UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,
Mathematics & Computer Science

Multi-Objective Algorithm Configuration on Multi-Modal Multi-Objective Optimization Problems

Master Thesis

submitted by

Oliver Ludger Preuß

Senior Principal Supervisor	Dr. Faiza Bukhsh
Principal Supervisor	Prof. Dr. Heike Trautmann
Daily Supervisor	Jeroen Rook, M.Sc.
Matriculation Number	s3055124
Contact Details	o.l.preus@student.utwente.nl
Submission Date	06.08.2023

in cooperation with

WWU MÜNSTER

WI & STAT

Contents

1 Introduction

The primary objective of multi-objective optimization (MOO) is to solve multi-objective optimization problems (MOP) characterized by multiple conflicting objectives. The goal is to find a set of solutions that form an optimal trade-off between these objectives, known as the Pareto front. Popular algorithms to solve MOPs, i.e., find the Pareto front, are evolutionary multi-objective optimization algorithms (EMOA) (Coello Coello et al., 2007) which are inspired by concepts of variation and selection from evolutionary theory. However, due to the complexity of MOPs, EMOAs usually only find an approximation of the Pareto front (Grimme et al., 2021). EMOAs typically have several configurable parameters that can influence their behavior, which can be beneficial depending on the given MOP to be solved. To find the best set of parameters automated algorithm configuration (AAC) can be used (Hoos, 2011; Rook et al., 2022).

A challenge that needs to be faced in MOO is the presence of multi-modality. Multi-modality describes the occurrence of multiple local and global optima. Local optima are optimal solutions within a certain neighborhood. Global optima are characterized by being the best solution overall for a MOP. A specific case of multi-modality is the multi-global case. The multi-global case is characterized by different points in the decision space corresponding to the same point in the objective space (Grimme et al., 2021) (Fig. 1).

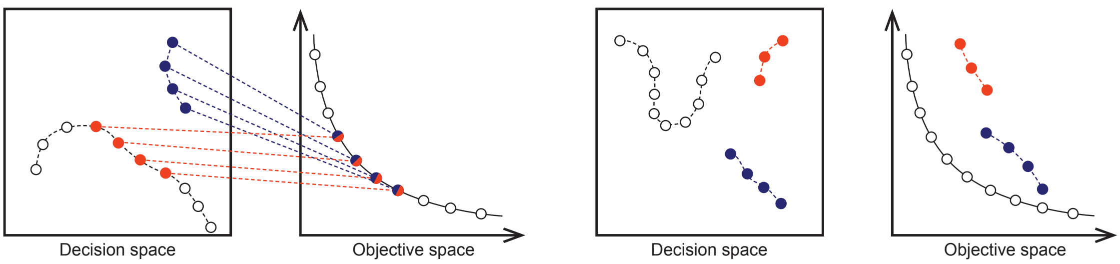


Figure 1 Examples of multi-modality for MOPs. On the left is the multi-global case. On the right is the multi-modal case with local optima. (Figure taken from Heins et al., 2022.)

Here, the focus will be on multi-global optima. When focusing on multiple global optima, it is not necessarily sufficient to focus on the convergence of the EMOAs towards the Pareto front but also find at best all solutions in the decision space that map to the Pareto front. This trade-off between finding a diverse Pareto set and converging towards the Pareto front depends on the configuration used for the chosen EMOA. Rook et al., 2022 showed that configurations selected favoring convergence in objective space negatively affected the diversity of the decision space and vice versa. To solve this problem algorithm configurations that simultaneously consider the diversity of the decision space and the convergence towards the Pareto front need to be selected. Finding these configurations is a MOP in itself resulting in multi-objective AAC (MO-AAC).

This thesis aims to investigate this MO-AAC problem. For that, three different **research questions** will be answered:

1. How competitive are EMOAs configured for both convergence towards the Pareto front and diversity in decision space compared to EMOAs configured for a respective single objective?
2. How configurable are EMOAs?
   (a) Which EMOA is most versatile?
   (b) Which EMOA is most competitive?
3. How does the trade-off between the convergence towards the Pareto front and diversity in decision space when configuring for both simultaneously look like?
   (a) What is the extent of this trade-off
   (b) How do the configurations differ on this trade-off?

The remainder of this thesis is structured as follows: in chapters 2 and 3 the methodology will be outlined, followed by the experiments and their results in chapter 4, and finally the summary and conclusion will be provided in chapter 5.

2 Multi-Objective Optimization

This chapter introduces the fundamentals of multi-objective optimization (MOO). This includes a description of multi-objective optimization problems (MOP), quality measures to assess how well a MOP was solved, evolutionary multi-objective algorithms (EMOA) as tools to solve MOPs and different benchmarking sets that represent MOPs in order to be solved by EMOAs and assessed by quality measures.

2.1 Multi-Objective Optimization Problems

A MOP is characterized by multiple usually conflicting objectives. It is commonly denoted as a vector-valued function

$$\mathbf{f} : \mathcal{X} \to \mathbb{R}^m, \quad \mathbf{x} \mapsto (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x}))^\top,$$

with m real-valued single-objective functions $f_i : \mathcal{X} \to \mathbb{R}, i \in [m] := \{1, ..., m\}$ which are to be optimized simultaneously; w.l.o.g. a minimization of all objectives is assumed. The ranking of solutions in the single-objective case is straightforward. In the multi-objective case, this poses a challenge since multiple objectives usually are conflicting. Thus the notion of *(Pareto-)dominance* needs to be introduced. Given two solutions $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, we say that $\mathbf{x}$ *(Pareto-)dominates* $\mathbf{y}$, denoted as $\mathbf{x} \prec \mathbf{y}$, iff

$$\begin{aligned} f_i(x) &\leq f_i(y) \quad \forall_{i \in [m]} \text{ , and} \\ f_j(\theta) &< f_j(y) \quad \exists_{j \in [m]}. \end{aligned}$$

As a consequence, the solution of a MOP is not a single solution but rather a set of solutions where every single solution is not dominated by any other solution. This set of optimal trade-off solutions can be described as

$$\mathcal{X}_E = \{\mathbf{x} \in \mathcal{X} \mid \nexists\, \mathbf{x}' \in X : f(\mathbf{x}') \prec f(\mathbf{x})\}$$

and is called Pareto set with the corresponding image under $\mathbf{f}$ being the Pareto front (Fig. 2).

A specific property of MOPs is multi-modality. Multi-modality describes the presence of multiple local and global optima of a MOP. A local optimum describes a solution that is optimal within a certain neighborhood. A global optimum describes a solution that is overall optimal. A special case of multi-modality is multi-globality. Multi-globality occurs when different points in the decision space map to the same point in the objective space (Fig. 1). This thesis will focus on the multi-global case.

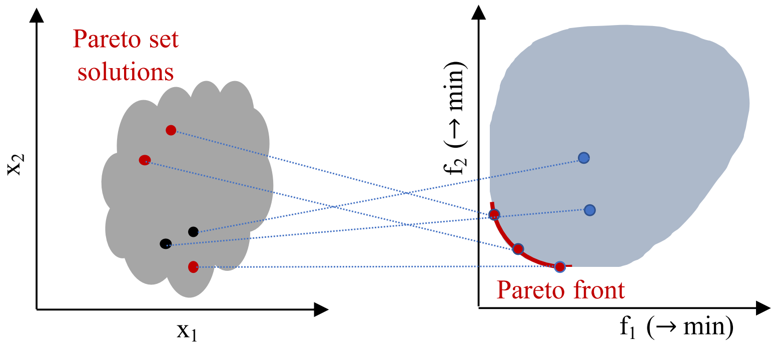


Figure 2 Exemplary visualization of Pareto set and Pareto front, feasible decision (grey) and objective space (blue)

2.2 Quality Measures

Quality measures are important in order to assess the performance of algorithms to solve a MOP. There are different ways to measure a MOPs solution quality. An overview of different measures was given by Zitzler et al., 2003. Here, the focus will be on two measures, *Solow-Polasky measure* (SP) (Solow and Polasky, 1994) which measures the diversity of solutions in the decision space and *dominated hypervolume* (HV) (Zitzler et al., 2003) which measures the quality of solutions in the objective space.

SP was designed to measure the diversity of species in biology. It was later adopted by Ulrich and Thiele, 2011 in the context of *Evolutionary Diversity Optimization*. It measures the pairwise distances of points in the decision space to assess its diversity. SP is defined as

$$SP(P) = \sum_{1 \le i,j \le \mu} M^{-1} \in [1, \mu],$$

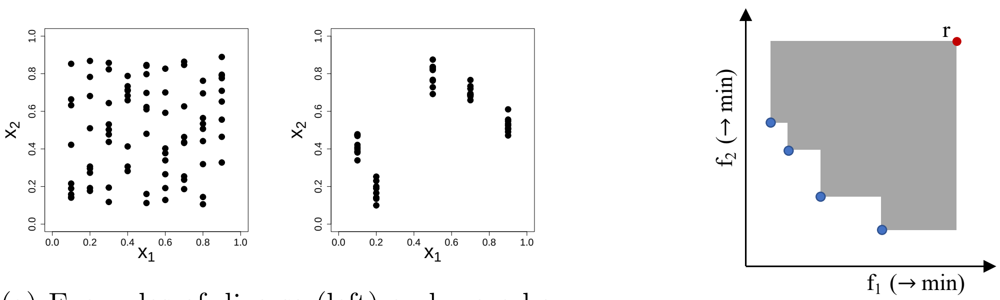
where $P = \{P_1, \dots, P_\mu\}$ is a population of μ individuals, and M^{-1} is the Moore-Penrose generalized inverse matrix of M with $M_{i,j} = \exp(-d(P_i, P_j))$ where d is the the distance between two individuals. If the points are spread out over the decision space, SP will be higher compared to the case when the points are clustered around a small area (Fig. 3a). A high SP is generally favorable since it indicates diverse solutions in the decision space. This is desirable, especially in the multi-global case since multiple points in the decision space can map to the same points on the Pareto front.

HV, also referred to as *S-metric*, is used to measure the quality of the Pareto front approximation To measure the quality of the Pareto front approximation, HV is used. HV measures the area enclosed by a set of non-dominated points X and an

anti-optimal fixed reference point r (see Fig. 3b).
It is defined as

$$HV(X, r) = \lambda_m\Big(\bigcup_{x \in X} [x \leq x' \leq r]\Big),$$

where λ_m is the m-dimensional Lebesgue measure and $\leq$ refers to weak Pareto dominance, i.e. no single point will improve all objectives simultaneously but possibly for certain objectives. A higher HV indicates a better proximity towards the Pareto front and spread of the solutions.



(a) Examples of diverse (left) and more homogeneous decision space according to SP.

(b) Example of the area calculated by HV.

Figure 3 Examples for SP and HV.

2.3 Evolutionary Multi-Objective Optimization

Evolutionary multi-objective optimization is an approach to solving MOPs with the help of principles inspired by evolution (Coello Coello et al., 2007). The general idea is to create a set of solutions, usually referred to as a population of individuals, of size μ. For all individuals, the fitness is evaluated, i.e., the objective function values of the MOP are computed. Out of the population, usually, but not necessarily, the fittest individuals are chosen as parents to create λ offsprings via variation. Variation can be the mutation of single individuals or the recombination of at least two individuals followed by an optional mutation. This creates a new population of size $\mu + \lambda$. To limit the size of the population to size μ a survival selection is performed. There are two typical approaches for this selection. The first one is to select the fittest μ individuals out of all parents and offsprings. The second one is to select μ individuals only from the offsprings. In the second case, it is important to have $\mu \leq \lambda$. This cycle of creating offsprings via variation and selection is repeated until a stopping criterion is met. Stopping criteria could be a number of iterations, a convergence criterion being met, or reaching a wall-time. The final population is a solution for a given MOP (Fig. 4). This solution is usually not the true Pareto front but rather an approximation of the Pareto front since MOPs are typically hard to solve.
An important property of evolutionary multi-objective optimization is stochasticity

introduced by parent selection, variation, and survival selection. This produces different results for multiple runs of the same evolutionary algorithm and MOP. Thus, aggregating solutions over multiple runs is important to counteract this stochasticity.

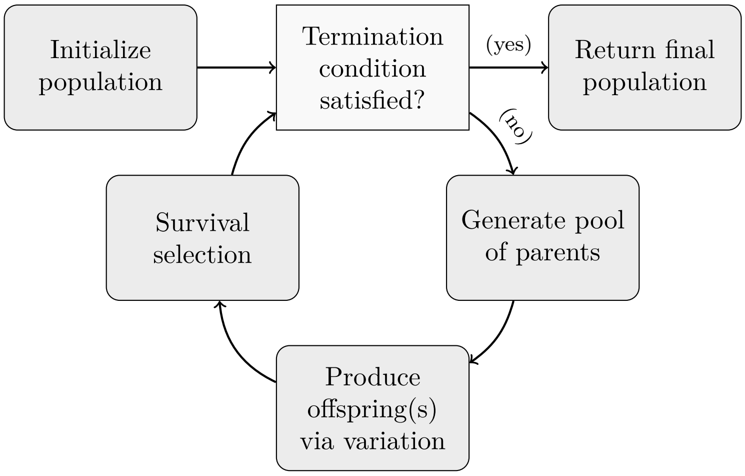


Figure 4 The general process of EMOAs.

Seven different evolutionary multi-objective algorithms (EMOA) are used for the subsequent experiments. The first four are NSGA-II (Deb et al., 2002), Omni-Optimizer (Deb and Tiwari, 2005), SMS-EMOA (Beume et al., 2007), and MOEA/D (Qingfu Zhang and Hui Li, 2007) which are classical EMOAs that intrinsically focus on convergence towards the Pareto front and thus may not be able to find diverse solutions in the decision space according to SP. An exception here might be Omni-Optimizer which was designed to favor a diverse decision space. The remaining three EMOAs utilize the gradient information of a MOP. HIGA-MO (Wang et al., 2017) focuses here on the gradient of the HV. MOGSA (Grimme et al., 2019) and MOLE (Schäpermeier et al., 2022) use the gradient to utilize landscape characteristics to move along local structures and preserve different solutions in the decision space.

NSGA-II

Non-Dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002) uses non-dominated sorting (NDS) as a primary and crowding distance (CD) sorting as a secondary criterion to identify solutions that need to be selected for the next generation. NDS sorts the population into different domination layers. Finding the domination layers is an iterative process. In each iteration, the non-dominated points of the population are identified, put into a domination layer, and removed from consideration for the next domination layer. This step is repeated until all individuals are assigned a domination layer. The first domination layer contains the currently non-dominated points. CD calculates the space around a point where no other individual is found. A low value means at least one other individual is close.

The overall idea of NSGA-II is the following: Perform a NDS and CD sorting on the current population consisting of μ parents and μ offsprings from the previous iteration. Next, take μ individuals as parents from the population based on a tournament selection. The tournament selection takes all individuals from the first to the last domination layer until adding all individuals of the current domination layer would increase the size of the pool of parents beyond μ. In this case, take the best individuals from the current domination layer with the highest CD until there are μ parents. Out of this pool of μ parents create μ individuals via variation. In the special case that there are more than μ individuals in the first domination layer immediately use the individuals with the highest CD. Repeat these steps until a stopping criterion is met.

Omni-Optimizer

In our MO setting, Omni-Optimizer (Deb and Tiwari, 2005, 2008) can be viewed as an extension of NSGA-II. The notion 'Omni', however, refers to that it is designed as to naturally simplify to algorithms for solving more basic optimization problems, including single-objective uni-optimal, multi-optima, and multi-objective uni-optimal optimization tasks. The underlying idea is to speed up convergence due to restricted, binary, tournament selection, combined with enlarging the non-dominated fronts by using ϵ- dominance (Deb, Mohan, et al., 2005) as ranking mechanism. Also, combining both objective and decision space crowding distance measures, so-called variable-space niching, helps to maintain diverse solutions, both in decision as well as in objective space. The latter property also makes this algorithm highly suitable for solving multi-global MO problems.

SMS-EMOA

S-metric Selection-EMOA (SMS-EMOA) (Beume et al., 2007) as $(\mu + 1)$ - strategy uses NDS as primary and HV-contribution as secondary selection mechanisms. The HV-contribution determines how much a single individual x' contributes to the overall HV of a set S of points, i.e. $HV(x'|S, r) = HV(S \cup \{x'\}, r) - HV(S, r)$.
The overall idea of SMS-EMOA is the following: First, a single offspring out of the current population is created. Then NDS on the population including the offspring is performed. Out of the worst domination layer, the individual with the worst HV-contribution is removed.These steps are repeated until a stopping criterion is met.

MOEA/D

Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) (Qingfu Zhang and Hui Li, 2007), specifically known for being also well-suited for many-objective problems with $m \geq 4$, uses decomposition to transform a MOP into SO sub-problems. One way of decomposing a MOP into an SOP is to create a weighted sum of the objective values. Other decomposition methods such as Tchebycheff or Boundary Intersection decompositions can be used as well.
The overall idea of MOEA/D is the following: As initialization create N decompositions of the MOP where each of the resulting SO sub-problems has its own weight vector and initial solution. Based on the weight vectors create a neighborhood for every SO sub-problem. This neighborhood is calculated based on the Euclidean distance of the weight vectors. After the initialization, repeat the following steps until a stopping criterion is met: First, for every sub-problem select two random solutions out of the sub-problems neighborhood. Create a new solution with variation based on the chosen neighboring solutions. If the new solution is better based on their fitness than the current solutions accept the new solution as the current solution. After that, check if the new solution is better for the neighboring sub-problem as well. If it is a better solution accept the new solution as the current solution for the sub-problem. As a last step add all sub-problem solutions to an external population and remove all dominated solutions. After the last iteration, this external population builds the solution of MOEA/D, i.e., the approximation of the Pareto front.

MOGSA

Multi-objective gradient sliding algorithm (MOGSA) (Grimme et al., 2019) uses a different approach than the previously explained algorithms. MOGSA was specifically designed to work with multi-modal problems and is essentially a deterministic local search approach. It focuses on MO gradients, i.e., the sum of normalized gradients of the individual objectives pointing to the direction of the largest possible simultaneous improvement. Conceptually, it exploits the presence of so-called ridges in decision space reflecting the boundary between basins of attraction for the MO gradient on which the algorithm can slide along from dominated basins to efficient sets. MOGSA uses efficient points and sets to explore the MOP. Efficient points and sets can either be local or global. A local efficient point is not dominated by any points in its neighborhood, and a global efficient point is not dominated by any point. Similarly, a local efficient set holds all points that are not dominated by any points in its neighborhood, and all points in a global efficient set are not dominated by any other points.

MOLE

Multi-objective landscape explorer (MOLE) (Schäpermeier et al., 2022) is an extension of MOGSA. MOLE improves the search for efficient points by using adaptive step sizes. Tracing the efficient set is improved by adaptive step sizes, direction prediction of points, redoing an MO descent if necessary, and the ability to decide whether a new point belongs to the current or a new efficient set which helps to identify potential unseen efficient sets. Additionally, MOLE extends MOGSA by a post-processing step in which areas with sparse information about the landscape are explored to find potential efficient points.
The overall idea of MOLE is the following: Perform the improved search for efficient points and subsequently trace the efficient sets. After an efficient set is explored, the post-processing step is performed. When a termination criterion is met, all points that belong to an efficient set are returned.

HIGA-MO

Hypervolume indicator gradient ascent multi-objective optimization (HIGA-MO) (Wang et al., 2017) uses gradient information to optimize a given problem just like MOGSA and MOLE. The difference is that HIGA-MO uses the gradients of the HV with respect to non-dominated points of all domination layers in the objective space, thereby directly searching for both global and local optima regarding HV.
The overall idea of HIGA-MO is the following: First, initialize the population randomly over the decision space. Next, repeat the following steps until the termination criterion is met. Perform NDS to sort all individuals into domination layers. Calculate the HV gradient for every individual in every domination layer, starting from the first domination layer, i.e., the overall non-dominated individuals, and perform a gradient ascent step to maximize HV resulting in a large number of (local) efficient sets that are archived along the algorithm run.

2.4 Benchmark Function Collections

Benchmark function collections are used as a standardized and commonly accepted way of evaluating the performance of algorithms across different settings like algorithm development or configuration and enable reproducible and comparable research within the community. They are designed to reflect specific problem properties and ideally reflect relevant real-world settings, which is, however, debatable with the current benchmark sets available. In the context of this thesis, they are used for automated algorithm configuration on, specifically, multi-modal multiobjective optimization problems.

Five different benchmark function collections are used. ZDT (Zitzler et al., 2003) consists of six bi-objective functions with a different set of characteristics representative of a different class of multi-objective optimization problem. DTLZ (Deb, Thiele, et al., 2005) is an extension of ZDT with a scalable decision and objective space dimensionality. MMF (Yue et al., 2019) consists of 20 functions that are either unimodal or multi-global, but not multi-local. The bi-objective BBOB (Brockhoff et al., 2016) consists of combinations of a subset of the single-objective BBOB benchmark collection (Hansen et al., 2009) and consists of 55 bi-objective functions.

3 Automated Algorithm Configuration

To answer the research questions in this thesis, multi-objective automated algorithm configuration (MO-AAC) is used. Rook et al., 2022 showed the potential for MO-AAC, especially on multi-modal MOPs as they showed that there is a trade-off between diversity in decision space and convergence in objective space. Yet, this line of research is unexplored. Additionally, MO-AAC approaches are rare with MO-ParamILS (Blot et al., 2016) being a model-free MO-AAC algorithm and MO-SMAC (Rook et al., 2023) being a model-based MO-AAC algorithm, which has not been used within the context of EMOAs yet. Thus, this thesis will provide an innovative approach to get insights on EMOAs using MO-AAC on which further research can be based.

To set the foundation for MO-AAC, we first outline the general algorithm configuration problem for the single-objective (SO) case. Given an algorithm A with parameter configuration space Θ, an instance set Π and a quality metric c, the goal of algorithm configuration is to find an optimal configuration $\theta^* \in \Theta$, referred to as *incumbent*, such that $c(\theta^*, \Pi) = \text{opt}_{\theta \in \Theta} c(\theta, \Pi)$, i.e., θ^* optimizes the quality c on the instance set Π (Fig. 5). Typically, $c(\theta, \Pi)$ is obtained by aggregating the quality on all instances $\pi \in \Pi$ for a fixed parameter configuration $\theta \in \Theta$, e.g. taking the mean $c(\theta, \Pi) = \frac{1}{|\Pi|} \sum_{\pi \in \Pi} c(\theta, \pi)$. The search for an optimal parameter configuration is *de facto* also an optimization problem and can be automatically solved, resulting in AAC (Hoos, 2011). Expensive configuration evaluations, mixed-type configuration spaces, and finding configurations on a set of instances are unique characteristics that are badly handled by *off-the-shelf* evolutionary algorithms. Fortunately, many single-objective AAC (SO-AAC) algorithms exist, such as irace (López-Ibáñez et al., 2016), ParamILS (Hutter et al., 2009), and SMAC (Hutter et al., 2011). An extensive overview of AAC algorithms can be found in Schede et al., 2022.

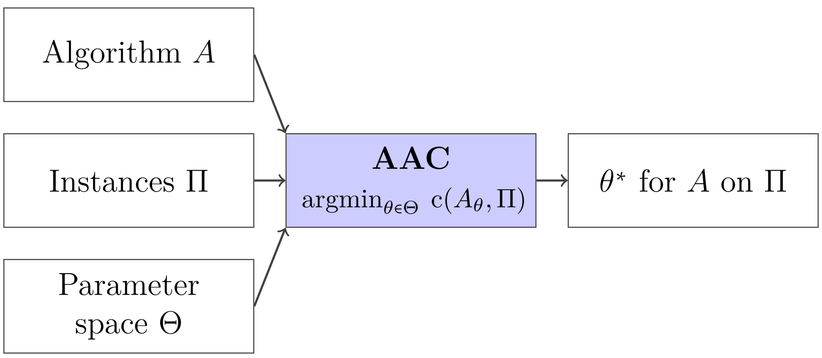


Figure 5 Concept of automated algorithm configuration.

In MO algorithm configuration the goal is to find a configuration that optimizes multiple quality metrics simultaneously, i.e., $c^i : \Theta \to \mathbb{R}, i \in [m], m \geq 2$. These quality metrics are usually conflicting which causes no single configuration to be optimal. Thus, MOO's notion of dominance can also be applied here. A configuration $\theta \in \Theta$ *dominates* another configuration $\theta' \in \Theta$, denoted as $\theta \prec \theta'$, iff

$$\begin{aligned} c^i(\theta) &\leq c^i(\theta') \quad \forall_{i \in [m]} \text{ , and} \\ c^j(\theta) &< c^j(\theta') \quad \exists_{j \in [m]}. \end{aligned}$$

W.l.o.g., minimization of all objectives is assumed. Given the notion of dominance, a configuration $\theta \in \Theta$ is Pareto-optimal if no other configuration in Θ dominates θ. The final incumbent in the MO case is a set of optimal trade-off configurations, i.e., the Pareto front, which can be described as

$$\Theta^* = \{\theta \in \Theta \ \mid \nexists\, \theta' \in \Theta : \theta' \prec \theta\}.$$

This description of the Pareto front for AAC is the same as the description of the Pareto front of MOO in chapter 2.1. Unfortunately, the EMOAs discussed in Chapter 2.3, again, cannot directly be used for MO-AAC despite the similar nature of the optimization problem because of the expensive configuration evaluations, mixed-type configuration spaces, and search for configurations on a set of instances. Expensive configuration evaluations restrict the number of function evaluations that can be performed within a reasonable time, whereas EMOAs usually require a large number of function evaluations to converge. Most mutation, cross-over, and gradient methods can handle either discrete or continuous parameters, which is not the case for algorithm parameter spaces. We seek to find configurations that yield the best overall performance on a set of instances. It is evident that non-competitive configurations can be detected while only being evaluated on a subset of these instances. Exploiting this characteristic can yield considerate improvements in the number of configurations that can be assessed during configuration which is not done by generic EMOAs. Nonetheless, the performance measures for MOO, i.e., HV and SP, described in Chapter 2.2 can be applied.

3.1 Sequential Model-based Algorithm Configuration

For the AAC in the subsequent experiments sequential model-based algorithm configuration (SMAC) (Hutter et al., 2011) and multi-objective SMAC (MO-SMAC) (Rook et al., 2023), an extension of SMAC to work in the multi-objective case, are used.

SMAC is a model-based automated algorithm configurator. During configuration, SMAC keeps track of the current best configuration, referred to as incumbent, and a run history of all trails of tested configurations with their achieved quality metric on an instance. This run history is used to train an empirical performance model (EPM) – usually in the form of a random forest (RF) – which is used to predict the quality metric of unseen configurations. The EPM takes the configuration and, optionally, instance features as input. This EPM is then used to predict the aggregated cost over all the training instances. Consecutively, an acquisition function (AF) is used to *score* each configuration. By default, this AF is the expected improvement (EI) (Jones et al., 1998), which balances the exploratory and exploitative nature of a configuration. It, therefore, uses the prediction and variance of the EPM, respectively. E.g., a configuration with a good prediction and low variance is considered to be exploitative, and a configuration with a lower prediction but high variance is more exploratory.

SMAC is an iterative algorithm with its first step choosing potential new incumbent configurations. To obtain new configurations the EPM is fitted on the complete run history at that moment, after which a random search and local search, using the 1-exchange neighborhood (Hoos and Stüzle, 2004) are performed until promising configurations, according to the AF, are found. These potential configurations, ranked on their AF score and interleaved with randomly sampled configurations, are tested against the incumbent in an intensify step.

The intensify step incrementally evaluates these new configurations on the instances where the incumbent also ran on, as long as the empirical performance of the new configuration is not worse than that of the incumbent. When the new configuration is evaluated on all the same instances as the incumbent and has better aggregated performance, it becomes the new incumbent. Otherwise, it gets rejected. The first incumbent will be evaluated on one instance and after each run with a new configuration, the incumbent will be evaluated on one new instance. This way, the incumbent will be evaluated on more instances as the search progresses.

SMAC balances the time spend between finding new configurations and the intensify step, usually in a 1-to-1 ratio. To achieve this, the intensify step continues with evaluating configurations until it used the same amount of wall-clock time as the finding step did, with the one exception that at least two configurations from the list of potential configurations need to be evaluated. This process of alternating between finding and intensifying repeats until a termination criterion is met. A termination criterion can be reaching a wall-time limit or reaching a number of function evaluations (Figure 6).

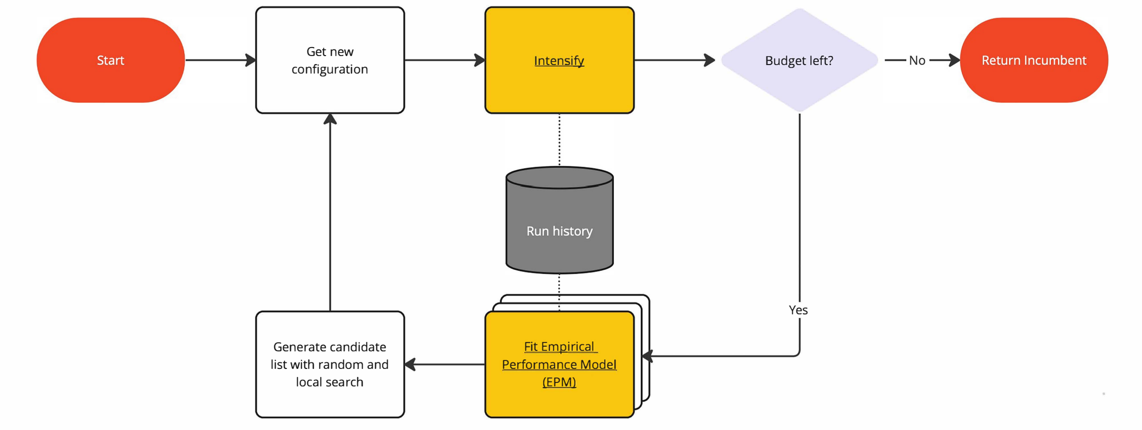


Figure 6 High-level overview of the iterative process of (MO-)SMAC.

3.2 From SMAC to MO-SMAC

MO-SMAC (Rook et al., 2023) is an extension of SMAC to work in the multi-objective domain. It modifies several key components in SMAC, namely incumbent, the AF and the intensification procedure.

The first – intuitive – change is the incumbent. Instead of being a single configuration, it is now a set of (non-dominated) configurations. This follows the explanation of the MO-AAC problem at the beginning of this chapter. The size of the incumbent is, by default, bounded to 8 to ensure a progressive increment of the number of instances the incumbent configurations have been evaluated on.

The second change is to use the predicted Hypervolume improvement (PHVI) as the acquisition function. This AF has an EPM for each objective and scores configurations based on how much they increase the Hypervolume over the existing configurations in the incumbent. The search and ranking procedures remain the same as in SMAC. PHVI is a purely exploitative AF, i.e., it does not make use of the variance of the prediction. However, exploration is still performed due to the interleaving of random configurations.

The third and last change is the intensify step. Here, three major changes are proposed; 1) the moment to intermediately compare the new challenger, 2) how to do this intermediate comparison, and 3) how to update the incumbent. A new configuration is only intermediately evaluated when the new aggregated cost dominates its cost of the previous comparison. This reduces the probability of falsely rejecting a configuration to become part of the incumbent, which is an inherent risk due to the noisy characteristics of partial evaluations. For the same reason, a new configuration is only compared against the incumbent, which is closest to it and is based on the

Euclidean distance in the objective space. As long as the selected incumbent configuration does not dominate the new configuration, the evaluation continues. Finally, when the new configuration is still non-dominated by the other configurations in the incumbent, it is added to the incumbent. Also, when existing incumbent configurations are now dominated, they are removed from the incumbent. In the case that the incumbent is larger than a pre-defined maximum, the configuration with the lowest crowding distance (Deb et al., 2002) is removed with the aim to keep a (in objective space) diverse set of configurations.

4 Experiments

In the subsequent experiments, various EMOAs will be configured to simultaneously have a diverse decision space and convergence towards the Pareto front using a set of multi-modal MOPs and MO-AAC. The results of the experiments will be analyzed to specifically answer the research questions provided in chapter 1.

4.1 Experimental Setup

The experimental setup is kept as close as possible to the experiments done by Rook et al., 2022 to build on top of their work. This enables the current experiments to be comparable and extend their research. The set of test instances includes all problems from ZDT, DTLZ, and MMF except ZDT5 and MMF13, and instances f_{46}, f_{47}, and f_{50} from bi-objective BBOB. This results in a total of 33 instances. All instances are bi-objective and have a 2-dimensional decision space. If the reference point for calculating the HV was unknown, it was obtained by taking the maximum values of the non-dominated set of solutions over the union of all EMOAs on the given instance.
To ensure the comparability of HV values across different instances and enable aggregation, the HV was normalized against the maximum obtainable HV for each instance during the configuration process. These maximum obtainable HVs are empirically approximated by combining all evaluations while running all considered algorithms 10 times with an evaluation budget of 100 000. For SP the absolute values were used since SP is not dependent on the instance but on the number of points that are considered. The population size μ was fixed to 100 to prevent the configurators from finding configurations where the population equals the number of function evaluations, which yields a high diversity in decision space but does not actually run the respective EMOA beyond the initialization of the population. MOLE and MOGSA can return a larger solution set than 100, as they do not have a population. In case they return more than 2 000 points, 2 000 points were randomly sampled without replacement to keep the SP computation – which relies on the matrix inversion of dense matrices – possible. A careful reader might wonder why we did not sample 100 points instead. This is motivated by the fact that by performing a subset selection that yields the highest SP is computationally infeasible. For AAC, SMAC is used for the SO configuration, while MO-SMAC is used for the MO configuration. The general process is the same for both cases, with the difference that SMAC will configure for SP and HV separately whereas MO-SMAC will configure the algorithms for SP and HV simultaneously. The previous experiments with SMAC are repeated because MO-SMAC is built on top of a different SMAC version (Lindauer et al., 2022) as was

used in the experiments of Rook et al., 2022 and the comparability between SMAC and MO-SMAC is otherwise not given. Considering the 7 different EMOAs a total of 21 different configuration scenarios were performed. Each scenario had a termination criterion of 250 algorithm calls. All other (MO-)SMAC parameters were set to default. Each of the seven algorithms had a function call budget of 20 000. A preliminary study showed that most algorithms converged at this point. Since SP and HV need to be maximized, but (MO-)SMAC is designed to minimize, the negative values of HV and SP were set as objectives.
Due to the small size of test instances, 10-fold cross-validation (CV) was used, as this was computationally feasible for the 21 configuration scenarios compared to leave-on-out CV. This was the only change compared to the experiments of Rook et al., 2022. In each fold, 10 separate configuration runs were performed to account for the stochastic behavior of (MO-)SMAC. Out of these 10 configuration runs, the incumbent solution for SMAC is selected based on the one that yielded the best performance on the training instances. For MO-SMAC all configurations from the incumbents are combined, and only the non-dominated configurations are selected as final incumbent. Again, this is based on their performance on the instances in the training partition of the fold. In addition to the cross-validation folds, the configuration scenarios also were configured on all instances. The final evaluation measurements for the configurations of the CV-folds are based on the test instances of every fold and for the no CV scenarios on all instances. During evaluation, all instances were evaluated 25 times with different random seeds on all EMOAs with their respective found non-dominated configurations. An overview of the experimental setup can be found in Figure 7.

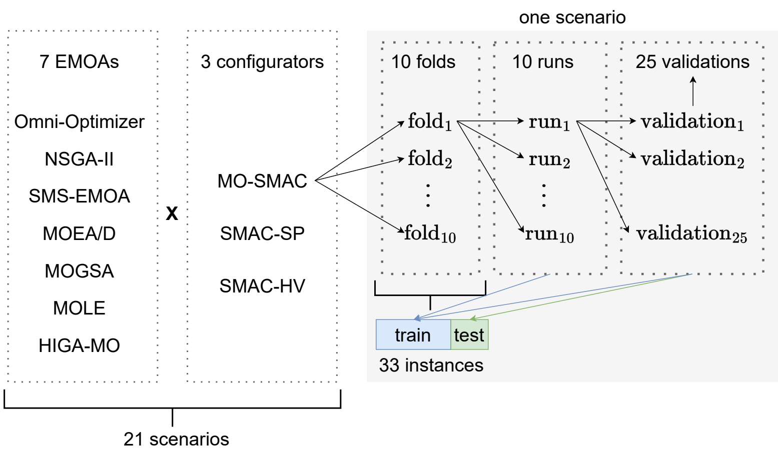


Figure 7 Overview of the relationship between configuration scenarios, folds, configuration runs, and validation.

4.2 Results

The experimental results are evaluated to answer the research questions asked in Chapter 1. For that, the following paragraphs are divided to represent a single research question. The first two research questions and the first half of the third research question use the results of the CV. Here only the validation instances are considered. The second half of the third research question uses the results for training and validating on all instances. Depending on the research question the results are aggregated in a different way. The specifics will be explained for each research question separately. Figure 8 gives an overview of what aspect of the results are investigated regarding every research question as it can be difficult to follow which parameter space or which configurations are being investigated.

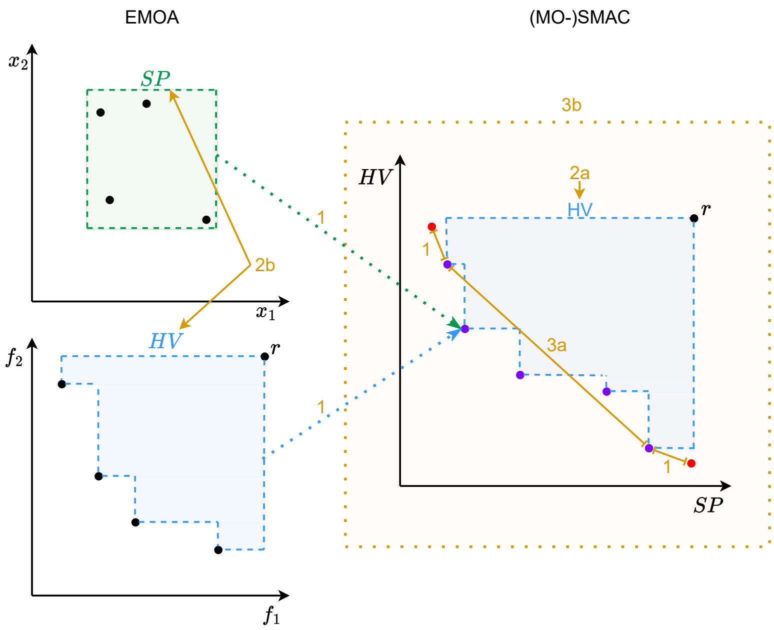


Figure 8 Overview of the different aspects that will be analyzed. The figures on the left side represent the function-space and the figure on the right represents the indicator-space. Displayed in orange are the respective research questions and it is shown which *space* is used to answer them.

1. How competitive are EMOAs configured for both convergence towards the Pareto front and diversity in decision space compared to EMOAs configured for a respective single objective?

Rook et al., 2022 showed the potential of AAC for multi-modal MOPs. In order to show that, they configured the EMOAs using SO-AAC for SP and HV separately. It is important that the MO-AAC for configuring SP and HV simultaneously does not substantially decrease the achieved performance on either SP or HV. Therefore, the

SO configurations' performance is compared to those of the MO configurations that achieve the highest SP and HV respectively.
This comparison shows that only MOEA/D and MOGSA have a decrease in SP regarding the best configuration of the MO-AAC, while MOGSA has the highest decrease in SP. All other EMOAs configured by MO-AAC are at least as good if not slightly better than the SO-AAC configured versions. When considering HV, there is a moderate improvement visible. MOGSA is an exception with as much improvement in HV as it has decreased in SP (Fig. 9).

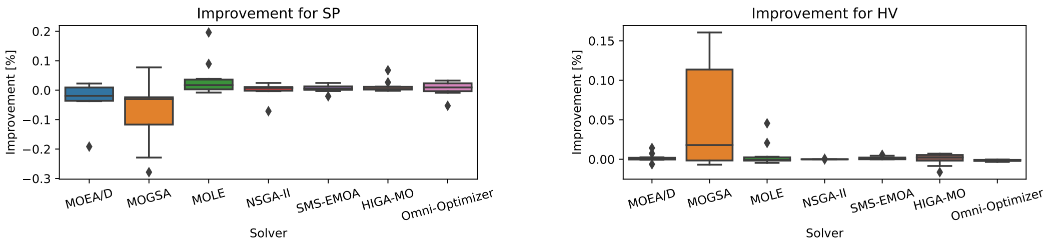


Figure 9 Relative improvement of the best MO configuration for SP and HV compared to the SO configuration respectively.

The comparison of the best configurations regarding SP and HV from the MO-AAC compared to the SO-AAC displays that the MO-AAC is capable of finding configurations with comparable performance to SO-AAC configurations. This shows that all other configurations found by MO-AAC will form a trade-off between SP and HV that lies in between the best configurations for SP and HV. The trade-off between the MO-AAC configurations as well as the performance of the SO-AAC configurations can be seen in Figure 13.

2. How configurable are EMOAs?

The second research question deals with the configurability of EMOAs. To answer this question two dimensions of configurability are looked at. The first one is *versatility* and the second one is *competitiveness*.
When talking about the versatility of an algorithm the capability to be usable in different settings is referred to. In the current setting of using MO-AAC to optimize EMOAs towards SP and HV, this capability includes finding configurations that optimize SP and HV simultaneously and either SP or HV specifically. This allows an EMOA to be usable specifically for a user's goals of having diverse points in the decision space, focusing on convergence towards the Pareto front or a trade-off between both.
The competitiveness of an EMOA describes how well an EMOA performs compared to other EMOAs on the same MOPs. EMOAs with high competitiveness are more

likely to be chosen for a given optimization problem. Even if an EMOA is versatile but can not compete with other EMOAs, it will not be chosen. Thus it is an important trait of an EMOA to have.

(a) Which EMOA is most versatile?

The performance of every non-dominated configuration is measured using SP and HV. Thus the configurations can be represented as points in the 2D space spanned by HV and SP. On these points, a HV can be calculated. This HV is calculated in the space of the performance measures of the non-dominated configurations. It is different from the HV previously calculated on the non-dominated solutions of an EMOA. As explained in Chapter 2.2 the HV measures the quality and the spread of solutions. Thus this method of evaluation is well suited to answer how versatile an EMOA is.

For every EMOA in total 2500 HVs are calculated. This is based on 10 folds, 10 configuration runs of (MO)-SMAC per fold, and 25 validation evaluations of the EMOA per configuration run. The final SP and HV for one validation evaluation is the mean over the validation instances of that fold. For each EMOA, every combination of fold, configuration run, and validation evaluation a ranking of the achieved HV is performed. The average over all 2500 rankings is used for the final ranking for every EMOA. The rankings are displayed in so-called Critical Distance plots where lower rankings mean better performance. Additionally, they display a critical distance, shown as thick vertical bars, to indicate that two entities in the plot do not have a statistically significant different ranking. The critical distance is calculated using a Nemenyi test (Nemenyi, 1963) with $\alpha = 0.1$, resulting in a critical distance of 0.16.

The rankings show that Omni-Optimizer has the highest HV overall followed by MOLE. The worst HV achieved MOGSA and SMS-EMOA while MOGSA is the worst EMOA overall. HIGA-MO, NSGA-II, and MOEA/D all achieved an average ranking where all three EMOAs do not have a significantly different HV than the others (Fig. 10).

The constant highest rank for Omni-Optimizer and second highest for MOLE show that for both EMOAs, high performing and diverse configurations can always be found. In contrast, the other EMOAs do not achieve a high HV and have considerably lower ranks.

(b) Which EMOA is most competitive?

In order to measure the competitiveness of the EMOAs, they are ranked based on their performance separately, considering their achieved HV and SP on their final

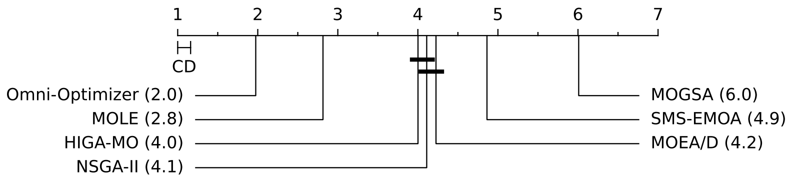


Figure 10 Rankings based on the HV of the configurations.

Pareto front approximations. These rankings are similarly calculated as the ranks in the previous research question with a critical distance of 0.28. Here the mean of the actual SP and HV values over all non-dominated configurations are considered. Thus it gives an overall performance value for all found non-dominated configurations. The final rank is the average of over 2500 ranks based on performance for every combination of fold, configuration run, and validation evaluation.
The general order of the ranking for SP and HV looks very similar. For SP Omni-Optimizer outperforms NSGA-II, but for HV NSGA-II is marginally, but not significantly, better than Omni-Optimizer. SMS-EMOA and MOEA/D achieve average ranks. Unlike the rankings from the first research questions, MOLE achieves a lower end ranking for SP and HV. HIGA-MO and MOGSA have the worst rankings (Fig. 11).

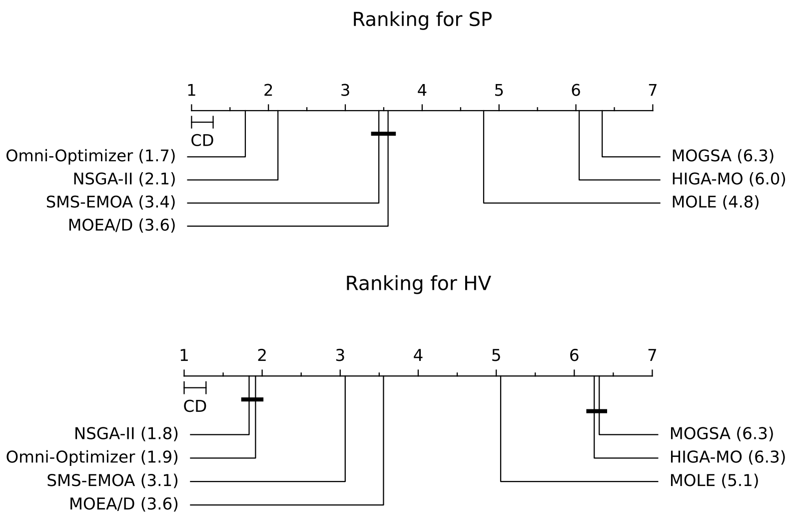


Figure 11 Performance rankings for SP and HV.

The rankings for versatility and competitiveness show similar overall results. Omni-Optimizer has in both cases the highest rank. Regarding competitiveness, it is closely followed by NSGA-II which is not surprising since Omni-Optimizer is an extension of NSGA-II. Overall, the worst rankings have HIGA-MO and MOGSA, which aligns with the versatility rankings. The order of the rankings regarding MOLE, SMS-EMOA, and MOEA/D changes depending on considering versatility or competitiveness but in both cases they have average rankings.

3. How does the trade-off between the convergence towards the Pareto front and diversity in decision space when configuring for both simultaneously look like?

MO-AAC aims to find multiple configurations that form a trade-off between the objectives. Investigating this trade-off and gauging how it affects the different configurations is of interest. First, the extent of the trade-off between SP and HV for different configurations will be investigated to get insights into this trade-off. Second, the configurations themselves will be looked at.

(a) What is the extent of the trade-off?

In order to assess the extent of the trade-off between SP and HV the loss of SP and HV for the best configurations regarding SP and HV will be compared, i.e., the two extreme solutions in HV and SP space. More specifically, the MO configuration with the best performance regarding SP will be compared to the MO configuration with the best performance regarding HV. The average over the median of the 25 validation runs for the validation instances is calculated in order in order to pick these configurations. The trade-off is calculated by dividing the worst performing configuration by the best performing configuration over the configuration runs. The loss is calculated by subtracting 1 from the trade-off. This way the actual loss can be seen instead of a ratio between SP and HV.
SP shows in general a higher loss than HV. MOEA/D and Omni-Optimizer both have the highest loss for SP and HV. This loss is up to a little over 10 % for SP and up to 7 % for HV. MOGSA and SMS-EMOA have the second highest loss for SP, up to about 5 %. All other EMOAs have a roughly 2.5 % loss on SP. MOGSA and MOLE have the second highest loss on HV with losses ranging up to 3 %. All other EMOAs have a loss lower than 2 % on HV.

The trade-off investigation shows that MO-AAC is able to find different configurations with a trade-off that underlines the results of the previous research question. The higher the trade-off is, the more versatile an EMOA is since there are configurations that specifically perform well on SP or HV. It further shows how sensitive an EMOA is regarding the chosen configuration parameter since they impact the functionality and thus the performance.
Most interesting is Omni-Optimizer, with a high loss on SP and HV. This reflects

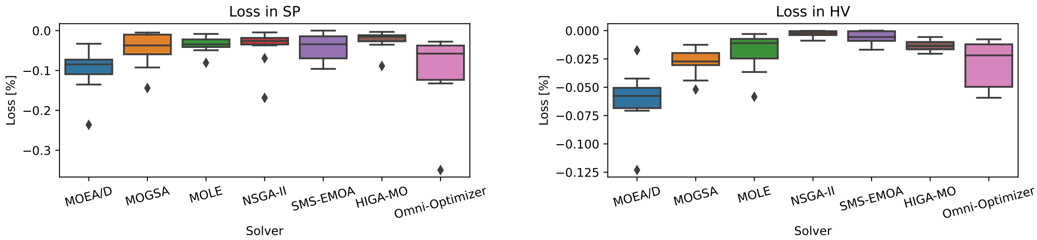


Figure 12 Relative loss for HV and SP on MO configurations.

the results of the second research question. It shows that the configurations of Omni-Optimizer have the highest HV which in combination with these results shows that these configurations have a good spread. Additionally, the ranking of the performance for Omni-Optimizer is very good, which underlines the quality of the configurations.

(b) How do the configurations differ on this trade-off?

For the previous research questions, the 10-fold CV was used. However, to get insights into specific configurations CV is unsuitable since it will produce a lot of configurations. Thus for the subsequent evaluations, the configuration and validation is done on the whole data set.
Since there are 10 configuration runs per EMOA, not all found configurations are non-dominated. Thus there will be a distinction between the amount of found, unique, and non-dominated configurations. For all EMOAs except Omni-Optimizer, all configurations are unique. Omni-Optimizer has with 42 configurations, 40 unique and 5 non-dominated by far the most configurations. NSGA-II is the only EMOA with the same amount of non-dominated configurations while having 27 unique configurations. MOEA/D and SMS-EMOA have 3 non-dominated configurations and 29 and 23 unique configurations respectively. MOLE and HIGA-MO each have 2 non-dominated and 19 and 17 unique configurations respectively. MOGSA is the only EMOA with only one non-dominated configuration while having 17 unique configurations (Tab. 1).

Algorithm	# configs	unique configs	non-dominated
MOLE	19	19	2
MOGSA	17	17	1
NSGA-II	27	27	5
HIGA-MO	17	17	2
MOEA/D	29	29	3
Omni-Optimizer	42	40	5
SMS-EMOA	23	23	3

Table 1 Number of configurations found.

Figure 13 shows the performance of all configurations. This includes the default, dominated, non-dominated, and SO configurations for SP and HV. Most of the non-dominated configurations dominate the default configuration for every EMOA. Every SO configuration is dominated by at least one configuration found by MO-AAC. This underlines the potential of MO-AAC and the results of the second research question. From the non-dominated configurations, a potential optimal trade-off configuration can be picked. Only for NSGA-II, this is not as clear as there is one configuration that is further off regarding SP.

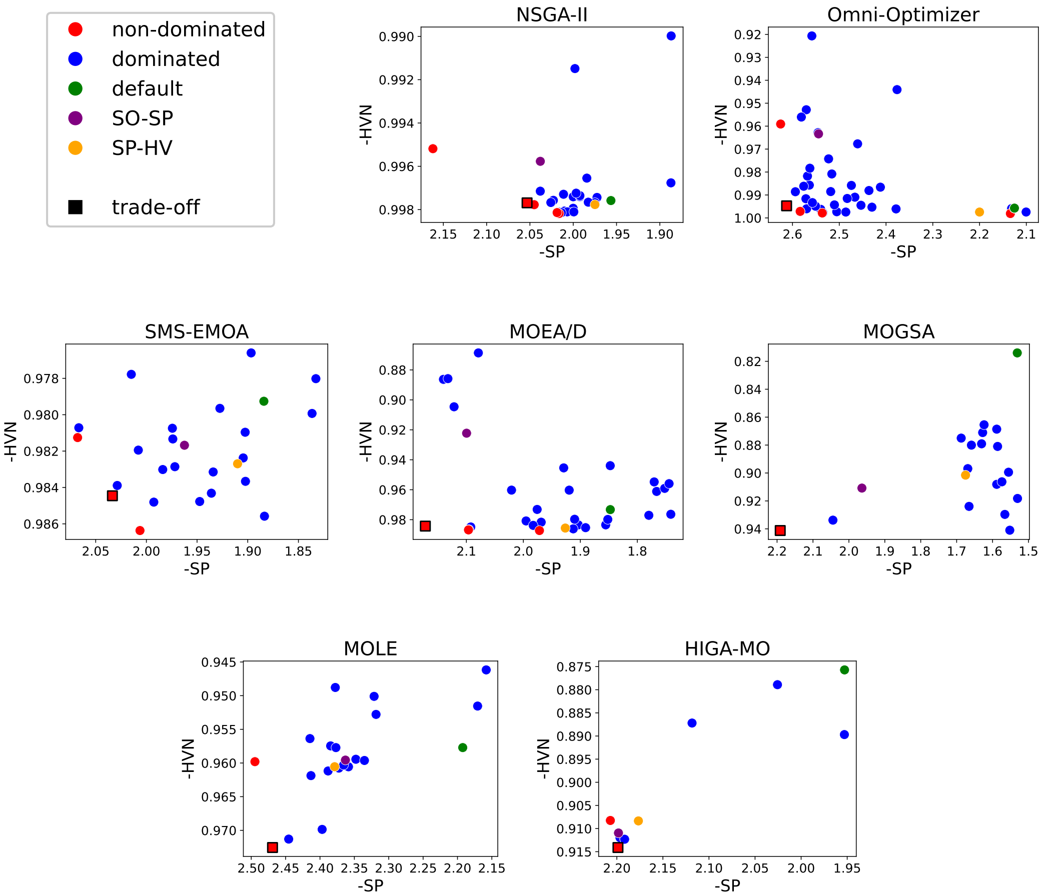


Figure 13 Configurations for every EMOA separately, trained and validated on all instances.

Figure 14 combines all non-dominated configurations from the sub-figures of Figure 13. It represents the results of the first research question in a visual way. Omni-Optimizer is the overall most performant EMOA in terms of HV of the non-dominated configurations and overall performance followed by NSGA-II. SMS-EMOA is comparable to MOEA/D in terms of SP and HV performance, as seen in the second part of the first research question. HIGA-MO and MOGSA are the worst performing EMOAs.

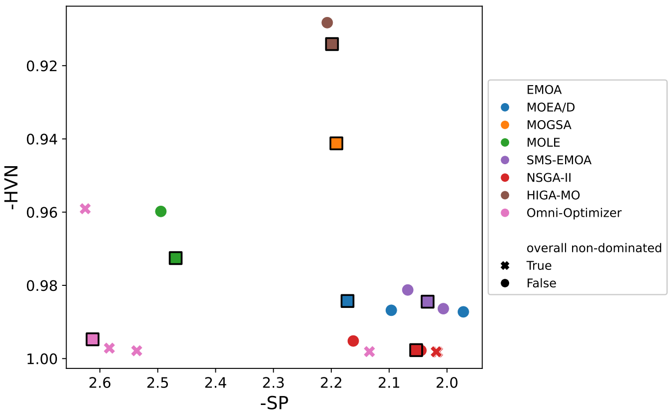


Figure 14 Non-dominated configurations for every EMOA combined, trained and validated on all instances.

Discussion

The results of the research questions show interesting insights into MO-AAC. The first one is that MO-AAC is capable of finding configurations that have comparable competitive performance to configurations found by SO-AAC (Fig. 9). This is underlined by Figure 13. Here it is visible that the MO-AAC configurations substantially outperform the SO-AAC configurations. This is counter-intuitive as SO-AAC should be able to find the best performing configuration for the respective measure. An explanation for this could be that MO-AAC is more robust regarding local optima due to the combined optimization of both measures where configurations that are accepted by SO-AAC would be dominated in MO-AAC. Thus MO-AAC might not get trapped in local optima like it could be the case in SO-AAC.

The second insight is the dominant performance of Omni-Optimizer. This performance is demonstrated in different aspects. First, the HV of the non-dominated configurations was calculated where Omni-Optimizer achieved the highest rank. A contributing factor here is the ability of Omni-Optimizer to find a lot of (non-dominated) configurations (Tab. 1) and more specifically well performing configurations which was demonstrated in the second aspect. The second aspect addressed the performance of the EMOAs regarding SP and HV. Here Omni-Optimizer shows dominant performance together with NSGA-II. For SP Omni-Optimizer shows better performance whereas on HV Omni-Optimizer and NSGA-II do not have a significantly different performance. This is unsurprising since Omni-Optimizer is an extension to NSGA-II to preserve diversity in decision space. Both aspects are also reflected in Figure 13 and Figure 14.

The third insight is that there are configurations that dominate the default configuration for every EMOA when considering SP and HV. Thus these configurations could be used as new default configurations. As a new default, the best trade-off configuration is proposed. This configuration is chosen visually on the basis of Figure 13. An overview of the actual parameters for the default, non-dominated, and proposed new default configurations can be seen in Figures 15 and 16 of the appendix.

5 Summary, Conclusion & Future Work

This thesis successfully investigated interesting aspects of MO-AAC for multi-modal MOPs and gained interesting insights into a so far rather unexplored field at the forefront of current research. In this context, various EMOAs were configured to simultaneously generate diverse solutions in the decision space and foster convergence towards the Pareto front. These configurations were performed using the model-based AAC algorithms (MO-)SMAC which demonstrated its high performance and potential for multi-objective configuration tasks. As test instances, a set of, also multi-modal, multi-objective optimization instances of different benchmark function collections were utilized.

The first research question investigated how competitive MO-AAC is compared to SO-AAC. The results show that MO-AAC is able to find configurations that have competitive, or even better, performance than configurations found by SO-AAC.
The second research question investigated the configurability of EMOAs. For that, the versatility and competitiveness of the considered EMOAs were compared. In both cases, Omni-Optimizer as a flexible, general-purpose optimizer interestingly achieved the best performance. Regarding the competitiveness of HV, NSGA-II has the same performance as Omni-Optimizer.
The third research question addressed the trade-off between different configurations for EMOAs individually. Here, Omni-Optimizer showed the highest trade-off regarding HV and the second highest trade-off regarding SP. The highest trade-off for SP showed MOEA/D. As an evaluation method to further analyze the trade-off behaviour, the non-dominated configurations were visualized in the space spanned by the chosen performance indicators SP and HV. Thereby, the performance of configurations found by MO-AAC could be nicely compared to the default and respective SO-AAC configurations. It was clearly visible that the MO-AAC configurations outperformed the other configurations and Omni-Optimizer achieved the overall best performance. The best trade-off configuration for every EMOA can potentially be picked out of these visualizations and proposed as a new default configuration. However, the best suited configuration in practice of course depends on the specific underlying optimization problem and the preferences of the decision makers involved. Without this knowledge, a best-compromise configuration in the knee region of the resulting Pareto front approximation would be a good candidate.

The experimental results promise that there is a lot to gain by further pursuing this line of research fostering algorithm design, understanding of algorithm behaviour, and automated configuration. A straightforward extension will be investigating scalability and generalizability of results, both in terms of enlarging decision space and possibly also objective space dimensionality. Also, other benchmark sets as e.g. recently provided in Schäpermeier et al., 2023 will be included. Experimental results then will be further detailed and more specifically analyzed focusing on specific benchmark sets and problem characteristics separately.

Moreover, MO configuration studies internally solve specific MOO problems themselves, and it is crucial to analyze the characteristics of the resulting optimization landscapes in order to get a detailed understanding of problem hardness and structural properties such as multimodality. This will further help in understanding the performance differences of different configurators applied to the underlying scenario. Respective experiments will thus include a comparison to MO-ParamILS and potentially specific racing approaches as well. The robustness of resulting EMOA configurations is also an issue and could even be integrated as a third performance criterion into the MO configuration scenario. However, different notions of robustness exist which could be explored in this regard.

Additionally, automated configurators do have parameters themselves which should be analyzed further regarding parameter importance and sensitivity of results regarding the chosen settings within SO- and MO-SMAC. However, one has to be careful not to end up in a 'vicious circle' of meta-configuring configurators.

A Configurations Overview

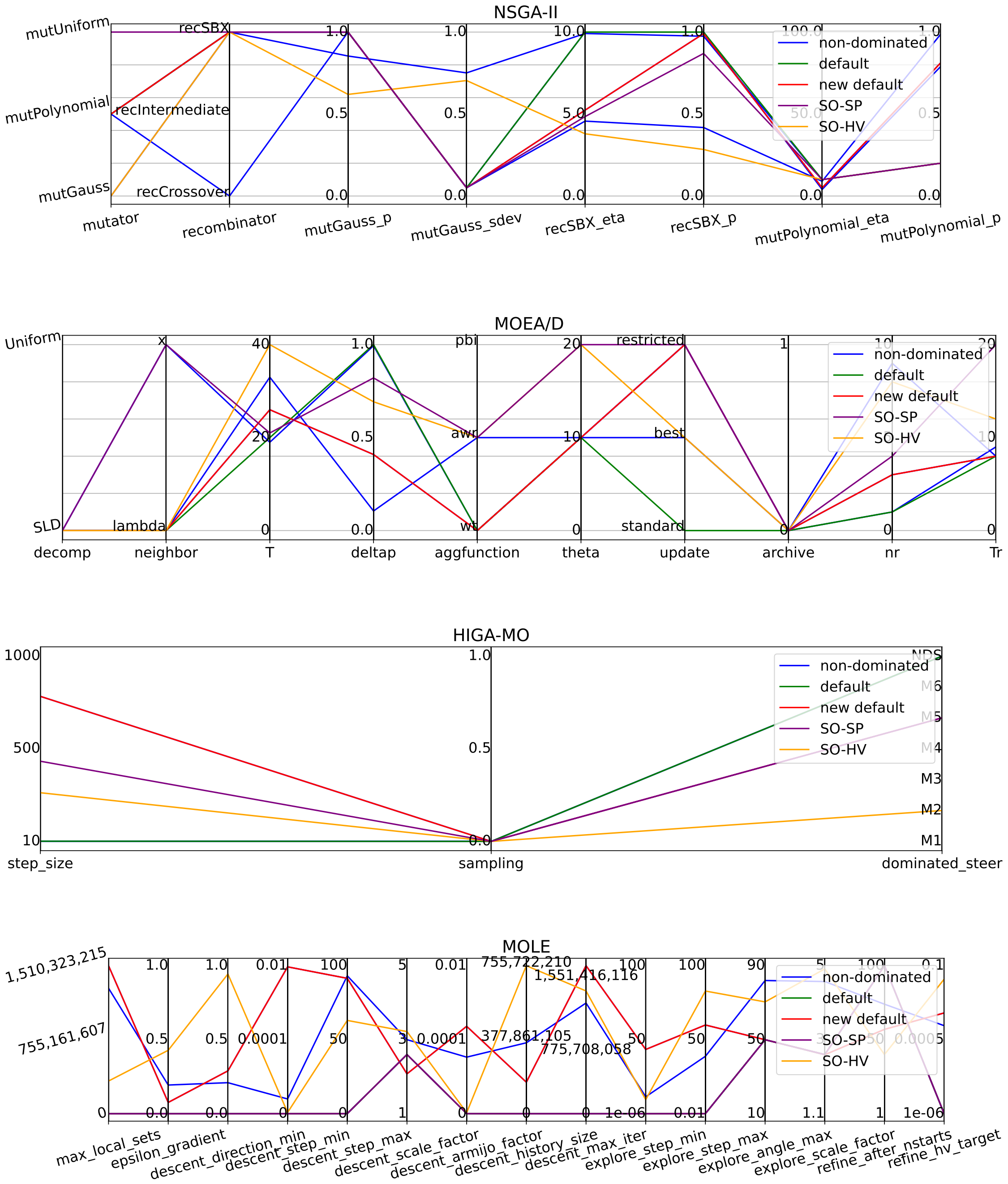


Figure 15 Summary of all non-dominated, default, SO, and proposed new default configuration parameters for NSGA-II, MOEA/D, HIGA-MO, and MOLE. The configurations correspond to the configurations of Figure 13. All configurations were found training and testing on all instances.

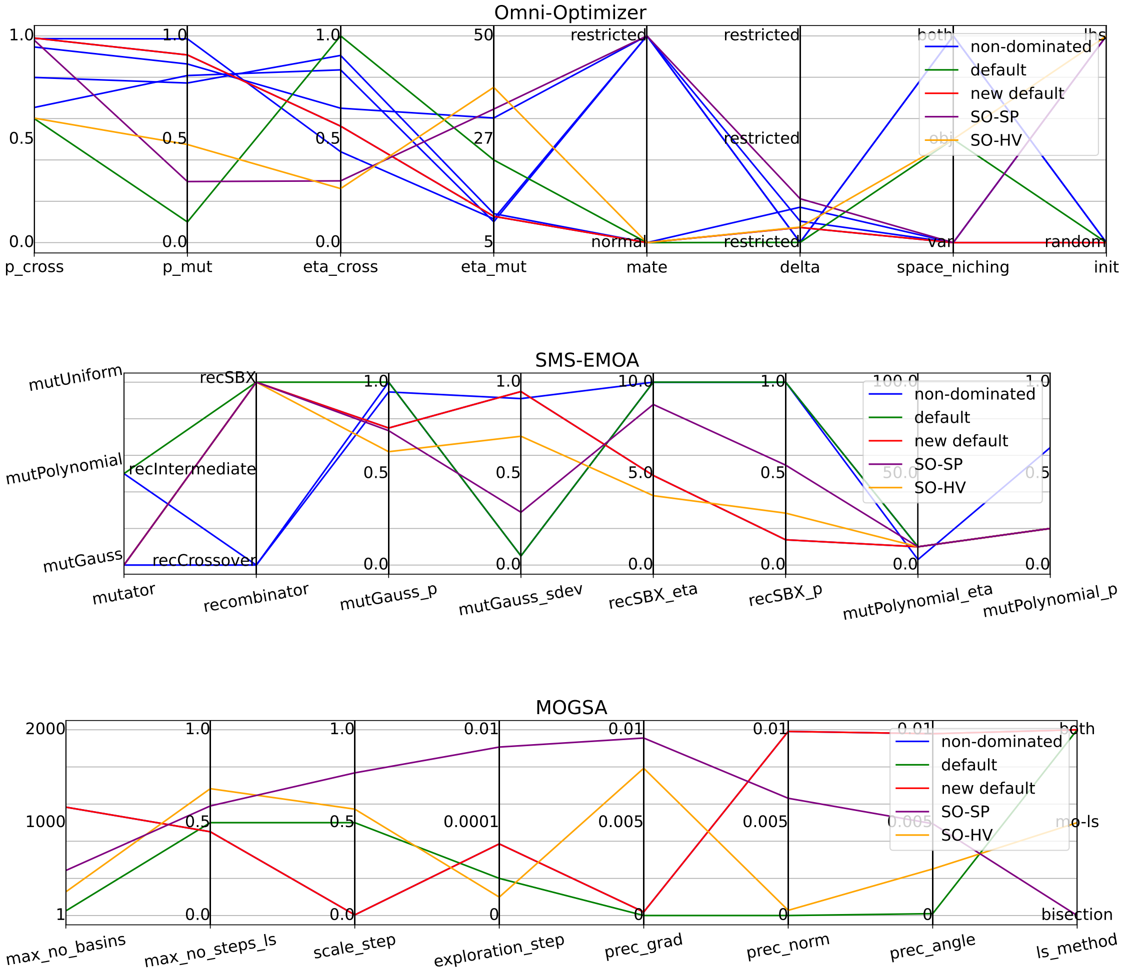


Figure 16 Summary of all non-dominated, default, SO, and proposed new default configuration parameters for Omni-Optimizer, SMS-EMOA, and MOGSA. The configurations correspond to the configurations of Figure 13. All configurations were found training and testing on all instances.

References

Beume, N., Naujoks, B., & Emmerich, M. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, *181*(3), 1653–1669. https://doi.org/10.1016/j.ejor.2006.08.008

Blot, A., Hoos, H. H., Jourdan, L., Kessaci-Marmion, M.-É., & Trautmann, H. (2016). MO-ParamILS: A Multi-objective Automatic Algorithm Configuration Framework. In P. Festa, M. Sellmann, & J. Vanschoren (Eds.), *Learning and Intelligent Optimization* (pp. 32–47). Springer International Publishing. https://doi.org/10.1007/978-3-319-50349-3_3

Brockhoff, D., Tusar, T., Auger, A., & Hansen, N. (2016). Using Well-Understood Single-Objective Functions in Multiobjective Black-Box Optimization Test Suites. https://doi.org/10.48550/ARXIV.1604.00359

Coello Coello, C. A., Lamont, G. B., & Van Veldhuisen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems* (2nd ed). Springer.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. https://doi.org/10.1109/4235.996017

Deb, K., Mohan, M., & Mishra, S. (2005). Evaluating the -Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions. *Evolutionary Computation*, *13*(4), 501–525. https://doi.org/10.1162/106365605774666895

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2005). Scalable Test Problems for Evolutionary Multiobjective Optimization. In A. Abraham, L. Jain, & R. Goldberg (Eds.), *Evolutionary Multiobjective Optimization* (pp. 105–145). Springer-Verlag. https://doi.org/10.1007/1-84628-137-7_6

Deb, K., & Tiwari, S. (2005). Omni-optimizer: A Procedure for Single and Multi-objective Optimization. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, C. A. Coello Coello, A. Hernández Aguirre, & E. Zitzler (Eds.), *Evolutionary Multi-Criterion Optimization* (pp. 47–61). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-31880-4_4

Deb, K., & Tiwari, S. (2008). Omni-Optimizer: A Generic Evolutionary Algorithm for Single and Multi-Objective Optimization. *European Journal of Operational Research (EJOR)*, *185*, 1062–1087.

Grimme, C., Kerschke, P., Aspar, P., Trautmann, H., Preuss, M., Deutz, A. H., Wang, H., & Emmerich, M. (2021). Peeking beyond peaks: Challenges and research potentials of continuous multimodal multi-objective optimization. *Computers & Operations Research*, *136*, 105489. https://doi.org/10.1016/j.cor.2021.105489

Grimme, C., Kerschke, P., & Trautmann, H. (2019). Multimodality in Multi-objective Optimization – More Boon than Bane? In K. Deb, E. Goodman, C. A. Coello Coello, K. Klamroth, K. Miettinen, S. Mostaghim, & P. Reed (Eds.), *Evolutionary Multi-Criterion Optimization* (pp. 126–138). Springer International Publishing. https://doi.org/10.1007/978-3-030-12598-1_11

Hansen, N., Finck, S., Ros, R., & Auger, A. (2009). *Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions* (Doctoral dissertation). INRIA.

Heins, J., Rook, J., Schäpermeier, L., Kerschke, P., Bossek, J., & Trautmann, H. (2022). BBE: Basin-Based Evaluation of Multimodal Multi-objective Optimization Problems. In G. Rudolph, A. V. Kononova, H. Aguirre, P. Kerschke, G. Ochoa, & T. Tušar (Eds.), *Parallel Problem Solving from Nature – PPSN XVII* (pp. 192–206). Springer International Publishing. https://doi.org/10.1007/978-3-031-14714-2_14

Hoos, H. H. (2011). Automated Algorithm Configuration and Parameter Tuning. In Y. Hamadi, E. Monfroy, & F. Saubion (Eds.), *Autonomous Search* (pp. 37–71). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-21434-9_3

Hoos, H. H., & Stüzle, T. (2004). *Stochastic local search: Foundations applications*. Morgan Kaufmann Publishers Inc.

Hutter, F., Hoos, H. H., Leyton-Brown, K., & Stuetzle, T. (2009). ParamILS: An Automatic Algorithm Configuration Framework. *Journal of Artificial Intelligence Research*, *36*, 267–306. https://doi.org/10.1613/jair.2861

Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential Model-Based Optimization for General Algorithm Configuration. In C. A. C. Coello (Ed.), *Learning and Intelligent Optimization* (pp. 507–523). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-25566-3_40

Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, *13*(4), 455–492. https://doi.org/10.1023/A:1008306431147

Lindauer, M., Eggensperger, K., Feurer, M., Biedenkapp, A., & Deng, D. (2022). SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization. *JMLR*, *23*(54), 1–9.

López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., & Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, *3*, 43–58. https://doi.org/10.1016/j.orp.2016.09.002

Nemenyi, P. B. (1963). *Distribution-free multiple comparisons* (Doctoral dissertation). Princeton University
Post-Hoc Test für Friedman-Test.

Qingfu Zhang & Hui Li. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, *11*(6), 712–731. https://doi.org/10.1109/TEVC.2007.892759

Rook, J., Benjamins, C., Bossek, J., Trautmann, H., Hoos, H. H., & Lindauer, M. (2023). MO-SMAC: Multi-objective sequential model-based algorithm configuration. In *Manuscript under review* (pp. 1–8).

Rook, J., Trautmann, H., Bossek, J., & Grimme, C. (2022). On the potential of automated algorithm configuration on multi-modal multi-objective optimization problems. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 356–359. https://doi.org/10.1145/3520304.3528998

Schäpermeier, L., Grimme, C., & Kerschke, P. (2022). MOLE: Digging tunnels through multimodal multi-objective landscapes. *Proceedings of the Genetic and Evolutionary Computation Conference*, 592–600. https://doi.org/10.1145/3512290.3528793

Schäpermeier, L., Kerschke, P., Grimme, C., & Trautmann, H. (2023). Peak-A-Boo! Generating Multi-objective Multiple Peaks Benchmark Problems with Precise Pareto Sets. In M. Emmerich, A. Deutz, H. Wang, A. V. Kononova, B. Naujoks, K. Li, K. Miettinen, & I. Yevseyeva (Eds.), *Evolutionary Multi-Criterion Optimization* (pp. 291–304). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-27250-9_21

Schede, E., Brandt, J., Tornede, A., Wever, M., Bengs, V., Hüllermeier, E., & Tierney, K. (2022). A Survey of Methods for Automated Algorithm Configuration. *Journal of Artificial Intelligence Research*, *75*, 425–487. https://doi.org/10.1613/jair.1.13676

Solow, A. R., & Polasky, S. (1994). Measuring biological diversity. *Environmental and Ecological Statistics*, *1*(2), 95–103. https://doi.org/10.1007/BF02426650

Ulrich, T., & Thiele, L. (2011). Maximizing population diversity in single-objective optimization. *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, 641–648. https://doi.org/10.1145/2001576.2001665

Wang, H., Deutz, A., Bäck, T., & Emmerich, M. (2017). Hypervolume Indicator Gradient Ascent Multi-objective Optimization. In H. Trautmann, G. Rudolph, K. Klamroth, O. Schütze, M. Wiecek, Y. Jin, & C. Grimme (Eds.), *Evolutionary Multi-Criterion Optimization* (pp. 654–669). Springer International Publishing. https://doi.org/10.1007/978-3-319-54157-0_44

Yue, C., Qu, B., Yu, K., Liang, J., & Li, X. (2019). A novel scalable test problem suite for multimodal multiobjective optimization. *Swarm and Evolutionary Computation*, *48*, 62–71. https://doi.org/10.1016/j.swevo.2019.03.011

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., & Da Fonseca, V. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, *7*(2), 117–132. https://doi.org/10.1109/TEVC.2003.810758