temporarily stores paths that could be the next shortest. We select the best one (i.e., the one with the smallest total weight) and add it to A as the next k -shortest path. This process is repeated until all k shortest paths have been found. The pseudocode of Yen's algorithm can be seen in Algorithm 3.

Algorithm 3: Yen's algorithm

Data: Visibility graph $G(V, E)$, source node s , target node t , number of paths k
Result: List A containing up to k shortest paths from s to t

```

1  $A[1] \leftarrow \text{Dijkstra}(G, s, t)$ ;
2  $B \leftarrow \emptyset$ ;
3 for  $k \leftarrow 2$  to  $K$  do
4   for  $i \leftarrow 1$  to  $\text{length}(A[k-1]) - 1$  do
5      $\text{spurNode} \leftarrow A[k-1][i]$ ;
6      $\text{rootPath} \leftarrow$  first  $i$  nodes of  $A[k-1]$ ;
7     foreach  $\text{path } p \in A$  do
8       if  $p[1:i] = \text{rootPath}$  then
9         Remove edge  $(p[i], p[i+1])$  from  $G$ ;
10      end
11    end
12    Remove all nodes in  $\text{rootPath}$  from  $G$  except  $\text{spurNode}$ ;
13     $\text{spurPath} \leftarrow \text{Dijkstra}(G, \text{spurNode}, t)$ ;
14    if  $\text{spurPath}$  exists then
15       $\text{totalPath} \leftarrow \text{rootPath} + \text{spurPath}$ ;
16      if  $\text{totalPath} \notin B$  then
17        Add  $\text{totalPath}$  to  $B$ ;
18      end
19    end
20    Restore all removed edges to  $G$ ;
21    Restore all removed nodes to  $G$ ;
22  end
23  if  $B$  is empty then
24    ▷ Terminate if no more alternative paths exist
25  end
26  Sort  $B$  by total path weight in ascending order;
27   $A[k] \leftarrow B[1]$ ;
28  Remove  $B[1]$  from  $B$ ;
29 end
30 return  $A$  ▷ Return list of k-shortest paths

```

2.3.3. Turbine Clustering

As FOWFs are often located further from shore, the wind farm size and number of turbines in these farms tend to increase. However, a larger number of turbines also increases the number of possible configurations for cable routing, making the optimization problem computationally challenging. In the simplest case, where a FOWF is routed in a single loop that starts and ends at the OSS, assuming identical turbines, cable types, and capacities, the number of unique routing configurations for a wind farm with N_{farm} can be expressed as:

$$\text{configurations} = \frac{N_{farm}!}{2} \quad (2.1)$$

Here, the division by two accounts for the fact that a routing sequence is considered identical when traversed in reverse. It is evident from this formula that the number of configurations grows factorially with N_{farm} , far exceeding linear, quadratic, or even exponential growth rates. For instance, with $N_{farm} = 15$ in a single loop, the number of unique routing configurations already reaches 10^{11} .

In practice, turbines in a wind farm are divided into smaller loops rather than a single large loop due to cable capacity constraints and the need for robustness in the event of cable failure or maintenance. This division into smaller loops reduces the solution space by limiting the number of possible routing configurations. Since turbines are routed in smaller loops, clustering them beforehand allows for a further reduction of the solution space, as the loops are then routed within predefined clusters. Clustering, in the context of the WFCR problem, can be defined as "splitting the group of turbines into smaller subgroups, by maximizing the resemblance characteristics among individuals in the same cluster, while minimizing them for two elements belonging to

different subgroups” [34]. There are various characteristics by which turbines can be clustered, but as the objective in this thesis is to minimize cable length, it is intuitive that turbines positioned at opposite ends of the wind farm are unlikely to be routed within the same loop. Therefore, the similarity measure for clustering is based on turbine proximity, taking into account their relative positions. However, clustering introduces the challenge of determining an appropriate clustering algorithm and selecting the optimal cluster size. The cluster size must be larger than the maximum cable capacity to allow flexibility in the optimization. If the cluster size is set exactly equal to the maximum capacity, the optimization focus shifts primarily to clustering rather than routing, which is undesirable. Balancing these aspects is crucial to ensuring that clustering simplifies the problem without overly constraining the solution space.

While multiple clustering algorithms based on proximity exist, such as the K-Means algorithm, the sweep algorithm is the state-of-the-art approach currently used by Vattenfall, and alternative clustering methods are not considered within the scope of this thesis. The following section explains and illustrates the implementation of the sweep algorithm in this framework.

The sweep algorithm used in this framework is a modified version of the sweep heuristic proposed by Cazzaro et al. [6]. The algorithm takes the following inputs: the set V , which consists of all nodes from the visibility graph representing the positions of turbines, the OSS, and anchors in the 2D plane; the precomputed first shortest paths from the OSS to each turbine; a user-defined cluster size $N_{cluster}$; and user-defined starting index.

Figure 2.4 illustrates the fundamental concept of the sweep algorithm. The process begins by calculating the angles of the first shortest paths relative to the vertical axis (north) at the OSS, which serves as the initial sorting point, as shown in Figure 2.4a. These paths are then sorted in a clockwise order based on their relative angles. As the sorting progresses, if multiple paths share a common edge, the end node of this edge becomes the new point around which the next sorting is performed, as seen in Figure 2.4b. In this figure, the first shortest paths to WTG 1 and WTG 4 share an edge that is defined by the OSS node and the anchor node of WTG 2. Hence, this anchor node becomes the new sorting point. A new sweep is conducted around this new sorting point, starting by tracing back along the common edge before continuing the sweep in a clockwise direction to sort the paths that share this edge. If multiple paths continue to overlap beyond this new sorting point, the algorithm progresses along the next common edge and repeats the process until all paths are distinguished. In this example, the first shortest paths to WTG 1 and WTG 4 do not share an additional edge, so the new sweep encounters the remainder of the first shortest path to WTG 1 first, and then WTG 4. The algorithm then returns to the initial sorting point at the OSS and resumes the clockwise sweep from the angle where the first common edge was detected. Ultimately, in this example, the sweep-order of turbines is determined as [1, 4, 3, 2].

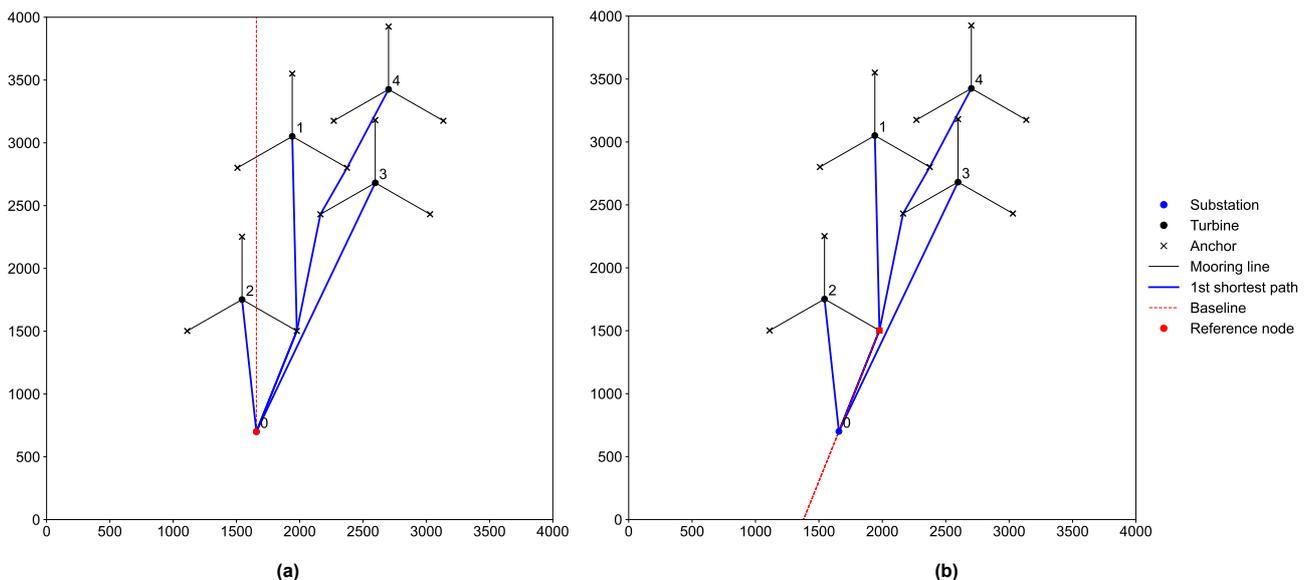


Figure 2.4: (a) Start of sweep algorithm with the OSS as initial sorting point and sweep line pointing north. (b) The sorting point is changed to the anchor node of WTG 2, with the sweep line extending back along the common edge between OSS and anchor node.

Once the turbines are sorted by their angular order, the algorithm assigns them to clusters. Clusters are filled sequentially based on the predefined cluster size $N_{cluster}$, starting from the chosen starting index. The first cluster consists of the first $N_{cluster}$ turbines from the starting index in the sorted list, the second cluster consists of the next $N_{cluster}$, and so on. If the total number of turbines N_{farm} is not an integer multiple of $N_{cluster}$, the last cluster contains the remaining turbines.

Figure 3.13 illustrates the impact of the starting index on cluster formation for the same wind farm layout as in Figure 2.4, with $N_{farm} = 4$ and $N_{cluster} = 2$, and the resulting sweep order $[1, 4, 3, 2]$. In Figure 3.13a, clustering begins with the first turbine in the sorted list, meaning the starting index is 0. Since $N_{cluster} = 2$, the first cluster consists of WTG 1 and WTG 4, while the second cluster contains WTG 3 and WTG 2. Figure 3.13b shows the effect of shifting the starting turbine to WTG 4 (starting index 1), which results in the first cluster containing WTG 4 and WTG 3, while the second cluster contains WTG 2 and WTG 1. Although shifting the starting index to the third or fourth turbine in the sweep-order list would result in different clusters by definition, the resulting clusters would still be one of the two combinations already illustrated in Figure 3.13. Since the clusters are solved individually and sequentially by the solver, the order in which they are solved does not impact the final result. Consequently, if N_{farm} is an integer multiple of $N_{cluster}$, the total number of unique clustering configurations is limited to $N_{cluster}$.

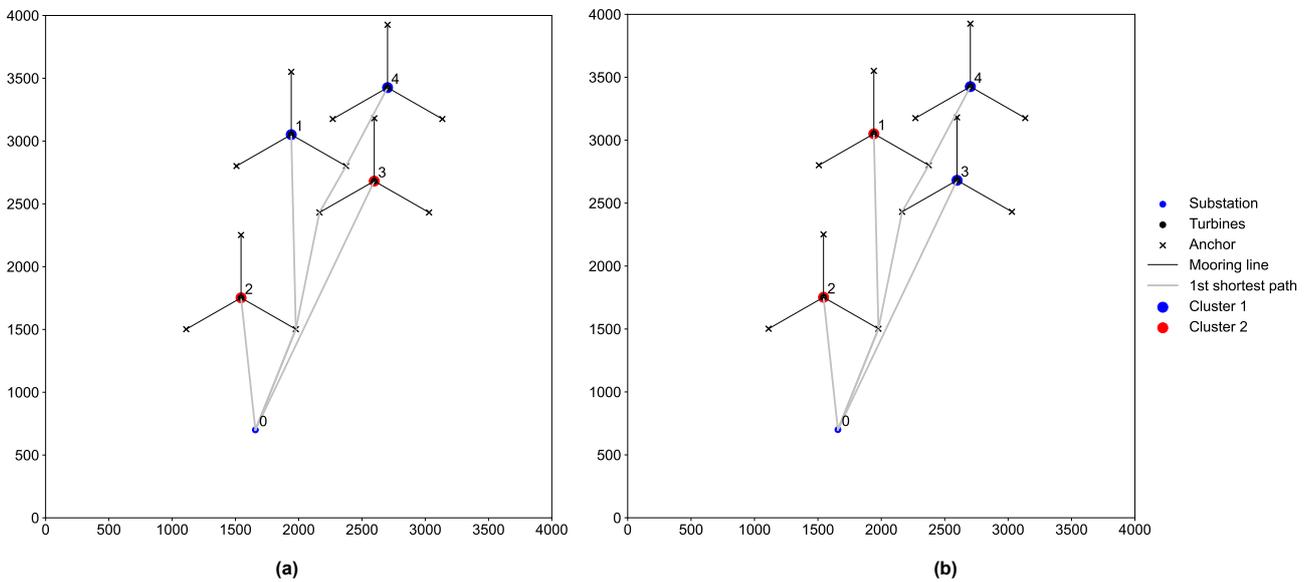


Figure 2.5: (a) Clustering configuration using the first-sorted turbine as starting index. (b) Alternative clustering configuration obtained by adjusting the starting index.

2.3.4. Cluster Boundary Creation

Originally, the WFCR was formulated as a single MILP model, solving the complete optimization in one step. However, as the number of turbines increases, the number of possible routing configurations grows factorially, leading to a rapid increase in computational complexity. To reduce the solution space, turbines are clustered, which reduces the number of possible configurations, the number of generated shortest paths, as these are no longer necessary for turbines in different clusters, and hence also the number of crossings between shortest paths. This restriction on the search space simplifies the optimization process and improves computational efficiency. Despite this reduction in solution space, the problem cannot yet be fully decomposed into independent subproblems, as crossings between shortest paths from different clusters may still occur.

To account for inter-cluster crossings, two main approaches can be considered. The first is to maintain the original formulation and solve the MILP model in one go. This ensures that the solver has full knowledge of all possible paths and constraints simultaneously, allowing it to find a feasible solution that satisfies all constraints, including inter-cluster crossings. However, this approach remains computationally demanding because MILP solvers rely on branch-and-bound and cutting-plane methods, which are highly sensitive to problem size. Solving a MILP model does not follow a simple linear increase in complexity with added variables and constraints. Instead, the solver must explore an extensive combinatorial search tree, where each additional variable and constraint increases the complexity exponentially. To mitigate this exponential growth

in computational complexity, the problem can be decomposed into smaller subproblems, effectively breaking the large search tree into a sequence of smaller trees. This sequential solving approach allows each cluster to be optimized individually, reducing the number of variables and constraints the solver must handle at any given time. However, as mentioned, due to the inter-cluster crossings, these subproblems cannot be solved independently, as previously solved clusters constrain the solution of subsequent clusters. Consequently, after optimizing the first cluster, its solution is stored, and additional constraints are dynamically introduced into the MILP model to ensure that paths selected in later clusters do not conflict with the fixed solution of the previous ones. This means that earlier decisions strongly influence later ones, potentially to suboptimal solutions if an early decision overly restricts the feasible space for later clusters. Therefore, the total outcome depends on the order in which the clusters are solved. It is important to note that in this context, "optimal" does not refer to the absolute best solution without clustering, but rather to the solution that is the best possible one (read: shortest total cable length) within the clustering approach.

To avoid these complications, an alternative approach is to define cluster boundaries that inherently eliminate inter-cluster crossings. This ensures that shortest paths from one cluster do not cross the boundaries of another cluster, preventing paths from different clusters from intersecting. By eliminating inter-cluster crossings by design, each cluster can now be optimized independently without referencing previous solutions, removing the need for dynamic constraints. In addition, these inter-cluster crossings do not need to be precomputed as a preprocessing step or considered during optimization. By enforcing boundaries during shortest path generation, the MILP model can truly be decomposed into independent subproblems, significantly simplifying the optimization process while ensuring that planarity constraints are met.

To generate the boundaries, a concave hull is first established around the cluster. To create the concave hull, the first step is to generate a triangulation of a set of input points representing the spatial locations around which the concave hull needs to be made. Figure 2.6b shows the Delaunay triangulation of the wind farm layout with $N_{farm} = 4$ in Figure 2.6a. The input points are the anchor nodes associated with the cluster's turbines and the OSS node. The Delaunay triangulation, which results in a set of non-overlapping triangles that cover the convex hull of the input points. Each triangle consists of three vertices, which correspond to the given input points. Additionally, each triangle has three neighboring triangles, except when an edge is part of the outer boundary of the convex hull, in which case it has no neighbor along that edge. This triangulation provides a framework for identifying and removing boundary edges.

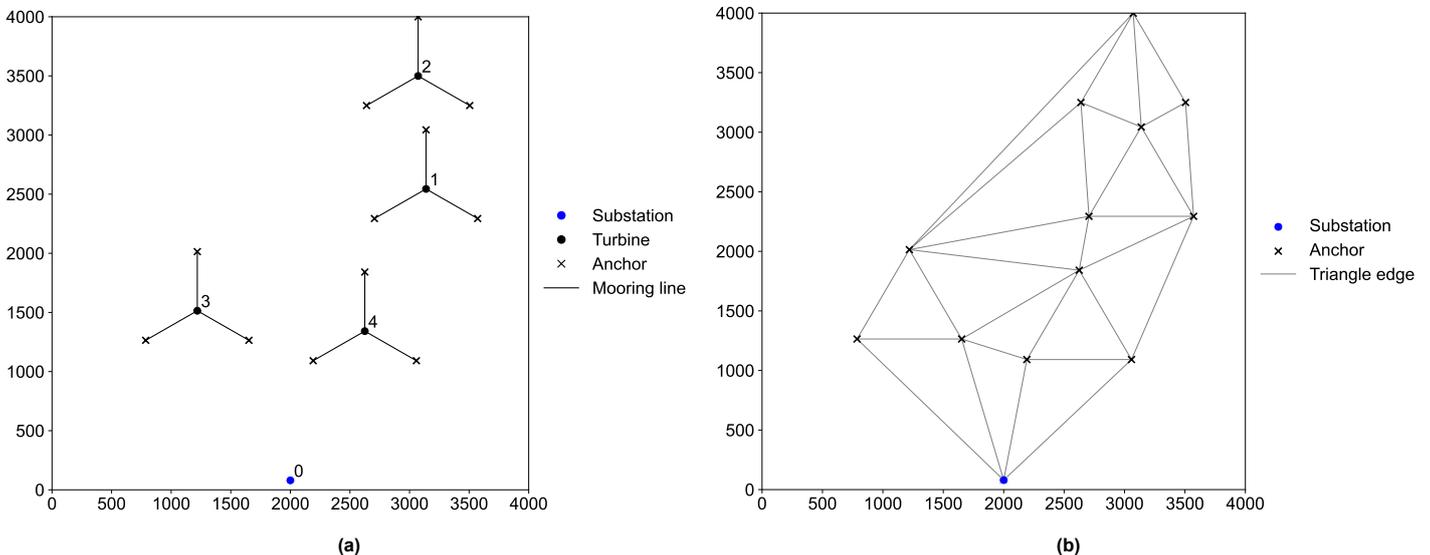


Figure 2.6: (a) Wind farm layout with $N_{farm} = 4$. (b) Delaunay triangulation based on OSS and anchor nodes of turbines in the layout of (a).

After triangulation, the algorithm detects the outer edges by identifying triangles that have at least one boundary edge, an edge with no neighboring triangle. Figure 2.7a shows the identified boundary edges of the Delaunay triangulation. These boundary edges form an initial polygonal boundary enclosing all the points. The key idea behind generating a concave hull is to iteratively refine the hull by removing long edges while

maintaining the connectivity of the hull. First, the boundary edges are sorted by length, and edges longer than a given boundary threshold length $L_{threshold}$ are marked for removal. Figure 2.7 shows the boundary edges marked for removal when $L_{threshold} = 1000$ m.

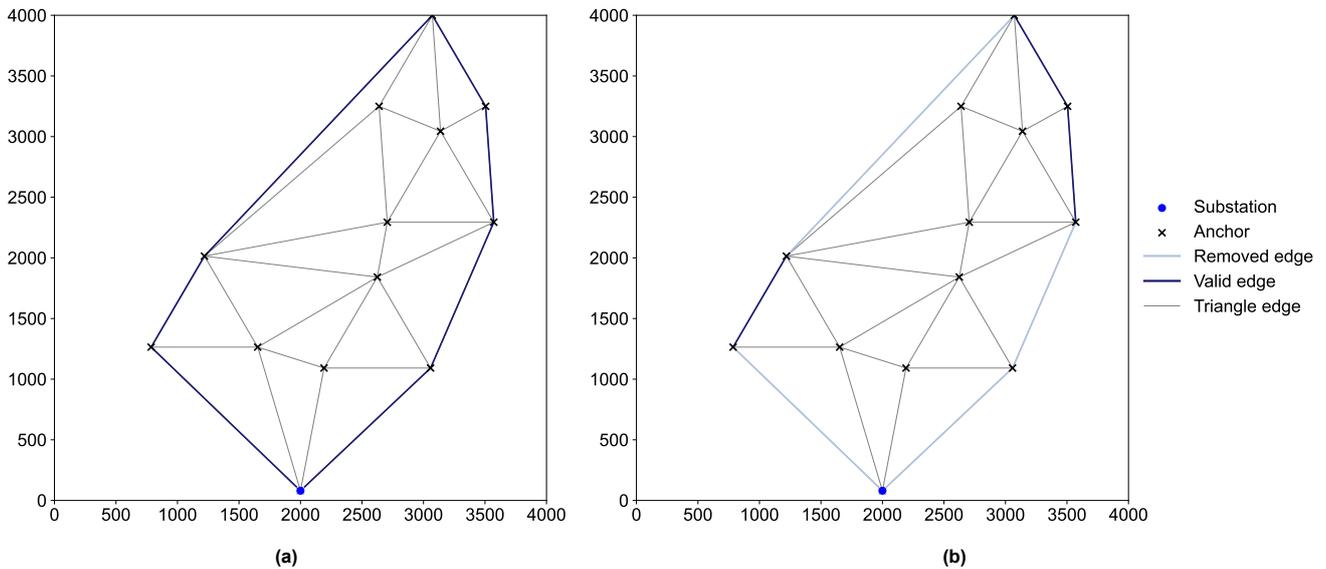


Figure 2.7: (a) Delaunay triangulation with identified boundary edges. (b) Boundary edges marked for removal when the threshold length is set to $L_{threshold} = 1000$ m.

If a triangle has only one boundary edge, i.e. it is connected to at least two other triangles, it is considered for removal. The algorithm then checks whether a triangle would disrupt the structure of the boundary. A triangle is removed only if its non-boundary vertex does not become a new boundary vertex in a way that would disrupt the overall boundary structure. Additionally, when removing triangles, the algorithm ensures that the remaining boundary does not create pinched regions where two sides of the hull meet at a single point. Once a triangle is removed, its neighbors are updated to reflect the change by marking its edges as new boundary edges. This process continues iteratively, with the longest boundary edges removed first, until no remaining boundary edge exceeds the length threshold. Figure 2.8 illustrates the resulting concave hull after iteratively removing all boundary edges that exceed the threshold for a threshold of $L_{threshold} = 1000$ m (Figure 2.8a) and $L_{threshold} = 2000$ m (Figure 2.8b).

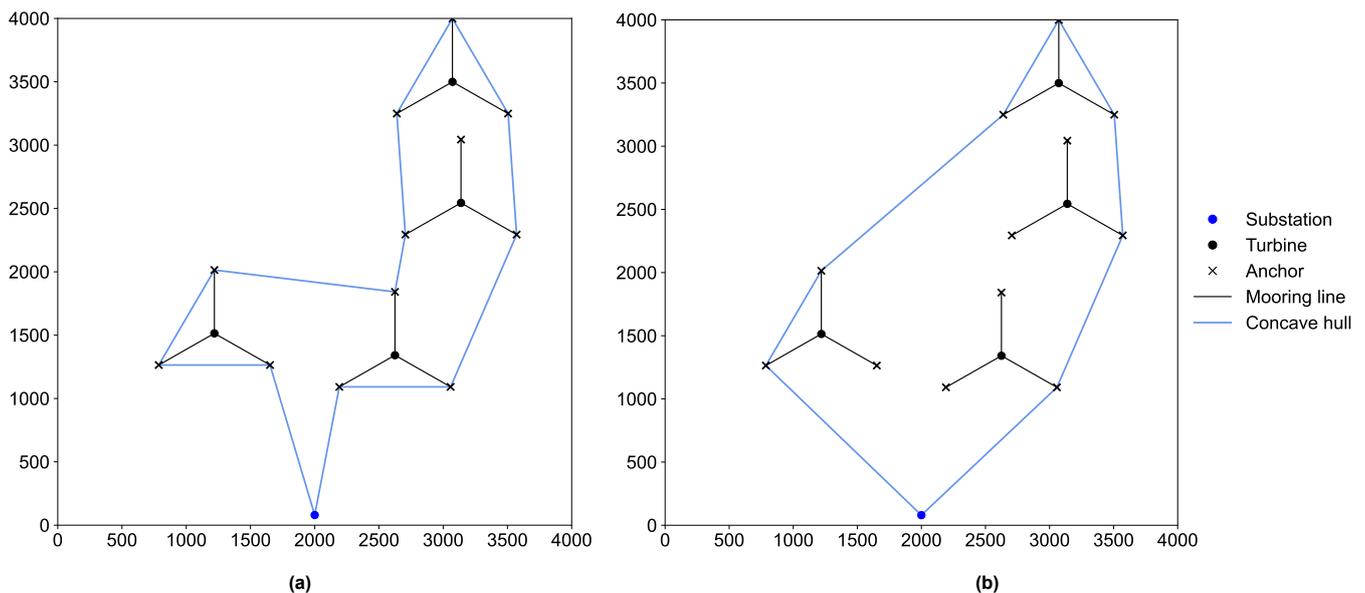


Figure 2.8: Final concave hull obtained after iterative boundary edge removal when (a) $L_{threshold} = 1000$ m, and (b) $L_{threshold} = 2000$ m.

Figure 2.9a shows the resulting concave hulls when the wind farm from Figure 2.6a is clustered into two clusters with $N_{cluster} = 2$ and $L_{threshold} = 10000$ m. Since the concave hulls for each cluster are generated independently, i.e. without considering the turbines and mooring systems from other clusters, the resulting boundaries may intersect mooring lines or concave hull edges from other clusters. This can introduce conflicts during optimization, as shortest paths generated within a cluster are not allowed to cross the boundaries of another cluster. To ensure that cluster boundaries account for the positions of turbines and mooring systems outside their own cluster, as illustrated in Figure 2.9b, Dijkstra's algorithm is used to replace the concave hull edges with the first shortest path that strictly adheres to a constrained set of nodes. This set of nodes is restricted to the cluster's own anchor nodes, the OSS, and concave hull nodes from other clusters. This restriction is essential to prevent situations where a cluster's boundary unintentionally encloses a turbine from another cluster, which could lead to infeasibilities in the routing solution. To enforce these restrictions within Dijkstra's algorithm, the original visibility graph is filtered by removing any edges where one or both nodes do not belong to the cluster's anchor nodes, the OSS, or the concave hull nodes of other clusters. This subgraph, along with the source and target node pairs from the cluster's concave hull edges, is then used as input to Dijkstra's algorithm to compute the first shortest path that can replace the concave hull edge. This process is repeated for each cluster.

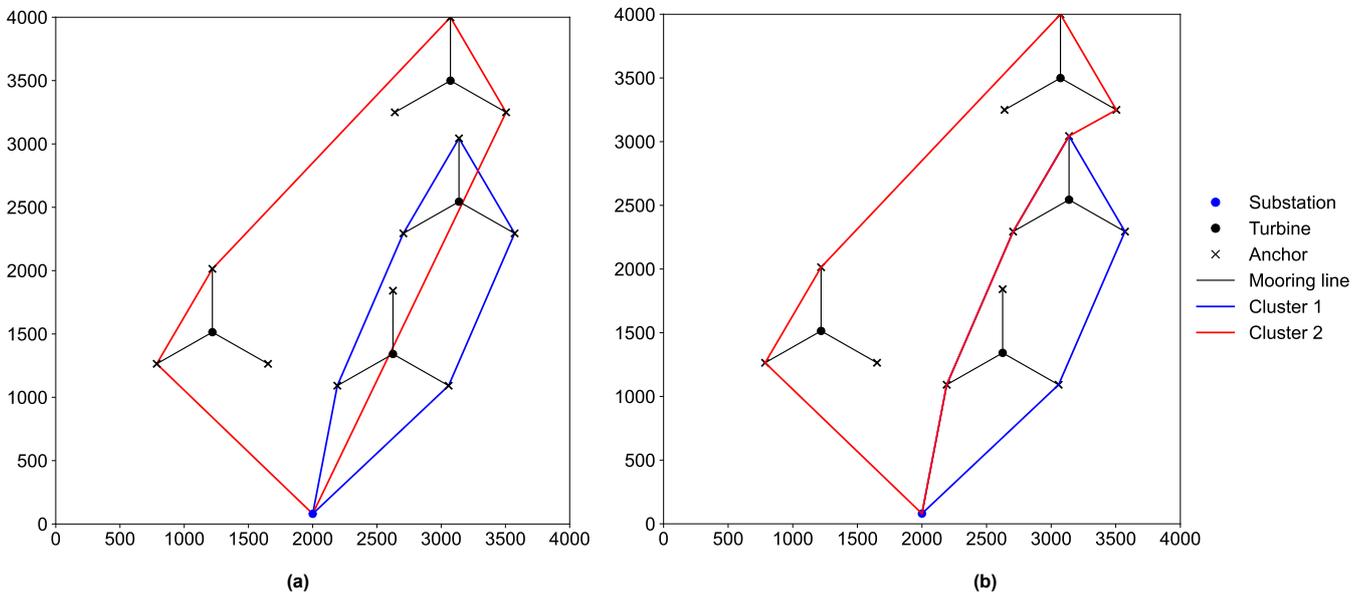


Figure 2.9: Transformation of concave hulls into final cluster boundaries. (a) Initial concave hulls generated independently for two clusters. (b) Final boundaries after replacing concave hull edges with shortest paths.

Although in Figure 2.9, the concave hull edges are replaced by shortest paths after applying a high length threshold, effectively reducing the hull to its convex hull, this does not mean that a convex hull would always be sufficient. For example, if both turbines from cluster 1 were moved slightly to the left, the convex hull edge from cluster 2 between the OSS and the southeast anchor of the turbine with the highest y-coordinate would likely be replaced by a shortest path looping entirely around cluster 1 to the right, rather than taking the leftward path as in the current configuration. This demonstrates that relying on convex hulls alone does not guarantee feasible boundaries in all cases.

2.3.5. Cluster-Specific Shortest Path Formation

Once the cluster boundaries are established, the k -shortest paths within each cluster are generated with certain restrictions. In the visibility graph for cluster X, only edges connecting nodes within the cluster, the substation, turbines, and anchor nodes of those turbines, are included, provided they do not cross the boundary of another cluster. In case another cluster's node is used to generate cluster X's boundaries, any edges linking cluster X's cluster nodes to this boundary node, although belonging to another cluster, are also allowed, as long as these edges do not cross another cluster's boundary. All other edges are removed from the visibility graph.

The resulting subgraph serves as input for Yen's algorithm, as discussed in Section 2.3.2, along with a list of

source and target nodes, which include all connection nodes (substation and turbine nodes), and the parameter k , which defines how many shortest paths need to be generated for each source-target pair. While the optimization framework supports multiple values of k , only the first shortest path ($k = 1$) is often chosen since the solver's ability to find a solution is not affected by k when the boundaries are properly defined for each cluster. Although a higher k could potentially lead to a better objective by reducing the total cable length, this is outside the scope of this thesis, as the primary research objective is to find a feasible solution.

2.3.6. Shortest Path Crossing Detection

Preventing IAC crossings is a critical constraint in optimizing the WFCR, as stated in Constraint C5 in Section 2.1. A key challenge in enforcing planarity constraints, such as avoiding cable crossings, is that the MILP model cannot dynamically check for cable crossings. Doing so would introduce nonlinear constraints, transforming the problem into a MINLP problem, which is significantly more complex to solve. To ensure that crossings between IACs do not occur, potential conflicts are identified before optimization begins, and constraints are introduced to prevent the solver from selecting conflicting paths. A crossing matrix is constructed as a preprocessing step and provided as input to the solver, allowing to determine whether any pair of shortest paths intersect.

Figure 2.10b illustrates an example of a crossing matrix resulting from the first shortest paths for a wind farm layout with $N_{farm} = 3$, shown in Figure 2.10a. The crossing matrix is a binary matrix where each element indicates whether two paths intersect: an entry of 1 signifies that the paths cross, i.e. at least one edge from one path intersects with an edge from another path, while an entry of 0 indicates no crossing. The rows and columns of the matrix are labeled according to the shortest path between two connection nodes, using the format (source node, target node, k), where k denotes whether it is the first, second, third, etc., shortest path. Constructing this matrix involves evaluating all possible shortest path pairs for intersections, except for paths with themselves or alternative k -shortest paths between the same two connection points, as only one path between two connection points can be selected at a time. These two exceptions are automatically assigned a zero entry.

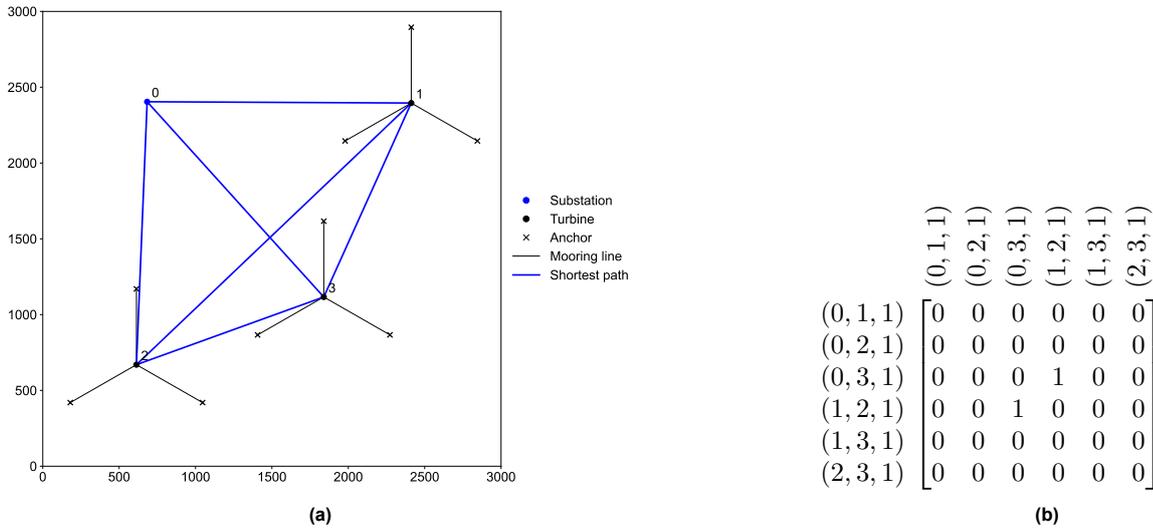


Figure 2.10: (a) First ($k = 1$) shortest paths for random wind farm layout with $N_{farm} = 3$ and the corresponding node labels to match the crossing matrix row and column headers. (b) The corresponding crossing matrix based on the first shortest paths from the layout in (a).

Each shortest path consists of one or more edges, each defined by two connection points: a source node and target node with coordinates in a 2D plane. Detecting intersections between two edges is a fundamental problem in computational geometry [32]. Consider two edges: edge P , defined by source node $P_1 = (x_1, y_1)$ and target node $P_2 = (x_2, y_2)$, and edge Q , defined by source node $Q_1 = (x_3, y_3)$ and target node $Q_2 = (x_4, y_4)$. The vectors representing these edges are:

$$\vec{P} = \begin{pmatrix} P_x \\ P_y \end{pmatrix} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix}, \quad \vec{Q} = \begin{pmatrix} Q_x \\ Q_y \end{pmatrix} = \begin{pmatrix} x_4 - x_3 \\ y_4 - y_3 \end{pmatrix} \quad (2.2)$$

To determine whether two edges intersect, the determinant of their 2D cross product is calculated:

$$\det = (y_4 - y_3) \cdot (x_2 - x_1) - (x_4 - x_3) \cdot (y_2 - y_1) \quad (2.3)$$

If $\det = 0$, the edges are either parallel or collinear. To distinguish between the two cases, the relative position of one edge with respect to the other is evaluated using the orientation parameters a and b :

$$a = (x_4 - x_3)(y_1 - y_3) - (y_4 - y_3)(x_1 - x_3) \quad (2.4)$$

$$b = (x_2 - x_1)(y_1 - y_3) - (y_2 - y_1)(x_1 - x_3) \quad (2.5)$$

If one of the orientation parameters a or b is non-zero ($\neq 0$), the edges are parallel and do not overlap. However, if $a = b = 0$, the edges are collinear, and further checks are needed to determine if they overlap. To confirm overlapping collinear edges, a bounding box check is performed:

$$\min(x_1, x_2) \leq \max(x_3, x_4) \quad \text{and} \quad \max(x_1, x_2) \geq \min(x_3, x_4) \quad (2.6)$$

$$\min(y_1, y_2) \leq \max(y_3, y_4) \quad \text{and} \quad \max(y_1, y_2) \geq \min(y_3, y_4) \quad (2.7)$$

If these conditions are satisfied, the edges overlap. However, overlapping edges are not considered crossings, as overlapping cables can be routed parallel to each other in practice. Moreover, since the shortest paths are constructed using the visibility graph, partially overlapping edges do not occur.

If $\det \neq 0$, the edges may intersect. The edges are considered to intersect if the following two conditions are satisfied:

$$0 \leq \frac{a}{\det} \leq 1 \quad (2.8)$$

$$0 \leq \frac{b}{\det} \leq 1 \quad (2.9)$$

Although intersections between path edges can be detected using traditional geometric methods as described above, additional checks are needed when paths share a common node. Figure 2.11 illustrates two cases where shortest paths pass through a common node but may or may not cross. These cases cannot be detected by standard geometric intersection checks because the intersection occurs at a node rather than along an edge. When a shared node is detected, all four edges connected to that shared node are examined. If any two edges overlap, the next edges along the paths are considered. The angles of all edges relative to the north are calculated and sorted. If edges from the two paths alternate in the sorted sequence around the shared node, the paths are considered to cross. This alternating pattern indicates that the paths spatially intertwine as they pass through the shared node. Conversely, if edges from the same path appear consecutively, the paths remain separate and can be routed parallel to each other.

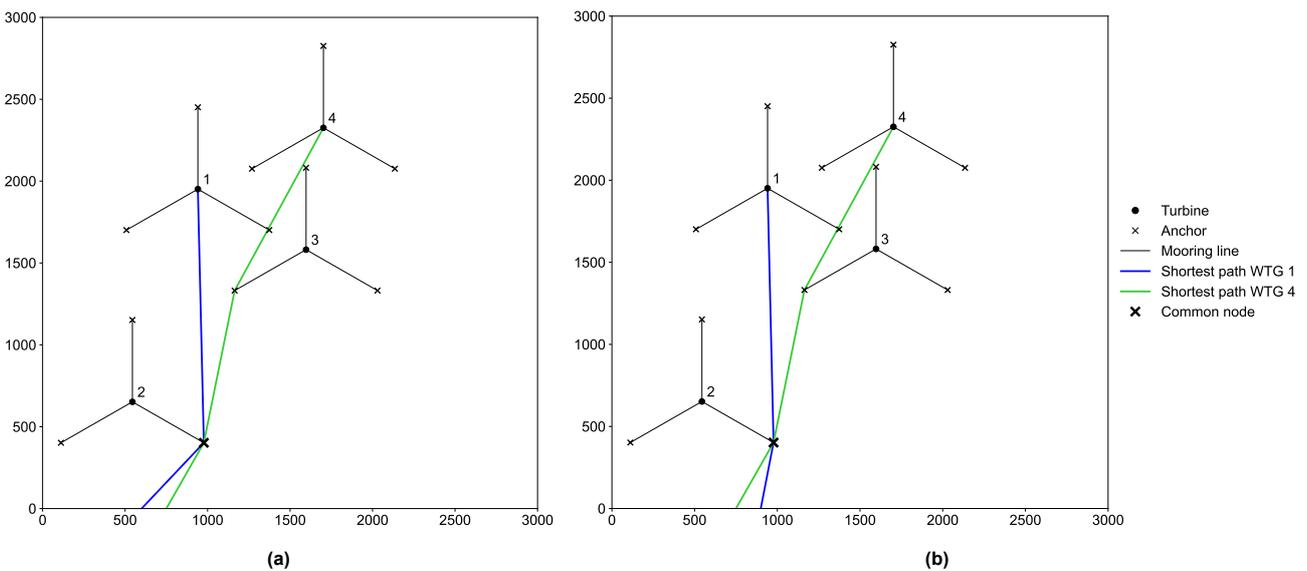


Figure 2.11: Examples of two shortest paths sharing a common node. (a) Paths do not cross as their path edges are adjacent. (b) Paths cross as their edges alternate.

2.4. Optimization

To solve the WFCR problem formulated in Section 2.1, a Mixed Integer Linear Programming (MILP) model is developed. The model is implemented in Python and solved using the open-source CBC solver and the PuLP library. Section 2.4.1 presents the complete mathematical MILP formulation as implemented, including the objective function, variables and constraints. In Section 2.4.2, the constraint that ensures a loop topology and prevents unintended cycles in this MILP model, known as the MTZ subtour elimination constraint, is further explained.

2.4.1. Mixed Integer Linear Programming Model

Let \mathcal{N} be the set of turbines to connect to substation s and $\mathcal{N}_s := \mathcal{N} \cup \{s\}$ the set of turbines and the substation. The objective of the optimization is to minimize the cable length, which can be mathematically expressed as:

$$\min \sum_{k=1}^{\mathcal{B}} \sum_{(i,j) \in \mathcal{N}_s} d_{ij} x_{ij}^k \quad (2.10)$$

where $x_{ij}^k \in \{0, 1\}$ is a binary decision variable that has value 1 if a shortest path (read: cable connection) between nodes i and j is used in loop k , where $i \neq j$. The parameter d_{ij} represents the shortest path distance between nodes i and j . The set \mathcal{B} consists of all loops, where $\mathcal{B} = \lceil |\mathcal{N}| / C \rceil$, and C is the cable capacity (sometimes also referred to as loop size). The objective is subject to a set of constraints, ensuring compliance with the problem formulation in Section 2.1:

C1. To ensure all generated electricity is transferred to the substation:

- o Ensure selected paths match the turbines in the loop:

$$t_j^k = \sum_{i \in \mathcal{N}_s} x_{ij}^k \quad \forall j \in \mathcal{N}, \forall k \quad (2.11)$$

where $t_j^k \in \{0, 1\}$ is a binary decision variable that is 1 if turbine j is included in loop k , and 0 otherwise.

- o Each turbine is assigned to exactly one loop:

$$\sum_{k=1}^{\mathcal{B}} t_j^k = 1 \quad \forall j \in \mathcal{N} \quad (2.12)$$

- o The total demand in each loop does not exceed the cable capacity C :

$$\sum_{j \in \mathcal{N}} t_j^k \leq C \quad \forall k \quad (2.13)$$

C2. To enforce a loop topology, the following constraints are enforced:

- o Ensuring connectivity within the loop, this constraint guarantees that each node maintains an equal number of entering and exiting cables:

$$\sum_{(i,j) \in \mathcal{N}_s} x_{ij}^k - \sum_{(i,j) \in \mathcal{N}_s} x_{ji}^k = 0 \quad \forall k \quad (2.14)$$

- o Avoid selecting both directions of the same path, as this will result in self-loops:

$$x_{ij}^k + x_{ji}^k \leq 1 \quad \forall (i, j) \in \mathcal{N}_s, i \neq j, \forall k \quad (2.15)$$

- o Eliminate subtours using MTZ constraints, preventing unintended self-loops and ensures the sequence of connections forms a valid loop that includes the substation:

$$u_j^k \geq u_i^k + 1 - M(1 - x_{ij}^k) \quad \forall i \neq j, \forall k \quad (2.16)$$

where u_j^k is a continuous decision variable that resembles the position of node j in the sequence of loop k , and M is a large constant set at $M = |\mathcal{N}|$ used for subtour elimination. The working principle and motivation behind using MTZ subtour elimination constraints is further elaborated on in Section 2.4.2.

C3. To ensure that balanced cable routing is applied, meaning that all loops should support an equal amount of turbines, or can vary at most by one turbine if the total number is not divisible evenly among the loops:

- An upper and lower bound on the amount of turbines in each loop is imposed due to cable capacity constraints:

$$\left| \sum_{j \in \mathcal{N}} t_j^k - \sum_{j \in \mathcal{N}} t_j^m \right| \leq 1 \quad \forall k, m, k \neq m \quad (2.17)$$

C4. Crossings of mooring lines are handled during preprocessing by removing edges that would violate this constraint from the visibility graph, as explained in Section 2.3.1. Thus, no additional constraints are needed in the model.

C5. Inter-cable crossings are handled by creating boundaries around clusters, and subsequently creating shortest paths that do not cross boundaries of other clusters, thereby inherently not creating any paths that result in crossings. Intra-cable crossings are handled by detecting potential crossings in preprocessing as explained in Section 2.3.6, and stored in a crossing matrix which is then used as input to the model.

- For each detected crossing pair (i, j) and (m, n) , at most one of the paths can be selected:

$$\sum_{k=1}^B x_{ij}^k + x_{mn}^k \leq 1 \quad \forall (\{i, j\}, \{m, n\}) \in \mathcal{C} \quad (2.18)$$

where \mathcal{C} is the set of all cable pairs that cross each other.

2.4.2. Miller-Tucker-Zemlin Subtour Elimination Constraint

The MTZ subtour elimination constraints are a technique in MILP designed to prevent subtours in routing problems. These constraints ensure that the selected paths form a single connected loop that includes the OSS, as in Figure 2.12a, and to not create isolated cycled that bypass it, as shown in Figure 2.12a. The fundamental idea behind the MTZ constraints is to introduce an ordering variable u_j^k , which represents the position of turbine j in the sequence of the loop k . This variable enforces a strict progression of nodes, ensuring that each turbine follows a structured path rather than forming independent closed cycles. As mentioned in Section 2.4.1, constraint (??) is implemented to achieve this:

$$u_j^k \geq u_i^k + 1 - M(1 - x_{ij}^k) \quad \forall i \neq j, \forall k \quad (2.19)$$

where u_j^k is a continuous decision variable that resembles the position of node j in the sequence of loop k , and M is a large constant typically set to a value greater than or equal to the total number of turbines in the loop; in this case, in this case $M = |\mathcal{N}|$. If a cable is used between nodes i and j , meaning $x_{ij}^k = 1$, the constraint simplifies to $u_j^k \geq u_i^k + 1$, ensuring that node j always has a strictly greater position index than node i , thereby enforcing a sequential order in the loop. If a cable is not used between nodes i and j , meaning $x_{ij}^k = 0$, the constraint becomes inactive, since M is a large number, preventing any unintended restrictions on the order of unconnected nodes.

To illustrate the mechanism of the MTZ constraint, consider an example where a cluster X consists of the following turbines eight turbines, labeled 1 to 8, and the OSS, labeled 0. The cable capacity $C = 4$, enforcing the MILP to make loops consisting of a maximum of four turbines. The goal is to establish valid cable connections forming loops while ensuring no subtours occur. Suppose the solver initially selects connections such that:

$$x_{0,1} = 1, \quad x_{1,3} = 1, \quad x_{3,5} = 1, \quad x_{5,7} = 1, \quad x_{7,0} = 1 \quad (2.20)$$

and

$$x_{0,2} = 1, \quad x_{2,4} = 1, \quad x_{4,6} = 1, \quad x_{6,8} = 1, \quad x_{8,0} = 1 \quad (2.21)$$

These sequences form two distinct loops, both starting and ending at the substation. The MTZ constraints enforce the following ordering variables:

$$u_1 \geq u_0 + 1, \quad u_3 \geq u_1 + 1, \quad u_5 \geq u_3 + 1, \quad u_7 \geq u_5 + 1 \quad (2.22)$$

and

$$u_2 \geq u_0 + 1, \quad u_4 \geq u_2 + 1, \quad u_6 \geq u_4 + 1, \quad u_8 \geq u_6 + 1 \quad (2.23)$$

This ensures a structured order within each loop, preventing any turbine from forming an independent subtour. If an unintended cycle were to occur, such as a loop between turbines 3, 5, and 7, without passing through the substation, the MTZ constraints would prevent a feasible solution by requiring a strict increasing sequence of positions. Hence, the constraints would dictate that $u_3 \geq u_5 + 1$, $u_5 \geq u_7 + 1$, and $u_7 \geq u_3 + 1$. This results in a contradiction, as it forces a cyclic dependency where each node must be both greater and smaller than another, making the solution infeasible. By maintaining this strict ordering, the MTZ constraints ensure that every turbine remains correctly assigned to its respective loop and ultimately connected to the OSS.

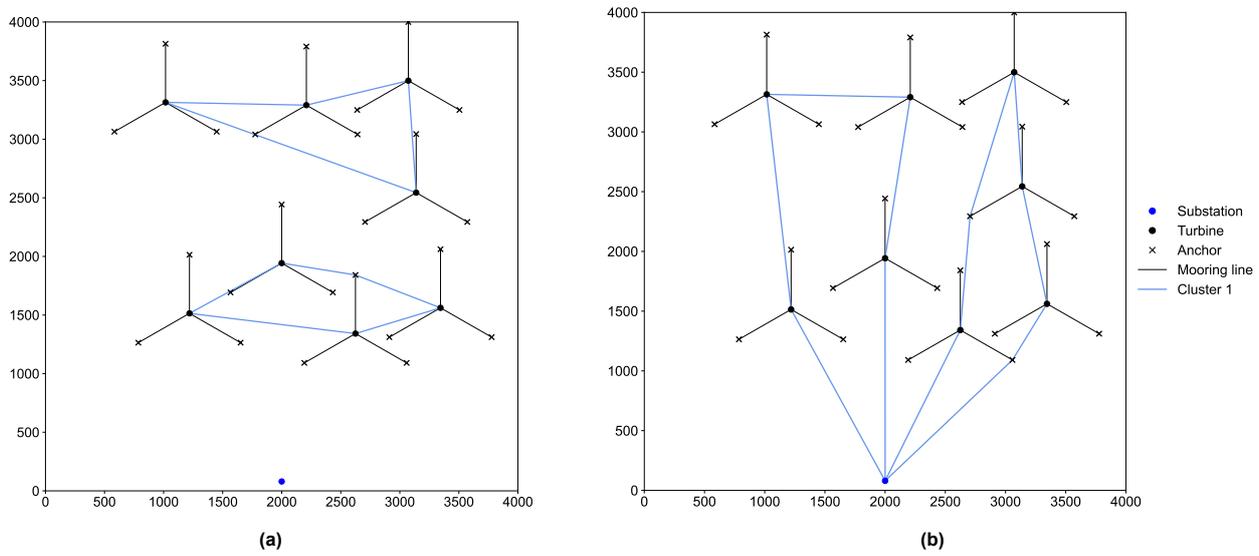


Figure 2.12: Optimized cable routing for random wind farm layout with $N_{farm} = 8$ and $N_{loop} = 4$ in case (a) no MTZ subtour elimination constraints are made, and (b) with MTZ subtour elimination constraints.

Compared to other subtour elimination methods, MTZ constraints are straightforward to implement, requiring only one additional constraint and one extra variable per node. In contrast, flow-based approaches typically introduce multiple constraints and auxiliary variables to track the flow between nodes, significantly increasing the problem's complexity and computational burden. This makes flow-based methods less efficient and less scalable for large-scale WFCR problems, where minimizing computational complexity is crucial.

2.5. Postprocessing

While the MILP model optimizes cable routing based on predefined constraints, it does not inherently verify whether the solutions comply with the industry standard for a minimum clearance distance. Ensuring compliance with this industry standard is crucial for the feasibility and safety of the cable layout. Since dynamically enforcing this constraint within the optimization process would significantly increase computational complexity, a postprocessing step is introduced. This step evaluates whether the optimized cable routing adheres to the required minimum clearance distance between mooring systems and IACs, as well as between IACs themselves. If violations are detected, adjustments can be made to refine the solution, such as modifying cluster and loop sizes, reclustering, or adjusting the wind farm layout. The remainder of the section explains how compliance with this industry standard is checked.

To ensure that IACs not only comply with the non-crossing requirement of Constraint C5 but also with the minimum clearance distance d_{min} , it must be ensured that if one or multiple cables are routed between two

turbines, they are able to maintain the required spacing between themselves and from the mooring system. Since anchors "extend" from the turbines, the shortest distance between two anchors of different turbines represents the narrowest space through which cables must be routed. This space between two anchors of different turbines is referred to as passages. To maintain the minimum clearance distance, the width of each passage must be determined to establish the maximum number of cables that can be routed through it. Since the anchor nodes of turbines are part of the set of nodes in the visibility graph, as explain in Section 2.3.1, and edges have weights based on their Euclidean distance, all edges between anchor points of different turbines can be used to define passages and calculate the maximum number of cables that they can accommodate:

$$C_{max,allowed} = \left\lfloor \frac{L_p - d_{min}}{d_{min}} \right\rfloor \quad (2.24)$$

where $C_{max,allowed}$ represents the maximum number of cables allowed through a passage, L_p is the length or width of the passage, and d_{min} is the minimum distance that must be maintained between IACs and between an IAC and anchor, as defined by industry standards.

The MILP model itself does not inherently verify whether this constraint is satisfied within the optimization process. To enforce it within the model, it is necessary to check which shortest paths potentially pass through specific passages. MILP model cannot dynamically check for cable crossings, as doing so would introduce nonlinear constraints, converting the problem into a MINLP, which is much harder to solve efficiently. Therefore, similar to the approach used for handling the crossing constraint, an input can be provided that lists all passages along with their widths and the cables that could be routed through them. A constraint can then be introduced to ensure that no more paths are selected simultaneously than the allowed number of cables. However, the number of passages to be checked increases significantly as the problem scales. Additionally, as with inter-cluster crossings, the number of cables that can be routed through a passage is dependent on other clusters. This interdependency still necessitates dynamically updating constraints. A purely cluster-independent MILP approach cannot accommodate such interdependencies, leading to infeasibility if subproblems are solved independently but violate passage capacity limits. Therefore, it is suggested to check for compliance with this industry standard as a postprocessing step and then manually refine the solution by updating variables such as cluster and loops size, reclustering, changing the wind farm layout, to name a few. The remainder of this section will explain how the passages are checked for compliance with this industry standard.

Firstly, standard straight-line paths that cross a passage can be identified using traditional geometric methods, as described in Section 2.3.6, and included in the list of cables routed through the passage. However, due to the inclusion of anchor nodes in the visibility graph, a cable route may pass directly through an anchor. In reality, the cable must deviate from the anchor while maintaining the minimum clearance distance. The direction in which the cable needs to adjust determines which passages from that anchor it will cross. These types of paths are referred to as paths with intermediate nodes, where the intermediate node represents the anchor through which the cable is routed. Additionally, cables may connect to an anchor node of a turbine before linking to the turbine itself. However, such cables can be routed either above or below, or left or right of the anchor, determining which passage the cable passes through. For example, if a cable enters a turbine at precisely 0 degrees from the north, it must be routed either to the left or right around the anchor pointing at 0 degrees north. These paths are referred to as paths with entry or exit nodes. The following sections first discuss paths with intermediate nodes, followed by paths with entry or exit nodes.

In Figure 2.13, the shortest path to WTG 4 can be seen. Rather than following a direct straight-line route, as this is not possible in this case, the path utilizes an anchor node belonging to WTG 1 as part of its route. Such an anchor node, which is part of the shortest path between two other nodes, is referred to as an intermediate node. In this case, the passage between the intermediate node of WTG 1 and another anchor node belonging to WTG 3 is examined in Figure 2.13a. When considering a small deviation of the actual cable routing from the shortest path to adhere to the required clearance distance to the anchor, it moves into the passage and it is clear that it crosses it. In Figure 2.13b, another passage, this time between the same intermediate node as in Figure 2.13a and an anchor node from WTG 4, is highlighted. When considering a small deviation of the actual cable routing from the shortest path to adhere to the required clearance distance to the anchor, it remains outside the passage, making it clear that the path does not cross it. To ensure that these cases involving intermediate nodes are handled correctly, a relative angle comparison algorithm is applied.

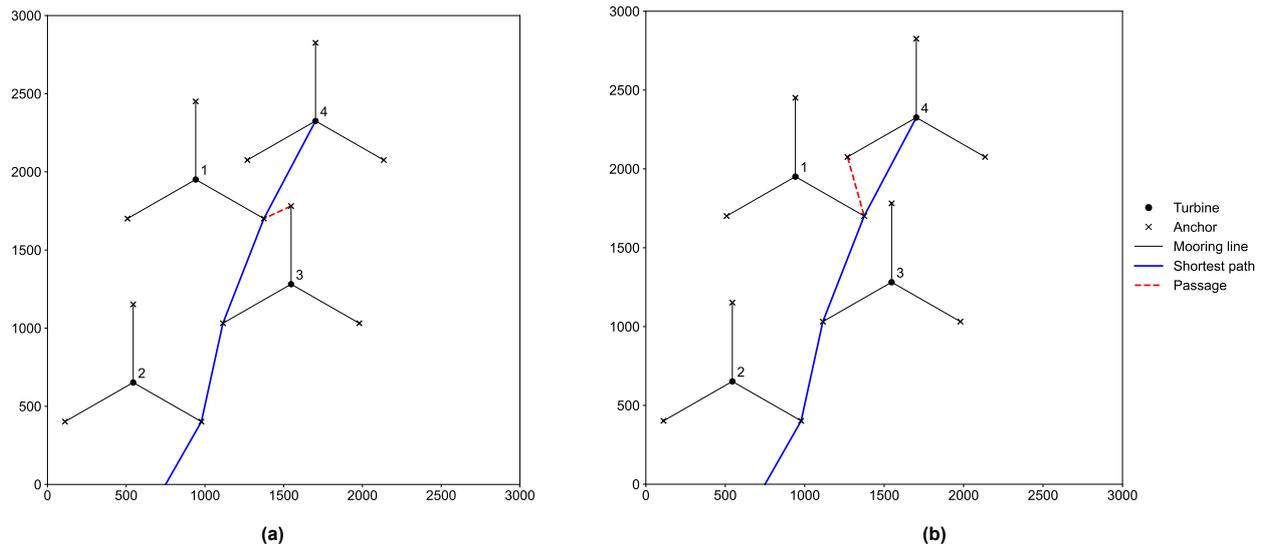


Figure 2.13: Passage crossings checked for special case where a shortest path uses an intermediate node. **(a)** The shortest path crosses the passage between anchor nodes of WTG 1 and WTG 3; **(b)** The shortest path does not cross the passage between anchor nodes of WTG 1 and WTG 4.

This algorithm involves comparing angles between edges of the path relative to the intermediate node. The algorithm uses four key edges in this scenario: 1. previous path edge, or reference edge, which connects the previous node in the path to the intermediate node; 2. next path edge, which connects the intermediate node to the next node on the path; 3. passage edge, which is the edge that defines the passage, connecting the intermediate node and another anchor node or a substation node; and 4. mooring line edge, which is the edge connecting the intermediate node and the center of the WTG the intermediate node belongs to. First, for each of these edges, the algorithm calculates the clockwise angle relative to the reference edge. Once the angles are calculated, they are sorted in ascending order. This sorted list indicates the relative positions of the edges in clockwise direction around the reference edge. Whenever the angle between the reference edge and the mooring line edge, θ_m , and the angle between the reference edge and the passage edge, θ_p , are adjacent in this sorted list, the shortest path does not cross the passage. However, if those two angles are not adjacent, the path is considered to cross the passage.

The second case that requires special attention involves paths that use an entry or exit node in their shortest path. In Figure 2.14, a passage between WTG 2 and WTG 3 is highlighted. The first shortest path from the OSS to WTG 2 is a straight-line. However, the second shortest path from the OSS to WTG 2 utilizes an anchor node from WTG 2 as an entry point to the turbine, which qualifies this anchor node as an entry/exit node. In reality, this second shortest path must deviate slightly above the entry node to maintain the required clearance distance. Routing below the entry node would result in the same trajectory as the first shortest path, which would not resolve the constraint violation that led to the selection of the second shortest path in the first place. As a result, the necessary deviation causes the path to cross the passage between WTG 2 and WTG 3.

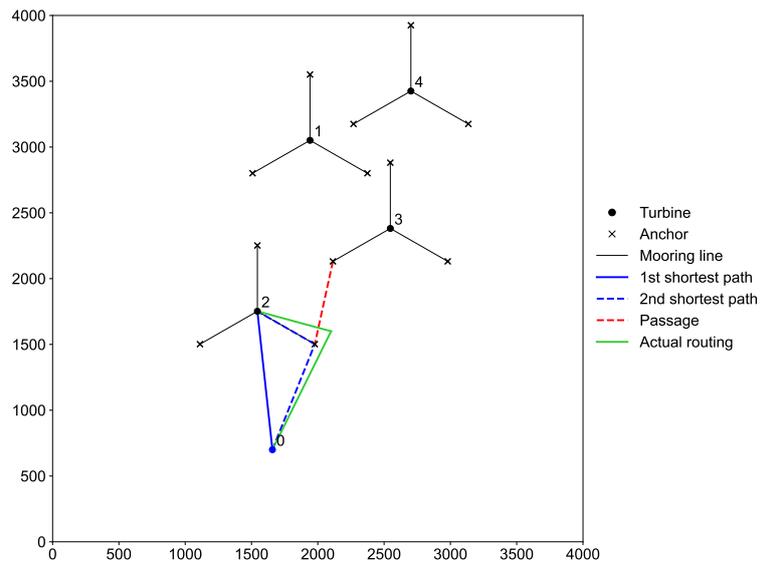


Figure 2.14: Shortest path uses entry node that belongs to passage.

Since the algorithm cannot determine in which direction the path will be routed around the entry or exit node in reality, it is added to the list of all possible passages it could pass through if it deviates from the anchor. This ensures that every potential crossing scenario is considered, accounting for both possible routing directions around the entry or exit node.

3

Results and Discussion

This chapter presents the results obtained from the MILP-based optimization framework as explained in Chapter 2 Methodology. Section 3.1 provides a detailed description of the scope and modeling assumptions. The case study in Section 3.2 illustrates the working principle of the tool on a real-life wind farm: the London Array wind farm. Section 3.3 emphasizes the challenges for finding feasible solutions introduced by mooring system constraints and the loop topology by analyzing the occurrence of passage violations and proposing adjustments to resolve these issues. Finally, Section 3.4 discusses the limitations and sensitivities of the optimization framework due to parameters in the turbine clustering and cluster boundary formation algorithms.

3.1. Site Description

The FOWFs to be routed in this thesis are based on predefined wind farm layouts, which are either randomly generated or manually specified to illustrate specific scenarios. A single substation is assumed to simplify the problem and eliminate the need for turbine partitioning, as the inclusion of mooring systems does not fundamentally alter the partitioning problem compared to BFOWFs. The positions of the OSS and turbines are fixed and not subject to optimization.

Each turbine is equipped with a mooring system that is explicitly modeled in the layout. This thesis assumes a uniform seabed, disregarding site-specific characteristic such as soil conditions, and considers depth irrelevant, effectively restricting the site and mooring system to a 2D plane. The mooring system comprises of three mooring lines and a corresponding number of anchors. This configuration offers geometrical stability, redundancy, and cost-efficiency [17]. Equal angular spacing of the mooring lines in the plane is crucial for maintaining the structural integrity of the turbines, as it enables the system to resist wind and wave forces from multiple directions [22]. Therefore, the mooring lines and anchors are distributed evenly around each turbine with an angular spacing $\phi = 120$ degrees. The orientation of the mooring systems is consistent across all turbines in the layout, with the mooring line at 0 degrees north designated as the upwind line. This alignment assumes that the upwind line is oriented to withstand the predominant environmental wind and wave forces, which are assumed to come from the north [9]. The structural stability of the turbine also depends on specifying both minimum and maximum mooring line lengths. While the selection of mooring line and anchor types is beyond the scope of this thesis, a catenary mooring configuration is assumed. The horizontal projection of the mooring line, referred to as the anchor radius, is set to $R_{anchor} = 500$ m. The actual mooring line length $l_{mooring}$ is typically defined as a function of water depth h , expressed through the scope S :

$$S = \frac{l_{mooring}}{h} \quad (3.1)$$

The scope of a catenary mooring system is typically between 7:1 and 5:1 [5]. The mooring line length can be approximated by:

$$l_{mooring} = \sqrt{R_{anchor}^2 + h^2} \quad (3.2)$$

Hence, the water depth can be approximated with:

$$h = \sqrt{\frac{R_{anchor}^2}{S^2 - 1}} \quad (3.3)$$

Substituting the assumed R_{anchor} and common values for the scope, this would mean that these FOWFs are located at sites with a water depth between roughly 70-100 m, which is realistic for a FOWF, as can be seen in Table 3.1. Although factors such as anchor type, environmental conditions, and turbine platform design can refine the required mooring line length, water depth remains the primary parameter determining the baseline requirement. Conversely, the maximum allowable mooring line length is constrained to manage costs and minimize slack, which could introduce instability and uncertainty [22].

Wind farm	Capacity	Turbines	Rotor diameter	Water depth
Hywind Scotland	30 MW	5 x Siemens SG 6 MW	154 m	90-120 m
WindFloat Atlantic	25 MW	3 x Vestas 8.4 MW	164 m	100 m
Kincardine Offshore Wind Farm	50 MW	5 x Vestas 9.5 MW, 1 x 2 MW	164 m	60-80 m
Hywind Tampen	88 MW	11 x Siemens Gamesa 8.6 MW	167 m	260-300 m

Table 3.1: Overview of wind farm parameters for commercially commissioned FOWFs.

The layout of the wind farm depends on the placement of the OSS and its associated field boundaries. As shown in Figure 3.1, two different substation placement scenarios for generating wind farm layouts are considered in this thesis: a circular template with the OSS centrally located (Figure 3.1a) and a rectangular template with the OSS positioned near the field boundary (Figure 3.1b). It is important to note that while these templates define specific field boundaries for turbine placement, they do not impose strict limitations on cable routing. In theory, cables could extend beyond the defined field boundaries, though this is not expected to occur frequently in practice. In both scenarios, turbines are restricted to placement within the green-shaded area, also referred to as the allowed turbine area, A_{farm} . The white buffer zone between the allowed turbine area and the field boundary has a width equal to R_{anchor} , ensuring that all nodes and paths remain within the field boundary. In both substation placement scenarios, an exclusion zone around the substation is defined by the parameters R_{anchor} and d_{OSS} . The parameter R_{anchor} accounts for mooring system radius, while d_{OSS} ensures a minimum distance between the OSS and any anchors.

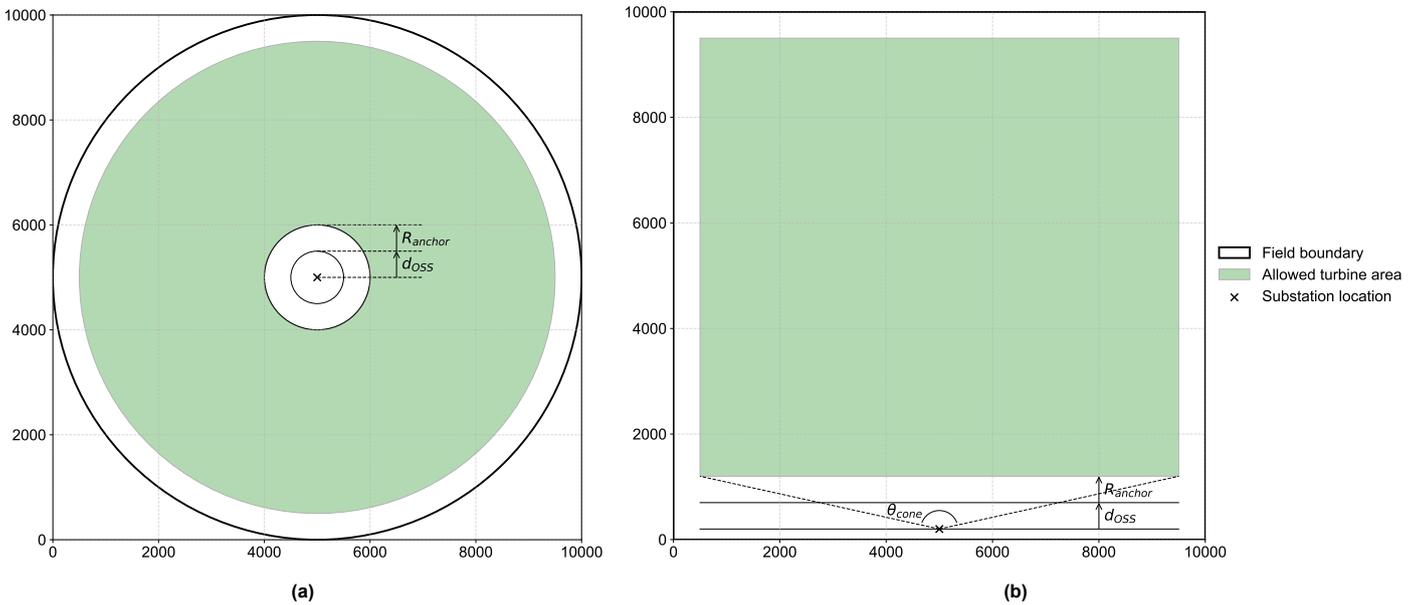


Figure 3.1: Substation placement scenarios with their corresponding allowed turbine areas A_{farm} : (a) Centrally-located substation; (b) Substation located near the field boundary.

A schematic illustrating the geometric parameters of neighboring turbines is provided in Figure 3.2. The minimum clearance distance d_{min} that should be applied according to industry standards between IACs or an IAC and the mooring system is 100 m. This distance is also applied between two anchors of different mooring systems to prevent soil weakening [22]. The distance between two anchor points of the same turbine is denoted as L_{anchor} and the footprint of the mooring system, or the area that is taken up by one FOWT is denoted as $A_{mooring}$. The horizontal distance between the centers of two turbines is denoted as D , while the

layout, ensuring that all turbines in all clusters are routed within a single loop. Since the London Array wind farm consists of BFOWTs, the model is set up with $R_{anchor} = 1$ m to mimic bottom-fixed foundations.

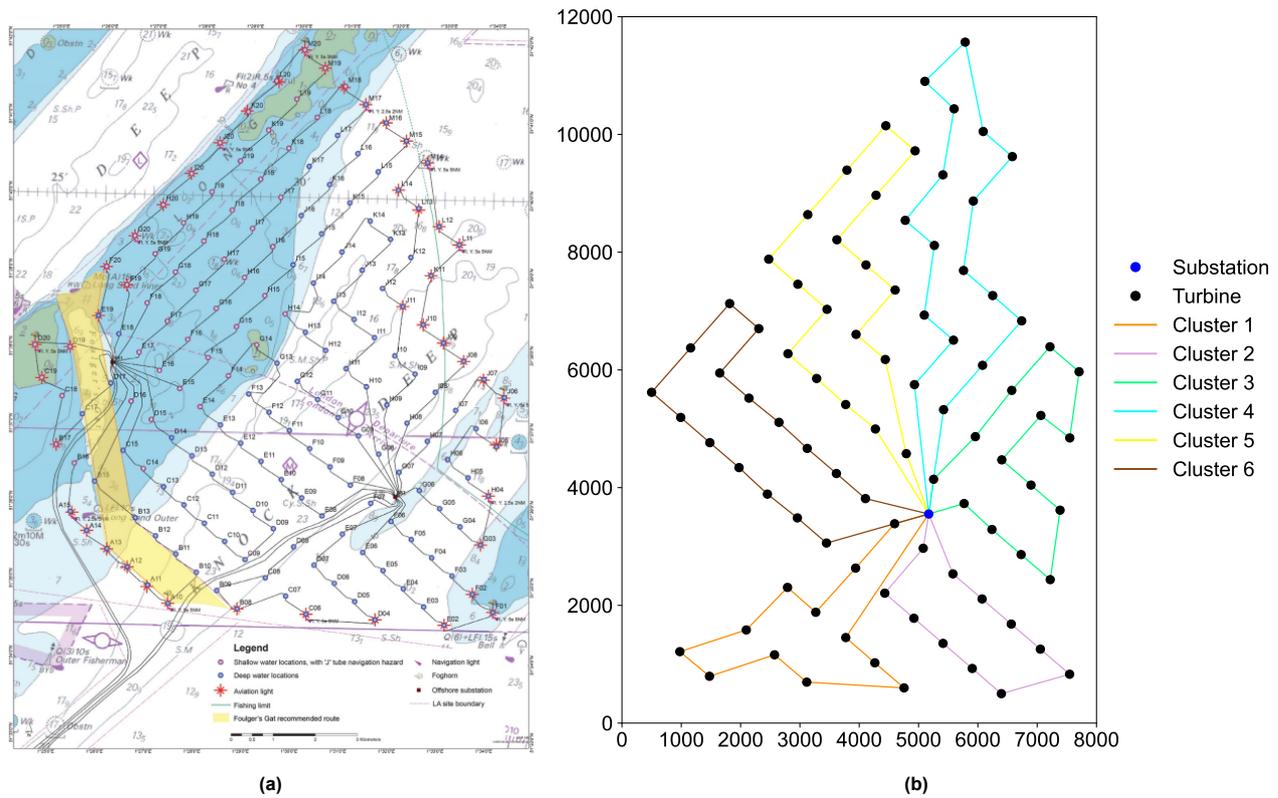


Figure 3.3: (a) Original farm and cable layout for the London Array BFOWF; (b) Optimized cable layout for the London Array BFOWF.

The optimized cable layout generated by the MILP model is shown in Figure 3.3b. Comparing the original cable layout and the optimized routing reveals that the routing for clusters 3, 4, 5, and 6 remains unchanged, while clusters 1 and 2 show minor deviations from the original routing. The total cable lengths of cluster 1 and 2 in both layouts are summarized in Table 3.2. A manual calculation of the original cable routing suggests that the MILP-optimized layout results in a shorter cable length for these clusters. However, it is important to consider that total cable length was likely not the sole objective in the original cable routing design of the London Array wind farm. Additional factors, such as obstacles, seabed conditions, or accessibility for maintenance, may have influenced the final routing decisions, potentially making the optimized cable routing proposed in Figure 3.3b infeasible in practice. However, the fact that the MILP model produces a cable routing with a reduced total cable length, while preserving most of the original structure, demonstrates that the optimization framework functions as intended.

Cluster	Original layout	Optimized layout	Difference
Cluster 1	13563 m	13415 m	-1.09 %
Cluster 2	9192 m	9090 m	-1.11 %

Table 3.2: Comparison of original and optimized cable lengths for cluster 1 and 2.

3.2.2. London Array Offshore Wind Farm as a Floating Offshore Wind Farm

To assess the feasibility of adapting the London Array layout for a FOWF, mooring lines and anchors are incorporated into the original layout following the geometric considerations outlined in Section 3.1. As shown in Figure 3.4a, the layout is already infeasible even before evaluating the optimized cable routing. This is due to violation of the minimum distance constraint between anchors and mooring lines, where the required clearance of $d_{min} = 100$ m is not always maintained (constraint C5). The resulting optimized cable routing is presented in Figure 3.4a. When analyzing the proposed optimized cable routing, four passages fail to meet the minimum spacing requirements as well, which are highlighted in Figure 3.4a. The passages are labeled

and referred to in order from the lowest to the highest y-coordinate. Table 3.3 presents the width of these passages, as well as the number of cables allowed and the actual number routed. Figure 3.4b shows a close-up of the most severe violated passage, passage 2, where seven cables are routed through a passage able to accommodate only one.

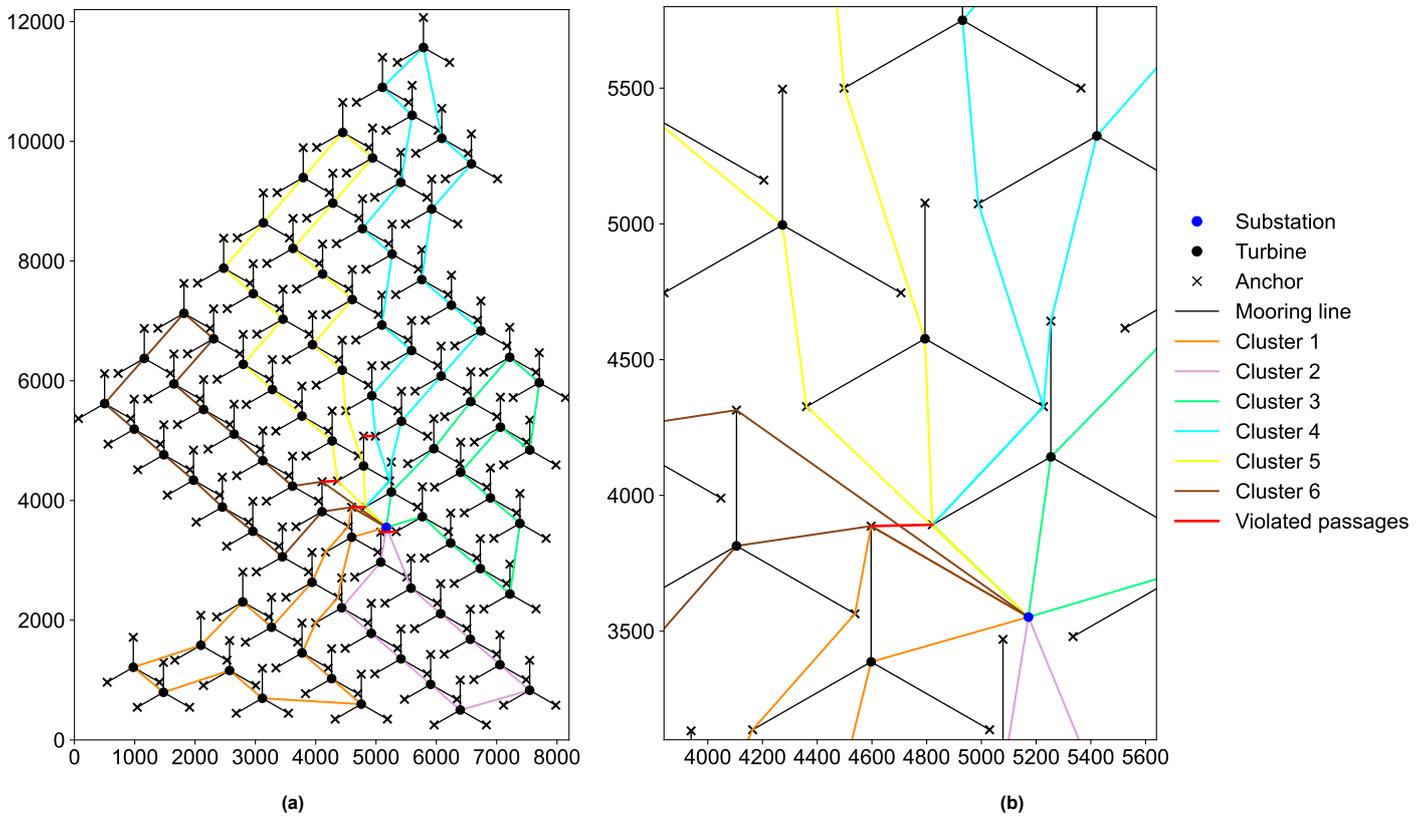


Figure 3.4: (a) Passage violations in the optimized cable routing in case the London Array wind farm would become a FOWF; (b) Close-up of most violated passage, passage 2, with its seven cables that are routed through the passage.

Passage	Width	Allowed cables	Routed cables
1	256.2 m	1	2
2	224.0 m	1	7
3	256.3 m	1	2
4	195.0 m	0	1

Table 3.3: Violated passages in the optimized cable routing of the London Array wind farm as a FOWF.

3.2.3. Modification and Refinement of London Array Offshore Wind Farm as Floating Offshore Wind Farm

To create a layout that complies with the geometric design principles outlined in Section 3.1 and Figure 3.2, the entire wind farm is uniformly scaled in both the x- and y-directions by a factor of 1.25. This scaling increases all distances by $1.25^2 = 1.5265$. As a result, the rows of turbines are now spaced approximately 1565 m apart, with a 1015 m distance between turbines. The proportions within the layout remain unchanged. The updated layout, cable routing, and passage violations are shown in Figure 3.5a. The passages that were originally violated in the unscaled layout are listed in Table 3.4, along with the violated passage after scaling, labeled as passage 5.

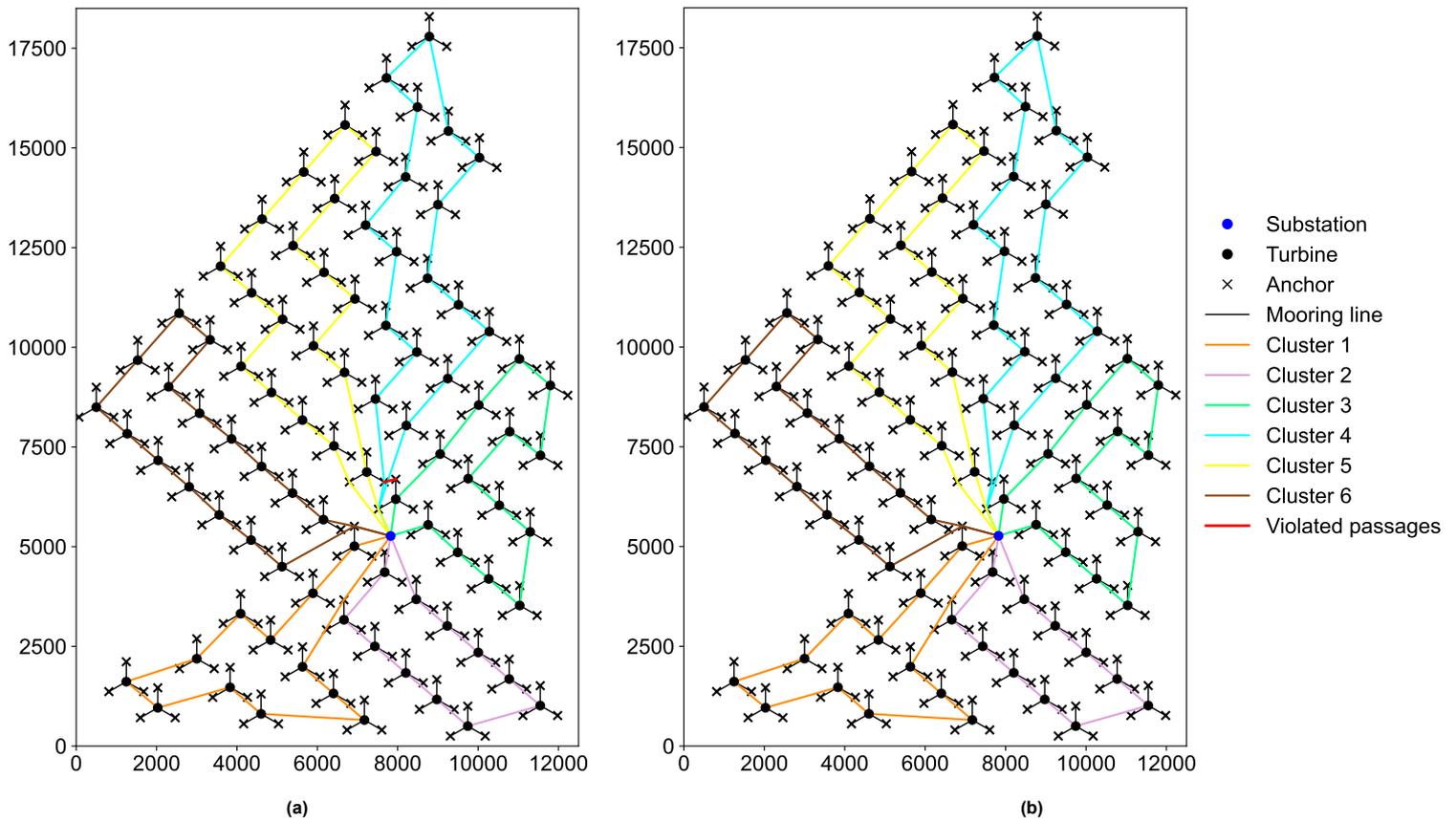


Figure 3.5: Comparison of passage violations before and after turbine position adjustment. **(a)** Optimized cable layout for the London Array FOWF scaled by a factor of 1.25, highlighting passage violations; **(b)** Passage violation resolved after shifting the turbine associated with the violated passage, ensuring sufficient clearance for the suggested cable routing.

The width of passage 5 in Table 3.4 indicates that increasing its width by approximately 3 m would make the cable routing feasible for this wind farm layout. Instead of scaling the entire wind farm again, a more efficient solution is to shift the turbine on the right side of passage 5 by 3 m along the positive x-axis, thereby increasing the passage width to the required length. Since no other cables pass through passages involving this turbine, this adjustment is not expected to introduce additional violations. After applying this manual modification to the turbine position, the optimization framework was rerun. The new optimization result maintained the same overall cable routing as before but successfully eliminated the passage violation, as indicated in Figure 3.5b, where no highlighted passages are shown. This confirms that small geometric adjustments in the wind farm layout can result in a feasible cable routing solution without fundamentally altering the optimal routing structure or requiring major changes to the overall layout.

Passage	Width	Allowed cables	Routed cables
1	780.7 m	6	2
2	735.4 m	6	6
3	783.3 m	6	1
4	691.0 m	5	1
5	297.1 m	1	2

Table 3.4: Violated passages in the non-scaled and scaled version of the London Array FOWF layout.

This analysis highlights the significant spatial constraints imposed by mooring systems when adapting a bottom-fixed wind farm layout to a floating offshore wind farm. The additional space required for mooring lines and anchor clearances makes it challenging to achieve a feasible layout that complies with even the minimal geometric constraints outlined in Section 3.1. Scaling the layout improved overall spacing and reduced passage violations, but a single critical violation remained. This issue was resolved through a minor turbine

adjustment, followed by rerunning the optimization, which confirmed that the framework provides robust routing solutions that can be further refined through small, user-driven modifications. These results demonstrate that the framework is a useful tool for initial layout optimization while allowing for practical adjustments to ensure compliance with design constraints.

3.3. Industry-Standard Violations

Section 3.2 demonstrated that while a given wind farm layout may be feasible for BFOWTs, the same layout may become infeasible when FOWTs are used. This infeasibility is due to passage violations, where the required minimum clearance between IACs and anchors cannot be maintained. This section aims to analyze the trends in the occurrence and causes of passage violations. To investigate these trends, a wind farm consisting of $N = 60$ turbines with randomly generated turbine positions is considered. The experiment is conducted both two OSS placement scenarios: one in which the OSS is located near the edge of the field and another where the OSS is centrally positioned, as described in Section 3.1. For each scenario, five different randomly generated layouts are evaluated to capture variations in passage violations. For each unique layout, the cable routing optimization is performed seven times, each with a different combination of cluster and loop sizes, as detailed in Table 3.5. The number of clusters, cluster size, loop size, and total number of loops influence the routing complexity as well as the occurrence and causes of passage violations. The root branches represent the total number of cables entering and exiting the OSS. The loops-to-turbine ratio, or loops-to-turbine density, provides a metric to assess the density of cables within each combination of cluster size and loop size.

Clusters	Cluster size	Loop size	Total loops	Root branches	Loops/turbine
3	15	15	3	6	0.0667
5	12	12	5	10	0.0833
6	10	10	6	12	0.1
6	10	5	12	24	0.2
10	6	6	10	20	0.167
10	6	3	20	40	0.333
10	6	2	30	60	0.5

Table 3.5: Cluster and loop configurations for randomly generated wind farms with $N_{farm} = 60$.

Figure 3.6a presents the total number of uniquely violated passages aggregated across the five randomly generated layouts for each scenario. The navy-colored bars represent the violated passages in wind farms where the OSS is positioned near the edge, while the light blue bars represent the violated passages where the OSS is centrally located. The results indicate that, for both OSS placement scenarios, the number of unique violated passages increases as the loop size decreases or as the total number of loops and cables increases. A single passage can be violated multiple times. For example, if a passage is wide enough to accommodate four cables but six cables are routed through it, the passage is considered violated twice. The total number of violations, accounting for multiple occurrences of the same passage being violated, is shown in Figure 3.6b.

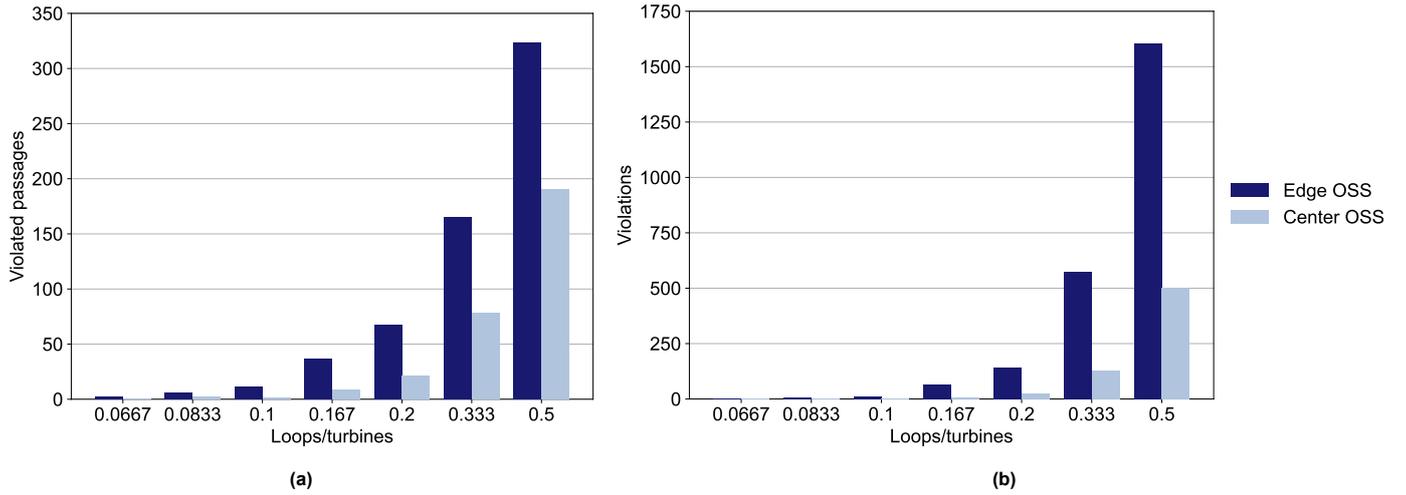


Figure 3.6: Comparison of (a) total number of unique violated passages and (b) total number of violations for different OSS placements across five random layouts.

The results presented in Figure 3.6b indicate that the total number of passage violations is more than three times higher when the OSS is positioned at the edge of the wind farm compared to when it is centrally located. Table 3.6 presents the allowed turbine area A_{farm} and corresponding packing efficiencies η_p for both OSS placement scenarios. The packing efficiency η_p can be expressed as:

$$\eta_p = \frac{N_{farm} \cdot A_{mooring}}{A_{farm}} \cdot 100\% \quad (3.6)$$

The increase in passage violations is particularly notable given that the allowed turbine area A_{farm} is approximately 25% larger for the edge OSS layout.

OSS position	A_{farm}	η_p
Edge	74.7 km ²	26.7 %
Center	60.5 km ²	32.2 %

Table 3.6: Comparison of allowed turbine area A_{farm} and packing efficiency η_p for $N_{farm} = 60$ for different OSS placement scenarios.

To understand the threefold increase in passage violations observed between the two OSS placement scenarios, it is useful to examine the spatial distribution of these violations. Figures 3.7a and 3.7b illustrate the locations of the violated passages for the highest loop-to-turbine density for the OSS positioned at the edge and the OSS positioned at the center of the wind farm, respectively. The size of each marker represent the amount or frequency of violations at a given passage, with larger circles indicating higher congestion levels. For both OSS placement scenarios, the most violated passages are concentrated in the vicinity of the OSS. It can also be noted that the passages closer to the OSS tend to be significantly more frequently being violated than those violated passage occurring farther away, as indicated by the larger marker sizes near the OSS. This behavior is expected, as the density of IACs increases as they merge (or converge) from multiple clusters into one connection point, the OSS.

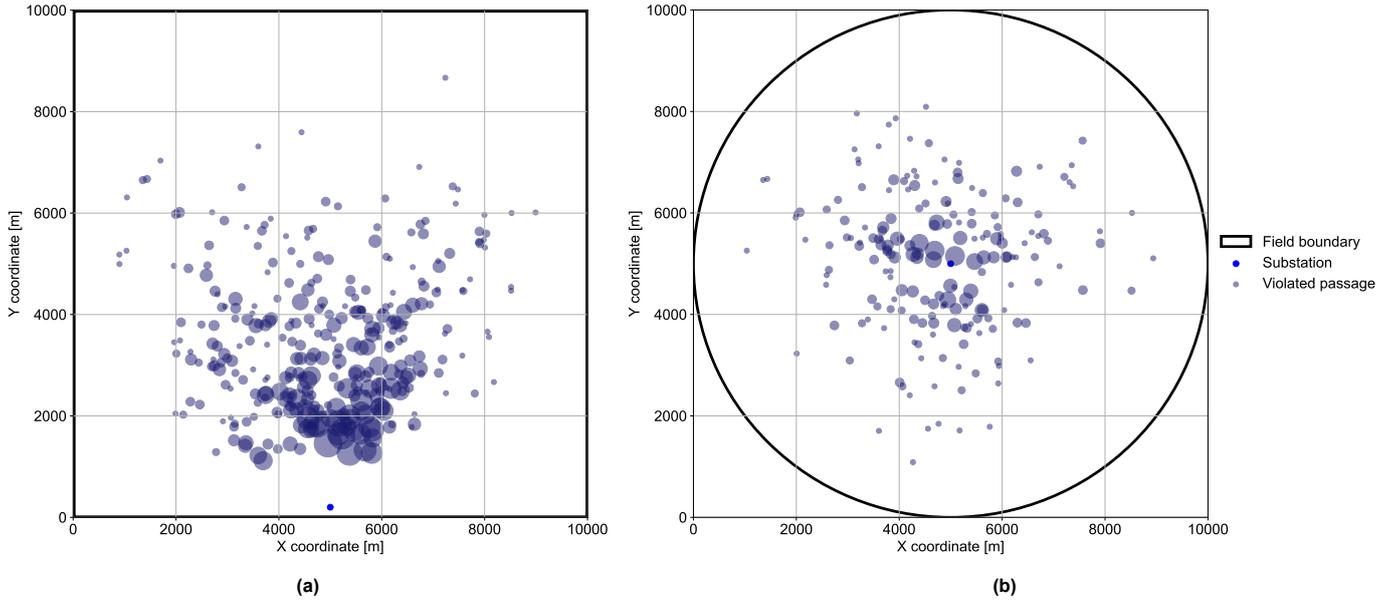


Figure 3.7: Spatial distribution of passage violations for the highest loop-to-turbine density scenario, comparing (a) OSS positioned at the edge and (b) OSS positioned at the center of the wind farm. The size of each marker represent the frequency of violations at that passage, with larger circles indicating a higher number of violations.

The increased number in passage violations in the edge OSS scenario can be attributed to the cable routing constraints imposed by the substation's location. When the OSS is located at the edge, all cables must enter and exit in a cone angle θ_{cone} , as can be seen in Figure 3.1b, that theoretically can be:

$$\theta_{cone} = \pi - 2\varphi = \pi - 2 \arctan \left(\frac{d_{OSS} + R_{anchor}}{\frac{w}{2} - R_{anchor}} \right) \quad (3.7)$$

where φ is the angle with the x-axis to the theoretically possible position of the first turbine at the left or rightmost position, d_{OSS} is a distance between the OSS and the first turbine, R_{anchor} is the anchor radius, and w is the width of the field (the width of the x-axis). In case of this experiment, where $w = 10000$ m, $d_{OSS} = 500$ m, and $R_{anchor} = 500$ m, this gives a maximum theoretical cone angle where the IACs can enter or exit the OSS of:

$$\theta_{cone} = \pi - 2 \arctan \left(\frac{500 + 500}{\frac{10000}{2} - 500} \right) = \pi - 2 \arctan \left(\frac{1000}{4500} \right) \approx 155^\circ \quad (3.8)$$

Conversely, when the OSS is centrally positioned, cables can be distributed over 360° , leading to "congestion" of passages. This observation is further reinforced by Figure 3.8, which illustrates the ratio of unique violated passages to the total number of violations for each OSS placement scenario. Notably, no bar is present at the 0.0667 x-axis entry for the centrally positioned OSS, as no violations occurred in any of the five random layouts. For both OSS placements, the first three loop/turbines entries where violations occur indicate a 1:1 ratio between unique violated passages and total violations. In other words, at lower cable densities, every violation corresponds to a distinct passage. However, as the number of loops per turbine increases, this ratio diverges, with passages experiencing multiple violations at a time. From a loop-to-turbine density of 0.167, the ratio of unique violated passages to total violations is between 1.75-2.1 times as high for the central OSS scenario compared to the edge OSS scenario. At the highest loop-to-turbine density, the 0.5 entry, this means that for every violated passage, an average of five excess cables are routed through the passage (besides the cables that are already allowed to be routed through the passage) when the OSS is at the edge, compared to three excess cables in the central placement scenario. This supports the hypothesis that a wider cone angle enables a better widespread cable distribution, reducing congestion at critical passages.

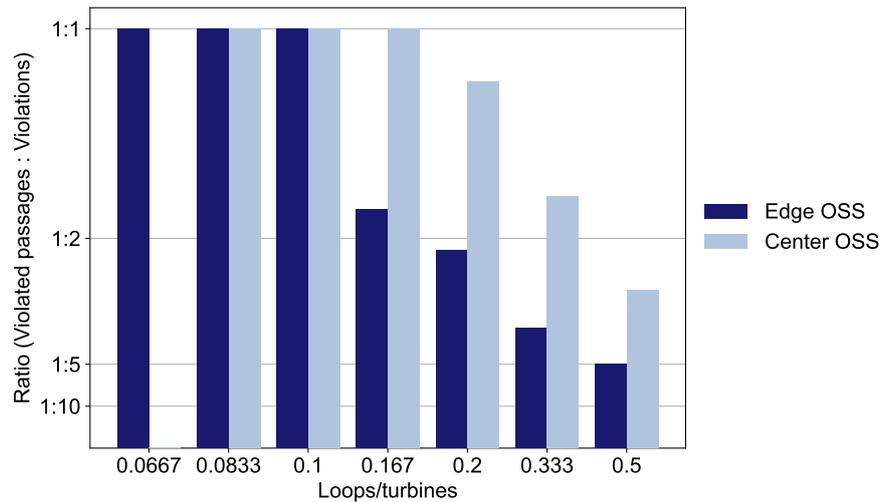


Figure 3.8: Ratio of unique violated passages to total amount of violations for seven loop-to-turbine densities for two OSS placement scenarios.

To gain insight into the occurrence and underlying causes of passage violations, the violated passages are categorized into two types: inter-cluster and intra-cluster passages. The term inter-cluster refers to violations occurring in passages between clusters, while intra-cluster refers to passage violations within a single cluster. A passage is classified as inter-cluster if it is located between anchor points of turbines belonging to different clusters, meaning the passage crosses a cluster boundary. Conversely, a passage is classified as intra-cluster if it is located between anchor points of turbines within the same cluster, meaning it does not cross a cluster boundary. Figure 3.9 presents the percentage of intra- and inter-cluster passages among all unique violated passages. As previously observed, no violations occur at the 0.0667 loop-to-turbine density for the centrally positioned OSS, and thus no bar is present for this configuration. The results indicate that at low loop-to-turbine densities, the majority of violations occur within intra-cluster passages. However, as the loop-to-turbine density increases, the proportion of inter-cluster violations increases significantly across both OSS placement scenarios.

This trend can be attributed to the fact that higher loop-to-turbine densities result in more clusters being created, thereby increasing the number of cluster boundaries. Consequently, a larger proportion of passages are classified as inter-cluster rather than intra-cluster. At low densities, most violations occur in passages that are inherently too narrow to accommodate even a single cable, as can also be seen from Figure 3.8 as the ratio between violated passages and total violations is 1:1 meaning that those passages that were violated are unable to accommodate even one cable, and since fewer cluster boundaries exist at these densities, the likelihood of an intra-cluster violation is higher. It is important to note that Figure 3.9 represents only the percentage of unique violated passages classified as inter- or intra-cluster. If we instead consider the total number of violations, thus including multiple occurrences of the same passage being violated, the intra-cluster violations would be significantly lower - so low, in fact, that a meaningful distinction between intra- and inter-cluster violations is nearly impossible to visualize in a similar figure. This is because intra-cluster passages, by definition, are confined within a single cluster. When the number of routed cables increases, the maximum by which these intra-cluster passages can be violated also increases at the same rate. However, inter-cluster passages, which are located at the boundaries between multiple clusters, tend to experience a much faster increase in violations. This is due to the fact that multiple clusters contribute cables to these passages, leading to a disproportionately higher number of violations compared to intra-cluster passages.

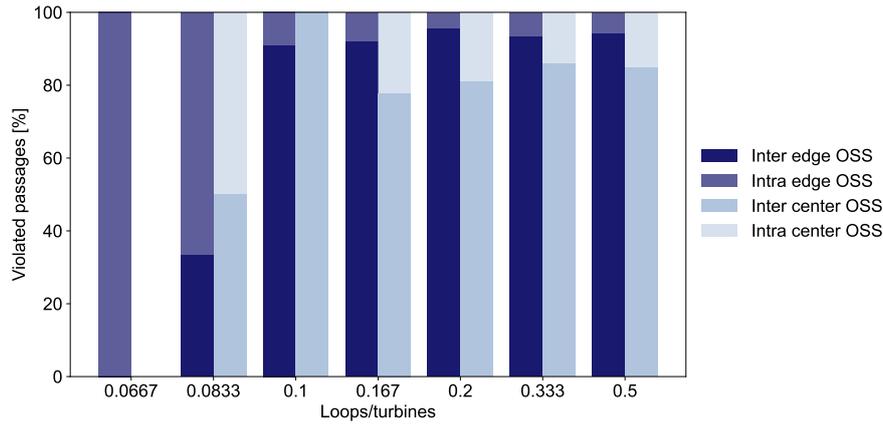


Figure 3.9: Proportion of inter- and intra-cluster violated passages across different loop-to-turbine densities for both OSS placement scenarios.

Figure 3.10 presents the aggregated spatial distribution of unique violated passages across five randomly generated wind farm layouts for three different turbine densities. Figure 3.10a corresponds to a wind farm with $N_{farm} = 40$, Figure 3.10b represents a wind farm with $N_{farm} = 60$, and Figure 3.10c shows the results for a wind farm with $N_{farm} = 80$. All wind farm layouts are clustered with a cluster size of $N_{cluster} = 10$ turbines and loop size $N_{loop} = 5$, resulting in a loop-to-turbine density of 0.2 for every turbine density. The marker size in the figure represents the severity of passage violations, with larger markers indicating more frequently violated passages. The results demonstrate that as the turbine density or packing efficiency increases, both the number of unique violated passages and the total number of violations increase. This trend is expected, as higher turbine densities lead to reduced passage widths, limiting the available space for cable routing and increasing the likelihood of passage violations.

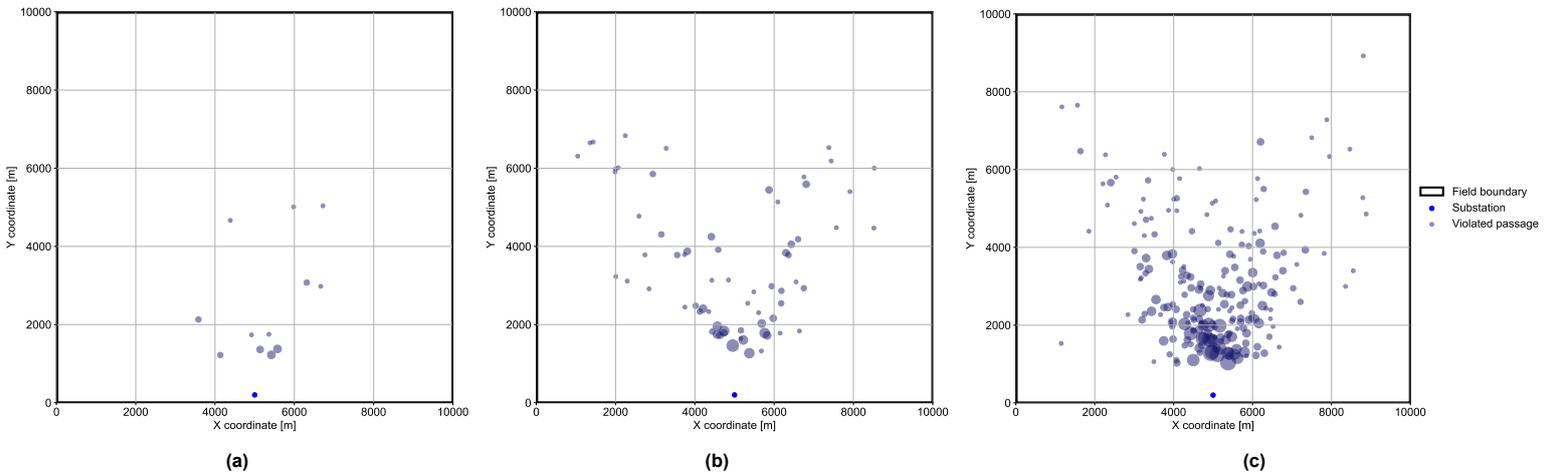


Figure 3.10: Spatial distribution of unique violated passages aggregated across five random wind farm layouts for three different turbine densities: (a) $N_{farm} = 40$, (b) $N_{farm} = 60$, and (c) $N_{farm} = 80$. Larger markers indicated passages with a higher frequency of violations.

3.4. Model Limitations

From Chapter 2 Methodology, it is known that turbines are clustered and boundaries around these clusters are established as preprocessing steps for the MILP model. However, both algorithms rely on manually defined input parameters: the cluster size and start index for the turbine clustering algorithm, and the threshold length for the boundary algorithm. Since these parameters are manually set, it is crucial to understand their impact on both the feasibility and objective of the optimized cable routing. This section examines the influence of these parameters, first focusing on those related to turbine clustering, followed by an evaluation of the boundary algorithm and its role in shaping the cable layout.

3.4.1. Turbine Clustering

The cluster size is inherently dependent on the loop size, as ensuring that the cluster size is a multiple of the loop size enables the optimization framework to prioritize cable routing rather than turbine clustering, aligning with the objective of this thesis. Smaller loops increase the cable length, leading to higher installation and material costs, as well as greater complexity in routing due to passage violations, which reduce the number of feasible solutions. Conversely, large loops are limited by cable capacity constraints and reduce system redundancy, making the cable layout less robust to turbine maintenance or cable failures. However, since decisions on cable type and capacity are outside of the scope of this thesis, and all turbines are assumed to be identical and generate a single unit of electricity, the loop size can be seen as the maximum cable capacity, and is hence the maximum number of turbines allowed in a single loop.

In addition to the cluster and loop size, the starting index of the clustering process also plays a significant role in the final cable layout. Since clusters are assigned based on a predefined sweep-order of turbines, the choice of the starting turbine affects the way clusters are formed. This subsection first examines the effects of cluster and loop size on the computational performance of the MILP model before analyzing the influence of the starting index on the optimization outcome and framework's performance.

3.4.1.1. Cluster and Loop Size

This section investigates the impact of different cluster and loop sizes on the performance of the MILP model, particularly focusing on computational time. The objective is to identify an "optimal" cluster size that balances computational efficiency with the need to keep the cluster size as large as possible, emphasizing cable routing optimization. The experiment is divided into two parts: seeing the scaling behavior in computational time when solving multiple clusters and the impact of cluster and loop size on a single cluster. The first part examines how computational time scales with the number of clusters, using a fixed cluster and loop size. This confirms whether computational time increases linearly as more clusters are solved sequentially. It is important to confirm this trend to ensure that the results from the second experiment, solving a single cluster of different cluster and loop size, can be extrapolated to larger farms. The second part investigates the impact of cluster and loop size when solving a single cluster on the computational time. By testing various cluster and loop sizes, this experiment seeks to identify the most computationally efficient cluster and loop size combination while maintaining a large cluster size to prioritize cable routing optimization. The findings from the first experiment justify using the fastest single-cluster and loop size combination as the optimal choice for larger farms, assuming a linear scaling trend.

The first part of the experiment starts with a single cluster of $N_{cluster} = 10$, arranged in two loops of $N_{loop} = 5$ each. Clusters are then added incrementally until the total farm size reaches $N_{farm} = 100$, while keeping the substation centrally positioned. To ensure that the observed computational trends are not affected by geometric complexity, each cluster is arranged in an identical layout. Since the layout is the same for each cluster, geometric differences do not impact the results. Additionally, the experiment is repeated three times to account for stochastic variations in solver performance. Since the layout of individual clusters do not impact the results, they are not explicitly shown. Figure 3.11 presents the resulting computational times for increasing number of clusters. A trend line fitted to the data suggests an approximately linear relationship between the number of clusters and computational time, with minor deviations indicating a weak quadratic trend. This behavior is expected, as solving the cable routing problem cluster by cluster results in a computational scaling pattern that is close to linear.

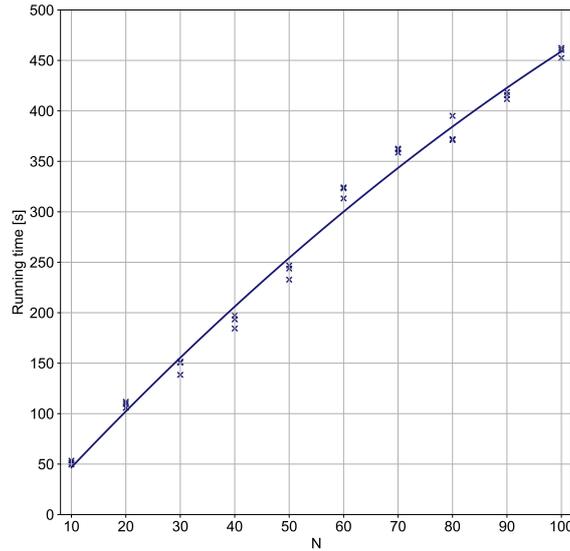


Figure 3.11: Computational time for sequentially solving clusters of $N_{cluster} = 10$ with two loops, increasing from one to ten clusters.

The second part of the experiment examines the average computation time required by the MILP model for wind farms of varying sizes, ranging from $N_{farm} = 8$ to $N_{farm} = 15$, each consisting of a single cluster. Therefore, in this experiment, $N_{farm} = N_{cluster}$. Multiple loop sizes N_{loop} are tested to evaluate their impact on computational time, with a minimum of $N_{loop} = 2$. Table 3.7 summarizes the number of loops L_N resulting from different cluster and loop size combinations. For example, in a cluster with $N_{cluster} = 10$, a loop size of $N_{loop} = 4$ results in three loops $L_N = 3$: one with $N_{loop} = 4$ and two with $N_{loop} = 3$ due to balanced routing. To maintain the optimization framework's emphasis on cable routing rather than clustering, the cluster size would ideally follow the relation $N_{cluster} = xN_{loop}$, where $x > 1$ and $x \in \mathbb{N}$.

$N_{cluster}$	N_{loop}											
	2	3	4	5	6	7	8	9	10	12	14	15
8	4	3	2	2	2	2	1	x	x	x	x	x
9	x	3	3	2	2	2	2	1	x	x	x	x
10	5	4	3	2	2	2	2	2	1	x	x	x
12	6	4	3	3	2	2	2	2	2	1	x	x
14	7	5	4	3	3	2	2	2	2	2	1	x
15	x	5	4	3	3	3	2	2	2	2	2	1

Table 3.7: Number of loops L_N inside a wind farm or single cluster of size $N_{farm} = N_{cluster} = 8, 9, 10, 12, 14, 15$ for different loop sizes N_{loop} .

Figure 3.12 shows the results of the second part of the experiment. Although Figure 3.12 presents results for a single cluster, the linear trend observed in Figure 3.11 suggests that this conclusion remains valid for larger farms with multiple clusters. While increasing cluster size, and thereby being able to increase the loop size while maintaining the relation $N_{cluster} = xN_{loop}$ where $x > 1$ and $x \in \mathbb{N}$, may bring solutions closer to a practical optimum, the results from Figure 3.12 indicate that the average running times of the MILP model increase rapidly. This is expected, as it is known from Section 2.3.3 that possible cable configurations grows factorially with $N_{cluster}$.

However, to decide on a good combination of cluster and loop size while maintaining reasonable time limits is highly context-dependent. During the development and testing phase of this optimization framework, an iterative approach was required, where multiple scenarios and configurations were explored to refine the model and ensure robustness under various constraints. In this context, a computational time of 300 seconds per cluster, as shown for $N_{cluster} = 12$, is considered long. For example, a wind farm with ten clusters of this size would require approximately 50 minutes for one iteration. Such durations hinder the iterative process, especially when constraint violations occur only in specific cases, requiring multiple adjustments and reruns.

Therefore, in the context of this thesis, smaller clusters, typically $N_{cluster} = 10$ with $N_{loop} = 5$, were chosen to maintain rapid solution iterations and expedite the development and testing phase. In reality, commercial offshore wind farms often feature strings, branches or loops with a greater number of turbines than $N_{loop} = 5$. If the framework were to be applied to an existing layout for the final cable routing layout, where the focus shifts to a less iterative use, these time constraints would be less critical. In such scenarios, larger cluster and loop sizes would be better as it will allow for more flexibility/less constrained in the solution space. However, since real-world loop sizes are generally larger, the cable routing problem naturally simplifies; fewer loops reduces the loop-to-turbine density, root branches, and total cable length, making it easier to comply with industry standards such as the minimum distance clearance between IACs itself and mooring system. Hence, if the MILP model can find a feasible solution for smaller loop sizes, as used in this thesis, it is reasonable to assume that larger real-world loop sizes would also yield valid routing solutions.

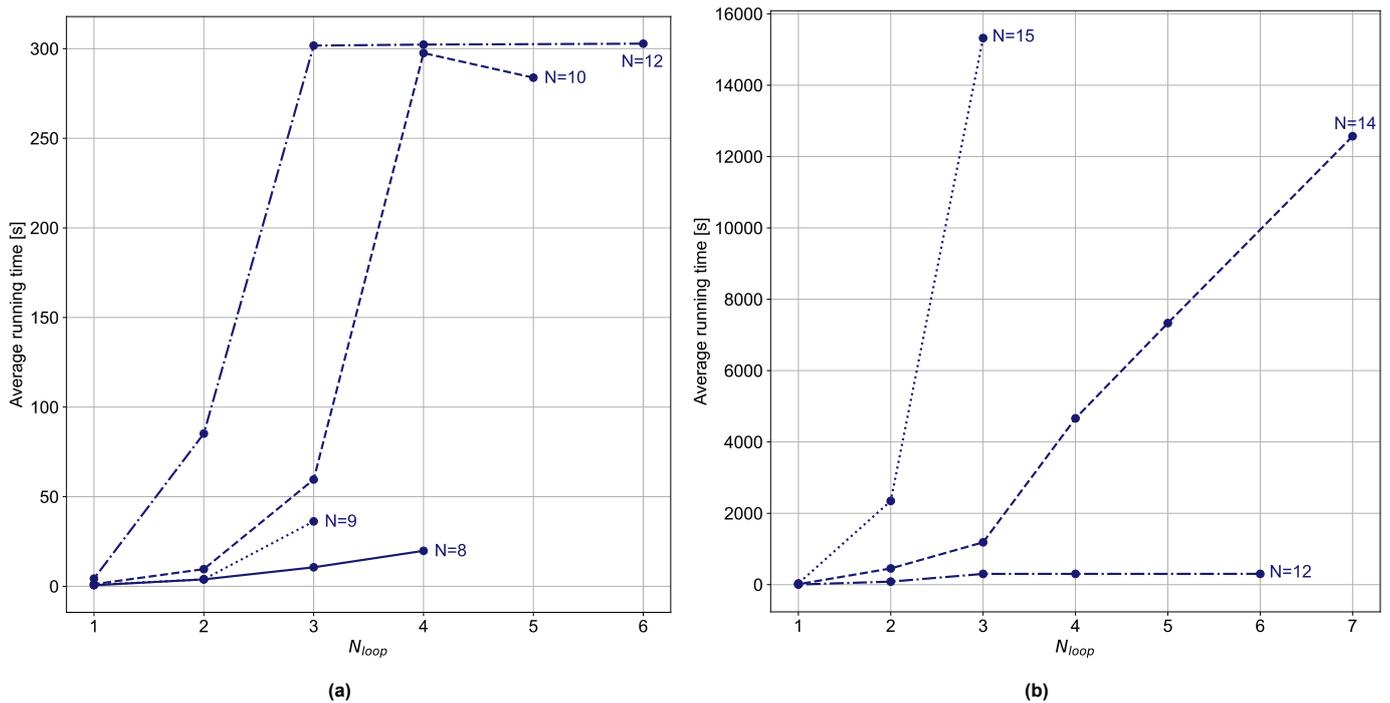


Figure 3.12: a) Average computational time for solving a single cluster with $N_{cluster}$ ranging from 8 to 12 and loop sizes N_{loop} varying from 1 to 6; b) Average computation time for solving a single cluster with $N_{cluster}$ ranging from 12 to 15 and loop sizes N_{loop} varying from 1 to 7. In both plots, $N = N_{cluster}$.

3.4.1.2. Start Index

Besides the key considerations for determining cluster and loop sizes, it is also important to highlight the impact of the start index on the model's performance and outcome. Clusters are formed sequentially based on the predefined cluster size, starting from a specific turbine known as the start index in the sweep-ordered list of turbines, as explained in Section 2.3.3.

Figures 3.13 and 3.14 illustrate how the choice of the starting index affects the cable length. In the default case shown in Figure 3.13a, the starting index leads to an inefficient configuration where WTG 1 and WTG 3 and WTG 2 and WTG 4 are grouped together, despite being geographically distant. In Figure 3.13b, the starting index is adjusted and the algorithm produces a more intuitive clustering, where turbines within the same cluster are closer to each other (Figure 3.13b).

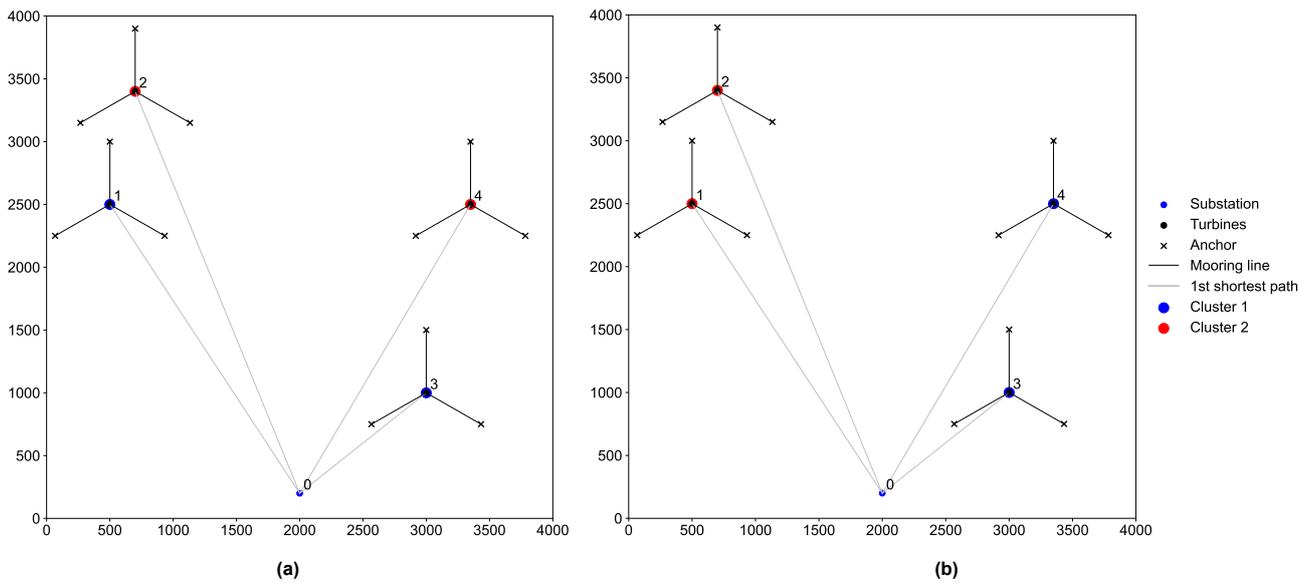


Figure 3.13: Effect of the starting index on cluster formation, where (a) the default start index leads to an intuitive less efficient clustering where turbines that are geographically distant are grouped together, and (b) adjusted starting index leads to a more intuitive clustering.

Figure 3.14 compares the resulting optimized cable routing for both cluster configurations. The suboptimal clustering in Figure 3.14a results in a total cable length of of 17157.91 m, while the cable routing in Figure 3.14b of the adjusted clustering has a total cable length of 12609.69 m, which is a decrease of 26.5 %.

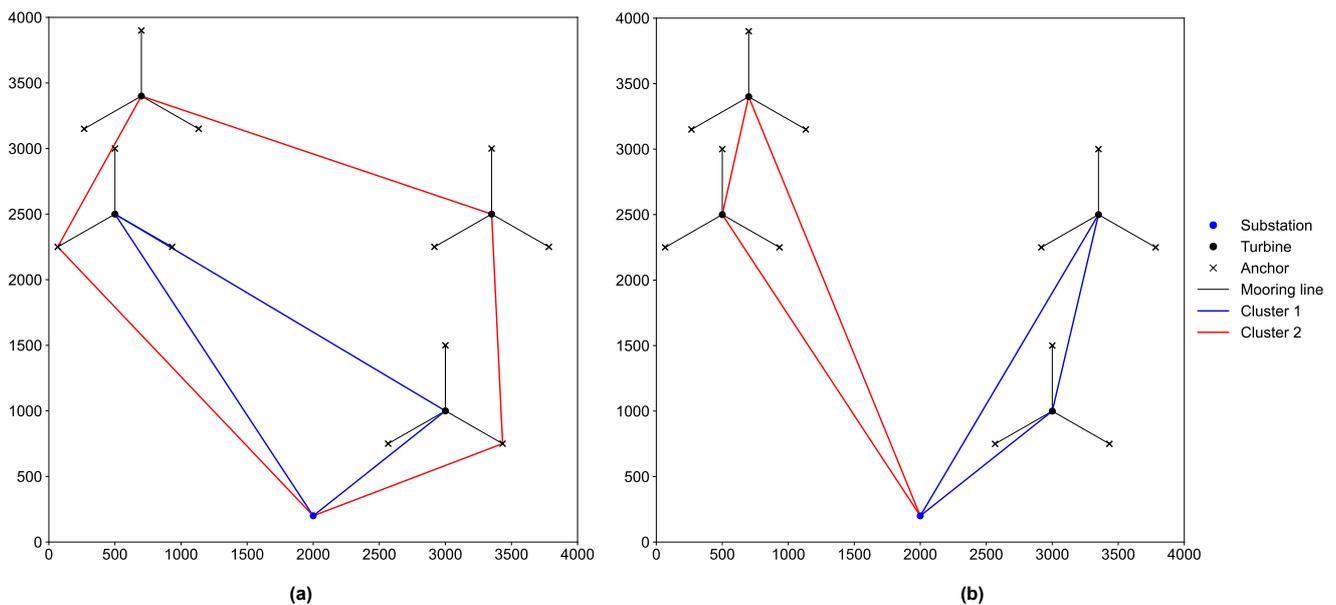


Figure 3.14: Optimized cable routing for (a) the suboptimal, default clustering, and (b) the adjusted clustering configuration.

Although in the example shown in Figures 3.13 and 3.14, the starting index only affects the model’s objective and does not impact the feasibility of the cable routing, this is not always the case. In larger wind farms, where the OSS is positioned at the edge of the field, and multiple turbines and clusters exist, the radial sweep algorithm used for clustering may lead to inefficient cluster configurations. For example, in a wind farm with an edge-positioned OSS, if a cluster is split such that some of its turbines are at the far left of the field and others are at the far right, the cluster boundary will span the entire width of the farm. This forces the cables for central clusters to be routed around the outer edges of the farm, similar to the inefficient layout seen in Figure 3.14a. This leads to longer cable lengths, unrealistic routing configurations, and potentially infeasible solutions. Even if feasible solutions are found, the computational time for the MILP model increases significantly. Therefore, for wind farms with an edge-positioned OSS, it is strongly recommended to select a starting index that results

in coherent, non-split clusters. In contrast, when the OSS is centrally located, the choice of the starting index has less impact on both the model's objective, computational time and feasibility of the solution. Due to the radial nature of the sweep algorithm and the wind farm layout, clusters will not be split in a way that places all other turbines behind the boundary of a single cluster.

3.4.2. Cluster Boundary Creation

From Section 2.3.4, the effect of the $L_{threshold}$ parameter on the concave hull construction was demonstrated. However, as this parameter is manually defined, it is crucial to assess its sensitivity and determine the conditions under which it yields valid cluster boundaries. A threshold length value is considered valid if the resulting concave hull leads to cluster boundaries that encloses all turbines within a given cluster without intersecting mooring lines, other cluster boundaries or enclosing turbines that do not belong to the designated cluster.

For this experiment, five random wind farm layouts with the OSS centrally positioned were generated for different amounts of turbines, namely $N_{farm} = 40$, $N_{farm} = 60$, and $N_{farm} = 80$, within a $10000 \text{ m} \times 10000 \text{ m}$ field, resulting in varying packing efficiencies, as summarized in Table 3.8. Each layout is clustered using two different cluster sizes with a single loop: $N_{cluster} = 20$, corresponding to a loop-to-turbine density of 0.05, and $N_{cluster} = 10$, resulting in a loop-to-turbine density of 0.1. Figure 3.15 illustrates the range of valid $L_{threshold}$ values for each case.

N_{farm}	A_{farm}	η_p
40	74.7 km ²	17.4 %
60	74.7 km ²	26.7 %
80	74.7 km ²	34.8 %

Table 3.8: Wind farm packing efficiencies η_p for different N_{farm} .

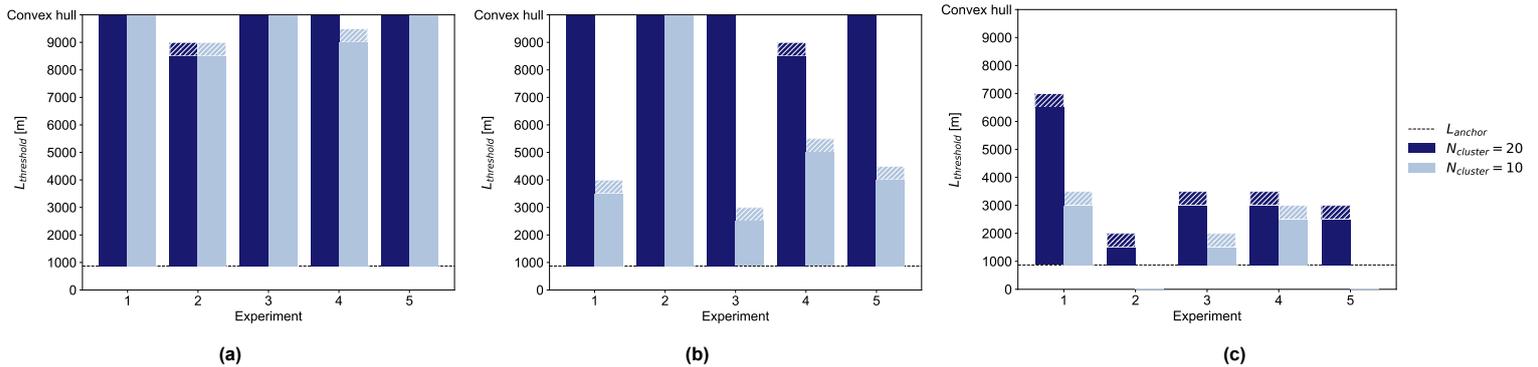


Figure 3.15: Sensitivity of the valid length threshold $L_{threshold}$ range for different turbine amounts, where (a) $N_{farm} = 40$, (b) $N_{farm} = 60$, and (c) $N_{farm} = 80$.

The hatched pattern represents uncertainty due to the discrete step size of 500 m used in threshold testing. For instance, in the second randomized wind farm layout, or Experiment 2, of Figure 3.15a, the concave hull at $L_{threshold} = 8500 \text{ m}$ creates valid cluster boundaries, while at $L_{threshold} = 9000 \text{ m}$ the boundaries become invalid. The actual transition between valid and invalid boundaries occurs somewhere in this range. Additionally, some experiments lack a bar, indicating that no tested $L_{threshold}$ value resulted in valid cluster boundaries. In cases where bars reach the top of the plot, this signifies that for all $L_{threshold} \geq L_{lower}$, the concave hull produces valid boundaries, implying that the convex hull is sufficient.

Analyzing the results across different configurations reveals that, irrespective of N_{farm} , the valid threshold range decreases as $N_{cluster}$ decreases. Furthermore, as N_{farm} , and, consequently, the packing efficiency η_p increases, the valid threshold range exhibits a decreasing trend. This is expected as the boundary creation replaces the concave hull edges with shortest paths allowed to use concave hull nodes of other clusters, and hence the more dense the wind farm, the more the boundary creation is affected by neighboring turbines.

Additionally, across all examined cases (for OSS edge and OSS center) where valid boundaries were obtained, a common lower threshold emerges at L_{anchor} . This is expected, as if $L_{lower} < L_{anchor}$ begin removing

edges between anchor points of the same turbine, leading to boundary inconsistencies. To illustrate effect, Figures 3.16a and 3.16b depict the Delaunay triangulation and the removed edges for $L_{threshold} = 870$ m and $L_{threshold} = 860$ m, respectively.

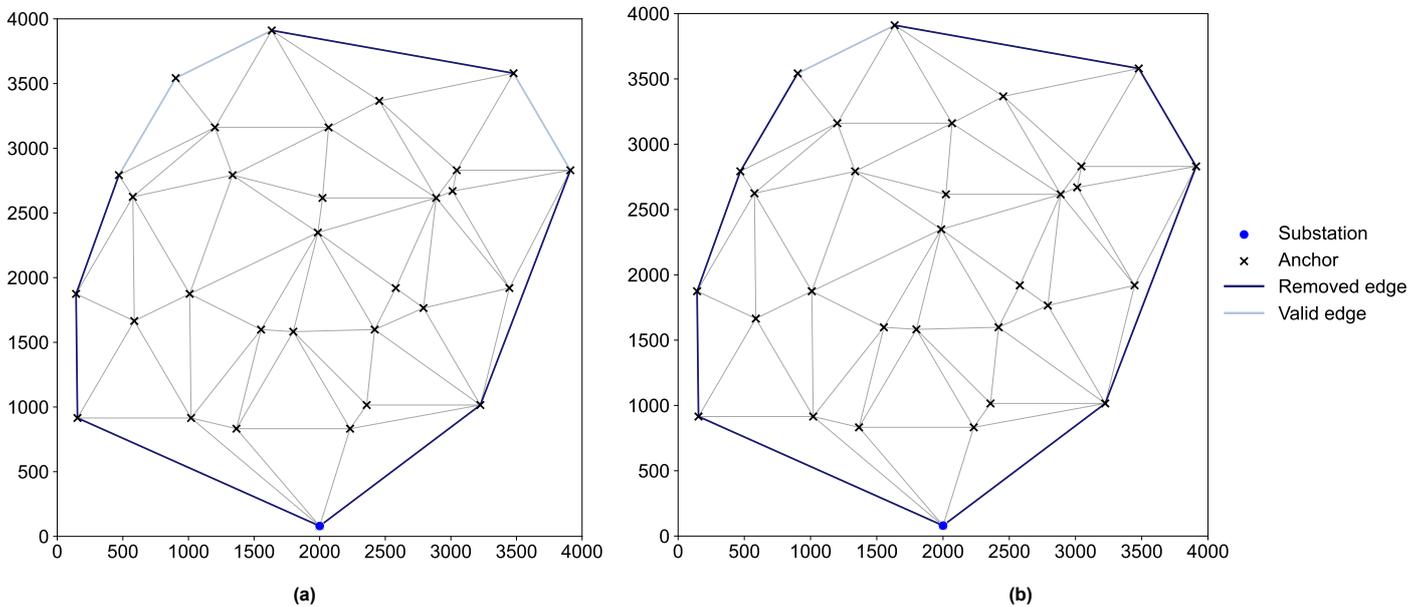


Figure 3.16: Delaunay triangulation as input for the concave hull construction and its removed edges due to $L_{threshold}$ for (a) $L_{threshold} = 870$ m, and (b) $L_{threshold} = 860$ m.

As these boundary edges are eliminated from the valid edge set, the resulting concave hull boundaries often intersect mooring lines or exclude turbines from their designated clusters. The corresponding concave hulls for both threshold values are shown in Figure 3.17.

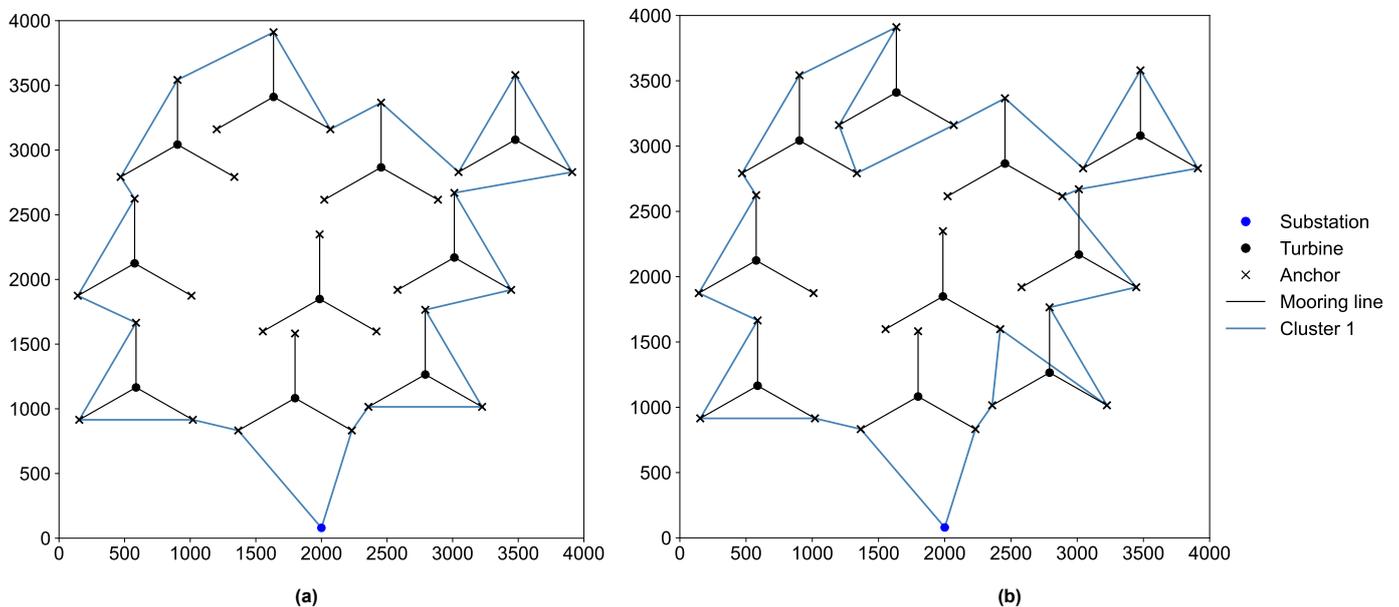


Figure 3.17: Concave hull boundaries for (a) $L_{threshold} = 870$ m and (b) $L_{threshold} = 860$ m.

In conclusion from this experiment, as the shortest paths that are created for the cluster individually are allowed to cross its own cluster boundaries, but not the cluster boundaries of other clusters, it is suggested to use a $L_{threshold} = 870$ m, as this value has the highest chance of resulting in feasible boundaries.

4

Conclusions and Recommendations

This chapter summarizes the key findings of this thesis by answering the main and sub research questions defined in Section 1.7 and provides recommendations for further research. The conclusion outlines how the WFCR problem for FOWFs was addressed, while considering loop topology and mooring system constraints in Section 4.1. In Section 4.2, the recommendations are presented, which focus on potential improvements in the optimization framework, clustering strategy, boundary formation, and compliance with industry standards, as well as further enhancements in computational efficiency and subtour elimination constraints.

4.1. Conclusions

This thesis aims to provide insights into how the WFCR for FOWFs can be optimized, while minimizing the total IAC length when considering a loop topology. The research focuses on understanding the integration of mooring systems into the wind farm layout and how the additional constraints and routing adjustments imposed by the mooring system and loop topology impact the optimization process. While the interaction between turbine placement and cable routing is a relevant topic, this thesis specifically focuses on developing feasible solutions for fixed FOWF layouts. The optimization framework employs an exact approach using MILP. This approach is chosen due to its ability to systematically address the complexities introduced by the mooring system and loop topology. By structuring the framework into three key phases, preprocessing, optimization, and postprocessing, it becomes possible to handle constraints separately, allowing for a more efficient solution process.

Given that the number of possible configurations for the cable routing increases factorially, the problem can be decomposed into smaller subproblems by clustering the turbines. Experimental results indicate a near-linear increase in computational time when solving multiple clusters sequentially, confirming that breaking down the problem into smaller subproblems can ease computational complexity. However, due to cable crossings between turbines that belong to different clusters, these subproblems can not be completely decoupled. Thus, although turbine clustering can divide the optimization problem into smaller subproblems, it does not result in independent subproblems; this needs an additional preprocessing step of forming cluster boundaries.

Besides the need for cluster boundaries, clustering the turbines to divide the WFCR into smaller subproblems, also influences the optimization process due to two manually defined parameters: the cluster size and the turbine with which the clustering is started, hereafter referred to as starting index. The cluster size plays a crucial role in balancing computational complexity with routing flexibility. A larger cluster size allows for larger loops, which can lead to a more efficient routing solution by reducing the number of cables and minimizing the total cable length. However, larger clusters also result in longer computational times, as each cluster contains more turbines, making the MILP model slower to solve. In addition to cluster size, the starting index of the clustering process significantly influences both the solution quality and the feasibility of the routing. A poorly chosen starting index can lead to geographically inefficient clusters, increasing the total cable length and even causing infeasibility when clusters split in a way that forces cables into unnatural detours. The impact of the starting index is particularly critical in wind farms where the OSS is positioned at the edge of the field. In this scenario, only one specific starting index ensures compact and well-distributed clusters. Conversely, in wind farms where the OSS is centrally located, every possible starting index leads to feasible clusters. This is because the clustering algorithm used in this thesis is based on a radial approach, which naturally aligns with the way turbines are distributed around a centrally placed OSS. As a result, centrally positioned OSS layouts

provide greater flexibility in clustering, reducing the likelihood of inefficient or infeasible cluster formations.

Since the need for cluster boundaries arises as a direct consequence of grouping turbines, the clustering process also influences the optimization process through this preprocessing step, particularly due to the way boundaries are defined. A more restrictive boundary formation results in tighter and more contained cluster areas, meaning that the shape of each cluster is less influenced by nearby turbines. In contrast, a more flexible boundary definition allows for larger, more relaxed cluster areas. However, increasing the flexibility of boundaries also raises the risk of clusters expanding too far, potentially overlapping with the space occupied by other clusters and their mooring systems. This can lead to infeasible configurations where turbines from one cluster become mistakenly included within another. Such cases make it impossible to establish clear cable connections without crossing into areas that belong to a different cluster. Despite this risk, allowing more flexibility in boundary formation, as long as it does not result in overlapping clusters, is generally preferred. Restrictive boundaries limit the available space for routing cables, reducing the number of possible solutions. However, in this thesis, the approach to defining cluster areas allows cable routes within a cluster to extend beyond its immediate boundary, while still preventing routes from crossing into other clusters. As a result, the exact boundary placement does not have as significant an impact on the total cable length as it would if cable paths were fully confined within each cluster's assigned area. For this reason, the recommended approach is to define boundaries as tightly as possible to minimize the likelihood of clusters overlapping. However, it is important to recognize that tighter boundaries create additional space between clusters that does not belong to any single group. In theory, this unassigned space could allow cable routes from different clusters to intersect, which might suggest that more relaxed boundaries should be used in combination with manual adjustments. This refinement would ensure that cluster areas align smoothly, while avoiding unnecessary gaps between them. Nonetheless, throughout this thesis, no instances have been observed where restrictive boundaries led to infeasibility due to unexpected intersections between cable routes from different clusters.

After the MILP model has identified a feasible solution, the proposed cable routing is evaluated to ensure compliance with industry standards regarding the minimum required spacing between cables, as well as between cables and the mooring system. Enforcing this constraint directly within the optimization model would significantly increase its complexity, requiring a more advanced mathematical formulation that is considerably more difficult to solve. To avoid this computational challenge, the clearance constraint is instead checked in a postprocessing step. However, this means that the optimization process itself does not inherently guarantee compliance with spacing requirements, and some routing configurations may still violate these constraints after the optimization phase. An analysis of these spacing violations revealed that higher densities of cable connections relative to the number of turbines result in more instances where the required clearances are not met. This increase was observed both in terms of the number of locations where violations occurred and the overall frequency of these violations. Additionally, most issues were found near the substation and at the boundaries between groups of turbines, where space for routing is more restricted. However, it was also observed that fewer violations occur when the substation is positioned at the center of the wind farm compared to when it is placed at the edge, despite the overall layout being more densely packed in the central placement scenario. These findings provide useful guidance for wind farm design, suggesting that reducing turbine density in the area surrounding the substation and choosing a central placement can help minimize clearance violations and improve the feasibility of the cable routing solution.

In cases where the optimized cable routing does not comply with the industry standard due to clearance violations, the solution still provides valuable insights into how to address these issues. Depending on the severity and nature of the violation, several corrective measures can be considered. If violations occur frequently throughout the farm, one possible modification is to scale the entire layout, increasing the distances between turbines to provide additional routing space. Alternatively, adjustments can be made to specific turbine positions to alleviate localized clearance issues. If the violations are primarily found at cluster boundaries, reclustered the turbines may be necessary to ensure better spatial distribution and reduce congestion at these critical points. The insights gained from analyzing these violations also provide guidance for future wind farm design, particularly in optimizing turbine placement and clustering strategies to minimize clearance conflicts.

In conclusion, this thesis presents an optimization framework for WFCR in FOWFs that considers loop topology and the impact of mooring systems, while minimizing the total IAC length. By employing a structured approach through preprocessing, optimization, and postprocessing, the model effectively balances computational efficiency with the feasibility of real-world implementation. The clustering of turbines into smaller subproblems helps manage computational complexity, though additional preprocessing steps, such as the formation of

cluster boundaries, are required to maintain solution feasibility. The study highlights the critical role of parameters such as cluster size, starting index, and threshold length in influencing both the optimization process and the final cable routing layout. Furthermore, the analysis of postprocessing results provides key insights into clearance violations and strategies for mitigating infeasibility, emphasizing the importance of considering spatial constraints in wind farm design. The findings of this thesis contribute to the broader field of offshore wind farm optimization by demonstrating how an MILP-based approach can be tailored to accommodate the unique constraints of FOWFs, offering a structured methodology for optimizing cable routing in these complex environments.

4.2. Recommendations for Future Research

This research has demonstrated the effectiveness of a MILP-based optimization framework for WFCR in FOWFs, but several areas remain for further improvement and refinement. Future research can build on the findings of this thesis by enhancing the clustering approach, refining the boundary generation process, improving passage width definitions, optimizing mooring system orientations, and exploring alternative mathematical optimization methods to improve computational efficiency.

One key limitation identified in this study is the dependence of clustering feasibility on the starting index, particularly in cases where the OSS is positioned at the edge of the wind farm. In this scenario, only one specific starting index ensures a feasible clustering, which may lead to inefficient solutions or even infeasibilities if not properly selected. While infeasible clustering due to the wrong starting index in this OSS placement scenario can be easily detected visually, the correct starting index can be determined by counting the remaining turbines of the split cluster that are positioned on the right side of the farm, as the sweep algorithm moves clockwise, and then increasing the starting index by this number of remaining turbines. However, a more systematic approach should be developed to automate this process and ensure optimal clustering. A possible improvement would be to implement a heuristic that automatically identifies the optimal starting index, ensuring that clusters remain compact and well-distributed without manual intervention. Alternatively, an entirely different clustering method that does not rely on a radial sweep approach should be investigated for wind farms with edge-positioned substations. In contrast, for wind farms where the OSS is centrally located, the clustering algorithm consistently produces feasible clusters regardless of the starting index. However, in large-scale wind farms, the best starting index in terms of minimizing total cable length cannot always be visually determined. Future research could explore measures to optimize clustering configurations, ensuring that the most efficient turbine groupings are identified automatically.

The formation of valid cluster boundaries is a critical preprocessing step that directly influences the feasibility of the optimization model. The current concave hull-based approach is manually tuned using a threshold length parameter, which affects the tightness of the boundaries. A smaller threshold length results in tighter boundaries that are less influenced by surrounding turbines, whereas a larger threshold allows for looser, more flexible boundaries but risks incorporating turbines from neighboring clusters. If a turbine from another cluster is unintentionally included within the boundary, the optimization model becomes infeasible, as it cannot establish valid shortest paths without violating cluster constraints. To improve boundary generation, a metaheuristic approach could be introduced to dynamically adjust boundaries based on turbine distribution, minimizing the risk of infeasibility, while maximizing available routing space. Such an approach could take into account the positions of neighboring turbines and adaptively adjust cluster boundaries to balance feasibility and routing flexibility.

Another important consideration is the definition of passage width for cable routing. In the current model, passage feasibility is determined based on the minimum clearance distance between anchor nodes of different turbines. However, the actual distance between an anchor node and the mooring line of another turbine's mooring system is not explicitly checked. In reality, this distance is often smaller than the anchor-to-anchor distance, meaning that clearance violations could still occur even if the model suggests compliance. Future work should refine the passage width definitions to account for these additional spatial constraints. Furthermore, as loop-to-turbine density increases, most violations identified in this study were classified as inter-cluster violations. To mitigate these issues, cable corridors could be introduced between clusters, ensuring additional spacing for routing flexibility and compliance with industry standards.

The orientation of the mooring system also plays a significant role in determining the available space for cable routing. In this thesis, all mooring systems were assumed to have identical configurations, with the upwind

line consistently pointing north. However, slight variations in the orientation of mooring lines and anchor placements could significantly impact the available passage widths. Future research could investigate the potential benefits of optimizing mooring orientations, allowing for alternating configurations within a single wind farm to maximize the spacing between cables and mooring lines. By strategically adjusting the placement of anchor points and mooring angles, it may be possible to create additional routing space without fundamentally altering the turbine layout.

A further area of investigation should focus on improving the computational efficiency of the optimization model. One of the primary factors influencing runtime is the inclusion of subtour elimination constraints, which prevent infeasible loops in the cable routing solution. These constraints introduce additional variables and forbidden turbine sequences, causing the model's computational complexity to grow factorially. Since subtour elimination has a major impact on runtime, alternative methods should be explored to reduce its computational burden. Possible solutions include constraint relaxation techniques or the development of more efficient constraint formulations that maintain feasibility, while reducing the number of required constraints.

In addition to refining individual model components, a broader recommendation for future research is to explore the use of matheuristic approaches, which combine exact optimization techniques with metaheuristic strategies. While MILP provides an exact solution methodology, it can be computationally expensive for large-scale wind farms. A matheuristic approach could leverage the strengths of both exact and heuristic methods, allowing for faster convergence while maintaining solution quality. For instance, a two-stage optimization process could be implemented, where a heuristic or evolutionary algorithm generates an initial feasible solution, which is then refined using MILP. By incorporating matheuristics, the optimization framework could be extended to handle larger wind farms with increased computational efficiency, while still maintaining high-quality cable routing solutions.

In conclusion, this thesis provides a structured framework for optimizing WFCR in FOWFs, but several areas for improvement remain. Enhancing the clustering methodology, refining boundary generation, improving passage width definitions, optimizing mooring system configurations, and addressing computational bottlenecks are all promising directions for future research. The implementation of heuristics, metaheuristics, and matheuristic approaches could significantly improve solution feasibility and computational efficiency, making the framework more applicable to large-scale offshore wind farms. By addressing these challenges, future research can further refine and extend the optimization framework, contributing to more efficient and feasible cable routing solutions for floating wind farms.

References

- [1] Alagab, S. M., Tennakoon, S., and Gould, C. "Review of wind farm power collection schemes". In: *Proceedings of the Universities Power Engineering Conference*. Vol. 2015-November. 2015. DOI: 10.1109/UPEC.2015.7339922.
- [2] Amaral, L. and Castro, R. "Offshore wind farm layout optimization regarding wake effects and electrical losses". In: *Engineering Applications of Artificial Intelligence* 60 (2017). ISSN: 09521976. DOI: 10.1016/j.engappai.2017.01.010.
- [3] Bauer, J. and Lysgaard, J. "The offshore wind farm array cable layout problem: A planar open vehicle routing problem". In: *Journal of the Operational Research Society* 66.3 (2015). ISSN: 14769360. DOI: 10.1057/jors.2013.188.
- [4] Bilgili, M., Yasar, A., and Simsek, E. *Offshore wind power development in Europe and its comparison with onshore counterpart*. 2011. DOI: 10.1016/j.rser.2010.11.006.
- [5] Campanile, A., Piscopo, V., and Scamardella, A. "Mooring design and selection for floating offshore wind turbines on intermediate and deep water depths". In: *Ocean Engineering* 148 (2018). ISSN: 00298018. DOI: 10.1016/j.oceaneng.2017.11.043.
- [6] Cazzaro, D., Fischetti, M., and Fischetti, M. "Heuristic algorithms for the Wind Farm Cable Routing problem". In: *Applied Energy* 278 (2020). ISSN: 03062619. DOI: 10.1016/j.apenergy.2020.115617.
- [7] Cazzaro, D. and Pisinger, D. "Balanced cable routing for offshore wind farms with obstacles". In: *Networks* 80.4 (2022). ISSN: 10970037. DOI: 10.1002/net.22100.
- [8] Cerveira, A., Pires, E. J., and Baptista, J. "Wind farm cable connection layout optimization with several substations". In: *Energies* 14.12 (2021). ISSN: 19961073. DOI: 10.3390/en14123615.
- [9] Chemineau, M., Castillo, F., Mechinaud, L., Arramounet, V., and Gilloteaux, J. C. "Design and costs benefits of shared anchors and shared mooring lines of floating wind turbines at farm level". In: *Journal of Physics: Conference Series*. Vol. 2626. 1. 2023. DOI: 10.1088/1742-6596/2626/1/012048.
- [10] Dahmani, O., Bourguet, S., Machmoum, M., Guerin, P., Rhein, P., and Josse, L. "Optimization and Reliability Evaluation of an Offshore Wind Farm Architecture". In: *IEEE Transactions on Sustainable Energy* 8.2 (2017). ISSN: 19493029. DOI: 10.1109/TSTE.2016.2609283.
- [11] DNV GL. *Energy Transition Outlook*. Tech. rep. DNV GL, 2020.
- [12] Ferguson, A., De Villiers, P., Fitzgerald, B., and Matthiesen, J. "Benefits in moving the inter-array voltage from 33 kV to 66 kV AC for large offshore wind farms". In: *European Wind Energy Conference and Exhibition 2012, EWEC 2012*. Vol. 2. 2012.
- [13] Fernández, A. and Spencer, T. *Net Zero RoadMap: A Global Pathway to Keep the 1.5 °C Goal in Reach*. Tech. rep. International Energy Agency, 2023.
- [14] Fernández, A. and Spencer, T. *Net Zero RoadMap: A Global Pathway to Keep the 1.5 °C Goal in Reach*. Tech. rep. International Energy Agency, 2023.
- [15] Fischetti, M. and Pisinger, D. "Optimizing wind farm cable routing considering power losses". In: *European Journal of Operational Research* 270.3 (2018). ISSN: 03772217. DOI: 10.1016/j.ejor.2017.07.061.
- [16] Fontana, C. M., Arwade, S. R., DeGroot, D. J., Myers, A. T., Landon, M., and Aubeny, C. "Efficient multiline anchor systems for floating offshore wind turbines". In: *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*. Vol. 6. 2016. DOI: 10.1115/OMAE2016-54476.
- [17] Fontana, C. M., Hallowell, S. T., Arwade, S. R., DeGroot, D. J., Landon, M. E., Aubeny, C. P., Diaz, B., Myers, A. T., and Ozmutlu, S. "Multiline anchor force dynamics in floating offshore wind turbines". In: *Wind Energy* 21.11 (2018). ISSN: 10991824. DOI: 10.1002/we.2222.
- [18] Gong, X., Kuenzel, S., and Pal, B. C. "Optimal Wind Farm Cabling". In: *IEEE Transactions on Sustainable Energy* 9.3 (2018). ISSN: 19493029. DOI: 10.1109/TSTE.2017.2771147.

- [19] Ho, W. C. "GA based algorithms for offshore wind farm collector cable optimization". In: *Journal of Physics: Conference Series*. Vol. 2362. 1. 2022. DOI: 10.1088/1742-6596/2362/1/012017.
- [20] International Energy Agency (IEA). *Offshore Wind Geospatial Analysis*. 2019.
- [21] International Energy Agency (IEA). *Investment for floating offshore wind*. 2022.
- [22] Janus, S. "Layout Optimisation of Floating Offshore Wind Farms: Minimisation of inter-array cable length while allowing for anchor-sharing in a floating offshore wind farm". PhD thesis. Delft Technical University, 2024.
- [23] Javid, M. A. "Understanding Dijkstra Algorithm". In: *SSRN Electronic Journal* (Oct. 2013). DOI: 10.2139/ssrn.2340905.
- [24] Kallinger, M. D., Rapha, J. I., Trubat Casal, P., and Domínguez-García, J. L. *Offshore Electrical Grid Layout Optimization for Floating Wind—A Review*. Sept. 2023. DOI: 10.3390/cleantechno15030039.
- [25] Klein, A. and Haugland, D. "Obstacle-aware optimization of offshore wind farm cable layouts". In: *Annals of Operations Research* 272.1-2 (2019). ISSN: 15729338. DOI: 10.1007/s10479-017-2581-5.
- [26] Lakshmanan, P., Sun, R., and Liang, J. *Electrical collection systems for offshore wind farms: A review*. 2021. DOI: 10.17775/CSEEJPES.2020.05050.
- [27] Lerch, M. *Technical-economic analysis, modeling and optimization of floating off shore wind farms*. Tech. rep. 2020. URL: <http://www.tdx.cat/?locale->.
- [28] Lerch, M., De-Prada-Gil, M., and Molins, C. "Collection Grid Optimization of a Floating Offshore Wind Farm Using Particle Swarm Theory". In: *Journal of Physics: Conference Series*. Vol. 1356. 1. 2019. DOI: 10.1088/1742-6596/1356/1/012012.
- [29] Lerch, M., De-Prada-Gil, M., and Molins, C. "A metaheuristic optimization model for the inter-array layout planning of floating offshore wind farms". In: *International Journal of Electrical Power and Energy Systems* 131 (2021). ISSN: 01420615. DOI: 10.1016/j.ijepes.2021.107128.
- [30] Lin, Z., Liu, X., and Lotfian, S. "Impacts of water depth increase on offshore floating wind turbine dynamics". In: *Ocean Engineering* 224 (2021). ISSN: 00298018. DOI: 10.1016/j.oceaneng.2021.108697.
- [31] Maienza, C., Avossa, A. M., Ricciardelli, F., Coiro, D., Troise, G., and Georgakis, C. T. "A life cycle cost model for floating offshore wind farms". In: *Applied Energy* 266 (2020). ISSN: 03062619. DOI: 10.1016/j.apenergy.2020.114716.
- [32] Mount, D. *CMSC 754: Lecture 1 | Introduction to Computational Geometry*. Tech. rep. Maryland, 2020.
- [33] Musial, W., Beiter, P., Spitsen, P., Nunemaker, J., Gevorgian, V., Cooperman, A., Hammond, R., and Shields, M. *2019 Offshore Wind Technology Data Update*. Tech. rep. 2019.
- [34] Pérez-Rúa, J. A. and Cutululis, N. A. "Electrical Cable Optimization in Offshore Wind Farms - A Review". In: *IEEE Access* 7 (2019). ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2925873.
- [35] Pérez-Rúa, J. A. and Cutululis, N. A. "A framework for simultaneous design of wind turbines and cable layout in offshore wind". In: *Wind Energy Science* 7.2 (2022). ISSN: 23667451. DOI: 10.5194/wes-7-925-2022.
- [36] Pérez-Rúa, J. A., Lumbreras, S., Ramos, A., and Cutululis, N. A. "Closed-Loop Two-Stage Stochastic Optimization of Offshore Wind Farm Collection System". In: *Journal of Physics: Conference Series*. Vol. 1618. 4. 2020. DOI: 10.1088/1742-6596/1618/4/042031.
- [37] Pérez-Rúa, J. A., Stolpe, M., Das, K., and Cutululis, N. A. "Global Optimization of Offshore Wind Farm Collection Systems". In: *IEEE Transactions on Power Systems* 35.3 (2020). ISSN: 15580679. DOI: 10.1109/TPWRS.2019.2957312.
- [38] Pérez-Rúa, J. A., Lumbreras, S., Ramos, A., and Cutululis, N. A. "Reliability-based topology optimization for offshore wind farm collection system". In: *Wind Energy* 25.1 (2022). ISSN: 10991824. DOI: 10.1002/we.2660.
- [39] Pérez-Rúa, J. A., Lund, R. S., Verelst, D. R., Abrahamsen, A. B., and Dykes, K. "Exact optimization of inter-array dynamic cable networks for Floating Offshore Wind Farms". In: *Renewable Energy* 237 (Dec. 2024). ISSN: 18790682. DOI: 10.1016/j.renene.2024.121647.

- [40] Pillai, A. C., Chick, J., Johanning, L., Khorasanchi, M., and De Laleu, V. "Offshore wind farm electrical cable layout optimization". In: *Engineering Optimization* 47.12 (2015). ISSN: 10290273. DOI: 10.1080/0305215X.2014.992892.
- [41] Pillai, A. C., Chick, J., Johanning, L., and Khorasanchi, M. "Offshore wind farm layout optimization using particle swarm optimization". In: *Journal of Ocean Engineering and Marine Energy* 4.1 (2018). ISSN: 21986452. DOI: 10.1007/s40722-018-0108-z.
- [42] Power Technology. *Hywind Pilot Park, Aberdeenshire*. 2018.
- [43] Qi, Y., Hou, P., Yang, L., and Yang, G. "Simultaneous optimisation of cable connection schemes and capacity for offshore wind farms via a modified bat algorithm". In: *Applied Sciences (Switzerland)* 9.2 (2019). ISSN: 20763417. DOI: 10.3390/app9020265.
- [44] Rentschler, M. U., Adam, F., and Chainho, P. "Design optimization of dynamic inter-array cable systems for floating offshore wind turbines". In: *Renewable and Sustainable Energy Reviews* 111 (2019). ISSN: 18790690. DOI: 10.1016/j.rser.2019.05.024.
- [45] Shen, X., Li, S., and Li, H. "Large-scale Offshore Wind Farm Electrical Collector System Planning: A Mixed-Integer Linear Programming Approach". In: *5th IEEE Conference on Energy Internet and Energy System Integration: Energy Internet for Carbon Neutrality, EI2 2021*. 2021. DOI: 10.1109/EI252483.2021.9713603.
- [46] Shin, J. S. and Kim, J. O. "Optimal Design for Offshore Wind Farm considering Inner Grid Layout and Offshore Substation Location". In: *IEEE Transactions on Power Systems* 32.3 (2017). ISSN: 08858950. DOI: 10.1109/TPWRS.2016.2593501.
- [47] Sijm, J., Beurskens, L., Marsidi, M., Niessink, R., Scheepers, M., Smekens, K., Van Der Welle, A., and De Wilde, H. *Review of energy transition scenario studies of the Netherlands up to 2050*. Tech. rep. 2020. URL: www.tno.nl.
- [48] Song, D., Yan, J., Gao, Y., Wang, L., Du, X., Xu, Z., Zhang, Z., Yang, J., Dong, M., and Chen, Y. "Optimization of floating wind farm power collection system using a novel two-layer hybrid method". In: *Applied Energy* 348 (2023). ISSN: 03062619. DOI: 10.1016/j.apenergy.2023.121546.
- [49] Sun, R., Abeynayake, G., Liang, J., and Wang, K. "Reliability and economic evaluation of offshore wind power dc collection systems". In: *Energies* 14.10 (2021). ISSN: 19961073. DOI: 10.3390/en14102922.
- [50] Taninoki, R., Abe, K., Sukegawa, T., Azuma, D., and Nishikawa, M. "Dynamic cable system for floating offshore wind power generation". In: *SEI Technical Review* 84 (2017). ISSN: 13434349.
- [51] *The Economics of Wind Energy*. Tech. rep. European Wind Energy Association, 2009.
- [52] Ulku, I. and Alabas-Uslu, C. "Optimization of cable layout designs for large offshore wind farms". In: *International Journal of Energy Research* 44.8 (2020). ISSN: 1099114X. DOI: 10.1002/er.5336.
- [53] Wei, S., Zhang, L., Xu, Y., Fu, Y., and Li, F. "Hierarchical Optimization for the Double-Sided Ring Structure of the Collector System Planning of Large Offshore Wind Farms". In: *IEEE Transactions on Sustainable Energy* 8.3 (2017). ISSN: 19493029. DOI: 10.1109/TSTE.2016.2646061.
- [54] Wu, X., Hu, Y., Li, Y., Yang, J., Duan, L., Wang, T., Adcock, T., Jiang, Z., Gao, Z., Lin, Z., Borthwick, A., and Liao, S. "Foundations of offshore wind turbines: A review". In: *Renewable and Sustainable Energy Reviews* 104 (2019). ISSN: 18790690. DOI: 10.1016/j.rser.2019.01.012.
- [55] Wu, Y. W. and Wang, Y. "Collection line optimisation in wind farms using improved ant colony optimisation". In: *Wind Engineering* 45.3 (2021). ISSN: 2048402X. DOI: 10.1177/0309524X20917319.
- [56] Xu, H., Rui, S., Shen, K., Jiang, L., Zhang, H., and Teng, L. *Shared mooring systems for offshore floating wind farms: A review*. 2024. DOI: 10.1016/j.enrev.2023.100063.
- [57] Yen, J. Y. *Finding the K Shortest Loopless Paths in a Network*. Tech. rep. 11. 1971, pp. 712–716.
- [58] Yi, X., Scutariu, M., and Smith, K. "Optimisation of offshore wind farm inter-array collection system". In: *IET Renewable Power Generation* 13.11 (2019). ISSN: 17521424. DOI: 10.1049/iet-rpg.2018.5805.
- [59] Young, D. *Predicting Dynamic Subsea Cable Failure for Floating Offshore Wind*. Tech. rep. 2018.
- [60] Zaaijer, M., Viré, A., Dos, R. B., Pereira, S., and Daneshbodi, A. *Introduction to wind turbines: physics and technology*. Tech. rep. 2023.

- [61] Zhong, W., Flemming, R., Zhong Shen, W., and Mikkelsen, R. "Study on wind turbine arrangement for offshore wind farms". In: *Citation* (2018).
- [62] Zuo, T., Meng, K., Tong, Z., Tang, Y., and Dong, Z. H. "Offshore wind farm collector system layout optimization based on self-tracking minimum spanning tree". In: *International Transactions on Electrical Energy Systems* 29.2 (2019). ISSN: 20507038. DOI: 10.1002/etep.2729.
- [63] Zuo, T., Zhang, Y., Meng, K., and Dong, Z. Y. "Collector System Topology for Large-Scale Offshore Wind Farms Considering Cross-Substation Incorporation". In: *IEEE Transactions on Sustainable Energy* 11.3 (2020). ISSN: 19493037. DOI: 10.1109/TSTE.2019.2932409.
- [64] Zuo, T., Zhang, Y., Meng, K., Tong, Z., Dong, Z. Y., and Fu, Y. "A Two-Layer Hybrid Optimization Approach for Large-Scale Offshore Wind Farm Collector System Planning". In: *IEEE Transactions on Industrial Informatics* 17.11 (2021). ISSN: 19410050. DOI: 10.1109/TII.2021.3056428.
- [65] Zuo, T., Zhang, Y., Meng, K., Tong, Z., Dong, Z. Y., and Fu, Y. "Collector System Topology Design for Offshore Wind Farm's Repowering and Expansion". In: *IEEE Transactions on Sustainable Energy* 12.2 (2021). ISSN: 19493037. DOI: 10.1109/TSTE.2020.3022508.