國立臺灣大學工學院工業工程學研究所

博士論文

Institute of Industrial Engineering

College of Engineering

National Taiwan University

Doctoral Dissertation

資料驅動最佳化之產業應用

Industrial Application of Data-Driven Optimization Techniques

羅迪納

Rudi Nurdiansyah

指導教授：洪一薰 博士

Advisor: I-Hsuan Hong, Ph.D.

共同指導教授：蘇哲平 博士

Co-Advisor: Jack C.P. Su, Ph.D.

中華民國 113 年 5 月

May 2024

doi:10.6342/NTU202403713

Acknowledgements

Undertaking this PhD has been a truly life-changing experience for me, and it would not have been possible without the support and guidance I received from many people.

Firstly, I would like to express my sincere gratitude to my supervisor, Prof. I-Hsuan Hong, for his guidance, support, and encouragement throughout this journey. His expertise, mentorship, and constructive feedback have been instrumental in shaping this thesis.

I also want to extend my gratitude to Prof. Jack C.P. Su, whose convincing discussions greatly influenced this work.

I am thankful for the funding received towards my PhD from The Islamic Development Bank and The Ministry of Education, Culture, Research, and Technology of Indonesia. Additionally, I acknowledge the funding provided for my research during my PhD from the Ministry of Science and Technology and the Ministry of Science and Technology, Taiwan, and National Taiwan University.

My appreciation goes to my colleagues and friends for their encouragement, discussions, and moral support during challenging times.

I owe a debt of gratitude to my family for their unwavering love, understanding, and encouragement throughout this journey. Their support has been my pillar of strength.

Lastly, I would like to acknowledge the participants of this study for their valuable contributions and cooperation, without whom this research would not have been possible.

中文摘要

在數位轉型時代，各行各業正利用先進的數據驅動優化技術來提高效率、降低成本並改善決策過程。這些技術結合了大數據、機器學習和人工智慧，通過創建模型和模擬來預測結果並提供最佳策略，徹底改變了傳統做法。然而，現實世界中的優化問題往往是複雜的、多目標的且資源密集型的，需要使用進化算法和基於模擬的優化等複雜的方法。替代模型等技術有助於減少計算成本，但會引入近似誤差。製造業、能源、海事和農業等行業從數據驅動優化中受益顯著，應對數據稀缺、噪聲和不平衡等挑戰。這些技術的框架包括數據收集、模型開發和計算，確保模型的穩健性和適應性。本研究探討了在製造排隊時間環生產系統、海洋渦輪模擬校準和海洋農場佈局優化中的應用，展示了性能和效率方面的顯著改進.

關鍵詞：數據驅動優化、數據建模、工業效率、優化算法

doi:10.6342/NTU202403713

Abstract

In the digital transformation era, industries are leveraging advanced data-driven optimization techniques to enhance efficiency, reduce costs, and improve decision-making processes. These techniques integrate big data, machine learning, and artificial intelligence, revolutionizing traditional practices by creating models and simulations to predict outcomes and suggest optimal strategies. However, real-world optimization problems are often complex, multi-objective, and resource-intensive, requiring sophisticated approaches like evolutionary algorithms and simulation-based optimization. Techniques such as surrogate models help mitigate computational costs but introduce approximation errors. Industries like manufacturing, energy, maritime, and agriculture benefit significantly from data-driven optimization, addressing challenges like data scarcity, noise, and imbalance. A framework for these techniques involves data collection, model development, and computation, ensuring robust and adaptable models. This study explores applications in manufacturing queue time loop production systems, marine turbine simulation calibration, and marine farm layout optimization, demonstrating significant improvements in performance and efficiency.

Keywords: data-driven optimization, data modelling, industrial efficiency, optimization algorithms

doi:10.6342/NTU202403713

Table of Contents

doi:10.6342/NTU202403713

doi:10.6342/NTU202403713

List of Figures

doi:10.6342/NTU202403713

List of Tables

doi:10.6342/NTU202403713

doi:10.6342/NTU202403713

Chapter 1 Introduction

In today's digital transformation era, industries across various sectors are increasingly adopting advanced data-driven optimization techniques to enhance efficiency, cut costs, and improve decision-making processes (Georgiadis et al., 2019; Shen et al., 2020; Brunton et al., 2021). The integration of these techniques is revolutionizing traditional industrial practices, enabling organizations to utilize the power of big data, machine learning, and artificial intelligence. Data-driven optimization refers to the process of using data analytics and computational algorithms to make informed decisions and optimize operational performance (Kim & Boukouvala, 2020). This approach leverages the large amounts of data generated in industrial environments to create models and simulations that can predict outcomes, identify inefficiencies, and suggest optimal strategies (Grillone et al., 2020; Seyedzadeh et al., 2020; Mandl & Minner, 2023).

However, many real-world optimization problems are highly complex, multi-objective, highly constrained, multi-modal, non-convex, large-scale, and fraught with ambiguities (Jin et al., 2018). Additionally, formulating these optimization problems is challenging, requiring appropriate specifications for representation, objectives, constraints, and decision variables (Jin & Sendhoff, 2009; Chugh et al., 2017; Sindhya 2014).

To tackle these challenges, researchers have developed various approaches and algorithms. An example of these approaches is Evolutionary algorithms (EAs). EAs typically assume that evaluating objectives and constraints is simple and inexpensive (Jin et al., 2018). However, in many real-world scenarios, evaluations can only be conducted using data collected from physical experiments, numerical simulations, or daily

doi:10.6342/NTU202403713

operations. These scenarios are known as data-driven optimization problems (Wang et al., 2016). The data involved can be distributed, noisy, heterogeneous, or dynamic, and can range from abundant to scarce, each posing unique challenges to the optimization process.

In certain cases, assessing the objective or constraint functions requires time-consuming or resource-intensive physical experiments or numerical simulations, commonly known as simulation-based optimization. For instance, evaluating a single function using computational fluid dynamics simulations might take up to an hour of computation time (Olson et al., 2021). Techniques such as surrogate models, including polynomial regression (Liu et al., 2020), Kriging model (Santos et al., 2023; Koziel et al., 2020), artificial neural networks (Cho et al., 2021; Yan et al., 2020), and radial basis function networks (Urquhart et al., 2020; Ji et al., 2023), are used to reduce computational costs. These models rely on limited training data to approximate the objective and constraint functions (Chugh et al., 2019), which can lead to approximation errors and potentially mislead the optimization search.

Conversely, when large amounts of data are available, acquiring and processing this data can significantly increase resource and computational costs (Zhou et al., 2014). Furthermore, evaluations based on data are approximations of exact evaluations, as the data is often (Liu et al., 2014), imbalanced (Wang et al., 2014; Wang & Yao, 2012), and noisy (Jin & Branke, 2005; Wang et al., 2015). These imperfections can introduce errors into the optimization process, complicating the search for optimal solutions.

Industries such as manufacturing, energy, maritime, and agriculture can greatly benefit from data-driven optimization. These sectors often deal with complex systems and large-scale operations where traditional optimization methods may be insufficient.

By adopting data-driven approaches, these industries can achieve significant improvements in various aspects of their operations, leveraging the vast potential of data to enhance performance and drive innovation.

A framework, as shown in Fig. 1.1, is utilized to illustrate the data-driven optimization techniques explored in this study. The framework in Fig. 1.1 begins with the crucial phase of data collection and pre-processing. Historical data is gathered from diverse sources, encompassing a wide array of information pertinent to the industrial processes under consideration. The pre-processing stage involves structuring the raw data, ensuring it is suitable for further analysis. Modeling forms a significant aspect of this phase, where decision variables, objectives, and constraints are defined. Decision variables represent the controllable factors in the system, objectives denote the desired outcomes or goals, and constraints outline the limitations within which the system operates.

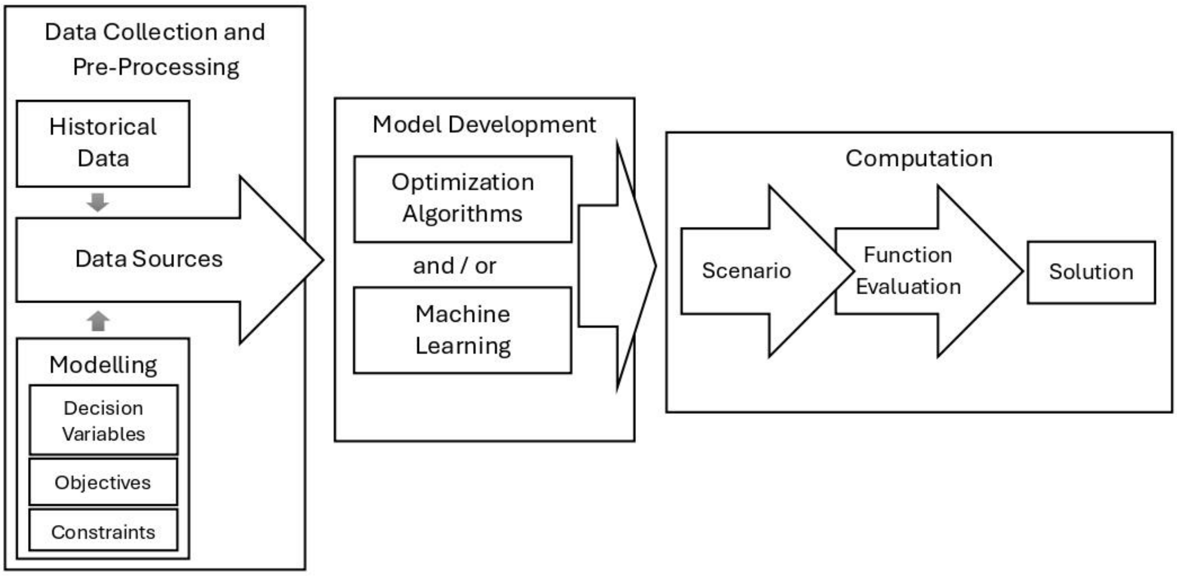


Fig. 1.1 Framework of data-drive optimization techniques.

doi:10.6342/NTU202403713

Moving on to the second phase, model development is a critical component where the pre-processed data is utilized to create sophisticated models. This involves the application of optimization algorithms and / or machine learning techniques. Optimization algorithms are mathematical procedures used to identify the best possible solutions within the defined constraints, aiming to maximize or minimize specific objectives. Examples include heuristics and metaheuristics algorithms. On the other hand, based on the data, machine learning techniques are employed to analyze patterns and make predictions. These methods can vary from basic regression models to advanced neural networks, equipping the framework with the ability to address a diverse range of industrial situations. By integrating these advanced methodologies, the framework ensures that the models developed are both robust and adaptable to changing industrial conditions.

The final phase of the framework is computation, where the developed models are put to the test under various scenarios. Each scenario represents a different set of conditions or inputs, allowing the evaluation of the model's performance in diverse contexts. Function evaluation is conducted to assess how well the model meets the defined objectives while adhering to the constraints. This step is crucial as it provides insights into the effectiveness and efficiency of the model. The ultimate goal is to derive a solution, which is the optimal set of decisions or actions recommended by the model. This solution aims to enhance industrial operations by optimizing performance, reducing costs, or improving resource utilization. By systematically following this framework, industries can leverage data-driven optimization techniques to achieve significant improvements in their processes and outcomes.

This study delves into three specific applications of data-driven optimization techniques: queue time loop production systems in manufacturing, calibration of marine

doi:10.6342/NTU202403713

turbine simulations, and marine farm layout optimization. In Chapter 2, we address the dynamic nature of queue-time loop in flow shop production systems using a deterministic model. This involves frequent updates of job arrivals and machine failures and recoveries to accurately reflect real-world dynamics. To tackle this challenge, we utilize combinatorial Benders' cut (CBC) to solve the problem and introduce the Phase-Step method (PS), a heuristic approach, to reduce the number of decision variables. The PS method focuses on the reduced model, disregarding relatively distant production schedules. Under the PS method, not all jobs are scheduled at every decision time point until the final stage. To enhance its effectiveness, we propose combining the PS method with CBC, resulting in CBC-PS. This approach allows us to manage the stochastic nature of a queue-time loop production system by making admission control decisions at each stage and then dynamically rescheduling production in real-time. Our findings indicate that CBC-PS outperforms three methods, i.e., reaction chains heuristics, threshold dispatching rule, and first-in-first-out, particularly in scenarios with high job arrival rates, in terms of minimizing the number of scrap jobs.

In Chapter 3, we investigate the utilization of artificial neural networks together with metaheuristics algorithms to calibrate turbulence parameters for a marine turbine simulation. We propose a computationally efficient method to determine the optimal values of adjustable parameters in the marine turbine simulator while minimizing the disparity between simulation outputs and physical experimental results. Our findings indicate a notable enhancement in performance by 16.97% compared to previous approaches, facilitated by the integration of machine learning, data modeling, and advanced optimization techniques. Additionally, we identify a straightforward opportunity for further performance improvement. A visual tool is devised utilizing descriptive statistics to quickly and easily assess the quality of sample data.

doi:10.6342/NTU202403713

In Chapter 4, we propose a bi-level optimization model addressing the marine current turbine layout problem, focusing on the design of the marine farm. To tackle this model, we introduce the Greedy Heuristic Algorithm (GHA), specifically designed to effectively handle such complexities. Given the inherent ambiguity of parameters in the marine farm, we develop a non-parametric scenario-based robust optimization model. Initially, each scenario within the proposed model is addressed using the GHA. Subsequently, leveraging the solutions obtained, we introduce a robust GHA to solve the robust optimization model. This aims to identify a robust solution employing the min-max relative regret decision rule. Our findings demonstrate that the robust solution performs admirably across all examined realizable scenarios.

The remainder of the thesis is organized as follows: Chapter 2 details the addressing of real-time admission control in a queue-time loop production system using the CBC-PS method. Chapter 3 discusses the calibration of turbulence parameters for the marine turbine using a surrogate model and metaheuristics algorithm. Chapter 4 addresses the robust bi-level optimization model of the marine current turbine layout problem using GHA. Chapter 5 provides a conclusion of the study.

Chapter 2 Real-time framework for admission control in queue-time loop production systems[1]

A typical flow shop production system constrained by queue times, comprises a loop with multiple steps where queue time is restricted between each step. To handle dynamic environments inherent in such systems, we regularly update machine availability and job arrivals, promptly rescheduling production accordingly. As the number of jobs increases, the problem's complexity grows exponentially, making computational time a critical factor in problem-solving success. A mixed integer linear programming (MILP) model designed for modeling queue-time loop production systems is involved in our proposed solution. To efficiently solve this model, we employ Combinatorial Benders' cut (CBC), which decomposes variables into integer and continuous components. Additionally, we introduce the phase-step (PS) method to reduce model size. By integrating both CBC and PS methods, known as CBC-PS, we streamline the production scheduling process in queue-time loop systems, effectively reducing computational burden. In our simulation study, CBC-PS outperforms existing heuristics such as first-in-first-out (FIFO), threshold dispatching (TH), and reaction chains (RC) by reducing scrap job instances by up to 39.1%.

[1] This chapter is rewritten based on an article authored by this thesis writer titled “Real-time admission control in a queue-time loop production system” which is published in Computers and Industrial Engineering in 2021 (Nurdiansyah et al., 2021).

doi:10.6342/NTU202403713

2.1 Problem background

Queue time constraints in flow shop scheduling problems arise in various industries, including food processing, semiconductor manufacturing, and steelmaking. These systems impose a maximum time limit on the queue between successive stages (Akkerman et al., 2007; Harjunkoski & Grossman, 2001; Joo & Kim, 2009; Scholl & Domaschke, 2000). Violating these constraints can lead to significant losses, including production inefficiencies or the generation of scrap jobs. Research by Hsieh et al. (2002) reveals that in semiconductor manufacturing, approximately 30% of stages in a 200-mm fabrication process have queue time constraints, resulting in over 50% of jobs being reprocessed due to exceeding these limits. For instance, in wafer fabrication, once treatments in furnace tubes are completed, the queue time for wafers to proceed to downstream stages is restricted to prevent particulate absorption in the air (Su, 2003). Similarly, the food industry must refrigerate certain prepared and cooked foods for a specific duration to prevent spoilage; otherwise, they must be discarded (Hodson et al., 1985).

When industries encompass numerous stages, whether with or without queue time constraints between successive stages, such as in the glass industry and semiconductor manufacturing, it becomes feasible to group all stages into several queue-time loops. A queue-time loop refers to a scenario where the entire production system comprises consecutive stages with queue time constraints. Within such a loop, all jobs are scheduled while adhering to these constraints. Jobs between any two loops can be temporarily placed in a storage without breaching the queue time limits. This study examines a loop with queue time limit comprising multiple machines or stages, enabling to solve the resulting smaller-scale problems within a reasonable timeframe and computational capacity.

doi:10.6342/NTU202403713

In practical scenarios, the arrival of jobs may occur randomly, and machines may fail unexpectedly, leading to disruptions in operations. Recoveries from these uncertain events pose challenges in capacity planning, thereby elevating the likelihood of violations of queue time constraints (Wu et al., 2012). Previous research by Wu et al. (2010), Wu et al. (2012), and Wu et al. (2016a) has devised Markov decision models to address the complexities of a two-stage production system with queue time constraints, incorporating considerations of machine availability status. Lee et al. (2005) introduced reaction chains (RC) as a mechanism to mitigate the influence of queue time control on tool availability in semiconductor manufacturing. Wu et al. (2016b) proposed a threshold dispatching rule (TH) to minimize capacity loss and rework rate by managing the work-in-process (WIP) level in a time-constrained production environment.

Prior studies in production control systems predominantly focus on the status of machine failures and recoveries. In contrast, our investigation accounts for the most recent updates on job arrivals and machine failures and recoveries, aligning with the approach adopted by Wang et al. (2018) and Hong et al. (2019). Notably, the queue time constraint formulation and the underlying causes of violations distinguish our study from Wang et al. (2018). Additionally, Hong et al. (2019) segment the continuous time window into discrete units, potentially resulting in larger problem sizes. To avoid this issue, our modelling approach directly addresses the starting and completion times of jobs within a continuous time framework.

This study addresses the challenges of applying stochastic models in real-world production systems. While multistage stochastic models are a potential solution, they often rely on estimated parameter distributions which may be unavailable or non-stationary (Madathil et al., 2018). To overcome these limitations, we formulate a novel queue-time loop scheduling system using Mixed Integer Linear Programming (MILP) model. Our approach utilizes a deterministic MILP model to capture the dynamic nature of the system. We achieve this by frequently updating machine availability and job arrival information, followed by prompt rescheduling based on this latest information. This process forms a closed-loop system. First, we obtain real-time information on machine availability (including failures and recoveries) and job arrivals. This data is then used to solve the MILP model, which provides admission control decisions for each stage in the production process. Based on these decisions, a new production schedule is generated. Finally, the loop closes by updating the machine and job arrival data to reflect the current state of the system. The core of the proposed framework lies in a rolling horizon mechanism. Similar to existing literature (Qin et al., 2018; Kamran et al., 2019), this mechanism triggers rescheduling based on a combination of events and time intervals. To account for the dynamic environment, we update machine availability and job arrivals at each decision point, enabling real-time response (Qin et al., 2018).

The concept of rolling horizon mechanisms for dynamic environments finds its roots in the work of Sethi and Sorger (1991). Tang et al. (2005) and Qin et al. (2018) utilize event-driven rescheduling, while Stauffer and Liebling (1997) and Shafaei and Brunn (1999) employ time-based intervals. Fang and Xi (1997) explored a mixed approach based on both events and cycles, highlighting the superiority of rolling horizon strategies over static scheduling.

Our work distinguishes itself by utilizing shorter rescheduling intervals compared to existing studies. While this approach offers faster response to system changes, it also leads to larger problem sizes. To address this, the Phase-step method (PS) is introduced in this study. This method reduces the size of the problem by disregarding the later production schedules where all jobs are not scheduled until the final machine or stage.

doi:10.6342/NTU202403713

The rolling horizon method is used by Kamran et al. (2019) and Addis et al. (2016) to allocate patients in the operating room scheduling problem. Both studies share constraints similar to ours, where the maximum waiting time must not exceed a specified duration. Table 2.1 provides an overview of relevant literature employing the rolling horizon method.

Given the computational complexity and time-consuming nature of frequently updating machine availabilities, job arrivals, and rescheduling production plans, a solution strategy based on combinatorial Benders' cuts (CBC) is adopted. CBC separates an MILP model into integer problems (defined as master problems) and continuous problems (defined as slave problems). The optimal solution is obtained through iterative solving steps. It identifies the smallest subset of variables that address infeasibility, resolving it by modifying at least one integer variable. Conversely, traditional Benders' decomposition relies on solving the problem using a dual solution approach, dividing it into subsets (Chen et al., 2012 and Taskn & Cevik, 2013).

Researchers have observed several benefits of CBC. In the branch and bound tree, Tanner and Ntaimo (2010) highlight its ability to reduce the number of nodes investigated and solution times. Compared to the branch and cut approach, Chen et al. (2012) emphasize its performance. Solution quality and computation time are noted by Verstichel et al. (2015) compared to exact approaches. CBC capability to handle small to medium-sized instances effectively is also demonstrated by Akpinar et al. (2017). Moreover, recent studies by Nurdiansyah and Hong (2018) indicate that CBC yields superior solutions with a reasonable computational time compared to the CPLEX solver.

doi:10.6342/NTU202403713

Table 2.1 Literature survey

Literature	Problem	Interval of reschedule	Solution method
Stauffer and Liebling (1997)	Job shop	Daily	Tabu search
Shafaei and Brunn (1999)	Open shop	Hourly	Dispatching rules
Tang et al. (2005)	Hybrid flow shop	-	Neural network
Addis et al. (2016)	Operating room	Weekly	Integer Linear Programming
Lv et al. (2017)	Hybrid flow shop scheduling	-	Heuristics algorithm
Qin et al. (2018)	Hybrid flow shop scheduling	-	Metaheuristics algorithm
Kamran et al. (2019)	Surgery room	-	Heuristics algorithm
Present study	Hybrid Flow shop with queue time constraints	In minutes	Combinatorial Benders' cut + Phase-step method

Addressing the sophistication of the problem, the phase-step (PS) method is proposed, reducing the model size. Heuristic methods, commonly used in practice to tackle real-world problems, have shown promising performance in flow shop scheduling challenges. The policy of production control in this study regularly updates job arrivals and machine availability followed by rearranging schedule in real-time. As a result, the previous production schedule becomes unusable at the current decision time point, eliminating the need to compute the schedule for all jobs until the final stage. The PS method achieves this by reducing the number of future stages considered. Additionally,

doi:10.6342/NTU202403713

we propose a combination of the CBC and PS methods, referred to as CBC-PS. Through an investigation spanning 200,000 minutes of plant operations, our findings indicate that the number of scrap jobs can be reduced up to 39.1% by the CBC-PS compared to RC, TH, and FIFO approaches.

The contributions of this research to the existing literature are in several ways. Firstly, it illustrates and demonstrates the utilization of a deterministic model to address dynamic production system challenges. Secondly, it groups consecutive stages with queue-time constraints into queue-time loops within the production system. Thirdly, the proposed PS method reduces the number of variables in the model. Lastly, theoretical findings reveal that the production schedule derived from the CBC-PS method aligns closely with the optimal production schedule. Next, we describe the problem and the modelling strategy, and formulates the MILP model in subchapter 2.2. Subchapter 2.3 explains the CBC, PS method, and combination of them. Subchapter 2.4 outlines the numerical study. Subchapter 2.5 explains the conclusion of the research.

2.2 Mathematical Model

In the semiconductor manufacturing, glass and food industries, there are various machines or stages in their manufacturing process that are subject to a queue time constraint between successive stages, forming what is termed as a queue-time loop. Our focus lies in a queue-time loop with numerous stages where each stage contains several machines identically. Allocation of the jobs to each machine to obtain the optimal objective function is the goal. Each job has to undergo processing on a designated machine, without interruption, following a consistent production flow. This research follows these assumptions.

doi:10.6342/NTU202403713

1. Jobs undergo successive stages following the identical production sequence.
2. Each machine handles only one task at a time, and every task must be executed on a machine at each stage without interruption.
3. A task may advance to the subsequent stage upon completion of the current one.
4. At each subsequent stage, distinct upper limits impose constraints on queue durations, preventing tasks that surpass these limits from further processing, resulting in their elimination.

Fig. 2.1 depicts the concept of a queue-time loop production system. Each stage is equipped with a single queue and M parallel machines. Let us assume that queue 1 has four jobs and stage 1 has two parallel machines. The first and second jobs undergo processing on the first and second machines at stage 1, while the third and fourth jobs await processing. Upon completion of stage 1, the first and second jobs transfer to queue 2, while the third and fourth jobs are handled at stage 1. Upon the entry of the first and second jobs into queue 2, the limit of queue time initiates. The queue time the first and second jobs in queue 2 is restricted by the queue time limit. If the violation of queue time limit occurs the jobs are discarded. These procedures are followed by all jobs at each stage till reaching the terminal stage, exiting this loop upon terminal stage completion.

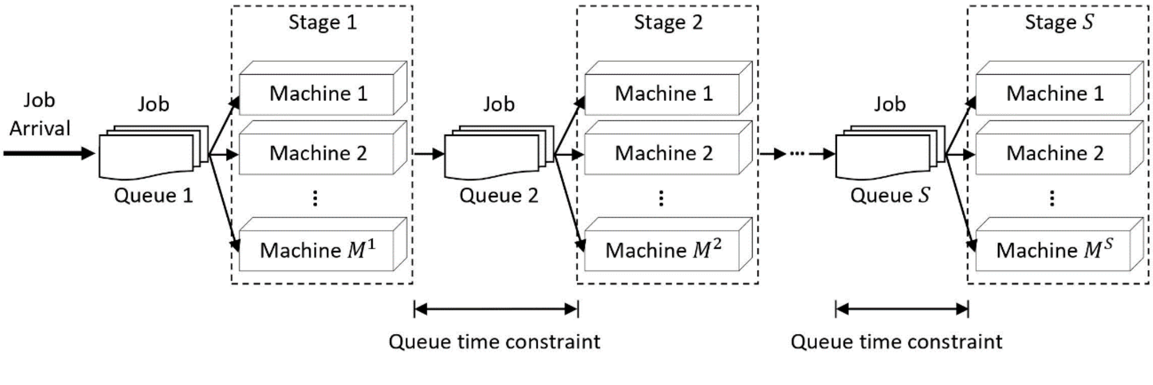


Fig. 2.1. Flow of queue-time loop system (adapted from Hong et al., 2019)

doi:10.6342/NTU202403713

We construct an MILP model for a queue-time loop scheduling problem. We notate the model's parameters and decision variables as follows:

Parameters

S number of stage;

M^f number of machines at stage *f*;

Ω_q bundle of jobs in queue *q*;

K_q number of jobs in queue *q*;

p^f processing time at stage *f*;

qT^f queue time constraint of stage *f*;

a_j^f beginning time of machine *j* can process at stage *f*;

T present time.

Decision variables

$s_{q,i}^f$ start time of job *i* in queue *q* at stage *f*;

$c_{q,i}^f$ finish time of job *i* in queue *q* at stage *f*;

$e_{q,i}^f$ queue time breach of job *i* in queue *q* at stage *f*;

$y_{q,i,j}^f$ binary variable (= 1 if job *i* in queue *q* is treated in machine *j* at stage *f*, = 0 otherwise).

We adapt Lee (2017) to formulate the queue-time loop scheduling problem as follows:

$$\underset{s_{q,i}^f,\, e_{q,i}^f}{\text{minimize}} \sum_{f=1}^{S} \sum_{q=f}^{S} \sum_{i=1}^{|\Omega|} \left(s_{q,i}^f + \frac{\alpha}{S-f} \cdot e_{q,i}^f \right). \tag{2.1}$$

Based on 2.1, the objective function is for minimizing the start time of all jobs at each stage and queue time breach, where α is the penalty weight of the breaching of queue time limit. We penalize heavily if the queueing time in a later step (downstream stage) exceeds its limit. This is because a queue overflow there means all the earlier steps (upstream stages) were also overloaded. Here are the specific constraints:

$$\sum_{j=1}^{M^f} y^f_{q,i,j} = 1, \forall f \in \{1,\dots,S\}, \forall q \in \{f,\dots,S\}, \forall i \in \Omega_q \tag{2.2}$$

$$s^f_{q,i} + \sum_{j=1}^{M^f} y^f_{q,i,j} \cdot p^f = c^f_{q,i}, \forall f \in \{1,\dots,S\}, \forall q \in \{f,\dots,S\}, \forall i \in \Omega_q \tag{2.3}$$

$$s^f_{q,i} \geq \sum_{j=1}^{M^f} a^f_j \cdot y^f_{q,i,j}\,, \forall f \in \{1,\dots,S\}, \forall q \in \{f,\dots,S\}, \forall i \in \Omega_q \tag{2.4}$$

$$s^f_{q,i} \geq c^{f-1}_{q,i}, \forall f \in \{2,\dots,S\}, \forall q \in \{f,\dots,S\}, \forall i \in \Omega_q \tag{2.5}$$

$$s^1_{1,i} \geq T, \forall i \in \Omega_1 \tag{2.6}$$

Constraint (2.2) specifies that each job must be processed on one of the machines at each stage. Constraint (2.3) determines the completion time of job i during operation q by summing the starting time of job i in operation q with the processing time of machine j at stage f. Constraint (2.4) requires that the starting time of job i in operation q on machine j must be greater than or equal to the allowable time. Constraint (2.5) ensures that job i can only begin processing at stage f if it has been completed at the previous stage. Constraint (2.6) mandates that the starting time of job i at the first stage must not be earlier than the decision time. At the start of each experiment, the time always begins at zero.

$$e_{q,i}^{f} = max\{(s_{q,i}^{f} - c_{q,i}^{f-1} - qT^{f}),0\}, \forall f \in \{2,\dots,S\}, \forall q \in \{f,\dots,S\}, \forall i \in \Omega_{q} \quad (2.7)$$

$$s_{q,i+k}^{f} + M\left(2 - y_{q,i,j}^{f} - y_{q,i+k,j}^{f}\right) \geq c_{q,i}^{f}, \forall f \in \{1, \dots, S\},$$
$$\forall q \in \{f,\dots,S\}, \forall i \in \{1,\dots,K_{q}-1\}, \forall k \in \{1,\dots,K_{q}-i\}, \forall j \in \{1,\dots,M^{f}\} \quad (2.8)$$

$$s_{q,i+k}^{f} \geq s_{q,i}^{f}, \forall f \in \{1, \dots, S\}, \forall q \in \{f, \dots, S\}, \forall i \in \{1, \dots, K_{q}-1\},$$
$$\forall k \in \{1,\dots,K_{q}-i\} \quad (2.9)$$

Referring Nurdiansyah et al. (2021), constraint (2.7) defines a queue time breach, which occurs if the queue time—calculated as the starting time of job i at the current stage minus the completion time of job i at the previous stage—exceeds the maximum allowable queue time at stage f. Constraint (2.8) specifies that if jobs i and $i+k$ are assigned to the same machine j at stage f, then both $y_{q,i,j}^{f}$ and $y_{q,i+k,j}^{f}$ must be set to 1, where K_{q} represents the number of jobs in operation q. This means that job i must be completed before job $i+k$ (i.e., $s_{q,i+k}^{f} \geq c_{q,i}^{f}$). If either $y_{q,i,j}^{f}$ or $y_{q,i+k,j}^{f}$ is not equal to 1, then constraint (2.8) does not impose any restrictions due to the large value of M in the model. Additionally, when jobs are assigned to different machines at the same stage, constraint (2.9) requires that each job's starting time must respect the order of jobs in the queue. The settings for nonnegative and binary variables are specified in constraints (2.10) to (2.13).

doi:10.6342/NTU202403713

$$s^{f}_{q,i} \geq 0, f \in \{1, \dots, S\}, q \in \{f, \dots, S\}, i \in \Omega_q \tag{2.10}$$

$$c^{f}_{q,i} \geq 0, f \in \{1, \dots, S\}, q \in \{f, \dots, S\}, i \in \Omega_q \tag{2.11}$$

$$e^{f}_{q,i} \geq 0, f \in \{1, \dots, S\}, q \in \{f, \dots, S\}, i \in \Omega_q \tag{2.12}$$

$$y^{f}_{q,i,j} \in \{0,1\}, f \in \{1, \dots, S\}, q \in \{f, \dots, S\}, i \in \Omega_q, j \in \{1, \dots, M^f\} \tag{2.13}$$

To manage the dynamic conditions in a queue-time loop production system using the proposed deterministic MILP model, we employ a rescheduling strategy. This strategy involves iteratively rescheduling jobs at each stage whenever there are updates to job arrival times and machine availabilities, as illustrated in Figure 2.2. At each decision time, the statuses of machine failures, recoveries, and job arrivals are updated, providing the input data for the MILP model. After solving the MILP model, the solution, which specifies the starting time of each job at each stage, is used to make the admission control decision. This updated admission control decision enables us to regenerate the production log file, which represents the production schedule from the previous decision point. The production log file is then revised based on the updated admission control decision, and this cyclical process continues until the predefined number of decision points is reached.

2.3 Solution approach

Initially, we explain the CBC approach for addressing the MILP model. Subsequently, we introduce the PS technique aimed at diminishing the model size.

2.3.1 Combinatorial Benders' cuts

The CBC method has shown promising results in terms of solution quality and computation time in various studies (Tanner & Ntaimo, 2010; Verstichel et al., 2015; Akpinar et al., 2017). Using CBC to address the proposed MILP model has the benefit of breaking the problem down into a master problem and a slave problem, thus simplifying it into two more manageable parts. According to Codato and Fischetti (2006), the master problem involves only integer variables for its objective function and constraints, while the slave problem deals with the remaining continuous variables and their constraints. In our model, the objective function is composed entirely of continuous variables. Therefore, the MILP model is divided into a master problem with its associated constraints and a slave problem with an objective function and constraints related to continuous variables. Below, we provide the formulations for both the master and slave problems.

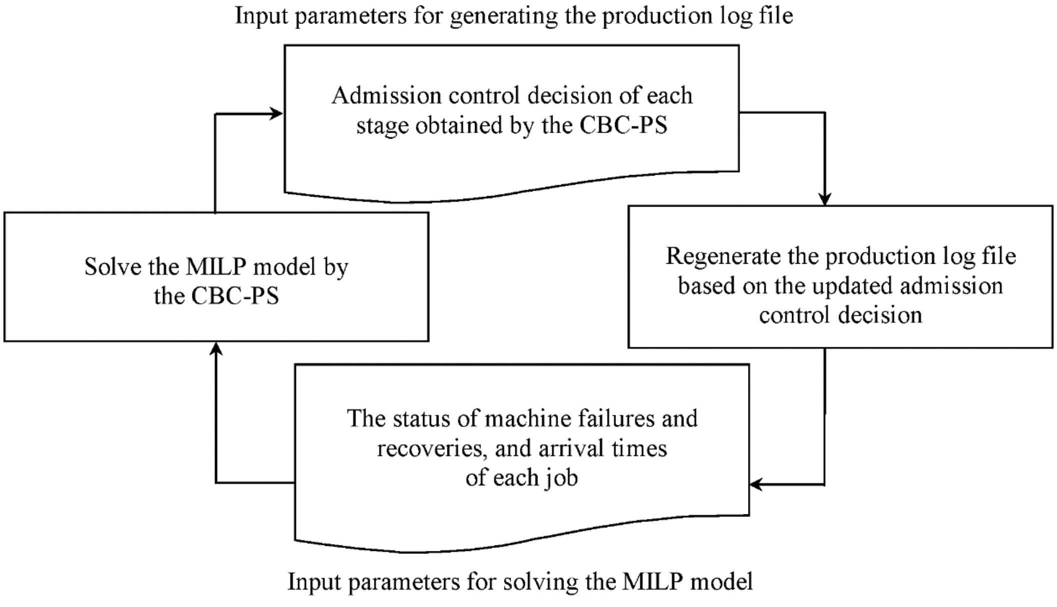

Input parameters for generating the production log file
Admission control decision of each stage obtained by the CBC-PS
Solve the MILP model by the CBC-PS
Regenerate the production log file based on the updated admission control decision
The status of machine failures and recoveries, and arrival times of each job
Input parameters for solving the MILP model

Fig. 2.2. The cyclic method (adapted from Hong et al., 2019).

doi:10.6342/NTU202403713

Masters: Constraints (2.2)

Slave: Objective function (2.1)

s.t.

Constraints (2.3) – (2.13)

In a specific scenario where the original MILP problem's objective function does not include integer variables, a bound inequality like (2.14) is introduced to the slave problem. In this context, the upper bound (UB) represents the objective function value of the current solution, and ε is a sufficiently small positive number. Unlike the standard CBC method, which only generates a cut when the slave problem is infeasible, this study creates a cut both when the slave problem is infeasible or feasible. This cut is then added back to the master problem. The purpose of this approach is to prevent the master problem from reproducing the same solution as in previous iterations; otherwise, the same constraint from the master problem is enforced.

Master (integer) problem handles allocation of jobs to machines at each stage, i.e. $y^{f}_{q,i,j}$, while subsequently, the continuous problem assesses the feasibility of solution derived from master problem. Notably, the slave problem involves continuous variables $s^{f}_{q,i}$, $c^{f}_{q,i}$, and $e^{f}_{q,i}$. In cases where the original MILP problem's objective function lacks integer variables, as described by Codato and Fischetti (2006), continuous problem introduces a bound inequality (2.14). In this context, the upper bound (UB) is defined as the objective function value of the current solution, and ε is a very small positive number. Our study deviates from the conventional implementation of the CBC approach. Traditionally, CBC introduces a cut only when the slave problem is found to be infeasible. However, our method incorporates a cut regardless of whether the slave problem produces a feasible or infeasible solution, subsequently reintroducing this cut to the master

doi:10.6342/NTU202403713

problem. This adjustment is designed to prevent the master problem from generating the same solution as in the previous iteration, thus avoiding the reapplication of identical constraints. Considering the model's objective function, as detailed in subchapter 2.2, focuses on minimizing the start time of all jobs at each stage and reducing queue time violations, CBC consistently aims to minimize these objectives with each iteration.

CBC initiates by addressing the master problem and obtaining the solution outlined in (2.2), denoted by $Y^* = \left\{ {y^f_{q,i,j}}^* : f = 1, \dots, S,\ i \in \Omega_q, j = 1, \dots, M^f, q = f, \dots, S \right\}$. Subsequently, Y^* is utilized in addressing the slave problem. After resolving the slave problem, in the event of infeasibility during the current iteration, cut (2.15) is formulated and reintegrated into the master problem. Conversely, if the slave problem proves feasible, the optimal objective value, denoted as Z^*, is determined, and the current best solution is updated using $(Y^*,\ Z^*)$. Furthermore, cut (2.15) is generated anew and reinserted into the master problem. The entire process concludes when we have infeasible master problem. **Algorithm 2.1** details the specific procedures of the CBC.

$$\sum_{q=1}^{S} \sum_{i=1}^{|\Omega_q|} \sum_{f=q}^{S} \left(s^f_{q,i} + \frac{\alpha}{S-f} \cdot e^f_{q,i} \right) \leq UB - \varepsilon \tag{2.14}$$

$$\sum_{j:\{y^f_{q,i,j} \in Y^* : {y^f_{q,i,j}}^* = 0\}} y^f_{q,i,j} + \sum_{j:\{y^f_{q,i,j} \in Y^* : {y^f_{q,i,j}}^* = 1\}} (1 - y^f_{q,i,j}) \geq 1 \tag{2.15}$$

The CBC algorithm swiftly attains the optimal solution by breaking down the initial MILP model into distinct master and slave problems. Nevertheless, the complexity of the problem discussed in this study escalates to NP-hard status in the scenario of two stages scheduling problem (Yang & Chern, 1995). With an increase in problem size, the number of decision variables also rises accordingly. The subsequent subchapter demonstrates that the PS method offers additional reduction in variables by using the domain expertise of a queue-time loop scheduling problem.

Algorithm 2.1 Modified CBC algorithm for a queue-time loop scheduling problem

```
Initialization
    1. Let $\varepsilon = 0.0005$ and $UB$ is set to be an infinite number.
    2. Include equation 2.14 to the slave continuous problem.
    3. Go to the Primary Step.
Primary Step
    1. Resolve the integer problem.
       if      infeasible integer problem occurs then
                   the best solution is $UB$.
                   Algorithm stops.
       else    Let $Y^*$ be a feasible solution to the integer problem.
                   Go to Step 2.
       end if
    2. Apply $Y^*$ to the continuous problem. Resolve the continuous problem.
       if      infeasible continuous problem occurs then
                   Construct 2.15 and include it to the integer problem.
                   Go back to Step 1.
       else    optimal objective value $Z^*$ of the continuous problem is obtained.
                   $UB = Z^*$.
                   Construct 2.15 and include it to the integer problem.
                   Go back to Step 1.
       end if
```

2.3.2 Phase-step method

In the proposed MILP model, scheduling all jobs at each stage of the queue-time loop production system is required. As the number of jobs and stages increases, the complexity of the problem also grows due to a substantial rise in variables. As a result, the CBC algorithm might find it challenging to deliver a solution in a reasonable amount of time. Our production control strategy operates in a dynamic environment, continuously updating the statuses of job arrivals, machine failures, and recoveries. It reschedules production on an ongoing basis at each decision point. The CBC then solves the updated model using the latest status information, as illustrated by the dashed arrows in Fig. 2.3.

Jobs are scheduled sequentially from the initial stage to the final stage. When applying the production control strategy at the current decision point, the previous production schedule becomes obsolete as it is updated according to the newly recalculated results.

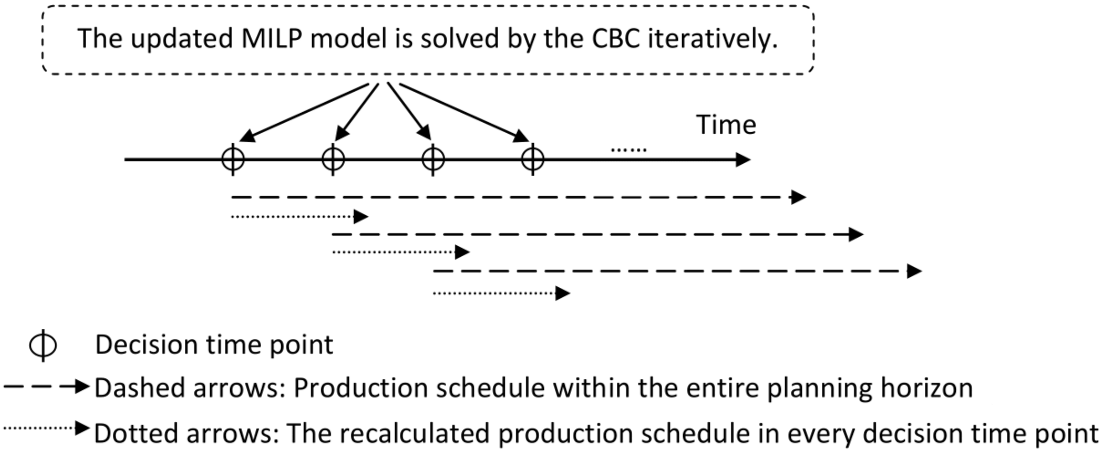

The updated MILP model is solved by the CBC iteratively.
Time
......
Decision time point
Dashed arrows: Production schedule within the entire planning horizon
Dotted arrows: The recalculated production schedule in every decision time point

Fig. 2.3. Solving scheme

Next, we present the PS method, which diverges from the approach of scheduling all jobs through to the final stage at every decision time point. We use the index q to denote the decision variables at each decision point. To reduce the number of decision variables, we discard production schedules that are significantly distant and concentrate on the streamlined model.

The PS method is based on three key definitions:

Definition 1: The mean processing time for stage f is obtained by dividing the processing time of stage f by the total number of machines available at that stage.

Definition 2: Stage f is classified as a phase-step if its mean processing time is greater than that of stage $f + 1$ and if the number of jobs in queue q exceeds the number of available machines at stage f.

doi:10.6342/NTU202403713

Definition 3: The first n jobs in stage f are referred to as necessary jobs if there are exactly n available machines at stage f, while all other jobs are labeled as non-necessary jobs.

The following examples demonstrate the defined concepts. If the mean processing time at stage 2 is greater than at stage 3, and the number of jobs at stage 2 exceeds the available machines, then stage 2 is classified as a phase-step, as shown in Fig. 2.4. Similarly, if stage 4 meets the criteria outlined in Definitions 1 and 2, it is also designated as a phase-step. Stage 2 is considered the current phase-step, while stage 4 is recognized as the next phase-step. For instance, if there are five jobs and only three machines available at stage 2, the first three jobs are classified as necessary, and the remaining two as non-necessary.

According to the PS method, jobs not at a phase-step are scheduled up to the next phase-step. Non-necessary jobs at a phase-step are scheduled up to the current phase-step, whereas necessary jobs are scheduled up to the subsequent phase-step. As illustrated in Fig. 2.4, all jobs at stages 1 and 2, excluding the necessary ones at stage 2, are scheduled up to stage 2, while the necessary jobs at stage 2 are scheduled until stage 4. Even with the reduction in decision variables and the size of the MILP model, the following proposition asserts that under the PS method, the production schedule for the smaller-scale problem will result in the same outcome as the full planning horizon.

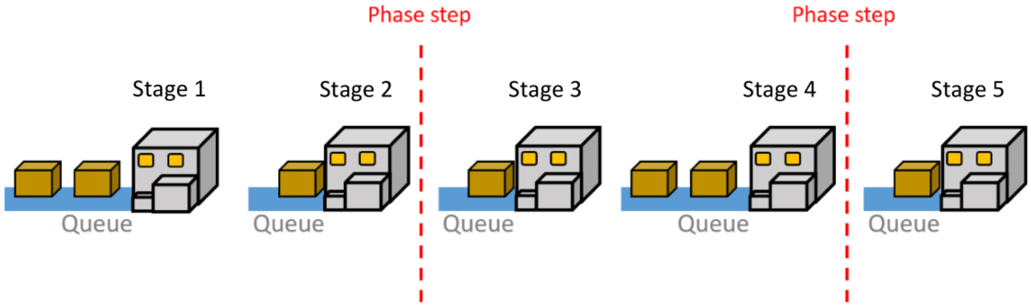


Fig. 2.4. Phase-step strategy

doi:10.6342/NTU202403713

Lemma 1. If there is only one machine available at each stage,

1. *the production schedule for essential jobs derived using the PS method at each decision point remains consistent with the production schedule for the entire planning horizon, and*
2. *the production schedule for non-essential jobs obtained from the PS method at each decision point is also consistent with the production schedule for the entire planning horizon.*

Proof.

Consider stage f as the second-last phase and stage g as the final phase. With only one available machine, the average processing time for stages f and g is equivalent to their respective processing times. Let p^f represent the processing time at stage f. At the previous decision point, we have the production schedule for non-essential jobs up to stage f and for essential jobs up to stage g. The completion times for non-essential and essential jobs at stages f and g, respectively, derived from the previous decision point's schedule, serve as the starting points for calculating the current production schedule. Let job i be an essential job and job $i+1$ a non-essential job. In the current production schedule, while job i needs to be scheduled from stage $g+1$ to the final stage, job $i+1$ needs to be scheduled from stage $f+1$ to stage g.

(a) To minimize the start time, job i in queue q is processed immediately at stage $g+1$ after completing stage g. Consequently, in the current production schedule, the start time of job i in queue q at stage $g+1$ is equal to the completion time of job i in queue q at stage g in the previous decision point's production schedule, as stated in (2.16).

$$s_{q,i}^{g+1} = c_{q,i}^{g} \tag{2.16}$$

doi:10.6342/NTU202403713

Clearly, since there is only one machine available at stage $g+1$, job i is assigned to this same machine. The queue time of job i at stage $g+1$ is calculated using (2.17).

$$s_{q,i}^{g+1} - c_{q,i}^{g} = k \cdot p^{g+1} + r \qquad k \in \mathbb{N}, r \in \mathbb{R}. \tag{2.17}$$

Here, k represents the number of jobs waiting to be processed at stage $g+1$, and r denotes the remaining processing time of the job currently on the machine. If the queue time of job i in queue q at stage $g+1$ is less than or equal to the queue time limit for stage $g+1$ as defined in (2.18), there is no violation of the queue time limit. Conversely, equation (2.19) indicates a violation.

$$s_{q,i}^{g+1} - c_{q,i}^{g} \leq qT^{g+1}. \tag{2.18}$$

$$s_{q,i}^{g+1} - c_{q,i}^{g} > qT^{g+1}. \tag{2.19}$$

Constraint (2.7) ensures a penalty is enforced on job i in queue q at stage $g+1$ if a queue time limit violation occurs, as outlined in (2.20).

$$s_{q,i}^{g+1} - c_{q,i}^{g} > qT^{g+1} + e_{q,i}^{g+1}. \tag{2.20}$$

If a penalty is enforced on job i in queue q at stage $g+1$, then job i is cancelled; otherwise, the start time of job i in queue q at stage $g+1$ is (2.21):

$$s_{q,i}^{g+1} = c_{q,i}^{g} + k \cdot p^{g+1} + r. \tag{2.21}$$

The finish time of job i in queue q at stage $g+1$ is (2.22):

$$c_{q,i}^{g+1} = s_{q,i}^{g+1} + p^{g+1}. \tag{2.22}$$

Job i enters stage $g+2$. The start time of job i at stage $g+2$ is equal to the finish time of job i at stage $g+1$ as (2.23):

$$s_{q,i}^{g+2} = c_{q,i}^{g+1}\ . \tag{2.23}$$

doi:10.6342/NTU202403713

The calculation of queue time of job i at stage $g+2$ is similar to the calculation of queue time of job i in stage $g+1$. If there is a violation of queue time limit at stage $g+2$, then a penalty is imposed on job i at stage $g+2$. Therefore, use the (2.16) – (2.23) to calculate the production schedule of job i from stage $g+1$ until the final stage to obtain the optimal production schedule. The starting time, queue time violation, and completion time of the necessary job calculated from stage $g+1$ until the final stage are identical to the starting time, queue time violation, and completion time calculated from the first stage until the final stage. Therefore, the current production schedule of necessary jobs obtained by the PS method is identical to the corresponding optimal production schedule of the entire planning horizon.

(b) The scheduling strategy of job $i+1$ from stage $f+1$ to stage g is similar to scheduling job i from stage $g+1$ to the final stage. Therefore, use (2.16) – (2.23) to calculate the starting time, queue time violation, and completion time of job $i+1$ from stage $f+1$ until stage g to obtain the optimal production schedule of non-necessary jobs. The starting time, queue time violation, and completion time of the non-necessary jobs calculated from stage $f+1$ until stage g are identical to the starting time, queue time violation, and completion time calculated from the first stage until the final stage. Therefore, the current production schedule of non-necessary jobs obtained by the PS method is identical to the corresponding optimal production schedule of the entire planning horizon.

Lemma 2: If each stage has multiple machines,

1. *the production schedule for necessary jobs derived using the PS method at each decision point remains the same as the corresponding production schedule for the entire planning horizon, and*

2. *the production schedule for non-necessary jobs obtained using the PS method at each decision point remains the same as the corresponding production schedule for the entire planning horizon.*

Proof.

(a) For the multiple machines case, the number of necessary jobs is the same as the number of available machines in each stage. Without loss of generality, assume there are m available machines at stage $g+1$. It is clear that there are m necessary jobs at stage $g+1$. Since m available machines are at stage $g+1$, then the mean processing time of stage $g+1$ is the processing time of stage $g+1$ divided by m existing machines in stage $g+1$. Let p_j^{g+1} denote the processing time of machine j at stage $g+1$. To minimize the start time, allocate m necessary jobs in parallel to the m existing machines to process, i.e. only one job is allocated to a machine in each stage. Thus, use (2.16) – (2.23) to compute the start time, the breach of queue time, and the finish time of the necessary jobs from stage $g+1$ to the last stage. Applying this method until the final stage optimizes the production schedule of the necessary jobs. This method generates an identical starting time, queue time violation, and completion time computed from stage $g+1$ to the final stage to the start time, breach of queue time, and finish time computed from the beginning stage to the last stage. Thus, the existing schedule of necessary jobs gained by the PS method is matching to the corresponding schedule of the whole planning horizon.

(b) For the case with m available machines in a stage, jobs $m+1, m+2, \ldots$ are non-necessary jobs. To minimize the starting time, assign these non-necessary jobs in parallel to the m available machines for processing, i.e. only one job is assigned to one machine in each stage. Therefore, use (2.16) – (2.23) to calculate the starting

doi:10.6342/NTU202403713

time, queue time violation, and completion time of those non-necessary jobs from stage $f+1$ until stage g to obtain the optimal production schedule. When the starting time, queue time violation, and completion time of the non-necessary jobs calculated from stage $f+1$ until stage g are identical to the starting time, queue time violation, and completion time calculated from the first stage until the final stage, then the current production schedule of non-necessary jobs obtained by the PS method is identical to the corresponding optimal production schedule of the entire planning horizon.

Proposition*:* The production schedule generated by the PS method is the same as the corresponding schedule for the whole planning horizon.

Proof.

According to Lemma 2, the optimal schedule for necessary and non-necessary jobs derived using the PS method is the same as the corresponding optimal schedule for the whole planning horizon. In a queue-time loop scheduling problem, all jobs are categorized as either necessary or non-necessary. Hence, the optimal schedule generated by the phase-step method matches the corresponding optimal schedule for the whole planning horizon.

A sufficient volume of jobs within a stage is crucial for the PS method to exclude non-necessary schedules for distant future stages within a rolling-based production schedule. The PS method is particularly effective when most jobs across stages are non-necessary, as it allows the elimination of a significant portion of variables related to future stages. Similar to the lookahead policy, the PS method explicitly considers specified future stages during schedule regeneration at each decision point (Romer & Mellouli, 2016).

By integrating the CBC with the PS method, a schedule for a queue-time loop scheduling problem can be created within a affordable computational time. While the CBC quickly resolves the model, the PS method provides the CBC with necessary information about future stages needed for scheduling jobs at each decision point.

2.4 Numerical study

Our hypothetical scenario mirrors the scale of a practical queue-time loop encountered in semiconductor manufacturing. We investigate a 4.6-month operation period (equivalent to 200,000 minutes) segmented into ten thousand decision time points. At each of it, we execute the cyclic process depicted in Fig. 2.2. Updates to job arrival statuses, machine availabilities, and production log file rescheduling occur every 20 minutes.

The queue-time loop system has six stages. There are two identical machines in stages 1, 2, 3, and 6 and there are three identical machines in stages 4 and 5. The processing times for stages 1, 2, 3, 4, 5, and 6 are 25, 45, 25, 105, 70, and 100 minutes, respectively. An infinite queue time limit is assumed for stage 1, as jobs preceding stage 1 can be buffered before entering the queue-time loop process. Queue time limits for stages 2 through 5 are set at 250 minutes, while stage 6 has a queue time limit of 350 minutes.

To demonstrate CBC's efficiency, the number of jobs varies in the system. We run the program for three minutes and compare the "feasible solution quality" gained by CBC and CPLEX. The results are shown in Table 2.2.

The same objective function values occur for both CBC and CPLEX for 10, 20, and jobs. However, for more than 30 jobs, CBC demonstrates superior performance over CPLEX.

doi:10.6342/NTU202403713

Subsequently, we examine ten thousand decision time points to assess the CBC-PS performance. We specifically analyze the impact of two parameters, job arrivals and queue time limits on throughput and the number of scrap jobs. The mean time between failures (MTBF) for stages 1, 2, 3, 4, 5, and 6 is set to 1100, 1200, 1100, 1400, 1200, and 1200 minutes, respectively. The mean time to repair (MTTR) for stages 1, 2, 3, 4, 5, and 6 is set to 25, 280, 25, 80, 115, and 85 minutes, respectively. We set two levels of job arrivals and queue time limit parameters, low and high levels. Table 2.3 details the four scenarios: LL, LH, HL, and HH. For instance, LH denotes low job arrival levels and a high queue time limit. Employing the schedule generated by CBC-PS, throughput and the number of scrap jobs are compared with FIFO, threshold (Wu et al., 2012), and reaction chain (Lee et al., 2005).

The results are illustrated in Fig. 2.5, with bar and dots charts indicating the number of scrap jobs and throughput, respectively. Fig. 2.5 illustrates that CBC-PS demonstrates superior performance over threshold (TH), reaction chain (RC), and FIFO methods in terms of reducing scrap jobs for all scenarios and maintains nearly identical throughput levels. CBC-PS notably minimizes scrap job numbers, particularly when job arrivals are frequent in the queue-time loop production system.

doi:10.6342/NTU202403713

Table 2.2 Value of objective function

Jobs number	Value of objective function	
	CPLEX	CBC
10	9,405	9,405
20	35,340	35,340
30	89,840	89,840
32	120,140	119,685
36	140,300	140,260
40	4,437,095	4,436,735
44	194,745,705	194,743,315
48	488,227,365	487,725,365
52	1,548,881,390	1,548,628,040
60	3,318,800,300	3,315,546,045
64	5,279,976,120	5,278,967,370
68	6,941,735,055	6,848,665,843
72	8,682,708,890	8,500,797,450
76	10,765,309,115	10,752,704,075
80	12,999,178,745	12,972,160,360
84	14,458,646,190	14,445,546,255
88	18,008,471,485	17,854,838,180
92	20,845,929,550	20,615,983,065
96	23,303,638,414	22,454,392,460
100	27,579,289,375	26,535,369,685

Table 2.3. Settings of parameter

	High	Low
Number of jobs/days	30	25
Limit of queue time limit	Infinite, 250, 250, 250, 250, 350	Infinite, 200, 200, 200, 200, 280

These findings illustrate that CBC-PS reduces the violation of queue time, leading to a significant decrease in scrap job numbers. Moreover, the PS method effectively reduces model variables, enabling CBC-PS to generate efficient schedule within a affordable timeframe. In the CBC-PS method, the PS method provides the model with information about distant future stage variables that can be temporarily disregarded at the current stage while CBC solves the updated model based on the latest statuses of job arrivals and machine failures and recoveries. Without employing the PS method, CBC struggles to solve experiments within a reasonable timeframe, given the significant increase in variables as job numbers rise. Over a 4.6-month plant operation period, job estimates exceed 3,400.

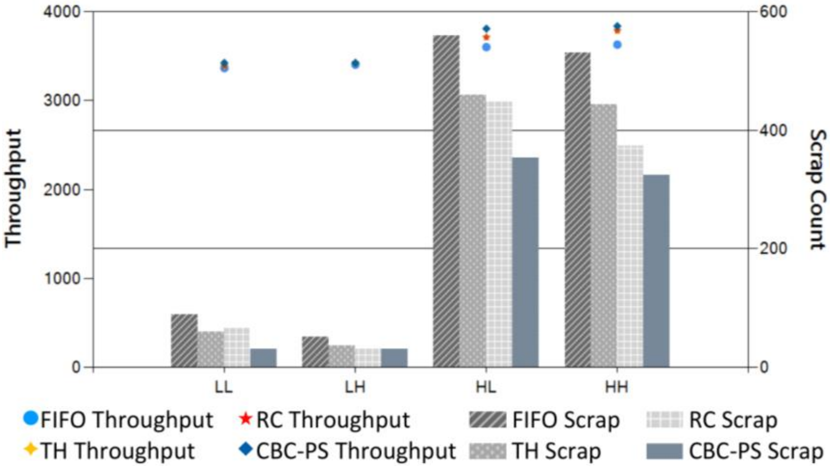


Fig. 2.5. The results of the base case

We further manipulate parameter settings by increasing MTBF, MTTR, or both by 1.5 times, with Figs. 2.6 – 2.8 displaying the results. Despite variations in parameter settings, outcomes remain consistent with the original MTBF and MTTR configuration. For high job arrival levels, CBC-PS consistently reduces scrap job numbers significantly compared to FIFO, TH, and RC.

doi:10.6342/NTU202403713

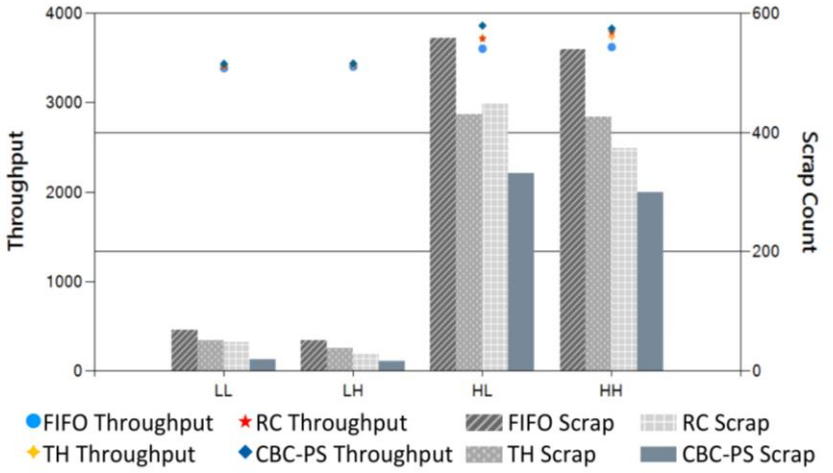


Fig. 2.6. Results if MTBF is increased 1.5 times

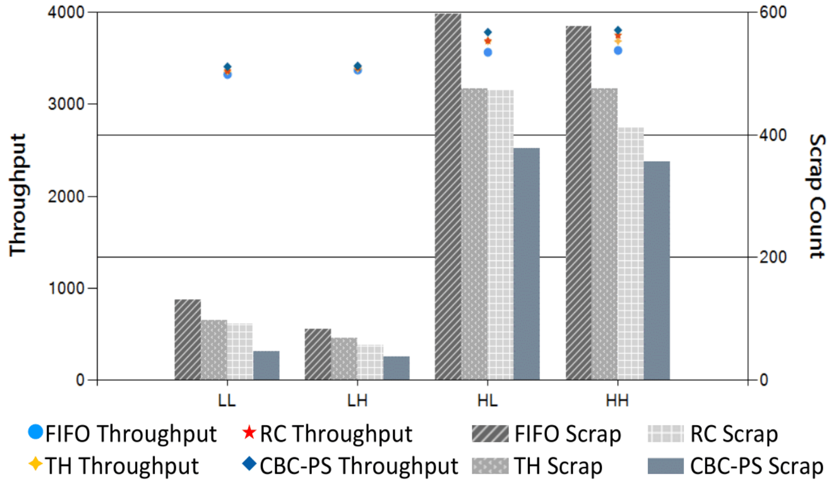


Fig. 2.7. Results if MTTR is increased 1.5 times

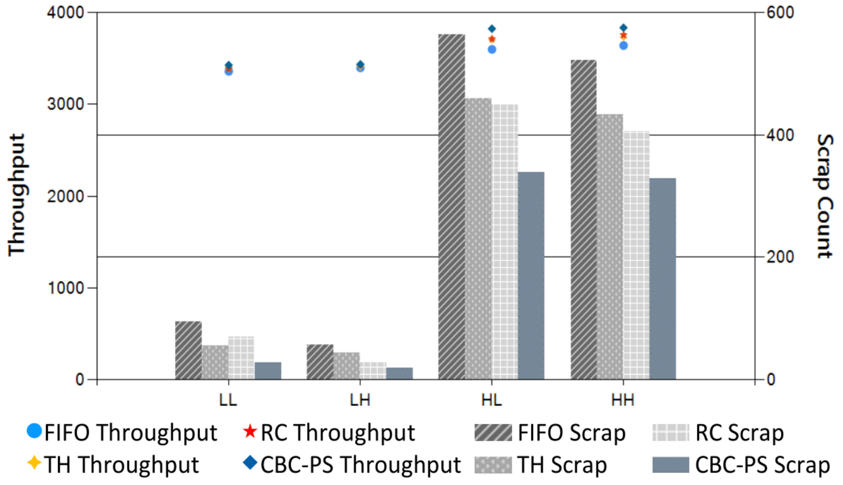


Fig. 2.8. Results if both MTBF and MTTR are increased 1.5 times

doi:10.6342/NTU202403713

2.5 Summaries

We devise a deterministic MILP-based approach to address the uncertainty in the problem, continuously updating job arrivals and machine statuses. At each decision point, we regenerate the production log file using these updates. Our method involves using CBC to tackle the MILP model, breaking it down into an integer problem and a continuous variable. To streamline the process and reduce computation time, we introduce the PS method, which simplifies the MILP model and decreases the number of variables. This combination results in even fewer variables. Comparing CBC to CPLEX solver, we find that CBC achieves superior objective function values within three minutes for large-scale problems with over 30 jobs. Furthermore, comparing CBC-PS to commonly used FIFO, TH, and RC heuristics demonstrates CBC-PS's superiority in minimizing scrap jobs, particularly under high job arrival rates.

doi:10.6342/NTU202403713

Chapter 3 Calibrating the turbulence characteristics using a surrogate model-based framework: A marine turbine simulation[2]

Harnessing sustainable energy from marine currents using marine turbines has become increasingly prominent in recent years. Evaluating the effectiveness of arrays of marine turbines necessitates extensive and computationally intensive simulations covering large domains. This study introduces a framework that utilizes surrogate models in conjunction with optimization algorithms to fine-tune the simulator's adjustable parameters, thereby minimizing the disparity between simulation outputs and real-world experimental data. We observe a 16.97% enhancement in performance compared to previous methods by employing more sophisticated surrogate models and optimization algorithms. Additionally, we identify a straightforward opportunity for further performance enhancement. Utilizing descriptive statistics, we use a visual tool for quickly and easily assessing the quality of sample data.

3.1 Problem background

The utilization of turbines to harness energy from marine currents has attracted significant attention, driven by the global focus on generating renewable energy (Ren et al., 2018; Brutto et al., 2016). However, concerns regarding turbine longevity and costly installation

[2] This chapter is rewritten based on an article authored by this thesis writer titled “A surrogate model-based framework to calibrate the turbulence parameters of a vegetative canopy model for a marine turbine simulation" which is published in Journal of Ocean Engineering and Marine Energy in 2023 (Nurdiansyah et al., 2023).

underscore the need for thorough investigation into their hydrodynamic performance prior to deployment (Nag and Sarkar, 2020; Bryden et al., 2007).

The wake properties for various turbine arrangements are explored by Nuernberg and Tao (2018), where they observe that the distance between turbines affects how quickly the air flow returns to normal after passing through them. Proximity between turbines leads to a notable reduction in current velocity recovery, whereas wider spacing facilitates better velocity recovery. Gaurier et al. (2020) conduct an experimental study to examine the impact of turbulence intensity on wakes, configuring two upstream turbines and one downstream turbine. Compared to upstream turbines, downstream turbines exhibit reduced velocity and amplified turbulence levels.

Mycek et al. (2014) investigate hydrodynamic performance properties of turbines in a flume. Their research employs an 18 m-long, 4 m-wide, and 2 m-deep flume tank, along with a marine turbine featuring a three-blade with a radius of 0.35 m. The turbine is exposed to a flow with an average velocity ($\boldsymbol{U}$) of 0.8 m/s and initial turbulence intensities ($\boldsymbol{I}$) of 3% and 15%. Reduced turbulence leads to smoother and more predictable water flow. Based on this, hydrodynamic performance, measured by current velocity and turbulence intensity, is captured in four sets of data points. These data are collected at ten points downstream of the turbine and organized into a 4×10 matrix. The matrix includes velocity and turbulence data for both low (3%) and high (15%) turbulence conditions. Figure 3.1 visually represents the test environment with data points collected at distances ranging from 1.2 to 10 times the turbine diameter.

In practical, conducting physical experiments like this one and similar ones is costly, inefficient, time-consuming, and environmentally harmful. Moreover, these experiments tend to be repetitive, as each new experiment necessitates conducting a prototype (Milne et al., 2013). A simulation model has been proposed, such as by Olson

doi:10.6342/NTU202403713

et al. (2021), Brunetti et al. (2019) and Dosi et al. (2021) to highlight its efficacy and suitability as reduced models to robustly tackle complex systems. This research introduces an surrogate model approach coupled with optimization techniques to imitate the hydrodynamic characteristics of a marine turbine.

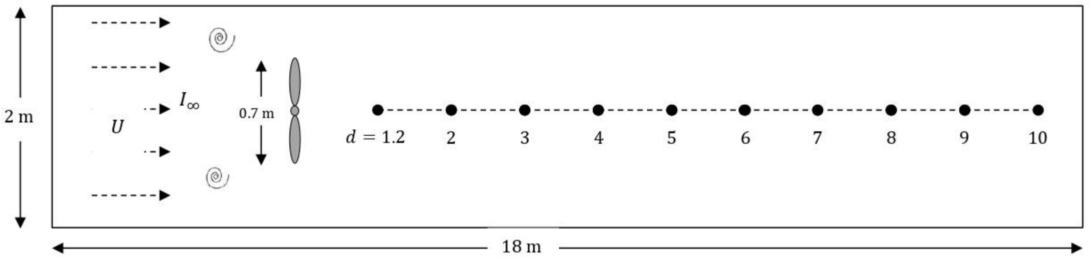


Fig. 3.1 Flume tank side view (adapted from Mycek et al., 2014)

OpenFOAM (Open-source field operation and manipulation), introduced by OpenCFD Ltd., is a widely used computer program in both academic and industry for computational fluid dynamics (CFD). Considering marine turbine arrays can span wide area, the actuator disc method in OpenFOAM is used since our simulations cover domains of this scale. Although other methods like the blade element method and actuator line models exist, they offer little additional resolution due to the lack of publicly available experimental and field data for marine turbines, making the additional computational cost unjustified. The actuator disc method in the OpenFOAM model approximates turbines and describes the turbulence of actuator disc based on an approach conducted to simulate currents through vegetative canopies. Réthoré et al. (2009) and James et al. (2020) have been applied the model to wind turbines and marine turbines.

Two hydrodynamic properties of downstream turbine, i.e., current velocity and turbulence intensity have been simulated by Olson et al. (2021) utilizing OpenFOAM using data from Mycek et al. (2014). With four outputs at ten distinct locations and two

doi:10.6342/NTU202403713

initial conditions, we generate a graph to display the performance of the reference model without any modifications, as depicted in Fig. 3.2. Fig. 3.2 highlights a substantial disparity between the reference model without any modifications and the physical experiment vectors. On average, the reference model without any modifications presents a 28.20% MAPE (mean average percentage error) compared to the physical experiment vectors. This model adjusts four continuous parameters ($\boldsymbol{\beta_p}, \boldsymbol{\beta_d}, \boldsymbol{C_{\varepsilon 4}}$ and $\boldsymbol{C_{\varepsilon 5}}$) to zero. While some researchers suggest values for these parameters (e.g., James et al., 2020;; Réthoré et al., 2009; Katul et al., 2004), the optimal settings remain undetermined. In this study, the adjustable parameters in the vegetative canopy model, i.e., $\boldsymbol{\beta_p}, \boldsymbol{\beta_d}, \boldsymbol{C_{\varepsilon 4}}$ and $\boldsymbol{C_{\varepsilon 5}}$, are calibrated to find their optimal value by minimizing MAPE.

However, there are limitations to using OpenFOAM. For instance, Olson et al. (2021) run a single setup of the OpenFOAM simulation model on the Sandia National Laboratories supercomputer for an hour. Additionally, due to the parameters are continuous, the number of iterations required to gain the optimal values of those parameters can be infinite. Consequently, extended simulation durations greatly limit the number of iterations that can be performed to find the optimal values.

To address these limitations, our goal is to minimize the Mean Absolute Percentage Error (MAPE) between the vectors generated by the OpenFOAM simulator and those obtained from physical experiments. We improve upon the approach of Olson et al. (2021) in three significant ways:

1. Surrogate Model Enhancement: We use an Artificial Neural Network (ANN) instead of the Gaussian model to construct the surrogate model, aiming to boost its performance. Numerous studies have shown the effectiveness of ANN as a meta-model for Computational Fluid Dynamics (CFD) simulations, especially for modeling complex, non-linear, and computationally intensive systems. For instance, Elsayed and

doi:10.6342/NTU202403713

Lacor (2013) employed ANN to analyze the performance of a gas cyclone separator, examining the impact of parameters like inlet height, inlet width, vortex finder diameter, and cyclone total height. Similarly, Gholami et al. (2015) used ANN to simulate flow characteristics in sharp bends within an open channel, while Lira et al. (2022) applied ANN for the analysis and optimization of micro-photocatalytic reactors. Despite ongoing efforts to refine the mathematical characterization of approximation properties for data-driven techniques utilizing ANN, as highlighted by Pichi et al. (2023), there are still limitations to these outcomes.

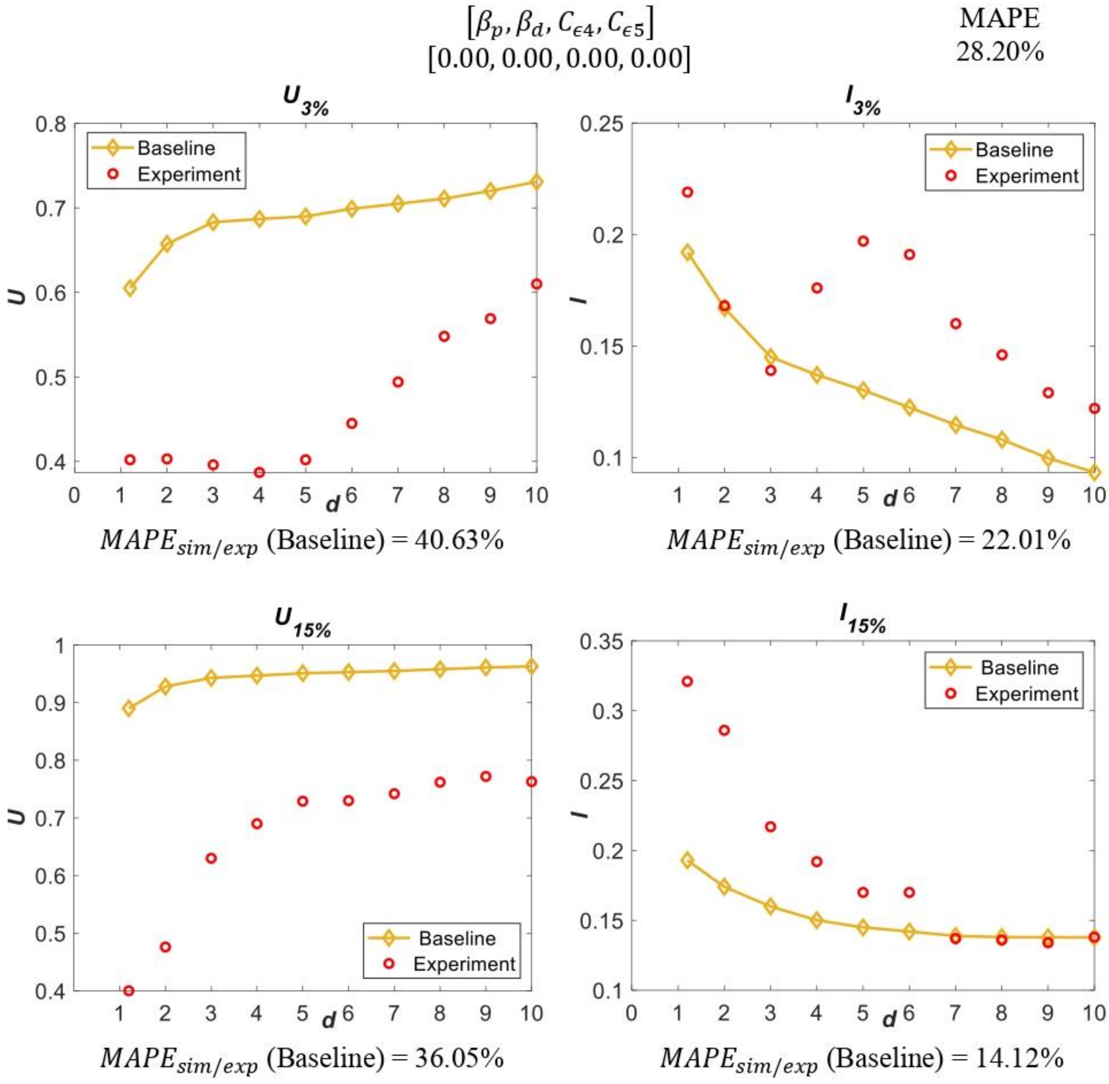


Fig. 3.2 The performance of the reference model without any modifications (adapted from Olson et al., 2021)

2. Advanced Optimization Algorithms: We implement more sophisticated optimization algorithms, including Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), to optimize the values of the adjustable parameters.
3. Further Parameter Calibration: We explore the potential for additional parameter calibration to achieve further improvements.

The discrepancy observed between the vectors derived from physical experiments and simulation allows us to exclude subsets of simulations during parameter calibration. As a result, we eliminate the need for running numerous costly and time-consuming simulations, leading to a performance enhancement of 16.97% compared to Olson et al. (2021).

Next, in subchapter 3.2, we outline the development of the OpenFOAM model, the framework of the surrogate model, and three error metrics utilized to assess performance of the surrogate model. Subchapter 3.3 delves into the surrogate model, while subchapter 3.4 elaborates on the PSO and GA. Subchapter 3.5 covers the calibration of adjustable parameters, followed by subchapter 3.6, which details a visual tool designed to aid researchers in swiftly evaluating the sample data's quality. Finally, subchapter 3.7 provides a conclusion to the study.

3.2 Framework and development

3.2.1 OpenFOAM model development

The Reynolds-averaged Navier–Stokes two-phase volume of fluid method alongside the $k - \varepsilon$ equations turbulence model is applied in the InterFoam solver applies to define the flow pattern. The openFOAM model development in this study refers to Olson et al. (2021) and Nurdiansyah et al. (2023). The resultant grids from the OpenFOAM settings representing the IFREMER flume validate the adequacy of using a surrogate model instead of the OpenFOAM model (Olson 2019).

3.2.2 Model of vegetative canopy

Vegetative canopy model integrates two essential parameters: k, denoting turbulent kinetic energy, and ε, representing the dissipation rate of that energy. These variables are defined by (3.1) and (3.2) respectively.

$$S_k = \frac{1}{2} C_x \left(\beta_p U^3 - \beta_d U k\right), \tag{3.1}$$

$$S_\varepsilon = \frac{1}{2} C_x \left(\beta_p C_{\varepsilon 4} \frac{\varepsilon}{k} U^3 - \beta_d C_{\varepsilon 5} U \varepsilon\right), \tag{3.2}$$

where $C_x(-)$ denotes the changed thrust coefficient (C_t) and U (m/s) describes the velocity of the flow at the actuator disc. Parameter β_p determines the portion of k transformed into kinetic energy of wake-generated due to drag, thereby characterizing the ratio of mean kinetic energy directly converted into turbulence. Equations (3.1) and (3.2) trust on U. However, turbine occurrence alters the local current field, necessitating updates to the exact thrust coefficient of the disc, $C_x(-)$, as (3.3) and (3.4) (Stallard et al., 2013).

$$C_d = 2 \frac{1 - \sqrt{1 - C_t}}{1 + \sqrt{1 - C_t}}, \tag{3.3}$$

$$C_x = \rho(A_d C_d), \tag{3.4}$$

where A_d (m^2) describes the flow-facing area of the actuator disc and ρ denotes the density of water.

For an actuator disc model, accurate thrust coefficient (C_t) and power coefficient (C_p) curves derived from experimental data or manufacturer specifications are crucial. Adjustment of the model to match the C_t/C_p of the turbine is necessary if they differ.

doi:10.6342/NTU202403713

While actuator disc size may be inconsequential for deeply submerged turbines, it can impact C_t/C_p curves. Unfortunately, this study lacks experimental data for various settings, highlighting a limitation. Access to such data would significantly enhance the method's validation. By employing the $k-\varepsilon$ turbulence equation and these variables, the model effectively simulates turbulent flow behavior within vegetative canopies, offering utility across applications such as marine hydrokinetic turbine design optimization.

3.2.3 Framework of surrogate model and its performance

The surrogate model framework is illustrated in Fig. 3.3. Initially, investigations are devised by changing values of adjustable parameters. Utilizing training and validation data gathered from the OpenFOAM, the surrogate model is trained. The fitting error is employed to assess the accurateness of the surrogate model's predictions, representing the comparison between the surrogate model result and the validation data using MAPE, as defined in (3.5).

$$MAPE_{srgt/sim} =$$

$$\frac{1}{N_d}\frac{1}{N_p}\sum_{m=1}^{N_d}\sum_{j}\sum_{e}\sum_{d}\left|\frac{Q_{j,e,d}^{S}-Q_{j,e,d}^{Sr}}{Q_{j,e,d}^{S}}\right|\cdot 100\%, \qquad (3.5)$$

$$j \in \{U, I\};\ \forall e \in E;\ \forall d \in D,$$

Where $Q_{j,e,d}^{Sr}$ and $Q_{j,e,d}^{S}$ denotes under initial turbulence intensity e at locations d, the surrogate model’s prediction and state variables j of the simulator’s, respectively; E as the initial turbulence intensity set; D as the downstream turbine locations set; N_d as the number of simulation datasets; and N_p as number of vectors multiplied by the number of d.

doi:10.6342/NTU202403713

If $MAPE_{srgt/sim}$ equals or exceeds the tolerable ratio, additional training data is required. Conversely, if $MAPE_{srgt/sim}$ falls below the tolerable ratio, an optimization algorithm is employed to minimize the predicted error, error between the surrogate model vectors and physical experiment vectors using MAPE as described in (3.6).

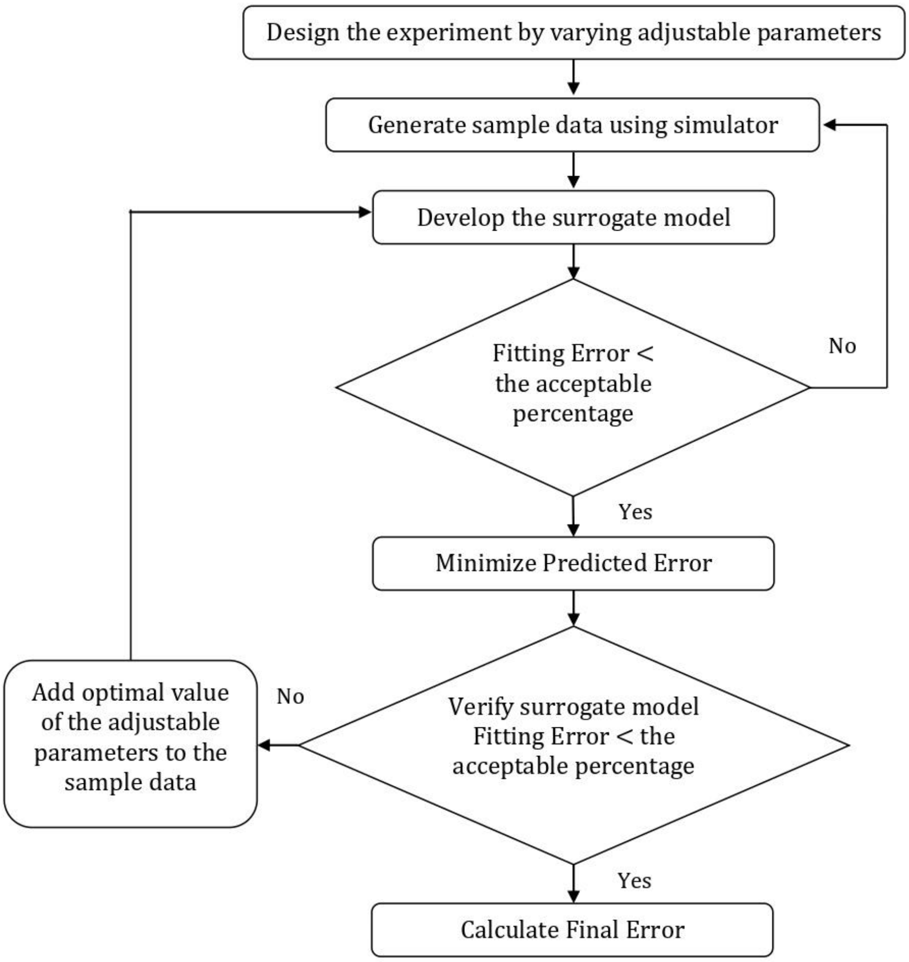

Design the experiment by varying adjustable parameters
Generate sample data using simulator
Develop the surrogate model
Fitting Error < the acceptable percentage
No
Yes
Minimize Predicted Error
Verify surrogate model Fitting Error < the acceptable percentage
No
Add optimal value of the adjustable parameters to the sample data
Yes
Calculate Final Error

Fig. 3.3. The framework (adapted from Olson et al., 2021).

$$MAPE_{srgt/exp} =$$

$$\frac{1}{N_p}\sum_j \sum_e \sum_d \left| \frac{Q^P_{j,e,d} - Q^{Sr}_{j,e,d}}{Q^P_{j,e,d}} \right| \cdot 100\%, \tag{3.6}$$

$$j \in \{U, I\};\ \forall e \in E;\ \forall d \in D.$$

doi:10.6342/NTU202403713

where under initial turbulence intensity e at locations d, $Q^{P}_{j,e,d}$ denotes the state variables j of the physical experiment.

Upon finding the optimal value, we verify that the fitting error remains below the tolerable ratio. If not, the optimal value and its vectors are incorporated into the training data and the surrogate model is retrained. The iterative strategy continues until $MAPE_{srgt/sim}$ is less than the tolerable ratio. Finally, the final error is computed, error between the calibrated simulator yield vectors and physical experiment data, using MAPE as defined in (3.7).

$$MAPE_{sim/exp} =$$

$$\frac{1}{N_p}\sum_{j}\sum_{e}\sum_{d}\left|\frac{Q^{P}_{j,e,d} - Q^{S}_{j,e,d}}{Q^{P}_{j,e,d}}\right| \cdot 100\%, \tag{3.7}$$

$$j \in \{U, I\};\ \forall e \in E;\ \forall d \in D.$$

3.3 Development of surrogate model

Artificial Neural Network (ANN), extensively employed for tasks like approximation or fitting, demonstrates characteristics such as self-learning, rapid training speed, adaptability, and resilience (Abdullah et al., 2022; Liu et al., 2018; Ding et al., 2011). The ANN imitates the functioning of the human brain through computer models comprising interconnected nerve cells treating information of the model (Papadopoulos et al., 2018). The ANN realizes input data delivered to the system to produce a yield. In the context of this research, input comprises vectors from the simulant, with several adjusts of parameter values serving as sample data, and yield consists of prediction vectors for ceertain sets of parameters.

doi:10.6342/NTU202403713

During the phase of training, ANN initially appoints random weights for individual input value. Backward propagation, employed as the learning rule in this study, iteratively adjusts the error between sample data and prediction vectors obtained by ANN. Partial differentiation, as outlined in (3.8), is performed during backward propagation, with the fitting error (3.5) serving as the loss function.

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial g}\frac{\partial g}{\partial h}\frac{\partial h}{\partial w}, \tag{3.8}$$

where $\boldsymbol{g}$ is the activation function, $\boldsymbol{h}$ is the weighted sum function, and $\boldsymbol{L}$ is the loss function, and $\boldsymbol{w}$ is the weight. The rectified linear unit (ReLu) is utilized as the activation function, as depicted in (3.9). We update the weight $\boldsymbol{w}$ using (3.10), where $\boldsymbol{w_{new}}$ denotes the recent weight, $\boldsymbol{w_{old}}$ represents the prior weight, and $\boldsymbol{\alpha}$ denotes the discovering degree. The phase of training concludes upon training entire sample data and finishing an period or epoch. We employ K-fold cross-validation is employed for constructing the model.

$$g(y) = \begin{cases} 0 \text{ for } y \leq 0 \\ y \text{ for } y > 0 \end{cases}. \tag{3.9}$$

$$w_{new} = w_{old} - \alpha\frac{\partial L}{\partial w}. \tag{3.10}$$

3.4 Optimization methods

The PSO algorithm and GA are utilized in this study to minimalize the $MAPE_{srgt/exp}$, aiming to align the surrogate model vectors similar to the physical experiments.

doi:10.6342/NTU202403713

3.4.1 PSO algorithm

The PSO algorithm is motivated by the collective manner of swarm, introduced by Kennedy and Eberhart (1995). In the PSO algorithm, particles or swarms are considered by their velocity, position, and the objective function value of their position. The simplicity and adaptability of the PSO algorithm make it well-suited for addressing complex optimization difficulties (Wang et al., 2021), with efficacy of PSO is confirmed by Sedighizadeh et al. (2021).

In the context of this research, a set of changeable parameters is represented by each particle. The positions of particle i in period t and $t-1$ are denoted as x_i^t and x_i^{t-1}, respectively. The speeds of particle i in period t and $t-1$ are denoted as v_i^t and v_i^{t-1}, respectively. The best solution obtained by particle i and the best solution obtained among all particles are denoted as l_i and g. $MAPE_{srgt/exp}$ as (3.6) is minimized by the PSO algorithm. If the stopping condition is satisfied, the PSO algorithm terminates, yielding the minimized $MAPE_{srgt/exp}$. Otherwise, particle speeds v_i^t and positions x_i^t are updated using (3.11) and (3.12), and the evaluation process repeats. In (3.11) and (3.12), c_1 and c_2 represent positive constants known as the learning factors, while r_1 and r_2 denote random numbers ranging from 0 to 1. Figure 3.4 illustrates the steps of the PSO algorithm.

$$v_i^t = v_i^{t-1} + c_1 r_1 (l_i - x_i^{t-1}) + c_2 r_2 (g - x_i^{t-1}), \tag{3.11}$$

$$x_i^t = x_i^{t-1} + v_i^t, \tag{3.12}$$

doi:10.6342/NTU202403713

3.4.2 GA

The genetic algorithm (GA), introduced by Holland in 1992, is an evolutionary metaheuristic inspired by natural genetic processes and has proven effective in optimizing natural systems (Gentils et al., 2017). Known for its speed and efficiency in handling computationally intensive problems (Hamdia et al., 2021), the GA has been widely used for solving numerical problems and making predictions (Zuo et al., 2019; Połap, 2020). During optimization, the GA uses selection, crossover, and mutation mechanisms among chromosomes, each representing a feasible solution, to identify the optimal solution.

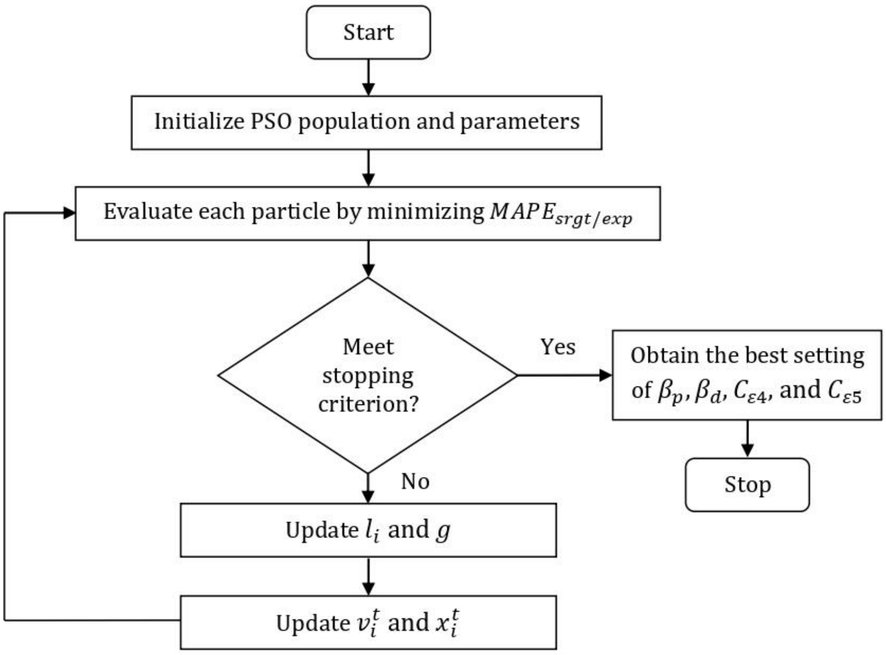

Start
Initialize PSO population and parameters
Evaluate each particle by minimizing $MAPE_{srgt/exp}$
Meet stopping criterion?
Yes
Obtain the best setting of β_p, β_d, $C_{\varepsilon 4}$, and $C_{\varepsilon 5}$
Stop
No
Update l_i and g
Update v_i^t and x_i^t

Fig. 3.4. The steps of the PSO algorithm.

In this study, we begin by generating chromosomes encoded as binary code and initializing the GA parameters. Each chromosome, composed of genes with a length of n, represents a feasible solution for the adjustable parameters. These chromosomes are substituted into $g(y)$ as shown in (3.9) within the surrogate model to generate prediction vectors. The GA then evaluates chromosomes using the predicted error (3.6). If the stopping criterion is met, the GA stops and retrieves the best solution for the adjustable

doi:10.6342/NTU202403713

parameters. Otherwise, it continues with selection, choosing chromosomes for reproduction, followed by crossover and mutation to increase chromosome diversity and prevent premature convergence. Crossover and mutation are conducted based on predetermined crossover and mutation rates. A one-point crossover operator is used for crossover, selecting a point in the two chosen chromosomes and exchanging the genes to the right of that point between them. Mutation uses a bit-flip mutation operator, flipping one gene in each of the selected chromosomes to another chromosome. After selection, crossover, and mutation, each chromosome is reassessed. Figure 3.5 illustrates the steps of the GA.

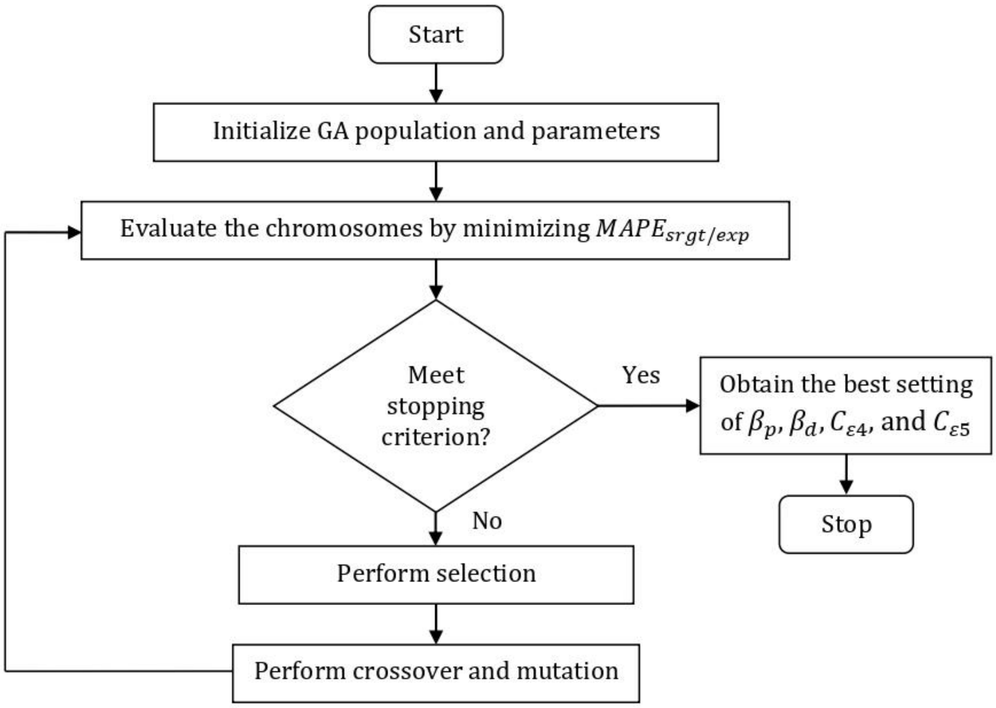

Start
Initialize GA population and parameters
Evaluate the chromosomes by minimizing $MAPE_{srgt/exp}$
Meet stopping criterion?
Yes
Obtain the best setting of β_p, β_d, $C_{\varepsilon4}$, and $C_{\varepsilon5}$
Stop
No
Perform selection
Perform crossover and mutation

Fig. 3.5. The steps of the GA.

doi:10.6342/NTU202403713

3.5 Results and discussion

ANN-PSO is used to denote when the results are from ANN paired with PSO and ANN-GA when the results are from ANN is paired with GA. For sampling the training data, $\beta_p, \beta_d, C_{\varepsilon 4}$, and $C_{\varepsilon 5}$ are set to have lower points at 0.1, 0.1, 0.1, and 0.1, respectively, while having upper points at 1, 4, 1, and 4, respectively. All parameters are divided into three points: lower point, upper point, and midpoint. The validation data has two levels, the average of first and second level, and the average of second and third level. This experiment yields 81 training data points and 16 validation data points. The parameters values are presented in Table 3.1. The sample data is generated by running the OpenFOAM at the Sandia National Lab.

Table 3.1. Settings of the adjustable parameters

Levels	β_p	β_d	$C_{\varepsilon 4}$	$C_{\varepsilon 5}$
For the training data				
Level 1	0.100	0.100	0.100	0.100
Level 2	0.550	2.050	0.550	2.050
Level 3	1.000	4.000	1.000	4.000
For the validation data				
Level 1	0.325	1.075	0.325	1.075
Level 2	0.775	3.025	0.775	3.025

To construct ANN, a fivefold cross-validation process is adopted. We set the crossover rate at 0.7, the number of chromosomes at 100, the binary code length at 10 for the GA parameters, and the mutation rate at 0.2. We set the number of particles at 50, and $c_1 = c_2 = 1$ for the PSO parameters. Both the PSO and the GA are set to run for 1000 iterations. We define the tolerable ratio in $MAPE_{srgt/sim}$ as 10%.

doi:10.6342/NTU202403713

Using the training data, we develop the ANN-PSO and calculate the initial $MAPE_{srgt/sim}$ to evaluate the surrogate model, finding it to be below 10%. We then determine the optimal values of the tolerable ratio to minimize the predicted error. As discussed in subchapter 3.2, we recalculated $MAPE_{srgt/sim}$ at the optimal values to verify the surrogate model's performance. However, the second $MAPE_{srgt/sim}$ exceeded 10%. Therefore, we augmented the training data by including the optimal values and their corresponding simulation outcomes. This time, the second $MAPE_{srgt/sim}$ was below 10%.

The ANN-PSO uses a total of 83 training data points (81 + 2). The initial and subsequent $MAPE_{srgt/sim}$ are 1.46% and 4.34%, respectively, indicating that the average error between the surrogate model predictions and the 16 validation data points is 1.46%. The error between the surrogate model prediction and the simulation outcome at the optimal setting is 4.34%, highlighting the surrogate model's effectiveness. The predicted error minimized by PSO is 13.81%. The optimal values of parameters β_p, β_d, $C_{\varepsilon 4}$, and $C_{\varepsilon 5}$ obtained by PSO are 0.91, 0.10, 1.00, and 1.03, respectively. Applying these values to the simulator, the final error compared to the physical experiment is 14.21%.

Following a similar procedure with the GA optimization algorithm, the ANN-GA requires only one update of training data, resulting in 82 training data points (81 + 1). The initial and subsequent fitting errors for ANN-GA are 1.10% and 5.50%, respectively. The minimized predicted error by GA is 15.70%, and the final error is 13.21%. The optimal values of parameters β_p, β_d, $C_{\varepsilon 4}$, and $C_{\varepsilon 5}$ obtained by GA are 0.98, 0.20, 0.73, and 0.87, respectively. Table 3.2 summarizes the results for both ANN-PSO and ANN-GA.

doi:10.6342/NTU202403713

Table 3.2. Results by ANN-PSO and ANN-GA

Surrogate model	Number of training data	1st Fitting Error	2nd Fitting Error	Predicted Error	Final Error
ANN-PSO	83	1.46%	4.34%	13.81%	14.21%
ANN-GA	82	1.10%	5.50%	15.70%	13.21%

Figure 3.6 illustrates the vectors of physical experiment alongside the yields of the reference vegetative canopy and the calibrated simulation results of both ANN-PSO and ANN-GA. In Figure 3.6, “Experiment,” “Baseline,” “Simulation ANN-PSO,” and “Simulation ANN-GA” correspond to the vectors of physical experiment, results of the reference model, and calibrated simulation results by ANN-PSO and ANN-GA, respectively.

We observe that our results, the calibrated values of tolerable ratio, outperforms the reference model (uncalibrated vegetative canopy). We decrease up to 13,21% obtained by ANN-GA. Figure 3.6 demonstrates that the simulation results of both ANN-GA and ANN-PSO closely equal to the physical experiment compared to the reference model.

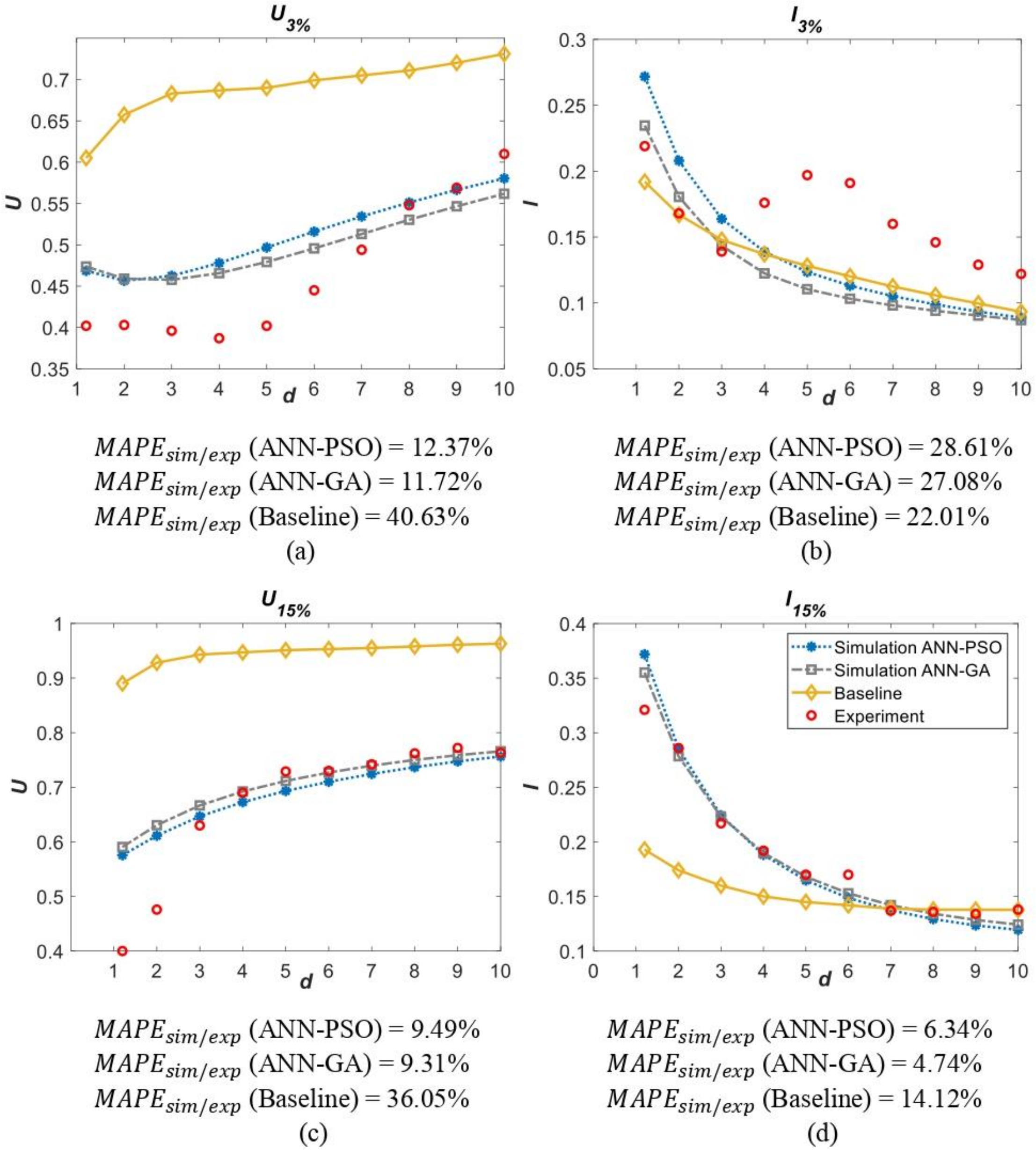
Further analysis of Figure 3.6 reveals that the results from calibration perform better as the turbulence intensity is 15% rather than 3%. $MAPE_{sim/exp}$ of $I_{15\%}$ and $U_{15\%}$ are 6.34% and 9.49% for ANN-PSO, respectively. They are lower than $MAPE_{sim/exp}$ of $I_{3\%}$ and $U_{3\%}$, i.e. 28.61% and 12.37%, respectively. Similarly, $MAPE_{sim/exp}$ of $I_{15\%}$ and $U_{15\%}$ are 4.74% and 9.31% for ANN-GA, respectively, compared to the final errors of $I_{3\%}$ and $U_{3\%}$, which are 27.08% and 11.72%, respectively.

In usual environments, a turbulence intensity of about 15% is expected. This value is notably higher than the unusually low value of 3%. Therefore, the adjusted simulator is anticipated to be beneficial for ecosystems with turbulence intensities of about 15%. A

doi:10.6342/NTU202403713

comparison of Figures 3.6I and 3.6(d) with 3.6(a) and 3.6(b) also suggests the need for additional calibration.

We suggest additional calibration is based on the remarks as follows. The pattern of physical observation displays performances that cannot be adequately captured by a second-order function. For instance, in Fig. 3.6(b), the red dot representing the physical experiment demonstrates a diminishing pattern up to d = 3, followed by an improving pattern until d = 5, and then another diminishing pattern.



$MAPE_{sim/exp}$ (ANN-PSO) = 12.37%
$MAPE_{sim/exp}$ (ANN-GA) = 11.72%
$MAPE_{sim/exp}$ (Baseline) = 40.63%
(a)

$MAPE_{sim/exp}$ (ANN-PSO) = 28.61%
$MAPE_{sim/exp}$ (ANN-GA) = 27.08%
$MAPE_{sim/exp}$ (Baseline) = 22.01%
(b)

$MAPE_{sim/exp}$ (ANN-PSO) = 9.49%
$MAPE_{sim/exp}$ (ANN-GA) = 9.31%
$MAPE_{sim/exp}$ (Baseline) = 36.05%
(c)

$MAPE_{sim/exp}$ (ANN-PSO) = 6.34%
$MAPE_{sim/exp}$ (ANN-GA) = 4.74%
$MAPE_{sim/exp}$ (Baseline) = 14.12%
(d)

Fig. 3.6. Simulation outputs

doi:10.6342/NTU202403713

Such complex behavior necessitates at least a third-order function for accurate description. Conversely, regardless of the calibration method applied, all simulation outputs from the simulator exhibit second-order functional patterns due to the simplification of detailed physical phenomena and inherent second-order character of the vegetative canopy system. Fig. 3.6(b) illustrates that a notable discrepancy exists between the vectors of physical observation and the simulation outputs from both ANN-GA and ANN-PSO for d = 4, 5, 6, with MAPEs of 27.08% and 28.61%, respectively. Similarly, in Fig. 3.6(a), the MAPEs of ANN-GA and ANN-PSO are 11.72% and 12.37%, respectively.

Further, this research employs a technique as same as that outlined in subchapter 3.5, with the only modification being the exclusion of errors associated with d = 4, 5, and 6 from calculating the value objective functions. The total number of training data points for ANN-PSO is 82. The initial and subsequent $MAPE_{srgt/sim}$ obtained are 1.22% and 3.50%, respectively. $MAPE_{srgt/exp}$ is 12.99%, and $MAPE_{sim/exp}$ is 12.53%. The values of parameters $\beta_p, \beta_d, C_{\varepsilon 4}$, and $C_{\varepsilon 5}$ gained by PSO are, respectively, 1.00, 0.10, 1.00, and 0.80.

The total number of training data points for ANN-GA is 83. The initial and subsequent $MAPE_{srgt/sim}$ obtained are 1.45% and 2.78%, respectively. $MAPE_{srgt/exp}$ is 11.81%, and $MAPE_{sim/exp}$ is 12.04%. The values of parameters $\beta_p, \beta_d, C_{\varepsilon 4}$, and $C_{\varepsilon 5}$ gained by GA are, respectively, 0.97, 0.15, 0.96, and 0.94. The results are summarized in Table 3.3.

doi:10.6342/NTU202403713

Table 3.3. Errors found by ANN-PSO and ANN-GA when excluding $d = 4, 5$, and 6

Surrogate model	Number of training data	1st Fitting Error	2nd Fitting Error	Predicted Error	Final Error
ANN-PSO	82	1.22%	3.50%	12.99%	12.53%
ANN-GA	83	1.45%	2.78%	11.81%	12.04%

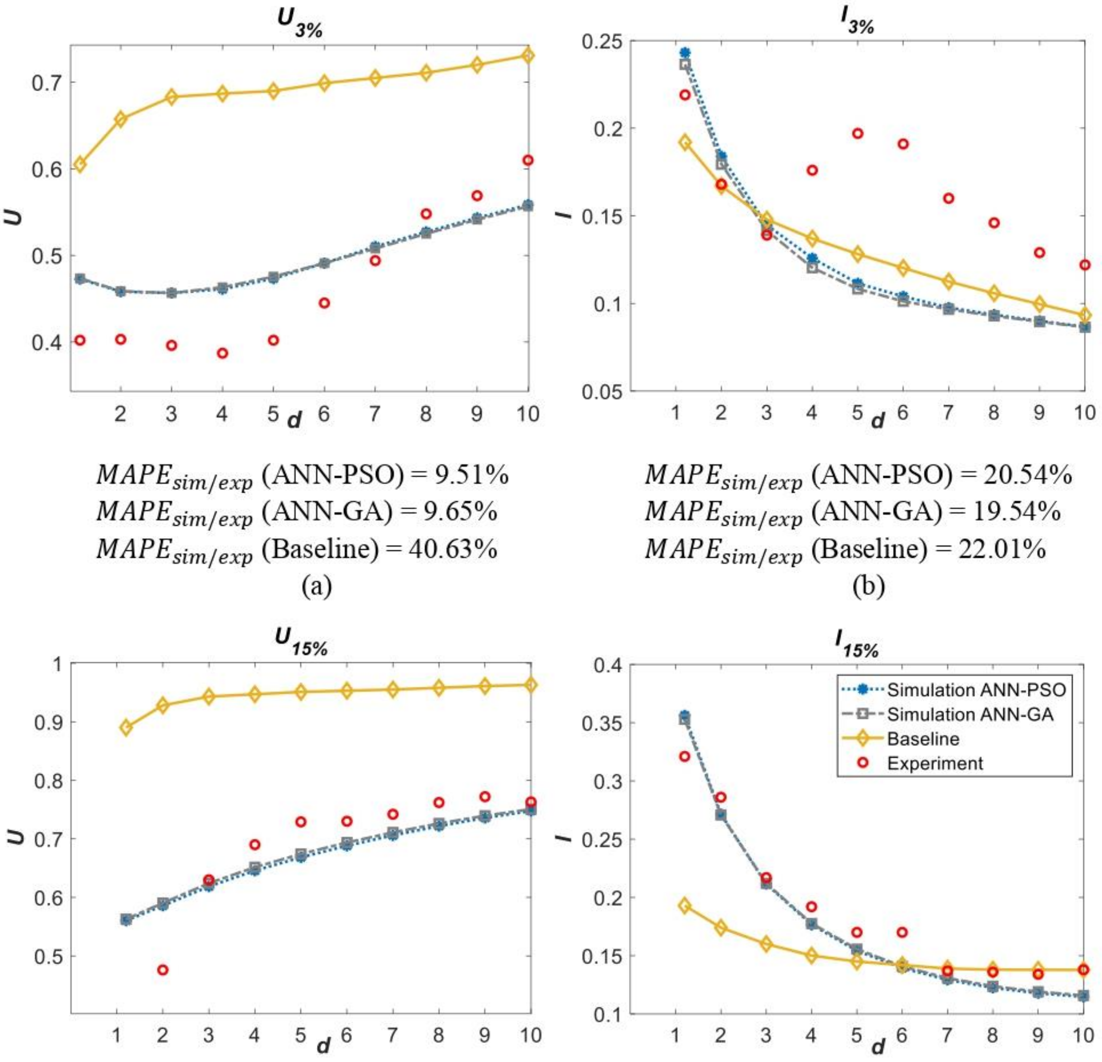
Figure 3.7 illustrates the vectors of physical experiment, results of the reference canopy model, and calibrated simulation results of ANN-GA and ANN-PSO by excluding d = 4, 5, and 6. Fig. 3.7(b) depicts the MAPEs of ANN-GA and ANN-PSO decreasing from, respectively, 27.08% and 28.61% to 19.54% and 20.54%. Comparably, Fig. 7(a) shows the MAPEs of ANN-GA and ANN-PSO decreasing from, respectively, 11.72% and 12.37% to 9.65% and 9.51%.

We improve $MAPE_{sim/exp}$ from 13.21% to 12.04% for ANN-GA and from 14.21% to 12.53% for ANN-PSO by excluding d = 4, 5, and 6. The common of the enhancement stems from the enhanced result for $I_{3\%}$, i.e., $MAPE_{sim/exp}$ for $I_{3\%}$ reduces from 27.08% to 19.54% for ANN-GA and from 28.61% to 20.54% for ANN-PSO. Though hypothetically, there is no guarantee that excluding locations where the simulator has known shortcomings will result in better performance, excluding them in the objective function to minimize $MAPE_{srgt/exp}$ enables further improvement in the Final Error.

In comparison to Olson et al. (2021), the result of this research gained by ANN-GA decreases $MAPE_{sim/exp}$ from 14.50% to 12.04%, indicating a 16.97% upgrade. Figure 3.8 illustrates the vectors of physical experiment, vectors of the simulation of ANN-GA

doi:10.6342/NTU202403713

excluding d = 4, 5, and 6, and those of GPR. Figure 3.8 demonstrates the superior fit of the vectors of simulation gained by ANN-GA to the vectors of physical experiment.



$MAPE_{sim/exp}$ (ANN-PSO) = 9.51%
$MAPE_{sim/exp}$ (ANN-GA) = 9.65%
$MAPE_{sim/exp}$ (Baseline) = 40.63%
(a)

$MAPE_{sim/exp}$ (ANN-PSO) = 20.54%
$MAPE_{sim/exp}$ (ANN-GA) = 19.54%
$MAPE_{sim/exp}$ (Baseline) = 22.01%
(b)

$MAPE_{sim/exp}$ (ANN-PSO) = 10.22%
$MAPE_{sim/exp}$ (ANN-GA) = 9.84%
$MAPE_{sim/exp}$ (Baseline) = 36.05%
(c)

$MAPE_{sim/exp}$ (ANN-PSO) = 9.85%
$MAPE_{sim/exp}$ (ANN-GA) = 9.13%
$MAPE_{sim/exp}$ (Baseline) = 14.12%
(d)

Fig. 3.7. Simulation outputs when excluding $d = 4, 5, 6$ for initial $I = 3\%$.

doi:10.6342/NTU202403713

3.6 Evaluating the quality of sample data

Recognizing the significance of assessing sample data quality prior to constructing the surrogate model, we employ a visual tool rooted in descriptive statistics for a swift evaluation of the sample data quality. Initially, we plot the vectors of physical experiment at each site denoted by blue dots with blue lines in Fig. 3.9, followed by the generation of boxplots based on the sample data at each location.

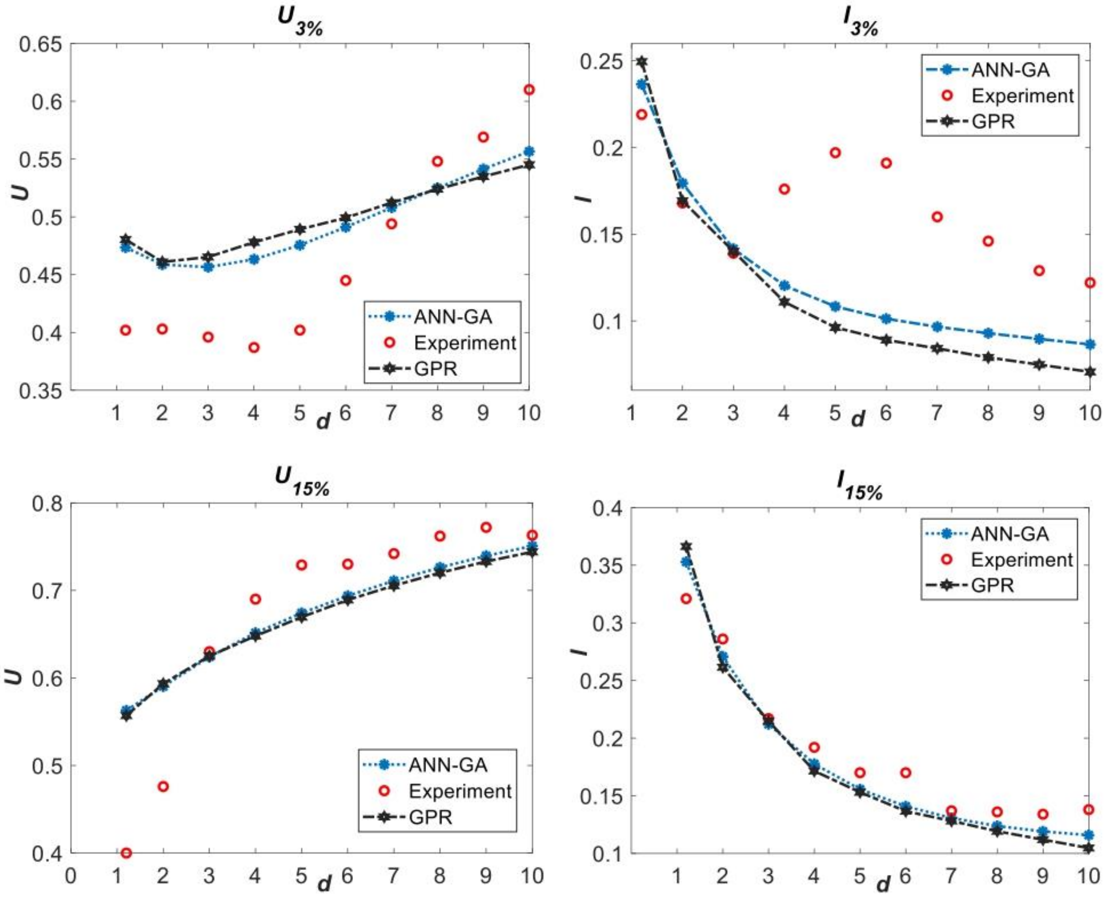


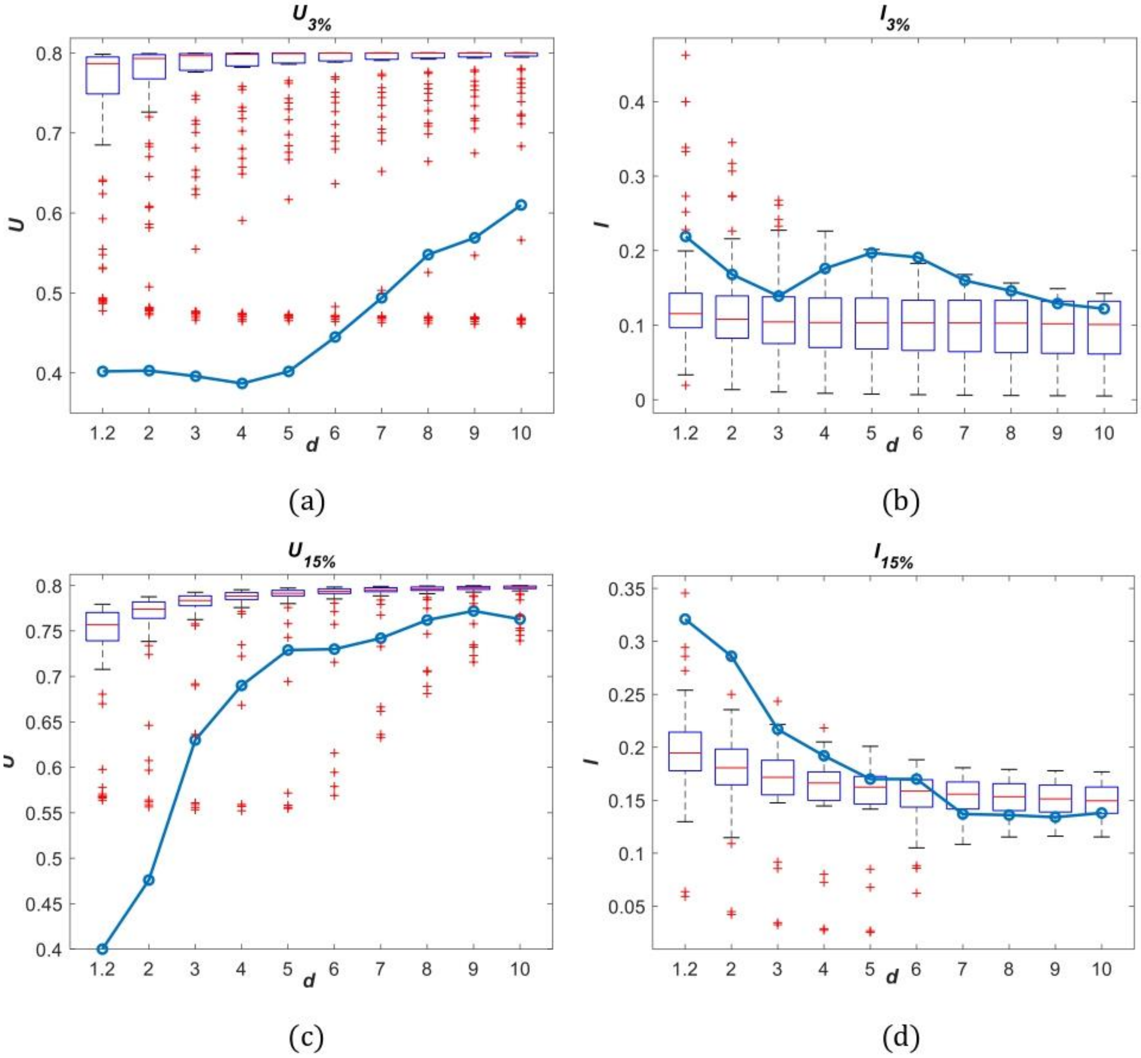
Fig. 3.8. ANN-GA and GPR performance

In Fig. 3.9(a), it is evident that the majority of the sample data in $U_{3\%}$ significantly deviates from the physical experiment vector. Conversely, Fig. 3.9(b) illustrates that the majority of the sample data in $I_{3\%}$ closely aligns with the physical experiment vector, indicating superior sample data quality for $I_{3\%}$ compared to $U_{3\%}$. When comparing Fig.

doi:10.6342/NTU202403713

3.9(a) and 3.9(c) to 3.9(b) and 3.9(d), it becomes apparent that the overall quality of training data is poorer for velocity U than for turbulence intensity I.

The representation in Fig. 3.9 underscores that the equal distance design utilized for generating sample data inaccurately captures the behaviors of the physical experiment vectors. Enhancing the quality of sample data warrants further investigation in upcoming studies.



(a)

(b)

(c)

(d)

Fig. 3.9. Physical experiment vectors and sample data.

doi:10.6342/NTU202403713

3.7 Summaries

This research employs an approach based on surrogate model to construct a simplified version of the behaviour of simulator, utilizing ANN to replicate its functions. Our study contributes significantly to two key aspects: firstly, we introduce a methodology integrating advanced optimization techniques such as GA and PSO with ANN to optimize changeable parameters in intensive systems. The ANN-PSO and ANN-GA performances are assessed using the disparity between outputs of physical experiment and results of calibrated simulator. ANN-GA and ANN-PSO achieve final errors of, respectively, 13.21% and 14.21%.

Secondly, we recognise and tackle an identified limitation of the simulator: its reliance on second-order functional shapes, which are insufficient for accurately modelling the behaviour observed in physical experiments, necessitating at least a third-order function. To relieve this issue, we selectively eliminate subgroups of simulation settings during parameter calibration. This refinement results in reduced final errors of 12.53% and 12.04%, respectively, for ANN-PSO and ANN-GA. Our methodology signifies enhancement over GPR used by previous study, yielding a remarkable 16.97% reduction in final error compared to their work.

The approach outlined in this research offers a fresh standpoint on parameter calibration for computationally intensive models, emphasizing the importance of addressing disparities between simulators and physical experiments. The calibrated simulator demonstrates usefulness in most natural environments with turbulence intensity around 15%. However, we note that the quality of sample data poses limitations to further improvement. To aid researchers and engineers in assessing data quality quickly, we develop a visual tool.

We design an experiment employing a simple equal0distance design to generate sample data. Despite improvements in final error through the surrogate model-based framework, our visual tool reveals inferior sample data quality for velocity compared to turbulence intensity in relation to the physical experiment's vectors. We suggest exploring more sophisticated experimental designs to generate sample data, which could lead to further enhancements in final error.

Chapter 4 An optimization model for marine current turbines layout problem against parameters ambiguity[3]

To improve the design and layout of marine farms, we introduce a non-parametric, scenario-based robust Bi-level Optimization model to address the ambiguity of environmental parameters in the Marine Current Turbine Installation problem. In this model, the upper level determines the number and locations of Marine Current Turbines (MCTs), while the lower level decides how to connect the installed MCTs. Initially, we solve all scenarios using a novel Greedy Heuristics Algorithm (GHA), which minimizes the cost per watt by accounting for wake effects and connection costs. The GHA outperforms both complete search and random search algorithms. Using the solutions obtained by the GHA, we then apply the min-max relative regret decision rule to identify a robust solution. A test case in Cook Inlet, Alaska, demonstrates that the robust solution performs well across all scenarios considered in our model.

4.1 Problem background

Covering two-thirds of the Earth's surface, marine environments have significant potential as a source of renewable energy (Ren et al., 2018). Recent projects like the MeyGen project in the United Kingdom, the Nova Scotia project in Canada, and the Raz Blanchard (Alderney) project in France highlight the increasing interest in marine energy

[3] This chapter is rewritten based on an article authored by this thesis writer titled “An Optimization Model for Marine Current Turbines Installation Problem against Parameters Ambiguity" which is submitted to Engineering Optimization (Nurdiansyah et al., 2024).

generation. Marine current converters are noted for their minimal impact on the underwater environment (Bahaj et al., 2011).

The power generation capacity of a marine farm is highly sensitive to the number and layout of its Marine Current Turbines (MCTs) due to wake effects, which involve the reduction of current velocity downstream of MCTs (Brutto et al., 2016). This reduction directly impacts power generation. Rao et al. (2016) suggest that an efficient setup can be achieved by arranging multiple MCTs in a single alignment. Additionally, Pinon et al. (2017) demonstrate that both turbulence intensity and the layout of MCTs significantly influence power generation.

However, previous studies have used fixed parameter values in their models, ignoring the dynamic nature of the marine environment, a crucial consideration in Marine Farm Design (MFD). Ambiguity, a form of uncertainty alongside randomness (Dang and Hong, 2013), means that parameter values are rarely constant in such environments. Therefore, these parameters are typically ambiguous.

Studies addressing ambiguity in the marine environment include Forbush et al. (2016), who use stochastic optimization techniques to handle the uncertainty associated with current velocity. Jónsdóttir and Milano (2019) employ stochastic differential equations to model short-term fluctuations in current velocity and turbulence intensity. In designing marine farms, Dai et al. (2017) and Ren et al. (2018) cluster daily current velocity data and subsequently calculate the probability associated with each cluster. The stochastic or probabilistic approach assumes that the probability distribution of ambiguous parameters is either known or can be estimated. However, as Hu et al. (2014) note, the necessary data may not always be available to accurately estimate these probability distributions.

doi:10.6342/NTU202403713

To address ambiguity, Gorissen et al. (2015) suggest that scenario-based robust optimization outperforms stochastic optimization while maintaining formulation tractability. Moreover, a non-parametric scenario-based robust optimization model eliminates the need to estimate the probability distribution of ambiguous parameters, which are included in the ambiguity set. In this study, we adopt a non-parametric scenario-based robust optimization approach to handle ambiguous parameters encountered in MFD. We propose a Bi-level Optimization model for the Marine Current Turbine Installation Problem (BO-MCTIP), which determines the number and location of MCTs in the upper-level model and establishes connections between them in the lower-level model. Our objective is to provide a robust decision that performs well across all scenarios with parameter ambiguity. Table 4.1 below provides details of the MFD methodologies.

Table 4.1. MFD methodologies

Reference	Model	Connection system	Parameters	Method
Brutto et al. (2016)	Single-level	No	Fixed	Deterministic
Rao et al. (2016)	Single-level	No	Fixed	Deterministic
Pinon et al. (2017)	Single-level	No	Fixed	Deterministic
Forbush et al. (2016)	Single-level	No	Ambiguous	Stochastic
Jónsdóttir and Milano (2019)	Single-level	No	Ambiguous	Stochastic
Dai et al. (2017)	Bi-level	Yes	Ambiguous	Probabilistic
Ren et al. (2018)	Bi-level	Yes	Ambiguous	Probabilistic
Our study	Bi-level	Yes	Ambiguous	Scenario-Based Robust optimization

doi:10.6342/NTU202403713

We explore all possible scenarios in the BO-MCTIP model using a novel Greedy Heuristic Algorithm (GHA) that incorporates Kruskal's algorithm (Kruskal, 1956) to solve the Minimum Spanning Tree (MST) problem. The GHA aims to minimize the cost per watt by considering wake effects and Submarine Power Cable (SPC) costs. We assess the effectiveness and efficiency of the GHA through experiments conducted in a marine farm with grid sizes ranging from 2×2 to 10×10 cells. The results show that the GHA outperforms both complete search and random search algorithms within a 24-hour computational time limit. At the design stage, where numerous what-if analyses are needed, an algorithm that can provide a good solution quickly is highly desirable.

Using the solutions obtained from the BO-MCTIP model, we apply a robust optimization model to find a robust solution using the min-max relative regret decision rule. We demonstrate the effectiveness of this robust optimization process with a case study in Cook Inlet, Alaska (US), showing that our robust solution performs well across all examined scenarios. To our knowledge, this is the first study to address ambiguous parameters in marine farms using non-parametric scenario-based robust optimization, while also considering wake effects and SPC connection costs in the BO-MCTIP model.

This paper is motivated by two main factors: first, the inadequacy of current models that use fixed parameters for marine farm design, which leads to suboptimal outcomes due to the dynamic and uncertain nature of the marine environment; and second, the need for resilient solutions capable of handling diverse environmental conditions, especially during the critical design phase that requires extensive what-if analyses. Additionally, there is a strong impetus to reduce the cost per watt generated by marine farms, considering both turbine placement and cable connection expenses.

In response to these challenges, this paper introduces two key innovations. First, we propose a non-parametric scenario-based robust BO-MCTIP model that addresses the inherent ambiguity of environmental parameters in marine settings, unlike previous approaches that rely on fixed parameter values. Second, we develop a Greedy Heuristic Algorithm (GHA) designed to efficiently identify robust solutions for the BO-MCTIP model. These solutions perform well across various environmental scenarios while minimizing the cost per watt, accounting for turbine placement and cable connection costs.

The rest of this chapter is structured as follows: Subchapter 4.2 describes the formulation of the BO-MCTIP model. Subchapter 4.3 outlines the robust optimization problem. In Subchapter 4.4, we propose the solution approach to the model and the robust optimization problem. Subchapter 5 presents the case study and its results. Finally, Subchapter 6 concludes and offers suggestions.

4.2 BO-MCTIP model

Figure 4.1 depicts our proposed BO-MCTIP model. The upper-level model is dedicated to the installation decision, optimizing the number and placement of MCTs while considering wake effects. The lower-level model handles the SPC connection decision, aiming to reduce the connection costs of the installed MCTs.

We define I as the set of MCTs to be installed in the marine farm and outline the parameters and decision variables as follows.

Parameters

N_{max} Maximum number of installed MCTs in the marine farm

d_{min} Minimum distance between two MCTs

DE Minimum power requirement generated by the marine farm

ρ	Marine water density
C_p	An MCT power coefficient
r	An MCT blade length
A_T	Area swept by an MCT blade (πr^2)
v_0	Inflow current velocity
C_T	An MCT's thrust coefficient
I_0	Turbulence intensity
x_i	MCT $i's$ location in the x-axis
y_i	MCT $i's$ location in the y-axis

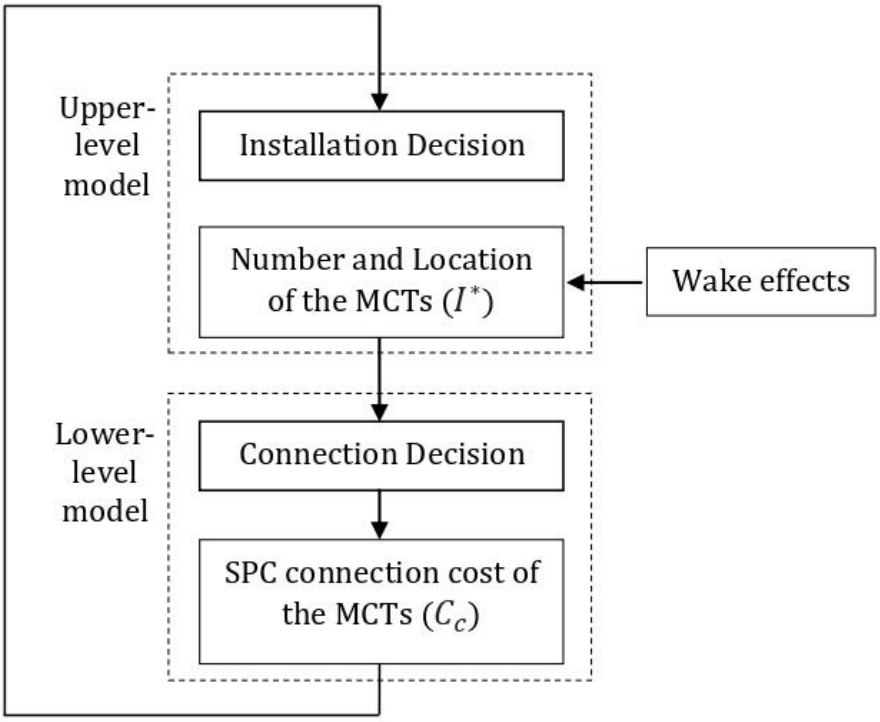


Fig. 4.1. Proposed BO-MCTIP model

dx_{hi} Distance between MCT h and i in the x-axis

dy_{hi} Distance between MCT h and i in the y-axis

d_{hi} Distance between MCT h and i

C_a An MCT’s cost.

C_{om} An MCT’s cost of operation and maintenance

P_c SPC cost per meter

Decision variables

t_i 0-1 variable that is equal to 1 if MCT i is installed at location (x_i, y_i), 0 otherwise.

l_{hi} 0-1 variable that is equal to 1 if MCT h and i are connected, 0 otherwise.

We start by dividing the planning area for MCT installation into J×K cells, each with a fixed size and uniform characteristics, where J and K are the total number of cells along the x- and y-axis, respectively. Each cell represents a potential location for an MCT, with (x_i, y_i) indicating the coordinates of MCT i within the marine farm. An MCT is placed in the center of each cell. Following Brutto et al. (2016), we assume that the marine current flows from the left, and we install the MCTs perpendicular to the flow direction to ensure they face a consistent flow.

4.2.1 The Analytical wake effects model

In our model, we incorporate wake effects to study the interactions between MCTs and their impact on downstream flow velocity. We employ the analytical wake effects model proposed by Brutto et al. (2016) for this analysis. According to Dai et al. (2017), this model provides greater accuracy compared to Jensen's model (Jensen, 1983), which often underestimates wake effects in scenarios with multiple MCTs.

Brutto et al. (2016) define a wake as the area behind an MCT where current velocity is reduced. For example, in the top right of Figure 4.2, downstream MCT 4 is within the wake generated by upstream MCT 1. As a result, the current velocity flowing past downstream MCT 4, denoted as v_{14}, decreases because upstream MCT 1 extracts energy from the incoming current flow, v_0.

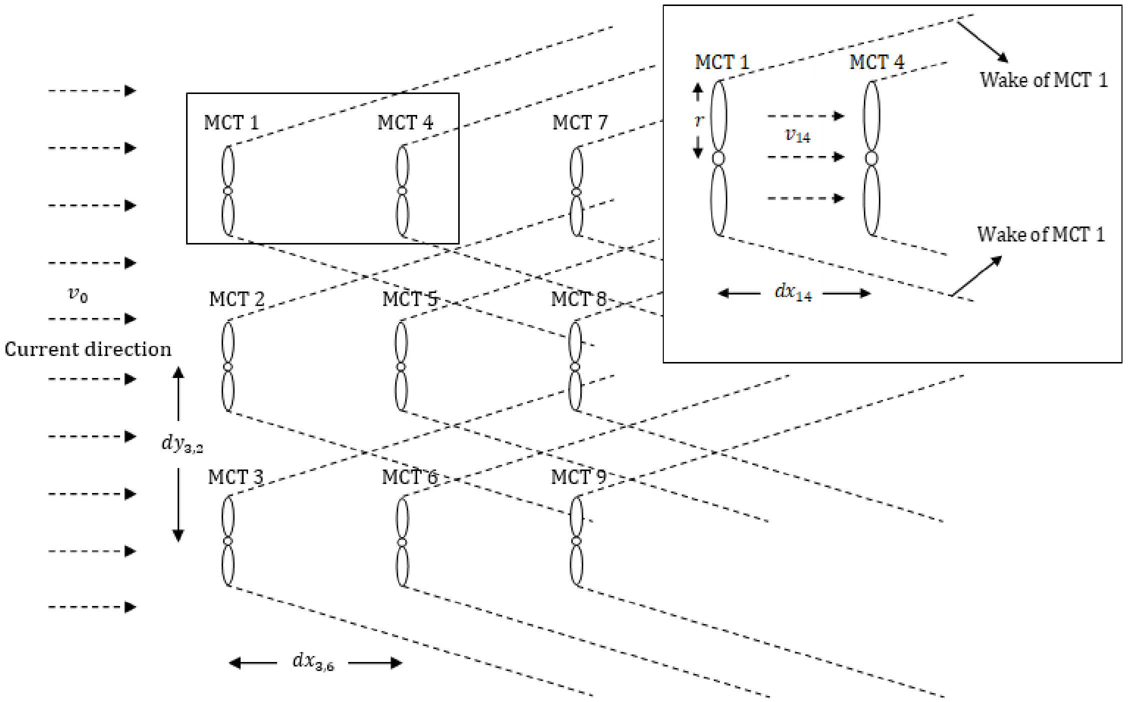


Fig. 4.2. Wake effects and MCTs

We denote the current velocity flow on downstream MCT i, caused by the wake effect of upstream MCT h, as v_{hi}, as

$$v_{hi} = \left(1 - \frac{1 - \sqrt{1 - C_T}}{[\sigma_{hi}/r]^2}\right) v_0, \quad h \neq i, \tag{4.1}$$

where σ_{hi} is the wake expansion of upstream MCT h on downstream MCT i. We evaluate the wake expansion as

$$\sigma_{hi} = \frac{r}{2.59}\left(-15.542 I_0^2 + 21.361 I_0 + 0.2184\right) \cdot \left[5.58\left(1 - e^{-0.051\frac{|dx_{hi}|}{2r}}\right) + 1.2\right], \;\; h \neq i, \tag{4.2}$$

which exponentially increases with the distance along the x -axis between upstream MCT h and downstream MCT i. This expansion is significantly influenced by turbulence intensity (Brutto et al., 2016). Figure 4.2 also shows the interactions between multiple MCTs: MCT 7 is within the wakes of MCTs 1, 2, and 4; MCT 8 is within the wakes of MCTs 1, 2, 3, and 5; and MCT 4 is only within the wake of MCT 1.

Next, we represent the current velocity of MCT i, denoted as v_i, under the influence of multiple wake effects, as

$$v_i = \left(1 - \sqrt{\sum_{i \in I} t_i \cdot \frac{O_{hi}}{A_T}\left(1 - \frac{v_{hi}}{v_0}\right)^2}\right) v_0, \quad \text{for } h \in W(i), h \neq i, \tag{4.3}$$

Where $W(i)$ represents the set of MCTs whose wakes impact MCT i, and v_{hi} is calculated as shown in equation (4.1). Brutto et al. (2016) assume that the total reduction in current velocity due to multiple wake effects is equal to the sum of the individual velocity reductions.

We further represent the elements of $W(i)$ as

$$W(i) = \{h: dx_{hi} < 0, |dy_{hi}| - r < \sigma_{hi}, h \neq i\} \tag{4.4}$$

and calculate the overlapping area of the wake generated by upstream MCT h on downstream MCT i, denoted as O_{hi}, as

$$O_{hi} = \begin{cases} \sigma_{hi}^2\left(\theta_{hi} - \dfrac{\sin(2\theta_{hi})}{2}\right) + r^2\left(\beta_{hi} - \dfrac{\sin(2\beta_{hi})}{2}\right); & \sigma_{hi} - r < |dy_{hi}| < \sigma_{hi} + r \\ A_T; & |dy_{hi}| \leq \sigma_{hi} - r, h \neq i. \\ 0; & |dy_{hi}| \geq \sigma_{hi} + r, \end{cases} \tag{4.5}$$

where

$$\theta_{hi} = cos^{-1}\left(\frac{\sigma_{hi}^2 + |dy_{hi}|^2 - r^2}{2|dy_{hi}|\sigma_{hi}}\right),\ h \neq i, \tag{4.6}$$

and

$$\beta_{hi} = cos^{-1}\left(\frac{\sigma_{hi}^2 - |dy_{hi}|^2 - r^2}{2|dy_{hi}|r}\right),\ h \neq i. \tag{4.7}$$

Figure 4.3 illustrates three potential conditions showing the affected areas of the wakes generated by upstream MCTs on the downstream MCT, as described in (4.5). The large circles represent the wake generated by upstream MCT h, while the small circles indicate the area swept by downstream MCT i. In Figures 4.3(a) and 4.3(b), the gray color indicates overlapping areas where the wake of MCT h partially and fully overlaps MCT i, respectively. Conversely, Figure 4.3(c) depicts no overlapping.

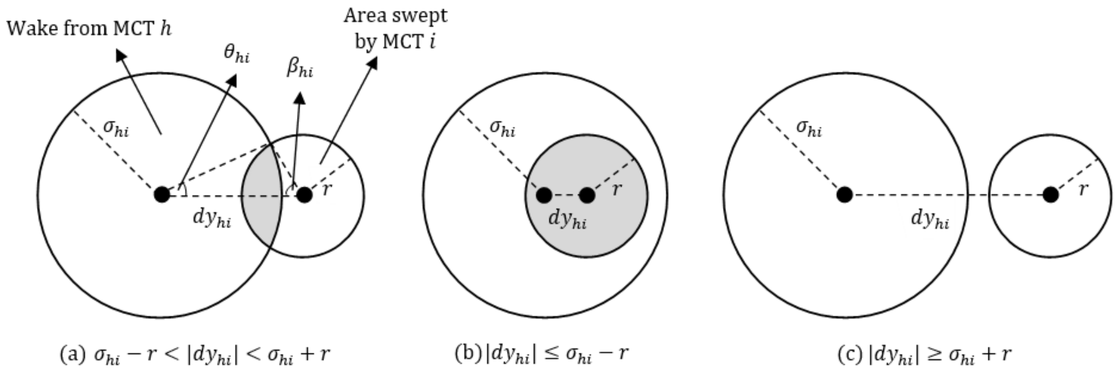


Fig. 4.3. Three potential conditions of the affected areas of the upstream MCT wakes on the downstream MCT (adapted from Mittal et al., 2016)

4.2.2 The Mathematical model for BO-MCTIP

The bi-level optimization framework, introduced by Bracken and McGill (1973), has been applied in numerous fields, including chemical reaction optimization (Abbassi et al., 2022), energy-traffic systems (Wang et al., 2017), and investment (Tsiodra and Chronopoulos, 2021). Following Nurdiansyah et al. (2024), we present the mathematical formulation of the upper- and lower-level models for our proposed BO-MCTIP model below.

Upper-level model

We propose a mixed-integer non-linear programming (MINLP) model to determine the installation decision. The objective function, defined in (4.9), seeks to minimize the cost per watt.

$$\min \frac{TC}{P_f}, \tag{4.9}$$

where

$$TC = M_c + C_c, \tag{4.10}$$

$$M_c = \left(\left(\frac{2}{3} + \frac{1}{3} e^{-0.00174 \cdot N^2}\right) \cdot C_a + C_{om}\right) N, \tag{4.11}$$

$$P_f = \sum_{i \in I} t_i \cdot P_i, \text{ and} \tag{4.12}$$

$$P_i = \frac{1}{2} \rho C_p A_T v_i^3, \quad \forall i \in I. \tag{4.13}$$

The total cost (TC), as calculated in (4.10), includes of two main components: the cost of installing the MCTs (M_c) and the cost of the SPC connection (C_c) linking all installed MCTs, determined in the lower-level model. Notably, C_c is influenced by the distances between MCTs, indicating interaction between the upper- and lower-level models. The installation costs (M_c) encompass two elements, C_a and C_{om}. We calculate M_c using (4.11), where $\left(\frac{2}{3}+\frac{1}{3}e^{-0.00174\cdot N^2}\right)N$ represents the scaling cost if a large number of MCTs are installed (N denotes the number of installed MCTs), as proposed by Mosetti et al. (1994). The total power generated by all installed MCTs within the marine farm, denoted as P_f, is obtained by summing the power generation of each individual MCT, as determined by (4.12). The power generation of individual MCT i, represented as P_i, is computed in (4.13) and is dependent on the velocity (v_i) calculated in (4.3). The constraints are as follows.

$$\sqrt{{dx_{hi}}^2+{dy_{hi}}^2} \geq d_{min}, \quad \forall h,i \in I; h \neq i. \tag{4.14}$$

$$\sum_{i \in I} t_i = N. \tag{4.15}$$

$$\sum_{i \in I} t_i \leq N_{max}. \tag{4.16}$$

$$P_f \geq DE. \tag{4.17}$$

Constraint (4.14) stipulates that MCTs must maintain a minimum clearance distance, denoted as d_{min}, from each other. Constraint (4.15) defines N. Constraint (4.16) limits the number of installed MCTs, ensuring it does not exceed N_{max}. Lastly, Constraint (4.17) ensures that the marine farm's power generation meets the minimum power requirement, denoted as DE. We define the nonnegative and binary variables as follows.

$$t_i \in \{0,1\}, \quad \forall i \in I. \tag{4.18}$$

$$N \geq 0. \tag{4.19}$$

$$v_i \geq 0, \quad \forall i \in I. \tag{4.20}$$

doi:10.6342/NTU202403713

$$P_i \geq 0, \ \ \forall i \in I. \tag{4.21}$$

$$P_f \geq 0. \tag{4.22}$$

$$M_c \geq 0. \tag{4.23}$$

$$TC \geq 0. \tag{4.24}$$

Lower-level model

We use the lower-level model to determine the connections between installed MCTs, treating it as a Minimum Spanning Tree (MST) problem. Let I^* denote the set of installed MCTs and Q represent the set of connections between any two installed MCTs, defined as $Q = \{\{h, i\}: h, i \in I^* \text{ and } h \neq i\}$. Additionally, d_{hi} represents the distance between two installed MCTs h and i within the marine farm. The set I^* is obtained from the upper-level model. The connections between MCTs are symmetric, meaning the distance between any two MCTs is the same regardless of the connection direction. The lower-level decision variable, l_{hi}, determines the connections between the locations of installed MCTs specified in the upper-level model. We mathematically model the connection costs in the marine farm as follows.

$$C_c = \min \ P_c \sum_{h,i \in I^*, h \neq i} l_{hi} {d_{hi}}^*, \tag{4.25}$$

subject to

$$\sum_{h,i \in I^*, \ \ h \neq i} l_{hi} = N^* - 1, \tag{4.26}$$

$$\sum_{h,i \in \delta(S), \ \ h \neq i} l_{hi} \leq |S| - 1, \quad \forall S \subseteq I^* \text{ where } |S| > 1, \tag{4.27}$$

$$l_{hi} \ \in \{0,1\}, \qquad \forall h, i \in I^*, h \neq i. \tag{4.28}$$

$$C_c \geq 0. \tag{4.29}$$

The objective function aims to minimize the cost of connecting all installed MCTs, as defined in equation (4.25). Constraint (4.26) specifies that there are exactly $N^* - 1$ connections, where N^* represents the number of installed MCTs determined in the upper-level decision. Constraint (4.27) is the cycle elimination constraint, ensuring no cycles by limiting the number of connections in every subset S to at most $|S| - 1$. Here, S is a subset of I^*, and $\delta(S) \subseteq Q$ represents the subset of connections linking installed MCTs in S. Constraints (4.28) and (4.29) define the binary variable and non-negativity settings, respectively.

Based on the above explanation, the two levels of the proposed bi-level optimization marine current turbine installation problem (BO-MCTIP) are connected in two ways. First, through decision variables: the upper-level model determines the number of Marine Current Turbines (MCTs) and their locations, defining I as the set of MCTs to be installed in the marine farm. The decisions made in the upper-level model directly determine the set of installed MCTs (and their locations), which become inputs (the set I^*) for the lower-level problem. The lower-level model then determines the optimal connections between these installed MCTs. However, the distances of these connections (part of the set Q) influence the SPC connection costs in the upper-level objective function.

Second, through cost calculations: the upper-level model minimizes the cost per watt, with one component of this cost (TC) being the SPC connection cost (C_c), significantly influenced by the distances between MCTs. The lower-level model focuses on minimizing the SPC connection cost (C_c). The optimal connection cost determined here directly impacts the value of the upper-level objective function.

doi:10.6342/NTU202403713

In summary, the upper-level model decides the number of MCTs and their placements, providing inputs for the lower-level model. The lower-level model, in turn, optimizes MCT connections, with the resulting distances feeding back into the upper-level's cost calculation. This illustrates the bi-level structure, featuring a distinct hierarchy, yet intertwined as the outcome of one directly influences the other.

4.3 Robust optimization model

Our proposed BO-MCTIP model assumes that all input parameters are constant. However, in reality, parameters such as inflow current velocity ($\boldsymbol{v_0}$), turbulence intensity ($\boldsymbol{I_0}$), marine water density ($\boldsymbol{\rho}$), and the power coefficient of the MCT ($\boldsymbol{C_p}$) are ambiguous. In mathematical optimization models, ambiguity in input parameters may result in impractical optimal solutions (Ben-Tal & Nemirovski, 2000). Therefore, our goal is to configure an effective MFD (installation and connection decisions) despite this ambiguity.

To identify an MFD that performs well across each investigated scenario, we set four ambiguous parameters ($\boldsymbol{v_0}$, $\boldsymbol{I_0}$, $\boldsymbol{\rho}$, and $\boldsymbol{C_p}$) to have scenario-specific values representing these parameters in our BO-MCTIP. It is important to note that the variable values $\boldsymbol{t_i}$ and $\boldsymbol{l_{hi}}$ may not be the most effective solution for every particular situation outlined by the BO-MCTIP model.

To obtain a robust solution, we employ the min-max relative regret decision rule as the robustness indicator, measuring the best worst-case percentage deviation from optimality among the investigated scenarios. In decision-making, this rule establishes a benchmarking orientation, guiding the decision-maker to consider the best possible outcome under any investigated scenario and to determine a solution that performs closely

doi:10.6342/NTU202403713

to this benchmark for each scenario (Kouvelis and Yu, 2013). In other words, a robust solution provides the minimum of the maximum percentage deviation among all investigated scenarios.

We define $\boldsymbol{\Omega}$ as the set of all investigated feasible input parameter scenarios, $\boldsymbol{\mathcal{Z}^*_{\omega}}$ represents the minimum cost per watt for scenario $\boldsymbol{\omega \in \Omega}$, and $\boldsymbol{\mathcal{R}_{\omega}}$ represents the corresponding cost per watt of an MFD for scenario $\boldsymbol{\omega \in \Omega}$. We then formulate the robust optimization model as follows:

$$\min \gamma \tag{4.30}$$

subject to

$$\gamma \geq \frac{\mathcal{R}_{\omega} - \mathcal{Z}^*_{\omega}}{\mathcal{Z}^*_{\omega}}, \qquad \forall \omega \in \Omega, \tag{4.31}$$

one set of the proposed BO-MCTIP constraints for each scenario $\boldsymbol{\omega \in \Omega}$,

where $\boldsymbol{\gamma}$ is the maximum relative regret of the cost per watt.

To implement the robust optimization model, we evaluate all investigated scenarios in the BO-MCTIP and apply the proposed algorithm to solve it for each scenario $\omega \in \Omega$, yielding the solution $\mathcal{Z}^*_{\omega}$. Next, we seek to find a robust solution that closely approximates $\mathcal{Z}^*_{\omega}$ under the min-max relative regret criterion. This process results in determining the robust MFD associated with the robust solution.

4.4 Solution approach

As discussed in subchapter 4.2, we acknowledge that the distance between MCTs must be sufficient to minimize wake effects on downstream MCTs while also recognizing that increasing these distances raises the SPC cost. To balance these two performance measures, we propose a novel Greedy Heuristic Algorithm (GHA) that minimizes the cost

doi:10.6342/NTU202403713

per watt by considering wake effects. This algorithm installs MCTs based on the greedy rule and uses Kruskal's algorithm to minimize the connection cost.

Our proposed GHA consists of a *finite enumeration step* and an *installation step*. In the finite enumeration step, we investigate the potential x-axis columns to install the MCTs by enumerating all possible combinations of the x-axis columns in an empty marine farm and then select the columns with the minimum cost per watt based on (4.9). In the installation step, we greedily install MCTs one by one at a y-axis row for each x-axis column selected in the finite enumeration step. Each cell in our marine farm is restricted to two possibilities, i.e., installing or not installing an MCT so that we have $2^{J \cdot K}$ solution possibilities. Our proposed GHA saves computational effort by only enumerating all possible combinations of the x-axis columns using the finite enumeration step, resulting in only 2^J possibilities.

In the Initialization Step, the pseudocode, **GHA**, begins with an empty marine farm that has $J \times K$ cells and initializes $\mathcal{Z}_{\omega}^{*}$ to be an infinite number. We initially set $N_{max} = J \cdot K$, $N = 1$, $j = 1$, $k = 1$, and $g = 1$. In the Main Step, for each scenario $\omega \in \Omega$ in the empty marine farm, the finite enumeration step investigates $\binom{J}{j}$, all possible combinations of j columns of the x-axis in J, by positioning MCTs for all rows in the selected j columns followed by applying Kruskal's algorithm to connect the installed MCTs. Then we select j columns with the minimum $\frac{TC_{\omega}}{P_{f_{\omega}}}$ among $\binom{J}{j}$ combinations, where TC_{ω} and $P_{f_{\omega}}$ are the total cost and the total power generation, respectively, for scenario $\omega \in \Omega$. In the finite enumeration step, we find the optimal MFD among $\binom{J}{j}$.

In the Installation Step, we utilize an empty marine farm at the beginning of the GHA and install N MCTs. For the j columns selected in the finite enumeration step, we

doi:10.6342/NTU202403713

install an MCT at row y_k, where k starts from 1 to K. If $N \leq N_{max}$, then we calculate P_{f_ω}. If $P_{f_\omega} < DE$, we record the updated MFD and go back to the installation step with $k+1$ and $N+1$. We install an additional MCT at row y_{k+1}. We install an MCT row by row from y_1 to y_K while evaluating P_{f_ω} at the selected column. If $P_{f_\omega} < DE$ after installing an MCT for each row, then we record the updated MFD and go back to the finite enumeration step with $j+1$, $k=1$, and $N+1$. We enumerate $\binom{J}{j+1}$ to find the optimal MFD among $\binom{J}{j+1}$ and go to the installation step. Using the recorded MFD, we update the MFD by first removing all installed MCTs at row y_k obtained in $\binom{J}{j}$ followed by installing an MCT at row y_k for each selected $j+1$ columns obtained in $\binom{J}{j+1}$ to obtain the minimum objective function value. This leads us to propose Collorary 1. The current step combines the MFD obtained in $\binom{J}{j}$ and $\binom{J}{j+1}$.

Corollary 1. *Removing first all installed MCTs at row y_k started from $k=1$ to K in the recorded MFD obtained in $\binom{J}{j}$ before installing an additional MCT to the columns in the x-axis selected in $\binom{J}{j+1}$ provides the minimum objective function value compared to installing an additional MCT without removing the installed MCTs.*

Proof. We obtain the optimal MFD in $\binom{J}{j}$ and $\binom{J}{j+1}$ since we enumerate all possible combinations of the MFD in $\binom{J}{j}$ and $\binom{J}{j+1}$ and select an MFD with the minimum wake effect (the power generation increases if the wake effect is minimized) and SPC cost so that the objective function value is optimal. Suppose that in the installation step, we install MCTs for all rows at the column in the x-axis selected in $\binom{J}{j}$ and record the MFD. To update the recorded MFD after finding the optimal MFD in $\binom{J}{j+1}$, installing an additional

MCT to the columns in the x-axis without removing the installed MCTs at y_k obtained in $\binom{J}{j}$ provides the greater objective function value than removing first the installed MCTs at y_k since the wake effect is not minimized in row y_k when the installed MCTs obtained in $\binom{J}{j}$ are not removed. ■

If $P_{f_\omega} \geq DE$, we apply Kruskal's algorithm. We obtain $\mathcal{Z}_\omega$ by calculating $\frac{TC_\omega}{P_{f_\omega}}$, and compare it to $\mathcal{Z}_\omega^*$. If $\mathcal{Z}_\omega < \mathcal{Z}_\omega^*$, then $\mathcal{Z}_\omega$ updates $\mathcal{Z}_\omega^*$ and we repeat to install an additional MCT at row y_{k+1} for the same j columns selected in the finite enumeration step. We want to minimize the SPC cost since installing an MCT at row y_k followed by installing an MCT at row y_{k+1} to the same selected column in the x-axis provides a lower cost than installing an MCT at row y_k followed by installing an MCT at a row other than y_{k+1} (see Corollary 2). If $\mathcal{Z}_\omega < \mathcal{Z}_\omega^*$ after MCTs are already installed at all rows, we record the MFD and go back to the finite enumeration step with $j+1$, $k=1$, and $N+1$.

Corollary 2. *In the installation step, installing an additional MCT at an adjacent row of the y-axis row to the same selected columns in the x-axis provides a lower cost compared to installing an MCT at the row other than the adjacent row in the y-axis.*

Proof. Suppose that we have installed an MCT at row $\boldsymbol{y_k}$. If an additional MCT is installed at row $\boldsymbol{y_{k+1}}$, then the SPC cost is $\boldsymbol{P_c \cdot dy_{hi}}$, which provides a lower cost compared to installing an additional MCT at row $\boldsymbol{y_{k+2}, y_{k+3}}$, etc., where the SPC cost is $\boldsymbol{P_c \cdot 2dy_{hi}, P_c \cdot 3dy_{hi}}$, etc., respectively. ■

If $\mathcal{Z}_\omega \geq \mathcal{Z}_\omega^*$, installing an additional adjacent MCT returns an inefficient MFD compared to the best MFD obtained until the current step. To avoid the potential issue of staying in the local minimum, the GHA continues to investigate the MFD by adding an MCT at the y-axis row for each selected column in the x-axis. If $k < K$, i.e., MCTs are not yet installed at all rows, then we record the MFD and continue to install an MCT at a y-axis row with $k+1$ and $N+1$. If $k=K$, i.e., MCTs are already installed at all rows, we record the MFD and go back to the finite enumeration step with $j+1$, $k=1$, and $N+1$. The GHA finishes searching for the solution of a specific scenario $\omega \in \Omega$ if either it collects K times of such inefficient MFDs or $N > N_{max}$. The best solution for a specific scenario $\omega \in \Omega$ is $\mathcal{Z}_{\omega}^{*}$ and the MFD for a specific scenario $\omega \in \Omega$ is obtained associated with $\mathcal{Z}_{\omega}^{*}$. The GHA terminates after examining all investigated realizable scenarios. To summarize the GHA, the finite enumeration step is always applied to an empty marine farm and neglects the previous results to ensure obtaining the optimal MFD between combination $\binom{J}{j}$ for $j = 1, 2, \dots, J$ and the installation step always installs MCTs one by one starting from the first row to the adjacent row in the y-axis.

An example of a marine farm with 3×3 cells illustrates the steps of the GHA. Since j starts from 1, we investigate $\binom{3}{1}$ as shown in Fig. 4.4 and evaluate them by using (4.9), where one dot denotes an MCT and the lines linking the dots are the SPC connections. The objective function values of the MFD in Figs. 4.4(a), (b), and (c) are the same without any wake effects. In Fig. 4.4(a) we assume the GHA selects x_1 column to install the MCT as the computer program chooses the smallest index for the same objective function evaluations.

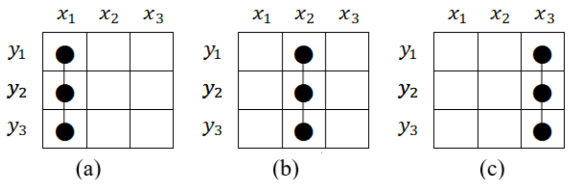


Fig. 4.4. The finite enumeration step when the total number of selected columns in the x-axis is 1 (j = 1)

doi:10.6342/NTU202403713

To illustrate the installation step, we use an empty marine farm (there is no recorded MFD). We first install an MCT ($N = 1$) at (x_1, y_1) as shown in Fig. 4.5(a). Since $N \leq N_{max}$, we then calculate P_{f_ω}. Since $P_{f_\omega} < DE$ and now $N = 2$, we continue to install an additional MCT at (x_1, y_2) as shown in Fig. 4.5(b). Since P_{f_ω} is still less than DE, we install another additional MCT at (x_1, y_3). We then apply Kruskal's algorithm if $P_{f_\omega} \geq DE$ as shown in Fig. 4.5(c) and obtain $\mathcal{Z}_\omega$. We set the best objective function value $\mathcal{Z}_\omega^*$ obtained as $\mathcal{Z}_\omega$, since $\mathcal{Z}_\omega < \mathcal{Z}_\omega^*$.

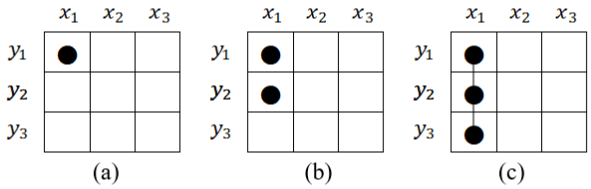


Fig. 4.5. Installing an MCT at row y_k followed by y_{k+1} from y_1 to y_K at column x_1

doi:10.6342/NTU202403713

GHA. The algorithm for solving scenario $\omega \in \Omega$ in the proposed BO-MCTIP

```
Initialization Step:
Set the empty marine farm with $J \times K$ cells.
Let $\mathcal{Z}_{\omega}^{*}$ be an infinite number.
Set $N_{max} = J \cdot K$, $N = 1$, $j = 1$, $k = 1$, and $g = 1$.
Go to Main Step.

Main Step:
Start.
For each scenario $\omega \in \Omega$, do
  Input parameters for scenario $\omega \in \Omega$.
  Finite enumeration step
    In an empty marine farm, enumerate $\binom{J}{j}$ by positioning MCTs for all rows in the $j$ columns.
    Apply Kruskal's algorithm to connect the MCTs.
    Calculate $\frac{TC_{\omega}}{P_{f_{\omega}}}$. Select $j$ columns with the minimum $\frac{TC_{\omega}}{P_{f_{\omega}}}$ among $\binom{J}{j}$ combinations.
  Installation step
    Use the recorded MFD. Use an empty marine farm if there is no recorded MFD.
    Install $N$ MCTs. Remove all installed MCTs at row $y_k$.
    Update the recorded MFD by installing an MCT at row $y_k$ for each $j$ columns selected in the finite enumeration step.
    If $N \leq N_{max}$, then calculate $P_{f_{\omega}}$.
      If $P_{f_{\omega}} < DE$, then
        If $k < K$, then record the MFD. Go to the installation step with $k + 1$ and $N + 1$.
        Else Record the MFD. Go to the finite enumeration step with $j + 1$, $k = 1$, and $N + 1$.
        End if
      Else Apply Kruskal's algorithm. Calculate $\frac{TC_{\omega}}{P_{f_{\omega}}}$ as $\mathcal{Z}_{\omega}$.
        If $\mathcal{Z}_{\omega} < \mathcal{Z}_{\omega}^{*}$, then $\mathcal{Z}_{\omega}^{*} = \mathcal{Z}_{\omega}$.
          If $k < K$, then record the MFD. Go to the installation step with $k + 1$ and $N + 1$.
          Else Record the MFD. Go to the finite enumeration step with $j + 1$, $k = 1$, and $N + 1$.
          End if
        Else
          If $g < K$, then
            If $k < K$, then record the MFD. Go to the installation step with $k + 1$, $g + 1$, and $N + 1$.
            Else Record the MFD. Go to the finite enumeration step with $j + 1$, $g + 1$, $k = 1$, and $N + 1$.
            End If
          Else $\mathcal{Z}_{\omega}^{*}$ is the best solution for scenario $\omega \in \Omega$. Return the MFD associated with $\mathcal{Z}_{\omega}^{*}$.
          End if
        End if
      End if
    Else $\mathcal{Z}_{\omega}^{*}$ is the best solution for scenario $\omega \in \Omega$. Return the MFD associated with $\mathcal{Z}_{\omega}^{*}$.
    End if
End for
Stop.
```

We record the MFD in Fig. 4.5(c) and go back to the finite enumeration step with $j+1$, $k=1$, and $N+1$. Now we investigate $\binom{3}{2}$ as shown in Fig. 4.6, where the MFD in Fig. 4.6(b) provides the minimum objective function value such that we select columns x_1 and x_3, and then go to the installation step. Now, $N=4$. Using the recorded MFD in Fig. 4.5(c), we remove the installed MCT at (x_1, y_1) and install an MCT at (x_1, y_1) and (x_3, y_1) as shown in Fig. 4.7(a) while comparing $\mathcal{Z}_\omega$ to $\mathcal{Z}_\omega^*$. If $\mathcal{Z}_\omega < \mathcal{Z}_\omega^*$, we replace $\mathcal{Z}_\omega^*$ by $\mathcal{Z}_\omega$. We continue to remove an MCT at (x_1, y_2) and install an additional MCT at (x_1, y_2) and (x_3, y_2) as shown in Fig. 4.7(b) since we need to install five MCTs. If $\mathcal{Z}_\omega \geq \mathcal{Z}_\omega^*$, the GHA continues to investigate the MFD by adding an MCT at a row in the y-axis for each selected column in the x-axis and terminates if K times of the inefficient MFDs are collected.

In Fig. 4.7, we assume that the MFD in Fig. 4.7(b) is the first MFD returning $\mathcal{Z}_\omega \geq \mathcal{Z}_\omega^*$. The GHA continues to remove an MCT at (x_1, y_3) and install an MCT at (x_1, y_3) and (x_3, y_3) as shown in Fig. 4.7(c). If the current $\mathcal{Z}_\omega \geq \mathcal{Z}_\omega^*$, the GHA continues to investigate other MFDs until there are three inefficient MFDs in the record. Since we have already installed an MCT for each row at columns x_1 and x_3, we go back to the finite enumeration step to investigate $\binom{3}{3}$ as shown in Fig. 4.8, where columns x_1, x_2, and x_3 are selected. In the MFD of the three columns, we remove the installed MCT at (x_1, y_1) and (x_3, y_1) and install an MCT at (x_1, y_1), (x_2, y_1), and (x_3, y_1) as shown in Fig. 4.9 ($N=7$). If $\mathcal{Z}_\omega \geq \mathcal{Z}_\omega^*$, the GHA terminates and returns the best MFD associated with the $\mathcal{Z}_\omega^*$ as the solution as shown in Fig. 4.7(a).

doi:10.6342/NTU202403713

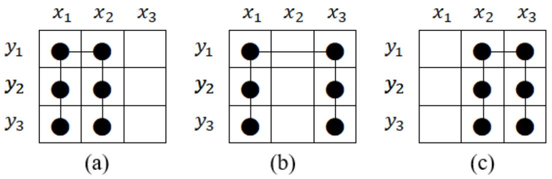


Fig. 4.6. The finite enumeration step when the total number of selected columns in the $\boldsymbol{x}$-axis is 2 (j = 2)

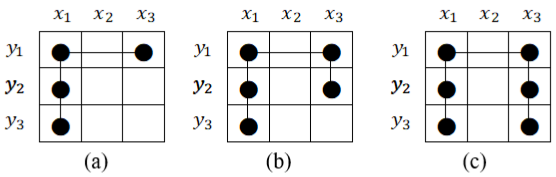


Fig. 4.7. Installing MCT one by one in the installation step when j = 2

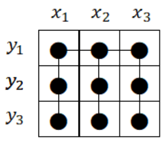


Fig. 4.8. The finite enumeration step when the total number of selected columns in the $\boldsymbol{x}$-axis is 3 (j = 3)

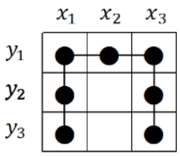


Fig. 4.9. Installing MCT one by one in the installation step when j = 3

4.4.1 Computational performance of the GHA

To evaluate the computational performance of our proposed GHA, we compared it with the complete search and random search algorithms using a marine farm ranging from 2×2 cells to 10×10 cells. We used the input parameters for the ambiguous variables from Scenario 1 of the case study (see subchapter 4.5). In the complete search algorithm, we incrementally increased the number of MCTs starting from one, examined all possible MCT locations, and then applied Kruskal’s algorithm. In the random search algorithm, we randomly generated the number and locations of MCTs and then applied Kruskal's algorithm. Each algorithm ran for a maximum of 4,320 minutes (3 days). The complete search algorithm stopped if it completed the search before 4,320 minutes, and the GHA stopped once it finished the solution search.

Table 4.2 shows that for marine farms with 2×2 cells to 7×7 cells, the complete search algorithm obtained the optimal solution. The random search algorithm found the same optimal solution as the complete search algorithm for marine farms with 2×2 cells to 5×5 cells. The GHA also achieved the same optimal solution as the complete search algorithm for marine farms with 2×2 cells to 7×7 cells but did so in less time.

For marine farms ranging from 8×8 cells to 10×10 cells, the complete search algorithm failed to find the optimal solution within 4,320 minutes. Similarly, the random search algorithm did not achieve the optimal solution within 4,320 minutes for marine farms with 6×6 cells to 10×10 cells. For marine farms with 8×8 cells to 10×10 cells, the GHA not only outperformed the complete search algorithm in terms of computational time but also in the quality of the solution found. This highlights GHA's efficiency in both solution quality and computational time for larger marine farms. However, since the complete search algorithm could not finish the solution search within 4,320 minutes for marine farms ranging from 8×8 cells to 10×10 cells, the optimality of the GHA solutions cannot be guaranteed. Compared to the random search algorithm, the GHA consistently outperformed it in terms of solution quality for marine farms with 6×6 to 10×10 cells. Figure 4.10, which plots the change in solution obtained by the GHA, complete search, and random search algorithms over 1,440 minutes for marine farms with 3×3 cells and 6×6 cells, demonstrates the significant reduction in computational time achieved by the GHA.

doi:10.6342/NTU202403713

Table 4.2. Computational performance comparisons

Marine farm	Proposed GHA		Complete search algorithm		Random search algorithm		Relative percentage (%)	
	$\mathcal{Z}_1^*$	Computational time (minutes)	$\mathcal{Z}_{C1}^*$	Computational time (minutes)	$\mathcal{Z}_{R1}^*$	Computational time (minutes)	$\frac{\mathcal{Z}_{C1}^* - \mathcal{Z}_1^*}{\mathcal{Z}_1^*}$	$\frac{\mathcal{Z}_{R1}^* - \mathcal{Z}_1^*}{\mathcal{Z}_1^*}$
2×2	0.5132	0.9	0.5132	9.5	0.5132*	4,320.0	0.00	0.00
3×3	0.4873	6.8	0.4873	371.2	0.4873*	4,320.0	0.00	0.00
4×4	0.4873	11.1	0.4873	1,486.3	0.4873*	4,320.0	0.00	0.00
5×5	0.4796	85.3	0.4796	2,344.6	0.4796*	4,320.0	0.00	0.00
6×6	0.4460	127.8	0.4460	3,120.5	0.4521*	4,320.0	0.00	1.36
7×7	0.4186	213.6	0.4186	4,296.1	0.4202*	4,320.0	0.00	0.38
8×8	0.4154	278.2	0.4186*	4,320.0	0.4292*	4,320.0	0.77	3.32
9×9	0.4064	373.8	0.4186*	4,320.0	0.4273*	4,320.0	3.00	5.14
10×10	0.4014	487.5	0.4186*	4,320.0	0.4298*	4,320.0	4.29	7.07

Note:

1. $\mathcal{Z}_1^*$, $\mathcal{Z}_{C1}^*$ and $\mathcal{Z}_{R1}^*$ are the solutions obtained by the GHA, complete search and random search algorithms, respectively.
2. * indicates the best the complete search algorithm or random search algorithm obtains within 4,320 minutes.

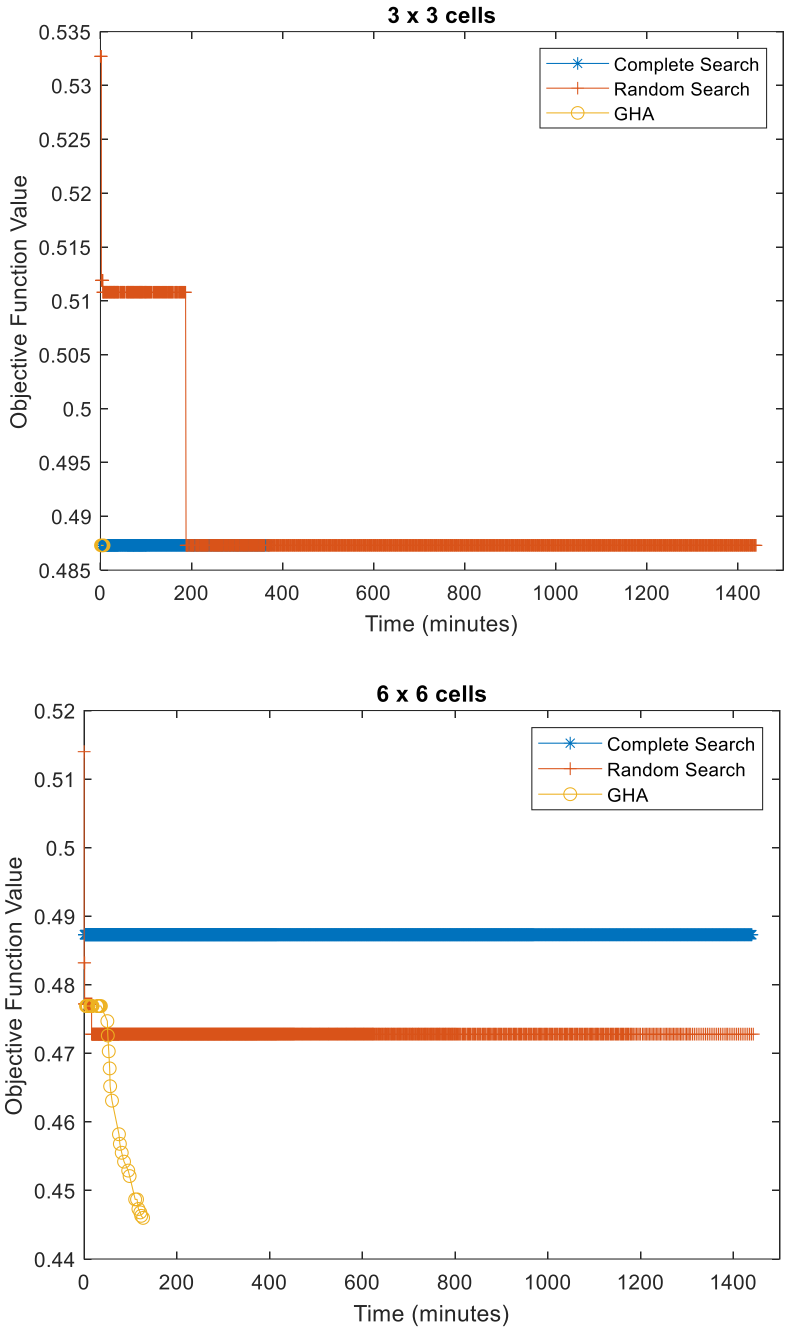


Fig. 4.10 Computational performance comparisons for the marine farm with 3×3 cells and 6×6 cells

4.4.2 Robust GHA

To solve the robust optimization model, we employ the pseudocode **robust GHA** and use the solution for each scenario $\omega \in \Omega$ in the BO-MCTIP obtained by the GHA to find a robust solution under the min-max relative regret. The robust GHA begins with an empty marine farm with $J{\times}K$ cells, initializes γ^{*} to be an infinite number, and initially sets $N_r = 1$ and $\mathscr{g} = 1$, where N_r is the number of MCTs to be installed in the robust model, and $\mathscr{g}$ iterates configuring the MFD. Following the GHA, the robust GHA also installs the MCTs one by one. At the beginning of the robust GHA, the MFD has at most one MCT ($N_r = 1$) so we set N_r as N_{max} in the GHA. In the Main Step, the robust GHA starts by configuring the MFD to have one MCT using the GHA. It calculates $P_{f_{\omega}}$ for each scenario $\omega \in \Omega$.

If $P_{f_{\omega}} < DE$, the robust GHA goes back to configure a new MFD using the GHA with $N_r + 1$ and if $P_{f_{\omega}} \geq DE$, it calculates $\frac{TC_{\omega}}{P_{f_{\omega}}}$ as $\mathcal{R}_{\omega}$. We also calculate the percentage deviation between $\mathcal{R}_{\omega}$ and $\mathcal{Z}_{\omega}^{*}$ as γ_{ω}, where $\mathcal{Z}_{\omega}^{*}$ is the minimum cost per watt for scenario $\omega \in \Omega$ obtained by the GHA. The robust GHA obtains γ by searching for the maximum percentage deviation value in γ_{ω}. To find the minimum of the maximum percentage deviation, we compare γ to γ^{*}. If $\gamma < \gamma^{*}$, then γ updates γ^{*} and the robust GHA goes back to configure the MFD using the GHA with $N_r + 1$. If $\gamma \geq \gamma^{*}$, the robust GHA continues to investigate the MFD by configuring the MFD with $N_r + 1$ to avoid staying in the local minimum. The robust GHA stops if $\gamma \geq \gamma^{*}$after K repetitions. The robust solution is γ^{*} and the robust MFD associated with $\gamma^{*}$is returned.

We use a similar example of a marine farm with 3×3 cells to illustrate the steps of the robust GHA. We use the four input ambiguous parameters with two extreme values from the case study (see subchapter 4.5) for a total of 16 scenarios. We set the DE to be 300,000 watts since the marine farm only has 3×3 cells. The robust GHA first configures the MFD to have at most one MCT using the GHA since we initialize $N_r = 1$ and then calculates ${P_f}_\omega$. Since one MCT cannot satisfy the DE for some scenarios, the robust GHA goes back to configure the MFD with two MTCs, and since two MCTs cannot satisfy the DE for some scenarios the robust GHA goes back to configure the MFD with three MCTs as shown in Fig. 4.11(a), where the DE is satisfied for all scenarios. The robust GHA then calculates $\mathcal{R}_\omega$, the corresponding cost per watt of the MFD with three MCTs for scenario $\omega \in \Omega$, followed by calculating γ_ω, the percentage deviation between $\mathcal{R}_\omega$ and $\mathcal{Z}_\omega^*$, as reported in Table 4.3 ($\mathcal{Z}_\omega^*$ is the minimum cost per watt for scenario $\omega \in \Omega$ obtained by the GHA). In Table 4.3, the maximum percentage deviation γ for the MFD with three MCTs is 10.11.

doi:10.6342/NTU202403713

Robust GHA. The algorithm for solving the robust optimization model

```
Initialization Step:
Set the empty marine farm with $J×K$ cells.
Let $\gamma^*$ be an infinite number.
Set $N_r = 1$ and $\mathscr{g} = 1$.
Go to Main Step.

Main Step:
Start.
Configure an MFD using the GHA.
For each scenario $\omega \in \Omega$, do
    Calculate $P_{f_\omega}$.
    If $P_{f_\omega} < DE$, then Go back to configure an MFD using the GHA with $N_r + 1$.
    Else Calculate $\frac{TC_\omega}{P_{f_\omega}}$ as $\mathcal{R}_\omega$. Calculate $\frac{\mathcal{R}_\omega - \mathcal{Z}_\omega^*}{\mathcal{Z}_\omega^*}$ as $\gamma_\omega$.
    End if
End for
$\gamma = \max_{\omega \in \Omega} \gamma_\omega$
If $\gamma < \gamma^*$, then $\gamma^* = \gamma$. Go back to configure an MFD using the GHA with $N_r + 1$.
Else
    If $\mathscr{g} < K$, then go back to configure an MFD using the GHA with $N_r + 1$ and $\mathscr{g} + 1$.
    Else $\gamma^*$ is the robust solution. Return the robust MFD associated with $\gamma^*$.
    End if
End if
Stop.
```

To find the minimum of the maximum percentage deviation, we compare γ to γ^*. Since $\gamma < \gamma^*$, γ updates γ^*, i.e., 10.11, and the robust GHA goes back to configure the MFD with four MCTs as shown in Fig. 4.11(b). Since the DE is satisfied for all scenarios, the robust GHA then calculates γ_ω and finds that γ for the MFD with four MCTs is 23.09 as reported in Table 4.3. Since $\gamma \geq \gamma^*$, γ^* remains 10.11. The MFD is the first MFD returning $\gamma \geq \gamma^*$ as shown in Fig. 4.11 (b). The robust GHA continues to configure the MFD with five MCTs as shown in Fig. 4.11(c), where γ for the MFD with five MCTs is 28.65 as reported in Table 4.3 so that $\gamma \geq \gamma^*$ and γ^* remains 10.11. The robust GHA continues to configure the MFD with six MCTs as shown in Fig. 4.11(d) and finds that γ for the MFD with six MCTs is 32.87 as reported in Table 4.3. Since $\gamma \geq \gamma^*$, γ^* remains

10.11. The robust GHA stops after three repetitions ($K = 3$) returning $\gamma \geq \gamma^*$. The robust solution γ^* is 10.11 and the robust MFD associated with γ^* is returned as in Fig. 4.11(a).

Table 4.3. Percentage deviations

Scenario	MFD with three MCTs	MFD with four MCTs	MFD with five MCTs	MFD with six MCTs
	γ_ω (%)	γ_ω (%)	γ_ω (%)	γ_ω (%)
1	10.10	16.63	17.71	18.64
2	10.11	16.63	17.72	18.64
3	10.11	16.63	17.71	18.64
4	10.10	16.63	17.70	18.63
5	10.10	23.09	28.63	32.87
6	10.11	23.09	28.65	32.87
7	10.11	23.09	28.64	32.86
8	10.10	23.09	28.63	32.86
9	2.19	8.25	9.25	10.11
10	0.00	5.93	6.91	7.75
11	0.00	5.93	6.91	7.75
12	0.00	5.93	6.91	7.75
13	2.19	14.24	19.39	23.31
14	0.00	11.80	16.83	20.67
15	0.00	11.80	16.83	20.67
16	0.00	11.80	16.83	20.67

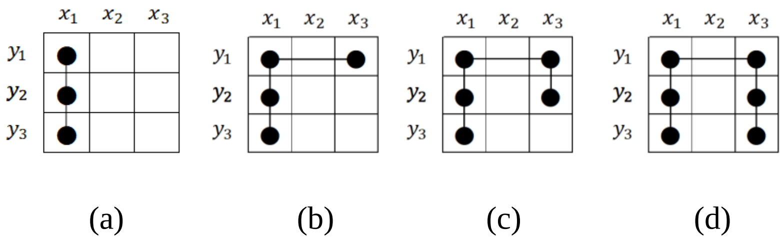
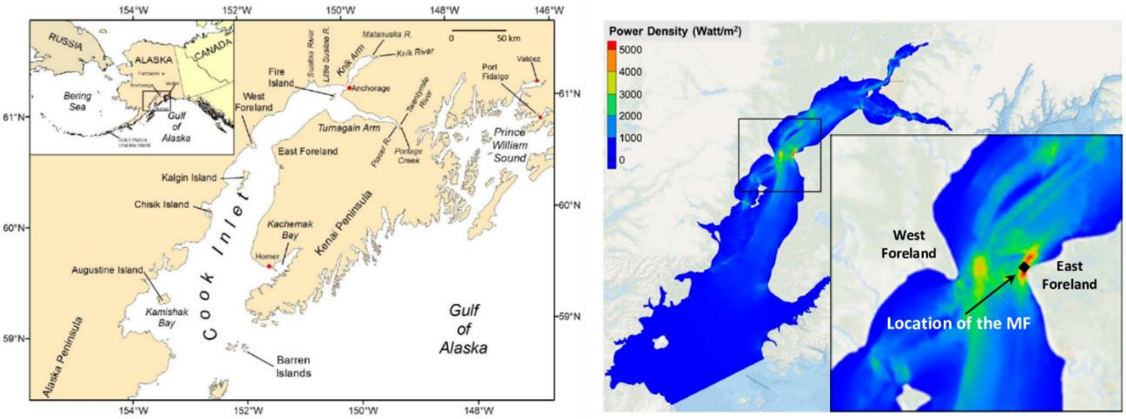


Fig. 4.11. Configure the MFD using the GHA

doi:10.6342/NTU202403713

4.5 Case Study

We validate the effectiveness of our solution approach with a case study of a hypothetical marine farm located in Cook Inlet, Alaska, as shown in Fig. 4.12. This area is characterized by substantial energy resources, suitable depths for installing MCTs, and proximity to Alaska’s industrial and population centers (Wang and Yang, 2020). Using power density location data from Wang and Yang (2020), we identify the site with the highest power density of Cook Inlet’s marine current at latitude 60.7° N and longitude 151.4° W, near East Foreland, where the average power density exceeds 5,000 watts per square meter (W/m²). In our setup, we position the MCTs 25 meters above the seabed, considering that the mean marine level at the site has a depth of 50 meters, as it experiences stronger currents compared to the bottom and surface layers (Wang & Yang, 2020).



(a) (b)

Fig. 4.12. (a) Cook Inlet and (b) Location of the hypothetical marine farm (adapted from Wang & Yang, 2020)

This study employs MCT parameters (cost, blade length, thrust coefficient) from Dai et al. (2017) and SPC cost per meter from Nieradzinska et al. (2016). Operation and

doi:10.6342/NTU202403713

maintenance cost,C_{om}, is assumed to be 2.5% of MCT cost, following Wu et al. (2020). Fixed values of these parameters are presented in Table 4.4.

Table 4.4. Fixed values parameters

Parameter	Value
Blade length in meter (r)	9
Thrust coefficient (C_T)	0.7
Cost of an MCT (C_a)	$799,712
SPC cost per meter (P_c)	$1,440
Operation and maintenance cost (C_{om})	$19,993

We employ a marine farm consisting of 10×10 cells (with N_{max} set to 100), following a similar configuration used by Ren et al. (2018). Each cell has a length and width of 10r or 90 meters, resulting in a total area of 810,000 square meters (or 9002 square meters). This setup ensures that d_{min} is 90 meters, allowing for adequate current recovery behind the upstream MCTs (Vennell et al., 2015). Based on the observations of Wang & Yang (2020), who note that the prevailing current flow generally occurs in a northwest direction, we position the MCTs perpendicular to this flow.

The four ambiguous parameters are represented by interval values as listed in Table 4.5. The inflow current velocity (v_0) at the site is measured by the US National Oceanic and Atmospheric Administration (NOAA) using Acoustic Doppler Current Profiler, where the current velocity typically ranges from 1.4 meters per second (m/s) to 3 m/s (Wang & Yang, 2020). Hence, the range of inflow current velocity is 1.4 m/s to 3 m/s. Specific datasets for turbulence intensity (I_0) and the density of marine water (ρ) at the hypothetical location are unavailable, so we use global data as a representation. Milne et al. (2013) indicate that turbulence intensity in the marine environment may vary from 6.6% to 15% using the Acoustic Doppler Velocimeter. Safarov et al. (2013) find that the density of ocean water ranges from 870.4 to 1099.5 kilograms per cubic meter (kg/m^3) using the Anton-Paar 4500 density meter, based on over 7,000 experiments conducted at various values of marine salinity, pressure, and temperature. Li (2014), who defines the power coefficient of the MCT as the ratio between the actual power output and a certain reference power output determined by the characteristics of the MCT and quantifies the power coefficient of different MCTs, observes that the power coefficient of the MCT varies from 38% to 49%.

Table 4.5 Range of ambiguous parameters

Parameter	Value
Inflow current velocity (v_0)	$1.4 - 3$ m/s
Turbulence intensity (I_0)	$6.6\% - 15\%$
Density of the marine water (ρ)	$870.4 - 1099.5\ \mathrm{kg/m^3}$
Power coefficient of the MCT (C_p)	$38\% - 49\%$

In a scenario-based approach, the ambiguous parameters' continuous values within an interval can result in an infinite number of distinct scenarios. To manage this, a set of extreme scenarios, incorporating the minimum and maximum values within the interval, is employed (Kasperski & Zieliński, 2016). Consequently, with four ambiguous parameters having two extreme values, there are 2^4 or 16 scenarios generated. For this study, the DE is established at 5,000,000 watts (5 MW).

The GHA yields solutions for these 16 scenarios, as documented in Table 4.6. Only six solutions are common across all scenarios: scenarios 1, 2, 3, 4, 9, 10, and 11 each have 45 MCTs; scenarios 5, 6, 7, 8, and 13 each have 31 MCTs; and scenarios 12, 14, 15, and 16 each have 47, 42, 41, and 68 MCTs, respectively. Notably, the inflow current velocity exhibits the most substantial impact on the objective function values; when set to 1.4 m/s

doi:10.6342/NTU202403713

in scenarios 9 through 16, the cost per watt tends to rise. This underscores the inflow current velocity's critical role in the marine farm's efficiency.

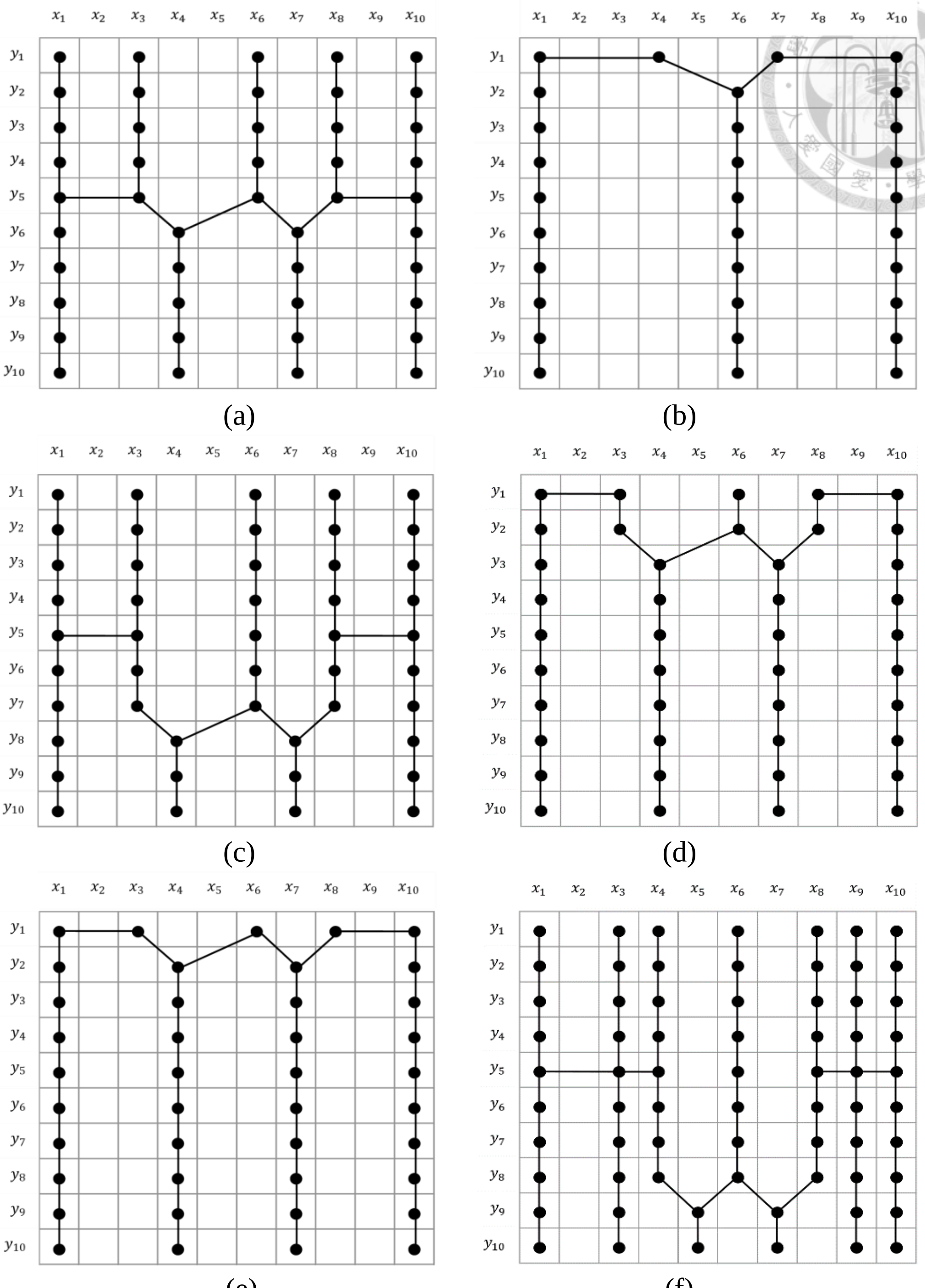
Table 4.6. Solutions of the BO-MCTIP model for each scenario

Scenario	Ambiguous Parameters				Solutions	
	v_0	I_0	ρ	C_p	N	$\mathcal{Z}_{\omega}^{*}$
1	3	15%	1099.5	49%	45	0.4014
2	3	15%	1099.5	38%	45	0.5176
3	3	15%	870.4	49%	45	0.5071
4	3	15%	870.4	38%	45	0.6539
5	3	6.6%	1099.5	49%	31	0.4574
6	3	6.6%	1099.5	38%	31	0.5898
7	3	6.6%	870.4	49%	31	0.5777
8	3	6.6%	870.4	38%	31	0.7450
9	1.4	15%	1099.5	49%	45	3.9500
10	1.4	15%	1099.5	38%	45	5.0935
11	1.4	15%	870.4	49%	45	4.9897
12	1.4	15%	870.4	38%	47	6.4370
13	1.4	6.6%	1099.5	49%	31	4.5006
14	1.4	6.6%	1099.5	38%	42	5.9216
15	1.4	6.6%	870.4	49%	41	5.7622
16	1.4	6.6%	870.4	38%	68	9.3146

Six MFD layouts are depicted in Fig. 4.13. MFD1, representing scenarios 1, 2, 3, 4, 9, 10, and 11, is shown in Fig. 4.13(a); MFD2, representing scenarios 5, 6, 7, 8, and 13, is shown in Fig. 4.13(b); while Figs. 4.13(c), 4.13(d), 4.13(e), and 4.13 (f) illustrate MFD3, MFD4, MFD5, and MFD6, respectively, corresponding to scenarios 12, 14, 15, and 16. These figures reveal that the distance between an upstream MCT and a downstream MCT exceeds the minimum distance, d_{min}, of 90 meters, suggesting that mitigating wake effects is more beneficial than overcrowding the farm with MCTs.

Additionally, the case study validates the robust optimization model. Table 4.7 reports the maximum relative regret, γ, where certain scenarios in MFD1 to MFD5 fail to meet the DE (refer to Table 4.8 – 4.12). Conversely, all scenarios in MFD6 meet the DE, as shown in Table 4.13. The robust GHA identifies the robust solution with the minimum of the maximum relative regret at 27.07%, achieved by installing 68 MCTs, feasible for all scenarios. Fig. 4.14 illustrates the layout of the MFD for the robust optimization model, termed the robust MFD.

doi:10.6342/NTU202403713


x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10}
y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8 y_9 y_{10}

(a) (b)

(c) (d)

(e) (f)

Fig. 4.13. MFDs for the 16 scenarios

doi:10.6342/NTU202403713

Table 4.7. Maximum relative regret

MFD	N	γ (%)	Feasibility for all scenarios
MFD1	45	4.09	Not feasible
MFD2	31	4.39	Not feasible
MFD3	47	5.45	Not feasible
MFD4	42	2.04	Not feasible
MFD5	41	1.37	Not feasible
MFD6	68	27.07	Feasible

Table 4.8. Percentage deviations between MFD1 and BO-MCTIP solutions for each scenario

Scenario	MFD1		BO-MCTIP Solution		% Deviation
	N	$\mathcal{R}_\omega$	N	$\mathcal{Z}_\omega^*$	$\frac{\mathcal{R}_\omega - \mathcal{Z}_\omega^*}{\mathcal{Z}_\omega^*}$
1	45	0.4014	45	0.4014	0.00
2	45	0.5176	45	0.5176	0.00
3	45	0.5071	45	0.5071	0.00
4	45	0.6539	45	0.6539	0.00
5	45	0.4760	31	0.4574	4.07
6	45	0.6138	31	0.5898	4.07
7	45	0.6013	31	0.5777	4.09
8	45	0.7754	31	0.7450	4.08
9	45	3.9500	45	3.9500	0.00
10	45	5.0935	45	5.0935	0.00
11	45	4.9897	45	4.9897	0.00
12	45	Cannot satisfy the DE	47	6.4370	-
13	45	4.6842	31	4.5006	4.08
14	45	6.0401	42	5.9216	2.00
15	45	5.9171	41	5.7622	2.69
16	45	Cannot satisfy the DE	68	9.3146	-

doi:10.6342/NTU202403713

Table 4.9. Percentage deviations between MFD2 and BO-MCTIP solutions for each scenario

Scenario	MFD2		BO-MCTIP Solution		% Deviation
	N	$\mathcal{R}_\omega$	N	$\mathcal{Z}_\omega^*$	$\frac{\mathcal{R}_\omega - \mathcal{Z}_\omega^*}{\mathcal{Z}_\omega^*}$
1	31	0.4190	45	0.4014	4.38
2	31	0.5403	45	0.5176	4.39
3	31	0.5293	45	0.5071	4.38
4	31	0.6825	45	0.6539	4.37
5	31	0.4574	31	0.4574	0.00
6	31	0.5898	31	0.5898	0.00
7	31	0.5777	31	0.5777	0.00
8	31	0.7450	31	0.7450	0.00
9	31	4.1232	45	3.9500	4.38
10	31	Cannot satisfy the DE	45	5.0935	-
11	31	Cannot satisfy the DE	45	4.9897	-
12	31	Cannot satisfy the DE	47	6.4370	-
13	31	4.5006	31	4.5006	0.00
14	31	Cannot satisfy the DE	42	5.9216	-
15	31	Cannot satisfy the DE	41	5.7622	-
16	31	Cannot satisfy the DE	68	9.3146	-

doi:10.6342/NTU202403713

Table 4.10. Percentage deviations between MFD3 and BO-MCTIP solutions for each scenario

Scenario	MFD3		BO-MCTIP Solution		% Deviation
	N	$\mathcal{R}_\omega$	N	$\mathcal{Z}_\omega^*$	$\frac{\mathcal{R}_\omega - \mathcal{Z}_\omega^*}{\mathcal{Z}_\omega^*}$
1	47	0.4017	45	0.4014	0.05
2	47	0.5180	45	0.5176	0.04
3	47	0.5074	45	0.5071	0.04
4	47	0.6543	45	0.6539	0.03
5	47	0.4945	31	0.4574	5.42
6	47	0.6376	31	0.5898	5.43
7	47	0.6246	31	0.5777	5.45
8	47	0.8054	31	0.7450	5.44
9	47	3.9526	45	3.9500	0.05
10	47	5.0968	45	5.0935	0.05
11	47	4.9930	45	4.9897	0.05
12	47	6.4383	47	6.4370	0.00
13	47	4.8658	31	4.5006	5.43
14	47	6.2743	42	5.9216	3.33
15	47	6.1465	41	5.7622	4.02
16	47	Cannot satisfy the DE	68	9.3146	-

Table 4.11. Percentage deviations between MFD4 and BO-MCTIP solutions for each scenario

Scenario	MFD4		BO-MCTIP Solution		% Deviation
	N	$\mathcal{R}_\omega$	N	$\mathcal{Z}_\omega^*$	$\frac{\mathcal{R}_\omega - \mathcal{Z}_\omega^*}{\mathcal{Z}_\omega^*}$
1	42	0.4016	45	0.4014	0.05
2	42	0.5179	45	0.5176	0.06
3	42	0.5073	45	0.5071	0.04
4	42	0.6542	45	0.6539	0.05
5	42	0.4667	31	0.4574	2.03
6	42	0.6018	31	0.5898	2.03
7	42	0.5895	31	0.5777	2.04
8	42	0.7602	31	0.7450	2.04
9	42	3.9522	45	3.9500	0.06
10	42	5.0962	45	5.0935	0.05
11	42	4.9924	45	4.9897	0.05
12	42	Cannot satisfy the DE	47	6.4370	-
13	42	4.5922	31	4.5006	2.04
14	42	5.9216	42	5.9216	0.00
15	42	5.8010	41	5.7622	0.67
16	42	Cannot satisfy the DE	68	9.3146	-

doi:10.6342/NTU202403713

Table 4.12. Percentage deviations between MFD5 and BO-MCTIP solutions for each scenario

Scenario	MFD5		BO-MCTIP Solution		% Deviation
	N	$\mathcal{R}_\omega$	N	$\mathcal{Z}_\omega^*$	$\frac{\mathcal{R}_\omega - \mathcal{Z}_\omega^*}{\mathcal{Z}_\omega^*}$
1	41	0.4018	45	0.4014	0.10
2	41	0.5182	45	0.5176	0.12
3	41	0.5076	45	0.5071	0.10
4	41	0.6546	45	0.6539	0.11
5	41	0.4635	31	0.4574	1.33
6	41	0.5977	31	0.5898	1.34
7	41	0.5856	31	0.5777	1.37
8	41	0.7551	31	0.7450	1.36
9	41	3.9544	45	3.9500	0.11
10	41	5.0992	45	5.0935	0.11
11	41	4.9953	45	4.9897	0.11
12	41	Cannot satisfy the DE	47	6.4370	-
13	41	4.5616	31	4.5006	1.36
14	41	Cannot satisfy the DE	42	5.9216	-
15	41	5.7622	41	5.7622	0.00
16	41	Cannot satisfy the DE	68	9.3146	-

doi:10.6342/NTU202403713

Table 4.13. Percentage deviations between MFD6 and BO-MCTIP solutions for each scenario

Scenario	MFD6			BO-MCTIP Solution		% Deviation
	N	$\mathcal{R}_\omega$	P_f (MW)	N	$\mathcal{Z}_\omega^*$	$\frac{\mathcal{R}_\omega - \mathcal{Z}_\omega^*}{\mathcal{Z}_\omega^*}$
1	68	0.4155	112.16	45	0.4014	3.51
2	68	0.5357	87.06	45	0.5176	3.49
3	68	0.5248	88.87	45	0.5071	3.49
4	68	0.6768	68.92	45	0.6539	3.50
5	68	0.5811	80.26	31	0.4574	27.04
6	68	0.7493	62.25	31	0.5898	27.04
7	68	0.7341	63.54	31	0.5777	27.07
8	68	0.9466	49.28	31	0.7450	27.06
9	68	4.0885	11.41	45	3.9500	3.51
10	68	5.272	8.85	45	5.0935	3.50
11	68	5.1646	9.03	45	4.9897	3.501
12	68	6.6596	7.01	51	6.4383	3.44
13	68	5.7184	8.16	31	4.5006	27.06
14	68	7.3737	6.33	42	5.9216	24.52
15	68	7.2236	6.48	41	5.7622	25.36
16	68	9.3146	5.01	68	9.3146	0.00

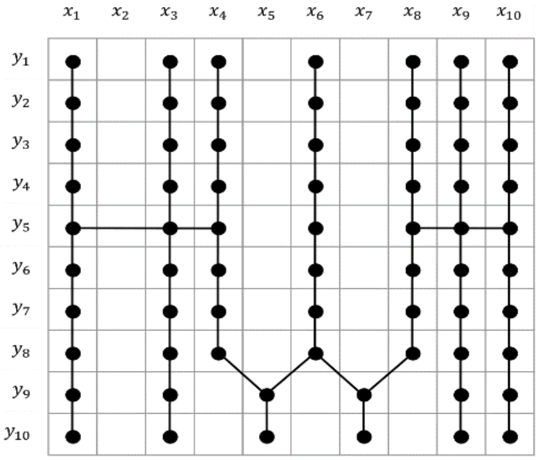


Fig. 4.14. Robust MFD

doi:10.6342/NTU202403713

These findings emphasize that meeting the minimum power requirement is crucial for preventing potential blackouts (Staffell & Pfenninger, 2018), with our robust solution approach ensuring this minimum requirement is satisfied.

4.6 Summaries

In this study, we introduce a BO-MCTIP model to determine the optimal number and locations of MCTs in a hypothetical marine farm. The upper-level model considers wake effects, which can reduce power generation, while the lower-level model focuses on the SPC connection cost of the installed MCTs. Our proposed GHA includes a finite enumeration step to account for wake effects and an installation step that strategically places MCTs in adjacent rows to minimize SPC connection costs.

We develop a non-parametric, scenario-based robust optimization problem to find a solution that meets a minimum power requirement and minimizes expected regret across 16 possible scenarios. A case study of a hypothetical marine farm in Cook Inlet, Alaska, validates our approach. The results indicate that MFD generally requires larger clearance distances among MCTs, and appropriate spacing can reduce the cost per watt. Additionally, marine current velocity significantly impacts MFD performance.

Future research could explore relaxing the homogenous features of cell sizes and depths. Moreover, considering a variable SPC connection cost based on the locations of installed MCTs could be an interesting direction for further investigation.

doi:10.6342/NTU202403713

Chapter 5 Conclusions

This study demonstrates the application of data-driven optimization techniques to address three specific industrial challenges: queue-time loop scheduling, turbulence-parameter calibration for marine turbines, and marine turbine layout optimization. The findings are as follows:

1. Queue-Time Loop Scheduling: To manage the dynamic nature of queue-time loops in flow shop scheduling, we propose the Phase-Step (PS) method in combination with Combinatorial Benders' Cut (CBC). This combination significantly outperforms traditional heuristics like First-In-First-Out (FIFO), Threshold Dispatching (TH), and Reaction Chains (RC), particularly under high job arrival rates, effectively reducing the number of scrap jobs.
2. Turbulence-Parameter Calibration for Marine Turbines: We employ artificial neural networks (ANN) coupled with optimization techniques such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) for turbulence-parameter calibration. Our approach achieves notable reductions in the final error, bridging the gap between simulation and experimental results by refining calibration strategies.
3. Marine Turbine Layout Optimization: For the marine turbine layout problem, we introduce the Greedy Heuristic Algorithm (GHA) integrated with Kruskal's algorithm. This method effectively minimizes power generation reduction due to wake effects and reduces installation costs. Validation through a scenario-based robust optimization approach highlights the importance of turbine spacing and the impact of marine current velocity on performance.

doi:10.6342/NTU202403713

The study showcases the efficacy of data-driven optimization techniques in tackling industrial problems, demonstrating superior performance compared to alternative methods. It underscores the broad applicability of these methodologies in industrial problem-solving due to their ability to provide effective solutions. Furthermore, the research emphasizes the practical implications of these methodologies in real-world applications, providing a comprehensive framework for addressing complex industrial challenges. The successful implementation of these techniques not only improves efficiency but also paves the way for future innovations in the field, reinforcing the potential for continued advancements and optimization in industrial practices.

References

Abbassi, M., Chaabani, A., & Said, L. B. (2022). An efficient chemical reaction algorithm for multi-objective combinatorial bi-level optimization. Engineering Optimization, 54(4), 665-686.

Abdelwahhab, M. A., Abdelhafez, N. A., & Embabi, A. M. (2023). Machine learning-supported seismic stratigraphy of the Paleozoic Nubia Formation (SW Gulf of Suez-rift): implications for paleoenvironment− petroleum geology of a lacustrine-fan delta. Petroleum, 9(2), 301-315.

Abdullah, F. A. R., Ningsih, N. S., & Al-Khan, T. M. (2022). Significant wave height forecasting using long short-term memory neural network in Indonesian waters. Journal of Ocean Engineering and Marine Energy, 8(2), 183-192.

Addis, B., Carello, G., Grosso, A., & Tanfani, E. (2016). Operating room scheduling and rescheduling: a rolling horizon approach. Flexible Services and Manufacturing Journal, 28(1–2), 206–232.

Agarwal, P., Sharma, A., & Kumar, N. (2022). A Soft-Computing Approach to Fuzzy EOQ Model for Deteriorating Items with Partial Backlogging. Fuzzy Information and Engineering, 1-15.

Akkerman, R., Van Donk, D. P., & Gaalman, G. (2007). Influence of capacity-and time-constrained intermediate storage in two-stage food production systems. International Journal of Production Research, 45(13), 2955–2973.

Akpinar, S., Elmi, A., & Bektas, T. (2017). Combinatorial Benders cuts for assembly line balancing problems with setups. European Journal of Operational Research, 259(2), 527–537.

doi:10.6342/NTU202403713

Amiri-Ramsheh, B., Safaei-Farouji, M., Larestani, A., Zabihi, R., & Hemmati-Sarapardeh, A. (2022). Modeling of wax disappearance temperature (WDT) using soft computing approaches: Tree-based models and hybrid models. Journal of Petroleum Science and Engineering, 208, 109774.

Bahaj, A. S. 2011. Generating electricity from the oceans. Renewable and Sustainable Energy Reviews, 15(7), 3399-3416.

Ben-Tal, A., & Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data. Mathematical Programming, 88(3), 411-424.

Bernier, V., & Frein, Y. (2004). Local scheduling problems submitted to global FIFO processing constraints. International Journal of Production Research, 42(8), 1483–1503.

Bracken, J., & McGill, J. T. (1973). Mathematical programs with optimization problems in the constraints. Operations Research, 21(1), 37-44.

Brunetti, A. N. N. A., Armenio, V., & Roman, F. (2019). Large eddy simulation of a marine turbine in a stable stratified flow condition. Journal of Ocean Engineering and Marine Energy, 5, 1-19.

Brunton, S. L., Nathan Kutz, J., Manohar, K., Aravkin, A. Y., Morgansen, K., Klemisch, J., ... & McDonald, D. (2021). Data-driven aerospace engineering: reframing the industry with machine learning. AIAA Journal, 59(8), 2820-2847.

Brutto, O. A. L., Thiébot, J., Guillou, S. S., & Gualous, H. (2016). A semi-analytic method to optimize tidal farm layouts–Application to the Alderney Race (Raz Blanchard), France. Applied Energy, 183, 1168-1180.

doi:10.6342/NTU202403713

Bryden, I. G., Couch, S. J., Owen, A., & Melville, G. (2007). Tidal current resource assessment. Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy, 221(2), 125-135.

Chen, C., Yaari, Z., Apfelbaum, E., Grodzinski, P., Shamay, Y., & Heller, D. A. (2022). Merging Data Curation and Machine Learning to Improve Nanomedicines. Advanced Drug Delivery Reviews, 114172.

Chen, J. H., Lee, D. H., & Cao, J. X. (2012). A combinatorial benders' cuts algorithm for the quayside operation problem at container terminals. Transportation Research Part E: Logistics and Transportation Review, 48(1), 266–275.

Cho, S., Kim, M., Lyu, B., & Moon, I. (2021). Optimization of an explosive waste incinerator via an artificial neural network surrogate model. Chemical Engineering Journal, 407, 126659.

Chugh, T., Sindhya, K., Hakanen, J., & Miettinen, K. (2019). A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. Soft Computing, 23, 3137-3166.

Chugh, T., Sindhya, K., Miettinen, K., Jin, Y., Kratky, T., & Makkonen, P. (2017, June). Surrogate-assisted evolutionary multiobjective shape optimization of an air intake ventilation system. In 2017 IEEE Congress on Evolutionary Computation (CEC) (pp. 1541-1548). IEEE.

Codato, G., & Fischetti, M. (2006). Combinatorial Benders' cuts for mixed-integer linear programming. Operations Research, 54(4), 756–766.

doi:10.6342/NTU202403713

Dai, Y., Ren, Z., Wang, K., Li, W., Li, Z., & Yan, W. (2017). Optimal sizing and arrangement of tidal current farm. IEEE Transactions on Sustainable Energy, 9(1), 168-177.

Dang, J. F., & Hong, I. H. (2013). The equilibrium quantity and production strategy in a fuzzy random decision environment: Game approach and case study in glass substrates industries. International Journal of Production Economics, 145(2), 724-732.

Ding, S., Su, C., & Yu, J. (2011). An optimizing BP neural network algorithm based on genetic algorithm. Artificial intelligence review, 36, 153-162.

Dosi, C., Iori, M., Kramer, A., & Vignoli, M. (2023). Successful implementation of discrete event simulation: integrating design thinking and simulation approach in an emergency department. Production Planning & Control, 34(13), 1233-1247.

Elsayed, K., & Lacor, C. (2013). CFD modeling and multi-objective optimization of cyclone geometry using desirability function, artificial neural networks and genetic algorithms. Applied Mathematical Modelling, 37(8), 5680-5704.

Fakhri, E., Thiebot, J., Bourguet, S., Machmoum, M., & Gualous, H. (2020, June). Optimization of Marine Current Turbines Arrangement Using a Genetic AlgorithmApplication to the Fromveur Strait (France). In 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE) (pp. 1057-1061). IEEE.

Fang, J., & Xi, Y. (1997). A rolling horizon job shop rescheduling strategy in the dynamic environment. The International Journal of Advanced Manufacturing Technology, 13(3), 227–232.

doi:10.6342/NTU202403713

Forbush, D., Polagye, B., Thomson, J., Kilcher, L., Donegan, J., & McEntee, J. (2016). Performance characterization of a cross-flow hydrokinetic turbine in sheared inflow. International Journal of Marine Energy, 16, 150-161.

Gaurier, B., Carlier, C., Germain, G., Pinon, G., & Rivoalen, E. (2020). Three tidal turbines in interaction: An experimental study of turbulence intensity effects on wakes and turbine performance. Renewable Energy, 148, 1150-1164.

Gentils, T., Wang, L., & Kolios, A. (2017). Integrated structural optimisation of offshore wind turbine support structures based on finite element analysis and genetic algorithm. Applied energy, 199, 187-204.

Georgiadis, G. P., Elekidis, A. P., & Georgiadis, M. C. (2019). Optimization-based scheduling for the process industries: from theory to real-life industrial applications. Processes, 7(7), 438.

Gholami, A., Bonakdari, H., Zaji, A. H., & Akhtari, A. A. (2015). Simulation of open channel bend characteristics using computational fluid dynamics and artificial neural networks. Engineering Applications of Computational Fluid Mechanics, 9(1), 355-369.

Gorissen, B. L., Yanıkoğlu, İ., & den Hertog, D. 2015. A practical guide to robust optimization. Omega, 53, 124-137.

Grillone, B., Danov, S., Sumper, A., Cipriano, J., & Mor, G. (2020). A review of deterministic and data-driven methods to quantify energy efficiency savings and to predict retrofitting scenarios in buildings. Renewable and Sustainable Energy Reviews, 131, 110027.

doi:10.6342/NTU202403713

Guo, S., Agarwal, M., Cooper, C., Tian, Q., Gao, R. X., Grace, W. G., & Guo, Y. B. (2022). Machine learning for metal additive manufacturing: Towards a physics-informed data-driven paradigm. Journal of Manufacturing Systems, 62, 145-163.

Hamdia, K. M., Zhuang, X., & Rabczuk, T. (2021). An efficient optimization approach for designing machine learning models based on genetic algorithm. Neural Computing and Applications, 33(6), 1923-1933.

Harjunkoski, I., & Grossmann, I. E. (2001). A decomposition approach for the scheduling of a steel plant production. Computers & Chemical Engineering, 25(11–12), 1647–1660.

Hodson, A., Muhlemann, A. P., & Price, D. H. R. (1985). A microcomputer based solution to a practical scheduling problem. Journal of the Operational Research Society, 36(10), 903–914.

Holland, J. H. (1992). Genetic algorithms. Scientific American, 267(1), 66-73.

Hong, I. H., Chou, C. C., & Lee, P. K. (2019). Admission control in queue-time loop production-Mixed integer programming with Lagrangian relaxation (MIPLAR). Computers & Industrial Engineering, 129, 417–425.

Hsieh, Y. L., Yu, H. C., & Yu, M. S. (2002). Process Q-time management and yield analysis system. In In Taiwan Semiconductor Manufacturing Company e-operations Symposium.

Hu, Z., Du, X., Kolekar, N.S. and Banerjee, A., 2014. Robust design with imprecise random variables and its application in hydrokinetic turbine optimization. Engineering Optimization, 46(3), 393-419.

IBM-ILOG (2014). Cplex optimization studio.

James, S. C., Johnson, E. L., Barco, J., & Roberts, J. D. (2020). Simulating current-energy converters: SNL-EFDC model development, verification, and parameter estimation. Renewable Energy, 147, 2531-2541.

Jensen, N. 1983. A note on wind turbine interaction Technical report Ris-M-2411. Roskilde (Denmark): Risø National Laboratory.

Ji, X., Lu, X., Li, H., Ma, P., & Xu, S. (2023). Blade optimization design of Savonius hydraulic turbine based on radial basis function surrogate model and L-SHADE algorithm. Ocean Engineering, 286, 115620.

Jin, Y., & Branke, J. (2005). Evolutionary optimization in uncertain environments-a survey. IEEE Transactions on evolutionary computation, 9(3), 303-317.

Jin, Y., & Sendhoff, B. (2009). A systems approach to evolutionary multiobjective structural optimization and beyond. IEEE Computational Intelligence Magazine, 4(3), 62-76.

Jin, Y., Wang, H., Chugh, T., Guo, D., & Miettinen, K. (2018). Data-driven evolutionary optimization: An overview and case studies. IEEE Transactions on Evolutionary Computation, 23(3), 442-458.

Jónsdóttir, G. M., & Milano, F. 2019. Modeling of Short-Term Tidal Power Fluctuations. IEEE Transactions on Sustainable Energy, 11(4), 2337-2344.

Joo, B. J., & Kim, Y. D. (2009). A branch-and-bound algorithm for a two-machine flowshop scheduling problem with limited waiting time constraints. Journal of the Operational Research Society, 60, 572–582.

Kadem, M., Garber, L., Abdelkhalek, M., Al-Khazraji, B. K., & Keshavarz-Motamed, Z. (2022). Hemodynamic modeling, medical imaging, and machine learning and their

doi:10.6342/NTU202403713

applications to cardiovascular interventions. IEEE Reviews in Biomedical Engineering, 16, 403-423.

Kamran, M. A., Karimi, B., Dellaert, N., & Demeulemeester, E. (2019). Adaptive operating rooms planning and scheduling: A rolling horizon approach. Operations Research for Health Care, 22, 100200.

Kasperski, A., & Zieliński, P. (2016). Robust discrete optimization under discrete and interval uncertainty: A survey. In Robustness Analysis in Decision Aiding, Optimization, and Analytics (pp. 113-143). Springer, Cham.

Katul, G. G., Mahrt, L., Poggi, D., & Sanz, C. (2004). One-and two-equation models for canopy turbulence. Boundary-layer meteorology, 113, 81-109.

Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In Proceedings of the IEEE international conference on neural networks (Vol. 4, pp. 1942-1948).

Khan, H., Kushwah, K. K., Maurya, M. R., Singh, S., Jha, P., Mahobia, S. K., ... & Sadasivuni, K. K. (2022). Machine learning driven intelligent and self adaptive system for traffic management in smart cities. Computing, 1-15.

Kim, S. H., & Boukouvala, F. (2020). Machine learning-based surrogate modeling for data-driven optimization: a comparison of subset selection for regression techniques. Optimization Letters, 14(4), 989-1010.

Kouvelis, P., & Yu, G. 2013. Robust Discrete Optimization and Its Applications (Vol. 14). Springer Science & Business Media.

Koziel, S., & Pietrenko-Dabrowska, A. (2020). Design-oriented computationally-efficient feature-based surrogate modelling of multi-band antennas with nested

doi:10.6342/NTU202403713

kriging. AEU-International Journal of Electronics and Communications, 120, 153202.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical Society, 7(1), 48-50.

Lee, Y.Y., Chen, C.T., & Wu, C. (2005, September). Reaction chain of process queue time quality control. In ISSM 2005, IEEE International Symposium on Semiconductor Manufacturing, 2005. (pp. 47–50). IEEE.

Li, Y. (2014). On the definition of the power coefficient of tidal current turbines and efficiency of tidal current turbine farms. Renewable Energy, 68, 868-875.

Lira, J. O., Riella, H. G., Padoin, N., & Soares, C. (2022). Computational fluid dynamics (CFD), artificial neural network (ANN) and genetic algorithm (GA) as a hybrid method for the analysis and optimization of micro-photocatalytic reactors: NOx abatement as a case study. Chemical engineering journal, 431, 133771.

Liu, Y., Shang, F., Jiao, L., Cheng, J., & Cheng, H. (2014). Trace norm regularized CANDECOMP/PARAFAC decomposition with missing data. IEEE transactions on cybernetics, 45(11), 2437-2448.

Liu, Z., Lesselier, D., Sudret, B., & Wiart, J. (2020). Surrogate modeling based on resampled polynomial chaos expansions. Reliability Engineering & System Safety, 202, 107008.

Liu, Z., & Pu, J. (2022). Analysis and research on intelligent manufacturing medical product design and intelligent hospital system dynamics based on machine learning under big data. Enterprise Information Systems, 16(2), 193-207.

doi:10.6342/NTU202403713

Lv, Y., Zhang, J., & Qin, W. (2017). A genetic regulatory network-based method for dynamic hybrid flow shop scheduling with uncertain processing times. Applied Sciences, 7(1), 23.

Madathil, S. C., Nambiar, S., Mason, S. J., & Kurz, M. E. (2018). On scheduling a photolithography process containing cluster tools. Computers & Industrial Engineering, 121, 177–188.

Mandl, C., & Minner, S. (2023). Data-driven optimization for commodity procurement under price uncertainty. Manufacturing & Service Operations Management, 25(2), 371-390.

Milne, I. A., Sharma, R. N., Flay, R. G., & Bickerton, S. (2013). Characteristics of the turbulence in the flow at a tidal stream power site. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 371(1985), 20120196.

Mittal, P., Kulkarni, K., & Mitra, K. (2016). A novel hybrid optimization methodology to optimize the total number and placement of wind turbines. Renewable Energy, 86, 133-147.

Monch, L., Fowler, J. W., Dauzere-Peres, S., Mason, S. J., & Rose, O. (2011). A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. Journal of Scheduling, 14(6), 583–599.

Mosetti, G. P. C. D. B., Poloni, C., & Diviacco, B. (1994). Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm. Journal of Wind Engineering and Industrial Aerodynamics, 51(1), 105-116.

doi:10.6342/NTU202403713

Mycek, P., Gaurier, B., Germain, G., Pinon, G., & Rivoalen, E. (2014). Experimental study of the turbulence intensity effects on marine current turbines behaviour. Part I: One single turbine. Renewable Energy, 66, 729-746.

Nag, A. K., & Sarkar, S. (2020). Experimental and numerical study on the performance and flow pattern of different Savonius hydrokinetic turbines with varying duct angle. Journal of Ocean Engineering and Marine Energy, 6, 31-53.

Nastro, F., Sorrentino, M., & Trifirò, A. (2022). A machine learning approach based on neural networks for energy diagnosis of telecommunication sites. Energy, 245, 123266.

Nieradzinska, K., MacIver, C., Gill, S., Agnew, G. A., Anaya-Lara, O., & Bell, K. R. W. (2016). Optioneering analysis for connecting Dogger Bank offshore wind farms to the GB electricity network. Renewable Energy, 91, 120-129.

Nuernberg, M., & Tao, L. (2018). Experimental study of wake characteristics in tidal turbine arrays. Renewable energy, 127, 168-181.

Nurdiansyah, R., & Hong, I. H. (2018, August). Combinatorial Benders' Cut for the Admission Control Decision in Flow Shop Scheduling Problems with Queue Time Constraints. In In IFIP International Conference on Advances in Production Management Systems (pp. 399–405). Cham: Springer.

Nurdiansyah, R., Hong, I. H., & Lee, P. K. (2021). Real-time admission control in a queue-time loop production system. Computers & Industrial Engineering, 161, 107617.

Nurdiansyah, R., Su, J. C., Hong, I. H., Olson, S. S., & Silva III, H. (2023). A surrogate model-based framework to calibrate the turbulence parameters of a vegetative

doi:10.6342/NTU202403713

canopy model for a marine turbine simulation. Journal of Ocean Engineering and Marine Energy, 9(3), 531-545.

Nurdiansyah, R., Hong, I. H., & Su, J. C. (2024). An Optimization Model for Marine Current Turbines Installation Problem against Parameters Ambiguity. Submitted to Engineering Optimization (under second round review).

Olson, S. (2019). Gaussian process regression applied to marine energy turbulent source tuning via metamodel machine learning optimization. https://digitalrepository.unm.edu/me_etds/165.

Olson, S. S., Su, J. C., Silva III, H., Chartrand, C. C., & Roberts, J. D. (2021). Turbulence-parameter estimation for current-energy converters using surrogate model optimization. Renewable Energy, 168, 559-567.

Olu-Ajayi, R., Alaka, H., Sulaimon, I., Sunmola, F., & Ajayi, S. (2022). Building energy consumption prediction for residential buildings using deep learning and other machine learning techniques. Journal of Building Engineering, 45, 103406.

Papadopoulos, V., Soimiris, G., Giovanis, D. G., & Papadrakakis, M. (2018). A neural network-based surrogate model for carbon nanotubes with geometric nonlinearities. Computer Methods in Applied Mechanics and Engineering, 328, 411-430.

Pichi, F., Ballarin, F., Rozza, G., & Hesthaven, J. S. (2023). An artificial neural network approach to bifurcating phenomena in computational fluid dynamics. Computers & Fluids, 254, 105813.

Pinon, G., Hurst, M. F., & Lukeba, E. (2017). Semi-analytical estimate of energy production from a tidal turbine farm with the account of ambient turbulence. International Journal of Marine Energy, 19, 70-82.

doi:10.6342/NTU202403713

Połap, D. (2020). An adaptive genetic algorithm as a supporting mechanism for microscopy image analysis in a cascade of convolution neural networks. Applied Soft Computing, 97, 106824.

Qin, W., Zhang, J., & Song, D. (2018). An improved ant colony algorithm for dynamic hybrid flow shop scheduling with uncertain processing time. Journal of Intelligent Manufacturing, 29(4), 891–904.

Quasim, M. T., Sulaiman, A., Shaikh, A., & Younus, M. (2022). Blockchain in Churn Prediction based Telecommunication System on Climatic Weather Application. Sustainable Computing: Informatics and Systems, 100705.

Rao, S., Xue, H., Bao, M., & Funke, S. (2016). Determining tidal turbine farm efficiency in the Western Passage using the disc actuator theory. Ocean Dynamics, 66(1), 41-57.

Ren, Z., Wang, Y., Li, H., Liu, X., Wen, Y., & Li, W. (2018). A coordinated planning method for micrositing of tidal current turbines and collector system optimization in tidal current farms. IEEE Transactions on Power Systems, 34(1), 292-302.

Réthoré, P. E., Sørensen, N. N., Bechmann, A., & Zhale, F. (2009, March). Study of the atmospheric wake turbulence of a CFD actuator disc model. In 2009 European wind energy conference and exhibition (pp. 1-9). EWEC.

Romer, M., & Mellouli, T. (2016). Future Demand Uncertainty in Personnel Scheduling: Investigating Deterministic Lookahead Policies Using Optimization and Simulation. In In Proceedings of the 30th European Conference on Modelling and Simulation (pp. 502–507).

doi:10.6342/NTU202403713

Rose, O. (1999). CONLOAD–a new lot release rule for semiconductor wafer fabs. In In Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future-Volume 1 (pp. 850–855). ACM.

Ruiz, R., & Maroto, C. (2005). A comprehensive review and evaluation of permutation flowshop heuristics. European Journal of Operational Research, 165(2), 479–494.

Safarov, J., Berndt, S., Millero, F. J., Feistel, R., Heintz, A., & Hassel, E. P. 2013. (p, ρ, T) Properties of seawater at brackish salinities: Extensions to high temperatures and pressures. Deep Sea Research Part I: Oceanographic Research Papers, 78, 95-101.

Santos, L. F., Costa, C. B., Caballero, J. A., & Ravagnani, M. A. (2023). Multi-objective simulation–optimization via kriging surrogate models applied to natural gas liquefaction process design. Energy, 262, 125271.

Scholl, W., & Domaschke, J. (2000). Implementation of modeling and simulation in semiconductor wafer fabrication with time constraints between wet etch and furnace operations. IEEE Transactions on Semiconductor Manufacturing, 13(3), 273–277.

Sedighizadeh, D., Masehian, E., Sedighizadeh, M., & Akbaripour, H. (2021). GEPSO: A new generalized particle swarm optimization algorithm. Mathematics and Computers in Simulation, 179, 194-212.

Sethi, S., & Sorger, G. (1991). A theory of rolling horizon decision making. Annals of Operations Research, 29(1), 387–415. Silver, E. A. (2004). An overview of heuristic solution methods. Journal of the operational research society, 55(9), 936–956.

doi:10.6342/NTU202403713

Seyedzadeh, S., Rahimian, F. P., Oliver, S., Glesk, I., & Kumar, B. (2020). Data driven model improved by multi-objective optimisation for prediction of building energy loads. Automation in Construction, 116, 103188.

Shafaei, R., & Brunn, P. (1999). Workshop scheduling using practical (inaccurate) data Part 1: The performance of heuristic scheduling rules in a dynamic job shop environment using a rolling time horizon approach. International Journal of Production Research, 37(17), 3913–3925.

Shen, F., Zhao, L., Du, W., Zhong, W., & Qian, F. (2020). Large-scale industrial energy systems optimization under uncertainty: A data-driven robust optimization approach. Applied Energy, 259, 114199.

Sindhya, K., Ojalehto, V., Savolainen, J., Niemistö, H., Hakanen, J., & Miettinen, K. (2014). Coupling dynamic simulation and interactive multiobjective optimization for complex problems: An APROS-NIMBUS case study. Expert systems with applications, 41(5), 2546-2558.

Staffell, I., & Pfenninger, S. (2018). The increasing impact of weather on electricity supply and demand. Energy, 145, 65-78.

Stallard, T., Collings, R., Feng, T., & Whelan, J. (2013). Interactions between tidal turbine wakes: experimental study of a group of three-bladed rotors. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 371(1985), 20120159.

Stauffer, L., & Liebling, T. M. (1997). Rolling horizon scheduling in a rolling-mill. Annals of Operations Research, 69, 323–349.

doi:10.6342/NTU202403713

Su, L. H. (2003). A hybrid two-stage flowshop with limited waiting time constraints. Computers & Industrial Engineering, 44(3), 409–424.

Sujitha, S., Shiva, R. V., Kulkarni, S., & Ponnappa, M. M. (2022, January). An implementation of soft computing approach of smart control for induction motor using ANFIS. In 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 1410-1413). IEEE.

Tang, L., Liu, W., & Liu, J. (2005). A neural network model and algorithm for the hybrid flow shop scheduling problem in a dynamic environment. Journal of Intelligent Manufacturing, 16(3), 361–370.

Tanner, M. W., & Ntaimo, L. (2010). IIS branch-and-cut for joint chance-constrained stochastic programs and application to optimal vaccine allocation. European Journal of Operational Research, 207(1), 290–296.

Taskın, Z. C., & Cevik, M. (2013). Combinatorial Benders cuts for decomposing IMRT fluence maps using rectangular apertures. Computers & Operations Research, 40(9), 2178–2186.

Tsiodra, M., & Chronopoulos, M. (2021). A bi-level model for optimal capacity investment and subsidy design under risk aversion and uncertainty. Journal of the Operational Research Society, 1-13.

Urquhart, M., Ljungskog, E., & Sebben, S. (2020). Surrogate-based optimisation using adaptively scaled radial basis functions. Applied Soft Computing, 88, 106050.

Vennell, R., Funke, S. W., Draper, S., Stevens, C., & Divett, T. (2015). Designing large arrays of tidal turbines: A synthesis and review. Renewable and Sustainable Energy Reviews, 41, 454-472.

doi:10.6342/NTU202403713

Verstichel, J., Kinable, J., De Causmaecker, P., & Berghe, G. V. (2015). A Combinatorial Benders' decomposition for the lock scheduling problem. Computers & Operations Research, 54, 117–128.

Wang, F., Zhang, H., & Zhou, A. (2021). A particle swarm optimization algorithm for mixed-variable optimization problems. Swarm and Evolutionary Computation, 60, 100808.

Wang, H., Jin, Y., & Jansen, J. O. (2016). Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system. IEEE Transactions on Evolutionary Computation, 20(6), 939-952.

Wang, H., Zhang, Q., Jiao, L., & Yao, X. (2015). Regularity model for noisy multiobjective optimization. IEEE transactions on cybernetics, 46(9), 1997-2009.

Wang, M., Srivathsan, S., Huang, E., & Wu, K. (2018). Job Dispatch Control for Production Lines with Overlapped Time Window Constraints. IEEE Transactions on Semiconductor Manufacturing, 31(2), 206–214.

Wang, S., Huang, G. H., Yao, Y., Xie, Y. L., & Zhen, J. L. (2017). An inexact fuzzy bi-level programming model for energy–traffic system planning under uncertainty: a case study of Urumqi city, China. Engineering Optimization, 49(8), 1441-1461.

Wang, S., Minku, L. L., & Yao, X. (2014). Resampling-based ensemble methods for online class imbalance learning. IEEE Transactions on Knowledge and Data Engineering, 27(5), 1356-1368.

Wang, S., & Yao, X. (2012). Multiclass imbalance problems: Analysis and potential solutions. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42(4), 1119-1130.

Wang, T., & Yang, Z. (2020). A Tidal Hydrodynamic Model for Cook Inlet, Alaska, to Support Tidal Energy Resource Characterization. Journal of Marine Science and Engineering, 8(4), 254.

Wu, C. H., Lin, J. T., & Chien, W. C. (2010). Dynamic production control in a serial line with process queue time constraint. International Journal of Production Research, 48 (13), 3823–3843.

Wu, C. H., Lin, J. T., & Chien, W. C. (2012). Dynamic production control in parallel processing systems under process queue time constraints. Computers & Industrial Engineering, 63(1), 192–203.

Wu, C. H., Chien, W. C., Chuang, Y. T., & Cheng, Y. C. (2016a). Multiple product admission control in semiconductor manufacturing systems with process queue time (PQT) constraints. Computers & Industrial Engineering, 99, 347–363.

Wu, K., Zhao, N., Gao, L., & Lee, C. K. M. (2016b). Production control policy for tandem workstations with constant service times and queue time constraints. International Journal of Production Research, 54(21), 6302–6316.

Wu, Y., Zhang, S., Wang, R., Wang, Y., & Feng, X. (2020). A design methodology for wind farm layout considering cable routing and economic benefit based on genetic algorithm and GeoSteiner. Renewable Energy, 146, 687-698.

Xie, C., Zhuang, X. X., Niu, Z., Ai, R., Lautrup, S., Zheng, S., ... & Fang, E. F. (2022). Amelioration of Alzheimer’s disease pathology by mitophagy inducers identified via machine learning and a cross-species workflow. Nature Biomedical Engineering, 1-18.

doi:10.6342/NTU202403713

Yan, S., Zou, X., Ilkhani, M., & Jones, A. (2020). An efficient multiscale surrogate modelling framework for composite materials considering progressive damage based on artificial neural networks. Composites Part B: Engineering, 194, 108014.

Yang, D. L., & Chern, M. S. (1995). A two-machine flowshop sequencing problem with limited waiting time constraints. Computers & Industrial Engineering, 28(1), 63–70.

Zhou, B., Pychynski, T., Reischl, M., Kharlamov, E., & Mikut, R. (2022). Machine learning with domain knowledge for predictive quality monitoring in resistance spot welding. Journal of Intelligent Manufacturing, 1-25.

Zhou, Z. H., Chawla, N. V., Jin, Y., & Williams, G. J. (2014). Big data opportunities and challenges: Discussions from data analytics perspectives [discussion forum]. IEEE Computational intelligence magazine, 9(4), 62-74.

Zuo, M., Dai, G., & Peng, L. (2019). Multi-agent genetic algorithm with controllable mutation probability utilizing back propagation neural network for global optimization of trajectory design. Engineering Optimization, 51(1), 120-139.

doi:10.6342/NTU202403713