Honda Research Institute EU
**Honda Research Institute Europe GmbH**
https://www.honda-ri.de/

# Model Guided Evolution Strategies for Dynamically Balancing Exploration and Exploitation

**Edgar Reehuis, Johannes Kruisselbrink, Markus Olhofer, Lars Gräning, Bernhard Sendhoff, Thomas Bäck**

**2011**

**Preprint:**

This is an accepted article published in The Biannual International Conference on Articial Evolution. The final authenticated version is available online at: https://doi.org/[DOI not available]

# Model-guided Evolution Strategies for Dynamically Balancing Exploration and Exploitation

Edgar Reehuis[1,2], Johannes Kruisselbrink[1], Markus Olhofer[2], Lars Graening[2], Bernhard Sendhoff[2], and Thomas Bäck[1]

[1] Natural Computing Group, LIACS, Leiden University
Niels Bohrweg 1, 2333 CA Leiden, The Netherlands
{ereehuis,jkruisse,baeck}@liacs.nl
http://natcomp.liacs.nl
[2] Honda Research Institute Europe GmbH
Carl-Legien-Straße 30, 63073 Offenbach/Main, Germany
{markus.olhofer,lars.graening,bernhard.sendhoff}@honda-ri.de

**Abstract.** Wide exploration of high-dimensional, multimodal design spaces is required for uncovering alternative solutions in the conceptual phase of design optimization tasks. We present a general framework for balancing exploration and exploitation during the course of the optimization that induces sequential exploitation of different optima in the search space by selecting on a solution's fitness and a dynamic criterion termed interestingness. We use a fitness approximation model as a memory representing the parts of the search space that have been visited before. It guides the optimizer toward those areas that require additional sampling to be correctly modeled, and are hence termed interesting. Next to applying the prediction error of the model as a measure of interestingness, we consider the statistical variation in the predictions made by multiple parallel models as an alternative approach to quantify interestingness. On three artificial test functions we compare these setups running on a canonical ES to the same ES extended with either archive-based novelty, niching, or restarting, and to simply evaluating a Latin Hypercube set of sample points.

**Keywords:** Multimodal optimization, interestingness, novelty, niching, multiobjective selection, prediction error, variation in prediction

## 1 Introduction

Conventional design optimization of real-world problems aims for efficient adaptation of free system parameters to improve an existing system on given quality criteria. Following such a procedure often results in only a marginal improvement of a principally known solution. Methods for finding innovative and conceptually new solutions therefore are of increasing interest, allowing development of competitive products that are sufficiently distinct from rival products. Determining

alternative solutions is generally part of the conceptual design phase where normally a search is performed on a simplified problem with a reduced parameter space, thereby decreasing the level of solution detail to facilitate efficient traversal of the design space. The simplified solutions found serve as a starting point in a following optimization phase on the fully parametrized problem. Ideally the initial search would however be performed on the actual problem as well since potentially good products may get obscured through the parameter reduction.

Evolutionary Algorithms (EAs) are generally capable of exploration but rely largely on random modifications, making the probability of identifying optimal solutions relative to the problem dimensionality. Evolution Strategies (ESs), featuring sophisticated adaptations guiding the sampling of new solutions, quickly result in local searches around good solutions found in an initial exploration phase, on a multimodal problem usually converging to a single, randomly picked optimum. Relying solely on a quality function to select new solutions mainly causes this rapid shift toward exploitation of a single optimum.

An effective way to overcome this problem is *restarting* the optimization process from random initial positions in order to identify multiple optima [1]. This has the chance however of repeatedly zooming in on the same optima while others are never found, depending on the basins of attraction. In [3] an archive of *novel* solutions is kept between restarts that contains solutions exceeding a novelty threshold according to some difference measure [9]. The most novel solution found (i.e., viewed from within the optimization process, not necessarily from a designer's point of view) is used to position the next restart in the least explored region of the search space. *Niching* methods [17, 14] and derivates [18, 15] on the other hand run on top of the optimizer and induce parallel local searches for a predefined maximum number of optima, but as such are not very suitable for highly multimodal landscapes [14]. Another method is *Continuous Tabu Search* that is centered around lists of recently visited and promising solutions that (temporarily) get excluded from the search [10], but is therefore dependent on appropriate continuous neighborhood definitions. We propose an approach that sequentially jumps from optimum to optimum by alternating between states of exploration and exploitation. Next to the fitness function, an additional criterion that is subject to change during optimization is used to guide the random search.

Recent papers report on the use of such dynamic criteria based on a memory of earlier seen solutions parallel to the static fitness function in a *Pareto*-based *multiobjective* selection scheme. Mouret [13] uses an archive-based method, like in [3] of solutions that were highly novel upon discovery [9] and calculates the *novelty* of a solution as the average distance to the nearest solutions in this archive and the current population. Graening et al. [5] use a *fitness model* (i.e., surrogate model of the objective space) to represent the current knowledge of the search space and calculate a solution's *interestingness* as the maximization of the *prediction error* made by the fitness model as compared to the actual fitness value of a solution. As the optimization proceeds the model improves in

the area that is being sampled and the Pareto selection will gradually steer the optimizer into an area where the prediction error remains larger.

The current study continues on the model-based approach by introducing an alternative expression for interestingness, namely the *variation in prediction* between multiple fitness submodels trained on the same data. Furthermore, a straightforward approach for attaining global memory is tested. The behavior of the different methods is analyzed on three artificial test functions. We include archive-based novelty *multiobjectivization* [13], *fixed radius* and *self-adaptive radius* niching [17], and restarting with increased population size [1] as benchmarks. All methods run on top of a simple, single-stepsize, comma ES, and use the same evaluation budget. Furthermore, we compare to the basic approach of evaluating a *Latin Hypercube* set [11] of sample points.

In the following section the suggested framework for balancing exploration and exploitation comprising optimizer and interestingness measure is laid out, after which the results of the experiments involving the artificial test functions are presented in Section 3. In Section 4 we conclude with discussion and outlook.

## 2 Model-guided Framework for Dynamically Balancing Exploration and Exploitation

We propose a model-guided optimization framework for the search of optimal, conceptually different solutions in multimodel fitness functions. The usage of the model differs from other model-based and model-assisted strategies [8, 7] in the sense that the model does not replace the fitness function but is used to provide an additional criterion to guide the search. Based on previous work [5], the framework targets the efficient handling of the trade-off between explorative and exploitative search. The aim is to identify many conceptually different solutions by means of exploration, while providing an optimal configuration for each of the identified concepts, referred to as exploitation. In an alternating pattern the suggested algorithm zooms in on a single optimum until enough data has become available to properly model that part of the fitness landscape, after which it continues exploring for alternative optima.

Fig. 1 depicts the overall framework of the model-guided optimization process, building upon a population-based optimization strategy such as an EA. The main difference to traditional optimization is that the search process is additionally guided by the interestingness of a solution, a dynamic additional criterion that gets taken into account next to the static fitness, in a multiobjective selection scheme. Applying interestingness comes with two major challenges, the construction of an adequate memory of already generated solutions and the definition of the indicator that quantifies interestingness. A universal approximation model predicting solutions' fitness is adopted to form an abstract representation of the solution space. A solution should be qualified as interesting if the solution vector is different from the already generated solutions, or if it resides in a rugged, hard-to-model area of the fitness landscape. Both requirements are

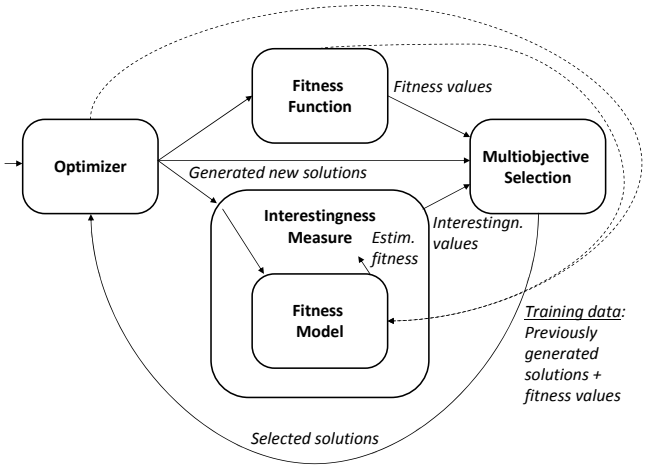



**Fig. 1. Model-guided Framework for Dynamically Balancing Exploration and Exploitation.**

inherently accounted for in using an interestingness indicator that is based on the prediction error of a fitness approximation model.

### 2.1 Framework Instantiation

The current study employs a $(\mu, 7\mu)$-ES with self-adaptation based on a single stepsize per individual ($\mu : \lambda$ should at least be $1 : 7$ [16]) as optimizer. The model of the fitness landscape is built using feed-forward neural networks with linear outputs and a single hidden layer of 10 sigmoidally activated nodes. The weights of the networks are trained with the *improved Rprop* backpropagation algorithm by Igel et al. [6] for a maximum of 1000 epochs. The networks are fully connected, including direct connections from input to output nodes, and contain a bias node. All connection weights are initialized in the interval $[-0.1, 0.1]$. The fitness model should represent the current knowledge of the function landscape best, hence there is no risk of overfitting and therefore no validation set is used (i.e., all available solution data is used for the actual training).

Two model-based interestingness indicators are investigated (defined further on), the prediction error (PE) of the fitness model directly and the variation in prediction (ViP) of multiple approximation submodels, the latter implicitly depending on the prediction errors of the different submodels. The question remains whether the aimed for dynamics are best induced by either a memory involving all generated solutions or only of the most recent ones. Therefore two types of models are considered: a *local* fitness model that per iteration is re-initialized and trained on the solution data from the *previous* $\gamma$ iterations, and a *global* memory consisting of all *distinct* previous local models, that is, every $\gamma$-th model. We term the latter a "horizontal ensemble" of fitness models referring

to the distinct training data used per model. In evaluating the interestingness the distinct models as well as the new local model are taken into account, for each evaluated solution adopting the model with the best prediction and lowest variation in case of PE and ViP respectively.

Multiobjective selection based on Pareto dominance is used, as proposed in [5] and [13], implemented using non-dominated sorting (and crowding distance sorting) as featured in NSGA-II [4]. Note that we apply this sorting procedure in combination with $(\mu, \lambda)$-selection (i.e., the new parent population is selected from the offspring only), as the aim is not to obtain and refine a Pareto front of optimal solutions but to steer the optimizer away from "sufficiently" exploited optima. An initial comparison between plus (i.e., taking parents into account as well) and comma selection showed slower dynamics and the chance of relapsing to standard ES convergence behavior using the first, identifying only one optimum. A disadvantage of using Pareto-based multiobjective selection is that the required degree of exploitation cannot be indicated. To what extent a certain optimum is exploited before the optimizer shifts the focus to a different region of the search space therefore depends on the shape of the fitness landscape and the capabilities of the applied modeling technique, together with the characteristics of the used interestingness measure.

**Prediction Error.** The interestingness of a real-valued solution vector $\mathbf{x} = (x_1, \ldots, x_n)$ with $n$ indicating the problem's dimensionality is calculated as the absolute difference between the model's fitness estimation $\tilde{f}(\mathbf{x})$ and the actual fitness value $f(\mathbf{x})$ [5],

$$\mathrm{PE}(\mathbf{x}) = \left| \tilde{f}(\mathbf{x}) - f(\mathbf{x}) \right|. \tag{1}$$

**Variation in Prediction.** The interestingness is taken as the variation between the predictions of multiple submodels that are trained on exactly the same data (as the training data is equal, this could be termed a "vertical ensemble" of submodels). In the current study multiple neural networks of the same architecture and type are trained. If sufficient data is available to correctly model a certain solution the fitness estimations of the different submodels should largely align. Otherwise, the predictions will be more diverse. Let $\tilde{f}_i(\mathbf{x})$ be the fitness estimation by submodel $i$ and $\mathrm{num}_{\mathrm{subm}}$ be the number of different submodels, then the variation in prediction is calculated using the *interquartile range* (IQR, the difference between the third and first quartiles of the data) of the fitness estimations,

$$\mathrm{ViP}(\mathbf{x}) = \mathrm{IQR}(\tilde{f}_1(\mathbf{x}), \ldots, \tilde{f}_{\mathrm{num}_{\mathrm{subm}}}(\mathbf{x})). \tag{2}$$

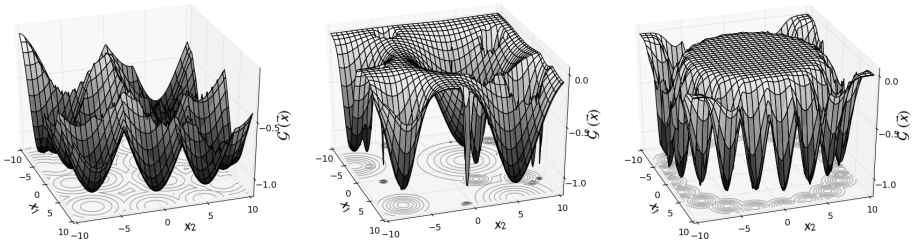
The IQR is a robust statistic in the sense that it is not prone to outliers. If most submodels "agree" but one is considerably different, this can have a great impact when for instance the sample variance is used as measure of variation.

## 3 Experiments

We run experiments on three artificial test functions, visualized in Fig. 2, that are variants of the *Gaussians* test function [5], which we define as

$$\mathcal{G}(\mathbf{x}) = -\max_{i\in\{1,\ldots,20\}} \exp(-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)), \tag{3}$$

with $\mathbf{x} \in [-10,10]^n$, $\boldsymbol{\mu}_i \sim \mathcal{U}([-10,10]^n)$, $\boldsymbol{\Sigma}_i = \mathrm{diag}((\sigma_1^2,\ldots,\sigma_{20}^2))$, and $\sigma_i = 2\sqrt{n}$. All variants feature 20 global optima regardless of the dimensionality, and problem instances with dimension 2, 5, and 10 are considered. *GaussiansEqual* uses the definition listed above, involving 20 kernels with equal standard deviations positioned randomly in the fitness landscape. *GaussiansVary* uses the default standard deviation for the first kernel and decreases it for the remaining $i \in \{2,\ldots,20\}$ via $\sigma_i = \sigma_{i-1} \cdot 0.8$. *GaussiansPlane* places 20 kernels with equal standard deviations $\sigma_i = 1$ on a two-dimensional plane in a circle with radius 10; in case $n > 2$ the remaining $n-2$ object variables in the kernel means $\boldsymbol{\mu}_i$ are set equal to the same random vector $\mathcal{U}([-10,10]^{n-2})$. This setup reflects the real-world case in which certain parameters (here the first two) can take various values while the remaining ones should be set precisely.




(a) GaussiansEqual (b) GaussiansVary (c) GaussiansPlane

**Fig. 2. Test Functions in 2D.**

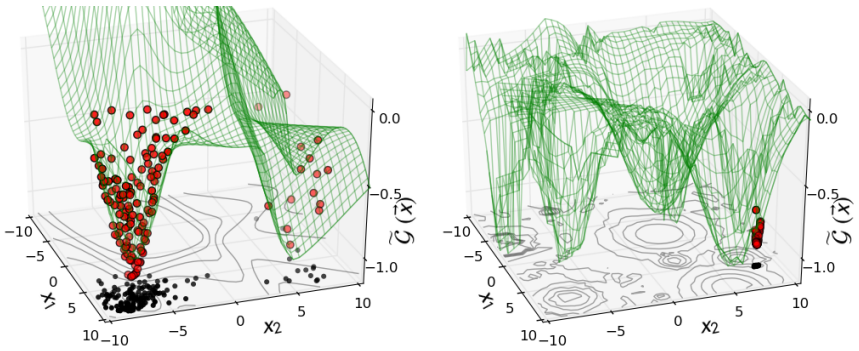
For performance assessment of the different methods we use the *Maximum Peak Ratio* (MPR) measure that reflects the quality and the quantity of the approximated optima in the set of all generated solutions. MPR was originally proposed in the context of niching and maximization problems with strictly positive fitness range [12], therefore we define a slightly adjusted version:

1. Normalize the Gaussians variants to maximization problems, strictly positive, using $f_{\mathrm{norm}}(x) = \frac{f(x)-\max}{\min-\max}$ with $\min = -1.0 - 1\% \cdot \mathrm{range}$, $\max = 0.0 + 1\% \cdot \mathrm{range}$, and $\mathrm{range} = 1.0$;
2. Assign each generated solution to the closest optimum with respect to Euclidean distance;
3. Per optimum, from the solutions assigned to it, use the fitness of the best solution; if it is less than 80% [12] of the actual fitness of the optimum, use 0;

4. Divide the sum of the selected fitness value per optimum by the sum of the actual fitness values of the optima.

The maximum MPR score is therefore 1.0, indicating that all optima were closely sampled.

The model-based methods PE and ViP are either guided by $\gamma$-*local* models as in [5] or the proposed horizontal *ensemble* of $\gamma$-local models. ViP uses 10 submodels per $\gamma$-local model. We take $\gamma = 5$ and at the heart of the optimization is a $(5, 35)$-ES applying single-stepsize self-adaptive mutation with learning rate $\tau = \frac{1}{\sqrt{2n}}$ [2]. The initial parent individuals are sampled uniformly within the initialization interval, and their initial stepsize $\sigma_{\text{init}}$ is set to $\frac{1}{4}$ of the initialization interval [17]. Each offspring is the product of 2 parents, using discrete recombination for the object variables $x_i$ and intermediary recombination for the stepsize [2].




(a) Local (b) Ensemble

**Fig. 3. Modeling Setups Compared.** Plots of the final fitness models obtained in a single run of PE on GaussiansVary 2D, using the 5-local fitness model setup in (a) and the ensemble of 5-local fitness models in (b). The dots represent the solution vectors used to train the last generated 5-local fitness model, in both cases.

As first benchmark we include archive-based novelty multiobjectivization [13], a similar Pareto setup that uses an archive-based novelty criterion to steer the optimizer into sparsely sampled regions, aiming for a wide sampling of the search space that becomes finer grained as the optimization continues. It runs on the same ES as the model-based methods. The novelty of a solution is calculated as the average distance to the nearest 15 solutions with respect to Euclidean distance, selected from an archive of novel solutions and the current population. A solution is added to the archive if its novelty exceeds the novelty threshold, which value is self-adapted during the optimization: if 1% of the fitness evaluations was used and no new solutions were added, the threshold is decreased by 5%, and if more than 4 solutions were added in one generation, the threshold is increased by 5%. Presumably the initial threshold value is to be set relative

to the size of the initialization domain but this relation is not clear. After some tuning it is set to 1.0, noting that due to its self-adaptation and the fact that the novelty score depends strongly on the current population, the exact initial value does not seem highly critical.

Two niching methods are included as benchmarks, fixed radius and self-adaptive niche radius [17], running on top of an $(20, 140)$-ES, except for population size equal to the $(5, 35)$-ES used for the Pareto setups[1]. Niching works by parallelly converging to a pre-indicated number of $q$ optima in the ES population guided by *niche radii*, with a single *niche* being used to zoom in on each optimum. In fixed radius niching the radius is calculated such that the entire search space is evenly divided over the $q$ desired niches, while in self-adaptive radius niching the length of each radius is coupled to changes of the stepsize and it decreases as is zoomed in on an optimum [17]. Although there is interplay between the different niches (e.g., in case of plus selection the individuals can move between niches), recombination is not used; offspring from a certain niche is generated as mutated copies of the niche best individual. We use $q = 20$ and all remaining settings are as prescribed in [17].

The last benchmark is the *IPOP-ES*[2], a scheme in which an underlying $(\mu, 7\mu)$-ES is restarted upon convergence by resampling the population and re-intializing stepsizes to $\sigma_{\text{init}}$, while increasing the population size between restarts as is aimed for more global search as the optimization proceeds [1]. The process is started from a $(2, 14)$-ES, besides the population size equal to the ESs mentioned above, and $\mu$ is increased by a factor of 2 at each restart. The ES is assumed to be converged if the average stepsize of the parent population falls below $10^{-3}$ times the initialization interval, similar to the convergence criterion used in [3].

For reference we include a comparison with standard ES behavior and the result of covering the search space with a Latin Hypercube (LH) set of sample points [11]. We use LH sampling as this offers multiple sets with mostly similar spread properties. For enforcing the population of the ES to remain within the initialization interval, we employ constraint handling that sets the interestingness or novelty score of an individual to a penalty value upon boundary violation; for the standard ES, niching, and IPOP-ES the fitness is deteriorated by instead adding a penalty value. For all tested methods we use a budget of 3500 function evaluations, and all methods have been implemented using the Shark Machine Learning Library v2.3.3[3].

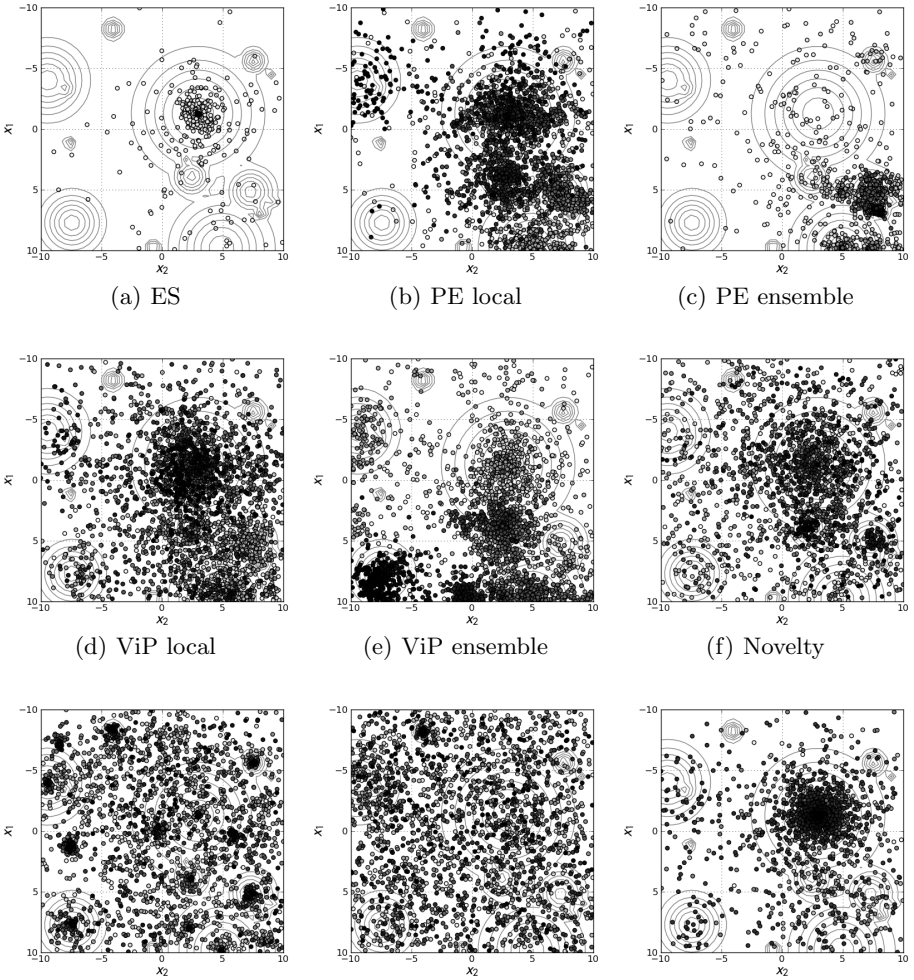
### 3.1 Results

In examining the outcome of the experiments, we start by visually analyzing the sampling behavior for GaussiansVary 2D based on a single run per method, plot-

[1] Running niching on a canonical ES allows for a fair comparison, but note that self-adaptive niching was proposed in combination with the cumulative adaptation mechanism of the stepsize in the CMA-ES [17].

[2] The IPOP-ES, derived from the IPOP-CMA-ES [1], is run on a canonical ES to allow for comparison.

[3] http://shark-project.sourceforge.net



(a) ES (b) PE local (c) PE ensemble

(d) ViP local (e) ViP ensemble (f) Novelty

(g) Niching fixed (h) Niching self-adaptive (i) IPOP-ES

**Fig. 4. Sampling Behavior on GaussiansVary 2D.** Plots of all solutions generated in the search space for a single run per method (dark: recent sol., bright: early sol.).

ted in Fig. 4. The gray tones of the sample points reveal the different dynamics: The standard ES converges quickly to the optimum with the largest attractor basin, while the model-based methods visit optima in a sequential pattern. Novelty multiobjectivization keeps sampling widely across the search space with extra focus on optimal regions, while the niching methods manage to zoom in on most of the optima in parallel. The IPOP-ES shows behavior similar to that of the standard ES, although the search space is sampled more widely because of the restart phases and iteratively increased population size. Of the model-based methods the local variants can be seen revisiting the largest attractor, while ViP

ensemble shows the clearest temporal separation between the visited optima. PE ensemble however gets stuck as the ensemble model is able to approximate most of the fitness landscape, nullifying the effect of the interestingness indicator as the area directly around an optimum must have higher interestingness than the optimum itself to be able to move away from it.

Fig. 5 plots the MPR scores over 30 runs per method. First of all the model-based approaches do not manage to outperform niching, with self-adaptive radius niching showing best performance across the test problems. Of the model-based variants PE ensemble clearly has the worst scores, while ViP local performs best. It benefits from the higher variation in its local model as compared to the ensemble of models. Novelty multiobjectivization has largely comparable results to those of ViP local, while the IPOP-ES shows slightly less performance. Furthermore, uninformed sampling across the entire search space works in low dimensions but as would be expected, evaluating an LH set in 10D is not useful.

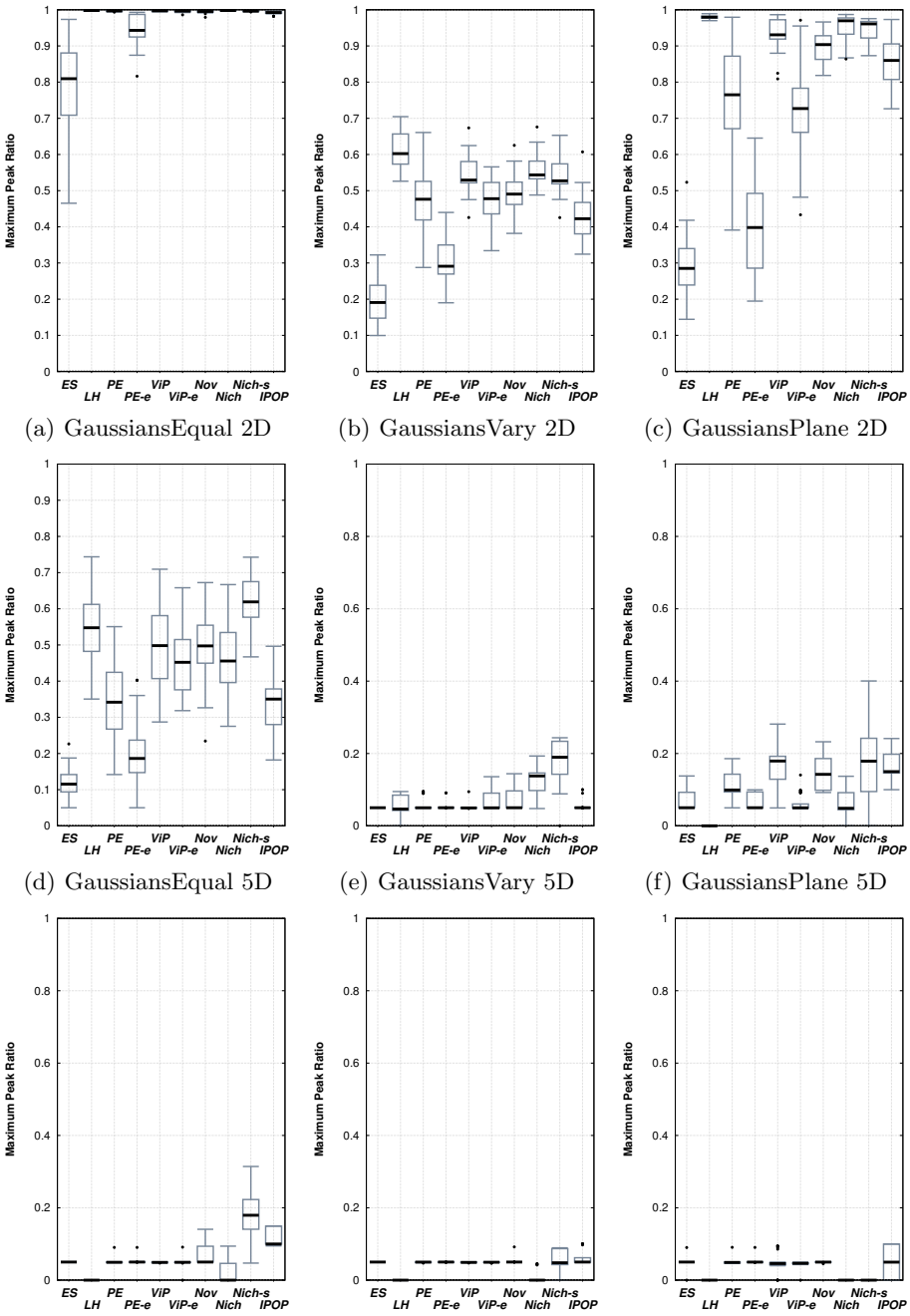
## 4 Discussion

This paper presented a model-guided framework that alternates between states of exploration and exploitation to find multiple optima in a multimodal landscape. The tested instantiation comprised a canonical ES with isotropic mutation, feedforward neural networks, and Pareto-based multiobjective selection on fitness and an interestingness indicator, applying either the model prediction error (PE) or the variation in prediction (ViP) between submodels. A comparison was made to archive-based novelty multiobjectivization, niching methods, restarting with increased population, and evaluating a Latin Hypercube (LH) set of sample points.

ViP local performed best of the proposed model-based approaches with respect to the Maximum Peak Ratio; the novelty scheme showed comparable performance to it while the restart setup performed slightly less. Self-adaptive niching however showed the best overall performance. On 10-dimensional problem instances all tested methods performed weakly, while still better than LH.

While elegant in the sense of omitting weight parameters on the fitness and interestingness values, the applied Pareto-based selection in the framework induces a dispersing effect, slowing down the ES in zooming in on optima. Conversely, it hinders the ES in escaping from sufficiently exploited optima, together with the small stepsize resulting after an exploitation phase. An alternative would be dynamically combining fitness and interestingness into a single objective, although this presumably suffers from the same problem while do requiring weighting.

Cuccu et al. [3] dismiss the dynamic aggregation in favor of restarting the ES after convergence, using an archive of visited points to position it in the least explored region of the search space. Instead, a global model could be trained between restarts on *all* sampled points and searched for the least *understood* region, e.g., applying ViP. Moreover, using restarts omits the need to perform a clustering step to isolate optima, required when applying the framework to black-box problems.




(a) GaussiansEqual 2D (b) GaussiansVary 2D (c) GaussiansPlane 2D

(d) GaussiansEqual 5D (e) GaussiansVary 5D (f) GaussiansPlane 5D

(g) GaussiansEqual 10D (h) GaussiansVary 10D (i) GaussiansPlane 10D

**Fig. 5. Results Overview.** The results of 30 runs per method on each test function instance are plotted, all runs involving 3500 function evaluations. The Maximum Peak Ratio indicates the quality and the quantity of the approximated optima in the set of all generated solutions per run and is to be maximized.

## References


1. Auger, A., Hansen, N.: A Restart CMA Evolution Strategy with Increasing Population Size. In: 2005 IEEE Congress on Evolutionary Computation, CEC 2005. pp. 1769–1776. IEEE Computer Society (2005)
2. Bäck, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press, Oxford, UK (1996)
3. Cuccu, G., Gomez, F., Glasmachers, T.: Novelty-Based Restarts for Evolution Strategies. In: 2011 IEEE Congress on Evolutionary Computation, CEC 2011. pp. 158–163. IEEE Computer Society (2011)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Trans. Ev. Comp. 6(2), 182–197 (2002)
5. Graening, L., Aulig, N., Olhofer, M.: Towards Directed Open-ended Search by a Novelty Guided Evolution Strategy. In: Schaefer, R., et al. (ed.) PPSN XI, LNCS, vol. 6239. pp. 71–80. Springer (2010)
6. Igel, C., Hüsken, M.: Empirical evaluation of the improved Rprop learning algorithms. Neur. 50, 105–123 (2003)
7. Jin, Y.: A comprehensive survey of fitness approximation in evolutionary computation. Soft Comp. 9(1), 3–12 (2005)
8. Jones, D., Schonlau, M., Welch, W.: Efficient Global Optimization of Expensive Black-Box Functions. J. Glob. Opt. 13(4), 455–492 (1998)
9. Lehman, J., Stanley, K.: Exploiting Open-Endedness to Solve Problems Through the Search for Novelty. In: Artificial Life XI. pp. 329–336. MIT Press (2008)
10. Mashinchi, M., Orgun, M., Pedrycz, W.: Hybrid optimization with improved tabu search. Appl. Soft Comp. 11(2), 1993–2006 (2011)
11. McKay, M., Beckman, R., Conover, W.: A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. Techn. 21(2), 239–245 (1979)
12. Miller, B., Shaw, M.: Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimization. In: 1996 IEEE Congress on Evolutionary Computation, CEC 1996. pp. 786–791. IEEE Computer Society (1996)
13. Mouret, J.B.: Novelty-Based Multiobjectivization. In: Doncieux, S., et al. (ed.) 2009 EvoDeRob Workshop, SCI, vol. 341. pp. 139–154. Springer (2011)
14. Preuss, M.: Niching the CMA-ES via Nearest-Better Clustering. In: Branke, J., et al. (ed.) 12th Annual Conference on Genetic and Evolutionary Computation, GECCO'10. pp. 1711–1718. ACM Press (2010)
15. Saha, A., Deb, K.: A Bi-criterion Approach to Multimodal Optimization: Self-adaptive Approach. In: Deb, K., et al. (ed.) SEAL 2010, LNCS, vol. 6457. pp. 95–104. Springer (2010)
16. Schwefel, H.P.: Evolution and Optimum Seeking: The Sixth Generation. John Wiley & Sons, Inc., New York, NY, USA (1993)
17. Shir, O., Emmerich, M., Bäck, T.: Adaptive Niche Radii and Niche Shapes Approaches for Niching with the CMA-ES. Ev. Comp. 18(1), 97–126 (2010)
18. Zechman, E., Ranjithan, S., Liu, L.: Niched Co-Evolution Strategies to Address Non-uniqueness in Engineering Design. In: 8th Annual Conference on Genetic and Evolutionary Computation, GECCO'06. Late breaking paper. ACM Press (2006)