

Chapter 3

Literature Review: Methods

3.1 Radial Basis Function interpolation

Radial basis functions (RBFs) have become increasingly popular over the last three decades. In essence, RBFs are typically used to interpolate multivariate scattered or gridded data [38]. The interpolation function, or the interpolant, is described as a linear combination of specified continuous basis functions, as shown in Equation (3.1):

$$s(\mathbf{x}) = \sum_{j=1}^N c_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) , \quad (3.1)$$

where ϕ is a specified basis function, $\tilde{\phi} : \mathbb{R}^d \rightarrow \mathbb{R}$, and $\tilde{\phi}(\mathbf{x}) = \phi(\|\mathbf{x}\|)$ is radial with respect to the Euclidean norm $\|\cdot\|$ [38]. It is shown in the literature that the use of these “classical” radial basis functions yields reliable approximation behaviour [38].

RBFs are often used in engineering, where the RBF returns a value as a function of the radius from a given centre. Furthermore, they were originally designed for interpolating scattered data [54]. The interpolation function, $s(\mathbf{x})$, which is a function value at any point in the domain under consideration, can be computed by a sum of radial basis functions as shown in Equation (3.2):

$$s(\mathbf{x}) = \sum_{j=1}^{n_b} \alpha_j \phi(\|\mathbf{x} - \mathbf{x}_{\mathbf{b}_j}\|) + p(\mathbf{x}) . \quad (3.2)$$

Here $\mathbf{x}_{\mathbf{b}_j} = [x_{\mathbf{b}_j}^1, x_{\mathbf{b}_j}^2, \dots, x_{\mathbf{b}_j}^d]$ are the data points, or centres, for which the values are known; $p(\mathbf{x})$ is a polynomial, $p(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$, which can differ according to the problem, n_b is the number of points

for which values are known and ϕ is a specific radial basis function with respect to the Euclidean distance [30]. The coefficients α_j are determined for the specific problem on which this method is applied. According to de Boer *et al.* [30], the values for the coefficients α_j as well as the coefficients for the polynomial (β_i) can be determined by solving the following system:

$$\begin{bmatrix} \mathbf{d}_b \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{b,b} & \mathbf{P}_b \\ \mathbf{P}_b^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}. \quad (3.3)$$

In this system, $\boldsymbol{\alpha}$ contains the coefficients, α_j , and $\boldsymbol{\beta}$ contains the coefficients for the polynomial p . $\mathbf{M}_{b,b}$ is an $n_b \times n_b$ matrix that contains the computed radial basis functions $\phi_{b,b_j} = \phi(\|\mathbf{x}_b - \mathbf{x}_{b_j}\|)$, and \mathbf{P}_b is an $n_b \times (d+1)$ matrix with row j represented by $[1 \ x_{b_j}^1 \ x_{b_j}^2 \ \dots \ x_{b_j}^d]$ in the case of a linear polynomial p . According to de Boer *et al.*, the system in Equation (3.3) can be solved using iterative techniques or by using the partition of unity method [30].

Two types of radial basis functions exist: global support functions and local support functions. Global support in radial basis function interpolation is a method that has the ability to fit functions to scattered data [48]. Global support functions support the whole area of interpolation and do not have a zero contribution outside a certain radius from a specific point. Dense matrix systems are created, as a consequence of global support functions [30]. When the size of the data set is very large, global support, where the interpolated value is influenced by all the data, becomes a time consuming and costly method to use [48]. Six examples of popular global RBFs are given in Table 3.1. These functions are used for different applications including neural networks, in computer graphics and for fluid mechanics [30]. The smoothness of the RBF, ϕ , has an effect on the reproduction quality of the interpolation function obtained; a smoother RBF results in a better reproductive quality [48]. In these functions, the constant a can be adjusted according to the problem. A large value for a results in a flat sheet-like function whereas a small value for a yields a narrow cone-like function [30].

Local support functions can handle much larger data sets effectively and allow for parallel implementation. However, local support functions have an increased complexity when d -dimensional problems are considered [48]. Despite the increased complexity of the local support methods, the global support methods use more computational time for larger data sets than local support methods [38]. There are a vast range of RBFs available that can be used for the interpolation of multivariate data [30]. Compact support functions are also used in various interpolation applications for multivariate data [30]. Compact or local support functions are functions which have the property that each point in the interpolation area only affects an interpolation area within a certain radius of that point. This is the property given by:

Table 3.1: Radial basis functions with global support [30]

No	Name	Abbreviation	$\phi(x)$
1	Thin plate spline	TPS	$x^2 \log(x)$
2	Multiquadratic biharmonics	MQB	$\sqrt{a^2 + x^2}$
3	Inverse multiquadratic biharmonics	IMQB	$\sqrt{\frac{1}{a^2 + x^2}}$
4	Quadratic biharmonics	QB	$1 + x^2$
5	Inverse quadratic biharmonics	IQB	$\frac{1}{1 + x^2}$
6	Gaussian	gauss	e^{-x^2}

$$\phi(x) = \begin{cases} f(x) & \text{if } 0 \leq x \leq 1 \\ 0 & \text{if } x > 1 \end{cases}, \quad (3.4)$$

where $f(x) \geq 0$. Scaling is generally used with local support functions, so that the support radius r supports the area of support by the compact support radial basis function, thus $\phi_r = \phi(\frac{x}{r})$ [54], [30]. In this case, only the points in a radius r around a given point are affected by that point [30].

Some examples of compact support functions are given in Table 3.2. The majority of compactly supported radial basis functions spans a radius of one. The radius in this case is scaled with r , such that, $\xi = \frac{x}{r}$ [30]. The first four functions are based on polynomials selected so that they have the lowest degree possible and still create a C^n continuous basis function where $n \in \{0, 2, 4, 6\}$. The last four functions are based on the thin plate spline equation and create C^n continuous basis functions with $n \in \{0, 1, 2\}$ [30]. Compactly supported radial basis functions have been successfully applied in various applications and yield good accuracy for the majority of the problems to which they have been applied [54].

The RBFs that are used to create the matrix in Equation (3.3) are said to be positive definite functions if the matrix is a positive definite matrix [45]. An $n \times n$ matrix is said to be positive definite if $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$, for all non-zero column vectors, \mathbf{x} , in Euclidean n -dimensional space [45]. Since the RBF is compactly supported, the RBF is automatically positive definite [35]. The matrix that is constructed from the positive definite functions, as in RBF interpolation, is also positive definite [59]. The strictly positive definite matrix is invertible, therefore the system in Equation (3.3) can be solved directly. The compactly supported radial basis functions are only strictly positive definite on \mathbb{R}^d , for a fixed maximum value for d . It is impossible for a function, $\phi(\xi)$, to be radial

Table 3.2: Radial basis functions with local support [30]

No	Name	$\phi(\xi)$
1	CP C^0	$(1 - \xi)^2$
2	CP C^2	$(1 - \xi)^4(4\xi + 1)$
3	CP C^4	$(1 - \xi)^6(\frac{35}{3}\xi^2 + 6\xi + 1)$
4	CP C^6	$(1 - \xi)^8(32\xi^3 + 25\xi^2 + 8\xi + 1)$
5	CTPS C^0	$(1 - \xi)^5$
6	CTPS C^1	$1 + \frac{80}{3}\xi^2 - 40\xi^3 + 15\xi^4 - \frac{8}{3}\xi^5 + 20\xi^2 \log(\xi)$
7	CTPS C_a^2	$1 - 30\xi^2 - 10\xi^3 + 45\xi^4 - 6\xi^5 - 60\xi^3 \log(\xi)$
8	CTPS C_b^2	$1 - 20\xi^2 + 80\xi^3 - 45\xi^4 - 16\xi^5 + 60\xi^4 \log(\xi)$

Table 3.3: Wu's compactly supported radial basis functions for different values of k and for $l = 3$ [35].

k	$\psi_{k,3}(r)$	d
0	$(1 - r)^7(5 + 35r + 101r^2 + 147r^3 + 101r^4 + 35r^5 + 5r^6)$	1
1	$(1 - r)^6(6 + 36r + 82r^2 + 72r^3 + 30r^4 + 5r^5)$	3
2	$(1 - r)^5(8 + 40r + 48r^2 + 25r^3 + 5r^4)$	5
3	$(1 - r)^4(16 + 29r + 20r^2 + 5r^3)$	7

on \mathbb{R}^d for all d , have compact support, and be strictly positive definite [35]. RBF's are thus defined as compactly supported functions, which are strictly positive definite and radial on \mathbb{R}^d (for some maximal d value) [35]. Wu's compactly supported RBFs is one set of compactly supported RBFs that are positive definite and radial on \mathbb{R}^d , for some d . Examples of Wu's compactly supported RBFs are given in Table 3.3. Wu's functions, $\psi_{k,l}$, are radial on \mathbb{R}^d and strictly positive definite polynomials, of degree $4l - 2k + 1$, on their support [35].

RBF interpolation is used in many different applications to fit or interpolate scattered data. Global or local support functions can be used depending on the problem and the size of the data set that is considered. There are different RBFs that can be used for both local and global support. With local support functions, it should be noted that the functions are radial on \mathbb{R}^d , and strictly positive definite only for a maximum d value. The different functions can be used in different applications in order to find the most accurate interpolation function for a particular problem.

3.2 Sampling techniques

When determining an equation to estimate the $p\text{CO}_2$ of the ocean from *in situ* data of the Southern Ocean, it is impractical and not feasible to use all the available data. Using the entire *in situ* data set would result in the processing of large amounts of data, increasing the run time and cost of the methods. It would be more feasible to use only a subset of the data to determine the equation that relates the oceanic properties to $p\text{CO}_2$. Part of the current research is to find a method that can be used to select a subset of points from the data set, such that the equation determined from this subset has minimal error in predicting the $p\text{CO}_2$ for the entire data set. This issue is explained in more detail in this section.

3.2.1 Random Sampling

It is possible to make a random selection of points from the data set. This is an unpredictable and unreliable method for sampling a set of points. It is inconsistent in the sense that it could sometimes yield a very good approximation of the CO_2 in the ocean and, on the other hand, with another random selection of points, it could yield a very poor approximation of the CO_2 in the ocean. An alternative, more structured sampling method, that could possibly yield more consistent predictions of the $p\text{CO}_2$, is therefore considered.

3.2.2 D-optimal Sampling

D-optimality, or D-optimal sampling, is a numerically robust method of sampling a set of points in a search space [46]. It is a method that can accurately sample data, irrespective of the scaling or the choice of units of the parameters [46]. Optimal sampling methods can be used in a wide range of different applications. It is of great interest in many applications, as it can improve precision and accuracy, and at the same time reduce the number of samples required in experimental design [50]. The sampling in experimental design can be approached mathematically [70]. The main aim of the sampling method is to choose the data points in such a manner that the optimal amount of information can be obtained from the measurements made [70].

D-optimal design is one of the most favoured cost functions used in experimental design — especially in non-linear design, where the D-optimal criterion is virtually the only criterion used [70]. The advantages of using D-optimality in various applications include:

- The scale invariance of the method, *i.e.* the results are independent of the scale or the units of the parameters.
- The ability to interpret the results geometrically when the D-optimal criterion is used.
- The method allows for the correlation between parameters.

The main aim in experimental design is to use a sampling method that will select the data points in such a way that the model parameters can be approximated with as little an error as possible. A low uncertainty in the parameter predictions is thus required [70]. The Fisher information matrix statistically describes the information about the experiment, and the inverse of this matrix presents information about the uncertainty of the approximation of the parameters. This matrix is therefore used in the D-optimality criterion since, by using it, the uncertainty in the parameter estimates can be minimized [70]. The Fisher information matrix, \mathbf{M} , can be written as:

$$\mathbf{M} = \mathbf{S}^T \mathbf{P} \mathbf{S}^{-1} . \quad (3.5)$$

Here, the matrix \mathbf{S} is the sensitivity matrix of the system $\mathbf{M}_{b,b}$ as described in Equation (3.3), and \mathbf{P} is the variance-covariance matrix of the error made during the measurement of the parameters.

The Cramer-Rao inequality is given by:

$$\text{cov}[\mathbf{p}] \geq \mathbf{M}^{-1} . \quad (3.6)$$

From this inequality, the inverse of the Fisher information matrix \mathbf{M} ($k \times k$) is the lower bound of the covariance matrix, $\text{cov}[\mathbf{p}]$, of the parameter estimates $\mathbf{p} = [p_1, p_2, \dots, p_k]$. This inequality is the basis of the D-optimal sampling method. The parameter covariance matrix, $\text{cov}[\mathbf{p}]$, is influenced by experimental factors such as the type of input function used and the methods used to collect the data [37]. The determinant of the covariance matrix is proportional to the volume of the parameter confidence region [37]. Thus, minimizing the determinant of the covariance matrix will result in less uncertainty and, similarly, the maximization of the determinant of the Fisher information matrix will ensure more accurate parameter estimation [37]. The Cramer-Rao inequality proves the suitability of the Fisher information matrix as a suitable criterion in D-optimality [47]. The Cramer-Rao inequality is powerful due to the ability of the Fisher information matrix \mathbf{M} to be expressed in terms of experimental design variables,

$$\mathbf{M} = \mathbf{M}(u, k, SS, e) , \quad (3.7)$$

where u refers to the input signal, k is the number of samples, SS refers to the specific sampling schedule used, and e refers to the measurement error [46]. The optimization can be carried out with respect to any of these variables depending on the specified problem [46].

In order to create a model with good accuracy and a small error (for the parameter estimates), the k optimal positions where measurements should be taken should be determined [47]. By using a numerical algorithm to search through the sample space of possible positions for measurements, and supplying an appropriate objective function, the optimal selection of positions can be obtained [47]. For the smallest error in parameter estimates, the subset of points from the complete data set should not be sampled by random selection, but by structured selection methods [47]. The Fisher information matrix gives an estimate of the uncertainties for the parameter estimates. D-optimality can be used to minimize the uncertainties of the parameter estimates. From the Cramer-Rao inequality, the left hand side should be minimized in order to minimize the uncertainty. This is equivalent to minimizing the inverse of the Fisher information matrix [47]. The Fisher information matrix itself can not be optimized, thus a scalar function of the matrix must be optimized [70]. Therefore, the D-optimality criterion aims to minimize the determinant of the variance-covariance Fisher information matrix, \mathbf{M}^{-1} , ($\min(\det(\mathbf{M}^{-1}))$), or similarly to maximize the determinant of the Fisher information matrix, \mathbf{M} , ($\max(\det(\mathbf{M}))$) [70]. These optimization methods, that are based on the Cramer-Rao inequality and on the use of the Fisher information matrix, yield methods that optimally sample the minimum possible number of points in order to estimate parameters accurately [47].

There are various optimality criteria that have been proven to be suitable for optimal sampling techniques [47]. Some of the most popular definitions for the objective function used for optimal sampling techniques include D-optimality (Equation (3.8)), A-optimality (Equation (3.9)) and E-optimality (Equation (3.10)) [47]. D-optimality is the most popular of these three because of its geometrical interpretation. The D-optimal design minimizes the volume of the covariance ellipsoids in the d -dimensional parameter space [47].

$$OF = \min[\det(\mathbf{M}^{-1})] \quad (3.8)$$

$$OF = \min[d^{-1}\text{trace}(\mathbf{M}^{-1})] \quad (3.9)$$

$$OF = \min [\max(\lambda_{M^{-1}})] , \text{ where } \lambda_{M^{-1}} \text{ are the eigenvalues of } \mathbf{M}^{-1} \quad (3.10)$$

3.3 Genetic algorithms

3.3.1 Background

Genetic algorithms are a set of computational search techniques, based on evolutionary processes [78], that approximate solutions to optimization problems [44].

The basic idea underpinning genetic algorithms originates from the work of John Holland, who first published on this topic in the 1970's [72]. Holland suggested that if any genetic pool of possible solutions for a particular population exists, it contains the solution or, at least, an improved solution to a given optimization problem [72]. He concluded that appropriate search techniques and, in particular, evolutionary techniques can be used to find the optimal (or improved) solutions to a given optimization problem [72]. Operations such as crossover and mutation are used to find the optimal solution [72] of the profit (or cost) function that needs to be optimized [78].

Darwin's principle of "survival of the fittest", where, as in nature, the "stronger" or "fitter" solutions are more likely to attract a mate and reproduce to hopefully create offspring that are just as fit or fitter, is the basis of genetic algorithms [72]. This imitation of living beings is used to design a powerful technique for solving complex optimization problems [40].

Genetic algorithms consists of the following main parts [44], [40], [72]:

- A genetic representation of the possible solutions to a given problem.
- An initial population which is created from the set of possible solutions to the optimization problem.
- A selection of the best solutions in the population based on a fitness or objective function.
- Recombination of the selected solutions by means of crossover.
- Random alteration of the solutions obtained by means of mutation.
- A repetition of the evolutionary process until the termination criterion is reached.

In general, a genetic algorithm refers to any population of solutions that applies selection and recombination operators to the possible solutions, in order to carry out a thorough search of the search

space [78]. By adapting several possible solutions in an evolutionary way, the genetic algorithm provides an efficient way of carrying out a directed search in the search space [72].

3.3.2 Genetic algorithms as optimization method

In essence, optimization seeks to minimize or maximize functions which generally have several restrictions on the variables under consideration [40]. In the process of optimization, a set of solutions (or values) that optimize a specific objective (or cost) function is required [19]. Traditional methods often give rise to problems when the optimal objective function needs to be determined using some initial guess. An incorrect initial value can cause the method to yield a local optimum instead of a global optimal value [19]. Since genetic algorithms work with a population of possible solutions, there is no need to make an initial guess that can affect the efficiency of the algorithm. This adds to the robustness of genetic algorithms [19] [72]. Genetic algorithms prove to be simple optimization methods, with minimal analytical calculations and flexibility of the algorithms, which can be applied using parallel processing. Genetic algorithms have been used successfully in many applications in various fields [40].

Genetic algorithms are often used for non-linear optimization problems in which the parameters can not be treated as independent variables. This type of scenario occurs regularly in natural phenomena [78]. Another application of genetic algorithms is the optimization of systems that are constantly changing [73]. Such situations arise when equilibrium assumptions lead to inaccurate results [73]. Furthermore, in such cases, the parameters tend to vary with time.

Genetic algorithms have a standard set of operators that are included to make the algorithm work. Of these operators only two are generally problem dependent, viz. the encoding of the possible solutions to the problem, and the fitness function which evaluates how well the solution solves the problem [78].

It should be noted that since genetic algorithms make use of randomness, the genetic algorithm is not the most efficient method to use if there exists a traditional optimization method by means of which the particular problem can be solved [78]. In such a case the genetic algorithm tends to be slower, since it does not take additional information like the gradient of the function into account [72]. Some researchers, however, use a genetic algorithm hybridized with other traditional methods to yield optimal results [78]. Cannelas *et al.* [23] also suggests that other numerical models could be combined with genetic algorithms in order to improve the optimal solution. Ideally, the random search of a genetic algorithm should be integrated with the local search of traditional methods, to

have a good balance between exploring the whole search space and directing the search towards optimal solutions [40].

A genetic algorithm is a stochastic algorithm in which randomness plays an essential role. Furthermore, it is used to search for the optimal solution in the solution space that contains all possible solutions to a particular problem [72]. The randomness with which the search is carried out by a genetic algorithm contributes to the fact that it does not get stuck at local optima, but searches the entire search space [40]. One of the most common hybridization methods of genetic algorithms, with traditional optimization methods, works by applying local optimization to newly produced offspring to move it to a local optimum value before returning it back to the population [40]. This hybridized method performs better than either one of the optimization methods operating in isolation.

A genetic algorithm searches for an “acceptably good” solution and does not necessarily yield the best possible solution [72]. This is because of the random search methods, which do not guarantee that improved offspring are created from reproduction [40]. A genetic algorithm can run for years without yielding a better solution than the solution produced in the first iteration [72]. This means that other methods, if it is possible to apply them, may yield better solutions in shorter time. However, genetic algorithms are worth using, if no other methods prove to work for the particular problem [72].

There are no specific requirements in order for genetic algorithms to be applied to a particular problem. The steps of the genetic algorithm (which follow below) can thus be applied to any optimization problem [72]. When using a genetic algorithm, each possible solution to the problem is represented using an appropriate encoding which is called a chromosome. The set of all possible solutions to the problem is called the solution space (or the search space). A generation in a genetic algorithm refers to one iteration of the genetic algorithm. The particular set of selected possible solutions, which is considered for a specific generation of a genetic algorithm, is called the population.

3.3.3 Steps in a genetic algorithm

The general steps in a genetic algorithm are as follows:

1. **Encoding**

Before a genetic algorithm can be applied to a problem, the problem needs to be encoded into a form that enables it to be processed by a genetic algorithm [40]. Traditionally this encoding is done (as in Holland’s work) by representing the problem (*i.e.* the possible solutions to the problem) in a binary encoding [40]. In many cases, however, the representation of the problem

in a binary encoding is either much more complex to deal with than real number encoding, or close to impossible to represent in binary encoding [40]. In such cases, other encodings can be used, and the genetic algorithm can then just be altered to operate with other encodings [40]. Encoding methods include [40]:

- Binary encoding.
- Real number encoding (best used for optimization problems using an optimization function).
- Integer or literal permutation encoding (best used in optimization problems which use a combination of solutions).
- General data structure encoding (generally when complex real-world problems are considered).

When using the original solution as the problem encoding, it is not necessarily advisable to use the whole solution for the encoding, as unnecessary data could be included and this could slow the algorithm down [40]. The operators below are explained from the viewpoint that a binary encoding is used.

2. Initialization

The initial population is randomly generated in the encoding that is decided upon in the first step. This is done by randomly generating possible solutions to the problem that exist in the solution space. This genetic code, that represents the individual, is called the “chromosome” of the individual [49]. These initial “solutions” are called the initial “parents” in the genetic algorithm, as from them the reproduction will take place to produce offspring. After this step is carried out, steps 3 to 6 are repeated until the termination criterion is reached.

3. Evaluation

The evaluation function is problem specific. It is a function that determines how well a solution solves a specific problem [78]. This function is usually some kind of optimization function, and should not take too long to process [78]. It is said that the evaluation function determines the fitness of the individual, such that a solution that solves the problem “better” is said to be a “fitter” solution [72]. The idea is that the fittest individuals should have a greater chance to reproduce, and possibly yield “just as fit” or “fitter” offspring [49].

4. Selection

The aim of selection is to select individuals for reproduction in such a manner that the offspring are likely to be fitter than the parents themselves [72]. In general, fitter individuals should be

given a higher chance to reproduce. Thereafter, the reproductive parents are chosen randomly from the set of chromosomes [72]. In general, an individual can be selected more than once to be a parent in one generation [7]. The selection step steers the algorithm in a direction that is likely to be the optimal region in the search space [40].

There are several selection techniques that can be used in a genetic algorithm, including:

- **Roulette wheel selection:** The selection method where a chromosome is selected from a pool of individuals. With this method, the probability for an individual to be selected from the population is proportional to the fitness of the individual [72]. A roulette wheel is set up with each sector representing an individual and the size of the sectors proportional to the probability of an individual being chosen, as shown in Figure 3.1. A linear walk is done through the roulette wheel, and chromosomes are randomly selected from this roulette wheel [72]. In this manner, individuals with a higher fitness have a higher probability of being selected without forcing the fitter individuals to be exclusively selected. This is a moderately strong selection technique [72].

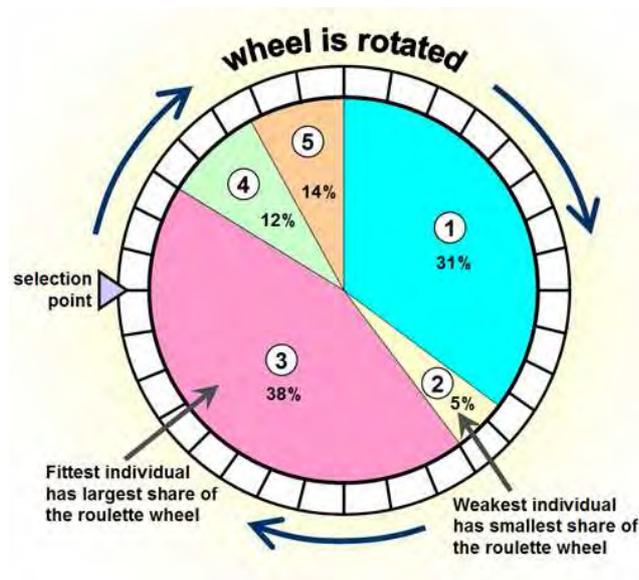


Figure 3.1: An example illustrating Roulette wheel selection [61].

- **Tournament selection:** The individuals compete against each other in a tournament and the “winners” are selected for reproduction [72]. Tournament selection allows for selective pressure, unlike roulette wheel selection, by allowing a number of individuals to compete against each other. Only the individual with the best fitness value is used as a parent for reproduction. The tournament competition is repeated until the required number of parents are selected for crossover. The new mating pool created has a higher fitness overall

than the mating pool from which the individuals are selected, and this creates the selection pressure [72]. This selection method is more efficient than roulette wheel selection [72].

- Random selection: The reproductive parents are selected in an entirely random manner [72].
- Rank selection: The population is ranked and each chromosome receives a fitness value from the ranking [72].

5. Crossover

Crossover is the process where two chromosomes (or parents) are used to create a child (or offspring) [72]. Crossover is an operator that is applied to the pool of parents in order to produce children that are likely to be “fitter” than the parents [72]. This enables the algorithm to withdraw the strong individuals from the gene pool and reproduce from them, in order to potentially produce stronger children [7].

There are several different types of crossover that can be applied to the pool of chromosomes, including:

- Single point crossover: A random point is selected on the chromosome and the two mating chromosomes exchange the part of the chromosome behind this point [72]. This is illustrated in Figure 3.2.

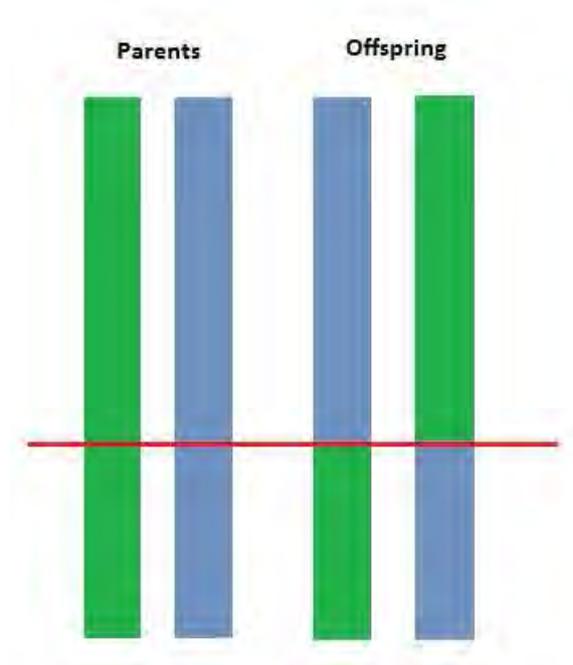


Figure 3.2: An example illustrating single point crossover.

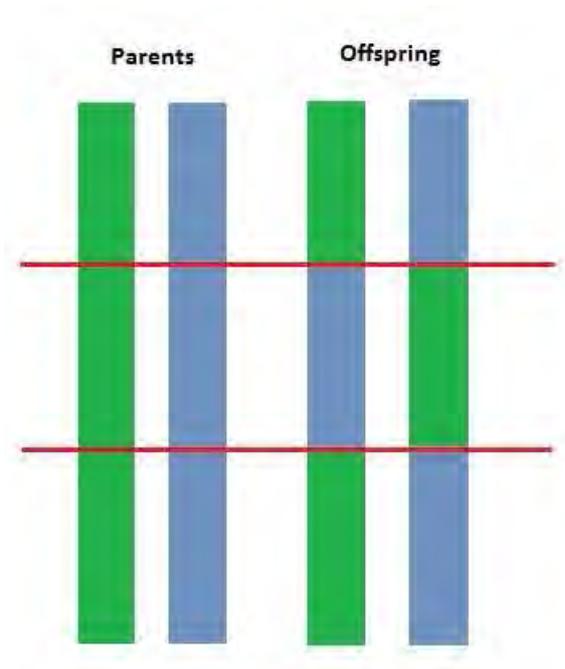


Figure 3.3: An example illustrating two point crossover.

- Two point crossover: Two random points on the chromosome are selected and the part of the chromosome between these two points are exchanged between the two mating chromosomes [72]. This is illustrated in Figure 3.3. This can cause building-blocks in the chromosome to be disrupted, decreasing the effectiveness of the genetic algorithm. However, it can also add to the diversity of the algorithm in the sense that more crossover points can contribute to a more thorough search of the solution space [72]. It depends on the type of problem considered, in some cases the one point crossover proves to be more effective, whereas in other cases the two point crossover turns out to be more effective [72].
- Three parent crossover: Three parents are selected for reproduction. Each bit of two of the parents are compared with one another, if the bit values are the same, the child receives the bit value in that bit position, otherwise, the child receives the bit value from the third parent [72].

The crossover fraction (or crossover probability) represents the proportion of the chromosomes in a generation that undergo crossover [7]. If this crossover probability is equal to one, all the parents in a generation undergo crossover. If the crossover probability is zero, none of the chromosomes in a particular generation undergo crossover. None of these probabilities are most efficient when implementing the genetic algorithm [7]; a value of about 0.6 to 0.8 is usually

chosen for the crossover probability. To obtain optimal results from the genetic algorithm, different alternatives have to be tried and tested [7].

6. Mutation

One of the most important factors that affects the performance of the genetic algorithm is the diversity in the set of chromosomes [7]. The mutation operator is one operator that increases the diversity of the population and makes sure that several parts of the search space is searched, in order to explore as much of the search space as possible [72]. The contribution of the mutation operator increases the chance of the algorithm generating chromosomes with higher fitness values [7], since it steers the search away from local optimum values [72].

Several types of mutation can be applied in a genetic algorithm implementation, including:

- Flipping: A random bit/bits are chosen and if the bit value is one, it is changed to zero, and if the bit value is zero, it is changed to one [72].
- Interchanging: Two random positions in the chromosome are selected and the bit values at these positions are interchanged between the two positions [72].
- Reversing: A random position is chosen, and the bits on either sides of this point are exchanged with one another [72].

The mutation probability is the probability that a chromosome in a particular generation will undergo mutation [72]. This value should not be too low, otherwise the diversity in the algorithm will be too low. On the other hand, making this value too high will cause the algorithm to carry out more of a random search than a structured search [72].

7. Replacement

During this cycle — from selection to mutation — two parents are selected from the gene pool and two children are produced from the reproduction; however, not all four of these chromosomes can be returned to the population — two of them must be replaced [72].

There are several ways to replace two chromosomes, including:

- Generational replacement: The new offspring automatically replaces the parents in the current generation. In this way, a chromosome can only reproduce from a chromosome in the same generation [72].
- Steady state replacement: The new chromosomes are returned to the population as soon as they are produced. In this way, the individuals of different generations can reproduce with one another [72].

- **Random replacement:** The newly created chromosomes replace two randomly selected chromosomes from the population of the current generation [72].
- **Weak parent replacement:** If the newly produced chromosome is “fitter” than the parent chromosome, then the parent is replaced by the child. This means that the two fittest chromosomes of the four chromosomes (two parents and two children) are returned to the population [72].
- **Both parents:** The child replaces the parent, and every chromosome only gets the opportunity to reproduce once [72].

8. Termination

The termination criteria are the criteria that determine when the algorithm terminates [72]. These criteria can include, amongst others, the following [72]:

- **Maximum generations:** A maximum number of generations that the algorithm can reproduce is specified.
- **Elapsed time:** A maximum running time for the algorithm is specified.
- **Stall generations:** The algorithm is terminated when there is no improvement of the chromosomes for a specified number of consecutive generations.
- **Stall time limit:** The algorithm is terminated when there is no more improvement in the chromosomes in a certain time frame.
- **No change in fitness:** The algorithm is terminated whenever the fitness does not change over a specified number of consecutive generations.

3.3.4 Motivation for using genetic algorithms

Genetic algorithms are a set of optimization algorithms that have proved to be very powerful in various applications. The advantages of using genetic algorithms are briefly discussed in this section.

The advantages of evolutionary computation — and specifically genetic algorithms include:

1. **Conceptual simplicity:** The derivatives of a function with respect to the parameters considered are not required for the algorithm, as in other traditional optimization methods [72] [49].
2. **Broad applicability:** Any problem that can be formulated as a function of optimization problems can be solved using genetic algorithms [72]. Problems that have discontinuous functions

and functions that can not be analytically determined can be solved using genetic algorithms [49].

3. **Hybridization with other methods:** Genetic algorithms have a lot of potential to be an effective optimization tool because it can be hybridized with other methods, including traditional optimization methods, to develop an optimal method for any particular problem considered [28], [72].
4. **Parallelism:** The evaluation of each solution by means of a fitness function can be dealt with in parallel — only the selection requires some sequential operation. Hence the run time of the algorithm is indirectly proportional to the number of processors available [72].
5. **Robust to dynamic changes:** Unlike traditional methods, the algorithm does not have to be restarted when the environment changes. Changes can be made to the method as the algorithm is running and the algorithm will simply adapt [72].
6. **Solves problems with no known solutions:** Without having an idea of what the possible solutions to a problem may be, and whether solutions exist, a genetic algorithm can be applied to find an optimal solution for the problem [72].
7. **Robust to non-linearity:** Complexity and non-linearity between the model and its parameters can be dealt with by a genetic algorithm [49].

Genetic algorithms can also be used to create the input for another algorithm and vice versa [28]. The design of other algorithms (or methods) can be supported by genetic algorithms [28]. Genetic algorithms prove to be more efficient than other machine learning techniques, like the Kalman Filter, when the function considered includes some discontinuities or other complexities [49]. Furthermore, genetic algorithms can be used to solve parameter estimation problems [49].

Another attribute of genetic algorithms that contributes to their robustness and success, in various applications, is that they can be modified to find better solutions and adapted to particular problem specifications [40]. The algorithm can be modified in two ways. The first way is to adapt the components of the genetic algorithm, including the encoding, the crossover and mutation operators, and the selection, in order to see which combination of operators works best for the particular problem [40]. Secondly, the evolutionary process can be adapted. This is a way to change the parameters, e.g. the mutation probability, or the crossover probability to solve the problem at hand [40]. Thus, by modifying the genetic algorithm, better results can be obtained (if the correct combination of modifications is used). These types of changes cannot be determined beforehand — they must be

tried and tested with each particular problem. Choosing the best set of parameters is of great importance in order to yield optimal results from genetic algorithms [49].

3.4 Conclusion

Various curve fitting and interpolation methods will be used in this research to investigate the empirical relationship between $p\text{CO}_2$ and other ocean variables. Least squares curve fitting and radial basis function interpolation are ways of determining a relationship between $p\text{CO}_2$ and other ocean variables. Furthermore, the empirical relationship will not be established using the entire data set. Rather, a set of points will be selected from the whole data set from which the relationship will be established. This can be done by means of D-optimal sampling, as this is shown to minimize the variance in the parameter estimates. Furthermore, genetic algorithms will be used to carry out the D-optimal sampling.

One of the challenges that is faced, is the processing of the data: to obtain a form of the data that can be worked with. The fact that the relationship between the ocean variables and the effect they have on one another is unknown creates challenges for the current research. Furthermore, the variables that should be used to establish this empirical relationship have to be considered carefully, in order to ensure that the most accurate empirical relationship is determined. Different fits will be done on the data in order to investigate whether a relationship between the $p\text{CO}_2$ and other ocean variables can be established.

An improved approach for estimating the $p\text{CO}_2$ in the Southern Ocean will contribute, not only to the country as a leading research location for the carbon-ocean studies, but also to global carbon investigations and climate studies. The results of this project will add to the research done on the effects of climate change and on ocean-atmosphere dynamics in the Southern Ocean. This project is part of a bigger project on the Southern Ocean and will contribute to the understanding of ocean-atmosphere dynamics.