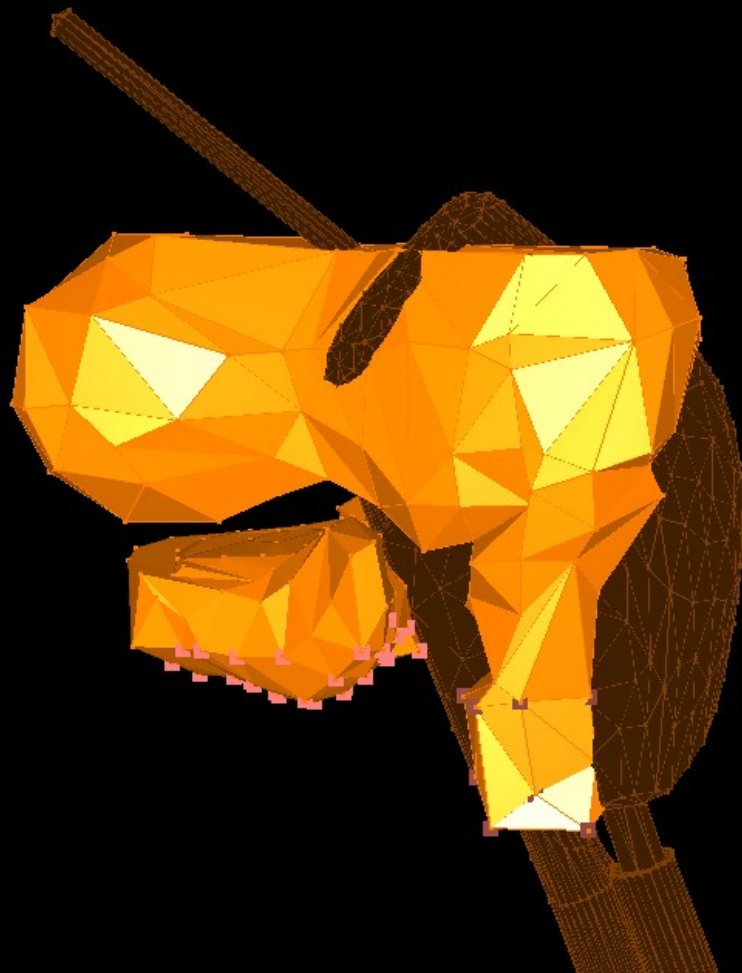


# A surrogate-assisted evolutionary algorithm based on inverse distance weighting

Applied to a multi-objective deformable image registration problem

Iwan Hoogenboom





# A surrogate-assisted evolutionary algorithm based on inverse distance weighting

## Applied to a multi-objective deformable image registration problem

by

Iwan Hoogenboom

to obtain the degree of Master of Science  
at the Delft University of Technology,  
to be defended publicly on Friday October 14, 2022 at 10:00 AM.

Student number:	4396634
Project duration:	September 16th, 2021 – Oktober 14th, 2022
Thesis committee:	Prof. dr. Peter A. N. Bosman    CWI, TU Delft, supervisor
	Dr. Tanja Alderliesten        LUMC, supervisor
	Dr. Frans A. Oliehoek        TU Delft
	Anton Bouter, MSc.            CWI
	Cedric J. Rodriguez, MSc.    LUMC

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



# Preface

In a sense, your life is like a expensive multi-objective optimization problem. You have many goals you want to achieve and only a limited amount of time. This thesis was one of my objectives of this last year. It marks the end of my (a little bit more than) eight years as student at the TU Delft. During this last year, I have explored the world of expensive multi-objective optimization with Evolutionary Algorithms. The goal being, designing an algorithm using the simple interpolation method of inverse distance weighting to assist in optimizing the complicated problem of beating cancer, in this particular case, cervical cancer.

I would like to thank Prof. dr. Peter Bosman, Dr. Tanja Alderliesten, MSc Anton Bouter and MSc Cedric Rodriguez for their guidance and support throughout the whole project. From helping me decide which ideas were worth investigating to explaining how the LUMC cluster worked. I would not have been able to do this project without their assistance.

*Iwan Hoogenboom  
Delft, Oktober 2022*



# Abstract

Solutions to many real-life optimization problems take a long time to evaluate. This limits the number of solutions we can evaluate. When optimizing with an Evolutionary Algorithm (EA) a frequently used approach is to approximate the objective using a surrogate function, replacing the time-consuming real evaluation. This surrogate model is combined with a so-called acquisition function, to select promising candidate solutions. The acquisition function balances the trade-off between exploration of parameter space and the exploitation of the surrogate. These candidates are subject to an expensive evaluation with the true objective function and are used to update the surrogate model. Iteratively applying this process can effectively optimize global optimization problems. In this work, we propose a new multi-objective optimization algorithm with inverse distance weighting as surrogate function, which we call IDW-SAEA (inverse distance weighting surrogate assisted evolutionary algorithm). We introduce a new objective to the optimization problem to improve exploration and reduce the complexity of the acquisition function. We show this algorithm is competitive with state-of-the-art kriging-based surrogate-assisted EAs on certain benchmark problems. Additionally, we use the algorithm to optimize a practical problem: a Finite Element Method simulation of the cervix region with applications in radiotherapy for cervical cancer.





# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Expensive optimization . . . . .	1
1.2	Virtual Phantom optimization for brachytherapy for cervical cancer . . . . .	3
<b>2</b>	<b>Background</b>	<b>5</b>
2.1	Optimization with Evolutionary Algorithms . . . . .	5
2.1.1	Optimization . . . . .	5
2.1.2	Basics . . . . .	5
2.1.3	Real-valued optimization . . . . .	6
2.1.4	Multi-objective optimization . . . . .	7
2.2	Expensive Optimization . . . . .	9
2.2.1	Basics surrogate assisted . . . . .	9
2.2.2	Inverse distance weighting . . . . .	9
2.2.3	Kriging. . . . .	10
2.2.4	Other surrogate types . . . . .	10
2.3	State-of-the-Art Evolutionary Algorithms . . . . .	10
2.3.1	MOAE/D-EGO . . . . .	11
2.3.2	K-RVEA . . . . .	11
2.3.3	AB-SAEA . . . . .	11
2.3.4	MAMaLgAM . . . . .	12
2.4	Virtual Phantom Simulation details . . . . .	12
<b>3</b>	<b>Design of Optimization Algorithm: IDW-SAEA</b>	<b>15</b>
3.1	Algorithm overview . . . . .	15
3.2	Optimizing the surrogate model . . . . .	15
3.3	Selection of candidate solutions . . . . .	19
3.4	Additional challenges and improvement of optimization algorithm . . . . .	21
<b>4</b>	<b>Experiments</b>	<b>23</b>
4.1	Setup . . . . .	23
4.1.1	State of the Art algorithm settings . . . . .	23
4.1.2	Benchmark optimization problems. . . . .	23
4.2	MAMaLgAM parameter tuning. . . . .	27
4.2.1	MAMaLgAM settings . . . . .	27
4.2.2	Results MAMaLgAM tuning . . . . .	27
4.3	IDW-SAEA parameter tuning . . . . .	28
4.3.1	Results settings IDW-SAEA . . . . .	29
4.4	Comparison state of the art on benchmarks . . . . .	34
4.5	Virtual phantom simulation . . . . .	36
4.5.1	Comparison of IDW-SAEA settings . . . . .	36
4.5.2	Comparison between IDW-SAEA, MAMaLgAM and K-RVEA . . . . .	36
<b>5</b>	<b>Discussion</b>	<b>39</b>
5.1	Performance Analysis . . . . .	39
5.2	Potential Improvements . . . . .	41
5.2.1	Exploration objectives as tool . . . . .	41
5.2.2	Inverse distance weighted regression . . . . .	42
5.2.3	Future work in Virtual Phantom optimization . . . . .	42
5.3	Limitations . . . . .	42

<b>6 Conclusion</b>	<b>45</b>
<b>A Appendix</b>	<b>51</b>

# Introduction

Evolutionary algorithms (EAs) are a specific type of artificial intelligence designed to find solutions for challenging optimization problems. These algorithms are inspired by Darwin's evolution theory. They exploit the information and interaction between the set of potential solutions (population) to gain insights into the optimization landscape. By effectively selecting and combining solutions, the overall population 'evolves' in the direction of the optimal solution to the problem. Compared to exact optimization methods, EAs are especially useful in problems where finding the optimum is infeasible to compute (NP-hard problems) or when the objective function, this is the function that determines the quality of a solution, is too complex because it contains non-linearities, discontinuities or definition gaps [29].

The second strength of EAs is in multi-objective (MO) optimization, where multiple objectives have to be optimized simultaneously. The different objectives always contain a trade-off between the objectives. In these problems, there does not exist one single solution which could acquire the best value for all objectives. Instead, there exists a set of solutions which represent the trade-off between the objectives. In MO optimization it is desired to get this set of solutions as complete as possible, thus modelling the whole trade-off. Single solution optimization algorithms, such as Gradient Descent, have to be run multiple times to create such a front, while EAs already have multiple solutions by using the population and therefore are great in MO optimization.

## 1.1. Expensive optimization

Although EAs are well suited for multi-objective problems, state-of-the-art EAs typically require the evaluation of many potential solutions. In many real-life optimization problems, the evaluation of potential solutions is computationally expensive [53, 50, 52]. A problem is considered expensive when its solutions are costly to evaluate, with an experiment using a large amount of time or many resources (e.g. money or valuable materials). For example, large aerodynamic simulations can take hours to run [33] and physical experiments can involve large financial costs. In standard EAs, the amount of evaluations used can easily go up to 10.000 and is often higher. When the problem is expensive, the evaluation of potential solutions becomes such a large factor in the runtime of the optimization process that generally a maximum of 3000 evaluations is used, often less [15]. Therefore when designing optimization algorithms for expensive to evaluate problems, reducing the required number of evaluations, could substantially reduce the total optimization time.

Currently, a multitude of algorithms has been developed for expensive multi-objective optimization with EAs [3, 15, 19]. This is often done by utilizing a quick-to-evaluate approximation, often function approximations, which replace the expensive evaluations to reduce the number of evaluations or reduce the time of each evaluation. Three different types of approximations are defined by Jin [30]: function approximation, problem approximation, and fitness approximation.

The first of these is a frequently used approach [15]. This approximation function, which is fast to evaluate and based on a set amount of true function evaluations (FE), replaces the true function. An optimization algorithm, (typically EA) optimizes the surrogate and selects the potential candidate solutions. The surrogate is usually updated in an iterative process: during optimization, the newly evaluated solutions are combined with the previously evaluated solutions to update the surrogate model.

Problem approximation is reducing the complexity of the problem itself by replacing it with a similar but simpler problem that is easier to solve, therefore reducing the cost of evaluating a solution. Running computational fluid dynamic simulations instead of wind tunnel experiments is an example of problem approximation.

Fitness, the metric to measure a solution quality, approximation estimates the fitness of a solution directly. The difference between this and function approximation is the method of determining the final fitness. In function approximation, the performance of a solution is based on the approximated values of the objective function. Fitness approximation does not approximate the objective function instead it directly tries to predict the fitness of a solution. An example of this is CSEA [42], where solutions are classified into two categories. The categories are then used in the selection of new solutions to evaluate.

An approximation function is also called a surrogate and an EA using a surrogate is often referred to as a surrogate-assisted EA (SAEA). The most common method of function approximation used is kriging [15] (also called Gaussian Process). This method is originally from geostatistics [17] and is popularized by the expensive optimization algorithm: EGO [34]. This method is usually preferable to other interpolation methods because of its uncertainty estimation. When selecting new potential solutions to evaluate, uncertainty estimation can be used to calculate a probability of improvement. A drawback of the kriging method is its time complexity of  $O(n^3)$  [23]. Even more complex methods for approximation exist, for example, using Deep Learning [55] or even using multiple surrogates together [24].

Although the performance of these complex approaches is promising, it is interesting to investigate if a simpler surrogate model could be used. A more computationally efficient ( $O(n)$ ) surrogate model is inverse distance weighting (IDW) interpolation [48]. IDW performs a weighted average of the objective values of its neighbours based on their distance to the interpolation point. To the best of my knowledge, there is no published research on the performance of IDW as an approximation function for multi-objective optimization. This makes it interesting to investigate. Kriging interpolation is a frequently used approach in function approximation, therefore it is a good method to compare against. This results in the following research questions of this thesis:

Is it feasible to design a multi-objective evolutionary algorithm using an inverse distance weighting surrogate function that is competitive with the state-of-the-art kriging-based surrogate-assisted evolutionary algorithms?

IDW does have limitations, the main one being the inability to extrapolate correctly as can be seen in Figure 1.1. We can see the 1-dimensional interpolation of the four points closely resembles their actual trend of  $y = x/1.7$ . However, when extrapolating the value achieved from IDW regresses to the mean. When minimizing this IDW interpolation, the best value of  $x$  is the already-known value  $x = 2.1$ . Since the objective score of this point is already known, re-evaluating this location will not lead us to a better solution. This directly prompts the following sub-question:

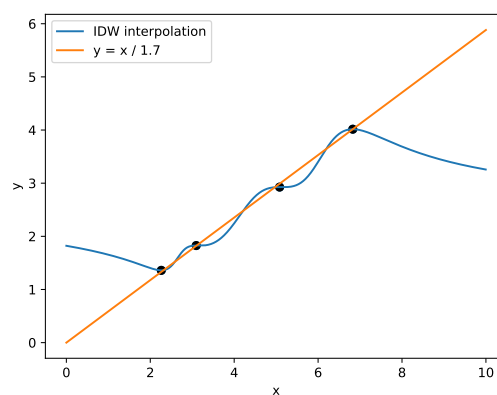


Figure 1.1: Inverse distance weighting of four points generated by the formula  $y = x/1.7$  with added noise. On the x-axis, we have the parameter to optimize and on the y-axis the objective value.

Which adaptations are required to counteract the limitations of inverse distance weighting?

An interesting sub-question is to investigate the difference in performance between the IDW surrogate-assisted EA (IDW-SAEA) and SAEAs using kriging methods. Resulting in another research question:

What are the advantages and disadvantages of the inverse distance weighting surrogate-assisted evolutionary algorithm with respect to state-of-the-art expensive kriging-based surrogate-assisted evolutionary algorithms?

## 1.2. Virtual Phantom optimization for brachytherapy for cervical cancer

Cervical cancer is the fourth most common type of cancer found in women accounting for more than half a million diagnoses and more than 300.000 deaths worldwide each year [12]. A possible course of treatment for cervical cancer consists of a two-step process. Firstly, the malignant cells are treated with external beam radiation therapy (EBRT). Thereafter, they are treated with internal radiation therapy (brachytherapy). For the planning of both treatments, a scan is made of the patient to capture the organ shapes during treatment. In the case of cervical cancer, an applicator is placed inside the vagina and uterus cavity. This applicator is then used to carefully guide the radioactive material close to the cancer tissue which kills malignant cells and stops them from growing and dividing. The time and location of the radioactive material are determined by the treatment plan which plans the amount of radiation the different tissues have to receive. Figure 1.2a illustrates the normal positioning of the vagina, cervix, uterus and bladder. Figure 1.2b shows the positions of the organs after placement of the applicator. Both the difference in bladder filling and the displacement of the applicator are creating large deformities of the surrounding organs.

To have optimal planning it is important to account for the dosage delivered by the EBRT. Currently, the medical procedure is to assume the worst-case scenario (the maximum dosage) of radiation delivered to the organs. By correctly keeping track of radiated tissue a more efficient treatment plan could be designed without increasing the risk of over-radiating the healthy organs at risk. A magnetic resonance imaging (MRI) scan is made before the placement of the applicator with a second scan once the placement is finished. Tracking the tissue deformation between scans is challenging due to large deformations induced by the applicator and vaginal padding. The padding is used to keep the applicator in place and restrict tissue movement during the radiation period. The resulting deformation can be seen on the MRI scan. To keep track of the tissue locations, deformable image registration (DIR) is required. DIR for brachytherapy is concerned with mapping two or more images onto the same coordinate system, this allows for calculating the transformations (translation, rotation, shrinking, stretching) of the organs. An additional difficulty is the content mismatch between the scan with and without the applicator. Current DIR methods are capable of modelling the deformations of the tissue to match the MRI scans with the applicator, however, these methods do not always keep into account the physical feasibility of the transformations [46].

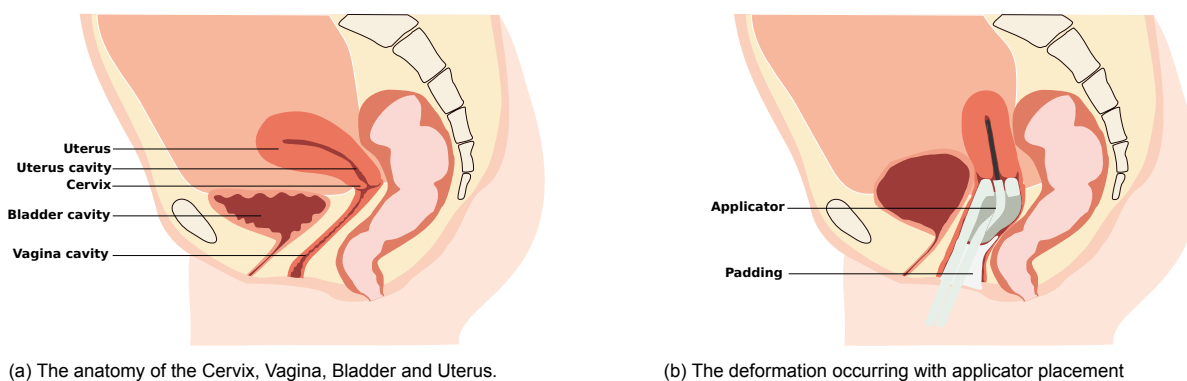


Figure 1.2: Schematic illustrations showing the cervix region before (1.2a) and after (1.2b) placement of the applicator.

A recent direction of research [46] pertains to the use of a virtual phantom of the cervix region. A virtual phantom is a possible manner for ensuring the physical feasibility of the deformations by creating

a mechanical model of the abdominal region based on MRI scans of the patient. On the model, we can run a mechanical simulation using established Finite Element Method (FEM). This mechanical simulation requires us to specify the organ forces and mechanical properties (details in section 2.4) such that the organ deformation aligns with the organ deformation seen during brachytherapy. These properties have a large variability between different patients and also are highly dependent on the placement of the applicator. Therefore these properties should be optimized on a patient-specific basis to achieve the most realistic results. The results of this simulation are compared to the MRI scan made post applicator placement to judge the correctness of the modelled deformations. The biomechanical nature of the virtual phantom allows checking for the feasibility of the transformations occurring. The ability to create matching and feasible deformations with the virtual phantom opens up the possibility to assume this deformation as ground truth. These ground truths can be used to validate the results of other DIR algorithms that do not take the biomechanical feasibility into account.

The two main organs to simulate are the bladder and the vagina-uterus. These organs are adjacent to each other in the abdominal region and therefore have a large amount of interaction. In addition, we work with simplified organ models, thus it is highly unlikely that a perfect matching exists. This results in a trade-off between finding the correct deformation of the two organs. A good result requires correct deformation of both organs, thus a bi-objective approach is most suitable. This allows modelling the trade-off between organ matching without requiring the preferred definition of the trade-off before simulation. Thus opening up the possibility for a clinical expert to later decide which matching is preferred.

Additional difficulties of this optimization problem are the runtime of a single simulation and the objective landscape. Currently, a single run of the virtual phantom requires approximately 15 to 25 minutes of computation time. This greatly impedes the number of evaluations that can be used. Secondly, the possibility of having incompatible biomechanical parameters, resulting in an unstable and crashing simulation, greatly inhibits gradient-based optimization methods. Therefore the choice of an EA seems a good fit, as they are capable of optimizing multi-objective and discontinuous landscapes. This practical problem introduces the last research question:

Can the evolutionary algorithm utilizing an inverse distance weighting surrogate optimize the Virtual Phantom Simulation more efficiently compared to state-of-the-art (surrogate-assisted) evolutionary algorithms?

The structure of this thesis is as follows. Chapter 2 contains the background information on EAs and the Virtual Phantom. Followed by a description of the designed optimization algorithm in Chapter 3. Then in Chapter 4 the experiments on both benchmark problems and the virtual phantom optimization are described. Thereafter these results and the algorithm are discussed in Chapters 5, followed by the mentioning of some limitations. Finally, a conclusion is drawn.

# 2

## Background

This section will explain the basics of evolutionary algorithms (EAs), explain what expensive optimization is, how to address expensive problems and finally a few different state-of-the-art (SOTA) EAs will be introduced. Furthermore, this section contains the details of the virtual phantom simulation.

### 2.1. Optimization with Evolutionary Algorithms

After a quick definition of optimization, the Genetic Algorithm [28], a type of EA, will be described. The following sections will explain the changes required to transfer to real-valued and multi-objective EAs.

#### 2.1.1. Optimization

In an optimization problem, the goal is to find the best, or as good as possible, solution for the problem. Optimization can be mathematically defined as:

$$\min_{x \in S} f(x)$$

where  $x$  is a solution,  $S$  is the set of all feasible solutions, and  $f$  is the optimization function. The set of feasible solutions is called the search space or decision space and the set of possible outcomes is called the objective space.

#### 2.1.2. Basics

EAs are a subset of the field of Evolutionary Computation. Evolutionary Computing algorithms are a set of optimization algorithms inspired by the principles of Darwin's theory of evolution: selection and variation. EAs use a set (*population*) of potential solutions (*individuals*) which is updated iteratively, where each iteration is called a *generation*. An individual is a list of values (*genotype*) which together form a single solution to the optimization problem. An individual will be evaluated using the *fitness function*, sometimes called *evaluation function* or *objective function*. This function assigns a (scalar) score (*fitness*) to each individual based on the quality of its genotype.

After (randomly) generating and evaluating an initial population, an EA iteratively applies selection and variation to generate a new population. New individuals are created by applying variation to solutions of the previous population (*parents*). These new individuals are called the *offspring* and inherit part of the genotype of their parents (variation). A parent with a higher fitness has a higher chance of surviving the process of selection and thus a higher chance of passing down its genotype. Repeatedly doing variation and selection will move the population in the direction of the optimal solution thus optimizing the problem.

In discrete optimization, the most common types of variation are mutation and crossover. Crossover is the exchange of genotype parts between two or more individuals. There are many different ways of exchanging information, but an easy example is one-point crossover. In one-point crossover a random point is chosen, up until the crossover point the genotype of the first parent is used and after the crossover point, the genotype of the second parent is used. Figure 2.1 shows the crossover point after the fifth digit. Taking the first five values from parent B and the last three from parent A results in a

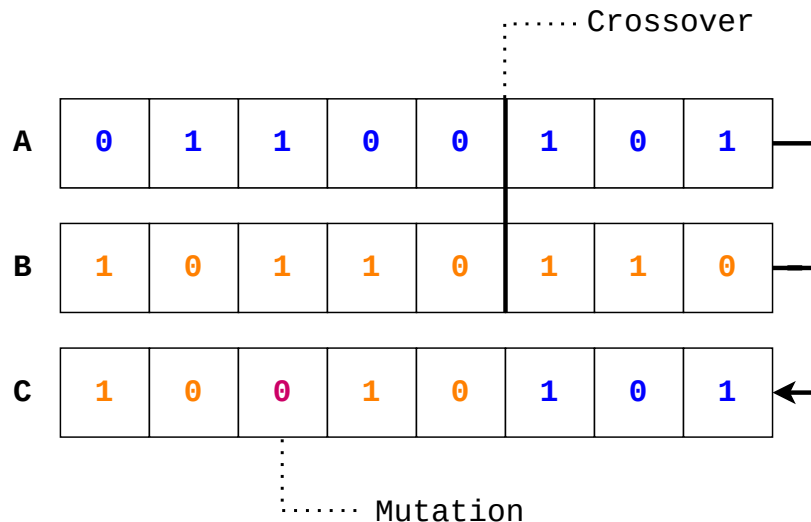


Figure 2.1: Two parents, A and B, with one-point crossover after the 5th digit. Mutation occurs on the 3rd digit to result in the child C.

new child. This offspring is a new combination of the parents' values. Mutation does not combine information of the parents but makes a random change to one individual without external information. The following binary-mutation rule is applied: randomly flip one of the values. It is now possible to reach different locations in parameter space independent of the parent solutions. Looking at Figure 2.1 again, we see the third digit mutating into a 0. This results in a new individual which was not possible to create no matter how the values of parents A and B would be exchanged.

Randomly selecting parents to generate offspring will not result in a better optimization process than random search. This is why there is a selection phase. The individuals who are selected to perform crossover and (possibly) mutate are once again inspired by a well-known biological principle: "survival of the fittest". The whole population will first be evaluated based on the objective function. Individuals with good fitness will be more likely to be selected as parents. This way the genotypes which contribute to good objective scores are more likely to exist in the offspring of the generation. By iteratively applying selection and variation the population will become more similar, which is called convergence. A population is called converged when the complete population has the same genes.

It is an important aspect of the algorithm to achieve convergence at the right moment in the optimization process. If the selection pressure is too large, premature convergence can occur. This means that all the diversity is lost before a solution of sufficiently high quality can be found, which does not allow for further exploration. It is therefore important that the convergence does not happen too early in the optimization process. Late convergence is also not preferred since it will reduce the speed of finding good solutions. If the algorithm converges very slowly there will be many weak individuals in the population which slows the optimization process by introducing non-promising offspring. Therefore a balance in convergence speed is required to be found.

Since the algorithm continues with generating new offspring every generation, and the optimal fitness value is generally not known, outside termination criteria have to be set. There are two types of termination criteria: resource-based and performance-based. Resource-based termination criteria cause the termination of the algorithm after a set amount of time or function evaluations. Performance-based termination terminates the algorithm when the desired fitness level has been reached or the population is fully converged thus no new solutions are introduced anymore.

### 2.1.3. Real-valued optimization

In real-valued optimization, the crossover becomes a lot less efficient since there are an infinite (practically limited by the computational precision) amount of values between two individuals. There exist multiple ways of incorporating real-valued individuals. It is possible to express real-valued numbers using a binary encoding, and directly optimize this using a discrete EA. This will scale badly: With four binary variables to encode a parameter within the range of  $[0, 1]$ , it is possible to divide into 8 different



steps giving a precision of  $\frac{1}{16} = 0.0625$ . Having multiple parameters to optimize the genotype of a single individual will increase the number of decision variables substantially and with that, the difficulty of finding the optimum also increases. A second even larger problem is the difference in magnitude between two numbers when using a bit representation: bit strings [0101] and [1101] only differ one bit in the encoding however the difference in real values ( $\frac{5}{16} = 0.3125$  and  $\frac{13}{16} = 0.8125$ ) is half the search space. Therefore most of the time a different algorithm is used, which is designed to handle real-valued inputs without encoding. An example of this is Evolution Strategies (ES) [44], where variation is done primarily by mutating a sample with the use of normal distributions for each variable. The most famous ES algorithm is CMA-ES [25].

Another approach is Estimation-of-Distribution Algorithms (EDAs), which is the type of algorithm used in this Thesis. The first designs of the EDAs were not real-valued but binary. This algorithm is known as the Univariate Marginal Distribution Algorithm [41]. The general idea is to create a probability distribution over all problem variables based on the population. Thus when each sample consists of a binary string of 0s and 1s, the algorithm would perform selection and then estimate the probability of 1s and 0s when sampling a random individual. These probabilities are then used to generate a new population by generating individuals based on the Bernoulli distribution of each value. Take for example a three-digit binary encoding. If after selection in the previous population 30% of the population has a value of 1 as the first digit, 40% a value 1 on the second digit, and 85% value 1 on the third digit, the next generation will be sampled using those values.

In real-valued EDAs the distribution estimated is no longer a Bernoulli distribution but a Normal distribution. From there on it is straightforward to sample the new population again based on the distribution. Iteratively performing selection, distribution estimation, and resampling is shown a promising alternative to ES [9]. This real-valued EDA is later improved into AMaLGaM [4] which was shown to be competitive with other state-of-the-art real-valued EAs.

#### 2.1.4. Multi-objective optimization

Many real world problems do not have one optimization criterion. Usually, there is a combination of different objectives to be optimized. These objectives will often conflict where the optimal value of both objectives can not occur in a single solution. An engineering example is minimizing the cost while maximizing the structural strength. Another example is taking a direct flight for a higher price, or a cheap flight with a connection in between. Both these problems create a trade-off where the balance between the objectives is challenging to define a priori. Mathematically this is defined as:

$$\min_{x \in S} \{f_1(x), f_2(x), \dots, f_m(x)\}$$

Where we have  $m$  objective functions  $f$  to minimize, using the same solution  $x$  which lies in decision space  $S$ . In multi-objective optimization, a single solution obtaining an optimal value for each of the objectives is often unachievable. Instead, the preferred result is a set of solutions that are all an efficient trade-off between the objectives. This set of optimal solutions is called the *Pareto set*. Their associated objective values are called the *Pareto front*.

In a multi-objective minimization problem, solution A is said to be *Pareto dominated* by solution B if for each objective the solution score of B is at least as good as A, and in at least one objective B scores better than A.

**Definition 1** Solution  $x_0$  is said to (Pareto) dominate solution  $x_1$  denoted as  $x_0 > x_1$ . If and only if.

$$\begin{aligned} \forall i \in \{1, \dots, m\} : f_i(x_0) \leq f_i(x_1) \wedge \\ \exists i \in \{1, \dots, m\} : f_i(x_0) < f_i(x_1) \end{aligned}$$

The goal of MO optimization is to find a set of solutions, that approximate the values of the Pareto set as good as possible. This resulting set of solutions is called the *approximation set*. The approximation set only contains non-dominated solutions and thus each solution models different options of the trade-off between the objectives. Together these solutions also create a front, the so-called *approximation front*.

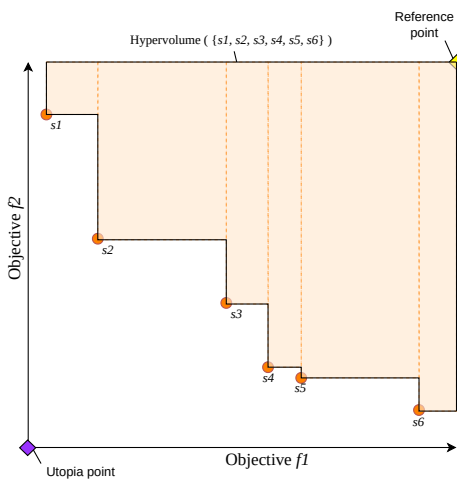
The concept of ranks of solutions within a population is an important part of the selection in this Thesis. Rank is a metric of how often a solution is dominated within a population. All non-dominated solutions in the population are given rank 0. Calculating the solutions belonging to rank  $i$  is done by

removing all solutions of rank  $i - 1$ , the solutions which are now non-dominated get assigned rank  $i$ . Repeat this until all solutions are assigned a rank.

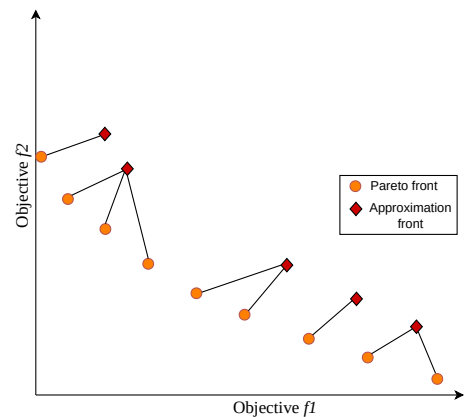
After multiple iterations of the EA, the number of rank 0 solutions can grow to the size of the population i.e. the complete population is the approximation front. When performing selection some of these non-dominated solutions have to be removed from the population to create space for new offspring. This is especially the case in the real-valued multi-objective optimization since often the objective scores are real-valued as well. Real-valued objective scores allow for an infinite number of non-dominated solutions. When solutions are discarded, the space dominated by the front changes and this makes it possible for the front to become worse. To avoid this an *elitist archive* is used. This archive stores all non-dominated solutions from the population. Every generation the new non-dominated solutions are compared to the existing ones in the archive. If any new solution is not dominated by one in the archive it is added and all solutions currently in the archive dominated by this new solution will be removed. Since the real-valued objectives could produce infinitely many solutions the elitist archive has to be managed, or else it might grow to extreme sizes. One method, designed by Luong and Bosman [38], is to set an upper bound on the maximum number of solutions. When more solutions than this bound are present in the elitist archive, the elitist archive is divided into hypercubes allowing only a single solution per hypercube. This way solutions that are close to each other in objective space will be removed.

In every multi-objective optimization problem, there are two goals to achieve: proximity and diversity. Good proximity means that the objective values of the non-dominated solutions are close to the Pareto front. A good diversity score is achieved by having a broad approximation front and having solutions close to all sections of the Pareto front. The most straightforward comparison of bi-objective optimization algorithms is plotting the approximation sets and comparing those visually. This allows us to see the strengths and weaknesses of each algorithm and sometimes directly indicates which algorithm is outperforming the other. When there are more than 3 objectives or the different algorithms dominate each other in different locations the visual inspection does not work well anymore. Another drawback is that it can be difficult to see the progression of the approximation set in a single figure.

This is where a metric like the hypervolume (HV) [36] is useful. Especially when the Pareto front is unknown, the HV is a good metric. In the HV metric, the dominated (hyper)volume in objective space is measured between the approximation set and a reference point. The reference point is a point which functions as an upper bound to the solutions of the approximation set. An example of the HV metric is seen in image 2.2a. The HV is indicated in orange. In this thesis, the HV is normalized by dividing the final result by the total possible area the HV could hypothetically span: from the reference point to the *Utopia point*. The utopia point is often not reachable, but it is the hypothetical best solution to the optimization problem. In the minimization problem illustrated in Figure 2.2a it is located in the origin.



(a) Hypervolume is the area (or (hyper)volume in more than two dimensions) between the reference point and the non-dominated solutions.



(b) For every point in the Pareto set the distance to the closest point in the approximation set is calculated, summed, and then divided by the size of the Pareto set to achieve the Inverted Generational Distance

Figure 2.2: Metrics used in this thesis: Hypervolume and Inverted Generational Distance

A second metric is the distance from Pareto front to approximation set ( $D_{P_F \rightarrow S}$ ) [10], also known

as Inverted Generational Distance (IGD) [16]. This metric requires the Pareto front to be known. It measures the distance from every solution of the Pareto front to the closest solution in the approximation set and averages these distances. Mathematically it is defined as:

$$IGD(S) = \frac{1}{|\mathcal{P}_F|} \sum_{p \in \mathcal{P}_F} \min_{s \in S} d(p, s)$$

Where  $\mathcal{P}_F$  is the Pareto front,  $p$  is a single solution of this front,  $S$  is the approximation set and  $s$  is a single solution from the approximation set. An illustration of the metric can be seen in 2.2b.

## 2.2. Expensive Optimization

Expensive optimization considers a specific type of sub-problem in the general optimization domain. A typical characteristic of expensive problems is the time or resource cost of an evaluation. When the cost of the evaluations itself is a large factor of your time and resource budget, the choice of which solutions to evaluate becomes of large importance. Therefore expensive optimization places additional importance on making every evaluation count. In EAs this is often done by using a so-called *acquisition function*. This is a function that manages which solutions to evaluate. It often aims to find a balance between promising solutions and explorative solutions, trying to quickly find the optimal regions to explore further. Multiple surveys on Expensive Optimization for EAs exist in literature [30, 49, 31, 3, 19, 15].

### 2.2.1. Basics surrogate assisted

As described in the introduction there exist multiple types of expensive optimization approximations for EAs. A common method consists of the use of the surrogate model, also called surrogate function, to approximate the original objective function. The main goal of a surrogate model is to predict the objective value in unknown areas i.e. it can be seen as inter- and extrapolation of the objective scores. EAs using these surrogates (or meta-models) are called surrogate-assisted EAs (SAEA) [31]. A simplified version of the general workflow (based on [15]) of SAEAs is as follows:

1. Initialize individuals with a sampling method and evaluate.
2. Build a surrogate, based on these individuals.
3. Optimize the EA population based on the fitness of the surrogate
4. Evaluate (a few of) the individuals in the final population
5. Update the surrogate model
6. If termination criteria are met: terminate. Otherwise, return to step 3.

Steps 2 through 5 are the steps where SAEAs differ and thus the ongoing research topics.

A crucial aspect of all SAEAs is Evolution Control (or model management) [32], which essentially boils down to the questions of when to update the surrogate model and which individuals to evaluate to update this model with. When the surrogate is updated after a predefined number of iterations, this is called Fixed Evolution Control. In adaptive evolution control, the choice of when or how to update is regulated by the algorithm. This allows for more flexibility but can be challenging to define correctly.

The choice of surrogate model and the way the EA interacts with it is critical to the success of the algorithm. There are multiple types of surrogates models. Often the surrogate function is used in the acquisition function. The acquisition function usually is a combination of expected value based on the surrogate and a measure of uncertainty. By combining these two, a new metric is designed to select appropriate candidates to evaluate with the expensive real function. Selecting the sample with the largest Expected Improvement [34] is one of these possible acquisition functions.

### 2.2.2. Inverse distance weighting

Inverse distance weighting [48] (IDW) is a simple interpolation method when interpolating between more than two data points. It multiplies a normalized weight with the value of the data point based on

the distance to the sample point. The IDW value of the interpolated point  $x$  is based on the known points  $N$  and is defined as:

$$IDWscore(x) = \sum_{i \in N} w'_i(x)v(i) \quad (2.1)$$

Where  $v(i)$  is the value of datapoint  $i$  and  $w'_i$  is the normalized weight of sample  $i$  defined as:

$$w'_i(x) = \frac{w_i(x)}{\sum_{j \in N} w_j(x)} \quad (2.2)$$

To normalize weights the individual weight is divided by the sum of all weights. Individual weight is calculated by:

$$w_i(x) = \frac{1}{d^2(x, i)} \quad (2.3)$$

where  $d^2(a, b)$  is the Euclidean distance between point  $a$  and  $b$ .

An advantage of IDW is its simplicity to use and understand but when the number of points increases the computation times becomes considerably slower [48]. Since the weights of the closest neighbours already are the most influential, the computation can be sped up by only computing the weights of the nearest neighbours.

### 2.2.3. Kriging

Kriging (also called Gaussian Process or the design and analysis of computer experiments (DACE) stochastic process model [47]) is a commonly used surrogate type [15]. In a Kriging model, the function is approximated (Equation 2.4) by a global prediction  $\mu(x)$  and a localized deviation based on Gaussian distribution  $\epsilon(x)$  with zero mean and nonzero standard deviation  $\sigma$  (Equation 2.5). The  $\mu(x)$  is typically a polynomial regression function. The result of a trained model with 10 data points can be seen in Figure 2.3.

$$y(x) = \mu(x) + \epsilon(x) \quad (2.4)$$

$$\epsilon(x) \sim N(0, \sigma^2) \quad (2.5)$$

For every untested point the Kriging surrogate can not only give a prediction of the value, but also an uncertainty  $\sigma(x_1)$  and with that uncertainty a confidence interval  $\mu(x_1) \pm \sigma(x_1)$ . This confidence level can then be used in the acquisition function to select the next candidates, for example by calculating the probability of improvement and the Expected Improvement [40].

A drawback of the Gaussian Process is its time complexity of  $O(n^3)$  [27, 22] for training the model where  $n$  is the number of samples. This can cause the training time to become problematic when a large number of data points are used.

### 2.2.4. Other surrogate types

Some other popular surrogate types are Radial Basis Functions (RBFs) and Neural Networks. RBFs are well known from classical machine learning and thus can be used as a form of regression model in EAs [45, 1]. Deep Learning with Neural Networks is currently a very popular machine learning methodology being applied in many different application areas [26]. Training a Neural network to predict fitness values is being used in new SAEAs [55, 37, 43]. While performance can be great, this method suffers from the standard deep learning drawbacks of being not easily explainable and potentially long training times.

## 2.3. State-of-the-Art Evolutionary Algorithms

In this section, a few notable multi-objective real-valued (expensive) EAs will be described. Starting with MOEA/D-EGO which is an older but well-known algorithm. Then K-RVEA and AB-SAEA are described,

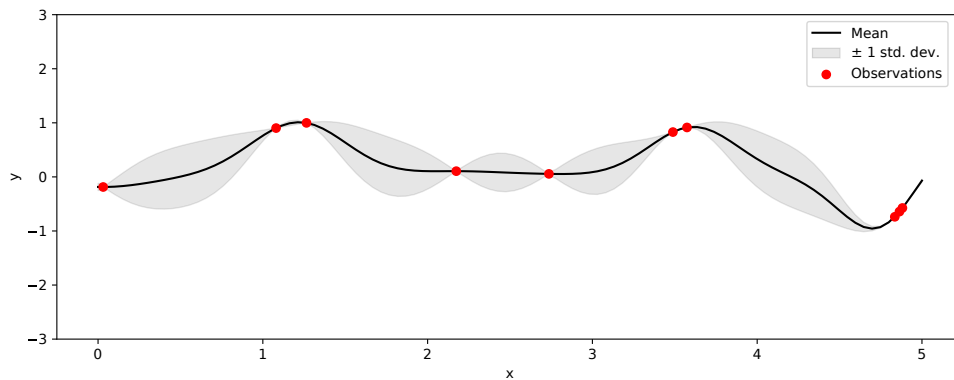


Figure 2.3: One-dimensional result after performing Kriging on nine observations (red). The mean (black) is shown, together with the uncertainty (grey).

which are both state-of-the-art. Last MAMaLGaM will be introduced being the only non-expensive EA in this section.

### 2.3.1. MOAE/D-EGO

MOAE/D-EGO [57] is a combination of the famous Effective Global Optimization (EGO) algorithm [34] and the multi-objective evolutionary algorithm framework called MOAE/D [56]. The MOAE/D framework decomposes the multi-objective problem into scalar optimization subproblems where each subproblem is solved simultaneously. Each subproblem consists of a weighted aggregation of the objectives. Each of these subproblems is optimized using information from their neighbouring subproblems. The neighbours are measured using the distance between the weight vectors. In the MOEA/D-EGO version of this framework, the subproblems are again transformed into an Expected Improvement score by constructing a Gaussian Process model for each aggregation and which is then further optimized to maximize the Expected improvement.

### 2.3.2. K-RVEA

K-RVEA [14] is a combination of the Reference Vector Evolutionary Algorithm (RVEA) [13] with Kriging. The algorithm is originally designed to perform in more than 3 dimensions, however, it is also possible to run it on a bi-objective problem. K-RVEA balances convergence and diversity with the use of the Kriging models and performs selection using the Angle Penalized Distance (APD) [13]. APD utilizes a set of uniformly initialized reference vectors. Each individual is then assigned to its closest reference vector. The fitness of each individual is measured using the length of the vector and the angle between the individual and the reference vector. Since the utopia point is located at the origin the shortest vectors are preferred to longer vectors for increased convergence. To keep diversity, solutions with small angles to the reference vectors are also preferred. These two sub-criteria are combined to perform the selection.

In the K-RVEA algorithm, the objective function is replaced with a kriging model to approximate the fitness. The first samples are initialized and evaluated. These are used to train the surrogate model for each objective. In the next phase, the optimization procedure of RVEA is run to generate and evaluate offspring where the original objective function is replaced by the surrogate. After  $w$  iterations, a fixed number based on sensitivity analysis, the surrogate model is updated by selecting  $u$  individuals from these last generations for re-evaluation. Selection is based on the uncertainty of the kriging model and the APD score. If more diversity is required, more uncertain samples are selected and when convergence is preferred, individuals with good APD values are selected. Which of the two is required is measured by the number of reference vectors without any individuals assigned to them. To limit the training time of the surrogate model, a selection of evaluated individuals was maintained instead of the complete set.

### 2.3.3. AB-SAEA

Adaptive Bayesian Surrogate-Assisted Evolutionary Algorithm (AB-SAEA) [54] is another Kriging-based extension based on RVEA. They use an acquisition function inspired by the lower confidence bound

(LCB), which tries to balance the expected mean and the uncertainty into a single function:

$$LCB(x) = \mu(x) - \kappa\sigma(x)$$

Where  $\kappa$  is the trade-off parameter between prioritizing exploitation (more focus on low  $\mu$ ) or exploration (higher focus on large  $\sigma$ ). They then improve this function by adding an adaptive weight based on the maximum number of evaluations where initially there is more exploration and then finishing gradually more weight is given to the exploitation of good areas. Next, an adaptive sampling strategy is introduced where in the first half of the real function evaluations different selection criteria are used than in the second half. Initially, only the angle to a reference vector is taken as the selection criterion. This will make sure every vector has at least one solution assigned. In the second half of the optimization process, the regular APD metric from RVEA is used, which optimizes convergence and exploitation, while keeping diversity.

#### 2.3.4. MAMaLGaM

Multi-objective Adapted Maximum-Likelihood Gaussian Model (MAMaLGaM) [4] is an Estimation of Distribution algorithm (EDA). In line with other EDAs, it utilizes a sampling strategy to generate offspring based on distribution estimates of the previous generation. Every generation, MAMaLGaM selects the 35% best solutions based on their rank to be divided into equally sized clusters based on the distance in objective scores. For each of these clusters, a multivariate Gaussian probability distribution over all problem variables with maximum likelihood is estimated, which in turn will be used to sample solutions for the next generation. The purpose of these clusters is to direct the search toward different parts of the Pareto front. This results in a more varied optimization direction, creating a better diversity among the population. The algorithm can model the dependencies between variables by learning a covariance matrix. Using this covariance matrix and the mean vectors, the new generation of individuals can be sampled. A second option is to use a factorization of the covariance matrix, where only a subset of variables is used to calculate the covariance. In the case of Univariate factorization, no dependencies are calculated.

MAMaLGaM contains a few mechanisms to improve performance, such as anticipated mean shift. This calculates the change of the mean between generations and adapts the parameters in that direction of the search space, aligning the shape of the estimated probability distributions with the direction of improvement. Another such method is the Adaptive Variance Scaling (AVS) based on the Standard Deviation Ratio (SDR). This changes the variance of the computed normal distribution: when the mean of the distribution does not coincide with the average direction of improvement, which is measured in terms of SDR, the AVS is used to increase the variance of the subsequent distributions. This allows for more effective exploration and traversal through the search space since a wider range of the search space is sampled. Expensive problems usually do not have many generations available to utilise these mechanisms before the FE limit is reached. The algorithm also features an interleaved multi-start scheme (IMS) where multiple populations (subgenerations) are initialized sequentially. These new populations are double the size of the previous population. Smaller populations are terminated once they are outperformed by a larger one, this way a new sub-generation with a larger population size can be initialized. IMS removes a large part of population size tuning since the population will increase automatically once a large population is deemed more successful. Finally, all non-dominated solutions found are stored in an Elitist Archive, to avoid the archive from being infinitely large. The implemented elitist archive is the same as described in [38] and explained earlier in section 2.1.4. MAMaLGaM also offers the option to run multiple optimizations in parallel to reduce the variability of a single run. These populations are completely independent except for their elitist archive.

## 2.4. Virtual Phantom Simulation details

The Virtual Phantom simulation is a FEM simulation of the deformation which occurs after placement of the applicator. This simulation is used to model physically feasible deformations of the vagina/cervix and bladder using the biomechanical properties of the regions of interest of the patient. The simulation is executed in the FEM open-source framework SOFA (v20.12.02). The FEM suitable mesh is created from the organ contour map and applicator contour. These contours are based on a diagnostic MRI scan and delineated by a radiation therapy technologist. The meshes are then rotated and translated into the brachytherapy planning MRI coordinate system of the target images. Using generic literature-

based mechanical properties of the patient, the applicator is inserted inside the vagina-uterus mesh.

Just using these mechanical properties does not result in a satisfactory simulation due to the large variability in mechanical properties of the organs between patients and patient-specific forces that occur during the applicator positioning. Therefore we optimize the biomechanical properties on a patient-specific basis. The details of each parameter can be seen in table 2.1.

Parameter	Lowerbound	Upperbound	Unit
1: Vagina Elasticity	0.001	15	$N/mm^2$
2: Force in left-right direction	-1	1	$kN$
3: Force in front-back direction	-1	1	$kN$
4: Force in head-toe direction	-1	1	$kN$
5: Padding on location 1	0	40	$mm$
6: Padding on location 2	0	40	$mm$
7: Padding on location 3	0	40	$mm$
8: Padding on location 4	0	40	$mm$
9: Bladder Elasticity	0.05	11.5	$N/mm^2$
10: Bladder filling	0	600	$cc$

Table 2.1: Lower and upper bound of every parameter used in the Virtual Phantom Simulation

After the deformation is simulated based on the simulation parameters, two binary images, each containing the vagina-uterus and bladder, are created. The similarity between these images and the target image (seen during the real brachytherapy MRI planning) is calculated. Our objectives are to measure the similarity between the target segmentation images and the results of our simulation, resulting in two objectives: Bladder similarity and vagina-uterus similarity. Since we want to measure volumetric overlap, the dice (or F1) similarity metric [20] is used. Dice score can be formulated as follows:

$$Dice = \frac{2(|O \cap T|)}{|O| + |T|} \quad (2.6)$$

Where  $|T|$  is the number of voxels assigned to the target image of an organ as delineated by the radiation therapy technologist,  $|O|$  are the number of voxels which belong to the organ after the simulation.  $|T \cap O|$  are the voxels where the target and output overlap and thus are correct. We use the two dice scores (one for each organ) as our objective functions. Figure 2.4 shows the overlap between a randomly chosen simulation result and the target deformation. The overlap ( $O \cap T$ ) are the voxels of the target image within the orange contour.

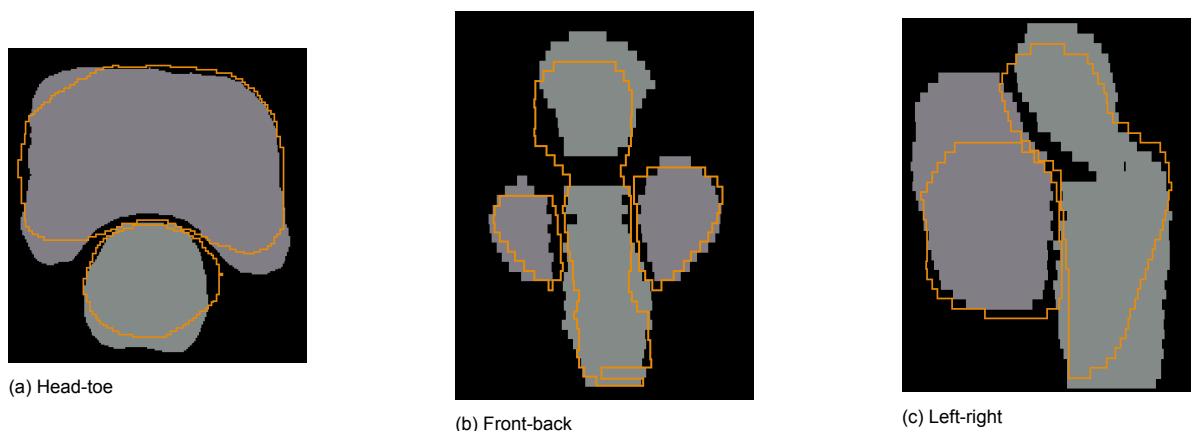


Figure 2.4: Target image and a solution contour in three orthogonal directions. Bladder and Uterus targets are grey and the contour of a solution is orange.





# 3

## Design of Optimization Algorithm: IDW-SAEA

In this section, the design of our Inverse Distance Weighting Surrogate Assisted Evolutionary Algorithm (IDW-SAEA) is described. First, a general overview is given. The later sections describe the optimization of the surrogate (3.2) and the selection of candidates (3.3). Section 3.4 describes changes made to improve the optimization.

### 3.1. Algorithm overview

The IDW-SAEA framework is shown in figure 3.1 with pseudocode in Algorithm 1. The algorithm is designed with bi-objective problems in mind since the Virtual Phantom simulation is a bi-objective problem. The problem objectives are described with  $f_1$  and  $f_2$ . These objectives are expensive to evaluate and thus approximated with a surrogate.

First, the initial population is initialized by latin hypercube sampling [39]. These samples are used as the initial points to calculate the surrogate function for each of the problem objectives. The surrogate based on IDW can be computed on the fly which means it does not require any training, thus the next step is to optimize the surrogate model (section 3.2). This is done by using the state-of-the-art multi-objective Estimation of Distribution Algorithm: MAMaLGaM [4]. However, any MOEA which is capable of solving the surrogate model would have been a suitable candidate. After MAMaLGaM has optimized the surrogate model a set of candidate solutions is selected (section 3.3). These solutions are evaluated using the expensive evaluation function and are added to the set of initial points. This process is repeated until the maximum amount of real function evaluations is reached. We use a fixed generation-based evolutionary control where after a predefined number of evaluations by MAMaLGaM, the surrogate model is updated. MAMaLGaM is reinitialized after every surrogate update, to reduce the risk of convergence to local optima.

**Note:** When looking critically at the algorithm described, one could argue it is not a SAEA since the EA is strictly used for the optimization of the surrogate model. In a sense, the EA is not assisted by the surrogate model but vice versa. However, in recent literature the term SAEA is used to describe algorithms where a similar optimization process is applied [14, 54, 42]. In particular, this type of optimization is referred to as generation-based evolution control [30]. Therefore the choice is made to adopt the SAEA name.

### 3.2. Optimizing the surrogate model

Approximation of objective values with IDW will give a weighted average based on the nearby evaluated solutions. A large drawback of IDW is the weak extrapolation (illustrated in Figure 1.1). When a new individual's genotype is outside the (hyper)volume enclosed by the evaluated samples, the objective score predicted with IDW will be an average of the (closest) neighbours and can never be better than the best already known point. Thus when an EA optimizes this surrogate model, the individuals with the best objective values are those identical to the non-dominated solutions of the already evaluated

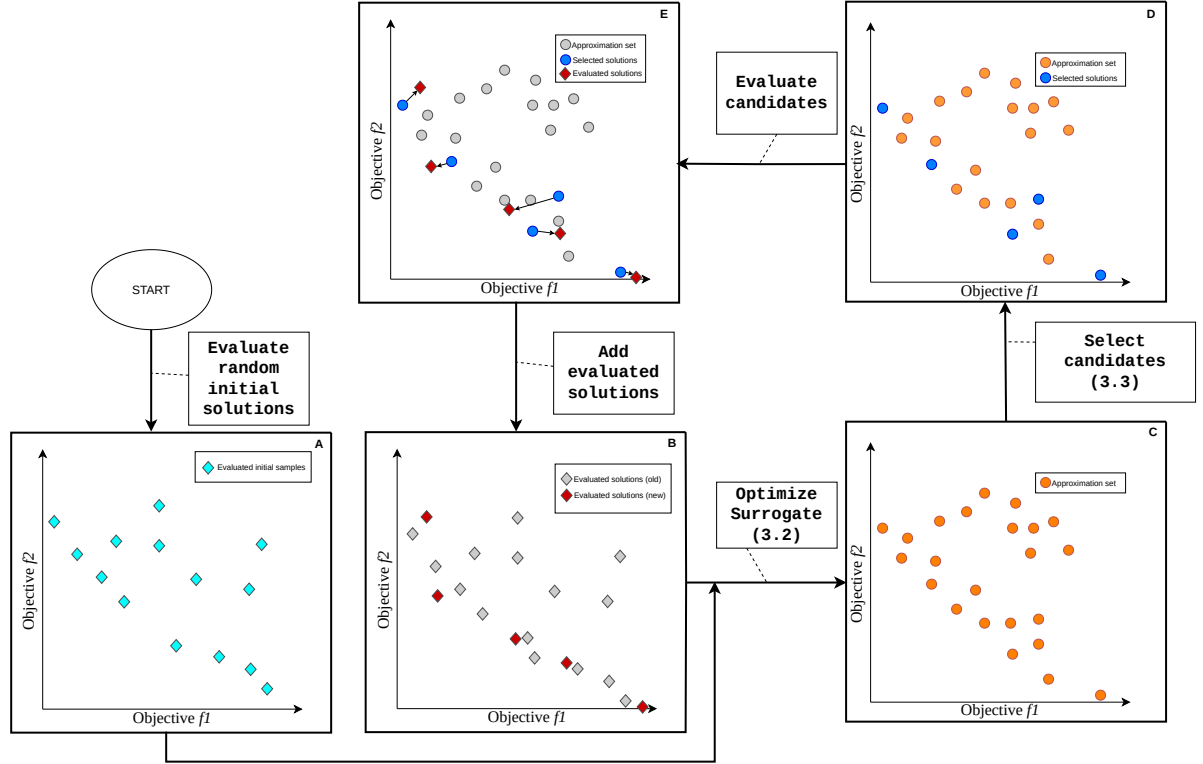


Figure 3.1: The optimization loop of IDW-SAEA. The initial solutions are evaluated (A). Based on these solutions the EA creates a surrogate model and optimizes this (section 3.2) resulting in an approximation set (C). From this approximation set, a few solutions are selected (section 3.3) to evaluate with the true evaluation (D). The selected solutions are evaluated resulting in new evaluated solutions (E). The union between the old and new evaluated solutions (B) is used to optimize the surrogate model again. This loop continues until the maximum amount of function evaluations is reached.

solutions. It also lacks a measure of uncertainty, which we do have in Kriging, making it not possible to easily compute Expected Improvement. A potential solution to these problems is adding an objective to the optimization problem: the exploration objective. The goal of this exploration objective is to create a reason for the EA to optimize in the direction of unknown solutions. The simplest implementation of such an exploration objective is based on the Euclidean distance between the individual in parameter space and the known samples. The best exploration objective score then is gained by maximizing the Euclidean distance between the individual and the closest evaluated samples (Formula 3.1).

$$F(x) = \arg \max_x \left\{ \min_i d(x, i) \right\} \quad (3.1)$$

This problem is similar to the Circles in a Square (CIAS) problem where the goal is to maximize the minimal distance between multiple points inside a square. The CIAS problem is not an easy optimization problem to solve [6] due to the non-smooth landscape. The problem formulation is thus relaxed. The relaxed definition of the CIAS problem as described in [11] is given in Equation 3.2). In the min-max approach the objective score is dominated and determined by only the closest evaluated sample, this results in a non-smoothness which is harder to optimize. Since the new function uses a summation, all the evaluated points are used in the objective score and because of the removal of the min-max operations the function becomes smooth. This helps the EA in optimizing for this objective since smooth functions are generally easier to optimize.

**Algorithm 1** IDW-SAEA

---

**Input:**  $FE_{max}$  = the maximum number of expensive evaluations;  
 $k$  = number of nearest neighbours used in IDW interpolation;  
 $u$  = number of solutions to evaluate every iteration;  
 $SE_{max}$  = the maximum number of surrogate evaluations used by MAMaLGaM;  
**Output:** All expensively-evaluated solutions.

---

- 1: Initialization: Generate the initial set of evaluated solutions  $E$  by sampling a set of solutions using Latin Hypercube Sampling and evaluating these solutions with the expensive objective functions  $f_1$  and  $f_2$ ;
- 2: **while**  $FE \leq FE_{max}$  **do**  
    /\* Optimizing Surrogate \*/
- 3:    $SE = 0$ ;
- 4:   Initialize population(s)  $N$  from MAMaLGaM using the non-dominated solutions from  $E$  and randomly generated solutions;
- 5:   Evaluate  $N$  with the surrogates  $\hat{f}_1$  and  $\hat{f}_2$  using  $k$  neighbours in  $E$ ;
- 6:   Evaluate  $N$  with the exploration objective;
- 7:   **while**  $SE \leq SE_{max}$  **do**
- 8:     Generate offspring  $N'$  using MAMaLGaM;
- 9:     Evaluate  $N'$  with the surrogates of  $\hat{f}_1$  and  $\hat{f}_2$  using  $k$  neighbours;
- 10:     Evaluate  $N$  with the exploration objective;
- 11:     Update Elitist Archive based on the three objectives;
- 12:      $SE = SE + |N'|$
- 13:   **end while**  
    /\* Updating the surrogate \*/
- 14:   Compute 2D ranks of the resulting approximation set  $A$ ;
- 15:   Compute upper bound  $ub$  and lower bound  $lb$  of the selectable ranks;
- 16:   Select  $u$  candidate solutions from  $A$  where  $lb \Rightarrow \text{rank}(u) \leq ub$  using greedy scattered selection;
- 17:   Evaluate candidate solutions  $u$  with the objective functions  $f_1$  and  $f_2$ ;
- 18:   Add the new evaluated candidate solutions to the set  $E$ ;
- 19:    $FE = FE + |u|$ ;
- 20: **end while**

---

$$\begin{aligned}
F_{CIAS}(x) &= \arg \max_x \left\{ \min_{i,j} d(i,j) \right\} \\
&= \arg \min_x \left\{ \max_{i,j} \frac{1}{d(i,j)} \right\} \\
&= \arg \min_x \left\{ \lim_{p \rightarrow \infty} \sqrt[p]{\sum_{i=0}^{|x|-1} \sum_{j=0}^{i-1} \left| \frac{1}{d(i,j)} \right|^p} \right\} \\
&= \arg \min_x \left\{ \lim_{p \rightarrow \infty} \sum_{i=0}^{|x|-1} \sum_{j=0}^{i-1} d(i,j)^{-p} \right\}
\end{aligned} \tag{3.2}$$

Unlike the CIAS problem where the distance between all points is calculated, the exploration objective is only based on the distance of the current individual to all the known samples. This allows for the removal of one of the summations which results in objective function  $F_{\text{exploration}}$  as defined in equation 3.3, with  $x$  being the individual solution,  $N$  the set of all evaluated solutions,  $n$  being a single evaluated solution, and  $d(x, n)$  the Euclidean distance between those two in parameter space. The choice of  $p$ , a parameter to increase or decrease the steepness of the resulting function, of  $-4$  was found to be a good value for the  $p$  norm [11].

$$F_{\text{exploration}}(x) = \sum_{n \in N} d(x, n)^{-4} \quad (3.3)$$

Figure 3.2 shows the exploration objective in the one-dimensional space. The exploration objective has to be minimized. The figure shows that the exploration value increases when the  $x$  value approaches an already evaluated sample. The best objective scores can be found with  $x < 3.5$  or  $x > 8.5$ . These samples are far away from the already known and therefore have the best scores in the exploration objective. Because of the  $p$  norm value of  $-4$  the exploration score is quick to increase or decrease. This allows for the values between the samples to decrease making sure the exploration does not only prioritize extrapolation but can also detect sparse regions during optimization.

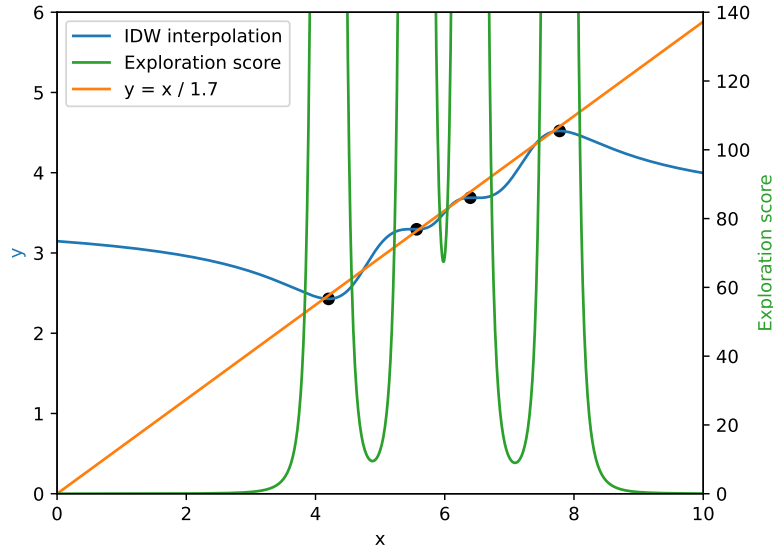


Figure 3.2: One-dimensional example containing 4 samples generated from the line  $y = x/1.7$  with a small amount of noise. In blue the estimated value based on the IDW is shown, in green the exploration score for every sample value of  $x$  is shown. The exploration score is lower when further away from known samples.

Next to the smoothness the new exploration function (Equation 3.3) has a second smaller benefit over using Equation 3.1: The new function prefers a sparse neighbourhood over a dense neighbourhood even if the closest neighbour is the same distance in every scenario. This promotes the less densely sampled areas of the search space, further improving the exploration of the algorithm.

Without the exploration objective, only the IDW interpolation would be optimized and (given successful optimization) the EA would always end up on the already known samples. The introduction of the new objective increases the dimensionality of the problem. The combination of the problem objectives and the exploration objective will allow more solutions into the front. Solutions which would be dominated based on the problem objective(s) can, with the introduction of the exploration objective, become non-dominated. In Figure 3.2 all solutions where  $x$  is smaller than the known sample at  $x = 4.1$  would not qualify for the approximation set of this one-dimensional problem, because the IDW interpolation prediction is worse than the value achieved at  $x = 4.1$ . The addition of the exploration objective enables all these solutions to become non-dominated. The predicted objective score is worse, however, the exploration score is better than every solution with larger  $x$  and where  $x < 4.1$ . All solutions where  $x$  is larger than 4.1 are dominated by the solutions where  $x < 4.1$  because in this case, the exploration score is better as well as the objective score estimation.

Since we want to optimize multi-objective problems, especially the Virtual Phantom simulation, we first consider bi-objective problems where adding the exploration objective would make this problem three-dimensional. Figure 3.3 shows a two-dimensional plot of an approximation front. The third (exploration) objective magnitude is visualized with a colour. The approximation front by design only contains non-dominated solutions. Therefore we can conclude that every solution which seems dominated in the

two-dimensional image, has a better exploration objective score than the solutions it is dominated by in the two problem objectives. Keeping this in mind we can start looking at how to select solutions from this approximation set in such a way that there is a good balance between exploration and exploitation.

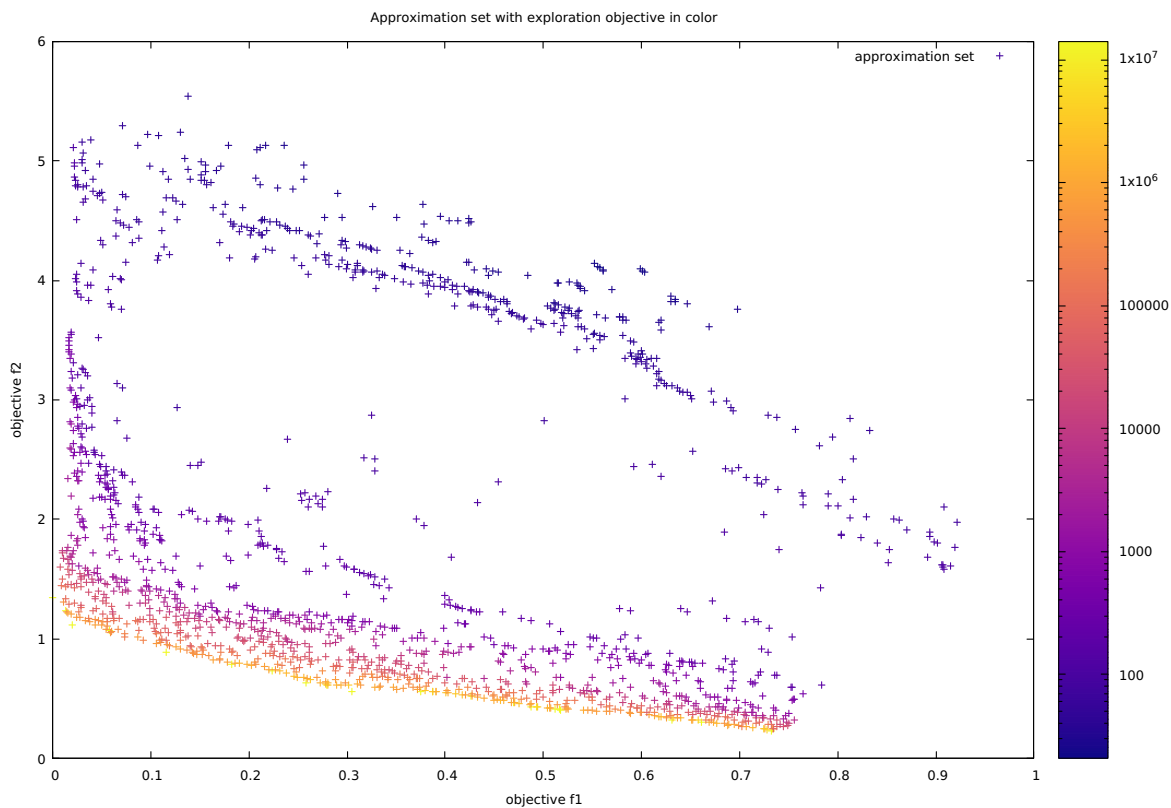


Figure 3.3: Approximation set of the ZDT1 problem during the optimization process. The objectives  $f_1$  and  $f_2$  are shown on the axis. The exploration objective is shown in colour. All these three objectives are to be minimized. The image shows the trade-off between closer to the problem minima and a better exploration score.

### 3.3. Selection of candidate solutions

After optimization of the three objectives, the best individuals for the real evaluation have to be selected. These individuals will be evaluated with the true objective function and used to update our surrogate. However, determining these best individuals a priori is a challenging task. After all, we want to evaluate the individuals which will improve the set of evaluated solutions the most, which is not known before evaluating. Individuals close to known samples should be avoided since they will most likely not be very different in fitness. On the other hand, individuals that are far away, thus have good fitness in the exploration objective, but have poor fitness in the problem objectives should also be avoided. The individuals that are only optimized on exploration objective, have a low probability of being an improvement in  $f_1$  or  $f_2$ . A balance between the problem objectives and the exploration objective has to be found when selecting the samples. Preferably we select individuals with good problem objective scores who are also far away from the known samples.

To help select individuals that are nearby the front but with enough distance, we introduce the 2D ranks selection methodology. Every individual in the final population is assigned a 2D rank based on the problem objectives, i.e., objectives  $f_1$  and  $f_2$ . The first rank will always be the already known samples, then subsequent ranks will contain solutions with a lower fitness in the problem objectives. However, since they are in the final approximation set, all solutions are non-dominated, thus these solutions will have a higher performance in the exploration objective. The highest ranks will have the most exploration and the lowest predicted interpolation score. We now have one metric which can be used to select the amount of exploitation and exploration.

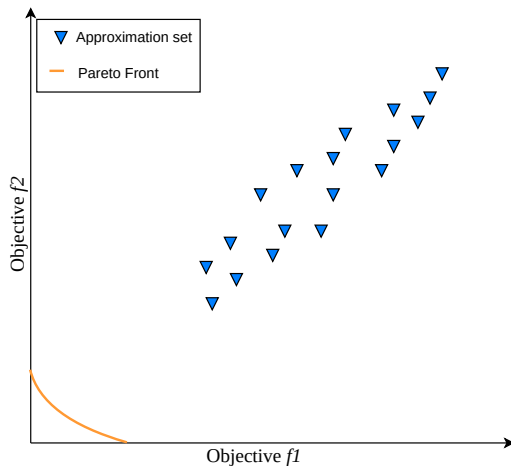
By applying a selection restrictions to certain ranks we can control if we want more exploration or

more exploitation. We propose three different methods of deciding the allowed ranks. The first method is to allow all individuals from the approximation set to be selected. This method does not utilize the ranks and will be used as a baseline. The second selection method is to only allow the lowest 15 percent of ranks to be selected. For example if we have in total 37 ranks, the allowed ranks would be ranks 0 to 5 ( $\lfloor 0.15 \cdot 37 \rfloor = 5$ ). The choice of selecting the 15% ranks is based on some preliminary testing. When the percentage cutoff is lower (<5%) the optimization has fewer early improvements and overall slower convergence. With a percentage cutoff which is higher (>50%), more explorative solutions are found which impede the exploitation once the approximation front nears the Pareto front. The 15% rank selection is a compromise that allows for some variation around the best known samples but still has a bias towards the problem objective scores and thus exploitation. This method is depicted in Figure 3.4b.

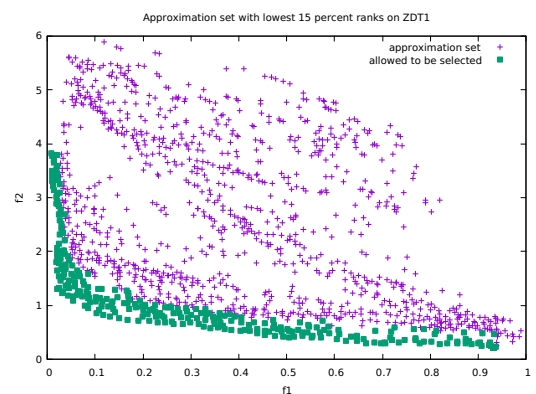
The third selection method is adaptive. This selection method is designed to increase performance in cases where the optimum is far away from the current population, in such a case there is no trade-off between the problem objectives (yet). Resulting in an approximation set which is shaped like a comet with a tail (Figure 3.4a). To decide if we are in this scenario we introduce a new metric called the Rank Sparsity (RS) metric. Rank Sparsity is defined as:

$$RS = \frac{|A|}{|R|} \quad (3.4)$$

Where  $|A|$  is the number of individuals in the approximation set and  $|R|$  is the number of ranks in the population. In cases where the samples are far away from the optimum, the algorithm will not yet find a large trade-off between the two problem objectives. This results in many ranks which only contain one or two individuals since the largest trade-off is between the exploration objective and problem objectives. Thus if the Rank Sparsity becomes higher it is more likely for the algorithm to be far away from the optimum. This is an indication that more exploration is required. In the adaptive selection method, once the rank sparsity reaches a value higher than 0.55, individuals from ranks between 65% to 95% of the total ranks will be allowed for selection. This way we promote more diversity between the known solutions and the candidates. By increasing the lower bound of the available ranks we force the candidate solutions to be further away from the known solutions. This creates more explorative solutions which have the possibility of a larger improvement compared to the lower rank solutions. We never allow the ranks 95% to 100% to be selected. These ranks are too focused on optimizing the exploration objective and have poor surrogate objective performance and consequently a very high probability of poor real-problem performance.



(a) The comet-shaped approximation set where there is no trade-off between objectives and a high rank-sparsity.



(b) Approximation set with the lowest 15 percent of ranks highlighted. These are the individuals which are eligible to be selected as candidates for evaluation.

Figure 3.4: Ranks selection illustrations

If the Rank Sparsity is lower than 0.05, meaning there are on average 20 individuals in the same

rank, the approximation front models the trade-off between both objectives. In such a case it is likely the approximation front is near the Pareto front. We want to perform small variation near the front since the solutions with a small deviation are the most promising to achieve actual improvements of the approximation set. Therefore we allow only the selection of solutions in rank 1. Rank 0 are the known individuals, which are always found since we introduce them in the starting population of the EA, thus rank 1 is the first rank consisting of new solutions. Selecting rank 1 will focus mainly on exploitation since we do not allow for far-away solutions and are only looking for small improvements to the best known samples. The choice of only rank 1 solutions is based on preliminary experiments with benchmark problems, these showed the rank 1 solutions containing the best solutions of the population when a low RS was obtained.

If neither of these two criteria hold, the adaptive selection will just select ranks lower than 30% to have some form of balance between exploration and exploitation. The formula of the rank selection can be seen in 3.5 and 3.6

$$\text{lowerbound} = \begin{cases} 1 & \text{if } RS \leq 0.05 \\ |R| * 0.65 & \text{if } RS \geq 0.55 \\ 1 & \text{otherwise} \end{cases} \quad (3.5)$$

$$\text{upperbound} = \begin{cases} 1 & \text{if } RS \leq 0.05 \\ |R| * 0.95 & \text{if } RS \geq 0.55 \\ |R| * 0.3 & \text{otherwise} \end{cases} \quad (3.6)$$

In every selection phase, ten individuals are selected to be expensively evaluated. In the VIR-SIM problem, the individuals can be evaluated in parallel. This allows for selecting multiple candidate solutions in every selection phase, allowing for faster optimization in the same amount of time. The individuals selected for evaluation are selected by doing a greedy selection method between lower bound and upper bound, inclusive. We want to evenly select solutions between these bounds. The selection algorithm starts by selecting the best performing individual in a random objective, followed by iteratively selecting the furthest away individual based on Euclidean distance in the normalized objective space from the already selected individuals. This is repeated until the ten individuals are selected to be subject to a real evaluation.

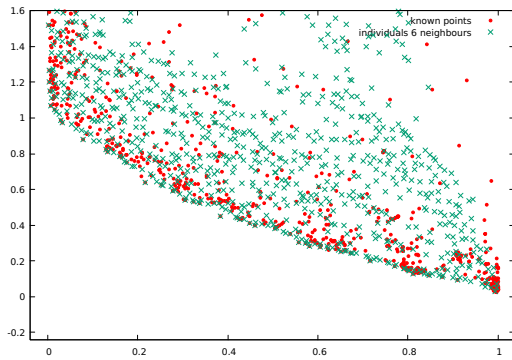
### 3.4. Additional challenges and improvement of optimization algorithm

With the addition of the exploration objective, solving the surrogate model with MAMaLGaM becomes a three-dimensional problem and therefore more difficult to optimize. This especially becomes harder the closer samples get to the Pareto front and when the number of evaluated samples increases in size. Since the selection of solutions to be evaluated depends on the individuals in the resulting approximation set, it is of importance that the surrogate is solved to at least acceptable quality. Acceptable quality is when the EA finds the previously evaluated solutions on which the surrogate model is built. In this section implemented strategies to improve the quality of the approximation set are described.

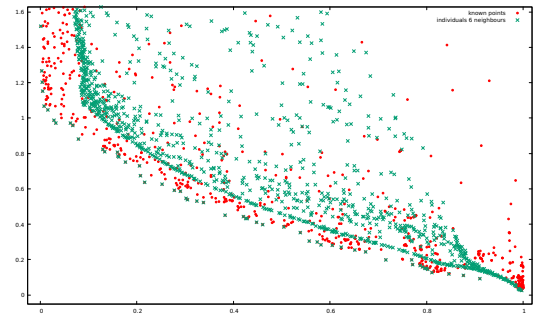
In standard IDW all samples will have a weighted influence on the interpolation point. This causes the objective space to be influenced by the closest few neighbours and the average of the whole set. This average creates a large number of local minima in cases where the interpolation point is not nearby a good sample. To reduce the effect of the global average the IDW is calculated using only the  $k$  nearest neighbours. To do this we replace the set of all known solutions ( $N$ ) in Equation 2.1 and 2.2 with the set of  $k$  nearest neighbours.

This results in the interpolation of only the nearby samples and less influence of the further away samples which reduces the difficulty of the landscape. A difference in approximation set between the different number of neighbours used in the IDW can be seen in figure 3.5b and 3.5a.

A second improvement is to use an alternative weighting function for IDW as introduced in [35] given in equation 3.7 which replaces equation 2.3. This new weighting function additionally reduces the effect of points which are far away. This reduces the effect of neighbours who are still far away from the individual's location to make sure the predicted objective score is not generalized to the mean.



(a) Samples and individuals with 6 nearest neighbours



(b) Samples and individuals with 300 nearest neighbours

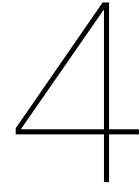
Figure 3.5: Comparison of 6 nearest neighbours and 300 nearest neighbours. In 3.5b can be seen what happens when the number of nearest neighbours is too large: All the objective values get pushed towards the mean value of the known samples.

$$w_i(x) = \frac{e^{-d^2(x,i)}}{d^2(x,i)} \quad (3.7)$$

To help MAMaLGaM optimize the surrogate model, the best known samples are included in its initial population. This is done by calculating the non-dominated solutions of the evaluated samples and adding these to the initial population of MAMaLGaM. In this way, we try to include as much prior knowledge as we have of the best solutions in the initial population.

Another challenge of the IDW interpolation is that a small change in the parameters will also give a small change in the objective space. After all, a small parameter change creates only a small change in the weights and since the IDW interpolation is smooth the effect will be small. This is a very good property of the algorithm since it allows the creation of a "front" of individuals between samples. However, it also allows for many extremely similar solutions. In some cases a single MAMaLGaM cluster became centered around a sample, only generating almost identical solutions. These inflated the total amount of 2D ranks in the final population when calculating the ranks. Additional to this, the elitist archive also becomes very similar which again degrades the performance of the algorithm. This is solved by setting the elitist archive size to a small enough number (in our case 1500) such that the maximum capacity is achieved relatively quick, at least before the optimization on the surrogate model is finished. This will trigger the hypercube-based archive pruning (as described in Section 2.1.4) and therefore remove solutions which are too similar.





# Experiments

In this section, the experimental setup is described. First, we describe the benchmark problems and algorithms used in the experiments. This is followed by a baseline test for the MAMaL GaM algorithm. MAMaL GaM is not designed with expensive optimization in mind, hence the parameters need to be tuned to increase performance with the low number of function evaluations. Thirdly experiments with the IDW-SAEA parameter settings are described. Thereafter, we compare the results of the MAMaL GaM baseline and IDW-SAEA with a few state-of-the-art expensive evolutionary algorithms utilizing the Gaussian Process on the benchmark problems. Lastly, the results are shown on the Virtual Phantom simulation problem.

## 4.1. Setup

This section defines the test problems and discussed the specifics of how the experiments are conducted.

### 4.1.1. State of the Art algorithm settings

In the benchmarks, we compare the IDW algorithm to the well-known MOAE/D-EGO algorithm and the two other kriging state-of-the-art expensive optimization algorithms: K-RVEA and AB-SAEA. Additionally, the performance of IDW-SAEA is also compared with MAMaL GaM without IDW implementation. Every algorithm has a budget of 1000 function evaluations (FE). Every algorithm is run 10 times to gain insights into the variability. Every expensive optimization algorithm is run using  $11 \cdot D - 1$  initial samples initialized using random Latin hypercube sampling, with  $D$  being the number of parameters to optimize. This is in line with state-of-the-art algorithms. The Latin hypercube sampling is implemented using the latinize C library <sup>1</sup>. MOEA/D-EGO, K-RVEA and AB-SAEA are all implemented using PlatEMO [51] and ran using default parameters. The hyperparameters of the SotA algorithms are the same as in their original papers.

### 4.1.2. Benchmark optimization problems

The algorithms were tested on the benchmark ZDT [58] problems, except for ZDT5 since this is a discrete parameter problem. In addition to these benchmarks, the genMED problem [5] with initialization range of  $[-10, 10]$  and  $[5, 10]$  was included. These two genMED problems were chosen to see how the algorithm performs on a relatively simple problem, but with a large exploration range. The  $[5, 10]$  problem was included to see the performance where the optimum is outside the initialization range. The last considered benchmark problem is BD2 (scaled) [8], which was chosen because of the difference in difficulty between the two objectives. The definitions of the problems are given in Figures 4.1 to 4.7. All problems are run with 10 parameters since the Virtual Phantom simulation also contains 10 parameters. Additionally, 10 dimensions is also a reasonable difficulty to be able to judge the performance of the algorithms.

---

<sup>1</sup>[https://people.sc.fsu.edu/~jburkardt/c\\_src/latinize/latinize.html](https://people.sc.fsu.edu/~jburkardt/c_src/latinize/latinize.html)

**ZDT1**

ZDT1 (Figure 4.1) is an  $n$ -dimensional bi-objective problem with a convex Pareto front. The definition of the objective functions can be seen in Figure 4.1a. As is the case with all ZDT problems, the first objective is defined by the first parameter. The second objective is defined by the other  $n-1$  parameters. The parameter range is equal for all parameters and ranges between 0 and 1.

**ZDT2**

ZDT2 (Figure 4.2) is a non-convex bi-objective function closely resembling the ZDT1 function, with a concave Pareto front.

**ZDT3**

ZDT3 (Figure 4.3) features a discontinuous front. The Pareto front is divided into five disconnected segments which are introduced by the sine function. In parameter space, there is no discontinuity.

**ZDT4**

ZDT4 (Figure 4.4) contains many local optima in the second objective which increases the difficulty of the problem. The parameter boundaries are also changed to a range of -5 to 5 except for the first parameter  $x_0$ .

**ZDT6**

ZDT6 (Figure 4.5) contains two challenges: the solutions are non-uniformly distributed along the front and the density of solutions is lower near the front.

**genMED**

genMED (convex) [5] (Figure 4.6) is quite a simple problem defined by the distance to the point  $[1, 0, 0 \dots 0]$  for objective 1 and distance to point  $[0, 1, 0 \dots 0]$  for the second objective. The problem is smooth, equally scaled and does not contain local minima. Additionally, both objectives are dimension-wise decomposable. The Pareto set is given by the formula  $x_1 = -x_1 + 1$  with  $x_0 \in [0, 1]$  and all other parameters equal to zero. The only challenge of this problem is the unconstrained parameter range. For practical purposes the parameters were constrained to  $[-1 \times 10^{10}, 1 \times 10^{10}]$ . The initialization contained two versions, one with an initialization range of  $[-10, 10]$  and a second initialization range of  $[5, 10]$ . These problems will be referred to as genMED $[-10, 10]$  and genMED $[5, 10]$  respectively.

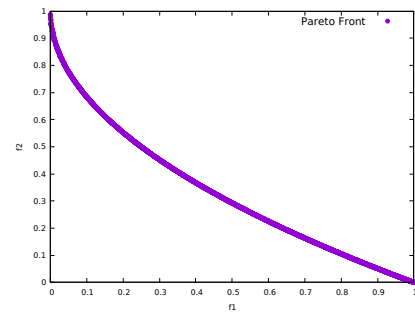
**BD<sub>2</sub> (scaled)**

BD<sub>2</sub> (scaled) [8] (Figure 4.7) is a bi-objective function which utilizes the Rosenbrock's function for its second objective while the first objective is the sphere function. This difference in difficulty between objectives will likely lead to premature convergence on objective 1. This benchmark is added to investigate if expensive optimization algorithms can optimize problems where one objective is substantially more challenging than the other.

$$\begin{aligned}
 f_1(x) &= x_0 \\
 f_2(x) &= g(x) \cdot h(x) \\
 g(x) &= 1 + 9 \cdot \sum_{i=1}^n \frac{x_i}{n-1} \\
 h(f_1(x), g(x)) &= 1 - \sqrt{\frac{f_1(x)}{g(x)}} \\
 x &\in [0, 1]
 \end{aligned}$$

(a) Equation

Figure 4.1: Equation and Pareto front of ZDT1

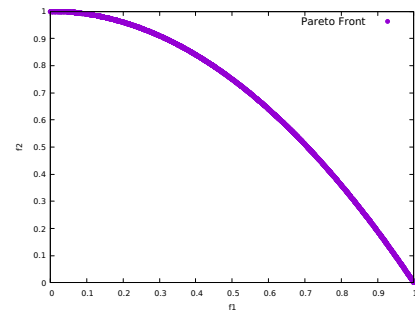


(b) Pareto Front

$$\begin{aligned}
 f_1(x) &= x_0 \\
 f_2(x) &= g(x) \cdot h(x) \\
 g(x) &= 1 + 9 \cdot \sum_{i=1}^n \frac{x_i}{n-1} \\
 h(f_1(x), g(x)) &= 1 - \left(\frac{f_1(x)}{g(x)}\right)^2 \\
 x &\in [0, 1]
 \end{aligned}$$

(a) Equation

Figure 4.2: Equation and Pareto front of ZDT2

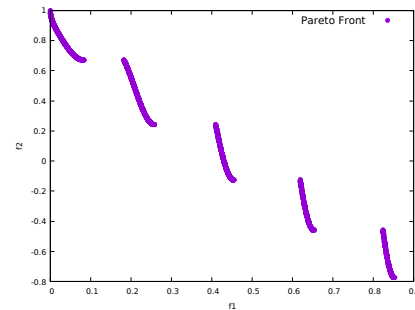


(b) Pareto Front

$$\begin{aligned}
 f_1(x) &= x_0 \\
 f_2(x) &= g(x) \cdot h(x) \\
 g(x) &= 1 + 9 \cdot \sum_{i=1}^n \frac{x_i}{n-1} \\
 h(f_1(x), g(x)) &= 1 - \sqrt{\frac{f_1(x)}{g(x)}} - \frac{x_0}{g(x)} \cdot \sin(10\pi x_0) \\
 x &\in [0, 1]
 \end{aligned}$$

(a) Equation

Figure 4.3: Equation and Pareto front of ZDT3

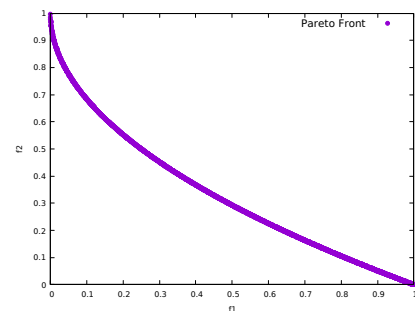


(b) Pareto Front

$$\begin{aligned}
 f_1(x) &= x_0 \\
 f_2(x) &= g(x) \cdot h(x) \\
 g(x) &= 1 + 10(n-1) + \sum_{i=1}^n (x_i^2 - 10 \cos(4\pi x_i)) \\
 h(f_1(x), g(x)) &= 1 - \sqrt{\frac{f_1(x)}{g(x)}} \\
 x_0 &\in [0, 1], x_i \in [-5, 5]
 \end{aligned}$$

(a) Equation

Figure 4.4: Equation and Pareto front of ZDT4

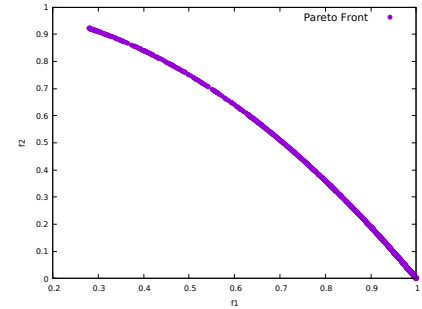


(b) Pareto Front

$$\begin{aligned}
 f_1(x) &= x_0 - \exp(-4x_0) \sin^6(6\pi x_0) \\
 f_2(x) &= g(x) \cdot h(x) \\
 g(x) &= 1 + 9 \cdot \left( \sum_{i=1}^n \frac{x_i}{n-1} \right)^{0.25} \\
 h(f_1(x), g(x)) &= 1 - \sqrt{\frac{f_1(x)}{g(x)}} \\
 x &\in [0, 1]
 \end{aligned}$$

(a) Equation

Figure 4.5: Equation and Pareto front of ZDT6

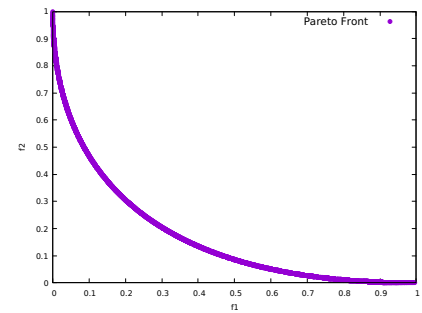


(b) Pareto Front

$$\begin{aligned}
 f_1(x) &= \frac{1}{2} \left( (x_0 - 1)^2 + \sum_{i=1}^n x_i^2 \right) \\
 f_2(x) &= \frac{1}{2} \left( x_0^2 + (x_1 - 1)^2 + \sum_{i=2}^n x_i^2 \right) \\
 x &\in [-1, 1]^{10}
 \end{aligned}$$

(a) Equation

Figure 4.6: Equation and Pareto front of genMED convex version.

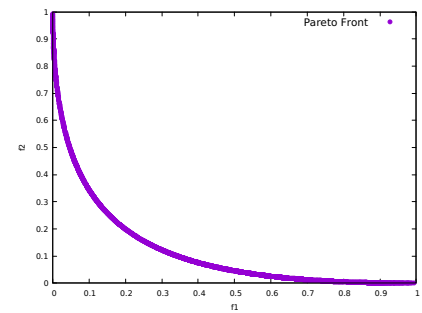


(b) Pareto Front

$$\begin{aligned}
 f_1(x) &= \frac{1}{n} \sum_{i=0}^{n-1} x_i^2 \\
 f_2(x) &= \frac{1}{n-1} \sum_{i=0}^{n-2} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2) \\
 x &\in [-1, 1]^{10}
 \end{aligned}$$

(a) Equation

Figure 4.7: Equation and Pareto front of BD2.



(b) Pareto Front

## 4.2. MAMaLGaM parameter tuning

This section describes the experiments conducted to find the best MAMaLGaM settings when ran with a low number of available evaluations. MAMaLGaM is not designed as an expensive optimization algorithm, therefore some benchmark settings might not be applicable. Especially the number of clusters and the choice between univariate or full factorization are important to be set correctly. The number of clusters directly influences the population size which in turn is important to determine the number of generations possible. The factorization influences which covariances are calculated in the covariance matrix. The full covariance matrix includes all the interactions between parameters. This calculation of the full covariance matrix is more time-consuming than using the Univariate factorization.

### 4.2.1. MAMaLGaM settings

MAMaLGaM, both full and univariate, was restricted to 1000 FE. The algorithm was run with the guidelines settings from [7]. The population size was determined by using the single-objective AMaLGaM formulas as described in the same publication and then multiplied by  $0.5 \times k$  where  $k$  is the number of clusters to create the population size for MAMaLGaM. This results in the following formulas with  $D$  representing the number of dimensions and  $k$  the number of clusters:

$$n = 0.5 \cdot k \cdot (17 + 3 \cdot D^{1.5}) \quad (4.1)$$

and

$$n = 0.5 \cdot k \cdot (10 \cdot D^{0.5}) \quad (4.2)$$

for the full covariance matrix and univariate factorization, respectively. We want the algorithm to have at least two generations, thus the population size must not be larger than 500. If the population size is larger we cannot do any variation and are thus doing random search. Therefore the following cluster sizes were experimented with:  $k = [2, 3, 4, 5, 8, 10]$  for both the univariate factorization and full covariance matrix, and  $k = 15$  for the Univariate model only. The overview of the settings used during this experiment can be seen in table 4.1. Each setting is run 30 times to reduce variability.

Parameter	Value
tau	0.35
population size	Equation 4.1 or 4.2
number of populations	1
number of clusters	$k$
distribution multiplier	0.9
standard-deviation ratio	1
Elitist archive size	100
maximum number of evaluations	1000
number of subgenerations	4

Table 4.1: Settings used when experimenting with cluster size for MAMaLGaM with 1000 evaluations. After the experiments  $k = 4$  is chosen with the Univariate model resulting in population size = 63.

### 4.2.2. Results MAMaLGaM tuning

For each run, the hypervolume metric is calculated using  $[10,10]$  as the reference point for the ZDT benchmark problems except for ZDT4 where a reference point of  $[250,250]$  is used. The reference point was set on  $[30,30]$  for the BD2 and genMED benchmarks. As can be seen in figure 4.8 the Univariate model performs better in all cases except ZDT4. In ZDT4 the performance of the Univariate and Full factorization are equal. The Univariate model with 2 to 5 clusters performs best, which specific version is problem dependent and not clearly defined. Therefore the Univariate model with 4 clusters is chosen since it is a balanced middle way. This will be the setting that is used for MAMaLGaM in the next sections. The results with the IGD metric can be found in the Appendix (Figure A.3).

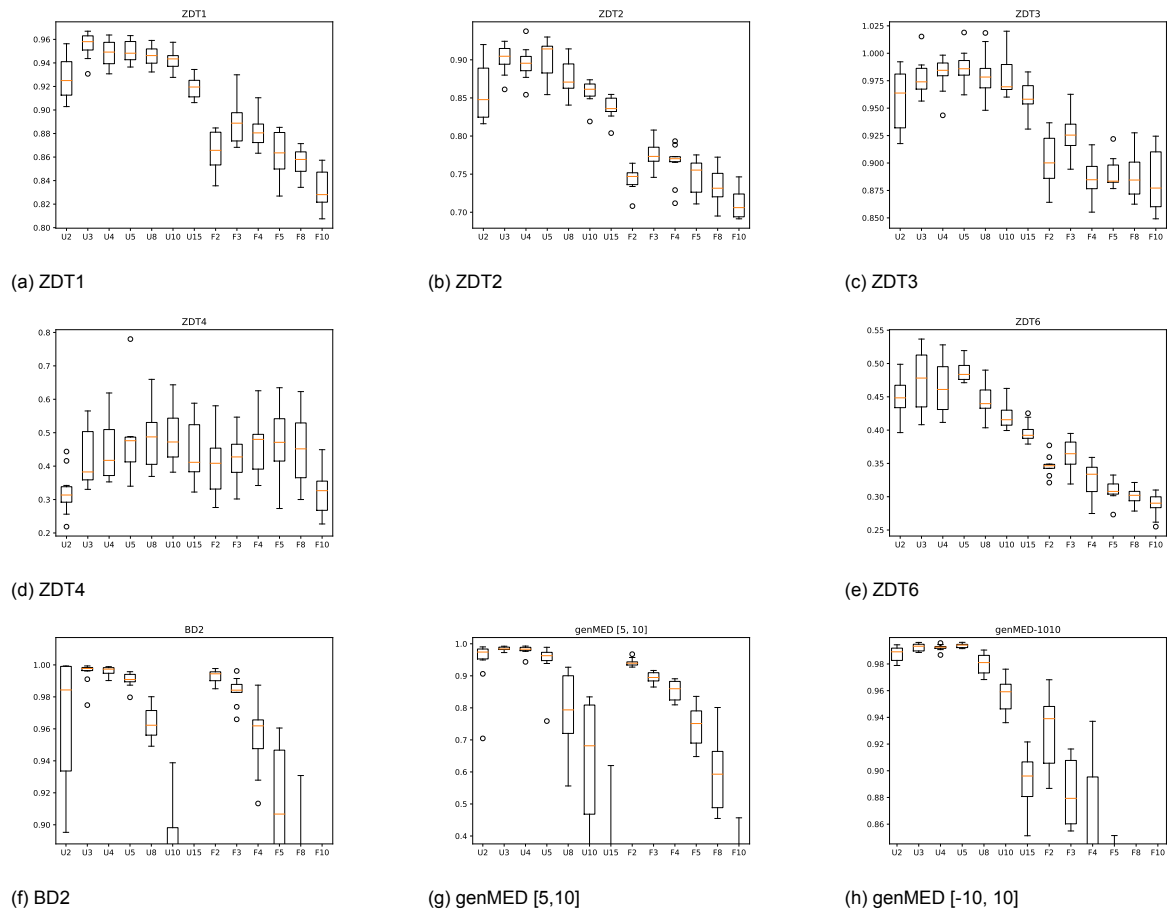


Figure 4.8: Hypervolume scores of MAMaLGaM with 1000 FE on different benchmark problems. Different numbers of clusters were experimented with in combination with the Univariate factorization (U) and Full covariance matrix (F), the number of clusters is indicated with the number after the U of F.

### 4.3. IDW-SAEA parameter tuning

The inverse distance weighting adaptation of MAMaLGaM has two settings: the number of neighbours to be used in the IDW calculation and the method for selecting ranks from which candidate solutions can be selected. A large number of neighbours results in a more averaged estimation of the objective space, which in turn has as an advantage a more robust estimation which is less sensitive to local minima. However, it also has a disadvantage in not always being capable of exploiting the best solutions since the objective value prediction will be influenced by more samples which can negatively impact the prediction. The difference between the final set of individuals can be seen in Figure 3.5. Too many neighbours restrict the optimization from finding candidate solutions which are near the problem objective front (Figure 3.5b). Preliminary testing showed good performance with a few neighbours. Therefore experiments with 2, 5 and 15 neighbours were conducted. The final parameters used by MAMaLGaM when optimizing the surrogate model in the IDW-SAEA loop can be found in table 4.2. IMS was used to potentially increase the population size to a size of 1600.

The second setting is the ranks from which the candidate solutions can be selected as described in section 3.3. Here we designed the adaptive selection rule and the fixed selection rule. Both of these were compared with the baseline where no rank-based selection occurs. These settings are referred to as S1, S2 and S3, respectively. These three selection methods were combined with 3 different sizes of nearest neighbours: 2, 5 and 15 neighbours to create 9 different setups to run experiments with. These combinations of settings are indicated with, e.g., S2N5 for selection setting 2 and number of neighbours equal to 5. These setups were each run 10 times on all the benchmark problems to investigate the performance of the setups.

Parameter	Value
tau	0.35
population size	400
number of populations	3
number of clusters	30
distribution multiplier	0.9
standard-deviation ratio	1
Elitist archive size	1500
maximum number of evaluations	150000
number of subgenerations	4

Table 4.2: Settings used for MAMaLgAM inside the IDW-SAEA. A full covariance matrix was used.

### 4.3.1. Results settings IDW-SAEA

Setting	[10,10] ZDT1			[10,10] ZDT2			[10,10]* ZDT3			[250,250] ZDT4			[10,10] ZDT6		
	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)
s1n2	0.9911	0.9931	0.9941	0.8983	0.9070	0.9671	0.8505	0.8700	0.9027	0.8537	0.8997	0.9439	0.6892	0.7483	0.7658
s1n5	0.9928	0.9950	0.9954	0.8986	0.9616	0.9862	0.8391	0.8891	0.8963	0.7969	0.8910	0.9586	0.7108	0.7312	0.7478
s1n15	0.9909	0.9932	0.9948	0.9130	0.9788	0.9867	0.8406	0.8679	0.8944	0.8128	0.8814	0.9404	0.6379	0.7342	0.7801
s2n2	0.9825	0.9892	0.9920	0.9410	0.9713	0.9783	0.8624	0.8781	0.9011	0.7993	0.8976	0.9746	0.6624	0.6913	0.7086
s2n5	0.9858	0.9888	0.9932	0.8994	0.9664	0.9810	0.8592	0.8866	0.8992	0.8192	0.8582	0.9474	0.6233	0.7094	0.7705
s2n15	0.9854	0.9877	0.9919	0.9525	0.9711	0.9781	0.8772	0.8915	0.9052	0.8258	0.8781	0.9105	0.6475	0.6797	0.7534
s3n2	0.9575	0.9674	0.9818	0.8939	0.9312	0.9658	0.8146	0.8484	0.8749	0.7991	0.8467	0.9082	0.5461	0.5906	0.6937
s3n5	0.9711	0.9804	0.9891	0.9023	0.9204	0.9745	0.8521	0.8621	0.8886	0.7698	0.7952	0.8298	0.5441	0.6205	0.6702
s3n15	0.9667	0.9810	0.9897	0.9445	0.9586	0.9733	0.8607	0.8765	0.9058	0.7540	0.8129	0.9169	0.5361	0.5693	0.6355

Setting	[6,6] GENMED [-10,10]			[30,30] GENMED [5, 10]			[40,40] BD2			[1,1] VIRSIM		
	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)
s1n2	0.9854	0.9929	0.9942	0.8961	0.9972	0.9986	0.9674	0.9715	0.9763	0.7230	0.7306	0.7343
s1n5	0.9229	0.9853	0.9927	0.9932	0.9980	0.9990	0.9735	0.9751	0.9754	0.7189	0.7273	0.7342
s1n15	0.9732	0.9793	0.9898	0.9637	0.9949	0.9977	0.9580	0.9728	0.9746	0.7215	0.7254	0.7283
s2n2	0.0092	0.9028	0.9797	0.9201	0.9669	0.9820	0.9471	0.9675	0.9750	0.7236	0.7274	0.7321
s2n5	0.8933	0.9666	0.9898	0.9652	0.9924	0.9989	0.9705	0.9744	0.9763	0.7233	0.7276	0.7311
s2n15	0.8056	0.9719	0.9898	0.2229	0.7598	0.9893	0.9087	0.9679	0.9737	0.7163	0.7234	0.7282
s3n2	0.9330	0.9510	0.9648	0.8579	0.9660	0.9901	0.9148	0.9721	0.9739	-	-	-
s3n5	0.9407	0.9729	0.9838	0.9322	0.9896	0.9969	0.9586	0.9715	0.9748	-	-	-
s3n15	0.8849	0.9550	0.9702	0.2285	0.5961	0.9819	0.8410	0.9671	0.9743	-	-	-

Figure 4.9: Hypervolume results of the different settings in all benchmark problems and the Virtual Simulation. For every problem, the reference point is indicated left of the name. \*ZDT3 uses the point [-0.773368556, -0.773368556] as the origin since the second objective can achieve negative values.

The results of the different settings are shown in Figures 4.9 and 4.10. For the ZDT problems, a reference point of [10,10] was chosen except for ZDT4 where a reference point of [250,250] was required to dominate all the solutions. The genMED[-10,10], genMED[5,10] and BD<sub>2</sub> had a reference points of [6,6], [30,30] and [40,40] respectively. The Virtual Phantom simulation (VIRSIM) problem uses a reference point of [1,1]. The tables show the 1st decile, median and 9th decile. Each column is coloured based on the performance of the settings with the worst performances in red and the best performance in green.

When we compare the results of the IGD metric and the HV metric the results are similar in most cases, this indicates that the algorithm does not have a large bias towards one of the two metrics. The VIRSIM does not have a known Pareto front which makes it impossible to compute the IGD metric for this problem. Figures 4.9 and 4.10 indicate that the HV metric is a good indication for the performance since the best HV scores often match the best IGD scores. Comparing the settings we see the 3rd selection method performing the worst with almost every problem achieving a worse result compared to selection methods 1 and 2.

The aggregated results from every selection setting (e.g. S1Nx) were compared with the use of the Mann-Whitney-Wilcoxon test. The pairwise comparison of the three selection settings over eight different problems results in 48 significance tests. To avoid false positives a Bonferroni correction is applied resulting in a significance value of  $0.05/48 = 0.0010$ . The results of these significance tests are shown in Table 4.3. Each test checked if the value of the aggregated selection is greater in the

Setting	ZDT1			ZDT2			ZDT3			ZDT4			ZDT6		
	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)
s1n2	0.04811	0.05418	0.06258	0.08029	0.41360	0.61852	0.10192	0.17407	0.25731	13.70935	24.73167	36.24438	1.98258	2.21895	2.84751
s1n5	0.02464	0.03022	0.04429	0.03831	0.09908	0.61608	0.09383	0.10900	0.26583	10.03331	26.92465	50.42925	2.25751	2.44252	2.63589
s1n15	0.02952	0.03842	0.04427	0.05099	0.06508	0.31988	0.11518	0.15752	0.24578	14.57213	29.32907	46.46635	1.87897	2.38453	3.30733
s2n2	0.09289	0.11846	0.12817	0.11275	0.13789	0.25553	0.16553	0.21987	0.31210	6.05755	25.26938	49.84028	2.60176	2.79493	3.04704
s2n5	0.05927	0.06706	0.08749	0.06655	0.11014	0.61431	0.15019	0.17565	0.24056	12.84416	35.10288	44.85750	1.93329	2.63781	3.45167
s2n15	0.05769	0.06929	0.08706	0.08540	0.11334	0.14737	0.10151	0.14966	0.19812	21.98235	30.15350	43.21747	2.09490	2.86881	3.15278
s3n2	0.19195	0.24638	0.30011	0.20669	0.33787	0.53605	0.37070	0.42206	0.54503	22.50259	37.96795	49.88894	2.71493	3.72682	4.22902
s3n5	0.10929	0.13519	0.16406	0.12518	0.29027	0.41414	0.22525	0.25400	0.31174	42.22491	50.84723	57.20683	2.96398	3.47992	4.26348
s3n15	0.10474	0.13563	0.19576	0.10887	0.16834	0.20990	0.16822	0.20102	0.25072	20.39329	46.40304	61.16764	3.33704	4.00563	4.27995

Setting	GENMED [-10,10]			GENMED [5, 10]			BD2		
	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)
s1n2	0.00911	0.01525	0.04107	0.02136	0.04811	2.47138	0.90997	1.08518	1.23838
s1n5	0.01212	0.03727	0.28930	0.01618	0.04120	0.11485	0.94400	0.95682	1.01236
s1n15	0.02550	0.06111	0.07914	0.03883	0.08648	0.68415	0.97161	1.03537	1.58464
s2n2	0.05968	0.30337	7.26908	0.29206	0.45852	1.53353	0.96100	1.23142	2.00449
s2n5	0.02493	0.11398	0.35762	0.02231	0.09618	0.49924	0.91079	0.98012	1.11962
s2n15	0.01897	0.07883	1.02576	0.17547	5.01041	22.03513	1.00580	1.21831	3.44079
s3n2	0.11845	0.15387	0.24840	0.20688	0.53531	2.93094	0.99692	1.06227	3.27807
s3n5	0.04523	0.07777	0.17378	0.07072	0.14758	1.13439	0.96615	1.08374	1.56428
s3n15	0.10403	0.13412	0.34720	0.25297	9.31410	21.94239	0.98452	1.24347	6.15565

Figure 4.10: IGD results of the different settings on all the benchmark problems. S1 stand for selection setting = 1 (adaptive selection), S2 is the fixed selection and S3 means no selection limits are enforced. The number after n indicated the number of neighbours. The results are highlighted with a comparative colour scale for each column. The table shows the top and bottom values of the interdecile range (IDR) in combination with the median.

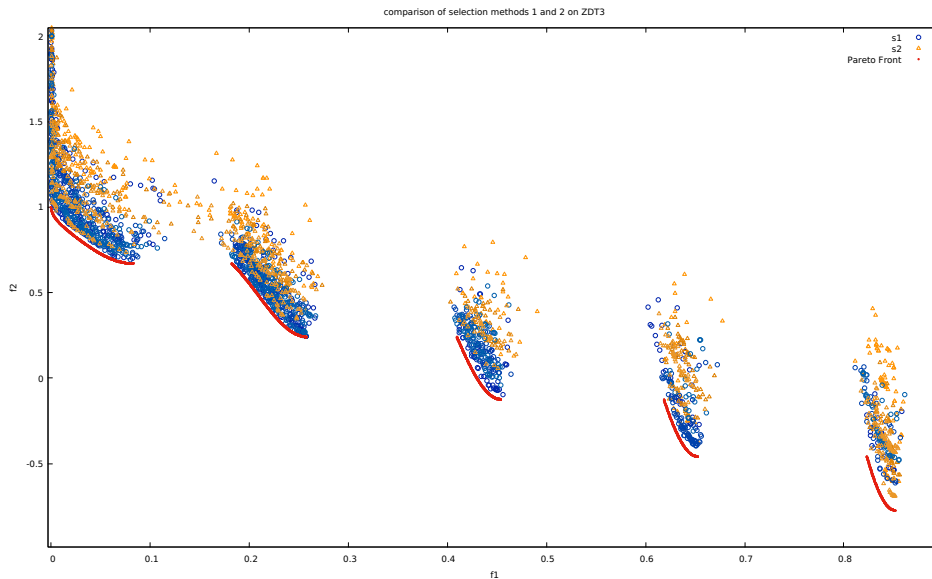
Comparison	Hypervolume			Inverted Generational Distance		
	S1 >S2	S1 >S3	S2 >S3	S1 <S2	S1 <S3	S2 <S3
ZDT1	<b>4.92E-08</b>	<b>4.96E-11</b>	<b>1.76E-07</b>	<b>4.05E-10</b>	<b>1.51E-11</b>	<b>1.46E-09</b>
ZDT2	0.7940	0.1556	<b>3.10E-04</b>	0.2601	0.0328	<b>1.99E-04</b>
ZDT3	0.9548	0.0667	<b>3.65E-04</b>	0.0512	<b>1.08E-06</b>	<b>7.15E-06</b>
ZDT4	0.1521	<b>7.65E-06</b>	<b>1.50E-04</b>	0.1521	<b>7.65E-06</b>	<b>1.59E-04</b>
ZDT6	<b>4.28E-04</b>	<b>4.88E-10</b>	<b>5.78E-08</b>	0.0016	<b>1.22E-09</b>	<b>7.37E-08</b>
genMED [-10, 10]	<b>2.80E-05</b>	<b>3.54E-08</b>	0.5206	<b>3.38E-05</b>	<b>3.54E-08</b>	0.5088
genMED [5,10]	<b>1.06E-05</b>	<b>1.34E-06</b>	0.3205	<b>2.16E-05</b>	<b>1.44E-06</b>	0.2410
BD2	0.0481	0.0012	0.1141	0.0425	0.0011	0.1170
VIRSIM	0.1290	-	-	-	-	-

Table 4.3: Significance results of comparing selection setting with Mann-Whitney-Wilcoxon test. Bonferroni correction is applied on the significance value of 0.05 resulting in a new significance value of 0.001. All significant results are shown in bold.

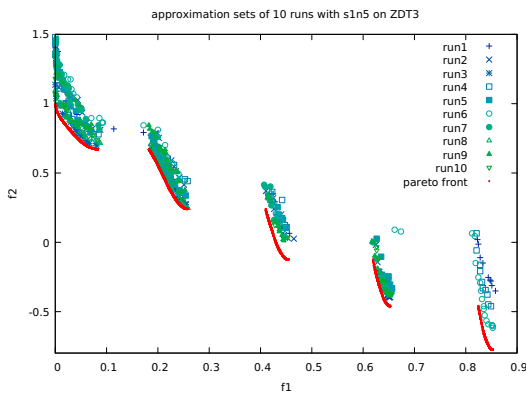
case of the HV metric or less in the case of the IGD metric. In almost all cases it is better to select one of the newly introduced selection methods (S1, S2) over the baseline (S3). The difference between selection methods 1 and 2 is more subtle: only in the ZDT1 problem and both the genMED problems S1 consistently outperforms S2.

An interesting case where the IGD and HV metrics contradict each other is ZDT3. When looking at the hypervolume metric the best performing selection method is the fixed selection range (S2) while if we use the IGD metric the adaptive selection method (S1) performs best. Figure 4.11a compares the S1 solutions with the S2 solutions. It clearly shows the adaptive selection converging more towards the Pareto front than fixed selection. Figure 4.11 shows the individual approximation fronts of all 10 runs of both S1N5 (Figure 4.11b) and S2N15 (Figure 4.11c) which respectively have the best IGD and HV scores on the ZDT3 problem. We see that S1N5 again converges more towards the Pareto front, but it only manages to discover the rightmost disconnected component 3 out of 10 times whereas S2N15 manages to discover it all 10 times. This indicates the trade-off which exists in the current selection methods, the adaptive selection converges faster but sometimes is too exploitative and thus unable to achieve a good spread. This trade-off between exploitation and exploration is a well-known trade-off which has to be balanced in EAs in general. This problem also occurs in the S1N2 version of ZDT2. Figure 4.14a shows adaptive selection again approaching the front more closely, however when comparing S1N2 (Figure 4.14b) with S1N15 (Figure 4.14c) we can see that the two neighbour variant has more clustered solutions in a single run which indicates a more local searching strategy. This time the trade-off is not with selection settings but with the amount of nearest neighbours used in

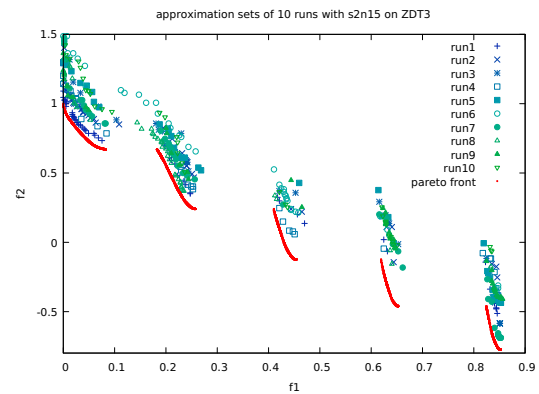




(a) Comparison of Adaptive selection method (S1) in blue and fixed selection method in orange on the ZDT3 problem. The adaptive selection approaches the Pareto front more than the fixed selection.



(b) S1N5: the best performing variant on IGD metric



(c) S2N15: the best performing variant on HV metric

Figure 4.11: Comparison of selection methods 1 and 2 on the ZDT3 problem. Overall and individual runs between setting S1N5 and S2N15. S1N5 converges closer to the Pareto front than S2N15 however it does not manage to find the 5th disconnected component only 3 out of 10 times.

the prediction of the objective values.

In the genMED problems, the adaptive selection performs significantly better than the other two approaches. The initial solutions are far away from the Pareto front thus the algorithm usually has to use the first few generations to more closely approach this. This is exactly the scenario in which the adaptive selection is targeted at. Being far away from the optimum values in both objectives results in the comet-shaped approximation set. Figure 4.12 shows the hypervolume over time for the three selection methods. This shows the advantage of adaptive selection with faster exploration in the early stages (<500 FE) and continuous exploitation in the later stages (>500FE). The fixed selection method has limited exploration which causes a slower convergence, whereas the baseline selection method (no selection criteria) explores fast initially. However, in the later stages, there does not exist enough focus on exploitation to push more toward the Pareto front. In the genMED problems, adaptive selection manages to balance the exploitation and exploration to achieve increased performance.

The number of neighbours used in the surrogate model does sometimes have a large influence on the performance (eg. ZDT2 S1N2 vs S1N15), however, there is no clear trend which indicates one number of neighbours is consistently better than another. The differences seem to be problem specific. Generally speaking, five neighbours seems to never be the worst choice, but the differences seem minimal.

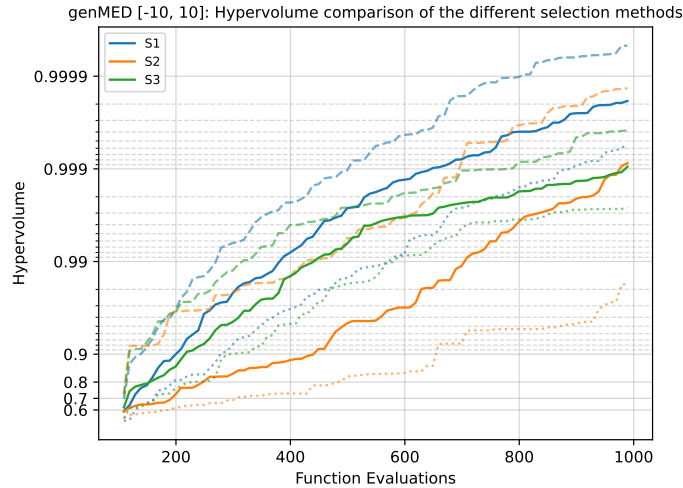
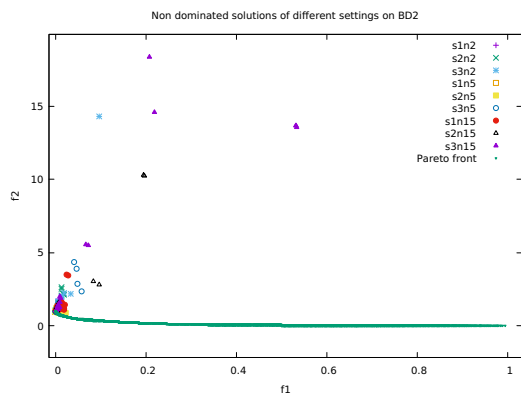
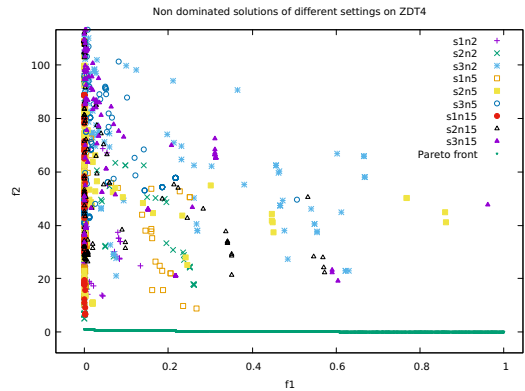


Figure 4.12: Comparison of the HV score between the adaptive selection method (orange), the fixed selection method (blue) and the no selection method (green). The interdecile range is shown with the dashed lines.



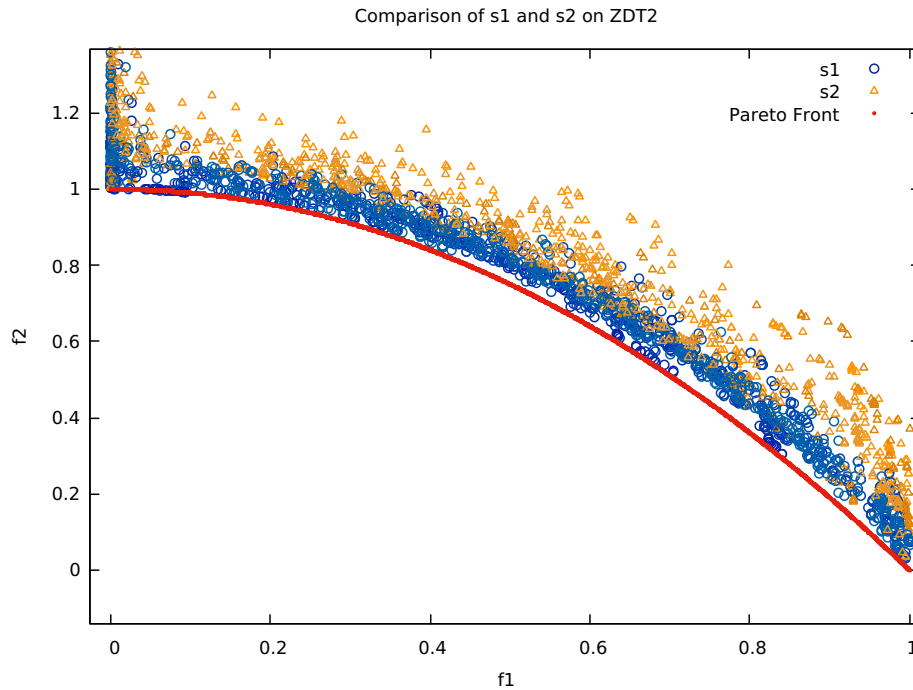
(a) Only objective f1 is optimized with most of the settings consistently finding the optimal solution where  $f_1 = 0$ . The EA was unable to find the optimal value in the second objective.



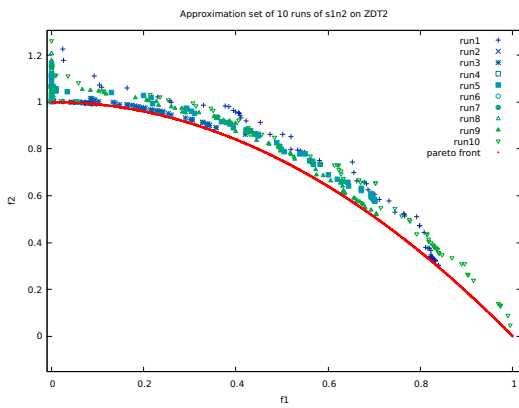
(b) No solution nearby the Pareto front has been found. The better performing settings do manage to achieve results which are optimal in objective 1 but do not manage to find the corresponding f2 objective value to achieve a Pareto front solution

Figure 4.13: All solutions of the BD2 4.13a and ZDT4 4.13b problem with different selection settings.

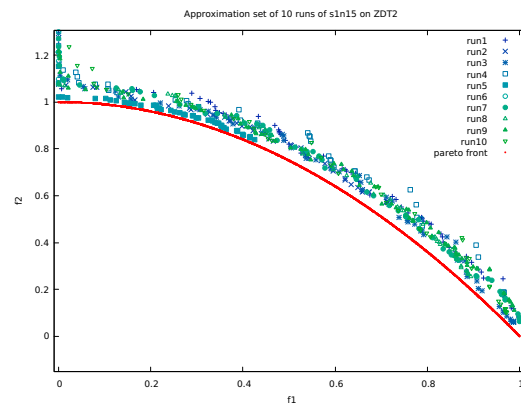
ZDT4 and BD2 are challenging problems which are not solved by the IDW-SAEA in the 1000 FE budget. In both cases, the IDW-SAEA only manages to optimize the first objective with some small explorations towards the second objective, but nowhere near the optimal value. The metrics do indicate a statistical difference between selection methods, however, Figure 4.13 demonstrates that this improvement isn't a better approximation front but just a few solutions which are closer to the Pareto front.



(a) Comparison of Adaptive selection method (S1) in blue and fixed selection method in orange on the ZDT2 problem. The adaptive selection approaches the Pareto front more than the fixed selection.



(b) S1N2: worst performing variant



(c) S1N15: best-performing variant

Figure 4.14: Comparison of the 10 individual runs between setting S1N2 and S1N15 on the ZDT2 problem. S1N2 is much more sparse than s1n15. The low amount of neighbours contributes to a more localized search which in the ZDT2 problem does not allow for a good exploration of the complete Pareto front.

## 4.4. Comparison state of the art on benchmarks

The comparison with the state-of-the-art algorithms is as follows: We compare IDW-SAEA with the best parameter settings to K-RVEA, MOEA/D-EGO, AB-SAEA and a normal version of MAMaLgAM using 4 clusters which was a good choice as shown in Section 4.2.2. The best IDW-SAEA variant is defined as the highest median HV score. HV was chosen over IGD because the VIRSIM does not have an IGD metric since the Pareto front is unknown.

The MOEA/D-EGO algorithm was not able to handle negative parameter ranges or large objective values. This would sometimes result in a negative mean value which subsequently was taken the square root of, resulting in imaginary numbers. To counteract this problem, when running MOEA/D-EGO on the genMED problem, the parameter range was scaled down to a range between 0 and 1. Before evaluation, the parameters would be multiplied by  $1e10$  to achieve the same ranges as the other algorithms. After evaluation, the outcome was divided by  $1e20$  to again reduce the objective values. Additionally, the k-means clustering distance was changed from  $1e - 5$  to  $1e - 15$  to account for the difference in parameter values.

	[10,10] ZDT1			[10,10] ZDT2			[10,10] ZDT3			[250,250] ZDT4			[10,10] ZDT6		
	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)
IDW-SAEA	0.99279	0.99499	0.99538	0.91304	0.97882	0.98669	0.87719	0.89151	0.90522	0.85372	0.89975	0.94389	0.68924	0.74827	0.76576
AB-SAEA	0.97574	0.98515	0.99448	0.99305	0.99317	0.99321	0.83572	0.83975	0.89917	0.83886	0.92520	0.95121	0.91946	0.94643	0.95670
K-RVEA	0.99613	0.99639	0.99651	0.99321	0.99322	0.99323	0.92179	0.92180	0.92182	0.81565	0.87712	0.91580	0.85799	0.88524	0.90703
MOEA/D-EGO	0.96583	0.97679	0.98674	0.92690	0.95901	0.97074	0.83200	0.85819	0.88335	0.65460	0.69738	0.73446	0.72108	0.79973	0.94244
MAMaLgAM	0.93428	0.95079	0.96469	0.85821	0.89668	0.92323	0.82026	0.83747	0.85763	0.79608	0.80909	0.86324	0.44707	0.48473	0.50149

	[90,90] GENMED [-10,10]			[230,230] GENMED [5, 10]			[30000,30000] BD2		
	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)
IDW-SAEA	0.99929	0.99983	0.99991	0.99913	0.99976	0.99989	0.99996	0.99997	0.99997
AB-SAEA	0.90277	0.95130	0.98523	0.14041	0.17141	0.19247	0.39161	0.99931	0.99995
K-RVEA	0.04971	0.13916	0.25497	0.00205	0.00737	0.02817	-	-	-
MOEA/D-EGO	0.13046	0.25540	0.50825	0.02264	0.05163	0.12516	0.88553	0.96558	0.98147
MAMaLgAM	0.99696	0.99859	0.99912	0.99562	0.99738	0.99867	0.99996	0.99996	0.99997

Figure 4.15: Hypervolume scores of all the algorithms on the benchmark problems. For every problem, the reference point is indicated left of the name. The significance value of the metric compared to IDW-SAEA is given after each problem. ZDT3 uses the point  $[-0.773368556, -0.773368556]$  as the origin since the second objective can achieve negative values.

	ZDT1			ZDT2			ZDT3			ZDT4			ZDT6		
	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)
IDW-SAEA	0.02464	0.03022	0.04429	0.05099	0.06508	0.31988	0.09383	0.10900	0.26583	13.70935	24.73167	36.24438	1.98258	2.21895	2.84751
AB-SAEA	0.00856	0.03561	0.09008	0.00498	0.00616	0.00830	0.03380	0.25542	0.28417	11.88240	18.37771	39.95631	0.09033	0.17334	0.42936
K-RVEA	0.00642	0.00685	0.00818	0.00772	0.01091	0.01256	0.01096	0.01228	0.01381	20.72259	30.39294	45.75456	0.54737	0.77603	1.15340
MOEA/D-EGO	0.05161	0.08683	0.17078	0.06303	0.08109	0.20791	0.22019	0.25945	0.42746	66.01739	75.27001	85.99327	0.14610	1.49941	2.36964
MAMaLgAM	0.29149	0.39712	0.46011	0.40359	0.58362	0.80437	0.41249	0.51885	0.59917	33.85138	47.40053	50.65259	4.68243	4.86336	5.24804

	GENMED [-10,10]			GENMED [5, 10]			BD2		
	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)	IDR (10)	median	IDR (90)
IDW-SAEA	0.00911	0.01525	0.04107	0.01618	0.04120	0.11485	0.94400	0.95682	1.01236
AB-SAEA	0.71282	2.92594	6.09934	182.11446	190.15992	202.97536	6.35018	18.49600	18245.682
K-RVEA	62.801	79.642	98.083	270.238	296.936	310.230	16805.662	35935.545	53245.599
MOEA/D-EGO	36.129	62.590	82.099	209.728	250.902	275.859	551.692	1027.681	3427.877
MAMaLgAM	0.08136	0.11979	0.20413	0.29479	0.33439	0.74009	0.86804	0.97831	2.34678

Figure 4.16: IGD of all the algorithms on the benchmark problems. Significance is indicated after each problem with (+) for better, (-) for worse and (=) for equal.

### ZDT

The best IDW version is compared to the state of the art algorithms and 4 cluster Univariate variant of MAMaLgAM in Figures 4.15 and 4.16. These figures show a clear pattern in the ZDT problems. K-RVEA and AB-SAEA perform very well. MOEA/D-EGO performs worse. Except for ZDT4, MAMaLgAM consistently performs worse than all expensive algorithms. IDW-SAEA competes relatively well, on the ZDT problems it achieves one statistically better performance, two equal performances and two problems where IDW-SAEA performs worse. IDW-SAEA statistically performs equal to or better than MOEA/D-EGO. K-RVEA is outperforming IDW-SAEA, with only the ZDT4 problem having equal performance and all other ZDT problems having a worse performance when using IDW-SAEA. Looking at the IGD values of the ZDT4 problem we discover that no algorithm managed to come close to the Pareto front with the lowest median IGD of 18.377. Therefore we can conclude the solutions for this

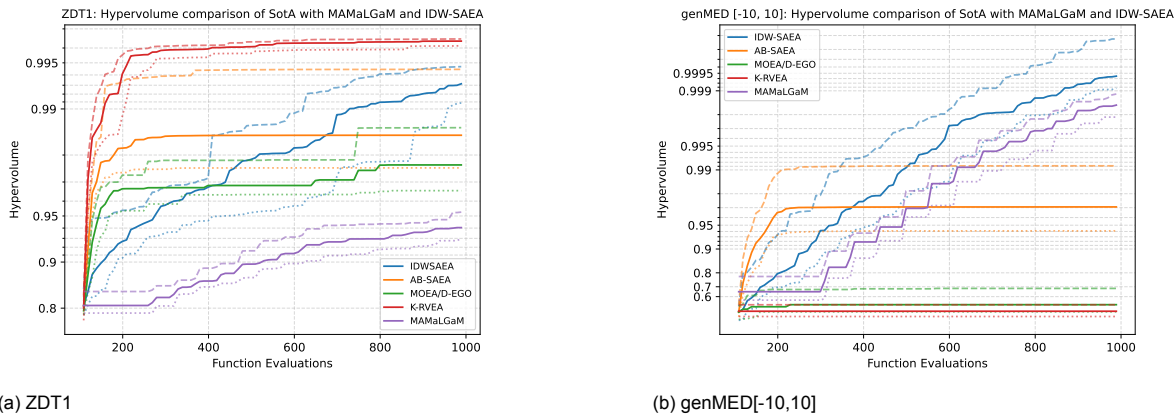


Figure 4.17: Hypervolumes over time on two Benchmark problems

problem are still very far away from the Pareto front because of the difficulty, making it a challenging problem to solve in 1000 FE.

Figure 4.17a shows the hypervolume value over time of all the algorithms on the ZDT1 problem. It shows the three SotA algorithms flatten out to a certain degree when passing the 300 evaluation mark whereas MAMaLGaM and IDW continuously improve over the complete function evaluations. These last two algorithms do have a slower start. However, IDW-SAEA overtakes MOEA/D-EGO at around 400 FE and AB-SAEA at around 700FE. This is an indication that with additional function evaluations the IDW-SAEA can continue to improve. This behaviour is replicated on all the ZDT problems which can be found in the Appendix.

### GenMED and BD2

The performance of IDW-SAEA is significantly better on the genMED and BD2 problems than the state-of-the-art algorithms. Firstly we see that MOEA/D-EGO and K-RVEA are unable to handle the large parameter search space of these problems. Both these two algorithms start by exploring the edge of the large search space and from there slowly improve their solutions, but they do not successfully use the initial samples to quickly point them in the direction of the Pareto front. This results in no improvements after the first few FE. Figure 4.17b shows AB-SAEA and MOEA/D-EGO failing to improve after 200 FE and K-RVEA failing to improve at all. IDW-SAEA and MAMaLGaM do continuously improve over time and significantly outperform the SotA. Comparing IDW and MAMaLGaM we can conclude that the IDW-SAEA starts earlier than MAMaLGaM but both algorithms improve similarly based on the FE. This late start of MAMaLGaM is caused by the algorithm having to initialize its subgenerations before it can start optimizing. One would expect it to start improving faster than IDW-SAEA once these subgenerations are initialized, but this is not the case. Similar behaviour is shown on genMED[5,10] (Figure A.2 in the Appendix).

The BD2 problem is very challenging. None of the algorithms managed to make any substantial progress in the difficult second objective. IDW-SAEA and MAMaLGaM did manage to discover the easy end of the Pareto front. Out of the SotA algorithms, AB-SAEA performed best by getting into the general vicinity of the Pareto front, but it did not consistently reach the front. This can be seen by looking at the difference between the IDR in Figure 4.16. This Figure shows a median IGD of around 18.5 for AB-SAEA with IDR values around 6.35 and 18245.68 indicating a very high variance in solutions. K-RVEA did not make any progress in solving BD2. After initialization, the algorithm became stuck in a very small area of the search space. This was because of the small relative values between the data points making the cosine metric effectively zero. MOEA/D-EGO did achieve progress in the correct direction, but with a median IGD of 1027.68, the front is comparatively far away. The performance was more consistent than AB-SAEA with much lower variance in the IDR, 551.69 and 3427.88 as 10th and 90th percentile.

## 4.5. Virtual phantom simulation

The runtime of a single evaluation with the Virtual Phantom simulation takes between 15 and 25 minutes. Due to time constraints, only the most promising settings were run instead of all the possible combinations of the IDW version. This resulted in not running the third selection setting: No selection. The runs were all ran for 1000 evaluations similarly to the benchmark problems and each setting was run 10 times to get a measure of the variance between runs. The IDW results are compared with the pure MAMaLGaM results and K-RVEA. MAMaLGaM uses the same parameters as described in section 4.2.2: Univariate factorization with 4 clusters and guideline parameters otherwise. The hypervolume calculation uses reference point [1,1].

### 4.5.1. Comparison of IDW-SAEA settings

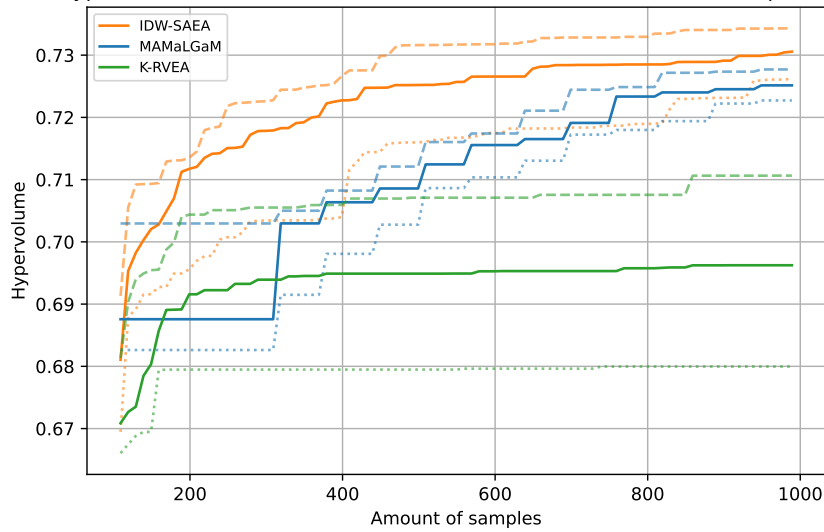
In Figure 4.9 we see the different versions of the IDW-SAEA on the VIRSIM problem. It can be seen that the S1N2 setting performs the best, but with a p-value of 0.0606 compared to S2N2 we can not consider this result significant. Nevertheless, we do see that the settings with a large number of neighbours do not perform very well. This is because of an interesting property of the Virtual Phantom Simulation: this problem is a FEM simulation thus it can crash. In the current design, these crashing runs get a dice score of 0, the lowest possible value resulting in an objective score of 1. When a large number of neighbours are utilized during the prediction, the possibility of one or multiple of these neighbours being a crashing run increases. The addition of such a crashing run in the weighting will heavily skew the objective score towards an objective score of 1 for both objectives. This could be an advantage to staying away from the crashing parameter setting, but it can also be a disadvantage if the best settings are nearby crashing simulations. These experiments indicate that it is more likely to be of disadvantage.

### 4.5.2. Comparison between IDW-SAEA, MAMaLGaM and K-RVEA

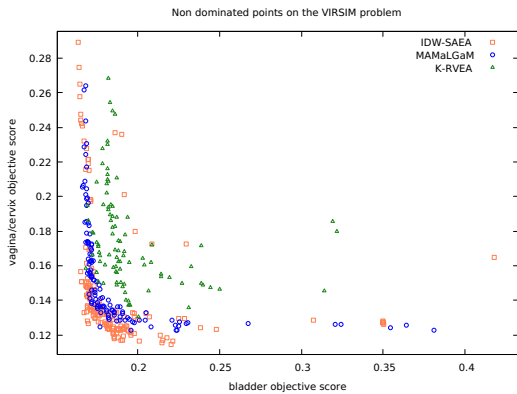
Figure 4.18a shows the comparison of IDW-SAEA (S1N2) with MAMaLGaM and K-RVEA. We see the IDW-SAEA achieving higher hypervolume than the other algorithms: p-value = 0.03783 compared to MAMaLGaM, p-value = 0.0006574 compared to K-RVEA. Looking at the behaviour of IDW-SAEA it starts with quick improvements early in the optimization process and continues to gradually improve. MAMaLGaM has a slow startup time and then quickly starts to catch up to IDW-SAEA. It achieves a better HV score than K-RVEA after the first 300 FE, which coincides with the first generation. Instead, K-RVEA has some early improvement until the 300FE and then stops with the improvement. This is similar behaviour to the benchmark problems where after some quick improvement K-RVEA optimization slowed down substantially. In contrast to the benchmarks, however, the solutions found in the first 300 FE are not as good.

The final non-dominated points can be seen in figure 4.18b. We see IDW-SAEA resulting in more solutions closer to the origin compared to the other algorithms while keeping a good spread with the best value in either objective also belonging to the IDW method. Interestingly the approximation front does not seem to contain the complete trade-off between the objectives: there are no solutions which are optimized entirely on one objective disregarding the score of the other objective. Figure 4.18c shows a more zoomed view of the front with an additional plot of the non-dominated solutions of the best MAMaLGaM run with 12000+ FE. This indicates that the IDW-SAEA approaches the possible Pareto front quite closely. One advantage of MAMaLGaM over IDW-SAEA is a smaller variance between runs. Figure 4.18b shows one run from IDW-SAEA where the solutions are worse than K-RVEA, this indicates that although the IDW method on average outperforms, it is not always capable of finding the best solutions in the first 1000 FE.

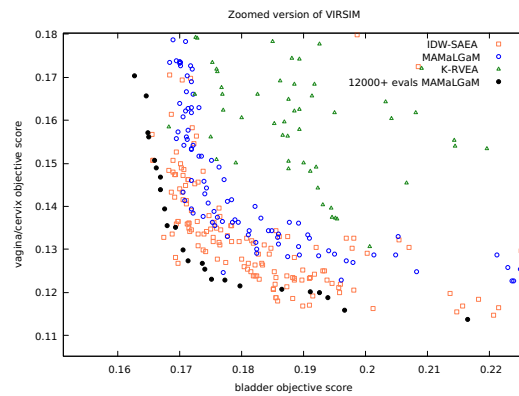
Hypervolume of MAMaLGA<sub>M</sub>, IDW-SAEA and K-RVEA on the VIRSIM problem



(a) Hypervolume comparison of the best settings of IDW-SAEA, K-RVEA and MAMaLGA<sub>M</sub>



(b) Aggregated non-dominated solutions of 10 runs of MAMaLGA<sub>M</sub>, K-RVEA and the best performing IDW (s1n2) settings.



(c) Aggregated non-dominated solutions of 10 runs of MAMaLGA<sub>M</sub>, K-RVEA and the best performing IDW (s1n2) settings. The approximation set of a single MAMaLGA<sub>M</sub> run with 12000+ FE is also displayed in black.

Figure 4.18: Comparison of IDW, K-RVEA and MAMaLGA<sub>M</sub> on the Virtual Phantom Simulation





# 5

## Discussion

In this section, we first analyze the overall performance of the algorithms and secondly give the weaknesses with potential improvement for the IDW Algorithm. Last some limitations to the study are discussed.

### 5.1. Performance Analysis

The results show that the IDW adaptation of MAMaLgAM does succeed in improving the algorithm to perform better with 1000 FE compared to the standard MAMaLgAM version. However, we also see that the state of the art is much better in quickly increasing the quality of its solutions in the ZDT problems. The SotA all are designed with a smaller limit of function evaluations, in their papers [57, 14, 54] the experiments only go up to 300 function evaluation maximum. We see this is reflected in their performance: The algorithms have a quick improvement in the first 400 FE. After these early evaluations, the improvement slows down considerably.

This directly relates to one of the strengths of the IDW-SAEA method. It continues to improve over time, albeit slower. Because of this, it manages to catch up on MOEA/D-EGO in many of the benchmark problems between the 500 and 800 function evaluations. The expectation is that if more function evaluations are allowed IDW-SAEA will continue to improve and eventually outperform the expensive SotA since these algorithms seem to stagnate. Eventually, non-expensive evolutionary algorithms such as MAMaLgAM will outperform all the SAEAs. The number of FE where this occurs will of course differ between problems. In the 0 to 1000 FE the IDW algorithm outperforms MAMaLgAM on every problem.

Overall comparing IDW-SAEA to SotA has different results highly dependent on the test problem. IDW-SAEA performs much better in the genMED, BD2 and the Virtual Simulation problems, whereas in the ZDT problems it only consistently manages to outperform MOEA/D-EGO. In every problem, it outperforms our non-expensive baseline EA: MAMaLgAM.

GenMED and BD2 are problems where the SotA performs especially poor. While the genMED problems in themselves are not particularly difficult, the chosen initialization and large parameter space seem to be very challenging for the SotA algorithms. Only AB-SAEA manages to gain real improvements compared to the initialization values, and even then these solutions are not on or near the Pareto front. Again after 300 FE it stops improving and gets stuck on its current location with only very small variations. In the genMED and BD2 problems, the IDW-SAEA performs better because of its robustness in exploration. In every iteration, the algorithm selects a few solutions with promising problem objectives and a few with promising exploration objectives. By doing this the algorithm will always have a few candidate solutions which are focused on exploitation and not waste all its evaluations exploring unknown regions in the search space like K-RVEA and MOEA/D-EGO. The only exception here is when the adaptive selection is in the exploration setting (low rank-density), however, in this scenario exploration is required.

Another reason why the IDW algorithm is relatively robust is because of the large amount of randomness still included in the algorithm. The estimation of objective scores is quite inaccurate when extrapolating. This results in a large set of individuals who have similar predicted objective scores but vary a decent amount. This incidentally adds to exploration and reduces the chance of the algorithm

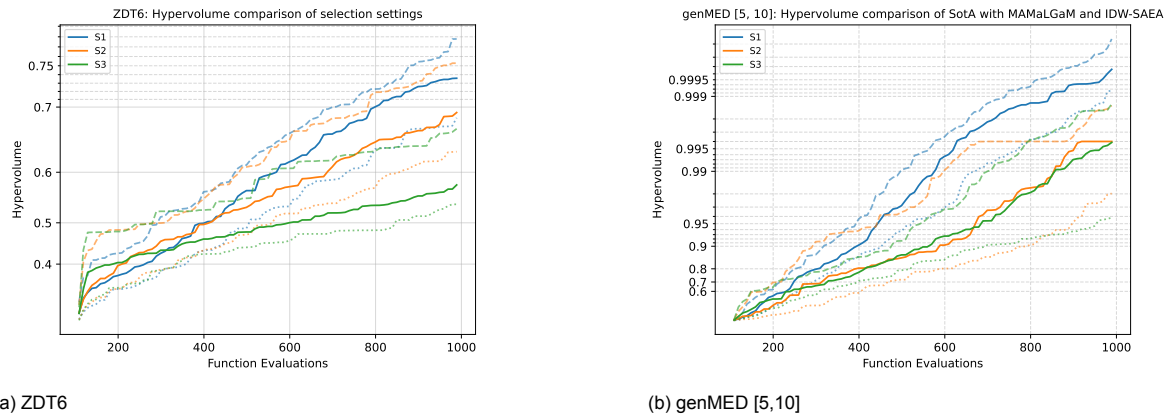


Figure 5.1: Comparison of the different selection settings on two Benchmark problems

achieving premature convergence. A disadvantage of this is of course the inability to directly make large progress at the start which is something the SotA is capable of doing.

Overall IDW-SAEA performs very well on problems which have a large parameter range and where the budget of FE is between 400 and 1000 FE. It requires a few function evaluations to get started because it cannot directly extrapolate to the Pareto front, however once started it does continue to improve. Future work would have to investigate how many FE IDW-SAEA outperforms non-expensive EAs since the expectation is that eventually, these non-expensive EA will outperform the expensive versions given enough FE.

The performance of IDW-SAEA on the Virtual Phantom simulation is good. One could argue that the performance of K-RVEA is subpar since on the ZDT problem it outperformed MAMaLGA. The genMED problems had the difficulty of extremely large parameter space but in the VIRSIM this range is not as large with the largest being the bladder filling between 0 and 600. Comparing the HV over time of K-RVEA in Figure 4.18a and Figure 4.17a, we do see similar functioning with the fast increase in first 300 FE and then a substantial reduction of the improvement. However, the early results are not as good as in the ZDT problems indicating that the VIRSIM problem is more challenging. Overall IDW-SAEA outperforms both K-RVEA and MAMaLGA.

Selection method 3 is consistently the worst-performing selection method. With this selection method, the Pareto front improves more slowly than with the other two methods. Since selection method 3 does not put any restrictions on the candidate solutions many of the solutions selected will be very far away from the 2D front. This is made worse by the fact that finding solutions with a good exploration objective score (far away from other solutions) is much easier than a solution which is predicted to have a good objective score. Thus not putting any restrictions on the selection method results in the exploration objective dominating the solutions selected by this method. This is not desirable since the exploration objective is a tool for exploration and not an intrinsic goal to optimize. This large amount of exploration does allow method 3 to sometimes get a head start compared to the other selection methods, since it takes more risks by selecting individuals with a good exploration it can make larger progress if it is lucky, but as shown this strategy is not sustainable. Figure 5.1a illustrates this behavior clearly.

Selection methods 1 and 2 are very comparable in performance on the ZDT problems and it is unclear to decide which method is the better performing version. Selection method 1 seems to have a slight edge but it is often not significant as seen in Figure 4.3. In the genMED and BD2 case selection method 1 is superior. The method allows for a quick early increase by selecting ranks with good exploration scores and also can continue exploitation once the front has been reached. Figure 5.1b shows this performance on genMED[5,10].

Overall the Adaptive selection method seems to be the best and a necessary introduction to select promising candidate solutions when using IDW in combination with the exploration objective. Even when introducing multiple mechanisms (keep best known solutions in population, calculate IDW with k-nearest-neighbours, filtering of solutions which are too similar) to help the optimization of the surrogate model, the selection method still makes a large difference in performance.

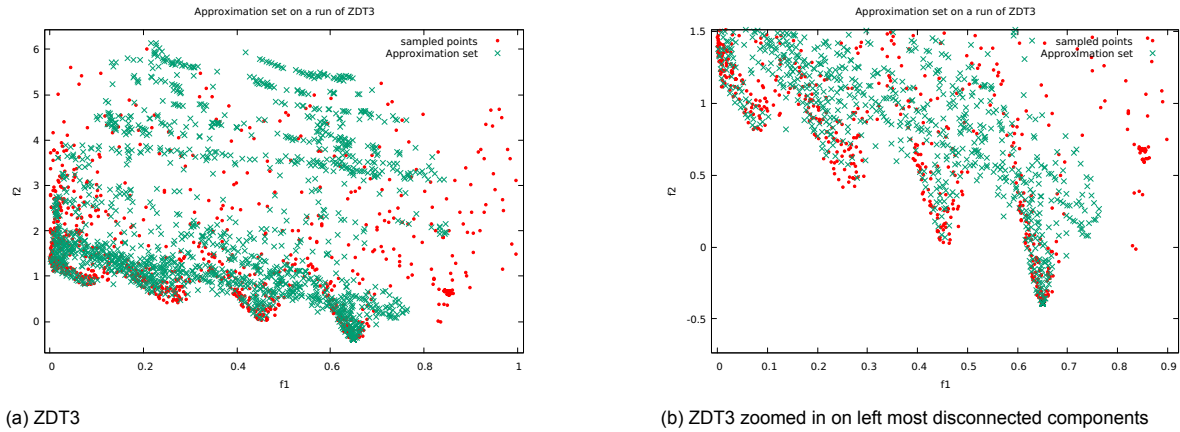


Figure 5.2: Approximation set during optimization of ZDT3. Figure 5.2b shows a sparse amount of solutions discovered in the second and third disconnected front. This indicates that MAMaLGaM inside of IDW-SAEA can not always consistently solve the surrogate model. Figure 5.2a shows a large number of the approximation set far away from the problem from around the  $f_2$  objective score of 4 and higher. This is a wasted effort in the optimization process since these samples will never be selected in this stadium.

## 5.2. Potential Improvements

Some possible improvements to the algorithm are discussed. First, some smaller changes are discussed. Followed by IDW in combination with a regression model and a problem-specific idea for the Virtual Phantom simulation is discussed.

### 5.2.1. Exploration objectives as tool

The performance of the IDW algorithm can still be improved. There are two main areas where it can be improved: the generation of candidate solutions and the selection of these solutions. Improving the candidate generation automatically improves selection, if all candidate solutions are viable to select there is no need for complicated selection strategies. The main problem currently with the generation of candidate solutions is the introduced exploration objective. This objective is required to promote exploration of the search space, however as already mention in the section above: there is currently too much importance on this objective resulting in situations like Figure 5.2a. We see a large number of individuals around the  $f_2$  objective score of 4 and higher, which will never be included in the final selection in this stage of the optimization process. When there are this many individuals far away from the front the performance of MAMaLGaM degrades: a large part of the computational power is invested in optimizing the 'useless' individuals, when this effort could be put into creating better solutions near the front. We want the final individuals to consist of many solutions with promising 2D objective scores with a varying degree of exploration and not be full of individuals which only exist because of their excellent exploration without having at least somewhat promising problem objective scores. This degradation at the problem objective can be seen in Figure 5.2b where the second and third components are only sparsely filled with the approximation set. Preferably every disconnected component would be densely filled like the fourth one.

There are some idea's still left to explore on how to achieve this result. Firstly to change the initialization of MAMaLGaM, because current random MAMaLGaM initialization does not utilize the knowledge of the current front in a very efficient way. Only the best solutions found so far are included, the rest is randomly initialized. Initializing all or more of the solutions based on the currently known samples we can avoid the effort of rediscovering the surrogate front every iteration. Allowing the algorithm to use this front as a starting point, it can immediately begin with exploring that specific area. The expectation is that this will result in a higher density of solutions near the front which are still optimized based on the exploration objective and additionally have a promising problem objective score. In short, we approach the problem from a different direction: Instead of finding the front from random solutions, we find faraway solutions starting from the front. There is an increased risk of premature convergence when using the approach, so the precise initialization procedure has to be designed with that in mind.

The second idea is to utilize Adaptive Steering [2]. This procedure reduces the available search

space by introducing constraints. Then utilizing constraint domination [18] the algorithm will avoid these areas and directly focus on exploring the preferred space. The constraints would have to be gradually introduced during the optimization and thoughtfully tuned to make sure they do not inhibit all forms of exploration. An example of a reasonable constraint for Figure 5.2a could be: only allow solutions where the  $f_2$  objective value is lower than 3. This would force all the solutions in the direction of the front leaving some space for exploration. Which setting would leave enough exploration space would have to be empirically tested, however since the current selection methodology already disregards all the solutions which are not in the first 30% ranks constraining these solutions should not be a problem, given the selection methodology would be adapted to this constraint. If adaptive steering would be successful there is no need for a selection strategy since all solutions are viable candidates. In the case of a low rank density (which occurs at the start of the genMED problems), the adaptive steering method should not be utilized. The experiments have shown that selecting very explorative samples is the best strategy in this case. Therefore adaptive steering should only begin when the optimization seems to slow down.

Another simple way to reduce the complexity of the surrogate model is to remove evaluated samples. During the research of this thesis, preliminary experiments have been done with removing samples: Keeping only the best 100 samples and trimming every other sample based on objective score. When trimming was used there was no improvement in regards to solving the surrogate model. Keeping the best 100 samples did allow for a better approximation set fit to the surrogate model: many solutions were found nearby the best samples so far. However, this filtering had a large drawback: The predicted objective scores were not accurate. This was caused by the loss of information by removing samples, i.e., individuals who had been evaluated with bad fitness before were not kept and the new 'unexplored' regions were seen as promising by MAMaLGaM.

### 5.2.2. Inverse distance weighted regression

Another area of weakness is the weak extrapolation with IDW. The best objective scores possible are those which have already been discovered. In [21] Emmendorfer and Dimuro introduce inverse distance weighted regression (IDWR): a combination between inverse distance weighting and weighted linear regression. This function could increase the performance by allowing the algorithm to extrapolate and therefore move in the direction of the Pareto front faster. If this would result in the same behaviour as the SotA (quick improvement and flattening out fast) has to be discovered empirically. Additionally, Emmendorfer and Dimuro have only shown an improvement in performance on two-dimensional problems. It is unknown how the formula performs with 10 decision variables.

### 5.2.3. Future work in Virtual Phantom optimization

Being a FEM simulation, the virtual phantom has the possibility of crashes and the possibility of organs intersecting after simulating, resulting in non-feasible solutions. When a crash or intersection occurs, it is impossible to assign a fitness score to it. Currently, the worst possible objective score of [1,1] is assigned. A bad score does drive the population away from these crashing parameters. However, when one of these [1,1] samples is included as a neighbour in the IDW, it heavily skews the predicted objective score. Deleting these samples is not a good idea, since in that case the crashing runs would be rediscovered and reevaluated again. This would cost valuable function evaluations.

Therefore a good alternative to experiment with would be to instead of using the [1,1] score when a crashing sample is included as a neighbour of interpolation, the crashing score is replaced in the interpolation weighting. We still want to include the information of this crashing sample, thus the surrogate prediction of this sample is used and the resulting predicted score is increased with the use of a crash penalty (for example +0.2 objective score). Doing this will reduce the negative effect when this solution is included as a neighbour while still making it a non-desirable location to be. Additionally, this would smooth the objective space of the Virtual Phantom Simulation allowing for easier optimization.

## 5.3. Limitations

There are a few limitations to this study. The first limitation is the relatively small number of runs for each test problem. Because the runtime of IDW-SAEA is currently not very optimized, the experiments took a long time to be executed. A single 1000 FE run on a benchmark problem took on average 1 hour, with a single VIRSIM run taking on average about 30 hours. On benchmark problems, most

of the time was spent computing the nearest neighbours for the surrogate and the covariance matrix of MAMaLGaM. Especially nearest neighbour computation could still be sped up by utilizing a more efficient implementation than repeated linear search. Because of the long runtimes in combination with the 9 different combinations of settings, the choice was made to only run every experiment for 10 runs. Although this often resulted in significant enough differences, more runs would give an even clearer indication of the difference in performance.

The second limitation is the test problems. The ZDT problems are quite diverse and with the addition of genMED and BD2 most types of two objective problems have been investigated, but it is still a limited amount. Especially the Virtual Simulation is very limited since only a single dataset from one patient was used in this experiment. More investigation with more datasets would be needed to more accurately determine the performance. The parameters of the adaptive and fixed selection methods would also benefit from more test problems and more thorough testing.

A third limitation is a difference in underlying algorithms when comparing Kriging to IDW. AB-SAEA and K-RVEA both use the RVEA algorithm to optimize the surrogate model, whereas IDW-SAEA uses MAMaLGaM. Although in theory, the underlying algorithm should be interchangeable, this should be investigated in the future to know for sure if the IDW with exploration objective is a good alternative to Kriging.



# 6

## Conclusion

In this work, we have shown the design of an evolutionary algorithm for expensive optimization, using a surrogate based on inverse distance weighting interpolation, called IDW-SAEA. The design introduced a third exploration objective which was used to stimulate the discovery of new solutions. The original problem objectives, for which a surrogate was estimated using IDW interpolation, and the exploration objective were optimized simultaneously. From the resulting solutions, a few were selected using the introduced adaptive selection methodology based on a new metric called rank density. The results of this algorithm were compared with three expensive SAEAs: AB-SAEA, K-RVEA and MOEA/D-EGO. IDW-SAEA was shown to be competitive with the SotA SAEAs, achieving slightly worse hypervolume and IGD scores in the ZDT test problems, but performing significantly better than the SotA in the genMED, BD2 and VIRSIM problems. IDW-SAEA starts slower than SotA SAEAs but continues improving when the other SA algorithms stagnate. The IDW-SAEA consistently outperforms non-expensive EA MAMaLGaM on every problem in the first 1000 function evaluations. With regards to the Virtual Phantom Simulation, IDW-SAEA manages to optimize this problem better than MAMaLGaM and K-RVEA. These results open the door to more experimenting on SAEAs with an separate objective dedicated towards exploration.





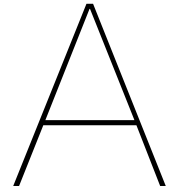
# Bibliography

- [1] Taimoor Akhtar and Christine A. Shoemaker. “Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection”. In: *Journal of Global Optimization* 64.1 (Jan. 2016), pp. 17–32. ISSN: 15732916. DOI: 10.1007/s10898-015-0270-y. URL: <https://link-springer-com.tudelft.idm.oclc.org/article/10.1007/s10898-015-0270-y>.
- [2] Tanja Alderliesten, Peter A. N. Bosman, and Arjan Bel. “Getting the most out of additional guidance information in deformable image registration by leveraging multi-objective optimization”. In: *Medical Imaging 2015: Image Processing*. Vol. 9413. SPIE, Mar. 2015, 94131R. ISBN: 9781628415032. DOI: 10.1117/12.2081438.
- [3] Maumita Bhattacharya. “Evolutionary Approaches to Expensive Optimisation”. In: *International Journal of Advanced Research in Artificial Intelligence* 2.3 (2013). ISSN: 21654050. DOI: 10.14569/ijarai.2013.020308. URL: [www.ijarai.thesai.org](http://www.ijarai.thesai.org).
- [4] Peter A N Bosman. “The Anticipated Mean Shift and Cluster Registration in Mixture-based EDAs for Multi-Objective Optimization”. In: *Proceedings of the 12th annual conference on Genetic and evolutionary computation - GECCO '10*. New York, New York, USA: ACM Press. ISBN: 9781450300728. DOI: 10.1145/1830483.
- [5] Peter A.N. Bosman. “On gradients and hybrid evolutionary algorithms for real-valued multiobjective optimization”. In: *IEEE Transactions on Evolutionary Computation* 16.1 (Feb. 2012), pp. 51–69. ISSN: 1089778X. DOI: 10.1109/TEVC.2010.2051445.
- [6] Peter A.N. Bosman and Marcus Gallagher. “The importance of implementation details and parameter settings in black-box optimization: a case study on Gaussian estimation-of-distribution algorithms and circles-in-a-square packing problems”. In: *Soft Computing* 22.4 (2018), pp. 1209–1223. ISSN: 14337479. DOI: 10.1007/s00500-016-2408-3.
- [7] Peter A.N. Bosman, Jörn Grahl, and Dirk Thierens. “Benchmarking parameter-free amalgam on functions with and without noise”. In: *Evolutionary Computation* 21.3 (Sept. 2013), pp. 455–469. ISSN: 10636560. DOI: 10.1162/EVCO\_A\_00094. URL: [https://dl-acm-org.tudelft.idm.oclc.org/doi/abs/10.1162/EVCO\\_a\\_00094](https://dl-acm-org.tudelft.idm.oclc.org/doi/abs/10.1162/EVCO_a_00094).
- [8] Peter A.N. Bosman and Dirk Thierens. “Adaptive variance scaling in continuous multi-objective estimation-of-distribution algorithms”. In: *Proceedings of GECCO 2007: Genetic and Evolutionary Computation Conference*. 2007, pp. 500–507. ISBN: 1595936971. DOI: 10.1145/1276958.1277067.
- [9] Peter A.N. Bosman and Dirk Thierens. “Expanding from discrete to continuous estimation of distribution algorithms: The IDEA”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 1917. 2000, pp. 767–776. ISBN: 9783540410560. DOI: 10.1007/3-540-45356-3\_75.
- [10] Peter A.N. Bosman and Dirk Thierens. “The balance between proximity and diversity in multiobjective evolutionary algorithms”. In: *IEEE Transactions on Evolutionary Computation* 7.2 (2003), pp. 174–188. ISSN: 1089778X. DOI: 10.1109/TEVC.2003.810761.
- [11] Anton Bouter et al. “Spatial redistribution of irregularly-spaced pareto fronts for more intuitive navigation and solution selection”. In: *GECCO 2017 - Proceedings of the Genetic and Evolutionary Computation Conference Companion*. Vol. 8. Association for Computing Machinery, Inc, July 2017, pp. 1697–1704. ISBN: 9781450349390. DOI: 10.1145/3067695.3082555.
- [12] Freddie Bray et al. “Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries”. In: *CA: A Cancer Journal for Clinicians* 68.6 (Nov. 2018), pp. 394–424. ISSN: 1542-4863. DOI: 10.3322/CAAC.21492.

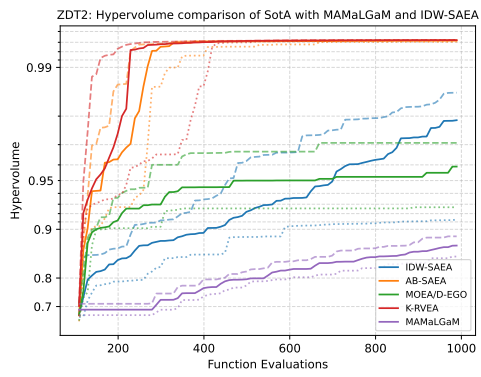
- [13] Ran Cheng et al. "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization". In: *IEEE Transactions on Evolutionary Computation* 20.5 (Oct. 2016), pp. 773–791. ISSN: 1089778X. DOI: 10.1109/TEVC.2016.2519378.
- [14] Tinkle Chugh et al. "A Surrogate-Assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-Objective Optimization". In: *IEEE Transactions on Evolutionary Computation* 22.1 (Feb. 2018), pp. 129–142. ISSN: 1089778X. DOI: 10.1109/TEVC.2016.2622301.
- [15] Tinkle Chugh et al. "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms". In: *Soft Computing* 23.9 (May 2019), pp. 3137–3166. DOI: 10.1007/s00500-017-2965-0.
- [16] Carlos A. Coello Coello and Margarita Reyes Sierra. "A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm". In: *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*. Vol. 2972. 2004, pp. 688–697. DOI: 10.1007/978-3-540-24694-7\_71.
- [17] Noel Cressie. "The origins of kriging". In: *Mathematical Geology* 22.3 (Apr. 1990), pp. 239–252. ISSN: 08828121. DOI: 10.1007/BF00889887. URL: <https://link-springer-com.tudelft.idm.oclc.org/article/10.1007/BF00889887>.
- [18] Kalyanmoy Deb et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE Transactions on Evolutionary Computation* 6.2 (Apr. 2002), pp. 182–197. ISSN: 1089778X. DOI: 10.1109/4235.996017.
- [19] Alan Díaz-Manríquez et al. *A review of surrogate assisted multiobjective evolutionary algorithms*. 2016. DOI: 10.1155/2016/9420460.
- [20] Lee R. Dice. "Measures of the Amount of Ecologic Association Between Species". In: *Ecology* 26.3 (July 1945), pp. 297–302. DOI: 10.2307/1932409.
- [21] Leonardo Ramos Emmendorfer and Graçaliz Pereira Dimuro. "A novel formulation for inverse distance weighting from weighted linear regression". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 12138 LNCS. Nature Publishing Group, 2020, pp. 576–589. ISBN: 9783030504168. DOI: 10.1007/978-3-030-50417-5\_43.
- [22] Michael T.M. Emmerich, Kyriakos C. Giannakoglou, and Boris Naujoks. "Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodells". In: *IEEE Transactions on Evolutionary Computation* 10.4 (Aug. 2006), pp. 421–439. ISSN: 1089778X. DOI: 10.1109/TEVC.2005.859463.
- [23] M. Gentile, F. Courbin, and G. Meylan. "Interpolating point spread function anisotropy". In: *Astronomy and Astrophysics* 549 (Oct. 2012). ISSN: 14320746. DOI: 10.1051/0004-6361/201219739. arXiv: 1210.2696. URL: <http://arxiv.org/abs/1210.2696> <http://dx.doi.org/10.1051/0004-6361/201219739>.
- [24] Ahsanul Habib et al. "A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization". In: *IEEE Transactions on Evolutionary Computation* 23.6 (Dec. 2019), pp. 1000–1014. ISSN: 19410026. DOI: 10.1109/TEVC.2019.2899030.
- [25] Nikolaus Hansen and Andreas Ostermeier. "Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation". In: *Proceedings of IEEE international conference on evolutionary computation*. IEEE, 1996, pp. 312–317.
- [26] William Grant Hatcher and Wei Yu. "A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends". In: *IEEE Access* 6 (2018), pp. 24411–24432. DOI: 10.1109/ACCESS.2018.2830661.
- [27] James Hensman, Nicolò Fusi, and Neil D Lawrence. "Gaussian processes for big data". In: *Uncertainty in Artificial Intelligence - Proceedings of the 29th Conference, UAI 2013*. 2013, pp. 282–290. arXiv: 1309.6835.
- [28] J H Holland. *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. 1975, viii, 183 p. ISBN: 0472084607.

- [29] Wilfried Jakob. "Applying Evolutionary Algorithms Successfully: A Guide Gained from Real-world Applications". In: (July 2021). DOI: 10.5445/IR/1000135763. arXiv: 2107.11300v1. URL: <http://arxiv.org/abs/2107.11300> <http://dx.doi.org/10.5445/IR/1000135763>.
- [30] Y. Jin. "A comprehensive survey of fitness approximation in evolutionary computation". In: *Soft Computing* 2003 9:1 9.1 (Oct. 2003), pp. 3–12. ISSN: 1433-7479. DOI: 10.1007/s00500-003-0328-5. URL: <https://link.springer.com/article/10.1007/s00500-003-0328-5>.
- [31] Yaochu Jin. "Surrogate-assisted evolutionary computation: Recent advances and future challenges". In: *Swarm and Evolutionary Computation* 1.2 (June 2011), pp. 61–70. ISSN: 22106502. DOI: 10.1016/j.swevo.2011.05.001.
- [32] Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. "A framework for evolutionary optimization with approximate fitness functions". In: *IEEE Transactions on Evolutionary Computation* 6.5 (Oct. 2002), pp. 481–494. ISSN: 1089778X. DOI: 10.1109/TEVC.2002.800884.
- [33] Yaochu Jin and Bernhard Sendhoff. "A systems approach to evolutionary multiobjective structural optimization and beyond". In: *IEEE Computational Intelligence Magazine* 4.3 (2009). Cited by: 111, pp. 62–76. DOI: 10.1109/MCI.2009.933094.
- [34] Donald R. Jones, Matthias Schonlau, and William J. Welch. "Efficient Global Optimization of Expensive Black-Box Functions". In: *Journal of Global Optimization* 1998 13:4 13.4 (1998), pp. 455–492. ISSN: 1573-2916. DOI: 10.1023/A:1008306431147. URL: <https://link.springer.com/article/10.1023/A:1008306431147>.
- [35] V. Roshan Joseph and Lulu Kang. "Regression-based inverse distance weighting with applications to computer experiments". In: *Technometrics* 53.3 (Aug. 2011), pp. 254–265. ISSN: 00401706. DOI: 10.1198/TECH.2011.09154. URL: <https://www.tandfonline-com.tudelft.idm.oclc.org/doi/abs/10.1198/TECH.2011.09154>.
- [36] J. Knowles and D. Corne. "On metrics for comparing nondominated sets". In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*. Vol. 1. 2002, 711–716 vol.1. DOI: 10.1109/CEC.2002.1007013.
- [37] Jinglu Li et al. "A classification surrogate-assisted multi-objective evolutionary algorithm for expensive optimization". In: *Knowledge-Based Systems* 242 (2022), p. 108416. ISSN: 0950-7051. DOI: <https://doi.org/10.1016/j.knosys.2022.108416>. URL: <https://www.sciencedirect.com/science/article/pii/S0950705122001691>.
- [38] Hoang N Luong and Peter A.N. Bosman. "Elitist archiving for multi-objective evolutionary algorithms: To adapt or not to adapt". In: *Belgian/Netherlands Artificial Intelligence Conference*. 2012. URL: <http://www.cwi.nl>.
- [39] M. D. McKay, R. J. Beckman, and W. J. Conover. "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code". In: *Technometrics* 21.2 (1979), pp. 239–245. ISSN: 00401706. URL: <http://www.jstor.org/stable/1268522> (visited on 06/12/2022).
- [40] J. Močkus. "On bayesian methods for seeking the extremum". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 27 LNCS (1975), pp. 400–404. ISSN: 16113349. DOI: 10.1007/3-540-07165-2\_55. URL: [https://link.springer.com/chapter/10.1007/3-540-07165-2\\_55](https://link.springer.com/chapter/10.1007/3-540-07165-2_55).
- [41] H. Mühlenbein and G. Paaß. "From recombination of genes to the estimation of distributions I. Binary parameters". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 1141 (1996), pp. 178–187. ISSN: 16113349. DOI: 10.1007/3-540-61723-x\_982. URL: [https://link-springer-com.tudelft.idm.oclc.org/chapter/10.1007/3-540-61723-x\\_982](https://link-springer-com.tudelft.idm.oclc.org/chapter/10.1007/3-540-61723-x_982).
- [42] Linqiang Pan et al. "A Classification-Based Surrogate-Assisted Evolutionary Algorithm for Expensive Many-Objective Optimization". In: *IEEE Transactions on Evolutionary Computation* 23.1 (2019), pp. 74–88. ISSN: 1089778X. DOI: 10.1109/TEVC.2018.2802784.

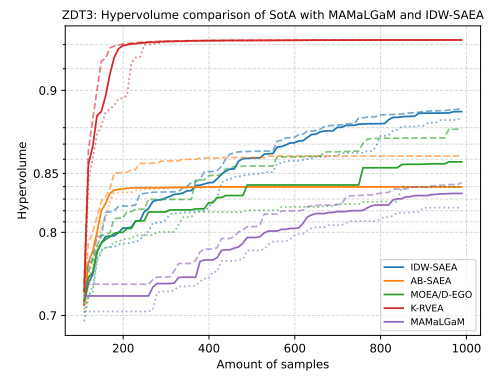
- [43] Linqiang Pan et al. "A Classification-Based Surrogate-Assisted Evolutionary Algorithm for Expensive Many-Objective Optimization". In: *IEEE Transactions on Evolutionary Computation* 23.1 (2019), pp. 74–88. DOI: 10.1109/TEVC.2018.2802784.
- [44] Ingo Rechenberg. "Evolutionsstrategien". In: *Simulationsmethoden in der Medizin und Biologie*. Springer, 1978, pp. 83–114.
- [45] Rommel G. Regis. "Multi-objective constrained black-box optimization using radial basis function surrogates". In: *Journal of Computational Science* 16 (Sept. 2016), pp. 140–155. ISSN: 18777503. DOI: 10.1016/j.jocs.2016.05.013.
- [46] B. Rigaud et al. "Deformable image registration for dose mapping between external beam radiotherapy and brachytherapy images of cervical cancer". In: *Physics in Medicine and Biology* 64.11 (May 2019), p. 115023. ISSN: 13616560. DOI: 10.1088/1361-6560/ab1378. URL: <https://iopscience.iop.org/article/10.1088/1361-6560/ab1378%20https://iopscience.iop.org/article/10.1088/1361-6560/ab1378/meta>.
- [47] Jerome Sacks et al. "Design and analysis of computer experiments". In: *Statistical Science* 4.4 (Nov. 1989), pp. 409–423. ISSN: 08834237. DOI: 10.1214/ss/1177012413.
- [48] Donald Shepard. "A two-dimensional interpolation for irregularly-spaced data function". In: *Proc. Ass. Comput. Mach.* (1964), pp. 517–524.
- [49] L. Shi and K. Rasheed. "A Survey of Fitness Approximation Methods Applied in Evolutionary Algorithms". In: Springer, Berlin, Heidelberg, 2010, pp. 3–28. DOI: 10.1007/978-3-642-10701-6\_1. URL: [https://link.springer.com/chapter/10.1007/978-3-642-10701-6\\_1](https://link.springer.com/chapter/10.1007/978-3-642-10701-6_1).
- [50] Timothy W Simpson et al. "Approximation methods in multidisciplinary analysis and optimization: a panel discussion". In: *Structural and multidisciplinary optimization* 27.5 (2004), pp. 302–313.
- [51] Ye Tian et al. "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization". In: *IEEE Computational Intelligence Magazine* 12.4 (2017), pp. 73–87.
- [52] Felipe AC Viana et al. "Special section on multidisciplinary design optimization: metamodeling in multidisciplinary design optimization: how far have we really come?" In: *AIAA journal* 52.4 (2014), pp. 670–690.
- [53] G. Gary Wang and S. Shan. "Review of Metamodeling Techniques in Support of Engineering Design Optimization". In: *Journal of Mechanical Design* 129.4 (May 2006), pp. 370–380. ISSN: 1050-0472. DOI: 10.1115/1.2429697. eprint: [https://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/129/4/370/5922708/370\\\_\\_1.pdf](https://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/129/4/370/5922708/370\__1.pdf). URL: <https://doi.org/10.1115/1.2429697>.
- [54] Xilu Wang et al. "An adaptive Bayesian approach to surrogate-assisted evolutionary multi-objective optimization". In: *Information Sciences* 519 (May 2020), pp. 317–331. ISSN: 00200255. DOI: 10.1016/j.ins.2020.01.048.
- [55] Yuan Yuan and Wolfgang Banzhaf. "Expensive Multiobjective Evolutionary Optimization Assisted by Dominance Prediction". In: *IEEE Transactions on Evolutionary Computation* 26.1 (2022), pp. 159–173. DOI: 10.1109/TEVC.2021.3098257.
- [56] Qingfu Zhang and Hui Li. "MOEA/D: A multiobjective evolutionary algorithm based on decomposition". In: *IEEE Transactions on Evolutionary Computation* 11.6 (Dec. 2007), pp. 712–731. ISSN: 1089778X. DOI: 10.1109/TEVC.2007.892759.
- [57] Qingfu Zhang et al. "Expensive multiobjective optimization by MOEA/D with gaussian process model". In: *IEEE Transactions on Evolutionary Computation* 14.3 (June 2010), pp. 456–474. ISSN: 1089778X. DOI: 10.1109/TEVC.2009.2033671.
- [58] E. Zitzler, K. Deb, and L. Thiele. "Comparison of multiobjective evolutionary algorithms: empirical results." In: *Evolutionary computation* 8.2 (June 2000), pp. 173–195. ISSN: 10636560. DOI: 10.1162/106365600568202. URL: <https://dl.acm.org/doi/abs/10.1162/106365600568202>.



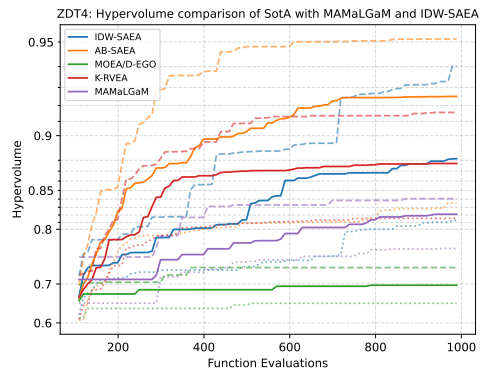
# Appendix



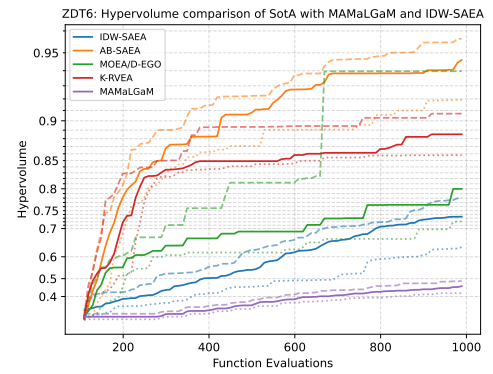
(a) ZDT2



(b) ZDT3



(c) ZDT4



(d) ZDT6

Figure A.1: Hypervolume over time of the ZDT problems

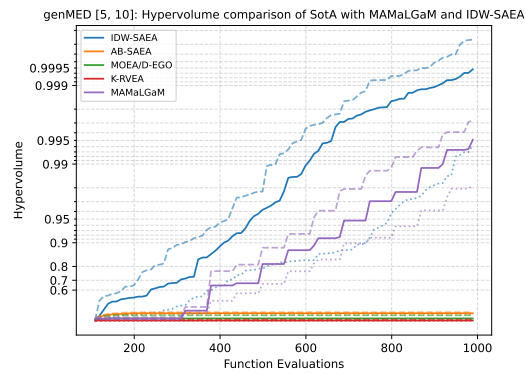


Figure A.2: Hypervolume over time on the genMED [5,10] problem.

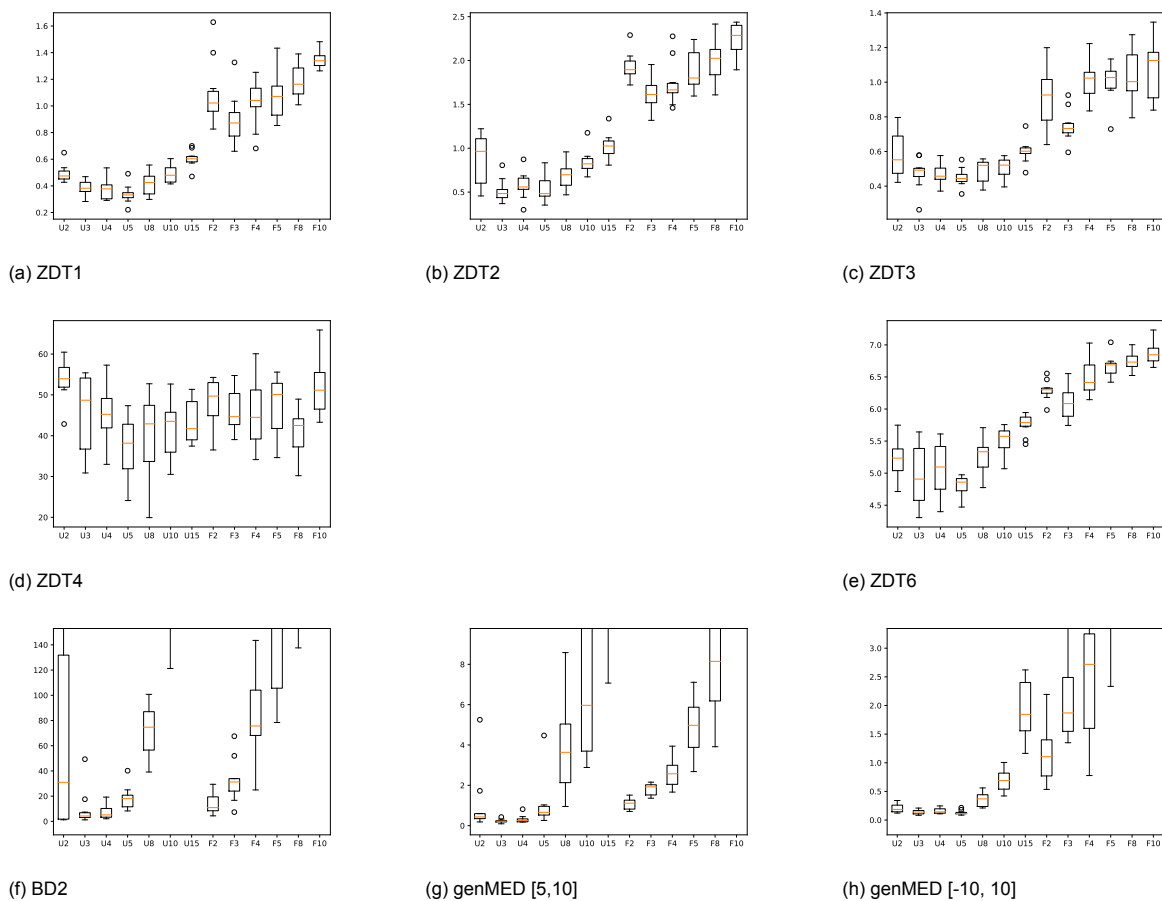


Figure A.3: IGD scores of MAMaLGaM with 1000 FE on different benchmark problems. Different number of clusters were experimented with and also the Univariate FOS (U) and Full FOS (F), number of clusters is indicated with the number after the U of F.