Optimally weighted ensembles of surrogate models for sequential parameter optimization

Echtenbruck, M.M.

Citation

Echtenbruck, M. M. (2020, July 2). *Optimally weighted ensembles of surrogate models for sequential parameter optimization*. Retrieved from https://hdl.handle.net/1887/123184

Cover Page

The handle http://hdl.handle.net/1887/123184 holds various files of this Leiden University dissertation.

Author: Echtenbruck, M.M.
Title: Optimally weighted ensembles of surrogate models for sequential parameter optimization
Issue Date: 2020-07-02

Chapter 4

Building Ensembles Using Convex Linear Combinations

In this chapter, the taxonomy specified in chapter 3 is used to develop a new ensemble method. This approach is studied fundamentally, by first evaluating ensembles of only two surrogate models in detail and then proceeding to ensembles with more surrogate models. Last, experiments are carried out on objective functions based on physical models. The results show to what extent combinations of models can perform better than single surrogate models and provide insights into the scalability and robustness of the approach.

As concluded from the insights gathered in Chapter 3, the preferred method of ensemble building for the considered task of optimizing expensive black box functions is supposed to be a combination of the predictions of several strong surrogates. It appears to be an interesting idea to linearly combine several models into more complex models. A convex linear combination of the models' predictions is both easy to calculate also for several heterogeneous models and comprehensive in terms of meaningfulness.

This Chapter is structured as follows. In Section 4.1 the ensemble building method is defined, thoroughly tested on different settings and analyzed for its strengths and weaknesses. Also, the influence of the RMSE on the behavior of the ensemble is regarded.
In Section 4.2 the method is extended for the use of three base models. To further extend the method for the use of more than three models, the resulting size of the search space has to be taken into account. This is done in Section 4.3.

Additional adjustments have to be made to ensure the functioning of the method with a larger set of models. This is done in Section 4.4. Lastly, in Section 4.5, the method is tested on a set of physical functions using different settings for the previously made adjustments.

4.1 Binary Ensembles

To investigate, whether and why combinations of models by linear convex combination could be beneficial it is reasonable to start with binary combinations of models, to enable an in-depth analysis and a better understanding of the functioning of such ensembles. The main questions are:

- Can such combinations of models compete with, or even improve on the performance of single models?

- Given the answer is positive, how can the observed behavior be explained?

In the following, convex combinations of models (CCM) will be referred to as ensemble model or CCM, while the original models will be referred to as base models.

We focus on positive weights since we do not want to select models that make predictions that are anti-correlated with the results. Given a weight α, where $\alpha \in \{0.0, 0.1, \ldots, 1.0\}$, the ensemble model can be defined as the linear combination of the models a and b as follows:

$$\hat{y} = \alpha \times \hat{y}_a + (1 - \alpha) \times \hat{y}_b \tag{4.1}$$

The prediction $\hat{y}$ of the ensemble model is a weighted sum of the predictions $\hat{y}_a$ and $\hat{y}_b$ of the base models a and b. With this definition of the combination, it is also ensured, that the total weight sums up to one.

To get an impression whether such a combination can be beneficial the two base models and resulting ensemble models are tested and compared against each other in a first simple experimental setup.

The experiment is carried out on an instance of a GLG function, which was introduced in Section 2.3, of dimension=4 featuring 40 Gaussian components (for additional parameters cf. Section 2.3, Table 2.1). A sample of points (design) is evaluated on the objective function. For the sampling of the points, a Latin hypercube design featuring 40 design points is generated.

The two base models are Kriging with exponential correlation function (referred to as a) and Gaussian correlation function (referred to as b). The models are evaluated by calculating the RMSE of the predictions made during a leave-one-out cross-validation on the 40 design points. Both base models are fitted to the data and then used to predict the left out point. The predictions $\hat{y}$ of the ensemble models are calculated as convex combinations of the predictions of the base models as specified in (4.1).

Since randomness has been induced into the experiment by using the Latin hypercube design, the evaluation process has been repeated 50 times in order to receive a meaningful result.

To get a first quick insight into the result data, for each repetition the rankings of the RMSEs have been calculated. The models with $\alpha = 0.6$, $\alpha = 0.8$ and $\alpha = 0.9$ dominate this comparison, each performing best 8 times out of 50. The base models, a and b, performed best only in four respectively two cases out of 50. Figure 4.1 shows the distribution of the ranks of each model.
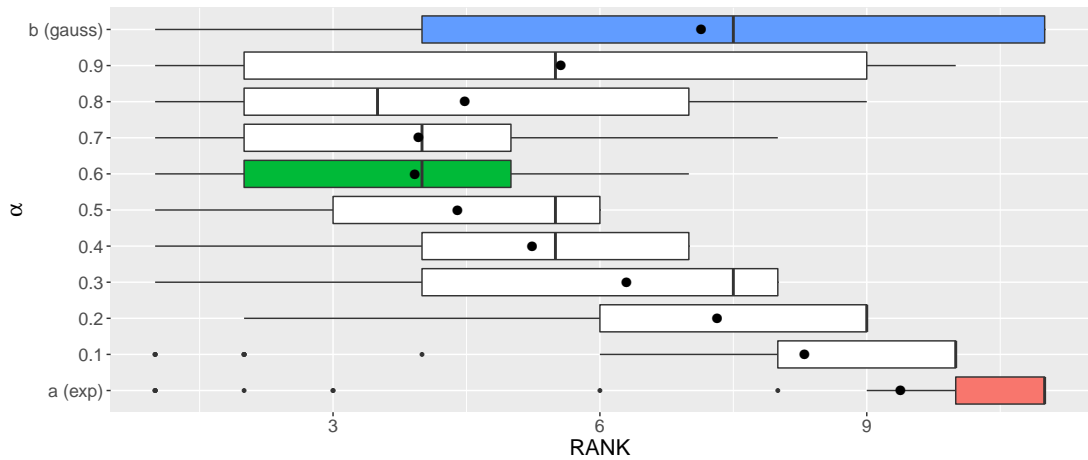
Figure 4.1: Boxplot over the repetition wise ranks of all models. The models are defined by an α-weighted linear combination of the two base models. The results of the base models are depicted on the outer rows and colored red (exponential kernel), and blue (gaussian kernel) respectively. The model combination chosen as best with $\alpha = 0.6$ is colored green. The mean value of each result bar is marked by a dot.

Model a (exponential) performed best in four of the cases but worst in 36. Model b (gaussian) shows a larger variation in its performance. It has been the best

performing model in two cases and performed worst in 14. In none of the cases an ensemble model was performing worst. Overall a parabolic tendency can be seen in the performance. This indicates that linear combinations of the models are indeed beneficial.

4.1.1 Detailed Analysis on Transparent Test Cases

It can clearly be stated that for this first experiment setup the combination of two models is beneficial for the overall prediction. In this section we will have a closer look at possible explanations for the successful result and will address the following questions:

- Are there problem features that encourage using ensembles?

- And is this result generalizable?

A larger number of experiments is carried out to analyze the performance and benefits of the ensembles.

As a consistent method for evaluating the performance and automatically choosing the best model, the following approach is proposed: Model-wise mean-, median- and 3rd quartile-values are calculated. The resulting values are ranked and then summed up to one final ranking. The model that achieved the lowest value is recommended as the best choice. In Figure 4.1 the model recommended as the best choice by this method is colored green. Applying this method, it might as well be more than just one model returned as best ensemble choice. In these cases, a decision has to be made about which model should be favored. Here, giving more importance to the median or 3rd quantile value means favoring a smaller distribution but taking the risk of larger outliers, while giving more importance to the mean value means having smaller outliers, but allowing for an overall wider distribution. Other criteria also might be taken into account.

For a better understanding of the underlying process and the strengths and weaknesses of the method, experiments are carried out on one-dimensional objective functions. The previous experimental setup has been preserved, except for the change of the objective functions' dimensionality.

Figure 4.2a shows exemplary results from these experiments. To allow for an easier visual comparison, the RMSE's of the models have been repetition-wise scaled to values from zero to one. The plot on the left hand side depicts these values. Applying the rule defined in Section 4.1.1 the model obtained by a linear

(a) Results on a GLG objective function

(b) Results on a piecewise assembled objective function.
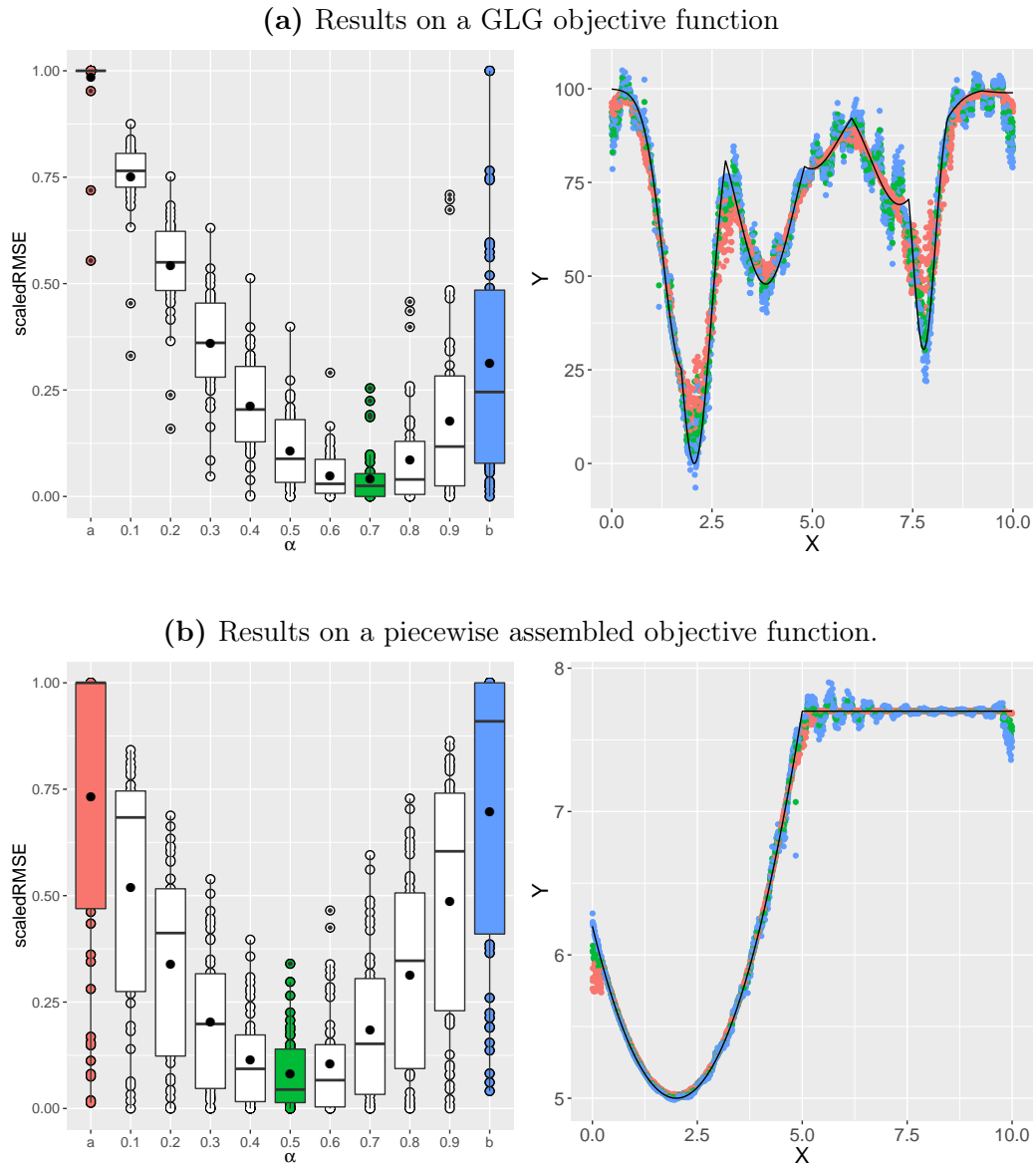
Figure 4.2: The plots show the results on two different one-dimensional objective functions. Each plot on the left shows the repetition-wise scaled RMSE's for each model. The α value defines the weight for the linear combination, the model that has been chosen as best is colored green. On the right hand side all predictions done during the leave-one-out cross validation for the base models and the best model are plotted against the objective function.

combination with $\alpha = 0.7$ is marked as best choice.

The plot on the right-hand side shows the predictions of the models versus the objective function. In this plot, only the two base models and the best ensemble are regarded. Each dot marks a single prediction made during the leave-one-out cross-validation. In every repetition, each model makes one prediction for every point of the design. This results in a total of 2000 prediction values for each model (40 design points × 50 repetitions).

As can be seen in the plot, the predictions of the model a (exponential kernel function), marked by red dots, seem to smooth the objective function: straight segments are well met while curved segments are smoothed out.

The predictions of the model b (Gaussian kernel function), marked by the blue dots, show signs of overfitting. Again, straight segments are well met, but when approaching local extrema, the predictions start to oscillate. So the linear combination of both predictions averages positive as well as negative outliers of base models. This seems to provide some benefit to the overall experiment outcome.

To confirm this assumption two additional experiments are carried out. For these experiments two objective functions are specified featuring corners that are not continuous differentiable. For one experiment a triangle objective function is used while the other features a piecewise assembled objective function. Figure 4.2b depicts the results on this function. A parabolic graph is joined together with a straight line graph. Special focus is laid on the joint of both function parts and on the straight part. Both base models succeed in describing the parabolic part of the function, but while base model a (red) fails at the borders and at the junction point, base model b (blue) is not able to describe the straight part of the function. The ensemble (green) benefits from this difficulties. We again find a strong parabolic tendency in the boxplot. Both base models have a rather large variance in their performance. The ensemble model marked as best choice has a smaller variance and performed better than the base models in nearly all cases.

The results on the triangle objective function happened to show a clear tendency towards base model b, which clearly outperformed base model a and thus was chosen best. This also is a good result, since it is desirable that the method only chooses an ensemble over a base model if it is actually better.

4.1.2   Detailed Analysis on Single Predictions

To better understand what is happening, we take a closer look at convex linear combination itself and how it influences the fitness of the resulting model. The main questions are:

- In what circumstances can the prediction of a CCM be better than the prediction of the base models?

- Is it possible that the prediction of a CCM is worse than the predictions of both base models?

To answer these questions, we consider two base models and related CCMs making a single prediction. For reasons of simplicity, we assume, that the real function value is zero and that the base models are making different combinations of prediction errors.
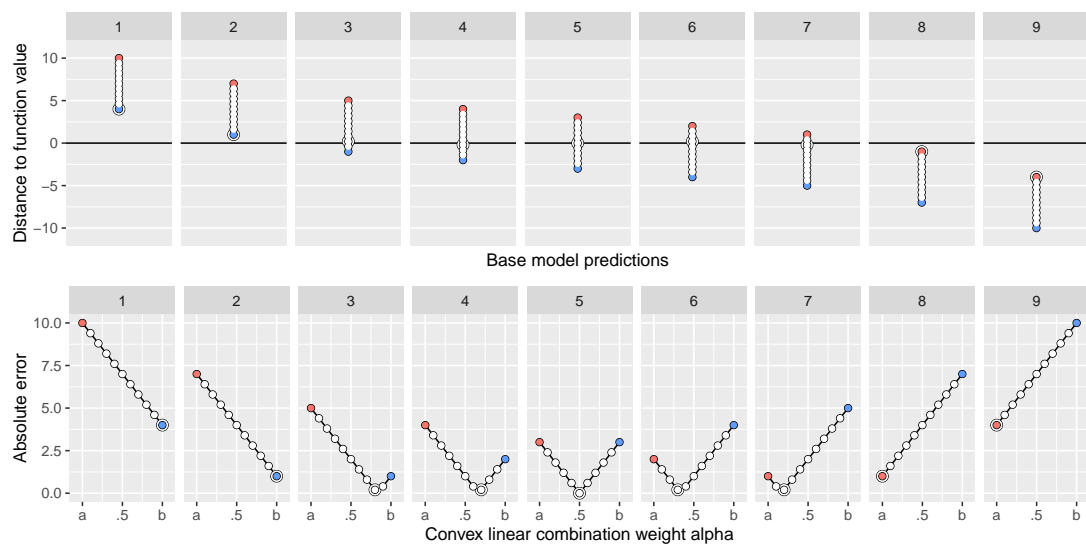
Figure 4.3: The upper row plots show the predictions of two base models and its related CCMs for a single point. The true function value of zero is marked with a line. In the lower row the related RMSEs are given. The best prediction and the related smallest prediction error are marked with an additional circle. It can be seen that in some cases the CCM performs better than both base models, but it never performs worse than both.

Figure 4.3 displays such a situation. In the upper row, the colored points mark the predictions of the base models, while the white points mark the predictions

obtained through convex linear combination. The true function value is set to zero, displayed by the horizontal line.

The lower row depicts the related resulting RMSEs. The best prediction and its related error are marked with an additional circle.

Since we are only considering a single prediction, the resulting RMSE equals the single prediction error. Thus, as assumed before, if the base models make opposing prediction errors, the prediction of the CCM will be better than the predictions of the base models (cf. Figure 4.3, Columns 3-7).

Whereas, when both base models make the same kind of prediction error, the prediction, as well as the prediciton error, of the CCM lies inbetween the predicions and the errors of the base models. (cf. Figure 4.3, Columns 1,2,8 and 9).

All other possible relations between prediction errors and resulting RMSE can be easily derived from this. So would a change of the distance between the base models predictions result in a steepened slope, while a larger distance between the two predictions and the real function value would result in a larger absolute error. The most important information we can read from these plots is that for a single prediction the convex linear combination of the predictions cannot be worse than the prediction of the weaker base model, independent of the weight chosen for the combination. Moreover, for some values of α the combined prediction can be better than the predictions of both base models given, that one model overestimates the actual function value while the other model underestimates it. In the experiment (cf. Section 4.1) this happened in 649 out of 2000 cases.

However, when evaluating the ensemble built by the proposed method of building convex linear combinations, this is not done based on a single prediction but on a set of predictions. Also, calculating the RMSE of a single prediction only yields the simple prediction error. Questions that also have to be considered are:

- Do these findings made for single predictions still apply for multiple predictions?

- How does the RMSE influence the evaluation of the models' overall performance?

To answer these additional questions, the analysis is extended to two predictions. Figure 4.4 displays several possible combinations of predictions for such a setup. Figure 4.4a shows a set of predictions where the one model always predicts a larger value than the other model, while in Figure 4.4b one model always returns a larger prediction for one point and a smaller prediction for the other point. In both plots, like before, the upper row shows the predictions, base model predictions are colored blue or red, and predictions of the linear combinations are colored

(a) One base model always predicts a smaller value than the other
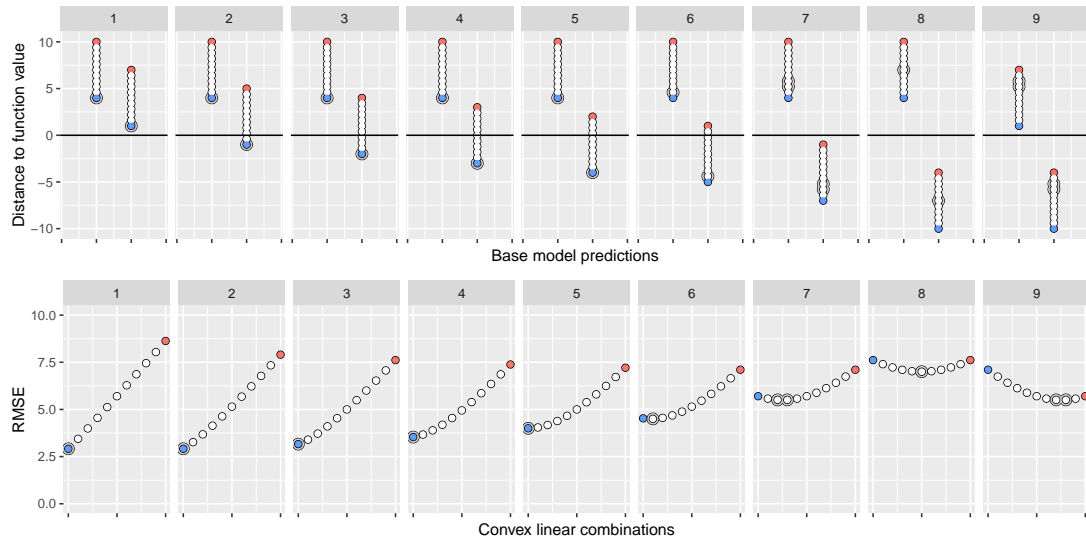
(b) Each model predicts larger in one and smaller in the other case
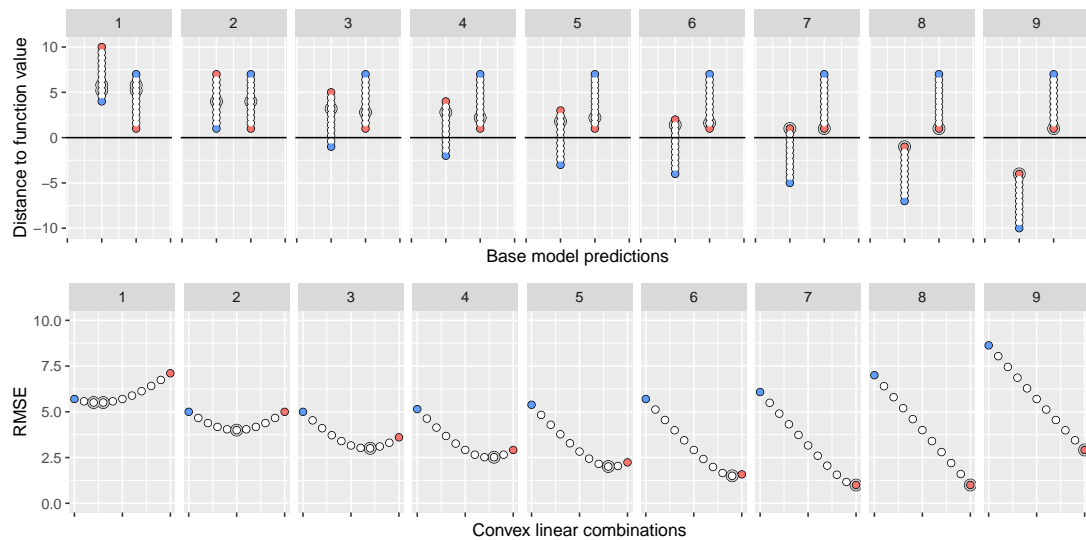
Figure 4.4: Both plots show a set of combinations of two predictions made by two base models (red and blue) along with the resulting predictions made generated through convex linear combining using a fixed step width of 0.1. The second row of each plot displays the related RMSE values resulting from the combination of the two predictions.

white. The lower row displays the relating RMSE values resulting from the two predictions.

We can see from these plots, that it does not suffice for the ensemble prediction to be better than the base models, when in one of the two cases the prediction spans the actual function value, while the combined predictions of one of the base models are better in terms of RMSE (cf. Fig. 4.4a items 2-5, 4.4b items 6-7). The combined prediction error, in terms of RMSE, rather benefits from opposing errors in general (cf. 4.4a items 6-9, 4.4b items 1-5).

As mentioned before, the CCM can make a better prediction on a single point only if the predictions of the base models span the true function value.
However, regarding more than one prediction, this evaluation is influenced by the characteristics of the RMSE. The overall fit of the CCM, in terms of RMSE, can be better than both base models also, when none of the base models predictions is spanning the true function value. For such an evaluation it is sufficient when one model returns a larger prediction than the other model for one point, and a smaller prediction for the other point.

Noticeable is also, that in contrast to the plots shown in Figure 4.3, the points marking the RMSE values in most of the cases lie on a parabolic curve. This might indicate that the search space, generated by the RMSE could also be convex. Interpreting the RMSE as a function describing a form of mean squared distance would support this claim. However, this assumption is based on observations only and would require further analysis.

4.1.3   Conclusion

The insights gained so far are promising and seem to make convex linear combinations an ideal choice for combining models. So far recognized advantages are:

- Due to the convex linear combination that is used for combination, the ensemble cannot perform worse than the weakest base model.

- The ensemble can perform better than the base models when compensating opposing prediction errors.

- A CCM is favored over a base model only if the overall fit of the ensemble model is actually better (in terms of RMSE), than the overall fit of both base models.

- The nature of the combination is intuitive and interpretable.

- The linear convex combination of predictions for a given set of weights is easy to compute.

4.2 Ternary Ensembles

Up to this point, the first experiments have been carried out, and the method showed promising results. However, the question arises if this method is scalable and generalizable. In this section the experiments are extended to a larger scale: The dimensionality of the objective functions is increased, and the method is adapted to enable the combination of three models. As before, Kriging models with different kernels are used, but now a third model using the spline correlation function is added.

For the linear combination of three base models also three weights are needed, that sum up to one as specified in (4.2).

$$\alpha, \beta, \gamma \in \{0.0, 0.1, \dots, 1.0\}, \quad \alpha + \beta + \gamma = 1 \tag{4.2}$$

Retaining the step size of 0.1 for the linear combinations results in 66 ensemble models.

As a first step towards problems of higher complexity, the dimensionality d of the objective function is set to 4. However, this change alone is not sufficient to gain a larger complexity, since without adjusting the number of Gaussian components used for generating the objective function, it rather gets less complex. Thus the number of Gaussian process trajectories is adjusted to forty times the dimension.

With increasing the complexity of the objective function also the size of the design should be adjusted to gain reasonable results. Desirable is a design that is preferably small yet allows for a proper fit of the models. To achieve this an experiment with varying design sizes is run beforehand. For each size, the fit of the base models to the objective function, in terms of RMSE, is compared to a simple mean predictor. The design size chosen for the experiment is that with the smallest design size, such that all base models are performing significantly better than the baseline predictor.
For the experiment presented in this Section, this occurred at a design size of 160.

Figure 4.5a shows an exemplary experiment result using three base models. The depiction of the result corresponds to the previously shown boxplots. The ternary plot also shows parts of these data but allows for a more intuitive interpretation than a boxplot would do.
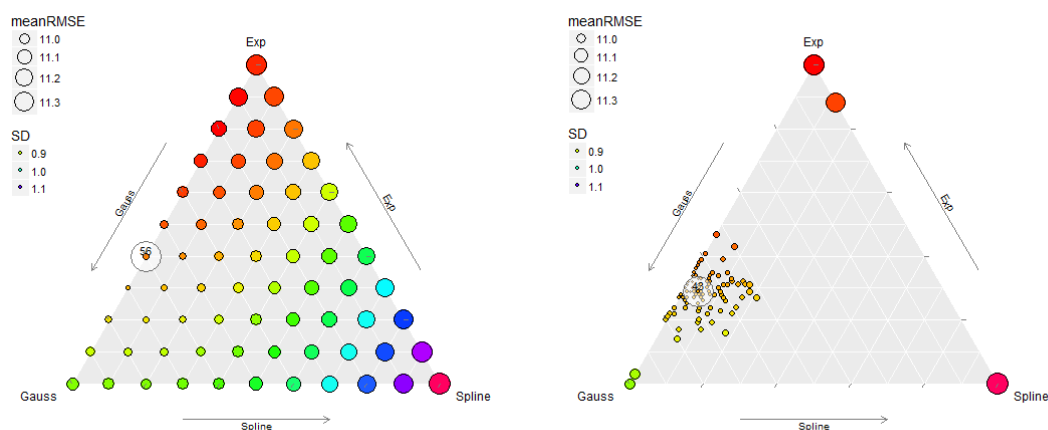
(a) The optimal linear combination has been found by a complete evaluations of all linear combinations using a fixed step size of 0.1.

(b) The optimal linear combination has been searched with a simple (1+1)-Evolution Strategy with 1/5th success rule (cf. [108]).

Figure 4.5: The plots show the results of the experiment set up with three base models. Each circle depicts the performance results for one model. The three base models are located on the corners of the triangle, models gained by linear combinations of only two models are located on the outer border. Circles on the inner region of the area show the results for models that were gained by linear combinations of all three base models. The size of the circles denotes the mean RMSE value, the color the standard deviation. The model proposed as best choice is marked by an additional white circle.

Each circle in the plot displays the performance of a single model. The base models are positioned on the corners of the triangle, models on the outer border are built from a linear combination of only two of the three base models. All circles on the inner area of the triangle are linear combinations of all three base models; its proximity to the corner relates to the share this model has to the linear combination. The size of the circle corresponds to the mean RMSE that the model

has achieved, the color to the standard deviation of the RMSE values. The best choice model, as defined in Section 4.1.1, is marked by a white circle.

In most of the experiments carried out, the model that is automatically chosen as the best performing model is located on the inner area of the triangle. For some of the experiments, the best model is located on the outer border of the triangle. From these results, it can be deduced that the method not only meets the demands but also performs better than a single base model only.

4.3 From Exhaustive Search to an Evolutionary Strategy

The approach proposed so far proved its functionality using a small set of homogeneous base models. Still, the underlying goal is to evolve a system that can handle a large set of heterogeneous base models. In this section, preparations are taken to enable the method to also handle a considerably larger search space. To achieve this, the method is adapted to use an evolution strategy instead of performing an exhaustive search of the complete search space.

When increasing the number of available base models, also the number of possible discretized convex combinations between these base models grows exponentially. The number of resulting convex combinations can be calculated using formula 4.3. The formula recursively counts the possible combinations of weight distributions under the premise that the available weight units have to be completely distributed among the available models, but also allowing that a model receives no weight.

$$f(s, r) = \sum_{i=0}^{r} f(s - 1, r - i), \quad f(s, 0) = 1, \quad f(1, r) = 1 \tag{4.3}$$

The relation between the number of models, the step size for the discretized convex combinations and the resulting number of linear combinations can be expressed as a function of s, the number of models, and r, the reciprocal of the step size or the number of available weight units respectively. Recursion stops if either only one model or no distributable weight unit remains. Since weights have to sum up to one, the remaining model has to take the remaining weight. Alternatively, if no weight remains, the remaining models receive no weight. In both cases, only one possible option remains.
Using three base models and a step size of 0.1 as defined in (4.2) results in

4. BUILDING ENSEMBLES USING CONVEX LINEAR COMBINATIONS

$f(3, 10) = 66$ linear combinations. Now thinking of combinations of 10 base models already results in $f(10, 10) = 92,378$ linear combinations.

The complexity of the search space, when increasing the number of models, quickly gets too large to do a complete evaluation of all possible convex combinations with a fixed step size of 0.1. Moreover, the restriction to a fixed step size of 0.1 might be too rough.

Looking at previous findings, the function that describes the performance of the models built by convex combinations up to this point only showed convex, unimodal characteristics. This seems to be expectable due to the nature of convex combinations and the use of the RMSE as the fitness function. We expect the function to show this characteristic also when combining a larger number of models.

Thus, instead of a complete evaluation of all linear combinations, an optimization step is introduced to find the best combination instead of performing an exhaustive search on a fixed grid. The allowed weights are extended to a precision of two decimal places, allowing for $f(3, 100) = 5,151$ possible ensemble combinations. Since the area around the optimum tends to build a plateau smaller differences in the weights distribution seems to be negotiable. This restricts the possible search space to a reasonable size without losing the possible best solution.

For the optimization step a (1+1)-evolutionary strategy (ES) with 1/5 success rate for step width adaption is introduced [109, 110]. Building on the assumption that the regarded search space is supposedly but not surely unimodal and convex this would be a reliable and robust search algorithm that performs well on a search landscape as assumed but can also handle more difficult search spaces. However, other search strategies like Particle Swarm Optimization ([111]) would also fulfill these criteria.

Each individual of the population, representing an ensemble, is distinctly defined by its weights vector $\mathbf{v} = (\alpha, \beta, \gamma)^T$. For the mutation of the parent individual a vector of three random samples of a normal distribution function with standard deviation according to the actual step width σ is added to the related weights vector. Here, σ is initialized with $\sigma_{init} = 0.16$ and adjusted by the factor 0.9 according to the 1/5 success rate rule.

To ensure that the offspring individual still meets the requirements needed for a valid weight vector, the resulting vector $\mathbf{v}$ is adjusted in three steps:

1) If the smallest of the weights is smaller than zero ($\min(\alpha, \beta, \gamma) < 0$), this value is subtracted from the vector $\mathbf{v}$ ($\mathbf{v} \leftarrow \mathbf{v} - \min(\alpha, \beta, \gamma)$) to ensure that

all weights are positive.

2) In a next step, the vector $\mathbf{v}$ is scaled such, that the weights sum up to one $(\mathbf{v} \leftarrow \mathbf{v}/(\alpha + \beta + \gamma))$.

3) Finally, the values α, β, γ are rounded to two decimal places such, that the premise $\alpha + \beta + \gamma = 1$ is not violated.

The modified ensemble building algorithm is run on the same experimental setup as already used in Section 4.2, to allow for a comparison of this method to the previously applied method using exhaustive search. Figure 4.5b displays the result of the optimization.

For this experiment we allowed a maximum of 100 individuals to be evaluated. Within these bounds already the 43rd evaluated individual has been the best individual found in this run, this individual is marked by a white circle. Table 4.1

Model	Ensemble weights			mean RMSE	SD
	Gauss	Exp	Spline		
Gauss	1	0	0	11.036	0.92
Exp	0	1	0	11.349	0.82
Spline	0	0	1	11.396	1.19
by exhaustive search	0.6	0.4	0	10.95	0.85
by optimization	0.67	0.29	0.04	**10.945**	0.87

Table 4.1: Comparison of the Model results in more detail. The left column names the considered models, in the first group the base models alone, in the second group the ensembles found by exhaustive search versus the ensemble found by optimization. The second column gives the weights on the base models. Of course the base models only use one model, marked by a 1, the other models have no weight. Whereas the ensembles combine two or three models respectively. The last columns give the mean and standard deviation of the RMSE over all repetitions. The result achieved by optimzation has the best result.

shows the precise performance values for the ensembles found in comparison to the base models used. From this, it can be said, that the ensemble found by optimization is not only able to compete with the result received from exhaustive search but also is marginally better. Thus, the optimization algorithm not only can receive comparable results in comparable time but also benefits from the ability to handle a smaller step size.

4.4   N-ary Ensembles

With Section 4.3 the basis is provided for setting up a large system of heterogeneous models. In the following, the set of base models is extended to the intended size and experiments are carried out to ensure that the performance of the method does not suffer from this change. The leading questions are:

- Can the ensemble method handle the heavily extended search space for the best model combination and still find a beneficial combination?

- Are further adjustments of the ensemble building method needed due to the extension of the model set?

To answer these questions the set of base models is extended, and the method is tested. To ensure, that the method can handle the increased number of base models it is also compared to the performance of the algorithm with only three models in the set. If the method can handle the higher number of base models, a solution should be found that is at least as good as the solution found when using only three models.

4.4.1   Extending the Base Model Set

To obtain a large set of homogeneous base models all models from the SPOT package are added to the system, using their default options. This results in a system containing 13 heterogeneous models:

- Gauss, Exp, and Spline refer to the same Kriging models that are already used in the previous experiments.

- Tree, Earth, LM, RandomForest, Tgp, Esvm, MLP, neuralnet, Qrnn, and RFMlegp are now newly added to the system (cf. Chapter 2).

The set of models here is rather diverse, chosen without preselecting appropriate models and the models are not tuned but used with default settings. Thus, their performances also have considerable variances. Due to these variances in their performances the design size of the experiment is enlarged to 200 points. Herewith, the previously formulated demand, that all models should perform considerably better than the baseline predictor is relaxed such, that at least about half of the base models fulfill the demand.

Figure 4.6 shows the performances of the models on the objective function, in comparison to a simple mean predictor. It can be seen, that from the set of
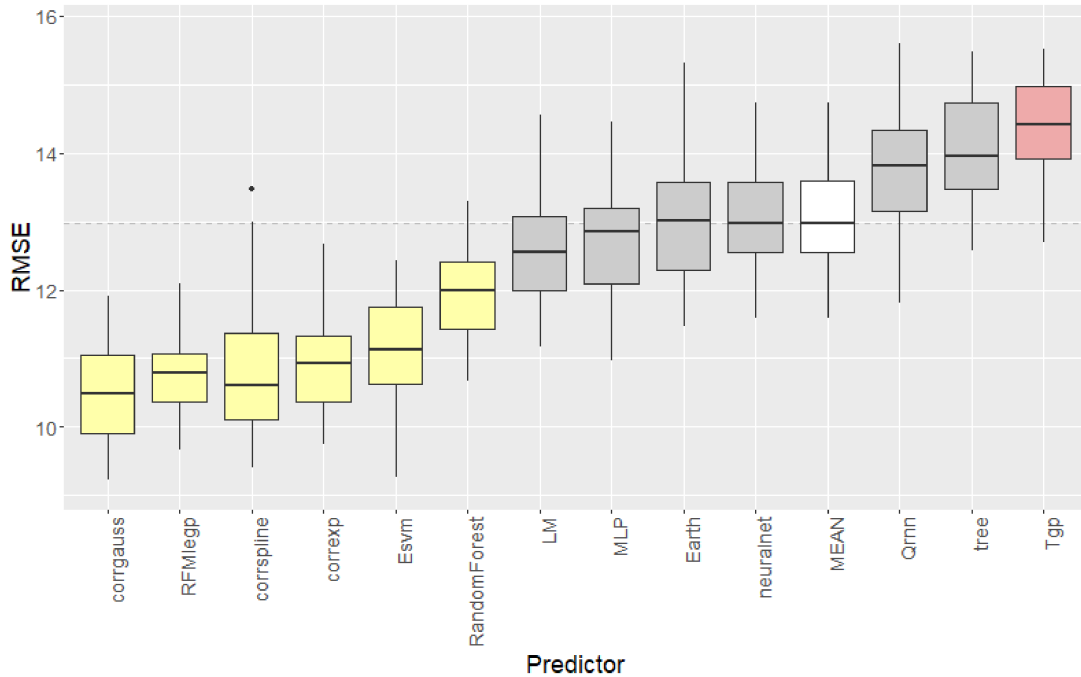
Figure 4.6: Comparison of the performance of the base models chosen for this experiment setup. Models are ordered by performance in terms of RMSE. Six base models perform significantly better than the mean predictor, these are shown in light yellow. All other models perform comparable (gray) or even worse (light red) than the mean predictor.

13 base models only six perform significantly better than the mean predictor, these models are colored light yellow. The remaining models perform comparable (gray), or even worse than the mean predictor (light red).

However, the size of the base model set and this diversity in the performances makes it harder for the search algorithm to find a good or even better solution. This fact makes this experimental setup an even more interesting setup to ensure that the search algorithm, also for larger model sets, is capable of finding a better solution if one exists.

4.4.2 Adaptation of the Evaluation Method

With this size of design, especially when evaluating more than three base models, the use of leave-one-out cross-validation is no longer practicable. Particularly with regard to problems of higher complexity an evaluation method that requires a

fixed number of model fitting processes would be preferable. Thus, for evaluation of the base models 10-fold cross-validation is applied.

Though these changes seem to be petty, they may implicate a change in the weights of the best ensemble solution. To serve as a reference, the previous experimental setup (cf. Section 4.2 and 4.3) is repeated using the larger design size of 200 points per repetition. Experiments are carried out once performing a leave-one-out cross-validation and once a 10-fold cross-validation for the evaluation of the base models. In both cases, the same set of designs is used, so that any differences in the results from the complete evaluation of these experiments solely originate from the change in the cross-validation.
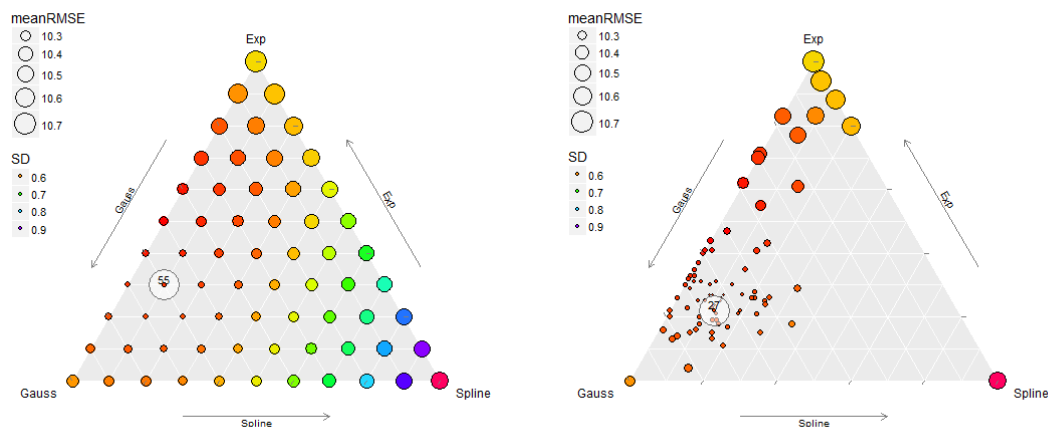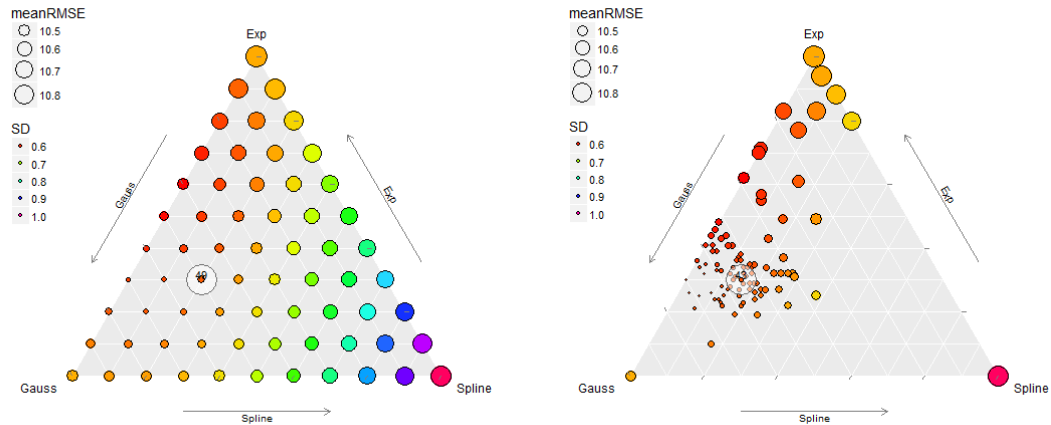

Figure 4.7: Results of the experiment carried out on a GLG objective function with a design size of 200. Base Model evaluation is done by leave one out cross-validation

The results for the experimental setup using leave-one-out cross-validation are shown in Figure 4.7. This experiment differs to the experiment presented in Section 4.2 Figure 4.5 only in the increased size of the design. In comparison to this experiment result, the best ensemble solution moved a little towards the inner area of the triangle.

Figure 4.8 shows the results for the experimental setup using 10-fold cross-validation. Here again, the best solution moved a little farther towards the inner area of the triangle.

From these results, it can be assumed that both, the increased design size as well

Figure 4.8: Results of the experiment carried out on a GLG objective function with a design size of 200. Base Model evaluation is done by 10-fold cross-validation

as the adjustment of the evaluation method, has an impact on the result of the experiments. Still, all experiments agree that indeed an ensemble solution exists that performs better than the base models solely. The result depicted in Figure 4.8 also serves as a reference for the following experiments using a large set of base models.

4.4.3 Performance Test Using a Large Base Model Set

With the preliminary experiment, it is already shown, that for the experiment setup carried out in Section 4.4.2, using the three kriging base models only, an ensemble combination exists, that performs better than these three models alone (cf. Figure 4.8).
Thus repeating this same experiment, only extending the set of base models while knowing that none of the new models performs better than the best of those three (cf. Figure 4.6), the search algorithm should be capable to at least reproduce a comparable solution or even find a better one. However, the ensemble algorithm as presented so far is not able to find neither a comparable, nor any ensemble solution that uses at least two of the base models. It presents Gauss alone as the best choice.

Figure 4.9 shows the progression of weight combinations that are evaluated during the search. Each line represents a single ensemble combination; the color of the

line corresponds to the optimization step it has been generated in. Each of the base models has been evaluated preceding the optimization. Therefor thirteen light red lines are shown, each featuring only one peak giving full weight to this base model. The remaining lines, representing the progress of the distribution of the weights during optimization, and therefore the different ensemble combinations tested, are entirely arbitrary. None of the random mutations seems to yield an improvement to the known best solution. The best solution that was found during the optimization gives all weight to the Gauss model. The bold white line presents this solution. The bold black line represents the best combination that has been found using the three kriging models only (cf. Figure 4.8), suggesting a mixture of these three base models, which the algorithm was not able to find.
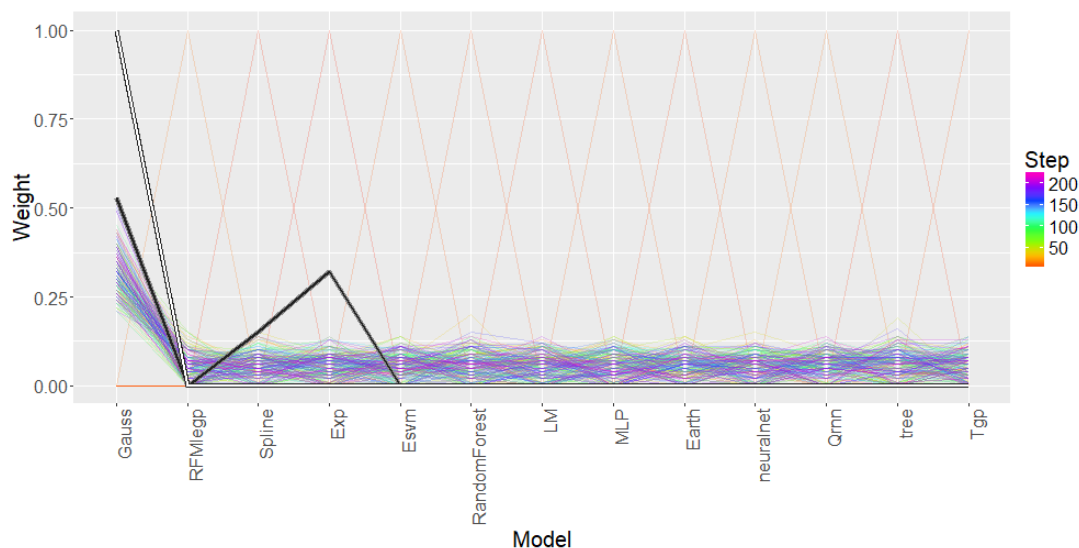
Figure 4.9: Developement of the ensemble weights during optimization. The black line marks the best ensemble found when only three models were part of the set. The white line marks the best solution found in this experiment run. The algorithm is not able to find the already known, better solution.

A possible explanation for this behavior is the number and performance of the base models added to the system. So far not only was the search space smaller but also the performance of the base models used was comparably strong. Now 13 heterogeneous base models are part of the system, only six of them are performing significantly better than the mean predictor.

This is aggravated by the fact that the search strategy performs a mutation of

the best solution by adding a random vector to the weights vector, which allows for small changes of all weights in every step (cf. 4.3). But with seven out of 13 models performing comparatively bad, the probability is high that with every mutation weights are adopted that worsen the overall performance of the new individual. Hence, the new individual generated has only very slim chances to perform better than the parent.

Possible solutions to overcome these problems might be:

1 *Restrict the number of genes that may be changed in one mutation step.* In a case where many of the base models are not beneficial for the system the chances of finding a beneficial mutation are getting larger.

2 *Minimize the overall search space by preselecting the most promising base models.* Before starting the search for the best weights, the set of models is reduced to the best performing of the models by a predefined rule.

3 *Start with a search space of a smaller dimension and enlarge during the search.* Starting the search, only using a small number of base models and then adding further base models as the search goes along, does not imply too much a priori intervention.

In the following subsections, these possible solutions are introduced in more detail and tested on the previously specified experiment setup. Possible advantages and disadvantages of the methods are discussed, and finally, all approaches are compared against each other.

4.4.4 Restriction of the Mutation

Based on the assumption that the method is not able to find a better offspring because of the rather large number of poorly performing base models in the set, the mutation is adapted to allow only a smaller number of weights to be changed in every mutation step. Doing so, it is more probable that less or none weight is given to the weaker models and with this, the probability of doing a beneficial mutation is increased.

To achieve this, a weights vector $\vec{v} = (v_1, v_2, \ldots, v_n)^T$ of n random samples of a normal distribution function with standard deviation of σ is drawn as before. However, to meet the demands of only a restricted number of weights to be changed, the smallest of these values are reset to zero, before being added to the parents' weights vector, such that only the required number of genes is changed.
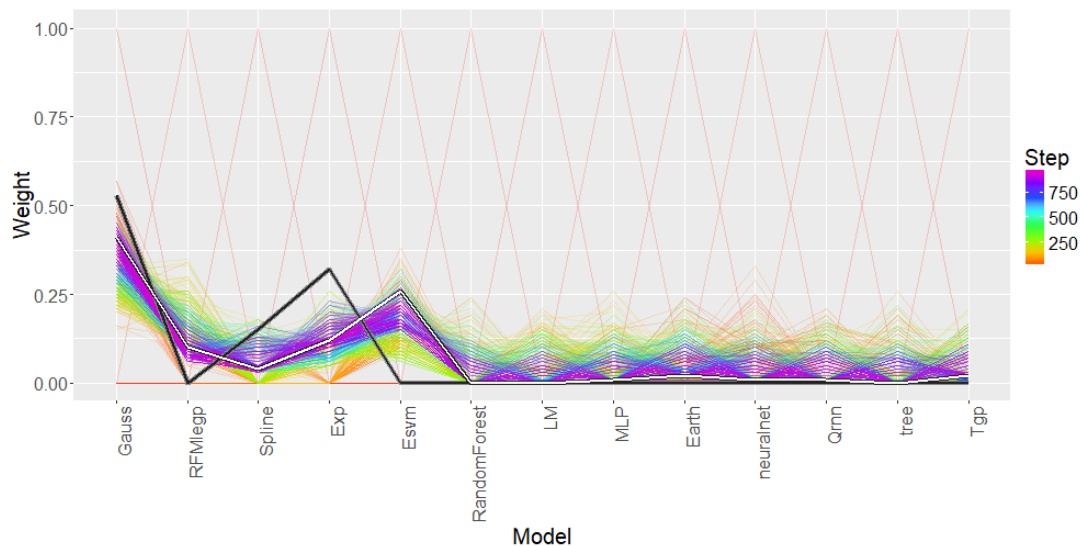
Figure 4.10: Developement of the ensemble weights during optimization using a
constrained mutation step. Again, the black line marks the ensemble already found
previously with only three base models available. The white line marks the best
solution found during this optimization.

The resulting weights vector is then adjusted to meet the demand, to sum up to
one as before (cf. Section 4.3). The approach is tested allowing three weights to
be changed in each mutation step.

Fig. 4.10 shows the development of the weights distribution during the optimiza-
tion process applying this approach. The lines show much more structure than
before. So can be read from the column for the Gauss model that for this model
a broad range of possible weights (about 20-50%) is tested in the earlier steps
of the optimization and then slowly settled down to weights around 38% in the
later steps of the optimization. Whereas for example for RandomForest a range
of weights is tested, but none of these is an improvement to the known best,
such that this model ends up with no weight. Other than the basic ES used so
far, this method can find combinations that are better than the single best base
model.

4.4.5   Preselection of Models

The main idea of this approach is to reduce the search space by choosing a
reasonable set of base models prior to the actual optimization. Models are selected

based on their single performance during the initial evaluation of the base models. With this preselection, the search space, in terms of the number of base models, may be reduced to a smaller number of base models excluding weaker models according to a predefined exclusion criterion.

This criterion might heavily depend on the objective function and has to be chosen carefully. Choosing too restrictive a criterion might a priori exclude models that would have been beneficial contributors to the ensemble while choosing too lenient a criterion might hamper the optimization algorithm. Assuming that the algorithms inability to find a better solution arises from the large number of models performing worse than the mean predictor leads to the conclusion that the comparison against the mean predictor might be a good indicator.
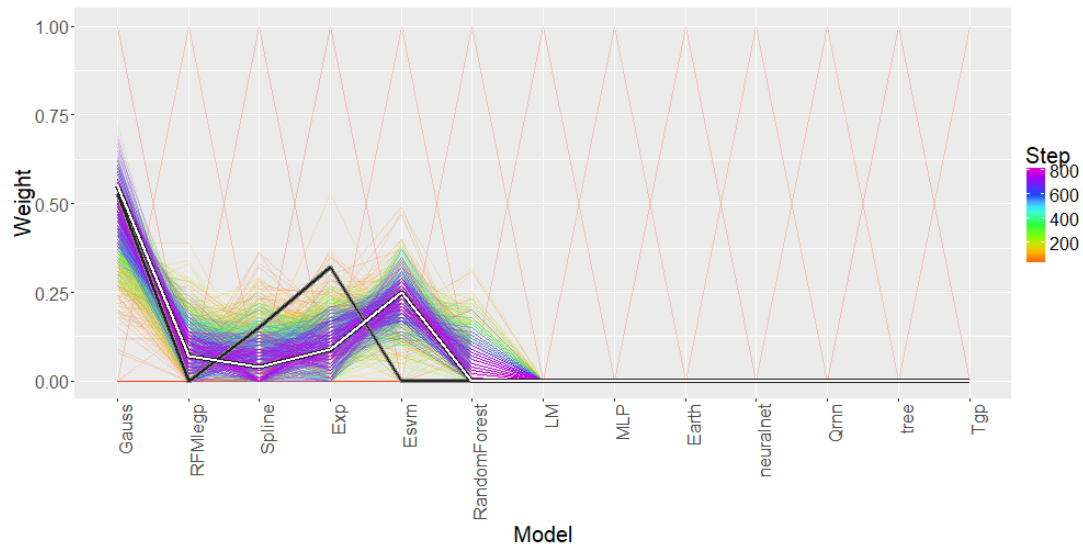
Figure 4.11: Developement of the ensemble weights during optimization using a preselected set of models only. The black line marks the ensemble already found previously with only three base models available. The white line marks the best solution found during this optimization.

This adaptation of the original ES is realized such, that models are chosen by their performance in comparison to the mean-predictor. Models with a third quantile performance, in terms of RMSE, worse than the mean predictor's first quantile performance are excluded. For the experiment setup regarded here, the base models Gauss, Exp, Spline, RandomForest, Esvm, and RFMlegp are preselected. Figure 4.11 shows the development of the weights distribution for this method. The search is restricted to the preselected models only, and the weights of these

models slowly and steadily settle down on the final solution.

4.4.6 Sequential Addition of Base Models

The previously introduced methods are realizable with small effort but also come with features that might as well be weaknesses. So is the restriction of the mutation as introduced entirely determined by the random choice of weights that is changed in each step. Whereas the preselection of models requires a selection rule to be specified and hence runs the risk to also exclude models that might have made a positive contribution to the ensemble otherwise.

This approach tries to combine the strengths of both ideas while doing without their weaknesses. The idea of this approach is to restrict the mutation to a small set of preselected models and then extending the set of models as the search proceeds. Before starting the optimization process, the base models are ranked by their performance, using the known fitness evaluation method (cf. Section 4.1). The selection of the models is based on this ranking. A small number of models is selected initially, and the remaining models are then added one by one following the ranking. Each time when adding another model to the search space, the search parameters of the (1+1)-ES are reset, and the search is started anew. In the following, the search steps between two consecutive resets are referred to as optimization tier.

Still, the question remains when the next model should be added to the system and how to proceed when the optimization process stagnates. For the experiments carried out in this section two adaptations of this idea are realized. Both start restricting the search to the three best base models and attribute a fixed number of minimum search evaluations to each tier. However, they follow different policies for the adaptation of the restriction of models and also for the handling of stagnation of the optimization process.

Sequential Addition with Stop on Stagnation
This adaptation of the approach adds the next best base model not before it succeeded in finding an improving distribution of the weights after adding the last model to the search space. With every addition of a model, the maximum number of search steps for the search is extended due to the number of active models. If no improving combination is found during the predefined minimum search interval, the search is continued until a combination is found or the search is finished.

Since the models are added to the system in order of their ranked performance, it might be more beneficial to extend the search on the actual tier than stepping on to the next. However, like the model selection approach, this may exclude lower ranked models, without ensuring if they would contribute to the system. Also, the problem of finding an improving distribution for the weights again may get harder with adding more models. But in opposition to the method of preselecting a set of base models for this optimization step no a priori decisions have to be made.
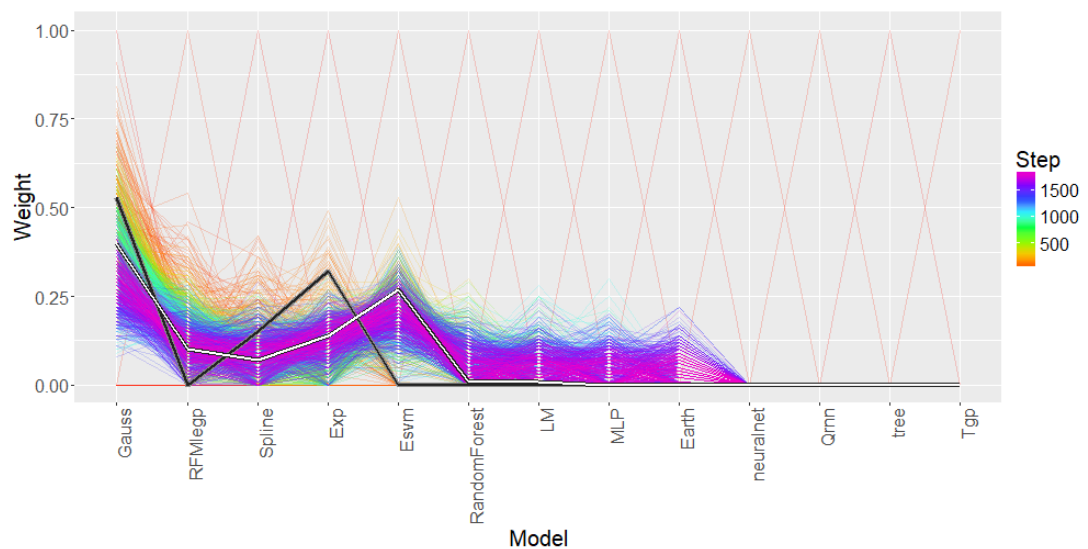
Figure 4.12: Developement of the ensemble weights during optimization while sequentially adding additional models. The black line marks the ensemble already found previously with only three base models available. The white line marks the best solution found during this optimization.

Figure 4.12 shows the development of the weights distribution for this method. The optimization process starts using only three base models: Gauss, Spline and RFMlegp. During the optimization process Exp, Esvm, RandomForest, LM, MLP, and Earth are consecutively added to the search space. After gaining no further improvement with the addition of Earth, the algorithm adds no further base models to the search space and eventually stops.

An interesting result is the performance of Esvm. The model was added to the search as the fifth model only, yet gained second most weight during optimization, while the three models added before gained less weight. Since the models are added to the system in order of their ranked performances in terms of RMSE,

75

this proves that a model ranked lower than another still may make a better contribution to the ensemble.

Sequential addition without Stop on Stagnation

Based on the idea, that models might be able to contribute to an ensemble though lower ranked than a model that doesn't, this method is adopted such, that every model of the set gets a chance to receive weight during the optimization.

In opposition to only extending the search space after finding an improving mutation, this adaptation of the algorithm drops the last model that was added after a fixed number of iterations, when the addition was not beneficial. Instead, it continues with the next best model. The algorithm stops when all models were at least for some time part of the search space, and after the last addition the search is stagnating or the maximal iteration count is reached.
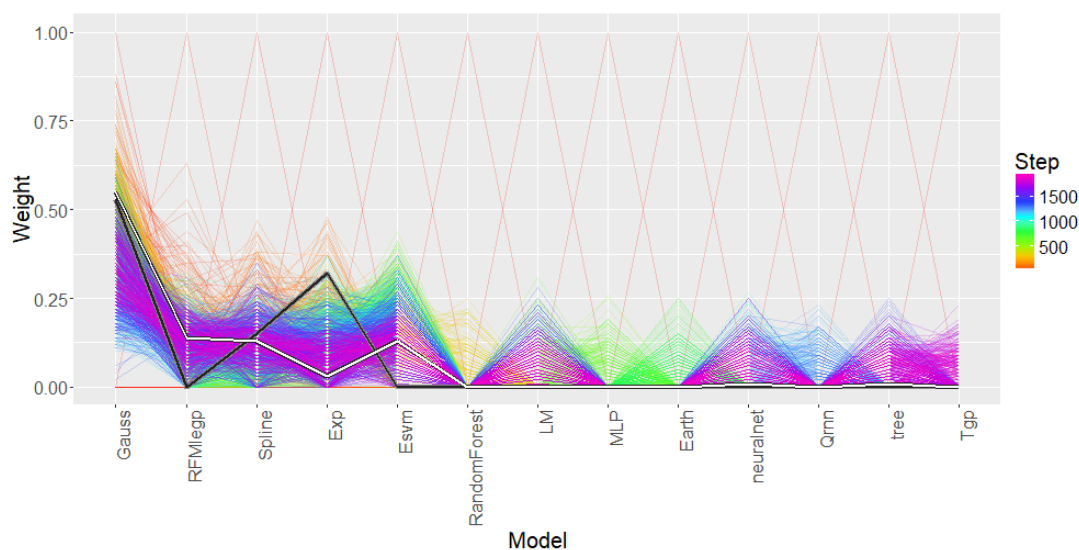
Figure 4.13: Development of the ensemble weights during optimization while sequentially adding additional models, and dropping non-beneficial models again. The black line marks the ensemble already found previously with only three base models available. The white line marks the best solution found during this optimization.

As before the algorithm starts only using Gauss, Spline, and RFMlegp. Then Exp and Esvm are added. RandomForest is added next but dropped again before adding LM. Next MLP and Earth are added but also dropped again. After that,

neuralnet is added and kept, then Qrnn but dropped again. At last tree and tgp are added and kept until the optimization finishes.

During the optimization, all base models were part of the search space for some time. Figure 4.13 shows the development of the weights during the optimization It can be seen, that some models were part of the system for some time during optimization and then dropped again. These models only have peaks in a small color range (i.e., RandomForest is peaked only by some yellow lines. Yellow is attributed to the earlier steps in the optimization).

The course of the optimization shows that indeed a model may make a beneficial contribution to the ensemble although a better-ranked model earlier was not able to do so.

4.4.7 Comparison of the Different ES-Adaptations

Results so far illustrate the behavior of the different approaches and show that all are able to overcome the initial inability to find an ensemble combination.
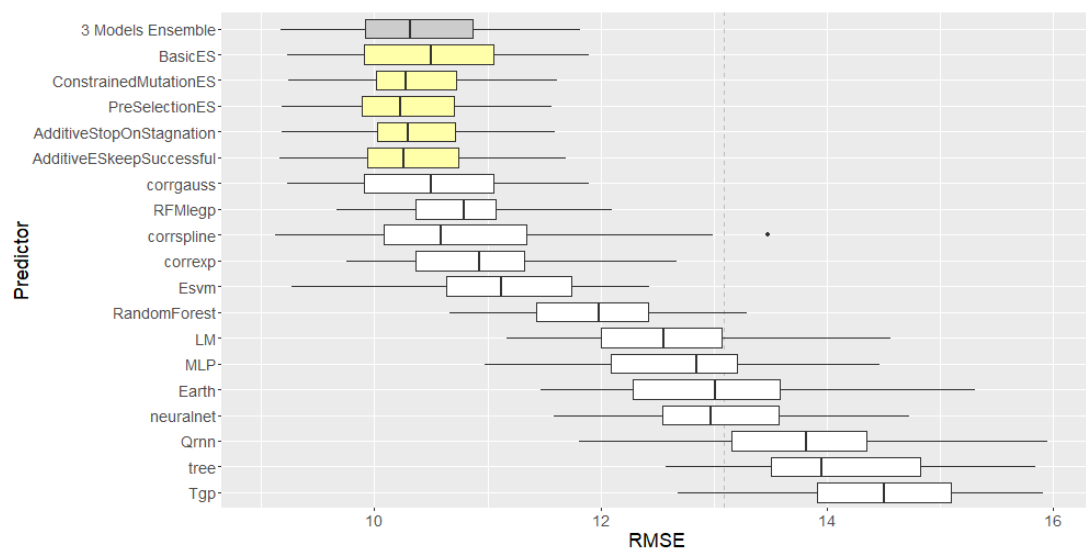
Figure 4.14: Comparison of the results found by the different optimizers with the base models and the ensemble built when only 3 base models were available.

Figure 4.14 shows a comparison of the performances of the base models (white), the ensemble model found with only three base models available (gray), and the ensembles found with the adapted search strategies using 13 heterogeneous models (yellow). A dashed line marks the mean performance of the baseline predictor.

4. BUILDING ENSEMBLES USING CONVEX LINEAR COMBINATIONS

The basic ES without any adaptations was not able to find an ensemble combination; therefore its result corresponds to the result of the Gauss model. It can be

Model	Ensemble weights													mean RMSE	SD
	Gauss	Exp	Spline	tree	Earth	LM	RandomForest	Tgp	Esvm	MLP	neuralnet	Qrnn	RFMlegp		
Gauss	1	0	0	-	-	-	-	-	-	-	-	-	-	**11.216**	0.87
Exp	0	1	0	-	-	-	-	-	-	-	-	-	-	11.621	0.71
Spline	0	0	1	-	-	-	-	-	-	-	-	-	-	11.489	1.06
exhaustive	0.5	0.3	0.2	-	-	-	-	-	-	-	-	-	-	10.449	0.62
optimization	0.53	0.32	0.15	-	-	-	-	-	-	-	-	-	-	**10.443**	0.62
constr. mut.	0.41	0.12	0.04	0	0.02	0	0	0.02	0.26	0.01	0.01	0.01	0.1	10.369	0.59
preselection	0.55	0.09	0.04	0	0	0	0	0	0.25	0	0	0	0.07	**10.334**	0.61
additive	0.4	0.14	0.07	0	0	0.01	0.01	0	0.27	0	0	0	0.1	10.355	0.60
keep succ.	0.55	0.03	0.13	0.01	0	0	0	0	0.13	0	0.01	0	0.14	10.376	0.61

Table 4.2: The table compares the results for the best base model and the different ensemble approaches. The left column names the considered models, in the first group the best performing base model only, in the second group the ensembles found by exhaustive search versus the ensemble found by optimization. The third group gives the results for the ensembles using the different adaptation strategies for the ES. The second column gives the weights on the base models. Since in the previous experiment only three base models were available, the others are crossed out. The last columns give the mean and standard deviation of the RMSE over all repetitions. The group ensembles using the adaptation strategies show the best results and the smallest standard deviations.

seen, that the ensembles found by the adapted search strategies can compete with the previous best solution (gray). All solutions found are performing comparably good.

The precise results for the ensemble solutions found using the adapted ES in comparison to the ensemble solution found with only three base models available (cf. Figure 4.8) and also to the best performing base model are displayed in Table 4.2.

From these results, it can be said that the different adaptations of the algorithms can handle the large search space, but it cannot be determined if one of these solutions is remarkably better than the others.

4.5 N-ary Ensembles on Higher Dimensional Physical Functions

So far a reliable method for ensemble building has been introduced and realized in different adaptations. On the experiment conducted all adaptations show comparable performance. To allow for a better comparison of the different adaptations of the method, in this section, the performance of the different adaptations are compared against each other and against the performance of the best performing base models alone.

Moreover, the initial step size σ_{init} of the (1+1)-ES is reconsidered. It is assumed, that with an increase of the dimension of the search space also the initial setting for the step width should be increased.

Therefore, the leading questions of this section are:

- Is one of the adaptations of the ensemble method preferable to the others?

- Moreover, should the initial step size σ_{init} adapted according to the dimensionality of the search space?

To answer these questions, additional experiments are carried out on a set of physical objective functions of higher dimensions (cf. Section 2.3.3).
For these experiments, the experimental setup has, in general, been left unchanged. Of course, for each experiment, the initial design size has been adjusted to 110, due to the higher dimensional functions. For the wing weight function, the initial design size has been set to 280, to ensure, that at least half of the base models perform better than the mean predictor.
Additionally, the experiments are used to get some insights into the effect of the initial step width σ_{init} on the optimization result. Therefore, each experiment is carried out twice with different values for σ_{init}. Here $\sigma_{init} = 0.16$, as used in earlier experiments is compared to $\sigma_{init} = 0.37$.

Figure 4.15 gives an overview of the performances of the different adaptations of the ensemble building method on the four physical functions. Each of the plots additionally shows the performance of the best performing base models. Further plots depicting the full comparison of all base models are also available in the Appendix (cf. Appendix A, Figures A.1-A.2).
From these plots, it can be read that each of the ensemble method adaptations can compete with the best performing base model. The Additive methods seem to have a small advantage over the remaining methods on the otl-circuit function and
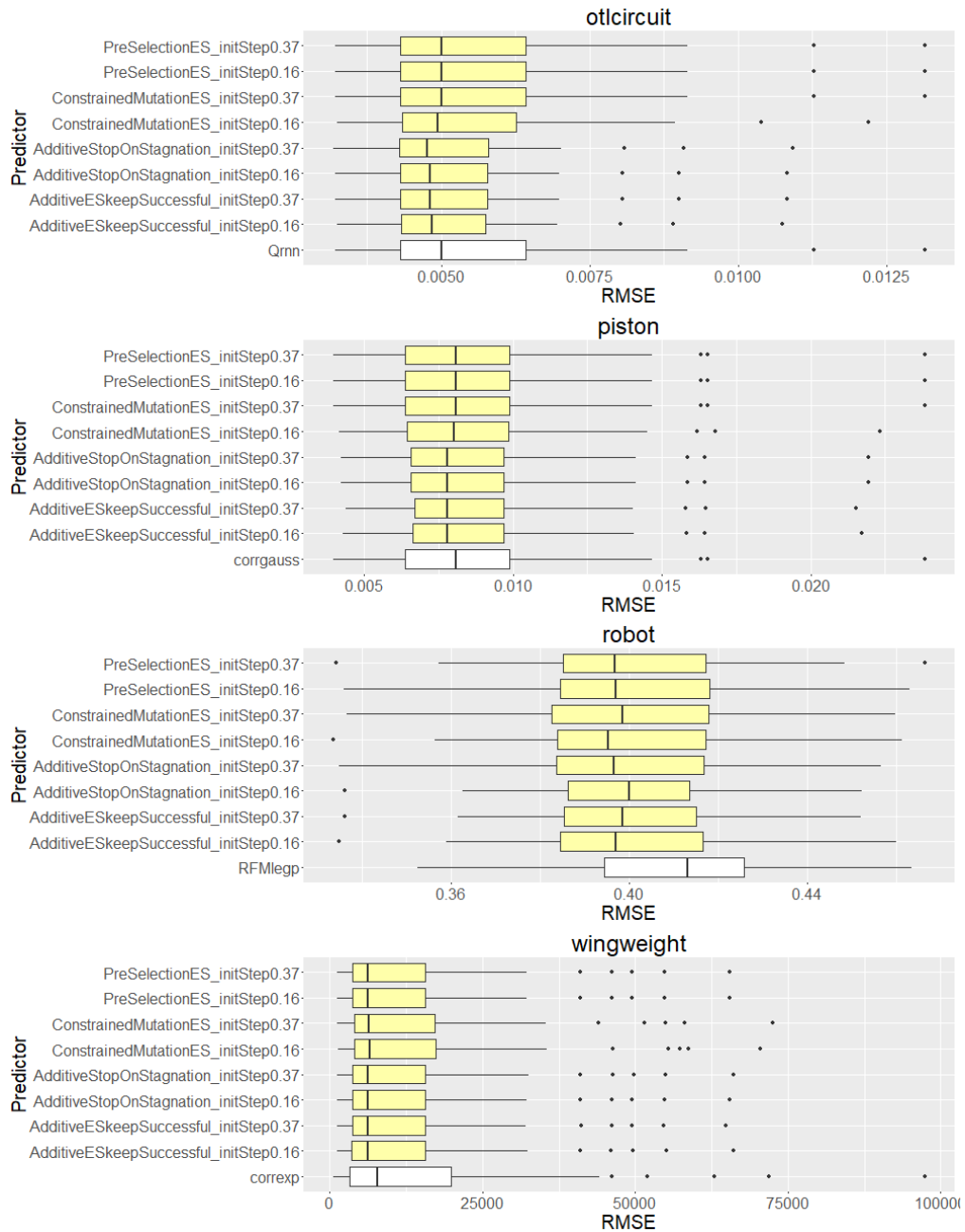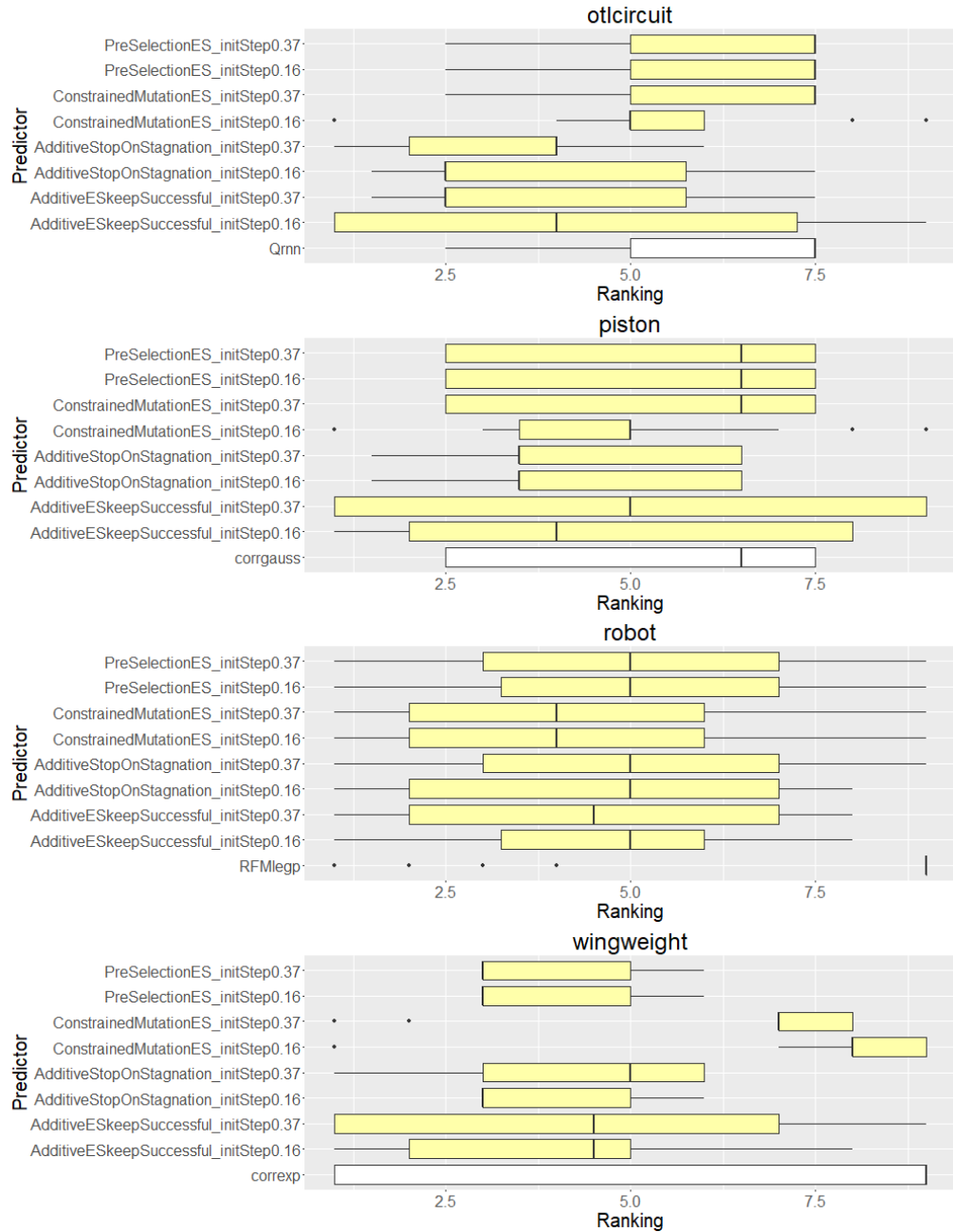
Figure 4.15: Comparison of the performances, in terms of RMSE, of the different adaptations of the ensemble building method and the best base model on the four physical functions. Ensemble results are colored yellow, the base model result is shown in white.

Figure 4.16: Comparison of the performances of the different adaptations of the ensemble building method and the best base model on the four physical functions. Results are repetition wise ranked. Ensemble results are colored yellow, the base model result is shown in white.

a smaller standard deviation on the piston function. On the robot arm function and the wing weight function the ensemble adaptations even show slightly better results, than the best base model.

To allow for a better comparison, Figure 4.16 shows the results of these experiments with the RMSEs achieved repetition wise ranked. From these results, it can be read, that the additive methods, in terms of ranking, on the otl-circuit function and the wing weight function perform better than the methods using pre-selection of the models or a constrained mutation.

FUN	AdditiveESkeepSuccessful		AdditiveStopOnStagnation	
	$\sigma_{init} = 0.16$	$\sigma_{init} = 0.37$	$\sigma_{init} = 0.16$	$\sigma_{init} = 0.37$
otl-circuit	0.0051659 ($\pm$ 0.00142)	**0.0051656** ($\pm$ 0.00144)	**0.0051656** ($\pm$ 0.00144)	0.0051669 ($\pm$ 0.00146)
piston	0.0086486 ($\pm$ 0.00323)	0.0086670 ($\pm$ 0.00319)	**0.0086355** ($\pm$ 0.00326)	**0.0086355** ($\pm$ 0.00326)
robot	0.4007355 ($\pm$ 0.02708)	0.4009263 ($\pm$ 0.02558)	0.4009612 ($\pm$ 0.02580)	0.4011576 ($\pm$ 0.02687)
wingweight	13492.530 ($\pm$ 15246.0)	**13457.750** ($\pm$ 15117.3)	13474.070 ($\pm$ 15180.1)	13505.770 ($\pm$ 15265.1)
Sum RankMN	13	**10.5**	11	17.5
Sum RankMD	14	19.5	16	**5.5**

FUN	ConstrainedMutationES		PreSelectionES	
	$\sigma_{init} = 0.16$	$\sigma_{init} = 0.37$	$\sigma_{init} = 0.16$	$\sigma_{init} = 0.37$
otl-circuit	0.0054833 ($\pm$ 0.00170)	0.0056067 ($\pm$ 0.00192)	0.0056067 ($\pm$ 0.00192)	0.0056100 ($\pm$ 0.00192)
piston	0.0086966 ($\pm$ 0.00338)	0.0087677 ($\pm$ 0.00354)	0.0087677 ($\pm$ 0.00354)	0.0087677 ($\pm$ 0.00354)
robot	0.4007446 ($\pm$ 0.02794)	**0.4003984** ($\pm$ 0.02682)	0.4015957 ($\pm$ 0.02743)	0.4012452 ($\pm$ 0.02815)
wingweight	14847.550 ($\pm$ 16888.8)	14398.630 ($\pm$ 16628.1)	13474.070 ($\pm$ 15180.1)	13474.070 ($\pm$ 15180.1)
Sum RankMN	21	22	25	24
Sum RankMD	19	27	22	21

Table 4.3: The Table gives the results of the different adaptations of the ensemble building method on the four physical functions and for two different settings of σ_{init}. Shown is the mean RMSE with standard deviation (in brackets). In the two last rows, the sums over the ranks of mean and median RMSE are given.

Table 4.3 lists all results in more detail. For each combination of ensemble building method adaptation with a given σ_{init} and a function, the mean RMSE with standard deviation (in brackets) is given. These mean RMSE values are function-wise ranked and summed up to obtain the sum of ranks over the mean RMSE values that are given in the second last row. Median RMSE values are considered accordingly. The sum of ranks over the median RMSE values is given in the last row.

For example, `AdditiveESkeepSuccessful` with an initial step width σ_{init} of 0.37 is ranked first on the Wing weight function (1) and shares the first place on the otl-circuit function (1.5), while on the Piston and the Robot Arm function it is ranked fourth place each (4; 4). In total it achieved a sum of ranks value over the

mean RMSE of 10.5, which is also the best value here.
Best evaluation in terms of sum of ranks over the median RMSE is achieved by
`AdditiveStopOnStagnation` with an initial step width σ_{init} of 0.37.

The results suggest that an additive approach with an initial step width σ_{init} of
0.37 might be the best choice for experiments of this dimension. However, for
more distinct results further experiments have to be carried out.

Additional plots for the development of the weights during optimization for the
different methods using an initial step width σ_{init} of 0.37 on the otl-circuit function
can be found in the Appendix (cf. Appendix A, Figures A.3 and A.4.

4.6 Conclusion

The primary goal of this chapter was to create an ensemble building strategy that
works reliably and as accurately as possible on arbitrary objective functions. A
method was aimed for, that can compete with the best performing base model for
each considered function. In the Sections 4.1 and 4.2, based on the Chapter 3, a
strategy was developed, which builds ensembles using convex linear combinations
of the models' predictions. The method was thoroughly analyzed, and convex
linear combinations showed to be an ideal choice for combining models.
The most important insights made and advantages recognized are:

- Due to the convex linear combination that is used for combination, in terms
  of RMSE, the ensemble cannot perform worse than the weakest base model.

- The ensemble can perform better than the base models when compensating
  opposing prediction errors.

- A CCM is favored over a base model only if the overall fit of the ensemble
  model is actually better (in terms of RMSE), than the overall fit of both
  base models.

- The nature of the combination is intuitive and interpretable.

- The linear convex combination of predictions for a given set of weights is
  easy to compute.

In Section 4.3, with the step from an exhaustive search to optimization using a
(1+1)-ES, the basis was provided for setting up a large system of heterogeneous
models. Then, in Section 4.4, further adaptations are made, and the set of base
models is extended to the intended size. First experiments showed that the

method was not able to handle the extended search space, which lead to further adaptations of the ensemble building method. Four different approaches were introduced and tested, and it was shown that the additional approaches are at an advantage in some cases. Another insight gained in these experiments is that indeed a model can make a beneficial contribution to the ensemble although a better-ranked model was not able to do so earlier.

In Section 4.5 additional experiments on physical functions were performed to allow for further comparison of the different approaches on functions of a higher dimension and with relation to real-world applications. Moreover, the question should be answered if, with an increasement of the search-space, also a larger value for the initial step width of σ_{init} would be recommendable. Still, the question which adaptation of the ensemble method is the best and if the initial step width σ_{init} should be adapted to the dimension of the search-space could not be answered satisfyingly and would require further analysis.

To a great extent, this chapter (Sections 4.1 through 4.3) is based on the article *"Building Ensembles of Surrogates by Optimal Convex Combination"* by Friese et al. [73]. Major parts from the original article were adopted verbatim. Of course, the text was adapted to fit the structure and notation of this thesis.