Increasing the value of a genetic algorithm based breakwater layout model

R.M. Schaly
Increasing the value of a genetic algorithm based breakwater layout model

Master of Science Thesis

by

R.M. Schaly

to obtain the degree of Master of Science
at the Delft University of Technology,

Student number: 4445821
Project duration: March 1, 2021 – November 25, 2021
Thesis committee: Prof. dr. ir. M. van Koningsveld, TU Delft, Chair
Dr. ir. B. Hofland, TU Delft
Ir. C. Parkinson, Arcadis
Dr. ir. E. van Berchum, Arcadis

An electronic version of this thesis is available at http://repository.tudelft.nl/.
Preface

Since I was born, I spent my summer holidays on a sailing vessel. This evolved in selecting my master choice (Hydraulic Engineering) and the topic for this master thesis where I investigated the possibilities to increase the value of a breakwater layout model. Following the design and execution of specific test cases, I investigated the value of a breakwater layout model and its applicability.

First and foremost, I would like to thank Professor Mark van Koningsveld for his time, passion, and excellent advice throughout my thesis. His positive attitude always kept me motivated. I would also like to thank him that he was always reachable, no matter what time and day it was, this inspired me to work on my project as much as possible.

From my graduation company, Arcadis, I would like to thank Chris Parkinson for the critical look at my research. His experience in the field of port layout added a lot of knowledge to my research. Furthermore, I would like to thank Erik van Berchum who supported me with the research structure by constantly reviewing the paper. He taught me to sometimes take a step back and look at the bigger picture before continuing. This will definitely help me in the future.

I would like to thank Bas Hofland for his advice on academic writing.

In addition to the members of my commission, I would like to thank Fedor Baart, who provided a crash course programming. Furthermore, I would like to thank the staff behind Mark van Koningsveld that supported me with knowledge in the field of breakwater layout and the scheduling of meetings with him.

Finally, I would like to thank my family and friends for their support throughout my time as a student. This time would not have been so excited without them.

R.M. Schaly
The Hague, 12 November 2021
Summary

A port is a maritime facility that acts as a logistic hub where land and sea transport is exchanged. In 2020 ports handled approximately 80% of all worldwide trade. This huge reliance on ports means that operability is crucial, the extent to which a port is available to handle cargo. Therefore, port operators have to ensure safe conditions in the harbour basin. To accomplish this, engineers have to design economically optimal ports that meet the design criteria matching the vessels calling the port. Therefore, the engineer must trade-off between capital and operational expenditures and downtime costs (missed revenues because a port is not operable). Port development is typically guided by a master plan that includes information about the port layout, the dimensions of water areas (approach channel, turning and manoeuvring basin and berthing pocket) and the allocation of cargo types and terminals to the available port area. The actual realisation of a new port or terminal typically follows steps like pre-feasibility, feasibility, construction, operation and maintenance. The feasibility phase is vital since this is the phase where critical design choices are made and evaluated.

To ensure sheltered water in the harbour basin, breakwaters are built to reduce wave impact and longshore currents. Therefore, breakwater layout is an essential aspect of port design. A typical breakwater layout consists of several elements: one or two breakwater(s), an approach channel, berthing pocket, manoeuvring basin, turning basin and quay wall and terminal. The breakwater design goes through several phases, in which the first layouts are elaborated in the conceptual design phase. These layouts are customised according to the design criteria and site conditions. Since any choice made during breakwater design will affect the dimensions or orientation of the water areas and terminal, making the design process iterative rather than sequential. This is time-consuming when done by hand and limits engineers from working out every possible design as a conceptual design has to be made in only two or three months, meaning that time is limited. Not all alternatives within the design space can be investigated, so designs are likely chosen based on expertise and a reasonable layout rather than that the optimal solution is chosen.

Translating the breakwater layout design process into an automation tool that searches for breakwater layouts and evaluates them can support the engineer and increase the solution space. A valuable model, useful for engineers, should be widely applicable, verified, validated, and high in performance. Performance refers to the ease of use of the model and the efficient use of computing power. First, as breakwater is site-dependent, a tool should convert site-specific conditions into breakwater layouts to be widely applicable. Second, to increase confidence among engineers, a model should be internally validated. This ensures that the breakwater layouts are only the result of the input data and not the results of design flaws of the model. In addition, the generalisability of the model should be validated to show the applicability of such a model. Third, a model creates attractiveness if it is verified. Therefore, it should not suddenly crash or behave unexpectedly, meaning it must be free of bugs. Fourth, the model should be user-friendly and produce results in an acceptable time, effectively using the computer power, to be useful in the conceptual design phase.

Teeling (2020) researched the further development of a genetic algorithm-based breakwater layout model that searches for economically optimal solutions for a certain set of breakwater design criteria and site data. Teeling (2020) added more accurate site data with the implementation of a
detailed bathymetry and wave model. The wave conditions are generated by REFRAC, a refraction model that applies the geometrical optics approximation (directional change when waves moving from deep to shallow water). The generated alternatives will be compared on aspects like capital expenditure, operational expenditure and downtime costs. Finally, the cheapest alternatives will be presented. Thereby the model can speed up the design process and explore a larger solution space quickly. However, improvements on the applicability of the model are still possible. Furthermore, no systematical analysis has yet been conducted for internal and external validation and verification of the model and into the model’s performance. This means that the confidence level of the use of the model is insufficient. This research investigated the increase of value for complex design software when confidence in the model is lacking by focusing on verification, validation and improvement of its performance and applicability.

Test-driven development was used to verify, validate and assess improvements in the model’s value. To test the software requirements, test cases were developed to verify, validate, and improve performance and applicability. Executing those tests will answer how the model generates breakwater layouts and assist by increasing the model’s value. If the model fails, the code was rewritten until the model passed.

During the research into the Teeling (2020) model, it appeared that it is possible to define a specific bathymetry and wave environment. Still, applicability could be further increased by the addition of the alongshore positioning of the primary breakwater. The alongshore position was in the Teeling (2020) model fixed at $x_0 = 0$. Creating a variable $x_0$ creates the opportunity for the model to search for the optimal position alongshore. With the implementation of the alongshore position, the solution space of the model increased.

The model was verified to ensure that the model generates breakwaters without containing any bugs; following the development and execution of a verification test case, various script and execution errors were found in the original program, which caused the program to fail in approximately one out of three runs. Eliminating these bugs resulted in 100% successful execution for the executed test cases.

The validation of the model could be split into internal and external validity. A base case was developed and executed for internal validation to check whether the input was translated into the model’s output. This showed that the model solely produced results based on the design criteria and site data. In addition, external validation was carried out. Therefore, the applicability of the model on different cases was tested. Several tests that focus on varying bathymetry and wave conditions had to trigger the model to generate the breakwater at a location with a certain configuration where site conditions are economically optimal. The results showed that the model blocked the waves with the primary breakwater for the test cases focusing on a single wave direction. In addition, the model was able to recognise differences in bathymetry and searches for the optimal location. Furthermore, the external validation test cases revealed an error in the downtime estimation. All test cases erroneously showed zero downtime, which was resolved by rewriting the downtime equation.

Finally, the performance of the model was examined. The performance focuses on the efficiency of the model itself and its user-friendliness. For example, a clear user manual was missing, which made it almost impossible to install and run the model. A manual has been designed where the genetic algorithm and the design criteria of the model are explained, as well as the installation of the model and how to run the model. Test-driven development has led to a decrease in the computation time of the model. First, Dask was added to the model, enabling parallel processing using
multiple cores from a computer processor. Second, the optimisation parameters were tuned, leading to more efficient use of computing power. These interventions resulted in a total reduction of 81.25% in computing time, decreasing from 160 to 30 minutes on average per run.

In conclusion, the model was improved significantly with test-driven development by verification, internal and external validation, and increased applicability and performance. The breakwater layout model can accelerate the conceptual design phase, and alternatives can be investigated within a larger solution space. Therefore, this model can contribute to the design of breakwaters in the conceptual design phase.

Although the value of the model increased, there is still room for improvement. To further increase the model’s applicability, it is recommended to add a percentage of occurrence to the wave data and have the outcome data checked by an expert. This will increase the accuracy and value of the model.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>v</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xi</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xv</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Breakwater layout optimisation</td>
<td>2</td>
</tr>
<tr>
<td>1.2 The Teeling (2020) model</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Problem statement</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Objective</td>
<td>4</td>
</tr>
<tr>
<td>1.4.1 Research question</td>
<td>4</td>
</tr>
<tr>
<td>1.4.2 Scope</td>
<td>4</td>
</tr>
<tr>
<td>1.5 Thesis structure</td>
<td>5</td>
</tr>
<tr>
<td>2 Background information</td>
<td>7</td>
</tr>
<tr>
<td>2.1 Port planning</td>
<td>7</td>
</tr>
<tr>
<td>2.1.1 Port layout</td>
<td>8</td>
</tr>
<tr>
<td>2.2 Breakwater design</td>
<td>9</td>
</tr>
<tr>
<td>2.2.1 Design guidelines</td>
<td>9</td>
</tr>
<tr>
<td>2.2.2 Conceptual design</td>
<td>10</td>
</tr>
<tr>
<td>2.3 Design optimisation</td>
<td>12</td>
</tr>
<tr>
<td>2.3.1 Mathematical models</td>
<td>12</td>
</tr>
<tr>
<td>2.3.2 Optimisation of the conceptual breakwater layout design</td>
<td>13</td>
</tr>
<tr>
<td>2.4 Summary</td>
<td>14</td>
</tr>
<tr>
<td>3 Parametric breakwater model and possible additions</td>
<td>15</td>
</tr>
<tr>
<td>3.1 Parametric breakwater model</td>
<td>15</td>
</tr>
<tr>
<td>3.1.1 The Teeling (2020) model</td>
<td>15</td>
</tr>
<tr>
<td>3.1.2 Validation</td>
<td>19</td>
</tr>
<tr>
<td>3.1.3 Aspects to improve</td>
<td>19</td>
</tr>
<tr>
<td>3.1.4 Focus</td>
<td>21</td>
</tr>
<tr>
<td>3.2 Positioning of the breakwater</td>
<td>22</td>
</tr>
<tr>
<td>3.3 Soil model</td>
<td>24</td>
</tr>
<tr>
<td>3.3.1 Soil characteristics</td>
<td>25</td>
</tr>
<tr>
<td>3.4 Summary</td>
<td>26</td>
</tr>
<tr>
<td>4 Software development</td>
<td>27</td>
</tr>
<tr>
<td>4.1 Genetic Algorithm</td>
<td>27</td>
</tr>
<tr>
<td>4.2 Methods</td>
<td>28</td>
</tr>
<tr>
<td>4.2.1 Comparison</td>
<td>29</td>
</tr>
<tr>
<td>4.2.2 Test-driven development</td>
<td>30</td>
</tr>
<tr>
<td>5 Research method</td>
<td>31</td>
</tr>
<tr>
<td>5.1 Alongshore position</td>
<td>31</td>
</tr>
<tr>
<td>5.1.1 Implementation</td>
<td>31</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Design process for a rock construction according to the Rock Manual (CIRIA-CUR, 2007, p. 6)</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>A possible port layout containing: the primary [PB] and secondary [SB] breakwater, shelter the harbour basin; the water areas, approach channel [AC], turning basin [TB], manoeuvring basin [MB] and berthing pocket [BP]; the quay wall [QW].</td>
<td>2</td>
</tr>
<tr>
<td>1.3</td>
<td>Thesis outline</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>Outline of Chapter 2</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Operability level risk for ports and site specific conditions across the world (Wiegel et al., 2021)</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>Important steps during the conceptual design phase and the feedback loop by evaluation</td>
<td>10</td>
</tr>
<tr>
<td>2.4</td>
<td>Design optimisation (Arora, 2017)</td>
<td>12</td>
</tr>
<tr>
<td>2.5</td>
<td>Typical algorithm design process for optimisation</td>
<td>13</td>
</tr>
<tr>
<td>3.1</td>
<td>Outline of the Teeling (2020) model description, with recommended further research by Teeling (2020) and the focus of this research</td>
<td>15</td>
</tr>
<tr>
<td>3.2</td>
<td>Breakwater generation decision variables for RHS = 0 and SEC = 1 (Woerlee, 2019)</td>
<td>18</td>
</tr>
<tr>
<td>3.3</td>
<td>An example of a breakwater layout generation with fitness score of 0.857 of the Teeling (2020) model and the economic optimisation for the breakwater layout model</td>
<td>19</td>
</tr>
<tr>
<td>3.4</td>
<td>Main drivers for site selection regarding UNCTAD. (1978) - Cape Town (“Google”, n.d.). Environmental impact and the geotechnical study concerns both land and sea area</td>
<td>22</td>
</tr>
<tr>
<td>4.1</td>
<td>Genetic algorithm process</td>
<td>27</td>
</tr>
<tr>
<td>4.2</td>
<td>Phases of the waterfall development method</td>
<td>29</td>
</tr>
<tr>
<td>4.3</td>
<td>Typical framework for test-driven development (Kawalerowicz &amp; Madeyski, 2013)</td>
<td>30</td>
</tr>
<tr>
<td>5.1</td>
<td>A result of the Teeling (2020) model, whereby the primary breakwater is fixed at $x_0 = 0$ (red arrow)</td>
<td>31</td>
</tr>
<tr>
<td>5.2</td>
<td>Top down method to breakdown the model to find the essential changes to the model for position the breakwater alongshore and in what sequence it must happen. The dotted rectangle highlights all places where to change the alongshore coordinates of the nodes. The mating, mutation and generator sections are part of the genetic algorithm</td>
<td>32</td>
</tr>
<tr>
<td>5.3</td>
<td>Wave field generated by REFRAC for a linear bathymetry</td>
<td>37</td>
</tr>
<tr>
<td>6.1</td>
<td>The implementation of $x_0$ prior and after the debugging. X-axis: alongshore distance [m]; y-axis: cross-shore distance [m]; colour bar: wave height [m]</td>
<td>40</td>
</tr>
<tr>
<td>6.2</td>
<td>The difference before and after debugging</td>
<td>40</td>
</tr>
<tr>
<td>6.3</td>
<td>The asymmetric contour bathymetry for the validation of the primary breakwater positioning. The red dashed rectangular represents the boundaries of the search area of the model</td>
<td>41</td>
</tr>
<tr>
<td>6.4</td>
<td>The wave field belonging to the alongshore positioning test case. X-axis: alongshore distance [m]; y-axis: cross-shore distance [m]; colour bar: wave height [m]</td>
<td>41</td>
</tr>
</tbody>
</table>
6.5 The results of the alongshore positioning validation test case. The wave direction is shown in the top corners of the figure. In the title of the figure, two values are indicated in brackets (fitness score, amount of generations before convergence) ........................................ 42

6.6 2D bathymetry profile for the asymmetric test cases with corresponding wave field for \( \theta = 0^\circ \) to shore normal ......................................................... 43

6.7 Linear bathymetry. The red dashed rectangular represents the boundaries of the search area of the model ................................................................. 44

6.8 The results of the base case, with linear bottom profile. The wave direction is shown in the top corners of the figure. In the title of the figure, two values are indicated in brackets (fitness score, amount of generations before convergence) ........................................ 44

6.9 Expected breakwater layout for wave directions coming from \(-30^\circ \) and \(+30^\circ \) to the shore normal, all other environmental variables are the same for both cases. The wave direction is represented in the top corners ..................................................... 46

6.10 The results of the test case with waves coming from -30\(^\circ\). The wave direction is shown in the top corners of the figure. In the title of the figure, two values are indicated in brackets (fitness score, amount of generations before convergence) ........................................ 47

6.11 The results of the test case with waves coming from \( \theta = +30^\circ \) to shore normal. The wave direction is shown in the top corners of the figure. In the title of the figure, two values are indicated in brackets (fitness score, amount of generations before convergence) ........................................ 47

6.12 2D bathymetry profile for the asymmetric test cases ......................................................... 48

6.13 Uniform wave field. \( X\)-axis: alongshore distance [m]; \( y\)-axis: cross-shore distance [m]; colour bar: wave height [m] ......................................................... 49

6.14 The results of the first bathymetry test case ................................................................. 49

6.15 Corresponding wave field to bathymetry shown in Figure 6.12a \( X\)-axis: alongshore distance [m]; \( y\)-axis: cross-shore distance [m]; colour bar: wave height [m] ......................................................... 50

6.16 Breakwater layouts for a bathymetry with a gentler slope at the left side. The wave direction is shown in the top corners of the figure. In the title of the figure, two values are indicated in brackets (fitness score, amount of generations before convergence) ........................................ 51

6.17 Corresponding wave field to bathymetry shown in Figure 6.12b \( X\)-axis: alongshore distance [m]; \( y\)-axis: cross-shore distance [m]; colour bar: wave height [m] ......................................................... 52

6.18 Breakwater layouts corresponding to a bathymetry profile with a gentler slope at the right side. The wave direction is shown in the top corners of the figure. In the title of the figure, two values are indicated in brackets (fitness score, amount of generations before convergence) ........................................ 52

6.19 Contour bathymetry of sudden deepening. The red dashed rectangular represents the boundaries of the search area of the model ......................................................... 53

6.20 Results for the sudden deepening test case, with a depth drop behind \( y = 1800 \) m. The wave direction is shown in the top corners of the figure. In the title of the figure, two values are indicated in brackets (fitness score, amount of generations before convergence) ........................................ 53

6.21 First downtime analysis test case ................................................................. 54

6.22 Second downtime analysis test case ................................................................. 55

6.23 The steps necessary to install and run the model ................................................................. 56

6.24 The results for the different population sizes, with a mutation rate of 0.25 and mutation size of 0.2. After a population size of 48 the increase in fitness score stagnates ......................................................... 57

6.25 The fitness score plotted against the computation time, for various mutation rate, with a population size of 48 and mutation size of 0.2 ......................................................... 58
6.26 Mutation size assessment, where the fitness score is plotted against the computation time. The population size equals 48 and the mutation rate 0.25

B.1 Breakwater model

D.1 Example input file for REFRAC

D.2 The input data for REFRAC with explanation on the most important variable parameters
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Typical elements considered for a port layout</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Advantages and disadvantages using a genetic algorithm for optimisation (Yang, 2020)</td>
<td>13</td>
</tr>
<tr>
<td>2.3</td>
<td>Comparison of the three genetic algorithm models (Elchahal et al., 2013; Teeling, 2020; Woerlee, 2019) and the requirements set by Ligteringen (Ligteringen, 2012). 'x' shows the model excludes the aspect, '-' the model incorporates the aspect but improvements add value to the model, '✓' indicates the model met the requirements for the conceptual design phase.</td>
<td>14</td>
</tr>
<tr>
<td>3.1</td>
<td>Guidelines according to the wet and dry infrastructure and downtime for the Teeling (2020) model</td>
<td>16</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparison of the different software development methods considered. A '+' means the model is favourable, '-' means not suitable for the objective of this research</td>
<td>29</td>
</tr>
<tr>
<td>5.1</td>
<td>The breakwater nodes and if they are fixed or variable</td>
<td>32</td>
</tr>
<tr>
<td>5.2</td>
<td>The guidelines applied for the various elements of the port layout</td>
<td>35</td>
</tr>
<tr>
<td>6.1</td>
<td>Design parameters applied to all test cases</td>
<td>39</td>
</tr>
<tr>
<td>6.2</td>
<td>CapEx, OpEx and downtime costs for the alongshore positioning</td>
<td>42</td>
</tr>
<tr>
<td>6.3</td>
<td>Number of errors found in the Teeling (2020) model</td>
<td>43</td>
</tr>
<tr>
<td>6.4</td>
<td>Internal validation results</td>
<td>45</td>
</tr>
<tr>
<td>6.5</td>
<td>CapEx, OpEx and downtime costs for a wave direction of −30° to shore normal</td>
<td>47</td>
</tr>
<tr>
<td>6.6</td>
<td>CapEx, OpEx and downtime costs for a wave direction of +30° to shore normal</td>
<td>48</td>
</tr>
<tr>
<td>6.7</td>
<td>CapEx, OpEx and downtime costs for test case 1</td>
<td>50</td>
</tr>
<tr>
<td>6.8</td>
<td>CapEx, OpEx and downtime costs for test case 2</td>
<td>51</td>
</tr>
<tr>
<td>6.9</td>
<td>CapEx, OpEx and downtime costs for test case 3</td>
<td>52</td>
</tr>
<tr>
<td>6.10</td>
<td>CapEx, OpEx and downtime costs for a sudden deepening profile</td>
<td>54</td>
</tr>
<tr>
<td>6.11</td>
<td>Comparison of the computation time between the Teeling (2020) model with and without alongshore position, and the updated model running on two or four cores</td>
<td>56</td>
</tr>
<tr>
<td>6.12</td>
<td>The average computation time and fitness score for the various population sizes</td>
<td>57</td>
</tr>
<tr>
<td>6.13</td>
<td>The mutation rate with corresponding computation time and fitness score</td>
<td>58</td>
</tr>
<tr>
<td>6.14</td>
<td>The varying mutation size with computation time and fitness score</td>
<td>58</td>
</tr>
<tr>
<td>A.1</td>
<td>Input parameter for Teeling (2020) and current model (Teeling, 2020)</td>
<td>80</td>
</tr>
<tr>
<td>E.1</td>
<td>Additional width to the approach channel (Woerlee, 2019)</td>
<td>92</td>
</tr>
<tr>
<td>E.2</td>
<td>Additional approach channel depth Woerlee (2019)</td>
<td>92</td>
</tr>
<tr>
<td>E.3</td>
<td>Results for the population size test case</td>
<td>93</td>
</tr>
<tr>
<td>E.4</td>
<td>Results for the mutation rate test case</td>
<td>94</td>
</tr>
<tr>
<td>E.5</td>
<td>Results for the mutation size test case</td>
<td>94</td>
</tr>
</tbody>
</table>
Introduction

In 2020 ports worldwide were responsible for 80% of all trade (UNCTAD, 2020). It is therefore important that these ports remain operational. So an efficient design, to ensure smooth handling of cargo is essential. In addition, it is important that the water areas of the harbour and berths are sufficiently protected against waves, to ensure safe access and calm conditions at the berth for vessels. Therefore, engineers need to shelter these water areas and berths by breakwaters.

Breakwater design

Breakwaters are used to ensure that vessels can enter and manoeuvre safely, and to reduce ship movement at the berths by minimising wave action inside the harbour basin (Bijker & Massie, 1976). This minimises port downtime. The design process of a breakwater is shown in Figure 1.1 from conception to the final design. In the conceptual phase, the first conceptual breakwater layouts are elaborated. The elements considered for the breakwater layout design are the approach channel angle and width, turning basin, quay wall, one or two breakwater(s) and minimal distance between these elements. In combination with the site data, a conceptual breakwater layout can be developed. Since any choice made during breakwater layout design will affect the approach channel, turning basin and quay wall design, the design process iterative rather than sequential. This is time-consuming when done by hand and limits engineers from working out every possibility, as time is scarce in the conceptual design phase, leading to designs being chosen based on expertise (CIRIA-CUR, 2007). Not all alternatives within the design space can be investigated, so a reasonable layout rather than the optimal solution is likely chosen.

The use of a model for the conceptual design process can support engineers in discovering more or better alternatives within the same time. Such automation requires a model that generates breakwater layouts by considering the design criteria and incorporating the site data. Therefore, a model should be widely applicable as breakwater design is not uniform worldwide. In addition, since there is little time available, a model must be easy to use and must give reliable results within an acceptable time. Furthermore, broadly speaking of a valuable model, it should be verified. Verification checks whether a model does not behave unexpectedly or crash and, therefore, may not contain
any bugs. When a model meets all these requirements, the confidence and attractiveness among engineers will increase.

1.1. Breakwater layout optimisation
Little research exists in the automation of breakwater layout design. The use of a root-finding algorithm developed by Rustell (2013) showed that for an LNG where the x- and y-location of the berth is known, the model could provide quick solutions. Nevertheless, the input data for this algorithm lacks accuracy and wave processes are a rough estimation. Another study, showed that the breakwater layout model can be translated into a genetic algorithm to speed up the design of detached breakwaters (Elchahal et al., 2013). However, computation time of the model is high and design criteria regarding safety aspects for vessels are not implemented. Furthermore, two recent studies validated the possibility of partly automating the conceptual layout design of a rubble mound breakwater by the use of a genetic algorithm (Teeling, 2020; Woerlee, 2019). Woerlee (2019) conducted a research on the development of a parametric model, by the principles of a genetic algorithm, and proved the concept of automation of the breakwater layout design. Teeling (2020) increased the applicability of this model by implementing detailed bathymetry and a wave model. The model’s goal is to optimise breakwater layouts towards an economical optimum by comparing alternatives on a fitness score. With the automation of the design process, many breakwaters are produced within a short period, covering a larger solution space than the traditional design process, which increases the chance to find the most optimal design. It is chosen to focus on the Teeling (2020) model to further investigate as design criteria, bathymetry and wave data are implemented and is applicable for a wide range of cargo vessels. The comparison between the four models is further elaborated in Section 2.3.

1.2. The Teeling (2020) model
Teeling (2020) continued to develop the Woerlee (2019) model by implementation of a wave model and detailed bathymetry to further increase the applicability of it. A possible layout generated by the model is visualised in Figure 1.2. The model generates the secondary and primary breakwater [SB, PB], respectively, to shelter the harbour basin for waves and longshore currents. Furthermore, the model generates the water areas containing an approach channel [AC], turning [TB] and manoeuvring basin [MB], and berthing pocket [BP]. The dimensions and alignment of these elements depend on the tuning of the design criteria determined by the safety guidelines and input data.

Figure 1.2: A possible port layout containing; the primary [PB] and secondary [SB] breakwater shelter the harbour basin; the water areas, approach channel [AC], turning [TB], manoeuvring basin [MB] and berthing pocket [BP]; the quay wall [QW].

- The approach channel:
  - The approach channel is assumed to be a straight line, where the orientation of the chan-
1.3. Problem statement

The length of the approach channel (sheltered by the breakwaters) depends on the time a tug can hook up on the vessel and the stopping length of the design vessel.

The width of the approach channel depends on the design vessel dimensions.

Inside the breakwaters, there must be enough space for turning, manoeuvring and berthing. The turning basin is located at the end of the approach channel, where the manoeuvring basin ensures the distance between the berthing pocket and the turning basin. The design vessel length determines the diameter of the turning basin. The dimensions of the berthing pocket depend on the length and width of the design vessel. The design vessel length estimates the quay wall length along with the required berths. The user defines the quay wall depth.

In addition to the design criteria, the Teeling (2020) model also needs input files created by the user. To incorporate the bottom profile at the site, a bathymetry file is needed created by the user. The model uses bathymetry to estimate the different elements’ costs, including segment costs and dredging and filling costs. In addition, a yearly siltation rate is used to estimate the dredging maintenance costs. Furthermore, the bathymetry is applied to achieve the accompanying wave fields. To simulate these wave fields, REFRAC is used, which is a refraction model that applies the geometrical optics approximation (Liu, 2009). REFRAC requires a bathymetry and offshore wave height and direction defined by the user to generate wave fields. This bathymetry and wave data, together with the design specifications (such as design vessel dimensions, limiting wave height in the approach channel and berths), estimates the dimensions and orientation of the water areas, quay wall and breakwaters. The total capital expenditures (CapEx) and operational expenditures (OpEx) are estimated with these dimensions. For each specific layout, a certain downtime is estimated, the time a port is not operable. If a port is inoperable revenue will be missed. The total costs of the port consists of the sum of the CapEx, OpEx and downtime costs. To optimise the breakwater layouts the model uses a Non-dominated sorting genetic algorithm (NSGA-II). This is an algorithm inspired by the process of natural selection whereby only alternatives are selected for the next generation that dominate all aspects compared to that of the other alternatives. For each generation, a certain fitness score is estimated, which are the client preferred costs divided by the CapEx, OpEx and downtime costs. Those fitness scores are compared and cheaper alternatives are passed to a next generation.

During the investigation of the Teeling (2020) model, it became clear that the model can be applicable for a wide range of cases through the implementation of a detailed bathymetry and wave model, including the implemented design criteria. However, Teeling (2020) stated the fact that room for improvement has not been excluded regarding its applicability. Therefore, the model should be investigated to explore potential gaps for improvement. In addition, to reflect the actual value of the model, verification and internal and external validation are required, and the model’s performance should be assessed for engineers to gain confidence in the model. Woerlee (2019) and Teeling (2020) have not systematically analysed the valuable aspects, so it is not clear whether this has been looked into. A useful model should correctly operate, without containing bugs and should, therefore, be verified. Without internal validation, it is not clear whether the design criteria are actually translated into the results. Simultaneously, the external validation examines whether the model responds to the bathymetry and the inserted wave fields. In addition, as a result, the external validation provides an answer to the consistency and reliability of the results. At last, since the conceptual design is time-consuming, a high-performance model will make it even more attractive to use. Aspects to consider are tuning the optimisation parameters to increase efficient use of the computing power to speed up processing and focus on the user-friendliness of the model.
1.4. Objective
The problem statement of this report points out that the Teeling (2020) model already incorporated essential aspects for breakwater design during the conceptual phase. However, engineers will not make use of the model until confidence has been well established. Engineers can obtain confidence as soon as the model is verified and internal and external validated, and the performance and applicability increase. In order to accomplish this, knowledge about the conceptual breakwater design is required to assess the reliability of the internal and external validation test results. In addition, these validation tests must focus on the consistency of the results to ensure the proper functionality of the model. Furthermore, verification is needed to track and resolve eventual bugs in the model. Next, by tuning the optimisation parameters, the performance and efficiency of the model will be increased. Also, the addition of a user manual will increase the user-friendliness of the model which makes it more attractive to use. Lastly, possible additions are being investigated to increase the applicability of the model further. As a result of this, the following research objective is formulated:

Investigate the possibilities to increase the value of the breakwater layout model by verification, validation and improvement of performance and applicability.

1.4.1. Research question
In line with the main objective, the following main research question is formulated:

How can the value of the genetic algorithm based breakwater layout model be improved?

Based on the main research question, five sub-questions are formulated:

1. What components are covered in the conceptual design of the breakwater layout?
2. What additions to the model are possible to increase the applicability?
3. How can possible errors be detected and resolved in the model?
4. How can the model be internally and externally validated?
5. How can the performance of the model be assessed and improved?

1.4.2. Scope
The conceptual design of the breakwater layout focuses on the input, site data and design criteria for the breakwater layout design. The impact on the environment and stakeholders is out of the scope of this research.

This research paper focus on the value improvement of the Teeling (Teeling, 2020) model. To assess the value of the model, the following aspects are considered:

- Applicability
- Number of bugs
- Internal validation
- External validation
- Performance

In this research, the assumption is made that the Teeling (2020) model's value only depends on these five aspects. This is a simplification from the real-world dynamics but is needed for interpretability. Furthermore, the following restrictions defined in the Teeling (2020) remain in effect for the breakwater layout model:
• The model is capable of designing a breakwater layout for one specific design vessel each run.

• The model is only applicable in areas where directional spreading of waves is negligible regarding shoaling and refraction.

• The model only assumes CapEx, Opex and downtime as costs for the breakwater.

• The downtime estimation equations in the model are solely based on the wave height.

• The model assumes a rubble mound breakwater. This is incorporated in the breakwater volume and cost.

• The model generates berths along a single quay, which is parallel to the shoreline.

• The model only addresses technical aspects of design. Social and environmental aspects are not included in the model.

• Extreme events, like ice, hurricanes, earthquakes or others, are not incorporated.

1.5. Thesis structure
After these brief introductory remarks, Chapter 2 considers the theoretical background of breakwater design and possible tools to support the design. It guides the reader through the literature background mandatory to answer the research question. Chapter 3 provides the reader with an introduction to the Teeling (2020) model and possible improvements. In addition, this chapter discusses the literature necessary to apply the improvements. Next, Chapter 4 elaborates on the methodology, after which Chapter 5 specifies the model adaptations and shows the results of the validation test cases. Chapter 6 highlights the adjustments made in the updated model compared to the Teeling (2020) model. The advantages and disadvantages of the adaptations in the breakwater layout model are being discussed. Chapter 7 discusses the updated breakwater layout model and the model’s uncertainties and limitations, and the current state of the model. Finally, Chapter 8 provides a conclusion together with some recommendations for further research. Auxiliary data and analysis of this thesis are presented in the appendices and a list of references referred to in the text. Figure 1.3 on the next page, shows the outline of the report.
1. Introduction

Figure 1.3: Thesis outline
Background information

The first section elaborates on port planning to show the essence of breakwater design, followed by a more specific elaboration on breakwater design in Section 2.2.2. Then, in Section 2.4 design optimisation and the breakwater layout design optimisation are discussed. Also, a suitable breakwater layout model is selected to further investigate. Figure 2.1 presents the outline of this Chapter.

2.1. Port planning

A port is a logistic hub that connects the land and sea for handling cargo. For the construction or expansion of a port, a port master plan should be developed, which should consist of the following factors (van Koningsveld et al., 2021):

- Strategic plan
- Cargo and shipping forecast
- Site data
- Nautical access
- Hinterland connections
- Environmental, safety and social aspects
- Stakeholders
- Financial and economic feasibility

The first step of port development, the pre-feasibility study, focuses on developing a strategic plan. At the pre-feasibility stage, little data is available, and mainly only already existing information can be used. Therefore, the strategic plan only provides a first idea of the project boundaries, a market
2. Background information

Analysis, the potential risks and the functional requirements [PIANC, 2019a]. A cargo forecast is essential for this to determine the economic and financial position of the port. The next step is the feasibility study, where the strategic plan is further investigated. Since little site data is available at the start of this phase, a site survey should be carried out. The site survey is essential as the first port layouts are designed during this stage in the conceptual design. The port layouts contain a first design of the port, together with an economical and financial evaluation. In addition, a risk analysis must be carried out to detect potential hazards for the project feasibility. At the end of this stage, two or three designs are selected to further investigate [Velsink, 1993]. All together should provide an answer to whether the project seems feasible. If this is the case, the next step will focus on obtaining the permits, land acquisition and arranging financing. If the permits are obtained, a detailed design is developed with a complete description of all components. After finishing the detailed design, the project can be realised. Project realisation focus on the construction, operation and maintenance.

2.1.1. Port layout

The design of the port layout is an essential aspect of port planning as it has a significant effect on the total costs of the port. The considering elements are the water areas and the port structures, which are shown in Table 2.1. The terminal includes the hinterland connections, storage area, number of berths, quay length and the required equipment.

Table 2.1: Typical elements considered for a port layout

<table>
<thead>
<tr>
<th>Water area</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot boarding</td>
<td>Breakwater</td>
</tr>
<tr>
<td>Anchorage</td>
<td>Terminal</td>
</tr>
<tr>
<td>Outer approach channel</td>
<td></td>
</tr>
<tr>
<td>Inner approach channel</td>
<td></td>
</tr>
<tr>
<td>Turning basin</td>
<td></td>
</tr>
<tr>
<td>Berthing and manoeuvring basin</td>
<td></td>
</tr>
</tbody>
</table>

With an expectation of an increase in seaborne trades, it is even more critical for ports to minimise downtime during the year. Sheltered water is an essential factor usually provided by either a natural shelter or a breakwater to ensure operability. Breakwaters provide calm conditions inside the harbour basin for vessels to safely sail towards the quay. Figure 2.2 shows the level of operability risk and which environmental factors are of importance across the world. It can be observed that site conditions differ worldwide. Therefore, breakwaters are customised to the specific site.

Figure 2.2: Operability level risk for ports and site specific conditions across the world [Wiegel et al., 2021]
2.2. Breakwater design

Knowing that breakwater design is extremely important to ensure operability, this section focuses on breakwater design. The design of breakwaters is highly customised as engineers deal with site-specific conditions that vary worldwide. Figure 2.2b shows the determining conditions that endanger the operability of ports. Depending on the site, an engineer can come up with a favourable breakwater type (rubble-mound, vertical, caisson, or others) and cross-section design and the optimal location and layout.

For the design of a breakwater, the following design phases are most often mentioned: (CIRIA-CUR, 2007):

- **Conceptual design.** The first breakwater layouts are elaborated, where the interaction between the different elements of the breakwater are incorporated.

- **Preliminary design.** The conceptual design is further developed with higher accuracy, where one layout is selected to be passed on to the next phase. Furthermore, in this stage, the cross-section of the breakwater is designed.

- **Detailed design.** Calculate the detailed structure dimensions and determine the final shape of the different breakwater elements.

The site study carried out during the feasibility study is used for the design phases of a breakwater. A first layout design is created in the conceptual design phase within the limited time available. Within this time limit, it is not possible to accurately optimise the breakwater design. Also, because the breakwater design is very dependent on the location, it is not possible to re-use a design of another location. As the concept phase highly influences the total costs of the breakwater, and since the breakwaters are a major cost factor for the overall port design, an optimised breakwater design can save costs (Yussef et al., 2020). The task for the designers is to provide the optimum combination of high operable ports with low breakwater costs. During the preliminary design, the breakwater alternatives are further developed, and the cross-sectional design of the breakwater is calculated. The results are provided to the client to check their requirements and match with the alternatives. Furthermore, the total costs are calculated more accurately, choosing one alternative at the end of this phase. In the last phase, a detailed design is created. Accurate drawings and stability and cost estimations are used to form the basis of the contracts. With these drawings, the construction phase can be carried out.

The breakwater layout design can be elaborated in different ways. Firstly, it is possible to do the entire design process by hand, where the engineer himself does all calculations. After that, a physical model is created to test the designed breakwater. Second, a combination of the use of numerical models and calculations by hand is possible, again translated into a physical model. The use of models can support engineers exploring a wider solution space and speed up the design process.

2.2.1. Design guidelines

For the design of a breakwater, several guidelines and studies are possible to apply. The following guides are most commonly used:

- **PIANC** (PIANC, n.d.)
- **The Rock Manual** (CIRIA-CUR, 2007)
- **EurOtop** (EurOtop, 2016)

Besides these guidelines, many others can be used. However, these are the most common and provide a broad knowledge of how engineers design a breakwater. An elaboration on how these guidelines are applicable for breakwater design is provided.
PIANC is the World Association for Waterborne Transport Infrastructure (formerly, Permanent International Association of Navigation Congresses). PIANC unite worldwide experts to develop economic, technical and environmental guidelines. Also, PIANC developed and still develops general and specified guidelines for the design of breakwaters. The guidelines support engineers during the design of breakwaters. Criteria are set in these manuals that ensure safe conditions within the harbour basin.

The Rock Manual focus on the use of rocks in the field of hydraulic engineering (CIRIA-CUR, 2007). Parts of the manual focus on the design of breakwaters. Compared to PIANC, the manuals do not cover criteria relating to the positioning of the various components of the breakwater design. However, the manual does support the engineer in the breakwater layout by naming the various processes influencing the breakwater layout. The main goal of this manual is to provide the engineer with equations to check cross-section geometry for the breakwater. The cross-sectional geometry means checking the breakwaters dimensions and the different breakwater types, checking the breakwater against instabilities due to wave energy, including the failure analysis, and providing the engineer with an overview of the favourable rock type.

EurOtop is a manual on wave overtopping for coastal structures (EurOtop, 2016). Breakwaters are exposed to waves, and in line with that, wave overtopping may occur. Wave overtopping can increase the operability risk level but also causes loss of life sometimes. Wave overtopping at breakwaters mean that waves fully or partly flow over the breakwater. This manual provides engineers equations to ensure no or less wave overtopping occurs. Furthermore, this manual provides the engineer equations for wave run-up. Wave run-up is assumed less important but helps the engineer predict the number of waves reaching the breakwater crest.

2.2.2. Conceptual design

The guidelines support an engineer during all design phases and provide reasonable estimations for the conceptual design. A typical conceptual design is made in only two to three months, according to a professional from Royal HaskoningDHV (Boer, 2021), covering the layout design of the breakwater and a feedback loop with the client. Therefore, an engineer will elaborate several alternatives mainly based on expertise. The conceptual design mainly focuses on the trade-off between CapEx, OpEx and downtime costs. Figure 2.3 highlights important steps for a typical conceptual design of a breakwater, where a design is generated by the input data and constantly evaluated to incorporate client preferences.

![Conceptual breakwater design](image)

Figure 2.3: Important steps during the conceptual design phase and the feedback loop by evaluation

**Input**

The input for the breakwater conceptual design consists of the site data, design criteria and the stakeholders. With the obtained input, a first conceptual design can be developed. An elaboration on the input data is provided below.
Site data  During the conceptual design phase, site data that should be incorporated are bathymetry and topography, wave conditions and water level, sedimentation and erosion and soil characteristics. The breakwater CapEx and OpEx are closely related to the site data:

- The topography determines the optimal location for the port and, therefore, also influence the breakwater location. For this location, a specific bathymetry is present. The bathymetry influences the breakwater costs by the needed breakwater volume.

- A breakwater must provide the necessary safety in the harbour and, therefore, consider the wave conditions and water level. The wave conditions are incorporated in the alignment and length of the breakwater. The water level is influenced by many factors, such as tide and extreme events, and should be incorporated in the breakwater height. This affects the required dimensions of the breakwater.

- The length and number of breakwaters determines how much sediment transport occurs. Critical is sediment transport alongshore, whether constant or variable. Therefore, a secondary breakwater may be needed to reduce the variance in sediment transport. A trade-off between the number of breakwaters and length and dredging costs arise.

- The soil characteristics determine the additional costs regarding the breakwater stability. In addition, there are certain costs associated with dredging a specific soil type, such as cheaper soil types (sand) and more expensive soil types (rock). Also, the re-use of dredged material, e.g. in site preparation for the onshore part of the port is an important aspect that can reduce the total costs of a port. Furthermore, this also influences the sediment and erosion around the port.

Design criteria  Mainly speaking for the strategic plan developed in the pre-feasibility study, a cargo forecast is estimated, which provides the cargo type and volume and the design vessel. However, the Teeling [2020] model is used to be applied after that phase already having those design parameters. For ports to be operable, vessels calling the port expect safe conditions during sailing outside and inside (including safe tug attaching) and cargo handling at the berth. For the safety of vessels, the guidelines are used to define the design criteria. Some common design criteria are:

- In case of strong off-shore wave conditions, the approach channel should be preferably aligned with the wave direction.

- Space between the breakwater and approach channel creates safer conditions for the vessels while sailing between the breakwaters.

- Bends in the approach channel should be avoided to minimise steering complications.

- The berths should preferably not be placed at the end of the approach channel to provide terminal safety.

Stakeholders  Stakeholders and environmental concerns may impact the breakwater design, but they are not addressed in the model, except in where they create limits on the spatial location of the breakwaters.

Design  An engineer designs a breakwater layout in the conceptual design phase using the design criteria and the available site data. The engineer’s objective is to design a breakwater layout that trades off the CapEx, OpEx and downtime costs. It does not mean that the downtime costs, therefore, have to
be zero. Unsafe conditions cause downtime for the port. The downtime is split up into operation and navigation downtime. Operation is governed by if ship movements exceed those allowable for the cargo handling equipment or cause reduced efficiency. It also includes having to leave a port early if an oncoming storm is likely to cause waves at the berth, exceeding the mooring system's limits. In this case, a ship has to leave before the conditions are too bad for a safe departure from the port. Then may have to wait outside until it is safe to return. Navigational downtime refers to the berth or port's inaccessibility due to bad weather. The construction and alignment of the breakwater can provide safe sailing and berthing. During the conceptual design, the engineer designed the breakwater layout.

The layout of the breakwater influences the dimensions and alignment of the water areas and quay wall (Ligteringen, 2012). Furthermore, the breakwater layout influences the dredging and filling costs, thus affecting CapEx and OpEx. Therefore, every choice made for one element influences the others, the design is rather iterative then sequential. In addition to the layout of the breakwater, a cross-section will also have to be designed. However, cross-sectional design take place in the preliminary phase.

**Evaluation**

During the design process, multiple conceptual designs are developed. Engineers can perform a multi-criteria analysis (MCA) to select the preferable layout. One aspect of an MCA covers the costs. As the costs estimated in the conceptual phase highly influence the total costs of the project, engineers should carefully consider the layouts (Yussef et al., 2020). However, by comparing the different designs, there is still the possibility that the optimal solution has not been estimated. Therefore, a potential gap in the use of computers arise.

### 2.3. Design optimisation

Understanding the design process helps to understand how breakwater design can be optimised and how computer models can support optimisation. First, it is elaborated on how a design can be optimised. Next, possibilities for conceptual breakwater design optimisation are investigated.

#### 2.3.1. Mathematical models

Optimisation is the process to find the best or most optimal result within certain limits. Optimisation can be used for different purposes, and one of them is design optimisation in the engineering field. Figure 2.4 represents the steps usually considered during design optimisation.

![Figure 2.4: Design optimisation (Arora, 2017)](image)

Design optimisation in the engineering field refers to finding optimum solutions with predefined criteria or constraints (Kelley, 2010). At the start of the design optimisation, the problem is formulated and consists of three components: selecting the optimisation variables, the design functions and the constraints to the design (Bhatti, 2012). The three components can be translated into a mathematical model. Within a mathematical model, a real-life problem can be solved by the use of equations and provides insight into the relationship between the equations and the results (Dundar et al., 2012). It should be noted that this does not necessarily require computers. However, in the design process, computers are gaining importance to execute these calculations to support with
the accuracy of and speeding up the design process. A mathematical model discussed above can be translated into an algorithm (Boult, 1987). Algorithms are well-defined sequences of actions presented at a computer to solve a problem. Figure 2.5 presents a typical algorithm design process used for optimisation.

The first step of the algorithm is to describe the problem and find out what a preferred solution is. With a clear problem description and objective, the necessary steps should be translated into a sequence that can be converted into an algorithm. Next, a suitable algorithm should be selected that is most suitable for the problem to increase efficiency. Algorithms are time-consuming and can be very precise, and therefore create a trade-off between efficiency and accuracy. The accuracy depends on the problem, but an effort should be taken to select the appropriate accuracy within its bounds to reduce the running time. Third, the code is written where the developer and user should understand the working of the model to explain the obtained results. Last, the model should be tested to discover bugs and potential increase of computation speed. By testing the algorithm, reliability for the model increases.

2.3.2. Optimisation of the conceptual breakwater layout design

The conceptual layout design of a breakwater is an iterative process, which requires multiple iterative steps to optimise the design. Numerical and physical models exist to model the impact of waves on the breakwater. These models are suitable for determining the CapEx, OpEx and downtime costs but take much time developing and designing them, while time is scarce in the conceptual design phase. The possibility of translating the iterative design process into an algorithm is possible since the design are mathematical rules that provide a result (Kicinger et al., 2005). The use of evolutionary algorithms, in particular the genetic algorithm, for civil engineering design process optimisation is a proven concept (Kicinger et al., 2005).

To become familiar with the background of a genetic algorithm, the application for engineering purpose will be described. The most important advantages and disadvantages for the use of a genetic algorithm for engineering purposes are shown in Table 2.2

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex problems</td>
<td>Formulation of the core equations</td>
</tr>
<tr>
<td>Parallelism</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.2: Advantages and disadvantages using a genetic algorithm for optimisation (Yang, 2020)

One advantage of a genetic algorithm is the ability to solve complex problems, the ability to deal with several optimisations and generate possible designs (Szczerbicka et al., 1998). Next, the genetic algorithm can search parallel for solutions and speed up a complex design process (Wang, 2001). To make use of this, the core of the genetic algorithm must be correctly defined. If the fitness function formulation, population size, mutation rate, mutation size and the selection criteria for the new population are not appropriately defined, the genetic algorithm probably does not converge or produce unusable results (Yang, 2020).
However, little research is available on the use of algorithms for the conceptual breakwater layout. Elchahal et al. (2013), Woerlee (2019) and Teeling (2020) investigated the use of a genetic algorithm to optimise the breakwater layout in the conceptual design phase. The similarity in all models is that they parameterise the breakwaters by nodes that reflect the optimisation problem’s variables. Diving in the models highlights the differences between the models. Elchahal et al. (2013) uses the genetic algorithm to optimise the layout for a detached breakwater, where the optimisation consists of a cost function. The costs regarding the detached breakwater are the breakwater length and the most feasible location, where the breakwater x-direction and y-direction can vary, considering wave disturbance criteria and navigational constraints. Woerlee (2019) and Teeling (2020) constructed a model using a genetic algorithm that searches for an optimisation of the rubble mound breakwater, between downtime costs and CapEx and OpEx with a fixed starting point of the primary breakwater and a variable starting point of the secondary breakwater. The Teeling (2020) model uses detailed design guidelines, bathymetry and REFRAC, a refraction model to simulate wave fields, for the generation of alternative breakwater layouts.

Rustell (2013) investigated the use of a root-finding algorithm to solve complex breakwater layout problems in the conceptual design phase. The model optimises an LNG breakwater by minimising the downtime costs and CapEx and OpEx. The breakwater layouts take as input a bathymetry, wind and wave time series, the tidal range, and the x- and y-location of the berth. Therefore, the starting points of the breakwaters are fixed. From these starting points, the model searches for the optimal breakwater layout.

Table 2.3 shows the comparison of the four models, testing against the conceptual design requirements according to Ligteringen (2012) provides the potential gap of the models. The Teeling (2020) model is chosen to further investigate. The model optimises for rubble-mound breakwaters used for cargo ports. Furthermore, it contains most of the site data, and all design criteria are already present.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Vessel type</td>
<td>All</td>
<td>Detached Not defined</td>
<td>Detached LNG</td>
<td>Rubble-mound Cargo</td>
<td>Rubble-mound Cargo</td>
</tr>
<tr>
<td>Bathymetry</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Geotechnical</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Oceanography</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Topography</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Environmental impact</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Stakeholders</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Design criteria</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

2.4. Summary

Breakwaters are hydraulic structures that provide safe conditions in the port basin. The breakwater layout design is developed during the conceptual design phase, where time is scarce. Therefore, engineers mainly choose alternatives based on expertise. This limits the solution space to the expertise of the engineers. By translating the design process into an algorithm, a wider solution space can be investigated. Recent research proved the concept of design automation by the means of a genetic algorithm. This research focus on the improvement and increase of value for the Teeling (2020) model.
This chapter provides information about the Teeling (2020) model. In Section 3.1, the core of the model and the output of the model are discussed. In addition, it is investigated how the former developers validate the model. Furthermore, the possible points for improvement are mentioned. After which, a topic is selected. In Section 3.2, background information is provided for the positioning of the breakwater. Last, in Section 3.3, the essential aspects that a soil model should cover is explained.

### 3.1. Parametric breakwater model

The original model (Woerlee, 2019) is revised and extended by Teeling (2020). This section provides an introduction to the model, where possible additions are highlighted according to Teeling (2020). After that, the focus for this research is elaborated. Figure 3.1 presents the outline of this section.

#### 3.1.1. The Teeling (2020) model

This section provides an elaboration on the Teeling (2020) model. The Teeling (2020) model is a parametric model, more specific a Non-dominated Sorting Genetic Algorithm (NSGA-II), which engineers can use during the conceptual design phase and is a revised version on the work of Woerlee (2019). The model uses design criteria, site data and input values that the user specifies in advance. As a result, the model designs multiple breakwater designs and approximates an optimal design based on a fitness score. The fitness score consists of a CapEx, OpEx and downtime cost estimation and tries to find a trade-off between those.

#### Genetic algorithm

A conceptual breakwater design is a complex process in which multiple parameters are optimised. A genetic algorithm is suitable for such a multi-objective optimisation (Woerlee, 2019). The Teeling
model uses a Non-dominated Sorting Genetic Algorithm (NSGA), more specific the NSGA-II type. A NSGA-II select alternatives based on their rank relative to the Pareto front. A Pareto front is the situation when changing one parameter always influences the other parameters. For the breakwater layout model, changing the CapEx or OpEx will then influence the downtime costs. The NSGA-II type represents the exposed, multi-objective, Pareto-optimum breakwater problem well according to Woerlee (2019). The genetic algorithm uses the constant trade-off between downtime costs and CapEx and Opex to find the optimum breakwater layout.

**Input**
The model requires an input file, in which the site data and design parameters are provided. Appendix A provide an overview of all parameters adjustable in the input file. The site data adjustable are a bathymetry and wave field.

**Bathymetry** The bathymetry file contains the possibility to vary the x, y and z-axis. A grid of 10 m is chosen as the default value but is adjustable upon the client preferences. With the bathymetry, dredging and filling costs are estimated.

**Wave model** The Teeling (2020) model uses the REFRAC wavefields to model the wave height variation in the x- and y-direction. REFRAC is a refraction model that applies the geometrical optics approximation (Liu, 2009). For REFRAC, a bathymetry profile is needed to incorporate depth variations, and the offshore wave height and direction need to be defined (The REFRAC team, 2019). Furthermore, the model contains design criteria that are translated into equations. The equations that determine the wet and dry infrastructure of the breakwater model are provided in table 3.1.

<table>
<thead>
<tr>
<th>Element</th>
<th>Guideline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approach channel</td>
<td>PIANC (2014)</td>
</tr>
<tr>
<td>Turning basin</td>
<td>Ligteringen (2012)</td>
</tr>
<tr>
<td>Harbour entrance</td>
<td>PIANC (2014)</td>
</tr>
<tr>
<td>Manoeuvring area</td>
<td>Ligteringen (2012)</td>
</tr>
<tr>
<td>Breakwater cross-section</td>
<td>Briganti et al. (2004)</td>
</tr>
<tr>
<td>Terminal</td>
<td>Ligteringen (2012)</td>
</tr>
<tr>
<td>Downtime</td>
<td>Ligteringen (2012)</td>
</tr>
</tbody>
</table>

**Optimisation**
The optimisation process is carried out by a NSGA-II, in which the fitness score is used to select the dominant alternatives. A sum of CapEx and OpEx, and downtime costs determine the total costs of the breakwater generation. Equation 3.1 describes the fitness score \( f_f \) with the preferred total costs \( PC \), CapEx and OpEx \( f_c(X_i) \) and downtime costs \( f_d(X_i) \). \( X_i \) is called a chromosome and defined by the decision variables. During the elaboration on the model run, a more detailed explanation is provided.

\[
f_f(X_i) = \frac{PC}{f_c(X_i) + f_d(X_i)}
\]

CapEx and OpEx are merged to one parameter in equation 3.1 but are expenditures at two different stages. Therefore, the model considers them separate. CapEx are costs during the construction process of the breakwater and contain the breakwater construction costs \( C_{bw} \) and the capital dredging/filling, and land costs \( C_{dr} \). Breakwater construction costs are the costs related to the volume and configuration of the breakwaters. Configuration of the breakwaters depends on the environmental
conditions, design parameters and the input values discussed above. Capital dredging contains dredging costs for the harbour basin and approach channel. Re-use of the dredged material is an input parameter defined by the client as a percentage of total dredged material. OpEx are maintenance driven costs during the lifetime of the breakwater. The OpEx are divided into the maintenance dredging costs $C_{dr,o}$, which ensures the usability of the port, and the breakwater maintenance costs $\Phi_{bw,m}$, defined by a percentage of the total breakwater costs. As the lifetime $t$ of a breakwater is more than one year, the OpEx are discounted $r$. Equation 3.2 provides the CapEx and OpEx costs estimation.

$$f_c(X_i) = C_{bw}(X_i) + C_{dr}(X_i) + \sum_{t=1}^{T_L} \frac{C_{dr,o}(X_i) + \Phi_{bw,m} \cdot C_{bw}(X_i)}{(1 + r)^t}$$

Downtime costs are indirect costs of the breakwater during events when the port is not accessible. Operational and navigational downtime together forms the downtime costs. Operational downtime occurs during unsafe conditions for (un)loading at the berth, while navigational downtime refers to the berth or port’s inaccessibility due to severe weather. Operational downtime is split into wave action inside the harbour’s shelter part inside the breakwater and wind interaction with vessels and equipment for container handling. Wave interaction corresponds to the breakwater’s functioning, creating safe (un)loading berths inside the port. Waves are entering the port domain causing ship movement at the berth. Vessels have restrictions on the amount of ship movement during berthing, incorporated by the breakwater design. The design criteria for maximum ship movement depends on the vessel type. Wave action can occur in the harbour basin when partly or total failure of the breakwater occurs, and several stakeholders suffer from this:

- Shipping lines
- Terminal operators
- Port authorities
- Shippers/consumers
- Related service operators

Navigational downtime refers to the safe sailing of vessels entering and inside the port and mooring and unmooring of the vessels at the berths. Waves travelling in the port cause unsafe conditions for vessels at the berth or while sailing. Vessels entering the port often fasten to tugboats, fastening on average may not proceed above a wave height of 1.5m (Ligteringen, 2012). The model assumes that operational downtime occurs whenever navigational downtime occurs. Equation 3.3 provides the downtime costs estimation, where total downtime costs $C_{w,o}$ are discounted over the years.

$$f_d(X_i) = \sum_{t=1}^{T_L} \frac{C_{w,o}(X_i)}{(1 + r)^t}$$

To obtain the net present value (NPV) for the OpEx and the downtime costs, the total costs per year are discounted with $t$ equal to the design lifetime, making it possible to compare the three costs; CapEx, OpEx and downtime.

**Model run**

Understanding the core of the model is essential for the examination of the generations. Each generation bases its design on the following decision variables:

- A primary breakwater, consisting of three breakwater segments and four nodes. For the primary breakwater, a binary decision variable, the right-hand side (RHS) variable, provides information on which side the primary breakwater is constructed. RHS is randomly generated
for each layout design; if $RHS$ equals 0 the primary breakwater is designed in the negative x-direction.

1. Node 0 is located on (0,0).
2. Node 1 is located at $(o, y_1)$, where the x component is fixed at $x = 0$ and the y component is variable along the y-axis.
3. Node 2 is located at $(x_2, y_2)$, where the x component is a variable along the x-axis and the y component is variable along the y-axis.
4. Node 3 is located at $(x_3, y_3)$, where the x component is a variable along the x-axis and the y component is variable along the y-axis.

- A secondary breakwater, consisting of two breakwater segments and three nodes. A binary decision variable is given to the secondary breakwater construction ($SEC$). The values for $SEC$ are randomly generated for each breakwater generation. If SEC equals one, a secondary breakwater is constructed.

1. Node 4 is located on $(x_4, 0)$, where the x component is a variable along the x-axis, and the y component is fixed at $y = 0$.
2. Node 5 is located on $(x_5, y_5)$, where the x component is a variable along the x-axis, and the y component is variable along the y-axis.
3. Node 6 is located on $(x_6, y_6)$, where the x component is a variable along the x-axis, and the y component is variable along the y-axis.

- A turning basin which is variable along the x-axis and the y-axis $(x_{tb}, y_{tb})$.
- An approach channel with the orientation as a variable ($\theta_{ch}$)
- A manoeuvring basin ensuring space between the quay wall and berthing pocket.
- A berthing pocket and the quay wall location depend on the user’s preference.

A chromosome consists of all decision variables, see equation 3.4 (Woerlee, 2019). Each decision variable has an accuracy of zero decimals, and therefore, all decision variables are integers.

$$X_i = (RHS, x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, x_5, y_5, x_6, y_6, x_{tb}, y_{tb}, SEC)$$

Figure 3.2: Breakwater generation decision variables for $RHS = 0$ and $SEC = 1$ (Woerlee, 2019)
Random breakwater layouts are generated with all decision variables randomly generated within predefined limits. These limits are mandatory in order not to get interference between components. For each generation equation 3.1 estimates a fitness score. The fitness score determines the parents of the mating pool, where after cross-over and mutation, new populations are generated. Cross-over uses the decision variables from the selected parents to generate new layouts. A mutation changes the values of the decision variables to develop a new design. These new layouts are added to the original population, which is generated from this new group. This process repeats itself until the convergence criteria are met for the optima breakwater layout. Figure 3.3 shows an example of a complete model run, where the best design and the economic optima are displayed.

![Figure 3.3: An example of a breakwater layout generation with fitness score of 0.857 of the Teeling (2020) model and the economic optimisation for the breakwater layout model]

### 3.1.2. Validation

Woerlee (2019) developed the core of the model and tuned the optimisation parameters (mutation size, mutation rate and population size). For the core of the model several test cases were performed to check whether designs are drawn. However, it is not clear whether it has been verified and validated or it is not explicitly documented. Therefore, it is impossible to judge whether the model is internal validated. Also, it is not clear for what cases the model can be applied since only two case studies were performed. Therefore, the model is not broadly externally validated. Teeling (2020) further improved the model by adding a detailed bathymetry and wave model. However, only the two features are validated and again the whole model is not assessed. Teeling (Teeling, 2020) stated an important note regarding the consistency of the model:

Teeling (2020) advised to perform three to four runs as one out of three gives unreliable results.

### 3.1.3. Aspects to improve

Teeling (2020) stated that further research is needed to further increase the model. The research performed by Teeling (2020) highlighted points for further research. This research will discuss the following:

- Client preference implementation
- Various cargo vessels entering the port
- Nautical safety constraints and their value
- Downtime estimation

Next to the points highlighted by Teeling (2020), other parameters are identified to improve the Teeling (2020) model:
3. Parametric breakwater model and possible additions

- Positioning of the breakwater
- Validation and verification of the model
- CapEx and Opex optimisation
- Varying cost of capital
- Cost estimation improvement and CapEx modelling
- Port expansion

An elaboration on the mentioned further research possibilities is provided below.

**Client preference implementation**
Client preferences and constraints are of high importance to further improve the Teeling (2020) model. Teeling (2020) shows a difference in the obtained design and the conceptual design caused by budget constraints. Implementing the constraints of the client regarding downtime allowance, reducing the costs by allowing more downtime, creates layouts that fits better with the client preferences.

**Incorporate different cargo vessels entering the port**
The Teeling (2020) model focuses on container vessels entering the port. However, larger ports handle multiple types of vessels. All cargo types have their constraints on maximum wave height to enter the port and their berthing productivity. Implementing the different cargo vessels creates a more specified breakwater design.

**Nautical safety constraints and their value**
The approach channel and port basin must adhere to prescribed constraints to ensure the nautical safety for vessels. These nautical safety constraints partly determine breakwater layout. Implementing the different nautical safety aspects creates designs that better match the constructor's preferences. Moreover, the possibility to value the importance of the several aspects for the constructor will advance this process.

**Downtime estimation**
The Teeling (2020) model base the downtime estimation on navigational downtime, while the port may stay operational at the berth resulting in a higher estimated revenue loss. However, this assumption is not necessarily true. Therefore, increasing the accuracy of the downtime determination by the interaction between the navigational and the operational downtime, resulting in a higher accuracy of the fitness score.

**Positioning of the breakwater**
The present model assumes a pre-selected site, where the x coordinates and y coordinates are fixed values. However, the cross-shore and alongshore position of the breakwater can influence the CapEx, OpEx and downtime costs. Along the coast, different bottom profiles influence the breakwater costs. The steepness of the bottom, for instance, affects wave height at the breakwater. Cross-shore variation of the terminal and breakwater searches for an optimal position of the breakwater. When positioning a breakwater cross-shore, it may be possible to use the cut and fill method. The cut and fill method refers to the use of the dredged material as land reclamation. Moreover, the position is influenced by the soil type below the breakwater. Therefore, the implementation of a soil model that fits the conceptual design phase could advance the breakwater position with more detailed information where dredging is less. A soil model in earlier design phases has more impact than using
3.1. Parametric breakwater model

a soil model in the determination of engineering parameters (McDowell et al., 2002). Therefore, the implementation of the breakwater position influences multiple aspects in the model, such as wave height, length of the breakwater, dredging volume, and cross-section of the breakwater.

Validation and verification of the model

The components of the breakwater model are validated by face validity, which means only judged on sight. Furthermore, the complete model is only validated by a case study and, therefore, it is unclear for what cases the model is applicable. To create more transparency for the tool, a validation study is possible. The validation of the complete model should focus on the internal and external components.

Varying cost of capital

The cost of capital for a structure represents the required return to make it worthwhile. It is used for comparing different investments, where the return of the investments must exceed the cost of capital. The cost of capital is built upon the cost of debt and the cost of equity of a company or section. As the breakwater can be placed in a special purpose vehicle (SPV), the cost of capital may change from the overall port’s perspective.

Cost estimation improvement and CapEx modelling

Three important reasons for cost estimation are managing costs, making strategic and operational decisions and planning and setting standards (Wouters et al., 2012). The first two statements are important for a breakwater cost estimation. CapEx/Opex is now estimated to add up the breakwater costs, capital dredging/filling and land costs, and discounting the maintenance costs and the breakwater over the years. However, costs for a breakwater are based upon more aspects. Adding all aspects together will create a more precise calculation of costs. Managing the costs for a breakwater provides insight into the activities and costs for CapEx/Opex modelling. The different breakwater layouts generated by the model have different costs. Decision making based on costs assists in differentiating between the designs. Therefore, creating a model for the CapEx will increase understanding of costs during the lifetime of the construction.

Developing a coupling between the damage parameters for the breakwater (a design choice) and the impact on maintenance requirements – hence a CapEx/Opex optimisation. Possibly link this to a Power BI dashboard to show degeneration and cost, maintenance interval, functionality, etc.

Port expansion

The Teeling (2020) model design breakwaters which come from scratch. Making it possible to use the existing port and breakwater as a fixed part and let the model design an expansion on the current port/breakwater.

3.1.4. Focus

Based on the point for further research discussed above and after discussing the model with professionals from Arcadis, it is chosen to focus on verification (debugging), internal and external validation, the performance of the breakwater model and implement the alongshore position of the breakwater into the model.

The potential improvements on the cost aspect are considered less important than validation of the model and modifications related to the positioning of the primary breakwater. The costs are detailed enough for a conceptual design and therefore eliminated for further investigation. The experts reviewed the Teeling (2020) model and suggested focusing on validating the model to increase the value of the model and increase confidence among engineers. Another aspect is the alongshore positioning of the primary breakwater, which increases the solution space of the model to find the
optimal position and layout. For the alongshore position, the model uses bathymetry, wave data and the already written code.

3.2. Positioning of the breakwater
The literature on the positioning of a breakwater is scarce, but broadly speaking, it resembles the same aspects considered during site selection of a port. Better site investigation results in a greater chance of project viability, preventing increasing costs, lowering construction time and OpEx, taking care of the health of employees and residents, and the environment (Clayton & Smith, 2013). The main drivers of site selection in the conceptual design phase are bathymetry, environmental impact, geotechnical, oceanography, topography (UNCTAD, 1978), see Figure 3.4.

The figure above highlights the aspects mentioned in subsection 2.2.2. Site data that deal with the position of the breakwater are bathymetry, geotechnical data, oceanography and topography. Also, the environmental study is important to investigate what impact the breakwater has on the environment. To create a complete conceptual design of a breakwater, these aspects are essential for the breakwater tool. Moreover, the optimal position of a breakwater depends on these aspects. An elaboration of how these aspects influence the position of the breakwater is provided below.

**Bathymetry**
Water depth and bottom slope determine the location of the port and breakwater. If building on steep slopes, water depth increases rapidly, which increases the costs of the breakwater elements as cross-section volume increases. With an increasing breakwater volume, the CapEx and OpEx increase as the volume increases. Moreover, water depth, wave height and wave energy are correlated. The decrease in water depth causes waves to break and decreases wave height. The Teeling (2020) model translates an offshore wave height to an onshore wave height. With the onshore wave height, the model generates possible breakwater layouts.

**Geotechnical**
Investigation after the geology of the possible sites is the base for the geotechnical assessment. A geology study provides information about the soil composition, which influences dredging costs and
3.2. Positioning of the breakwater

Geotechnical investigations are mandatory for maritime structures and land-based construction such as terminals to provide information about the load-bearing capacity and soil loading with accompanying viable structures and furnish information about foundations for the structure (Ezenwaka et al., 2014). The three main drivers for the layout of a breakwater concerning dredging and re-use of the material are (PIANC, 2000):

1. Volume of dredged material and the dredged material distribution.
2. Obtain the physical and mechanical properties of the soil, which influence the transport and dredging processes.
3. Whether the dredged material is possible for the cut and fill method.

In addition to answering the above aspects, the environmental assessment also uses the geotechnical study.

Oceanography

The most important oceanographic details for port planning are waves statistics, currents, tides and sedimentation. This research will only cover the wave data for the conceptual design phase. The nearshore wave data is obtained by translating the offshore wave data into onshore wave data. Wave period, wave height and wavelength are decisive factors determining the breakwater design. Teeling (2020) investigated three tools to estimate onshore wave data by offshore wave data; SwanOne, APEX wave ray and REFRAC. However, not only does wave data at the breakwater seaside influence breakwater design, but also wave overtopping, reflection, and transmission disturbs tranquillity in the harbour basin and therefore influence the breakwater alignment to reduce these effects (Boshek, 2009; Van der Meer & Sigurdarson, 2016).

Topography

The ease of hinterland connections to the port is one of the main drivers for site selection. With an increasing vessel size and increasing cargo volume, hinterland connections are of high importance regarding the competitiveness of the port (Acciaro & McKinnon, 2015). This, therefore, influences the port position alongshore and cross-shore. Also, the availability of land around the port is vital for the terminal.

Environmental impact

The construction of a port and breakwater requires dredging and dumping of materials. Dredging and disposal of these materials can cause plumes in the water due to suspension of fine materials (PIANC, 2016). Therefore, restrictions are set on the concentration fines in the water column. Besides the plumes, the design must take environmental damage to fish, plants and underwater structures into account (PIANC, 2010). If it is possible to position the port and breakwater somewhere with little or no impact, this is preferred. The conceptual design phase highlights these major construction elements on which the environmental assessment is based on (PIANC, 2009).

The models objective is to obtain an optimal location by striking a good balance between the aspects discussed above. Teeling (2020) already implemented oceanography and bathymetry in the breakwater model. Therefore, the alongshore positioning can be implemented. Therefore, by first letting the model search for the optimal location alongshore provides an answer to what the model is capable of in searching for the optimal location. If the alongshore position is working well, the cross-shore position is an addition to find the optimal location for the breakwater considering dredged and fill material together with the bathymetry and wave climate. To create a valuable cross-shore positioning model, a soil model is a possible addition. The soil model provides information about the soil conditions.
3.3. Soil model

Site investigation creates a more cost-effective design (Albatal et al., 2013). One of the highest unforeseen costs during construction arises because ground conditions are not estimated correctly (McDowell et al., 2002). A soil model search for the best location for the breakwater regarding the subsurface, where less dredging is required, or optimal use of dredged material is available (PIANC, 2000).

A soil model creates the opportunity to weigh up different positions for the breakwater. In the conceptual phase, the soil model must give information on the soil conditions and the soil layer thickness to clarify whether soil improvement is necessary and to what extent. The data used for the soil model is obtained during geology and geotechnical investigation. Such models provide information to the user if a potential risk occurs for the project (Parry et al., 2014). A soil model in the conceptual design phase must provide the depth, thickness and composition of the soil (Look, 2014). This data is available through a few cone penetration tests (CPT), single penetration tests (SPT) and visual observation. It is essential to check that the model covers enough data to determine the ideal position of the breakwater and that enough accuracy is guaranteed. During the conceptual phase, a soil model must provide a first indication about the subsurface, where positioning of the breakwater can rely on.

The soil type determines the strength of the subsoil and indirectly the costs associated with reinforcement or dredging. A place where minor dredging or soil improvement is necessary reduces CapEx for the breakwater and increases attractiveness. Therefore a good understanding of the possible soil types alongshore and cross-shore is essential, considering the following:

- Rock
- Sand
- Clay
- Silt and mud

Dredging and the possibility to apply the cut and fill method depends on the soil composition. Preference is given to build a breakwater on a hard bottom. If there is a soft layer above, this soft layer have to be dredged away (Sciortino et al., 2010). A vital soil property is the bearing capacity which determines the design conditions and is a limiting factor for the operations (J.P. van den Bos, 2018). The bearing capacity of a soil indicates the maximum load such that the ground does not exceed the ultimate settlement and no shear failure occurs. Four main soil mechanic problems related to the bearing capacity are sliding, squeezing, liquefaction and settlements. However, settlement problems are tackled during the final design phase and, therefore, out of the scope of this research. An elaboration on each parameter and how this affects the breakwater position is provided below, following the definitions of J.P. van den Bos (2018).

**Sliding**

Sliding is the process of a breakwater applying pressure on the subsoil with its weight, causing instabilities of the soil beneath on the front side or at the side slope. When designing a breakwater on a surface where the top layer is soft soil, a solution must be found for the risk of sliding. A soil model has to provide information about the dredging volume needed to reach the bottom layer with proper bearing capacity and consider the most optimal location.

**Squeeze**

By the combination of very weak subsoil and sturdy breakwater material, the very weak subsoil emerges instability. Deformation occurs below the construction, where over-stressing causes the
soil to deform without changing its volume. The stress transfers to the surrounding, and the breakwater sinks into the subsoil, called a squeeze. If squeezing is estimated as plausible during the concept design, a choice can be made to apply soil reinforcement or to include additional costs to the design.

**Liquefaction**
The risk of liquefaction occurs in the presence of loose sand. The power of sand lies in the force transfer from grain to grain. However, if there is a disturbance near the breakwater, there is a chance that the grains will move and water will drain from the pores. Water cannot drain from the pores in low permeable sand, and the body loses its strength.

Additionally, adding environmental factors to the soil model generates a tool that searches for the optimal location regarding geotechnical and environmental aspects.

### 3.3.1. Soil characteristics
There is a link between the above-mentioned soil mechanic problems and the soil characteristics. Soil characteristics are obtained during a site investigation and give insight into the bearing capacity of the soil. The following sections explain the soil characteristics that relate to the bearing capacity are (Roy & Bhalla, 2017).

**Specific gravity**
The specific gravity is the density of a substance related to the density of a reference substance (during this research, water is the reference substance, with \( \rho_w = 1000 \text{ kg/m}^3 \)). For higher specific gravity, the bearing capacity of the soil also increases (Maghvan et al., 2019). With an increasing bearing capacity, the soil becomes more suitable as a construction material.

**Consistency limits**
Consistency of a fine-graded soil refers to the cohesion of materials. Soil consists partly of water; when the water content in the material increases, the soil type will move from liquid to plastic to semi-solid to solid. Three transitions occur with each of their consistency limits: liquid limit, plastic limit and shrinkage limit.

**Particle sieve analysis**
A soil has specific particle distribution, obtained by a particle sieve analysis. The particle distribution indicates a granular material's particle size distribution, which helps identify suitable soils for a foundation. A well-graded soil contains particles with a wide range of sizes, while poorly graded soils have a narrow range. Soils with a well-graded distribution provide more strength than poorly graded.

**Compaction**
The others refer to the soil characteristics, compaction is a ground improvement method, increasing shear strength and bearing capacity and reducing settlements.

**Consolidation**
If a structure exerts pressure on the ground, the pore water pressure will increase, and the soil will shrink. The consolidation test is functional to check a structure founded on clay for possible settlements.
**Permeability**
It relates to the stability of the structure, which is not the goal of a soil model. Therefore, permeability is out of the scope of this research.

**Shear strength**
Shear strength is the maximum shear load a soil can withstand before shear failure occurs. It is the most important aspect of soil strength and provides information for the usability of the soil for engineering purposes.

### 3.4. Summary
This chapter focused on The Teeling (2020) model, which is a parametric model based on an NSGA-II, that support the engineer during the conceptual design of a breakwater. The model searches for the best breakwater layout regarding the technical requirements, bathymetry and wave data. A fitness score is drawn to compare the generations and covers the CapEx, OpEx and downtime costs. To increase the value of the model, further research is essential. It is chosen to implement the alongshore positioning of the primary breakwater. As the model include the bathymetry and wave data, the essential aspects for the alongshore position are present. Furthermore, this research focuses on verification (debugging), validation and performance to increase the model's value.
In Chapter 3 the Teeling (2020) model was explained and the possible additions were mentioned. This chapter takes a closer look at the genetic algorithm itself to provide information on the working of the code behind the model. Furthermore, a suitable software development method is selected to support the increase of the model’s value.

4.1. Genetic Algorithm

The genetic algorithm is inspired by the Darwin theory where natural selection plays a role (Kureichik et al., 2009). Figure 4.1 represents a typical genetic algorithm flowchart for optimisation purposes. From a starting population, the fittest individuals are selected to reproduce the new generation, called parents. After that, creating alternatives using crossover and mutation, whereby the worst alternatives are filtered out by selection. A new generation with the fittest individuals is selected. The breakwater is seen as a individual, consisting of a set of genes, where the position and dimensions are adjusted by means of the Darwin theory. The best alternatives are used for the generation of a new population and mutated towards an optimum. If the design criteria are met and convergence has reached, the genetic algorithm stops and returns the best.

Figure 4.1: Genetic algorithm process

From Chapter 3 it is clear that a genetic algorithm is a bunch of equations that require input data to come up with possible solutions or, for the breakwater layout model, possible designs. In addition to the equations and input, the optimisation parameters should be tuned to increase the model’s performance. The optimal values for the optimisation parameters differ per model and, therefore, cannot be obtained from the literature. Tuning attempts to ensure that the model quickly generates solutions while it does not tend towards a local optimum. A local optimum causes the model to search for possible solutions within very limited boundaries. Since the model is intended to explore
the largest possible solution area, a global optimum is preferred. The parameters considered are:

**Mutation rate** The frequency of mutation in a certain gene. For high mutation rates, the genetic algorithm easily escapes local optimum (Hassanat et al., 2019). However, the computation time of the model will increase enormously. Low mutation rates will increase the generation of the optimal solution, but this is probably only a local optimum.

**Mutation size** The standard deviation to the mutation rate. By increasing the mutation size, more significant differences between the alternatives are created. Therefore, the model should avoid a local optimum, but it is also possible that a less optimal solution is selected. A small mutation size can cause the model to reach a local optimum.

**Population size** The number of individuals representing the alternatives. By selecting a higher population size, the model probably covers a broader solution space. However, each extra generation increases the computation time.

### 4.2. Methods

To gain insight into a developed model, several software development methods are assessed. Software is a set of instructions that tells a computer what to do. The development of software encompasses computer science and focuses on creating, designing, and deploying the software. By selecting the appropriate software development method, the genetic algorithm should be kept in mind. This research focuses on verification and validation of the model, investigating its performance, and increasing its applicability. It is therefore preferred that the method can support the developer with all research objectives. The following software development methods are selected to discuss:

- Agile development
- Waterfall development
- Rapid application development

**Agile development**

Agile development covers many smaller development methods, like test-driven development, scrum, extreme programming, feature-driven development, etc. Agile development is an iterative development process. The following phases are common for an agile development method:

1. Gather technical requirements
2. Design the model
3. Develop the code
4. Testing the code
5. Development
6. Review

This research will focus on the test-driven development method as this is one of the most common agile development methods (Khanam & Ahsan, 2017).
4.2. Methods

Test driven development  The goal of test-driven development (TDD) is to develop a model that is focused on testing different cases and giving insight into the model (Beck, 2003). The first step in TDD is to develop the different test cases. After that, the code is written and validated by the test case. If the test fails, the code has to be rewritten. By repeating these steps, more insight is gained into the model and its limitations of the model. If all tests pass, the code is refactored, meaning that the code is rewritten following programming rules and the duplicate code is removed.

Waterfall development

The waterfall development method is a linear model with sequential phases, where each phase must be validated before going on to the next phase (Balaji & Murugaiyan, 2012). Figure 4.2 shows the different phases of this method. The requirements phase focuses on the need for the tool by the user. Next, the design phase realises the needs and writes the code in a programming language. After the code is ready, the implementation phase focuses on validating the written code by testing the smaller parts and the overall code. The verification phase quantifies the results obtained by the code. Last, the maintenance phase focus on the user. In the maintenance phase, customers need to install and apply the model.

Rapid application development

Rapid application development method (RAD) divides smaller tasks into teams to develop a tool. In the end, all the different tasks are coupled to create one working model. The first phase of RAD is the gathering of a general requirement for the model. With this requirement, specific requirements are created for different stages in the development. The next step is the user design. During this phase, multiple prototypes are built. The user then determines which features and functions of which prototypes are used for further development, which is the next step in the development of the model. Knowing the user preferences, the model is built. The last phase focuses on the finalisation of the model, covering all aspects of the model.

4.2.1. Comparison

For a complete overview of the advantages and disadvantages of the software development methods, they are compared to several criteria for the development of a conceptual breakwater model. These criteria also take into account the purpose of the research. Table 4.1 shows this comparison.

Table 4.1: Comparison of the different software development methods considered. A ‘+’ means the model is favourable, ‘-’ means not suitable for the objective of this research

<table>
<thead>
<tr>
<th>Method</th>
<th>Changing requirements</th>
<th>Transparency of the code</th>
<th>Validate the model</th>
<th>Validate model components</th>
<th>Refactoring</th>
<th>Meant for</th>
<th>Experience level</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDD</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>Team or one person</td>
<td>Beginner</td>
</tr>
<tr>
<td>Waterfall</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>Team or one person</td>
<td>Beginner</td>
</tr>
<tr>
<td>RAD</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>Team</td>
<td>Skilled</td>
</tr>
</tbody>
</table>

Knowing the current research is carried out by one person, RAD is already cancelled out. Furthermore, this research focuses on refactoring a current model, making the waterfall method also not applicable. Therefore, test-driven development method is estimated most suitable for this research.
4.2.2. Test-driven development

The model is already written to a large extent. However, adjustments and validation test cases are necessary to verify, internally and externally validate, assess the performance and implement new features to the model. TDD assists with creating targeted test cases and creating transparency for the model. Figure 4.3 shows a typical framework for test-driven development.

The first step for test-driven development is the writing of several test cases. If these test cases create generations that do not match the requirements, the code will be rewritten to produce usable values (Ivo et al., 2018). After rewriting the code, it should give usable values. It does not matter how it is written at this point, as long as it works. The next stage, refactoring, focus on the readability and maintainability of the code. The steps discussed above are constantly repeated for software development.
This chapter explains the use of test-driven development, selected in Section 4.2, to the research objectives: increase the applicability, verification, internally and externally validate and improve the performance of the model. First, in Section 5.1 the implementation of the alongshore position is explained. Then, in Section 5.2 the method to verify the model is explained. Next, in Section 5.3 the validation method is elaborated. After that, in Section 5.4 the method is explained to improve the performance of the model. Finally, the boundary problem for REFRAC is elaborated in Section 5.5.

5.1. Alongshore position

It is chosen to further improve the model’s applicability by implementing the alongshore positioning of the primary breakwater. TDD principles are applied to implement the alongshore position of the primary breakwater. A test case is developed whereby a favourable location regarding total costs exist for $x_0 > 0$. The primary breakwater in the Teeling (2020) model was fixed at $x_0 = 0$, as shown in figure 5.1 by the red arrow. Therefore, it is expected that the model will fail for this test. Adjustments to the code are needed to create the opportunity for the model to place the primary breakwater at $x_0 > 0$. An explanation of the implementation is provided below.

5.1.1. Implementation

Section 3.1.1 discusses the different elements of the Teeling (2020) model. The breakwater segments are generated based on certain values that the genetic algorithm assign to them. From the seven nodes that form the breakwater layout, two nodes have fixed values defined in the python

![Figure 5.1: A result of the Teeling (2020) model, whereby the primary breakwater is fixed at $x_0 = 0$ (red arrow)](image-url)
Figure 5.2: Top down method to breakdown the model to find the essential changes to the model for position the breakwater alongshore and in what sequence it must happen. The dotted rectangle highlights all places where to change the alongshore coordinates of the nodes. The mating, mutation and generator sections are part of the genetic algorithm code: node 0 and node 1. The possibility of these nodes becoming variable will cause the model to search for the optimal alongshore position. Table 5.1 indicates the different nodes and if their x- and y-values are variable or fixed. The model is broken down to understand where each node is defined. A suitable way for doing this is by applying the top-down method. This method tackles the bigger problem by dividing it into smaller steps. Figure 5.2 visualises the top-down application for the position of the breakwater. The first step in finding the optimal location for the breakwater is to create a model with free variables along the shoreline.

Table 5.1: The breakwater nodes and if they are fixed or variable

<table>
<thead>
<tr>
<th>Node</th>
<th>( x_0 )</th>
<th>( y_0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>fixed</td>
<td>fixed</td>
</tr>
<tr>
<td>1</td>
<td>fixed</td>
<td>variable</td>
</tr>
<tr>
<td>2</td>
<td>variable</td>
<td>variable</td>
</tr>
<tr>
<td>3</td>
<td>variable</td>
<td>variable</td>
</tr>
<tr>
<td>4</td>
<td>variable</td>
<td>fixed</td>
</tr>
<tr>
<td>5</td>
<td>variable</td>
<td>variable</td>
</tr>
<tr>
<td>6</td>
<td>variable</td>
<td>variable</td>
</tr>
</tbody>
</table>

The implementation of the free variables is provided in figure 5.2. There are two ways to let the
model search for an optimal alongshore position:

1. Let the model define an x-value, with a uniform distribution, before the genetic algorithm starts. However, with this implementation, the x-value is not inserted in the genetic algorithm meaning, which means that the genetic algorithm will not use that value for the next generation. At the end of the run, the location with the highest fitness score is used as starting value for the next run. This will possibly speed up the design process but pry with the idea behind an optimisation tool and enormously increase computation time.

2. Put the x-value inside the genetic algorithm. Suppose \( x_0 \) and \( x_1 \) are inside the genetic algorithm, the model rates the position of the x-values for the following generation. The running time will probably rise a bit, but the concept of a genetic algorithm and optimisation tool remains.

Since this is an optimisation tool based on an algorithm and efficiency is expected, option one is only used to test if the breakwater nodes can vary. If there is an opportunity to implement the variable breakwater nodes inside the algorithm, option two is preferred.

A genetic algorithm comprises several classes, the effect of an adjustment is examined for each layer in the model. Therefore, the first step is to highlight the variables responsible for the model to generate the primary breakwater at \( x = 0 \). The classes that affect the alongshore position of the breakwater are generator, mating and mutation classes. Changing the position is carried out at every class at once so that no errors occur and the model does not overwrite a position defined by another class. The model is set up in such a way that the starting value of node 0 and 1 are \( x_0 = 0 \) and \( x_1 = 0 \). To investigate whether it is possible to change this value and what consequences this has on the execution of the model, the values are changed to a value bigger than zero. If the possibility arises for this model to have a \( x_0 \) and \( x_1 \) different from zero, the next step is to apply the genetic algorithm to the x-values. For the implementation of the x-values, a grid similar to the other x-values is applied. As Teeling (2020) based the wave processes on the grid of Woerlee (2019) no distinction may arise by the addition of the positioning of the breakwater alongshore. Therefore, the rectangular grid developed by Woerlee (2019) with a step size depending upon the user, but usually 10 meters, is applied for the variable breakwater nodes.

The final step of the implementation is the validation of the alongshore positioning. A test case whereby the bathymetry has a favourable location in the middle of the x-axis should test whether the model can place the primary breakwater there.

5.2. Verification

Before further work can be done on the model and to increase confidence for it, verification of the model is necessary. Verification focus on whether the model function as expected and not suddenly terminate. Therefore, the model should be free of bugs. Without verification, adjustments made to the model, without knowing whether it contains errors, it can be mistakenly thought that the adjustment leads to an error. In addition, verifying the model provides more knowledge of the code needed for the next steps of this research.

TDD support by debugging and refactoring the code. As the code is already written, a test case is created that contains linear bathymetry and waves coming from two directions. The creation of the bathymetry and generating the wave fields can be done in a separate Python file, which requires an excel file\(^1\). The linear bathymetry is saved under the tab 'Homogeneous Slope'. By running that Python file, multiple files are created. With these bathymetry files, REFRAC can be run, also done

\(^1\)At the time of publication, a public document is available from where the model can be run.
5. Research method

in a separate Python file. Using this bathymetry and wave data, a run is executed multiple times. By adding breakpoints to the code, the execution is interrupted at specific points defined by the developer. This helps by splitting the code into smaller parts to ensure that no errors occur until that specific point. Once the model is clear of bugs, the code is refactored.

5.3. Validation

The goal of validating the model is to add reliability to the model and its results and thereby increase the model’s value. The reliability of a genetic algorithm increases when the model shows explicable results and cost consistency during different runs for the same test case. A distinction is made between internal and external validity. Internal validity indicates whether there are no other factors than the input that influence the results of the model (Ruthruff et al., 2010). External validity shows that the model applies to the areas in which validity is obtained (Ruthruff et al., 2010). Both internal and external validity requires cost consistency between the tests. However, the genetic algorithm is a black box, dependent on multiple variables, making it almost impossible to get the same results. Therefore, this research assumes that the model can find consistent results if the model produces five runs within a cost accuracy of 10% to each other for the same test case.

While validating the model, a focus must lie on the coordinates of the approach channel, primary (and secondary) breakwater, manoeuvring basin, terminal and turning basin. However, the model may generate layouts that are not thought of beforehand; it will then have to be investigated whether these designs are explicable by the applied equations. For example, the model could design a breakwater layout where the turning basin is outside the breakwater; this is not common among engineers. But since all technical requirements according to Briganti et al. (2004), Ligteringen (2012) and PIANC (2014) are present and the bathymetry and wave data are incorporated, it is possible to explore whether this could be a possible alternative. It can provide a new view of the already existing ideas about breakwater layout and port design.

5.3.1. Internal validation

The model consists of design criteria and input data whereby the algorithm finds possible breakwater layouts. However, the former developers did not internally validate the model. Applying TDD and, therefore, develop a base case to internally validate the model. The base case focus on whether the input is translated into the output. The model has multiple input values that all generate a piece of the breakwater layout. As all elements together generate one possible breakwater layout, every element has to exactly generate what it should obtain as a working breakwater model. Table 5.2 shows the elements of the model and what they must generate or which guidelines they should follow as provided in Chapter 3. The first thing in creating more value for the model is to test that all elements are printed and follow the specified guidelines. If not, the model will be changed until it translates the guidelines in the result.
Table 5.2: The guidelines applied for the various elements of the port layout

<table>
<thead>
<tr>
<th>Element</th>
<th>Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approach channel angle</td>
<td>Approach channel angle should follow guidelines by PIANC (2014).</td>
</tr>
<tr>
<td>Approach channel dimensions</td>
<td>Approach channel dimension should follow guidelines by PIANC (2014)</td>
</tr>
<tr>
<td>Breakwater</td>
<td></td>
</tr>
<tr>
<td>Primary</td>
<td>Breakwater nodes &amp; breakwater segments Briganti et al. (2004)</td>
</tr>
<tr>
<td>Secondary</td>
<td>Breakwater nodes &amp; breakwater segments Briganti et al. (2004)</td>
</tr>
<tr>
<td>Manoeuvring basin</td>
<td>Turning basin dimension should follow guidelines by PIANC (2014)</td>
</tr>
<tr>
<td>Terminal</td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>Predefined by the user</td>
</tr>
<tr>
<td>Berths</td>
<td>The amount of berths are defined by the user upon running the model</td>
</tr>
<tr>
<td>Quay wall</td>
<td>Quay wall coordinates should follow coordinates of the primary breakwater + required berth space by Ligteringen (2012)</td>
</tr>
<tr>
<td>Turning basin</td>
<td>Turning basin dimension should follow guidelines by Ligteringen (2012)</td>
</tr>
</tbody>
</table>

The test should validate:

- The manoeuvring basin and approach channel dimensions follow the design criteria of PIANC (PIANC, 2014).

- The model plot all nodes and breakwater segments as intended according to Briganti et al. (2004).

- The terminal area, berths and quay wall and turning basin follow the design rules of Ligteringen (Ligteringen, 2012).

In addition to the validation of the various elements, the complete model is assessed. The layouts are rated by face validity, which judges the generations created by the model based on expertise. In addition, the CapEx, OpEx and downtime costs are used to compare different layouts on cost consistency. These costs also provide the developer insight into the estimated costs by the model.

5.3.2. External validation

External validation focus on whether the model is applicable for a wide range of cases. Therefore, various test cases are developed to test and validate the model's functioning by assessing the recognition by the model of the input data. The input data refer to the bathymetry, wave data, and input excel file.

To ensure that the test cases only study the specific aspect, it is essential to define the boundaries for each test case. The various test cases designed for the tool assess:

1. Waves

2. Bathymetry

The wave test cases focus on the variability of wave height and wave direction. For the bathymetry, the bottom slope and depth profile are checked. No specific sequence must be followed in running the different test cases, but this research will first focus on the wave variation.

Waves

The first aspect to investigate is the capability of the model to search for the best wave field and align the breakwater such that wave influence inside the harbour basin is the least. The first step in the designing of the test cases is to specify boundaries. The wave test case should only differ in wave height alongshore and cross-shore without any other aspect influencing the alignment and positioning of the breakwater. As REFRAC is not capable of varying wave height alongshore, only the
recognition of the breakwater model against varying wave direction is validated. Two wave scenarios are tested, in which waves come from -30° or +30°. For the wave cases, linear bathymetry is applied.

**Bathymetry**
The bathymetry test cases are divided into a slope variation alongshore and a sudden-deepening cross-shore. The bathymetry influences the wave field. Therefore, some experiments will test the overall impact of changing bathymetry, thus with a wave field influenced by the bathymetry. While another test only focus on the bathymetry differences while applying a uniform wave field. Such wave field is created by using a linear bottom profile.

The slope variation affects the dredging CapEx and OpEx, but also the wave height at the coast. One of the test cases only focuses on the slope variation on its own; therefore, to obtain uniform wave data alongshore, run REFRAC with a linear slope and waves coming from two directions to the coast to prevent an optimal wave location. With no favourable location regarding the wave field, the model’s ability to recognise a favourable location concerning the bathymetry is tested. A uniform wave field with an asymmetric slope will test the ability of the model to locate the optimal breakwater layout regarding the bottom profile. Comparing the outcome of the different test cases provides an answer if the model recognises the difference in bathymetry and wave field. Furthermore, a test case with a sudden-deepening profile can test the model to recognise optimal locations alongshore and cross-shore. If alongshore a sudden-deepening occurs, the model is expected to select the position where its less deep.

**5.4. Performance**

**5.4.1. Usability**
A hardly understandable model can scare the use of it. Therefore a user manual will be made for the breakwater layout model. The user manual should contain the installation and execution of the model and the core of the model. The breakwater model is a complex model, which requires certain actions in advance to execute it. Furthermore, one file will be created to ease the use of the model and provide structure for it by storing the case files in one directory.

**5.4.2. Efficiency**
Tuning the optimisation parameters and using the optimal computer performance unit (CPU) result in higher efficiency for the genetic algorithm. Therefore, TDD is used to develop and run a base case to tune the optimisation parameters. The genetic algorithm parameters are adjusted during these runs to observe the differences in computation time and accuracy, where accuracy is measured as a fitness score. The goal is to find a trade-off between the computation time and accuracy, where accuracy hardly, but computation time decreases (Aslansefat, n.d.).

In addition, the use of Dask will increase the optimal use of the CPU. Dask is an open-source python library and enables the user to compute parallel by running the model on multiple cores (Böhm & Beránek, 2020). This will possibly lower the computation time of the model without influencing the accuracy. By the implementation of Dask, the number of cores is adjustable. The amount of cores available depends on the random-access memory (RAM) available in the computer. When working on a local computer, the optimal number of cores can be tested. Using the optimal number of cores will lead to lower computation time. Dask is implemented by installing the Dask environment and addition of it to the Python code.

**5.5. REFRAC**
REFRAC has a boundary problem that causes waves to pile up at the extremes of the bathymetry boundaries. If the model uses the whole bathymetry, unrealistic values for the wave height at the
5.5. REFRAC

boundaries exist, as shown in Figure 5.3a (red circle). This figure presents waves coming from the right where it can be observed that the waves at the left boundary pile up. This boundary problem emerged as REFRAC cannot interpret that the coastline continues after $x = 0$ m. To correct for this error in REFRAC, the boundary values of the models solution space were set at $x = 3000$ m, $x = 7000$ m, $y = 0$ m and $y = 4000$ m. The accompanying search area in the bathymetry is shown as a red dashed rectangular in Figure 5.3b.

(a) Wave height corresponding to a linear slope. The red circle presents the boundary problem where waves pile up.

X-axis: alongshore distance [m]; y-axis: cross-shore distance [m]; colour bar: wave height [m]

(b) Linear bathymetry. The red dashed rectangular represents the boundaries of the search area of the model

Figure 5.3: Wave field generated by REFRAC for a linear bathymetry.
This chapter presents the results from the methodology as described in Chapter 5. In Section 6.1, the alongshore position of the breakwater is implemented and validated. Whereafter, in Section 6.2, the results of verification of the model are presented. Then, in Section 6.3, the model is internal and external validated. Last, in Section 6.4, the performance of the model is assessed. The following applies to all results:
All test cases have been run five times to test for consistent fitness score, as discussed in Section 5.3.

Furthermore, Table 6.1 provides information for the default values applied to all test cases. The design vessel size determines the length and width of the water areas and quay wall. The soil type influences the dredging and filling costs; for the test cases, sand soil is applied. The offshore wave height belonging to these test cases equals 2 m. To calculate the fitness score, a total preferred total breakwater cost of 180,000,000 euros is used.

<table>
<thead>
<tr>
<th>Element</th>
<th>Amount</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Design vessel</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Length overall</td>
<td>300</td>
<td>m</td>
</tr>
<tr>
<td>Beam</td>
<td>32</td>
<td>m</td>
</tr>
<tr>
<td>Draught</td>
<td>12</td>
<td>m</td>
</tr>
<tr>
<td>Entrance speed</td>
<td>3</td>
<td>m/s</td>
</tr>
<tr>
<td><strong>Number of berths</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Present phase</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>Future</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>Soil type</td>
<td>sand</td>
<td>-</td>
</tr>
<tr>
<td>Coastline orientation</td>
<td>0</td>
<td>°</td>
</tr>
<tr>
<td>Terminal length</td>
<td>200</td>
<td>m</td>
</tr>
<tr>
<td>Breakwater total costs</td>
<td>180,000,000</td>
<td>€</td>
</tr>
</tbody>
</table>

6.1. Alongshore positioning of the breakwater
The first step in the positioning of the primary breakwater was to check the possibility of the model to vary alongshore. As mentioned in Section 5.1.1, the first step was forcing the model towards an x-value other than zero. By doing this, multiple errors arose as shown in Figure 6.2a. The alongshore position of $x_0$ and $x_1$ was used for the definition as starting points of terminal/quay wall and
the primary and secondary breakwater. Therefore, intersection occurred between the breakwaters and the water areas (red rectangle in Figure 6.2a), where the primary breakwater intersect with the approach channel. Furthermore, the two breakwaters sometimes intersected with each other (red circle in Figure 6.2a). Furthermore, the starting and ending points of the terminal did not move with the primary breakwater. This resulted in a shorter terminal than should (red circle in Figure 6.2a). These errors were caused by the fact that the coordinates defined by the genetic algorithm are uniformly distributed, all depending on the starting points of the terminal and quay wall, which depends on the breakwater coordinates $x_0 = 0$ and $x_1 = 0$. To solve these bugs, all hard coded parts were replaced with $x_0$ and $x_1$, shown in Figure 6.1.

![Image](a) Hard coded definition for $x_0$  
(b) Implementation of a $x_0$ determined by the genetic algorithm

**Figure 6.1:** The implementation of $x_0$ prior and after the debugging.  
X-axis: alongshore distance [m]; y-axis: cross-shore distance [m]; colour bar: wave height [m]

After these errors were known, $x_0$ and $x_1$ could be implemented in the genetic algorithm by following the method of Chapter 5 and implement them in the intersection criteria. The result is shown in Figure 6.2b. This chapter provides information on which classes define the $x$-coordinates; mating, mutation and generator. Inside these classes, $x_0$ and $x_1$ define the terminal, turning basin and quay wall starting points which were initially set to zero (as visible above). The steps necessary for the implementation of an alongshore positioning of the breakwater are denoted in Appendix C, where every step is indicated.

![Image](a) A breakwater layout plot after implementation of a variable alongshore position. Three main errors arose, the primary breakwater was placed inside the approach channel (red circle), the secondary breakwater intersected the primary breakwater (red rectangle) and the terminal was too short (red circle)  
(b) Breakwater layout after debugging

**Figure 6.2:** The difference before and after debugging

### 6.1.1. Validation

To validate the implementation of the alongshore positioning a test case with favourable conditions away from the boundaries was created, as shown in Figure 6.3. It was expected that the model would place the breakwaters in the vicinity of $x = 1000$ m or $x = 3000$ m seen from the model’s perspective (or $x = 4000$ m and $x = 6000$ m in Figure 6.3). The CapEx and OpEx are much lower in the middle
6.1. Alongshore positioning of the breakwater

than in the sides. The wave field corresponding to the asymmetric bathymetry is shown in Figure 6.4.

Figure 6.3: The asymmetric contour bathymetry for the validation of the primary breakwater positioning. The red dashed rectangular represents the boundaries of the search area of the model.

Figure 6.4: The wave field belonging to the alongshore positioning test case. X-axis: alongshore distance [m]; y-axis: cross-shore distance [m]; colour bar: wave height [m]

Figure 6.5 presents the results for the asymmetric bathymetry and wave field. As can be observed, the primary breakwater was positioned around $x = 1000$ m, with a secondary breakwater placed at the right side of the terminal. Furthermore, the approach channel was generated in the direction of the deeper parts of the bathymetry, resulting in lower CapEx and OpEx. Between the runs a cost consistency of 9.94% was obtained.

Figure 6.5 presents the results for the asymmetric bathymetry and wave field. As can be observed, the primary breakwater was positioned around $x = 1000$ m, with a secondary breakwater placed at the right side of the terminal. Furthermore, the approach channel was generated in the direction of the deeper parts of the bathymetry, resulting in lower CapEx and OpEx. Between the runs a cost consistency of 9.94% was obtained.

(a) Optimal layout run 1, fitness score = 0.919  (b) Optimal layout run 2, fitness score = 0.935  (c) Optimal layout run 3, fitness score = 1.01
6. Results

(d) Optimal layout run 4, fitness score = 0.956
(e) Optimal layout run 5, fitness score = 0.913

Figure 6.5: The results of the alongshore positioning validation test case. The wave direction is shown in the top corners of the figure. In the title of the figure, two values are indicated in brackets (fitness score, amount of generations before convergence).

Table 6.2: CapEx, OpEx and downtime costs for the alongshore positioning

<table>
<thead>
<tr>
<th>Unit</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Run 4</th>
<th>Run 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapEx/OpEx</td>
<td>€195,865,071</td>
<td>€192,483,329</td>
<td>€179,217,822</td>
<td>€188,284,097</td>
<td>€197,044,365</td>
</tr>
<tr>
<td>Downtime costs</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fitness score</td>
<td>0.919</td>
<td>0.935</td>
<td>1.01</td>
<td>0.956</td>
<td>0.913</td>
</tr>
</tbody>
</table>

6.2. Verification

Verification is essential to track bugs and solve them. When setting up the verification test, it appeared that there were already errors in preparing the bathymetry. Therefore, first in-depth knowledge of how the model can be run was obtained. Preparation work was needed to start the model. By the creation of the bathymetry files only eight files were created where ten were necessary to run REFRAC. Therefore, the code was adjusted to create ten bathymetry files. Appendix D provides an elaboration on the input for REFRAC. After the wave data was present, the model could be run.

Another point that was detected regarding REFRAC was for waves travelling perpendicular towards an asymmetric bathymetry as shown in Figure 6.12b. REFRAC produces Not a Number-values (NaN-values) for the regions where refraction influences the wave propagation. Figure 6.6b shows a wave field generated by REFRAC for the bathymetry shown in Figure 6.6a, containing a white field (representing NaN-values) where the water and waves split, because REFRAC does not incorporate directional spreading. The Teeling (2020) model was not able to use NaN-values as wave height, causing the model to terminate when waves on an asymmetric bathymetry (Figure 6.6a) travel perpendicular to the coast. By adjusting the code, it is now possible to insert perpendicular wave fields to an asymmetric bathymetry.

The bugs encountered were:

- The input file, which should create wave data and bathymetry data, generated fewer files than necessary to run the model.

- The model is built to find the best position of the breakwater on the left or right side of the x-axis. After running the first test case, where no preference was given to the placement of the breakwater, the model terminated when it tried to position the breakwater to the left side.

- Several mathematical errors were present. These errors meant that the model sometimes gave negative x-values when this should be positive or vice versa. In addition, the wave direction was inserted as a negative value into the model, resulting in a mirrored wave direction.
6.3. Validation

The validation focused on the model’s capability to translate the input, which consists of the site data and design criteria, into the output. An essential factor for validation is the cost consistency between the runs. The results are considered consistent if the difference in fitness score between the highest and lowest is within 10%, according to Section 5.3.

First, a base case was developed, with a linear slope and waves coming from two directions with no wave height difference. The base case is meant to internally validate the model. Furthermore, the base case is used to show the results where no extremes occur. This can be used to check whether any problems arose with more extreme bathymetry or varying wave fields during external validation. An expectation is provided for the external validation test cases related to bathymetry and wave fields, about the breakwater layout.

6.3.1. Internal validation

A base case is useful to internally validate the model. The base case tested a linear slope and a uniform wave field. An internal check ensures that each element defined in the code is validated to see if the dimensions of the elements (turning basin, berthing pocket, quay wall, approach channel and breakwater(s)) met the input criteria. The genetic algorithm should generate the shape and dimensions of the elements by following the guidelines shown in Table 5.2. These guidelines are: Briganti et al. (2004), Ligteringen (2012) and PIANC (2014). Mainly the dimensions of the design vessel determine the dimensions of the water areas and quay wall.
6. Results

Figure 6.7: Linear bathymetry. The red dashed rectangular represents the boundaries of the search area of the model.

Base case run

For the base case, a homogeneous slope shown in Figure 6.7 was applied to internal validate the model. Based on the principle of a genetic algorithm where the model tries to find an optimal location, the breakwater can be placed on the left or right side. A uniform wave field and homogeneous slope were applied for the base case, with waves coming from $-30^\circ$ and $+30^\circ$ to the shore normal. All wave directions are displayed relative to the normal. With this, more advantageous locations regarding the wave environment were removed.

(a) Optimal layout run 1, fitness score = 1.09
(b) Optimal layout run 2, fitness score = 1.024
(c) Optimal layout run 3, fitness score = 1.001
(d) Optimal layout run 4, fitness score = 1.024
(e) Optimal layout run 5, fitness score = 1.037

Figure 6.8: The results of the base case, with linear bottom profile. The wave direction is shown in the top corners of the figure. In the title of the figure, two values are indicated in brackets (fitness score, amount of generations before convergence)
Figure 6.8 shows the results of the base case. The primary breakwater was placed on the right side, with a secondary breakwater covering the waves from the left. Furthermore, comparing the highest fitness score, Figure 6.8a, and the lowest fitness score, Figure 6.8c, shows a maximum fitness score difference of 8.89%.

From the base case, an internal check was executed based on the output of the python script. The parameters are shown in Table 6.1 together with the wind and current velocity and direction, the equations applied for the generation are checked. The equations are provided in appendix E. Table 6.4 shows the internal check for all elements as given by the model. For the primary and secondary breakwater, the amount of nodes depends on one criterion. Nodes 3 and 6 (the breakwater tips in Figure 6.8) should not be plotted if they are close to nodes 2 and 5. Furthermore, the approach channel angle depends on nautical safety criteria for limiting wave heights inside the approach channel and the minimum and maximum angle between the waves and the approach channel. As the genetic algorithm constantly changed the angle values, multiple runs were studied to find the optimal solution. While running the base case, the inner and outer approach channel width constantly changed, which was expected, as it depends on the alignment of the approach channel concerning the current, wind and wave direction were taken into account. Concluding, the model generated all elements as expected.

<table>
<thead>
<tr>
<th>Element</th>
<th>Criteria</th>
<th>Output value</th>
<th>Check</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approach channel angle</td>
<td>Must lie between the minimum and maximum allowable angle</td>
<td>Plots between the limits</td>
<td>Validated</td>
</tr>
<tr>
<td>Approach channel dimensions</td>
<td>Depends on the current velocity, wind direction and speed</td>
<td>Right equation according to the current velocity and wind speed</td>
<td>Validated</td>
</tr>
<tr>
<td>Breakwater</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Primary</td>
<td>Must be present if limiting wave height is exceeded</td>
<td>Four nodes</td>
<td>Validated</td>
</tr>
<tr>
<td>Secondary</td>
<td>If sec = 1 a secondary breakwater is necessary, sec = 0 no secondary necessary</td>
<td>Plot nodes for sec = 1, no nodes for sec = 0</td>
<td>Validated</td>
</tr>
<tr>
<td>Maneuvering basin</td>
<td>No intersection between berthing pocket and other elements</td>
<td>No intersection between berthing basin and other elements</td>
<td>Validated</td>
</tr>
<tr>
<td>Area</td>
<td>200 m</td>
<td>200 m</td>
<td>Validated</td>
</tr>
<tr>
<td>Berths</td>
<td>3</td>
<td>3</td>
<td>Validated</td>
</tr>
<tr>
<td>Quay wall</td>
<td>1076 m</td>
<td>1076 m</td>
<td>Validated</td>
</tr>
<tr>
<td>Turning basins</td>
<td>750 m</td>
<td>750 m</td>
<td>Validated</td>
</tr>
</tbody>
</table>

6.3.2. External validation
Waves
Two test cases were developed for the validation of the model for wave data. The wave data provides a direction and wave height to the model. By varying the wave data, the model should generate layouts in reach with the expected layouts beforehand. The expectations are based on the position and how the breakwater should block the waves.

Wave direction variation Two single wave direction test cases were developed to validate the model’s capability of recognising differences in the wave field. Figure 6.9 shows two sketches of the expected variation of wave direction. The sketches are for illustration purposes only to indicate the expected approach channel and breakwater layout. The sketches illustrate the approach channel, turning basin, primary breakwater, secondary breakwater and coastline. These elements are directly influenced by the wave direction.
Results

(a) The breakwater layout expectation where the waves are coming from -30 degrees. The approach channel is aligned to the right and is sheltered by the primary and secondary breakwater.

(b) The breakwater layout expectation where the waves are coming from +30 degrees. The approach channel is aligned to the left and is sheltered by the primary and secondary breakwater.

Figure 6.9: Expected breakwater layout for wave directions coming from $-30^\circ$ and $+30^\circ$ to the shore normal, all other environmental variables are the same for both cases. The wave direction is represented in the top corners.

Figure 6.9a shows the breakwater sketch for waves coming from $-30^\circ$. If waves are coming from this direction, it is expected that the primary breakwater will cover the approach channel from that side. The primary breakwater ensures that the downtime will be low inside the approach channel and at the berth. The secondary breakwater seems superabundant, but due to diffraction, it may be necessary to avoid downtime. As the fitness score is based on the total costs, the model should assess the costs for the secondary breakwater against the downtime costs without this breakwater. Figure 6.9b shows the breakwater sketch for waves coming from the $+30^\circ$. The same applies here as for waves coming from $-30^\circ$. The primary breakwater should cover the approach channel from waves coming from $+30^\circ$, and the secondary breakwater is not mandatory but again depends on the total costs.

Test case: $\theta = -30^\circ$ to shore normal

The first test case, where waves come from $-30^\circ$, generated the primary breakwater at the left side in all configurations (figure 6.10), without a secondary breakwater. Table 6.5 provides the CapEx, OpEx and downtime costs. It is remarkable that all runs showed zero downtime. Figure 6.10b shows the generation with the highest fitness score, while the fitness score corresponding to Figure 6.10c has the lowest fitness score. Comparing both generations gives a cost variation of 6.10%.

(a) Optimal layout run 1, fitness score = 1.172

(b) Optimal layout run 2, fitness score = 1.193

(c) Optimal layout run 3, fitness score = 1.121
6.3. Validation

(d) Optimal layout run 4, fitness score = 1.122  
(e) Optimal layout run 5, fitness score = 1.194

Figure 6.10: The results of the test case with waves coming from -30°. The wave direction is shown in the top corners of the figure. In the title of the figure, two values are indicated in brackets (fitness score, amount of generations before convergence).

Table 6.5: CapEx, OpEx and downtime costs for a wave direction of -30° to shore normal

<table>
<thead>
<tr>
<th>Unit</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Run 4</th>
<th>Run 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapEx/OpEx</td>
<td>€</td>
<td>€</td>
<td>€</td>
<td>€</td>
<td>€</td>
</tr>
<tr>
<td>Downtime costs</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fitness score</td>
<td>1.172</td>
<td>1.193</td>
<td>1.121</td>
<td>1.122</td>
<td>1.194</td>
</tr>
</tbody>
</table>

Test case: θ = +30° to shore normal

The second test case tested the model to waves coming from +30°. Figure 6.11 presents the results for the waves coming from θ = +30°. Similar to the results for θ = -30°, the primary breakwater blocked the waves, and no secondary breakwater was generated. Table 6.6 provides the CapEx, OpEx and downtime costs. All runs again showed zero downtime. Figure 6.11a shows the generation with the highest fitness score, while the fitness score corresponding to Figure 6.11c has the lowest fitness score. Comparing both generations gives a maximum fitness score variation of 5.82%.

Figure 6.11: The results of the test case with waves coming from θ = +30° to shore normal. The wave direction is shown in the top corners of the figure. In the title of the figure, two values are indicated in brackets (fitness score, amount of generations before convergence).

(a) Optimal layout run 1, fitness score = 1.089  
(b) Optimal layout run 2, fitness score = 1.053  
(c) Optimal layout run 3, fitness score = 1.058  
(d) Optimal layout run 4, fitness score = 1.119  
(e) Optimal layout run 5, fitness score = 1.104
Table 6.6: CapEx, OpEx and downtime costs for a wave direction of $+30^\circ$ to shore normal

<table>
<thead>
<tr>
<th></th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Run 4</th>
<th>Run 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapEx/OpEx</td>
<td>€ 165,334,803</td>
<td>€ 170,875,261</td>
<td>€ 170,164,492</td>
<td>€ 160,915,430</td>
<td>€ 163,013,946</td>
</tr>
<tr>
<td>Downtime costs</td>
<td>€ 0</td>
<td>€ 0</td>
<td>€ 0</td>
<td>€ 0</td>
<td>€ 0</td>
</tr>
<tr>
<td>Fitness score</td>
<td>1.089</td>
<td>1.053</td>
<td>1.058</td>
<td>1.119</td>
<td>1.104</td>
</tr>
</tbody>
</table>

Bathymetry
For the bathymetry, several test cases were designed to check if the model recognises differences in bathymetry. The model should place the breakwater along the x-axis and determine the optimal location. The model’s aspects in its assessment for the position are the breakwater segment, downtime, filling and dredging costs.

Asymmetric profile
Asymmetric bottom profiles are shown in Figure 6.12. Breakwater costs are less for a more gentle slope as the total breakwater volume is lower. It was expected that the model will place the primary breakwater at the left side and a possible secondary breakwater on the right side for the specific bathymetry shown in Figure 6.12a. Based on the bathymetry presented in Figure 6.12b, the primary breakwater was expected to be placed on the right side, and a possible secondary breakwater on the left side. However, also wave height and direction influence the alignment of the breakwater. At the gentler slope side, wave height is higher than the steeper slope due to refraction and shoaling, and, therefore, it was expected that the model generates a breakwater at the steeper slope side to have lower waves at the breakwater. To investigate whether the model generates other designs, different wave fields were used in different test cases:

- Uniform wave field, where the effect of depth variations on wave propagation were neglected (test case 1)
- Waves from $\theta = -30^\circ$ and $\theta = +30^\circ$ to shore normal (test case 2, 3)

Three test cases are presented hereupon. First, a uniform wave field is applied to the asymmetric bathymetry profiles, with waves coming from $-30^\circ$ and $+30^\circ$ to shore normal. With a uniform wave field, the model should only base the position and shape on bathymetry differences. Then, two test cases focus on the interaction of waves and bathymetry, where wave height and direction vary depending on the wave direction and bottom profile, with waves coming from $-30^\circ$ and $+30^\circ$ to shore normal. Expected was that refraction and shoaling influences the wave height and pattern and,
therefore, the wave environment should also influence the breakwater layout position and shape.

**Test case 1**

The two bathymetry profiles in Figure 6.12 are mirrored, and as the wave field is uniform along the coastline as shown in Figure 6.13, the designs belonging to the bathymetry were expected to be mirrored. First, the bathymetry shown in Figure 6.12a was tested. Expected was that the breakwater is plotted at the gentler slope, between x = 0 m and x = 1800 m. To check for coincidence, the mirrored bathymetry was also tested as it was assumed that the model recognises the difference in bathymetry.

![Figure 6.13: Uniform wave field. X-axis: alongshore distance [m]; y-axis: cross-shore distance [m]; colour bar: wave height [m]](image)

![Figure 6.14: The results of the first bathymetry test case](image)
Table 6.7: CapEx, OpEx and downtime costs for test case 1

<table>
<thead>
<tr>
<th></th>
<th>Unit</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Run 4</th>
<th>Run 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapEx/OpEx</td>
<td>£</td>
<td>141,382,954</td>
<td>148,392,415</td>
<td>149,817,386</td>
<td>144,450,969</td>
<td>140,955,364</td>
</tr>
<tr>
<td>Downtime costs</td>
<td>£</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fitness score</td>
<td>-</td>
<td>1.273</td>
<td>1.213</td>
<td>1.201</td>
<td>1.246</td>
<td>1.277</td>
</tr>
</tbody>
</table>

Figure 6.14 shows the configurations for the first test case. The first five runs correspond to a gentler slope between $x = 0$ m and $x = 4000$ m, where the sixth run is a generation for the mirrored bathymetry. An apparent similarity in breakwater layout is visible for the first five runs, where a primary breakwater was built at the gentler slope, and a secondary breakwater was situated in the transition zone around $x = 1200$ m. Table 6.7 provides the CapEx, Opex and downtime costs for the five runs. The highest fitness score, shown in Figure 6.14c, and the lowest fitness score, shown in Figure 6.14e, differ 6.3% of each other. Figure 6.14f shows the generation of the model, under the same conditions, but applied to the mirrored bathymetry as shown in Figure 6.12b. The generation showed resemblance with the generations for the mirrored bathymetry. Remarkable is that downtime for asymmetric bathymetry was also zero.

**Test case 2**

Figure 6.12a shows the bottom profiles used in the third test case. The corresponding wave field is shown in Figure 6.15.

Figure 6.15: Corresponding wave field to bathymetry shown in Figure 6.12a. X-axis: alongshore distance [m]; y-axis: cross-shore distance [m]; colour bar: wave height [m]

A distinction between $x = 0$ m till $x = 4000$ m and $x = 6000$ m till $x = 10,000$ m can be observed in the bathymetry, with a transition region between $x = 4000$ m and $x = 6000$ m. The breakwater was expected to be placed on the left side due to lower CapEx and OpEx for the breakwater shown in test case 1. On the other hand, by observing the wave field belonging to the bathymetry, wave heights are higher at the gentler slope. Therefore, the model should weigh up the pros and cons of each location. The interaction between the bathymetry and wave field, resulted in the optimal breakwater layouts shown in Figure 6.16.
6.3. Validation

Figure 6.16: Breakwater layouts for a bathymetry with a gentler slope at the left side. The wave direction is shown in the top corners of the figure. In the title of the figure, two values are indicated in brackets (fitness score, amount of generations before convergence)

Table 6.8: CapEx, OpEx and downtime costs for test case 2

<table>
<thead>
<tr>
<th>Unit</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Run 4</th>
<th>Run 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapEx/OpEx</td>
<td>€159,586,465</td>
<td>€162,778,079</td>
<td>€148,226,617</td>
<td>€153,061,224</td>
<td>€156,168,662</td>
</tr>
<tr>
<td>Downtime costs</td>
<td>€0</td>
<td>€0</td>
<td>€0</td>
<td>€0</td>
<td>€0</td>
</tr>
<tr>
<td>Fitness score</td>
<td>1.128</td>
<td>1.106</td>
<td>1.214</td>
<td>1.176</td>
<td>1.15</td>
</tr>
</tbody>
</table>

The results obtained from the second test case, with the gentler slope between $x = 0$ m and $x = 4000$ m, shows that four of the five runs generated a primary breakwater on the right side with the approach channel directed to the left. This was expected as the waves were coming from the right. Table 6.8 provides the CapEx, OpEx and downtime costs for the five runs. Again all runs have zero downtime costs and only CapEx/OpEx. The highest fitness score is shown in Figure 6.16c and the lowest fitness score is shown in Figure 6.16b, the fitness score differ 9.76%.

Test case 3

The third test case applied the bathymetry profile with a gentler slope between $x = 6000$ m and $x = 10,000$ m, shown in Figure 6.12b. For this test case the wave field is shown in Figure 6.17. With this wave field and bathymetry, the same expectation holds as for test case two. Relating to CapEx and OpEx the breakwater was expected at the gentler slope regarding test case 1, but regarding the wave field, waves are less extreme at the steeper slope.
6. Results

(a) Wave field belonging to bathymetry of Figure 6.12b, waves coming from $\theta = -30$

(b) Wave field belonging to bathymetry of Figure 6.12b, waves coming from $\theta = +30$

Figure 6.17: Corresponding wave field to bathymetry shown in Figure 6.12b. X-axis: alongshore distance [m]; y-axis: cross-shore distance [m]; colour bar: wave height [m]

(a) Optimal layout run 1, fitness score = 1.159
(b) Optimal layout run 2, fitness score = 1.253
(c) Optimal layout run 3, fitness score = 1.141
(d) Optimal layout run 4, fitness score = 1.249
(e) Optimal layout run 5, fitness score = 1.196

Figure 6.18: Breakwater layouts corresponding to a bathymetry profile with a gentler slope at the right side. The wave direction is shown in the top corners of the figure. In the title of the figure, two values are indicated in brackets (fitness score, amount of generations before convergence)

Table 6.9: CapEx, OpEx and downtime costs for test case 3

<table>
<thead>
<tr>
<th>Unit</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Run 4</th>
<th>Run 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapEx/OpEx</td>
<td>€155,215,015</td>
<td>€143,669,382</td>
<td>€157,756,354</td>
<td>€144,086,452</td>
<td>€150,457,642</td>
</tr>
<tr>
<td>Downtime costs</td>
<td>€0</td>
<td>€0</td>
<td>€0</td>
<td>€0</td>
<td>€0</td>
</tr>
<tr>
<td>Fitness score</td>
<td>-</td>
<td>1.159</td>
<td>1.253</td>
<td>1.141</td>
<td>1.249</td>
</tr>
</tbody>
</table>

Figure 6.18 shows the results of the third test case. The results show the same pattern as for the third test case. The primary breakwater was placed at the steeper slope side, where the wave field is calmer, and the waves are lower at the gentler slope. Furthermore, a cost consistency between the
highest, Figure 6.18b and lowest, Figure 6.18c, fitness score of 7.78% obtained. It can be observed from Table 6.9 that again all layouts have zero downtime costs.

**Sudden deepening** A sudden deepening profile was constructed to test the model’s ability to recognise the depth drop in the bathymetry cross-shore. Figure 6.19 shows the sudden deepening profile, with a huge drop between $y = 1000$ m and $y = 1800$ m. The model should recognise this drop and place the breakwaters between $y = 0$ m and $y = 1800$ m to reduce breakwater segment costs. Figure 6.20 shows the generations belonging to the sudden deepening profile, with their costs displayed in Table 6.10.

![Figure 6.19: Contour bathymetry of sudden deepening. The red dashed rectangular represents the boundaries of the search area of the model](image)

(a) Optimal layout run 1, fitness score = 1.14
(b) Optimal layout run 2, fitness score = 1.056
(c) Optimal layout run 3, fitness score = 1.06
(d) Optimal layout run 4, fitness score = 1.088
(e) Optimal layout run 5, fitness score = 1.106

Figure 6.20: Results for the sudden deepening test case, with a depth drop behind $y = 1800$ m. The wave direction is shown in the top corners of the figure. In the title of the figure, two values are indicated in brackets (fitness score, amount of generations before convergence).
Table 6.10: CapEx, OpEx and downtime costs for a sudden deepening profile

<table>
<thead>
<tr>
<th>Unit</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Run 4</th>
<th>Run 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapEx/OpEx</td>
<td>€157,856,737</td>
<td>€171,001,021</td>
<td>€169,823,321</td>
<td>€165,486,807</td>
<td>€162,685,625</td>
</tr>
<tr>
<td>Downtime costs</td>
<td>€0</td>
<td>€0</td>
<td>€0</td>
<td>€0</td>
<td>€0</td>
</tr>
<tr>
<td>Fitness score</td>
<td>-</td>
<td>1.14</td>
<td>1.056</td>
<td>1.06</td>
<td>1.088</td>
</tr>
</tbody>
</table>

The generated layouts showed that the breakwater was constructed closer to the shore than in the former test cases and reach no further than approximately y = 1800 m, which was expected. The downtime costs for all generations were again zero. Regarding cost consistency, the highest fitness score that can be observed in Figure 6.20a, and the lowest fitness score in Figure 6.20b, revealed a cost consistency between the runs of 7.95%.

**Downtime analysis**

The results of all test cases showed zero downtime costs for the breakwater layouts. The model was expected to optimise the CapEx, OpEx and downtime costs and find a balance between the three. Therefore, an extra test case was developed to examine the cause why the model always optimises towards zero downtime.

The validation test cases used the same bathymetry as the base case, so a linear bathymetry with a slope of 0.01. To exclude the possibility of having layouts where the harbour basin and approach channel shelter for some wave directions, more than two wave directions were used. When multiple wave directions are used, there will most likely be a design where waves cannot be excluded completely, causing downtime to occur.

![Figure 6.21: First downtime analysis test case](image)

(a) Scatter plot for CapEx, OpEx and downtime costs. Black arrows indicate the downtime costs. (b) Best layout for the downtime analysis. In the title of the figure, two values are indicated in brackets (fitness score, amount of generations before convergence).

Figure 6.21a shows a scatter plot for the CapEx, OpEx, downtime costs and total costs. The figure shows, in pink, that if only 0.05% downtime occurs per year, the downtime costs are already 65 million euros (arrow). The model assumed three container terminal berths within the breakwater. This would mean that for every hour that a berth can not be exploited, the port revenue loss equaled 4.95 million euros. Reviewing downtime tariffs estimated by Arcadis for the same design vessel, the hourly downtime costs are around 3,000 to 5,000 euro. Pachakis and Kiremidjian [2004] found downtime costs in the order of 60,000 dollar per day or 2,500 dollar per hour. Considering annual inflation of roughly 2%, this also adds up to approximately 4,000 dollars or 3,400 euros. So, if, for example, 1% downtime would occur, the total costs would rise by 1.3 billion euros on account of downtime only. This means the downtime costs in the breakwater model were too high, which may
force the model to reduce the downtime costs to zero.

The second downtime test case focused on the examination of the downtime equation. In the downtime cost equation, one parameter is the total downtime. The downtime should be presented as a percentage per year. However, the model produced decimal numbers. So instead of 1.8% downtime, the model renders a downtime of 180%. The downtime costs calculation was adapted so it will provide the correct percentages. Figure 6.22 shows a scatter plot with more accurate estimations for the downtime. The economic optimisation graph clearly shows that a lower downtime does not necessarily resulted in lower total costs. The lowest total costs (dark red dot in Figure 6.22a) now also shows total costs where downtime is not equal to zero.

6.4. Performance

6.4.1. Usability

User-friendliness is an important aspect of a complex model to create attractiveness for a model among engineers. As mentioned in Section 6.2 was the use of the model very complicated. Several Python files were used to prepare the input for the model and run the model. To increase the user-friendliness of the model, these separate files were merged into one Python file. In addition, it was also been added that it is possible to link a case name to certain input data. This ensures that the results are stored within a certain folder on the computer, which ensures clarity. The following input can now be adjusted in this document:

- Case name
- Design vessel dimensions
- Bathymetry
- Wave height
- Wave direction

Furthermore, understanding the installation, execution and principles ease the use of the model. Teeling (2020) added a read me in the file directory for the installation and the execution of the model. The information in this read me was very concise, which made it difficult to install the right
files and execute the model. Therefore, an extended user manual was created to ease the use of the model, the link to it can be found in Appendix B.

A brief summary is added to introduce the reader to the main steps before execution is possible. Figure 6.23 provides the steps required for the execution of the model. First, the model and Python (at least version 3.6) should be installed on the computer. Next, a virtual environment should be developed, where all python packages are installed. After this, the complete model (in which bathymetry and wave data are created) can be run from one Python file.

![Image of the steps necessary to install and run the model](image)

**Figure 6.23: The steps necessary to install and run the model**

### 6.4.2. Efficiency

The implementation of Dask reduced the computation by creating the opportunity for the model to process parallel on the computer cores. However, with the implementation of the alongshore positioning, computation time was expected to increase by the addition of a new variable. Therefore, the effect of the alongshore positioning and Dask was investigated by running the base case three times (to reduce the chance of outliers). First, the base case was inserted in the Teeling (2020) model to check the computation time. Next, the model with the alongshore positioning, but without the dask server was assessed. Last, the updated model was run on two and four cores (for Dask) to check the total running time. Table 6.11 shows the difference in computation time.

<table>
<thead>
<tr>
<th>Model version</th>
<th>Computation time [min]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teeling (2020)</td>
<td>160</td>
</tr>
<tr>
<td>Teeling (2020) with refactoring and alongshore position</td>
<td>118</td>
</tr>
<tr>
<td>Current model on 2 cores</td>
<td>104</td>
</tr>
<tr>
<td>Current model on 4 cores</td>
<td>98</td>
</tr>
</tbody>
</table>

The current version of the model, running on four cores, provided the lowest computation time. However, even a drop in computation time was visible after the adjustments and modifications to the Teeling (2020) model (debugging and refactoring).

Next, the genetic parameters were assessed. Each test was run three times per value for the population size, mutation rate, and mutation size (to reduce the chance on outliers).

**Population size**

The population size has an enormous influence on the models’ performance. For higher population size, the model has a larger population to generate layouts from. Therefore, it was expected that fitness score would rise for a higher population size. However, all these populations affect the computation time. Therefore, a trade-off between computation time and fitness score is essential. By the tuning of the population size, five different population sizes were tested. Figure 6.24 shows the spread of the population size with the corresponding computation time and fitness score. The values behind the graph can be found in Appendix E for the population size and mutation rate and size.
6.4. Performance

Figure 6.24: The results for the different population sizes, with a mutation rate of 0.25 and mutation size of 0.2. After a population size of 48 the increase in fitness score stagnates.

Table 6.12: The average computation time and fitness score for the various population sizes

<table>
<thead>
<tr>
<th>Population size [-]</th>
<th>Computation time [min:sec]</th>
<th>Fitness score [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>111:11</td>
<td>1.148</td>
</tr>
<tr>
<td>60</td>
<td>53:37</td>
<td>1.12</td>
</tr>
<tr>
<td>48</td>
<td>37:02</td>
<td>1.144</td>
</tr>
<tr>
<td>24</td>
<td>22:37</td>
<td>0.959</td>
</tr>
<tr>
<td>12</td>
<td>10:02</td>
<td>0.72</td>
</tr>
</tbody>
</table>

From Table 6.12 it is visible that a population size of 120, 60 and 48 outperforms the other population sizes regarding the fitness score. However, with a population size of 48, computation time drops enormously without losing accuracy.

Mutation rate

The mutation rate determines the variation of the chromosomes between the generations. A higher mutation rate will support the model in escaping local optimum. However, this will also lead to less accurate solutions. It can be observed that a mutation rate of 0.25 has the highest average fitness score, with a big difference to the second highest average of a mutation rate of 0.15. Furthermore, the computation time not enormously increased between a mutation rate of 0.15 and 0.25. The best trade-off can be observed, therefore, for a mutation rate of 0.25.
Figure 6.25: The fitness score plotted against the computation time, for various mutation rate, with a population size of 48 and mutation size of 0.2

Table 6.13: The mutation rate with corresponding computation time and fitness score

<table>
<thead>
<tr>
<th>Mutation rate [-]</th>
<th>Computation time [min:sec]</th>
<th>Fitness score [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>51:40</td>
<td>0.523</td>
</tr>
<tr>
<td>0.35</td>
<td>30:14</td>
<td>0.619</td>
</tr>
<tr>
<td>0.3</td>
<td>25:25</td>
<td>0.517</td>
</tr>
<tr>
<td>0.25</td>
<td>29:00</td>
<td>1.118</td>
</tr>
<tr>
<td>0.2</td>
<td>24:36</td>
<td>0.911</td>
</tr>
<tr>
<td>0.15</td>
<td>26:13</td>
<td>0.958</td>
</tr>
<tr>
<td>0.1</td>
<td>19:07</td>
<td>0.837</td>
</tr>
</tbody>
</table>

**Mutation size**

The mutation size was expected to influence the standard deviation for the mutation rate and, therefore, affect the fitness score and computation time. Observing Table 6.14 shows that the mutation size hardly influences the fitness score and computation time. The best trade-off between computation time and fitness score was obtained for a mutation size of 0.2. The scatter plot visible in Figure 6.26 shows all results for the mutation size.

Table 6.14: The varying mutation size with computation time and fitness score

<table>
<thead>
<tr>
<th>Mutation size [-]</th>
<th>Computation time [min:sec]</th>
<th>Fitness score [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>20:07</td>
<td>1.063</td>
</tr>
<tr>
<td>0.2</td>
<td>21:25</td>
<td>1.139</td>
</tr>
<tr>
<td>0.18</td>
<td>31:15</td>
<td>1.107</td>
</tr>
<tr>
<td>0.15</td>
<td>35:07</td>
<td>1.077</td>
</tr>
<tr>
<td>0.1</td>
<td>37:38</td>
<td>1.112</td>
</tr>
</tbody>
</table>
6.5. Summary

By using test-driven development, several bugs were tracked and solved. The potential errors were numbered to quantify the effectiveness of test-driven development for debugging. By solving these errors, the convergence rate increased by 50%, from 66.67% towards 100% for the executed test cases. Furthermore, the internal check showed that all components are plotted following the guidelines. Furthermore, the model generated breakwater layouts considering bathymetry, wave height, and wave direction and is, therefore, considered externally validated. Lastly, the computation time was sped up using multiple cores by debugging and tuning the optimisation parameters. Also, revision of the genetic algorithm parameters and the addition of Dask lead to a decrease of 81.25% computation time, from 160 minutes to 30 minutes on average.

Figure 6.26: Mutation size assessment, where the fitness score is plotted against the computation time. The population size equals 48 and the mutation rate 0.25
In this chapter, a comparison of the Teeling (2020) model and the updated model is provided. First, the modifications and adjustments to the Teeling (2020) model are discussed. Then, the performance of the updated model is compared to that of the Teeling (2020) model. Lastly, a conclusion on the differences between the updated model and the Teeling (2020) model is provided.

7.1. Adjustments

7.1.1. Verification

Before eventual additions or modifications to the Teeling (2020) model were made, the model was verified to debug the model. Applying test-driven development, the model was vetted to find bugs. First, in the code of the Teeling (2020) model, tasks were sometimes performed several times, resulting in higher computation time. Next, the Teeling (2020) model contained some typing errors, for example the model could not find the optimal location for the primary breakwater, left or right. By rewriting these parts, the model can switch the primary breakwater from left to right of the x-axis. Last, the Teeling (2020) model sometimes generated elements that intersected with other elements. For example, the primary breakwater sometimes intersected with the approach channel or turning basin. The model should check that there is no intersection between the different elements. However, by the generation of these elements, the intersection criteria were not specific enough and, therefore, intersection sometimes appeared.

Debugging of the Teeling (2020) model was essential as already several features were implemented and it was unsure if bugs were present. Addition of more features would probably making it harder to find bugs in a later stadium. In summary, the following bugs have been fixed:

- Intersection between elements of the breakwater model.
- Typing errors that cause miscalculations and caused an error when the primary breakwater was placed on the right.
  - A huge typing error was found in the downtime estimation; decimal downtime estimation instead of a percentage downtime.
- Possibility to handle Not a Number values, essential for waves perpendicular to an asymmetric coastline.
- Execution errors that caused the model to crash or go into an endless loop.
7.1.2. Positioning of the breakwater

In the Teeling (2020) model the position of the primary breakwater starting points was fixed, \( x_0 = 0 \) and \( x_1 = 0 \). Therefore, the Teeling (2020) model was not meant for site selection. With the adjustments to the alongshore position of the primary breakwater, this assumption was partly cancelled out. By the implementation of a primary breakwater that searches for an optimal location alongshore results in the modifications of the following aspects:

- Terminal
- Intersection

**Terminal**

The starting point of the terminal was hardcoded in the Teeling (2020) model. This induced problems when the primary breakwater became variable. Therefore, modifying the starting and ending point to the varying \( x \)-coordinate of the primary breakwater created a terminal that moves with the primary breakwater.

**Intersection**

Intersection criteria inside the code changed with implementing a primary breakwater that has no fixed starting point. The following aspects take \( x_0 = 0 \) and \( x_1 = 0 \) as starting points: turning basin, approach channel, manoeuvring basin, berthing pocket and terminal. The \( x \)-coordinates of the primary breakwater and the end of the terminal were used for the model to check if there was no intersection between elements. The \( x \)-coordinate for the end of the terminal was based on \( x_0 = 0 \). However, by applying a varying \( x \)-coordinate, this method was not applicable anymore. The model now takes as input values the varying \( x_0 \) and \( x_1 \) and the terminal location following the primary breakwater.

**Positioning**

By implementing a varying primary breakwater alongshore, the model can search for the optimal location alongshore. The test cases validated the ability of the model for alongshore positioning.

7.1.3. Downtime analysis

The external validation test cases showed that the model aimed to decrease the downtime towards zero. A deeper analysis in the downtime was performed that first focused on a varying wave field. This resulted in the revealing of too high downtime costs per hour in the Teeling (2020) model, 4.65 million euros. The adjustment of the downtime equation resolved the downtime error.

7.2. Performance

The generations were executed on a Desktop, Intel core i9 10th gen, 32GB RAM. Therefore, the results obtained are not comparable to other computers.

**Convergence**

The updated model uses the same optimisation algorithm principles, a genetic algorithm, as the Teeling (2020) model. For a genetic algorithm to approach an optimal breakwater layout, the model must converge towards an optimum. If the model converges too early, probably the model has not reached an optimum. The Teeling (2020) model converged after 100 to 150 generations with a convergence rate of 0.001. The updated model showed convergence after 90 to 189 generations, where averages were used to remove outliers. For the extreme cases with asymmetric bathymetry, convergence sometimes happened in more than 150 generations. However, Teeling (2020) did not
use that extreme cases making it hard to compare the two models for consistency on such extreme cases.

**Running time**
Comparing the running time of the Teeling [2020] model and the updated model indicates the influence of the adjustments and additions. Several bugs were fixed during test-driven development, and the code was rewritten to lower the running time. In addition, Dask was added to the model, which enables selecting the number of cores whereon the model runs. When using four workers, the running time decreases from 38.75% to 98 minutes. After tuning the optimisation parameters, the computation time decreases even further to about 30 minutes on average. Resulting in a total decrease of 81.25% from 160 to 30 minutes on average.

**Consistency**
Consistency is an important factor to assess the value of the breakwater model. The results showed that all runs were within a 10% cost consistency, which was set as a criterion in Chapter 5 for consistency.

**7.3. Conclusion**
With the adjustments made to the model, the running time decreased enormously. By applying the test-driven development, several errors were cancelled out. These errors caused some parts of the code to be duplicated and, even worse, sometimes caused the model to have an infinite loop, which was kept unnoticed by the former developer. Before adjustments and debugging, the total running time was at least 160 minutes. By adding Dask and tuning the optimisation parameters, the running time decreased to 30 minutes while accuracy did not drop.

To ensure that the model does not give incorrect values or terminates, the model was debugged. The value of the model increases when it can consistently produce results. In addition, debugging gave insight into the model, which was needed to implement the alongshore position. The addition of the positioning of the primary breakwater, the model can search alongshore for the most optimal location. This resulted in a larger solution space as the model can now search alongshore for the most optimal location.
This report aimed at improving the value of the breakwater layout model. This chapter first discusses the applied method in Section 8.1. Then, the purpose of the model is discussed in Section 8.2. Finally, in Section 8.3 the limitations of the research are presented.

8.1. Software development method
Test-driven development (TDD) was used for the improve in applicability with the implementation of the alongshore positioning. Repeatedly running and adjusting the developed test case for the positioning exposed several errors regarding the direction of the x-value and supported solving them. As a result, TDD is considered a suitable method to implement new features in the model.

Furthermore, this research also pointed out that it can be used for verification (debugging), internal and external validation and improvement of performance:

- To quantify the use of TDD, the errors found were counted. In total, fifteen typing errors and six execution errors were found during debugging in Chapter 6. TDD supported the developer by tracing and testing the code and writing test cases that the model should solve.

- TDD resulted in the internal and external validation of the model by using several test cases. Furthermore, during the external validation process TDD proved its added value by solving the error in the downtime equation. This error was caused by the fact that the downtime was calculated as a decimal number instead of a percentage, resulting in hundred times higher downtime costs.

- Tuning the optimisation parameters was aided by TDD by testing and modifying the code until the best trade-off between the fitness score and computation time was found. Furthermore, TDD supported with the implementation and validation of Dask, by comparing the computation time with and without Dask.

However, the disadvantage of using TDD as a software development method is that a flaw in the code exposed in later test cases may turn the previous test runs invalid. The downtime error was discovered after the validation test cases. However, the downtime does not influence the working of the model but only affect the optimisation equation. Therefore, less optimal solutions may be found but not affecting the verification and validation of the model. Furthermore, the test cases developed for TDD were constructed by one person. It is, therefore, possible that certain aspects were not incorporated because the researcher has not thought of this.
8.2. The purpose of the breakwater layout model
This research focused on the validation and in-depth understanding of the breakwater model to increase its value. The breakwater model is meant to be an optimisation tool during the conceptual design phase, in which it can support an engineer by generating many alternatives within a short period. The model can carefully examine a wide solution space to find possible alternatives by varying the breakwater nodes during the runs and selecting the cheapest design. With this, the model can sharpen a business case by comparing several designs and indicating the corresponding costs. Furthermore, the genetic algorithm is a black box, potentially resulting in solutions an engineer never thought of. This may help the engineer to think of alternatives to its designs.

However, the accuracy of the model can be discussed. Currently, the purpose of the tool is to optimise the breakwater layout design. To speak of an optimisation model that is useful for engineers, the results and input must have a certain accuracy. First, the verification and validation test cases showed that all results were within a 10% cost margin. However, between the results, differences were visible in orientation of the primary and secondary breakwater. Therefore, the breakwater layout model can generate possible alternatives to support engineers. But, it will also be necessary to check whether the results can show more resemblance to each other to create a model that works towards an optimum layout. Second, the site data inserted in the model must also have a certain accuracy. This accuracy should match the intended conceptual design phase. The use of REFRAC, however, posed some problems for the accuracy of the estimated wave height and direction:

- As REFRAC is run prior to the wave model, the approach channel is not incorporated for the wave propagation and, therefore, shoaling and refraction are neglected in the approach channel. These wave processes still have a significant influence on wave propagation in the harbour basin.

- Only diffraction is implemented in the model to simulate wave propagation influenced by the breakwaters. However, reflection significantly influences the diffraction coefficient (Boshek, 2009). By neglecting reflection, less accurate wave heights are obtained.

- REFRAC is unsuitable for creating a wave climate where wave direction, height and occurrence can be inserted.

Because the wave data is not accurate enough, wave heights inside the harbour basin and around the breakwater tips are not estimated correctly. This can influence the downtime estimations and whether the layouts fulfill the design criteria.

Nevertheless, the choice can be made to use the breakwater model for a quick scan design, where ‘black box’ alternatives are elaborated by the model. Where after, the engineer can choose one alternative from the model and further examine how feasible the layout is. Still, improvement on the wave climate is needed.

8.3. Limitations of the research
The research has the following limitations:

- The use of REFRAC posed the following limitations:
  - REFRAC neglects reflection, diffraction and wave breaking. Teeling (2020) used rules of thumb to estimate the effect of diffraction on the wave height inside the harbour basin. Still, the effect of reflection is not included in the wave height calculation, while this is essential for good wave studies inside the harbour basin (Maroudi & Reijmerink, 2020).
Furthermore, without wave breaking, waves will pile up towards the coast, giving unreliable wave heights. However, wave breaking does not influence the wave height inside the harbour basin and, therefore, is only essential for the design of the breakwater cross-section. As, most breakwaters are constructed outside the breaking zone wave breaking is not of great importance.

- REFRAC has a boundary problem where waves pile up at the side they are propagating to. Therefore, two boundaries at $x = 3000$ m and $x = 7000$ m were set to exclude regions where waves pile up. This bounds the solution space of the model.

- The wave height estimated inside the harbour basin lacked on accuracy. REFRAC was run prior to the breakwater layout model and, therefore, run without the breakwater and approach channel present in the bathymetry. Thus, the effect of refraction and shoaling in the approach channel is not incorporated, while this can have a huge influence on the design of the approach channel to reduce wave propagation inside the harbour basin (Li et al., 2000).

- It was not possible to simulate a wave rose, a wave climate that should incorporate the wave height, direction and occurrence per year, by REFRAC. As a result the model assumed that there is no difference in occurrence for the inserted wave heights and directions. Therefore, it is not possible for the model to search for layouts whereby an allowable downtime occurs, which might result in the chance that less optimal solutions are chosen.

- The aspects considered to value the model were objective aspects (applicability, performance, verification and validation). However, assessing the results with an expert can help gain even more value for the model and provide an answer whether the model will be useful for and applied by engineers.

- The downtime was estimated solely based on the wave height thereby the wave period was neglected. Since the model is intended for the conceptual design phase, it is allowable to only consider wave height for the downtime estimation.

- Verification of the model was only based on the executed test cases.

- The optimisation parameters were sequentially tuned (population size -> mutation rate -> mutation size). Other sequences have not been tested.

- When calculating the downtime, the model assumed that operational downtime always occurs when there is navigational downtime. This is not necessarily true.

- Due to the limited time available, priority was given to implement the alongshore position to increase the applicability. The alongshore position was chosen as it enormously increases the solution space for the breakwater layout model with the possibility to compare alternatives along the shoreline.

- The technical aspects of breakwater design only determine the alongshore position. However, social and environmental aspects also influence the position of the port and breakwater(s).
Conclusions and recommendations

9.1. Conclusions
This research examined the possibilities to increase the value of a breakwater layout model through verification, validation and improvement of performance and applicability. Therefore the following research question was formulated:

How can the value of the genetic algorithm based breakwater layout model be improved?

Knowledge about the conceptual design phase was necessary to assess the value of a breakwater layout model. In the conceptual design phase, engineers have to develop breakwater layouts, based on the site data and design criteria, within limited time. When done by hand, engineers can not investigate a wide solution space and, therefore, choose alternatives mostly on expertise. At the end of the conceptual design phase, the total estimated costs are calculated to determine the economically most favourable alternative. Thus, a valuable model has to contain all design criteria, have the possibility to enter the site data and compare alternatives to select which one will be most optimal to apply within an acceptable time while no errors may occur.

Test-driven development (TDD) principles were applied to assess the model’s value. Test-cases were developed and executed to debug and validate the model and increase the applicability and performance of it. A conclusion is provided for each valuable aspect, mentioned in the research objective:

Applicability TDD was used to implement the capability for the model to position the primary breakwater alongshore. The positioning has been implemented and validated in four steps:

1. Development of a test case whereby a favourable location exist at $x_0 \neq 0$, forcing the primary breakwater towards $x_0 > 0$
2. Implementation of the code to a manually fixed value larger than zero
3. Refactoring the code by implementing the $x_0$ in the algorithm to ensure it is included in the optimisation
4. Re-running the test case and validate the alongshore positioning of the breakwater layout model

Verification An important aspect of a valuable model is that no errors occur to prevent unexpected behaviour or sudden termination. Therefore, TDD was used to verify the model. A test case was set up and executed, which resulted in fifteen typing errors and six termination errors.
These errors were found by performing repeated tests, putting breakpoints in the code to ensure the code works until that specific point. These errors caused the model to fail in about one out of three runs, eliminating these errors resulted in 100% successful execution for the tested cases.

**Validation**  To add value to the model, internal and external validation was needed. If the model is not internally validated and it is thus unknown if the generated results are solely due to the input, it will have little value. Therefore, it was investigated by the development of test cases whether all functional requirements were correctly implemented and translated into the results. For this, examination of the code was necessary, which also revealed that the model is capable of the breakwater layout design for a broader range of cargo vessels. To externally validate the model, the capability of the model to construct a breakwater layout with respect to the bathymetry and wave data was tested. First, the bathymetry test cases showed that the model generated the breakwater(s) where downtime costs and capital expenditures (CapEx) and operational expenditures (OpEx) were lowest. Second, the wave field test cases validated the model’s recognition that a cheaper solution exists in blocking the waves with one breakwater when the waves come from one side. Another aspect that external validation added to the model’s value was the uncovering of the downtime miscalculation. All test cases showed zero downtime, which was uncommon as most ports prefer little downtime instead of higher breakwater costs because little downtime is relatively cheaper than the increase in CapEx and OpEx. Solving this showed that the model could optimise downtime costs and CapEx and OpEx, thereby restoring the optimisation function of the model.

**Performance**  The use of the model is attractive when it is user-friendly and efficient. The breakwater layout model is very complex, so by the addition of a user manual it is easier to install, use and understand. Furthermore, with the addition of one input python file, it is possible to run the model and change important parameters from this python file. Another aspect of the model’s performance is the efficient use of the computer resources. Therefore, by adding Dask and tuning the optimisation parameters of the model, computation time decreased on average with 81.25%, from 160 to 30 minutes. In addition, tuning of the parameters, an ascending fitness score was visible towards a population size of 48 where after the fitness score increase stagnates. Therefore, a population of 48 was selected. With this population size a mutation rate of 0.25 was chosen as it showed best trade-off between computation time and fitness score. At last, a mutation size of 0.2 was selected as this also provided the best trade-off between computation time and fitness score. Remarkable is that the mutation size hardly influenced the fitness score of a run.

In conclusion, within the research scope and by the executed test cases, the model improved significantly with the use of test-driven development by verification, internal and external validation and increase in applicability and performance. The breakwater layout model can accelerate the conceptual design phase, and alternatives can be investigated within a larger solution space. Therefore, this model will contribute to the design of breakwaters in the conceptual design phase.

### 9.2. Recommendations

During the validation of the model, it emerged that the model performs all implemented functions well. However, improvements are still recommended to create a full working conceptual design model for breakwaters. The aspects are arranged on level of importance, starting with the aspect that is strongly advised to improve.
Downtime estimation
The downtime estimation in the current model is still too simplistic, and client preferences for the amount of downtime can not be defined. Therefore, three points related to the downtime are advised to be further investigated, where the first is a huge step towards a model that is closer to reality:

- The model loops through the wave data loaded to the model. However, it is not possible to give a certain percentage of occurrence to simulate a realistic wave climate. Therefore, the model will calculate the downtime on the most adverse waves and think that they occur just as often as the more favourable conditions. The addition of a probability of occurrence to certain wave fields will solve this problem.

- Assessing the correlation between the two types of downtime can advise if navigational and operational downtime should be separately incorporated in the model.

- The model optimises the total costs, including the downtime costs. However, an amount of downtime desired by the customer can be added. Some ports prefer a higher percentage of downtime when vessels call less frequently. By increasing the allowable amount of downtime, CapEx and OpEx can decrease. Since the downtime is calculated for a port that wants to be 100% available, the costs for the downtime will not be passed on to the customer as these downtime costs are not loss of income. This will ultimately lead to lower total costs for the client.

Wave model
Currently REFRAC is used to model the waves for the breakwater layout model. However, REFRAC does not incorporate directional spreading, reflection, wave breaking, wind and current influence. Furthermore, the approach channel is constructed after REFRAC is run and, therefore, refraction and shoaling in the approach channel are neglected. These can influence the wave processes inside the harbour basin while this is important for the downtime estimation. In drawing up the problems that REFRAC entails, three main advises are given:

1. It is advised to investigate other wave modelling tools that are more accurate. The addition of wave breaking, reflection, wind and current influence and directional spreading will make the model more accurate and, therefore, more valuable and reliable to an engineer. A wave modelling tool that may be applicable is SWAN. However, SWAN has to be rerun every generation. Therefore, it should be kept in mind that the computation time will increase. For this, it must be considered whether the more accurate solutions is worth the higher computation time.

2. If it is chosen to still use REFRAC, it is advised to implement equations that approximates the wave height inside the harbour basin due to reflection, shoaling and refraction. Two possible solutions to include these processes that may be investigated:

   a) A solution in line with the optimisation tool is implementing shoaling and refraction equations inside the breakwater model to estimate wave propagation due to the approach channel. This solution will probably increase the running time of the breakwater model. However, it provides interaction between the breakwater layout and the wave processes.

   b) Another solution is to approximate shoaling and refraction influence on the wave height inside the harbour basin. Using numerical models and developing several test cases can estimate a coefficient for refraction and shoaling. These coefficients must be linked to certain situations. In the model, these situations will be implemented through criteria, after which the model calls the correct coefficients for the calculation of the wave propagation. This makes it possible to estimate the effect of these two wave processes on the wave height in the harbour.
Purpose of the model
In Section 8.2, the purpose of the model was discussed. The current state of the model is the perfect moment to choose the future purpose of the model. For an optimisation tool, the accuracy of the input data has to be more accurate. For a tool that serves as a sketch tool for engineers or a visualisation tool for clients and engineers, lower accuracy criteria are obligatory than for an optimisation tool. Still, the results should be judged by an expert if the model can already serve as a visualisation tool.

Validation
With the validation of the breakwater model, fictional test cases were developed. These test cases were meant to validate the recognition of the breakwater model for favourable bathymetry and wave conditions and externally validate the model. However, the model’s value can further increase by validating the model with experts. Applying the model to previous case studies and having the results reviewed by experts in breakwater design will show whether the model can produce valuable results according to these experts opinions.

Soil model
After implementing the primary breakwater alongshore position, the primary and secondary breakwater cross-shore position can be added. The addition of the cross-shore position demands an extra addition to provide more detailed information about the soil conditions. For an optimal cross-shore position for the breakwater, a soil model will be very effective as it provides information about the soil conditions in the designated area. Regarding section 3.3, the soil model must fit into the concept phase. Therefore, it is recommended to divide the soils into four main types: rock, clay, sand and silt, and mud. Each soil type has its soil mechanic problems as sliding, squeezing and liquefaction.

Optimisation parameters
One of the points that this research focused on was gaining insight into the core of the model. The genetic algorithm parameters were studied and optimised. A lower mutation rate makes it more difficult for the model to arrive at a local optimum, but it does allow for more detailed designs. However, a too low mutation rate at the start of the run is not advised due to the local optimum problem. This is also visible during the test runs in section 6.4. The lower mutation rates result in lower computation time and lower fitness scores because the local optimum is harder to escape. When it is possible to decrease the mutation rate during the run, this is beneficial for the optimisation. A decreasing mutation rate during the run allows the model to start with a broad solution space through which many alternatives are compared and work towards an increasingly specific search area.

Vessel type
The scope of the Teeling (2020) model was focused on container vessels. For applicability reasons, the model would benefit if it could handle different vessel types. During the examination of the model, it became clear that the equations used depend on the design vessel dimensions, tug time and limiting wave height. When comparing these equations with the design formulas of other vessels, it appeared that the formulas are applicable for several types of vessels. Therefore, the model is expected to be applicable for other vessel types. However, no research has been done on this. Validating this provides information if the model is broader applicable than is thought until now. For example, by validating a roll-on/roll-off terminal, tugging time reduces to zero. The model has
9.3. Final remark

not yet been tested on zero tugging time. It is, therefore, possible that the model will give unreliable results. During the validation, the inner and outer approach channel and the location of the breakwaters should therefore be taken into account.

Furthermore, if the model can deal with several vessel types, a possible addition would be to apply the model to multiple vessel types for one breakwater and port design. Ports are often intended for multiple vessel types, which means that the model is currently insufficient to design these types of ports. The model will have to design different terminals to be widely applicable, as some vessel types need a different terminal design. Finally, the most stringent design vessel requirements must be considered when designing the breakwater and approach channel.

9.3. Final remark
This research increased the model's value by verification, internal and external validation, and improved applicability and performance. Still, before the model can be used in practice, even more adjustments will be needed. This study can be used for further development of the model.
Bibliography


Aslansefat, K. (n.d.). How to validate the correctness of an evolutionary optimization algorithm.


Google. (n.d.). [https://www.google.nl/maps/place/Cape%20Town,%20South%20Africa/@-33.8988717,18.400808,16716m/data=!3m1!1e3!4m5!3m4!1s0x1dcc500f8826eed7:0x687fe1fc2828aa87!8m2!3d-33.9248685!4d18.4240553](https://www.google.nl/maps/place/Cape%20Town,%20South%20Africa/@-33.8988717,18.400808,16716m/data=!3m1!1e3!4m5!3m4!1s0x1dcc500f8826eed7:0x687fe1fc2828aa87!8m2!3d-33.9248685!4d18.4240553)


UNCTAD. (1978). *Port development: A handbook for planners in developing countries*. UN.


Table A.1 provide the input values for the current model.
### A. Input values

<table>
<thead>
<tr>
<th>Category</th>
<th>Symbol</th>
<th>.py symbol</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Constraints</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$x_{max}$</td>
<td>$x_{\text{max}}$</td>
<td>m</td>
<td></td>
<td>outermost point for primary breakwater on x-axis</td>
</tr>
<tr>
<td>$y_{max}$</td>
<td>$y_{\text{max}}$</td>
<td>m</td>
<td></td>
<td>outermost point for primary breakwater on y-axis</td>
</tr>
<tr>
<td>$L_{\text{max}}$</td>
<td>$l_{\text{max}}$</td>
<td>m</td>
<td></td>
<td>outermost distance for secondary breakwater in x-direction</td>
</tr>
<tr>
<td><strong>Navigation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$H_c$</td>
<td>$h_{\text{ch}}$</td>
<td>m</td>
<td></td>
<td>limiting wave height in sheltered channel</td>
</tr>
<tr>
<td>$H_{\text{berth}}$</td>
<td>$h_{\text{berth}}$</td>
<td>m</td>
<td></td>
<td>limiting wave height at berth</td>
</tr>
<tr>
<td>$v_{\text{S,min}}$</td>
<td>$v_{\text{Smin}}$</td>
<td>m/s</td>
<td></td>
<td>minimum vessel speed</td>
</tr>
<tr>
<td>$t_{\text{tug}}$</td>
<td>$t_{\text{tug}}$</td>
<td>min</td>
<td></td>
<td>tugging time</td>
</tr>
<tr>
<td>$u_{\text{CWN}}$</td>
<td>$u_{\text{CWN}}$</td>
<td>m/s</td>
<td></td>
<td>limiting cross-wind speed for navigation</td>
</tr>
<tr>
<td>$u_{\text{LWN}}$</td>
<td>$u_{\text{LWN}}$</td>
<td>m/s</td>
<td></td>
<td>limiting longitudinal wind speed for navigation</td>
</tr>
<tr>
<td>$u_{\text{WO}}$</td>
<td>$u_{\text{WO}}$</td>
<td>m/s</td>
<td></td>
<td>limiting wind speed for operations</td>
</tr>
<tr>
<td>$u_{\text{CC}}$</td>
<td>$u_{\text{CC}}$</td>
<td>m/s</td>
<td></td>
<td>limiting cross-current speed for navigation</td>
</tr>
<tr>
<td>$u_{\text{LC}}$</td>
<td>$u_{\text{LC}}$</td>
<td>m/s</td>
<td></td>
<td>limiting longitudinal current speed for navigation</td>
</tr>
<tr>
<td><strong>Vessel</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_s$</td>
<td>$l_s$</td>
<td>m</td>
<td></td>
<td>length over all of design vessel</td>
</tr>
<tr>
<td>$B_s$</td>
<td>$b_s$</td>
<td>m</td>
<td></td>
<td>beam of design vessel</td>
</tr>
<tr>
<td>$D_s$</td>
<td>$d_s$</td>
<td>m</td>
<td></td>
<td>draught of design vessel</td>
</tr>
<tr>
<td>$v_s$</td>
<td>$v_s$</td>
<td>m/s</td>
<td></td>
<td>entrance speed of a vessel</td>
</tr>
<tr>
<td><strong>Costs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Core</td>
<td>$c_{\text{core}}$</td>
<td>/m$^3$</td>
<td></td>
<td>unit price for breakwater core material</td>
</tr>
<tr>
<td>Under</td>
<td>$c_{\text{under}}$</td>
<td>/m$^3$</td>
<td></td>
<td>unit price for breakwater underlayer material</td>
</tr>
<tr>
<td>Armour</td>
<td>$c_{\text{armour}}$</td>
<td>/unit</td>
<td></td>
<td>unit price for breakwater armour layer material</td>
</tr>
<tr>
<td>Dd</td>
<td>$c_{\text{dd}}$</td>
<td>/m$^3$</td>
<td></td>
<td>dredging-dumping costs of sediment</td>
</tr>
<tr>
<td>Df</td>
<td>$c_{\text{df}}$</td>
<td>/m$^3$</td>
<td></td>
<td>dredging-filling costs of sediment</td>
</tr>
<tr>
<td>TEU</td>
<td>$c_{\text{TEU}}$</td>
<td>/TEU</td>
<td></td>
<td>(un)loading unit rate per twenty-foot equivalent</td>
</tr>
<tr>
<td>Land</td>
<td>$c_{\text{land}}$</td>
<td>/m'</td>
<td></td>
<td>land cost per running meter of coastline stretch</td>
</tr>
<tr>
<td>Breakwater Maintenance</td>
<td>$\psi_{\text{bw,m}}$</td>
<td></td>
<td></td>
<td>ratio of annual breakwater maintenance costs w.r.t. construction</td>
</tr>
<tr>
<td>r</td>
<td></td>
<td></td>
<td></td>
<td>discount rate for present-day value</td>
</tr>
<tr>
<td>$T_L$</td>
<td></td>
<td>yr</td>
<td></td>
<td>design lifetime of breakwater</td>
</tr>
<tr>
<td><strong>Terminal</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$n_b$</td>
<td>$n_b$</td>
<td></td>
<td></td>
<td>number of berths second phase</td>
</tr>
<tr>
<td>$n_{bf}$</td>
<td>$n_{bf}$</td>
<td></td>
<td></td>
<td>number of berths in future</td>
</tr>
<tr>
<td>$y_f$</td>
<td>$y_f$</td>
<td>m</td>
<td></td>
<td>fill-up distance in y-direction</td>
</tr>
<tr>
<td>$m_b$</td>
<td>$m_b$</td>
<td></td>
<td></td>
<td>berth occupancy factor</td>
</tr>
<tr>
<td>$P_b$</td>
<td>$p_b$</td>
<td>MPH</td>
<td></td>
<td>hourly production per berth</td>
</tr>
<tr>
<td>TEU-factor</td>
<td>$f_{\text{TEU}}$</td>
<td></td>
<td></td>
<td>TEU-factor</td>
</tr>
<tr>
<td><strong>Bathymetry</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$z_0$</td>
<td>$z_0$</td>
<td>m</td>
<td></td>
<td>ground level above MSL at y = 0</td>
</tr>
<tr>
<td>$\alpha_b$</td>
<td>$\alpha_b$</td>
<td></td>
<td></td>
<td>slope of the sea bed</td>
</tr>
<tr>
<td><strong>Coast</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\theta_s$</td>
<td>$\theta_s$</td>
<td>$^\circ$</td>
<td></td>
<td>orientation of offshore-directed normal to coastline w.r.t. North</td>
</tr>
<tr>
<td><strong>Environment</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$s_{\text{max}}$</td>
<td>$s_{\text{max}}$</td>
<td></td>
<td></td>
<td>maximum directional spreading parameter</td>
</tr>
<tr>
<td><strong>Water levels</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HAT</td>
<td>$\text{HAT}$</td>
<td>m</td>
<td></td>
<td>high astronomical tide level above MSL</td>
</tr>
<tr>
<td>LAT</td>
<td>$\text{LAT}$</td>
<td>m</td>
<td></td>
<td>low astronomical tide level below MSL</td>
</tr>
<tr>
<td><strong>Breakwater</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_l$</td>
<td>$h_l$</td>
<td>m</td>
<td></td>
<td>level of leeward armour layer below MSL</td>
</tr>
<tr>
<td>$h_c$</td>
<td>$h_c$</td>
<td>m</td>
<td></td>
<td>core height above HAT</td>
</tr>
<tr>
<td>$t_u$</td>
<td>$t_u$</td>
<td>m</td>
<td></td>
<td>thickness of underlayer</td>
</tr>
<tr>
<td>$t_a$</td>
<td>$t_a$</td>
<td>m</td>
<td></td>
<td>thickness of armour layer</td>
</tr>
<tr>
<td>$b_{\text{crest}}$</td>
<td>$b_{\text{crest}}$</td>
<td>m</td>
<td></td>
<td>breakwater crest width</td>
</tr>
<tr>
<td>$l_{:x_f}$</td>
<td>$l_{:x_f}$</td>
<td>-</td>
<td></td>
<td>front slope of breakwater</td>
</tr>
<tr>
<td>$l_{:x_r}$</td>
<td>$l_{:x_r}$</td>
<td>-</td>
<td></td>
<td>rear slope of breakwater</td>
</tr>
<tr>
<td>$n_v$</td>
<td>$n_v$</td>
<td></td>
<td></td>
<td>armour layer porosity</td>
</tr>
<tr>
<td>$D_a$</td>
<td>$D_a$</td>
<td>m</td>
<td></td>
<td>nominal diameter of armour units</td>
</tr>
<tr>
<td><strong>Sediments</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t_{s,0}$</td>
<td>$t_{s,0}$</td>
<td>m/year</td>
<td></td>
<td>annual siltation thickness without secondary breakwater</td>
</tr>
<tr>
<td>$t_{s,max}$</td>
<td>$t_{s,max}$</td>
<td>m/year</td>
<td></td>
<td>annual siltation thickness with breakwater gap &gt;= 1000m</td>
</tr>
<tr>
<td>$t_{s,min}$</td>
<td>$t_{s,min}$</td>
<td>m/year</td>
<td></td>
<td>annual siltation thickness with breakwater gap equal to channel width</td>
</tr>
<tr>
<td>$\text{S}_{\text{type}}$</td>
<td>$s_{\text{type}}$</td>
<td></td>
<td></td>
<td>soil type (1 = mud, 2 = sand/clay, 3 = rock/coral)</td>
</tr>
<tr>
<td><strong>Dredging</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_f$</td>
<td>$h_f$</td>
<td>m</td>
<td></td>
<td>average fill level of terminal above MSL</td>
</tr>
<tr>
<td>$x_b$</td>
<td>$x_b$</td>
<td>-</td>
<td></td>
<td>bank slope of approach channel</td>
</tr>
<tr>
<td><strong>Grid</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\delta x y$</td>
<td>$\delta x y$</td>
<td>-</td>
<td></td>
<td>step size for grid</td>
</tr>
<tr>
<td>Layout</td>
<td>$\text{lay_dir}$</td>
<td>-</td>
<td></td>
<td>$r$ = rightward direction, $l$ = leftward direction, $\text{np}$ = no preference</td>
</tr>
</tbody>
</table>

Table A.1: Input parameter for Teeling [2020] and current model (Teeling, 2020).
This chapter provides a manual how to download the python script and what to install before execution of the tool is possible. The code is published using GitHub. The following link will take you to the breakwater model:

https://github.com/TUDelft-CITG/GA-BreakwaterLayout

or scan the qr code provided in figure B.1 to find the Python script of the Breakwater model. At the time of writing access to the model is only possible on request.

Figure B.1: Breakwater model
Alongshore positioning

The goal of changing the alongshore position was to change \( x_0 \) and \( x_1 \) from fixed values to implemented in the genetic algorithm. Furthermore, the quay wall was based on \( x_0 = 0 \) and was a key factor for the determining of possible intersection between approach channel, turning basin and breakwaters, quay wall and maneuvering basin.

Notes:

1. \( \text{Self} \cdot x_0 = 0 \), means a fixed breakwater. To define the node, None values are used. Be aware by changing this, as most of the code generates it's layouts based on this value.

2. \( \text{V.lfq} \rightarrow \text{self.lfq} \) or \( \text{v.lq} \rightarrow \text{self.lq} \) is to create a terminal that moves with the moving breakwater. This causes no intersects between the breakwaters and the turning basin, quay wall or approach channel. It means: \( \text{v.lq} + \text{self.xo} \), the quay wall/terminal length + the \( x \)-value of the breakwater.

3. Other comments are mentioned after the statement.

Every subsection discusses the changes and additions to the parts of the code.

**Generator**

**Changes**

**Init:**

- \( \text{Self} \cdot x_0 = 0 \rightarrow \text{None} \).
- \( \text{Self} \cdot x_1 = 0 \rightarrow \text{None} \).

**Coords:**

- \( \text{V.lfq} \rightarrow \text{self.lfq} \).

**Turning basin stop:**

- \( \text{Self} \cdot x_2 < \text{v.lq} \rightarrow \text{self.x2} < \text{self.lq} \)

**Channel edges:**

- \( \text{V.lfq} \rightarrow \text{self.lfq} \)

**Node6:**
• V.lfq > self.lfq

Flip layout:

• Self.x0 = -1 -> -self.x0 - 1
• Self.x0 = -1 -> -self.x1 - 1
• Self.lfq = -v.lfq -> -self.lfq

Additions Coords:

• self.x0 = self.x1 = int(random.uniform(0,var.xmax)) to create variable breakwater coordinates along the x-axis
• self.lq = self.x0 + v.lfq

Chromosome:

• ‘x0’. To add the variable self.x0 to the genetic algorithm

**Mutation**

*Changes*

Mutation:

• Self.x0 = 0 -> self.original.x0
• Self.x1 = 0 -> self.original.x1

Intersecting:

• V.lfq -> self.lfq

Set rhs:

• V.lfq -> self.lfq

Sec node:

• V.lfq -> self.lfq

Turn coords:

• V.lfq -> self.lfq

Set sec:

• V.lfq -> self.lfq

Additions: Set rhs:

• Self.x0 = -self.x0
• Self.x1 = -self.x1

An extra function was created which was called prim node. Prim node create the \(x_0\) and \(x_1\) x-coordinates.
85

- if gene == 'x0':
  self.x0 = self.x1 = int(np.random.normal(allele, abs(self.mut_size * allele)))
  if self.sec == 1:
    d = f.min_perpendicular_distance(self.x1, self.y1, self.x2, self.y2, self.x3, self.y3, self.x6, self.y6)
  else:
    d = 99999
  loop = 0
  while f.min_perpendicular_distance(self.x1, self.y1, self.x2, self.y2, self.x3, self.y3, self.xtb, self.ytb)
    < 0.5 * v.dtb + var.xb * (v.d_hb + max(-v.d_hb, var.z0 - self.y3 / var.alpha_b)) or self.intersecting(1, 2) or f.inequality_rhs(self.x0, self.rhs, var.l_max, pos=2) or f.inequality_rhs(self.x0, self.rhs, self.x3) or f.inequality_rhs(self.x0, self.rhs, self.x3, pos=1):
    # print('x4 not valid')
    self.x0 = int(np.random.normal(allele, abs(self.mut_size * (allele - self.lfq))))
    self.x1 = self.x0
    loop += 1
    if loop > 20:
      # print('Mutation of x4 is negated')
      self.x0, self.x1 = allele, allele
      break
  if self.sec == 1:
    d = f.min_perpendicular_distance(self.x1, self.y1, self.x2, self.y2, self.x3, self.y3, self.x6, self.y6)
  else:
    d = 99999
- if self.sec == 1:
  d = f.min_perpendicular_distance(self.x1, self.y1, self.x2, self.y2, self.x3, self.y3, self.x6, self.y6)
  else:
    d = 99999

Chromosome:
  'x0'

Generate layout:
  if random.random() < self.mut_rate:
    self.prim_node('x0', self.x0, None). To place the self.x0 and self.x1 in the genetic algorithm generation.

Mating
Changes Child:
  - Self.x0 = 0 -> None
  - Self.x1 = 0 -> None
  - Self.lq = v.lq -> None
  - Self.lqf = v.lfq -> None

Turn basin:
- `self.x0 = 0 -> self.x0`
- `v.lfq -> self.lfq`

**Intersecting:**
- `V.lfq -> self.lfq`

**Secondary:**
- `V.lfq -> self.lfq. Here it was to help x4 not intersect with the quay wall or primary breakwater.`

**Stopping length:**
- `Self.lq = -v.lq -> -self.lq`
- `Self.lfq = -v.lfq -> -self.lfq`
- `Self.lq = v.lq -> self.lq`
- `Self.lfq = v.lfq -> self.lfq`

**Copy:**
- `Self.lq = -v.lq -> -self.lq`
- `Self.lfq = -v.lfq -> -self.lfq`
- `Self.lq = v.lq -> self.lq`
- `Self.lfq = v.lfq -> self.lfq`

**Generate layout:**
- `self.lq, self.lfq = -v.lq, -v.lfq -> -self.lq, -self.lfq`

**Best solution:**
- `self.lq = -v.lq -> -v.lq + self.x0`. Here, it’s not possible to replace `v.lq + self.x0` with `self.lq`, as otherwise the genetic algorithm will randomly choose a quay length, which is defined on forehand.
- `self.lq = v.lq -> v.lq + self.x0`

**Additions** Create a new function for \( x_0 \) and \( x_1 \), called `prim_node`:

- `self.x0 = self.x1 = int(random.uniform(self.p1['x0'], self.p2['x0']))`
  - if `self.p1['rhs'] == self.p2['rhs']:`
  - `self.x0 = self.x1 = int(random.uniform(self.p1['x0'], self.p2['x0']))`
  - `elif self.p1['rhs'] == self.rhs:`
  - `self.x0 = -self.x0`
  - `self.x1 = -self.x1`
  - `else:`
  - `self.x0 = self.p2['rhs'] == 1:`
  - `self.x0 = -self.x0`
  - `self.x1 = self.p2['rhs'] == 1:`
  - `self.x0 = -self.x0`
  - `self.x1 = -self.x1`
- `Self.lq = self.x0 + v.lq`
- `Self.lfq = self.x0 + v.lfq`

Intersecting:
- `Self.primnode()` to run the primnode() equation for `self.x0` and `self.x1`.

Primary:
- `Self.primnode()` to run the primnode() equation for `self.x0` and `self.x1`.

Secondary:
- `Self.primnode()` to run the primnode() equation for `self.x0` and `self.x1`.

Stopping length:
- `Self.primnode()` to run the primnode() equation for `self.x0` and `self.x1`.

Generate layout:
- `Self.primnode()` to run the primnode() equation for `self.x0` and `self.x1`.
- `Self.lq = -v.lq -> -v.lq + self.x0`

Best solution:
- `Self.lq = v.lq -> None`
- `Self.x0, Self.x1 = 0, 0 -> self.c['x0'], self.c['x1']`
An example input file for REFRAC is shown in figure D.1.

Figure D.1: Example input file for REFRAC

Figure D.2 shows the aspects notable and which can determine the wave field obtained by REFRAC.

(a) Wave period and offshore wave height (b) Wave energy averaging over x- and y-cells (c) The adjustable wave direction highlighted with the red rectangular.

Figure D.2: The input data for REFRAC with explanation on the most important variable parameters
It is recommended to read the REFRAC manual for a complete overview.
Validation

E.1. Equations
This section provides the reader the equations applied by the model to generate the various elements depending on the design criteria.

E.1.1. Port layout
The breakwater layout model incorporates criteria for nautical safety of the port dimensions (Woerlee, 2019).

**Channel width**  The channel width is estimated by Equation E.1 in accordance to (PIANC, 2014) and implemented by Woerlee (2019).

\[
W = W_{BM} + \sum W_i + W_B
\] (E.1)

In which \(W_{BM}\) is the basic manoeuvring basin, \(W_i\) represents the additional width, \(B_s\) is the width of the design vessel and \(W_B\) is the bank clearance. The additional width \(W_i\) is determined by table E.1.

**Channel depth**  The depth of the approach channel is estimated using Equation E.5 according to (PIANC, 2014), implemented by Woerlee (2019).

\[
d_{ch} = \begin{cases} 
1.1D + 0.015B_s + d_b & \text{Inner channel} \\
1.3D + 0.015B_s + d_b & \text{Outer channel} 
\end{cases}
\] (E.2)

In which \(D\) is the draught of the vessel and \(d_b\) is the bottom type factor.
Table E.1: Additional width to the approach channel (Woerlee, 2019)

<table>
<thead>
<tr>
<th>Width</th>
<th>Condition</th>
<th>Outer channel</th>
<th>Inner channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prevailing cross-wind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mild</td>
<td>$u_{cw} &lt; 7$ m/s</td>
<td>0.2 $B_s$</td>
<td>0.2 $B_s$</td>
</tr>
<tr>
<td>Moderate</td>
<td>$7 \leq u_{cw} &lt; 16$ m/s</td>
<td>0.4 $B_s$</td>
<td>0.6 $B_s$</td>
</tr>
<tr>
<td>Strong</td>
<td>$16 \leq u_{cw} &lt; 24$ m/s</td>
<td>0.7 $B_s$</td>
<td>n/a</td>
</tr>
<tr>
<td>Prevailing cross-currents</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negligible</td>
<td>$u_{cc} &lt; 0.1$ m/s</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Low</td>
<td>$0.1 &lt; u_{cc} &lt; 0.25$ m/s</td>
<td>0.25 $B_s$</td>
<td>0.2 $B_s$</td>
</tr>
<tr>
<td>Moderate</td>
<td>$0.25 &lt; u_{cc} &lt; 0.75$ m/s</td>
<td>0.7 $B_s$</td>
<td>0.6 $B_s$</td>
</tr>
<tr>
<td>High</td>
<td>$0.75 &lt; u_{cc} &lt; 1.0$ m/s</td>
<td>1.2 $B_s$</td>
<td>n/a</td>
</tr>
<tr>
<td>Prevailing longitudinal current (ulc)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>$u_{lc} &lt; 0.75$ m/s</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Moderate</td>
<td>$0.75 &lt; u_{lc} &lt; 1.5$ m/s</td>
<td>0.1 $B_s$</td>
<td>0.2 $B_s$</td>
</tr>
<tr>
<td>High</td>
<td>$u_{lc} &lt; 1.5$ m/s</td>
<td>0.2 $B_s$</td>
<td>0.2 $B_s$</td>
</tr>
<tr>
<td>Beam and stern quartering wave height (Hbq)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>$H_{bq} \leq 1$ m</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Moderate</td>
<td>$1 &lt; H_{bq} &lt; 1.5$ m</td>
<td>0.2 $B_s$</td>
<td>0.4 $B_s$</td>
</tr>
<tr>
<td>High</td>
<td>$H_{bq} \geq 1.5$ m</td>
<td>0.4 $B_s$</td>
<td>n/a</td>
</tr>
<tr>
<td>Aids to navigation</td>
<td>Good</td>
<td>0.2 $B_s$</td>
<td>n/a</td>
</tr>
<tr>
<td>Bottom surface</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smooth and soft</td>
<td></td>
<td>0.1 $B_s$</td>
<td>0.2 $B_s$</td>
</tr>
<tr>
<td>Rough and hard</td>
<td></td>
<td>0.2 $B_s$</td>
<td>n/a</td>
</tr>
<tr>
<td>Depth of waterway</td>
<td>Relatively small</td>
<td>0.2 $B_s$</td>
<td>0.4 $B_s$</td>
</tr>
</tbody>
</table>

Table E.1 shows the standard values for the bottom type according to PIANC (2014).

Table E.2: Additional approach channel depth Woerlee (2019)

<table>
<thead>
<tr>
<th>Bottom type</th>
<th>Inner channel</th>
<th>Outer channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mud</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sand/clay</td>
<td>0.4 m</td>
<td>0.5 m</td>
</tr>
<tr>
<td>Rock/coral</td>
<td>0.6 m</td>
<td>1.0 m</td>
</tr>
</tbody>
</table>

**Channel length**  The inner approach channel is estimated using Equation (E.3) according to PIANC (2014), implemented by Woerlee (2019).

$$L_{ch,in} = \begin{cases} L_{mild} + L_{final} & \text{for } H_s < 1.5\cdot3.0 \text{ m and } v_s < 2.5\cdot3.0 \text{ m/s} \\ L_{rough} + L_{final} & \text{for } H_s > 1.5\cdot3.0 \text{ m and } v_s > 2.5\cdot3.0 \text{ m/s} \end{cases}$$ (E.3)

In which $L_{mild}$ is the time necessary to tie up a tugboat to the design vessel, $L_{rough}$ length needed to slow down the design vessel to 2 m/s and tie up a tugboat and $L_{final}$ is the final stopping distance:

$$L_{mild} = t_{tug} \cdot v_{s,min}$$

$$L_{rough} = \max((v_s - v_{s,min})^2 L_s; t_{tug} \cdot v_{s,min}) + L_s$$

$$L_{final} = 1.5 \cdot L_s$$

**Turning basin**  The turning basin estimations were implemented by Woerlee (2019):

- The diameter of the turning basin equals $2.5 \cdot L_s$.
- The depth of the turning basin equals the depth of the inner approach channel.
E.2. Computation time

Depth of the manoeuvring basin and berthing basin  
Equation [E.4] provides the estimation model for the depth of the manoeuvring basin and depth at the berths, implemented by Woerlee (2019).

\[ d_{bb} = LAT + D + d_b + 0.7 \]  

(E.4)

Harbour entrance  
The minimum width of the harbour entrance equals \( L_s \).

Quay  
The quay length is estimated using Equation [E.5] according to Ligteringen (2012), implemented by Woerlee (2019).

\[ L_q = \begin{cases} 
L_s + 2 \times 15 & \text{for } n_b = 1 \\
1.1 \times n_b \times (L_s + 15) + 15 & \text{for } n_b > 1 
\end{cases} \]  

(E.5)

E.2. Computation time

This section provides the results for the tuning of the optimisation parameters.

E.2.1. Population size

<table>
<thead>
<tr>
<th>Population size [-]</th>
<th>Computation time [min]</th>
<th>Fitness score [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>101.29</td>
<td>1.105</td>
</tr>
<tr>
<td>120</td>
<td>121.08</td>
<td>1.19</td>
</tr>
<tr>
<td>120</td>
<td>110.55</td>
<td>1.149</td>
</tr>
<tr>
<td>60</td>
<td>68.5</td>
<td>1.10</td>
</tr>
<tr>
<td>60</td>
<td>60.23</td>
<td>1.18</td>
</tr>
<tr>
<td>60</td>
<td>79.15</td>
<td>1.08</td>
</tr>
<tr>
<td>48</td>
<td>36.14</td>
<td>1.106</td>
</tr>
<tr>
<td>48</td>
<td>42.43</td>
<td>1.149</td>
</tr>
<tr>
<td>48</td>
<td>31.39</td>
<td>1.177</td>
</tr>
<tr>
<td>24</td>
<td>23.12</td>
<td>0.955</td>
</tr>
<tr>
<td>24</td>
<td>26.28</td>
<td>0.984</td>
</tr>
<tr>
<td>24</td>
<td>16.53</td>
<td>0.938</td>
</tr>
<tr>
<td>12</td>
<td>9.37</td>
<td>0.749</td>
</tr>
<tr>
<td>12</td>
<td>10.5</td>
<td>0.702</td>
</tr>
<tr>
<td>12</td>
<td>10.2</td>
<td>0.709</td>
</tr>
</tbody>
</table>
E.2.2. Mutation rate

Table E.4: Results for the mutation rate test case

<table>
<thead>
<tr>
<th>Mutation rate</th>
<th>Computation time</th>
<th>Fitness score</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>55.08</td>
<td>0.521</td>
</tr>
<tr>
<td>0.4</td>
<td>51</td>
<td>0.437</td>
</tr>
<tr>
<td>0.4</td>
<td>48.12</td>
<td>0.612</td>
</tr>
<tr>
<td>0.35</td>
<td>31.05</td>
<td>0.631</td>
</tr>
<tr>
<td>0.35</td>
<td>26.1</td>
<td>0.543</td>
</tr>
<tr>
<td>0.35</td>
<td>32.08</td>
<td>0.684</td>
</tr>
<tr>
<td>0.3</td>
<td>18.09</td>
<td>0.512</td>
</tr>
<tr>
<td>0.3</td>
<td>25.19</td>
<td>0.538</td>
</tr>
<tr>
<td>0.3</td>
<td>31.27</td>
<td>0.631</td>
</tr>
<tr>
<td>0.25</td>
<td>34.21</td>
<td>1.151</td>
</tr>
<tr>
<td>0.25</td>
<td>28.55</td>
<td>1.121</td>
</tr>
<tr>
<td>0.25</td>
<td>24.25</td>
<td>1.081</td>
</tr>
<tr>
<td>0.2</td>
<td>28.48</td>
<td>0.937</td>
</tr>
<tr>
<td>0.2</td>
<td>21</td>
<td>0.932</td>
</tr>
<tr>
<td>0.2</td>
<td>22.4</td>
<td>0.864</td>
</tr>
<tr>
<td>0.15</td>
<td>29.34</td>
<td>1.002</td>
</tr>
<tr>
<td>0.15</td>
<td>24.34</td>
<td>0.931</td>
</tr>
<tr>
<td>0.15</td>
<td>23.51</td>
<td>0.942</td>
</tr>
<tr>
<td>0.1</td>
<td>25.09</td>
<td>0.901</td>
</tr>
<tr>
<td>0.1</td>
<td>9.15</td>
<td>0.75</td>
</tr>
<tr>
<td>0.1</td>
<td>21.46</td>
<td>0.86</td>
</tr>
</tbody>
</table>

E.2.3. Mutation size

Table E.5: Results for the mutation size test case

<table>
<thead>
<tr>
<th>Mutation size</th>
<th>Computation time</th>
<th>Fitness score</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>16.10</td>
<td>1.054</td>
</tr>
<tr>
<td>0.3</td>
<td>18.26</td>
<td>1.034</td>
</tr>
<tr>
<td>0.3</td>
<td>25.05</td>
<td>1.102</td>
</tr>
<tr>
<td>0.2</td>
<td>24.12</td>
<td>1.167</td>
</tr>
<tr>
<td>0.2</td>
<td>21.15</td>
<td>1.154</td>
</tr>
<tr>
<td>0.2</td>
<td>17.29</td>
<td>1.095</td>
</tr>
<tr>
<td>0.18</td>
<td>26.35</td>
<td>1.056</td>
</tr>
<tr>
<td>0.18</td>
<td>34.50</td>
<td>1.138</td>
</tr>
<tr>
<td>0.18</td>
<td>33.00</td>
<td>1.128</td>
</tr>
<tr>
<td>0.15</td>
<td>39.54</td>
<td>1.11</td>
</tr>
<tr>
<td>0.15</td>
<td>35.24</td>
<td>1.055</td>
</tr>
<tr>
<td>0.15</td>
<td>30.42</td>
<td>1.067</td>
</tr>
<tr>
<td>0.1</td>
<td>37.25</td>
<td>1.096</td>
</tr>
<tr>
<td>0.1</td>
<td>31.53</td>
<td>1.085</td>
</tr>
<tr>
<td>0.1</td>
<td>42.16</td>
<td>1.154</td>
</tr>
</tbody>
</table>