Towards Self-Adjusting Weighted Expected Improvement for Bayesian Optimization

Carolin Benjamins, Elena Raponi, Anja Jankovic, Carola Doerr, Marius Lindauer

Towards Self-Adjusting Weighted Expected Improvement for Bayesian Optimization

Carolin Benjamins
Leibniz University Hannover
Hannover, Germany

Elena Raponi
Sorbonne Université, CNRS, LIP6
Paris, France

Anja Jankovic
Sorbonne Université, CNRS, LIP6
Paris, France

Carola Doerr
Sorbonne Université, CNRS, LIP6
Paris, France

Marius Lindauer
Leibniz University Hannover
Hannover, Germany

ABSTRACT

In optimization, we often encounter expensive black-box problems with unknown problem structures. Bayesian Optimization (BO) is a popular, surrogate-assisted and thus sample-efficient approach for this setting. The BO pipeline itself is highly configurable with many different design choices regarding the initial design, surrogate model and acquisition function (AF). Unfortunately, our understanding of how to select suitable components for a problem at hand is very limited. In this work, we focus on the choice of the AF, whose main purpose it is to balance the trade-off between exploring regions with high uncertainty and those with high promise for good solutions. We propose Self-Adjusting Weighted Expected Improvement (SAWEI), where we let the exploration-exploitation trade-off self-adjust in a data-driven manner based on a convergence criterion for BO. On the BBOB functions of the COCO benchmark, our method performs favorably compared to handcrafted baselines and serves as a robust default choice for any problem structure. With SAWEI, we are a step closer to on-the-fly, data-driven and robust BO designs that automatically adjust their sampling behavior to the problem at hand.

CCS CONCEPTS

• **Computing methodologies → Randomized search**.

KEYWORDS

Bayesian Optimization, Self-Adjusting Weighted Expected Improvement, Acquisition Function Schedules, Upper Bound Regret

1 INTRODUCTION

Black-box problems f are challenging because we do not know the underlying structure of the landscape. While we can sequentially query different points x and learn from these how to choose the next promising points, there is no direct information on the direction of making progress or how to trade off exploration and exploitation. This is especially challenging when we have a low number of available function evaluations in relation to the size of the search space $\mathcal{X}$. Formally, we want to find the minimum x^* of our function f:

$$x^* \in \arg\min_{x \in \mathcal{X}} f(x) \tag{1}$$

We focus in our paper on Bayesian optimization (BO) [9, 19], as a well-studied and sample-efficient approach for expensive black-box optimization. The main idea of BO is to use a probabilistic surrogate model (e.g., a Gaussian Process), iteratively refining an approximation of the problem landscape that guides the optimization process. BO starts with an *initial design* or *design of experiment* (DoE), obtained from sampling strategies, e.g., random sampling, Sobol sequence or Latin Hypercube desing [4, 15]. With these initial points, the surrogate model is built to capture the uncertainty of the true cost on unobserved points. The *acquisition function* (AF) (a.k.a. infill criterion) is a utility function to trade-off exploration of underexplored areas and exploitation of presumable promising areas. The point with the highest acquisition function value is queried next, and the surrogate model is adjusted with the new observation. These steps are repeated for a given overall optimization budget.

Besides accurate probabilistic surrogate models and other bells and whistles [3, 5, 16], the exploration-exploitation trade-off is crucial for successful and efficient optimization. Since the landscape of the black-box optimization problem is unknown, it is a-priori unknown which AF should be chosen for the optimization problem at hand. Even worse, since each problem has its unique landscape, we need different exploration-exploitation trade-offs [1, 2].

Because there are different choices of AFs, e.g., Probability of Improvement (PI) [14], Expected Improvement (EI) [20], Upper Confidence Bound (UCB) [8] and Thompson Sampling (TS) [23], selecting a suitable one for the problem at hand remains challenging. Furthermore, in the past, the choice of an AF has been considered *static* over the BO process. Prior works suggest that mixed AF-strategies [12, 13] or even very simple schedules switching from EI to PI can improve anytime performance of BO; however, for each problem different schedules, incl. static ones, perform best [2].

Selecting an AF-schedule with a meta-learned selector based on the exploratory landscape analysis (ELA) features [18] of the initial design factors in the problem at hand and further improves performance [1]. Nevertheless, this approach has its limitations. First, it requires a large initial design compared to the overall budget, and the ideal size of it is unknown. Second, the selector is trained for a specific budget, and it is unclear how it transfers to other dimensions and budgets.

In this work, we instead aim for an *self-adjusting yet simple* approach to adapt the exploration-exploitation trade-off in a data-driven way throughout the optimization process. For this, we propose to adaptively set the weight α of Weighted Expected Improvement (WEI) [22] in an online control fashion. The crucial questions to answer here are (i) *When* should we adjust α? And (ii) *how* should we adjust α?

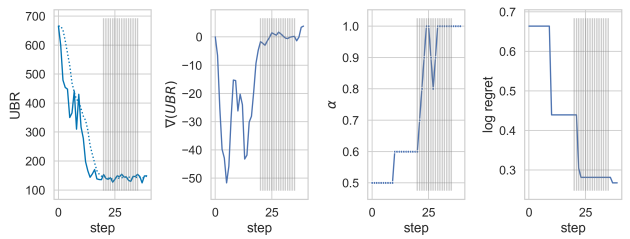

Figure 1: With SAWEI we self-adjust the exploration-exploitation trade-off parameter α based on the Upper Bound Regret (UBR) (left). Whenever the gradient of UBR (2nd left) becomes 0 (marked by vertical lines), we adjust α (2nd right), further reducing the log regret (right).

We propose a new method, dubbed Self-Adjusting Weighted Expected Improvement (SAWEI). Inspired by a termination criterion for BO [17], we adjust the weight α whenever BO tends to converge, indicated by the Upper Bound Regret (UBR) and adjust α opposite to the dominant search attitude of exploration vs exploitation. We demonstrate the effectiveness of our method SAWEI on the BBOB functions of the COCO benchmark [11] against established AF and handcrafted baseline AF-schedules for α.

2 SELF-ADJUSTING WEIGHTED EI

In our method SAWEI, we adaptively set the weight α of WEI to steer the exploration-exploitation trade-off. WEI is defined as [22]:

$$WEI(\mathbf{x};\alpha) = \alpha \underbrace{z(\mathbf{x})s(\mathbf{x})\Phi[z(\mathbf{x})]}_{\text{exploitation}} + (1-\alpha) \underbrace{s(\mathbf{x})\phi[z(\mathbf{x})]}_{\text{exploration}} \tag{2}$$

with $z(\mathbf{x}) = (f_{\min} - \hat{y}(\mathbf{x}))/s(\mathbf{x})$, $f_{\min}$ being the lowest observed function value, $\hat{y}(\mathbf{x})$ and $s(\mathbf{x})$ the predicted mean and standard deviation from the surrogate model, and ϕ and Φ being the PDF and CDF of a Gaussian distribution, respectively. α weights the exploration and exploitation terms. For example, $\alpha = 0.5$ recovers standard EI [20] and $\alpha = 1$ has a similar behavior as PI [14].

When To Adjust. In order to be able to set α adaptively we need an indicator of the progress of the optimization. Recently, Makarova et al. [17] proposed a termination criterion to stop BO for hyperparameter optimization. If the Upper Bound Regret (UBR) falls under a certain threshold, they terminate. UBR estimates the true regret at iteration k by:

$$\bar{r}_k := \min_{\mathbf{x}\in G_k} \mathrm{UCB}_k(\mathbf{x}) - \min_{\mathbf{x}\in \mathcal{X}} \mathrm{LCB}_k(\mathbf{x}) \tag{3}$$

with G_k being the history of all evaluated points and $\mathcal{X}$ being the entire search space. The first term estimates the worst-case function value of the best-observed point, a.k.a. the incumbent, and the second term the lowest function value across the whole search space. This means the smaller the gap between both terms becomes, the closer we are at asymptotic function value *under the current settings of the optimizer*. Instead of using UBR to stop the optimization process, it serves as an indicator for us when to switch or adjust components. Our rule is: When the gradient of UBR over the last n steps becomes close to 0, we adjust the exploration-exploitation attitude with α. Please find the details in Appendix A.

How to Adjust. The remaining question is how to adjust α, by how much and into which direction. We propose a rather simple, yet effective additive change by Δ_α. We set $\Delta_\alpha = 0.1$ to allow for gradual changes. We determine the sign of Δ_α by the recent optimization *attitude*: Depending on whether the EI- or PI-term of Equation (2) is larger for the last selected point, the current search attitude was either exploring or exploiting, respectively. We inspect the attitude and adjust α in the *opposite* direction, i.e. to provide a chance for more exploration or exploitation in contrast to the currently dominating attitude.

3 EXPERIMENTS

3.1 Baselines

We compare our data-driven, self-adjusting method SAWEI to (i) the well-established best practice of simply using a single AF and (ii) hand-designed schedules of α , see Table 1. We start with static schedules of $\alpha \in \{0, 0.5, 1.\}$, either fully exploring, EI, or a modulated PI*.[1] We can also adjust α whenever there is an incumbent change and either increase (WEI Turn Up), decrease α (WEI Turn Down) or define the direction via the attitude (WEI Turn Auto). Further, we define a schedule from EI to modulated PI (and vice versa) as a step function with 5 steps. In addition, we compare to hard switches from EI to PI [1] as well as the Gutmann-Sobester pulse cycling through α [10, 22].

Table 1: Baselines. PI* denotes WEI($\alpha = 1$).

Explore	$\alpha = 0.0$
EI	$\alpha = 0.5$
PI*	$\alpha = 1.0$
WEI Turn Auto	$\Delta_\alpha \sim$ attitude
WEI Turn Down	$\alpha_0 = 1., \Delta_\alpha = -0.1$
WEI Turn Up	$\alpha_0 = 0.5, \Delta_\alpha = +0.1$
EI → PI* (Linear)	5 steps
PI* → EI (Linear)	5 steps
EI → PI	switch after 25 %
EI → PI	switch after 50 %
EI → PI	switch after 75 %
Gutmann-Sobester Pulse	Cycle $\alpha \in [0.1, 0.3, 0.5, 0.7, 0.9]$

3.2 Setup

We evaluated SAWEI and the baselines on the BBOB functions of COCO benchmark [11] in a fixed budget setting. We set the dimensionality of the synthetic functions to 2 and the budget of the initial design to 10 and of the surrogate-based optimization to 40 function evaluations. Our implementations were built upon the BO tool SMAC3 (v2.0.0b1) [15]. We use a standard GP as configured in SMAC's BlackBoxFacade and SMAC optimizes the acquisition function with a combination of local and random search which also applies to minimizing LCB in Equation (3) for calculating the UBR. We

[1] Following Eq. (2), setting $\alpha = 1$ results in a modulated PI and not the original PI.

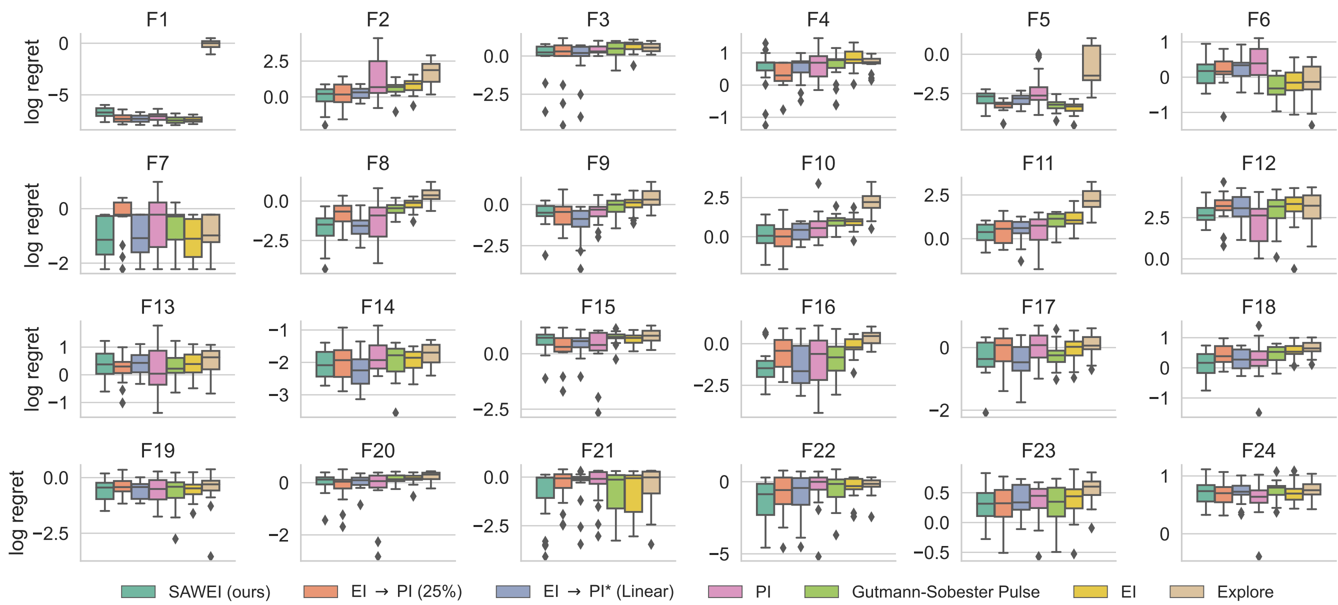

Figure 2: Box-Plots of all 24 BBOB functions of COCO. Each shows the final log regret after 20 repeated runs with different random seeds.

set $\beta_t = 2\log(dt^2/\beta)$, $\beta = 1$ for UCB/LCB as in SMAC. You can find the code here: https://github.com/automl/SAWEI/tree/GECCO23. We set our convergence check horizon to $n = 1$, i.e. check whether the last gradient is close to 1. Our evaluation protocol repeats the optimization 20 times with different random seeds and calculates the interquartile mean (IQM) across seeds to robustly estimate the regret per function. For each schedule, we then determine the rank for each of the 24 BBOB functions and compute the global rank across functions. In the plots over optimization steps, we show the mean and standard error across all the functions.

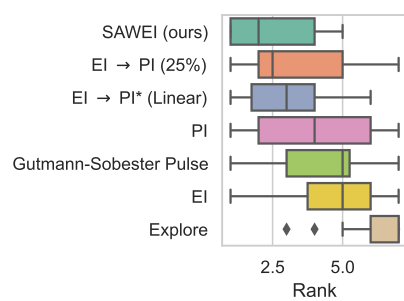

Figure 3: Ranks across BBOB Functions

3.3 Results

Overall Results. Our method SAWEI ranks first across the BBOB functions, followed by a switch from EI to PI and by the step function from EI to PI*, see Figure 3. This supports the intuition of starting with exploration and later leaning towards exploitation. Also, the popular default choice EI ranks second to last.

When we take a look at the ranks across all variants (Table 2 in the Supplementary), SAWEI is quite robust to the exact time of adjustment and for how long the attitude is tracked (see Appendix B for details). Interestingly, the modulated PI, being fairly exploitative, is pretty competitive, although the initial design was quite small. Adjusting α after an incumbent change is not beneficial (WEI Turn X schedules).

For the log regret (see Figure 2) we can observe that not one schedule is optimal for each function and we should aim to exploit the complementarity of the different schedules in future work. But what we can also observe is that our SAWEI is a robust default choice.

Evolution of α. The general schedule set by SAWEI is increasing, i.e. changing the attitude from exploring to exploiting. The slope of α is smaller or bigger depending on the problem. On a closer inspection, one interesting thing to note here is that sometimes SAWEI resembles a hard switch from EI to PI and sometimes it is almost a linear increase across the optimization budget, resembling the step function, see Figure 4 (right). On this particular schedule, WEI Turn Down (ranking less desirably overall so not a robust default choice, see Supplementary Table 2), performs quite well. This α schedule is very opposite to the schedule traversed by SAWEI, suggesting that per function schedules might be very dissimilar yet lead to similar performance. SAWEI also works on highly multi-modal functions with weak global structure (F20-F24).

4 LIMITATIONS AND FUTURE WORK

Although SAWEI performs best on average across a diverse set of functions, our first results have several limitations. First of all,

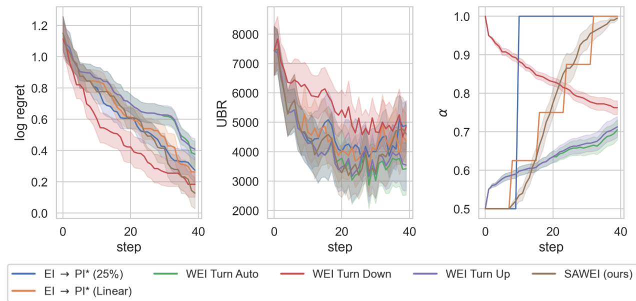

Figure 4: Exemplary insights on BBOB Function 18. SAWEI increases α from exploring to exploiting.

we only considered 2-dimensional, artificial functions so far. As a next step, a study of SAWEI on higher-dimensional functions and black-box problems from practical applications, e.g., HPO [7] is required. Also, for specific α WEI does not select configurations lying on the Pareto front of exploration and exploitation as EI and UCB do [6]. In addition, as a first step *how to adjust* is a heuristic offering potential for a data-driven approach. Furthermore, SAWEI is still oblivious to the general landscape structure and focuses on rather local properties. Future work could consider a meta-learned approach with ELA features [1] or guided by reinforcement learning [21].

5 CONCLUSION

In this paper, we addressed the problem of robustifiying Bayesian Optimization on expensive black-box functions. The main observation is that (i) black-box functions follow different problem structures and (ii) more exploitation is often required in later optimization stages. Thus the optimization process requires self-adjustment. Instead of choosing among many different complementary acquisition functions, we proposed in this work to self-adjust weighted expected improvement in a data-driven manner, which allows for different exploration-exploitation tradeoffs in different stages. In particular, we suggested *when* and *how* to adjust: (i) Whenever the Upper Bound Regret (UBR) [17] does not change anymore and (ii) adjust in the opposite direction of the current search attitude. On the BBOB functions of the COCO benchmark [11] our method ranks favorably compared to handcrafted baselines and is a robust default choice for *any* problem structure.

ACKNOWLEDGMENTS

Carolin Benjamins and Marius Lindauer acknowledge funding by the German Research Foundation (DFG) under LI 2801/4-1. Carola Doerr and Elena Raponi acknowledge funding by the ANR T-ERC project *VARIATION* (ANR-22-ERCS-0003-01), and by the PRIME programme of the German Academic Exchange Service (DAAD) with funds from the German Federal Ministry of Education and Research (BMBF), and by CNRS, project number 20-51-15009.

REFERENCES

[1] C. Benjamins, A. Jankovic, E. Raponi, K. van der Blom, M. Lindauer, and C. Doerr. 2022. Towards Automated Design of Bayesian Optimization via Exploratory Landscape Analysis. In *Meta-Learning Workshop @ NeurIPS'22*.

[2] C. Benjamins, E. Raponi, A. Jankovic, K. van der Blom, M. Laura Santoni, M.s Lindauer, and C. Doerr. 2022. PI is back! Switching Acquisition Functions in Bayesian Optimization. In *Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems Workshop @ NeurIPS'22*.

[3] J. Bossek, C. Doerr, and P. Kerschke. 2020. Initial Design Strategies and Their Effects on Sequential Model-Based Optimization: An Exploratory Case Study Based on BBOB. In *Proc. of GECCO'20*. 778–786.

[4] E. Brochu, V. Cora, and N. de Freitas. 2010. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. *arXiv:1012.2599v1 [cs.LG]* (2010).

[5] A. Cowen-Rivers, W. Lyu, R. Tutunov, Z. Wang, A. Grosnit, R. Griffiths, A. Maraval, H. Jianye, J. Wang, J. Peters, and H. Ammar. 2021. An Empirical Study of Assumptions in Bayesian Optimisation. *arXiv:2012.03826 [cs.LG]* (2021).

[6] G. De Ath, R. Everson, A. Rahat, and J. Fieldsend. 2021. Greed is Good: Exploration and Exploitation Trade-offs in Bayesian Optimisation. *ACM Trans. Evol. Learn. Optim.* 1 (2021), 1–22.

[7] K. Eggensperger, P. Müller, N. Mallik, M. Feurer, R. Sass, A. Klein, N. Awad, M. Lindauer, and F. Hutter. 2021. HPOBench: A Collection of Reproducible Multi-Fidelity Benchmark Problems for HPO. In *Proc. of NeurIPS'21 Datasets and Benchmarks Track*.

[8] Alexander I. J. Forrester, András Sóbester, and Andy J. Keane. 2008. *Engineering Design via Surrogate Modelling - A Practical Guide.* John Wiley & Sons Ltd.

[9] R. Garnett. 2023. *Bayesian Optimization.* Cambridge University Press.

[10] H. Gutmann. 2001. A radial basis function method for global optimization. *J. Global Optimiz.* 19, 3 (2001), 201–227.

[11] N. Hansen, A. Auger, R. Ros, O. Mersman, T. Tušar, and D. Brockhoff. 2020. COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting. *Optimization Methods and Software* (2020).

[12] M. Hoffman, E. Brochu, and N. de Freitas. 2011. Portfolio Allocation for Bayesian Optimization. In *Proc. of UAI'11*. 327–336.

[13] K. Kandasamy, K. Vysyaraju, W. Neiswanger, B. Paria, C. Collins, J. Schneider, B. Poczos, and E. Xing. 2020. Tuning Hyperparameters without Grad Students: Scalable and Robust Bayesian Optimisation with Dragonfly. *Journal of Machine Learning Research* 21 (2020), 81:1–81:27.

[14] H. Kushner. 1964. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Fluids Engineering* (1964), 97–106.

[15] M. Lindauer, K. Eggensperger, M. Feurer, A. Biedenkapp, D. Deng, C. Benjamins, T. Ruhkopf, R. Sass, and F. Hutter. 2022. SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization. *JMLR* 23, 54 (2022), 1–9.

[16] M. Lindauer, M. Feurer, K. Eggensperger, A. Biedenkapp, and F. Hutter. 2019. Towards Assessing the Impact of Bayesian Optimization's Own Hyperparameters. In *IJCAI'19 DSO Workshop*.

[17] A. Makarova, H. Shen, V. Perrone, A. Klein, J. Faddoul, A. Krause, M. Seeger, and C. Archambeau. 2022. Automatic Termination for Hyperparameter Optimization. In *International Conference on Automated Machine Learning, AutoML 2022, 25-27 July 2022, Johns Hopkins University, Baltimore, MD, USA (Proceedings of Machine Learning Research, Vol. 188)*, Isabelle Guyon, Marius Lindauer, Mihaela van der Schaar, Frank Hutter, and Roman Garnett (Eds.). PMLR, 7/1–21.

[18] O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph. 2011. Exploratory landscape analysis. In *Proc. of GECCO'11*. 829–836.

[19] J. Mockus. 1989. *Bayesian approach to global optimization. Theory and Applications.* Kluwer Academic Publishers.

[20] J. Mockus, V. Tiesis, and A. Zilinskas. 1978. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization* 2, 117-129 (1978).

[21] G. Shala, A. Biedenkapp, N. Awad, S. Adriaensen, M. Lindauer, and F. Hutter. 2020. Learning Step-Size Adaptation in CMA-ES. In *Proc. of PPSN'20*. 691–706.

[22] A. Sobester, S. Leary, and A. Keane. 2005. On the Design of Optimization Strategies Based on Global Response Surface Approximation Models. *J. Glob. Optim.* 33, 1 (2005), 31–59.

[23] W. Thompson. 1933. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika* 25, 3/4 (1933), 285–294.

A HOW TO DETECT THE ADJUSTMENT POINT

First, we apply a moving interquartile mean (25 %-75 % quartiles) with window size 7 to UBR to smooth the curve because we are interested in a general signal. Then, we calculate the absolute gradients. We signal time to adjust when the last absolute gradient is close to 0 with an absolute tolerance of ϵ times the last observed maximum of the absolute gradient.

B RANKS OF ALL VARIANTS

In Table 2 we list the ranks of all variants with our selected method marked in bold. PI* denotes the modulated PI. The sensitivity to the gradient controlling when to adjust is denoted by ϵ (see Appendix A). We can either base our attitude only on the last EI and PI terms ("last") or based on the attitudes until the last incumbent change ("until inc change"). We see that SAWEI is fairly robust to ϵ and until when the attitude is tracked (n).

Table 2: Ranks of All Schedules. PI* denotes the modulated PI.

Schedule	Rank
SAWEI $\epsilon = 0.1$ (last)	7.583
SAWEI $\epsilon = 0.5$ (last)	8.583
SAWEI $\epsilon = 0.25$ (last)	9.083
SAWEI $\epsilon = 0.1$ (until inc change)	9.167
EI → PI* (Linear)	9.333
SAWEI $\epsilon = 0.05$ (until inc change)	9.333
SAWEI $\epsilon = 0.5$ (until inc change)	9.667
EI → PI* (25%)	9.750
SAWEI $\epsilon = 0.25$ (until inc change)	9.750
EI → PI (25%)	9.833
EI → PI* (50%)	11.667
PI*	11.917
SAWEI $\epsilon = 0.05$ (last)	12.000
WEI Turn Up	12.250
WEI Turn Down	13.167
EI → PI* (75%)	13.333
PI	13.833
EI → PI (75%)	14.083
WEI Turn Auto	14.917
EI → PI (50%)	15.600
Gutmann-Sobester Pulse	18.167
EI	19.000
PI* → EI (Linear)	19.250
Explore	22.500

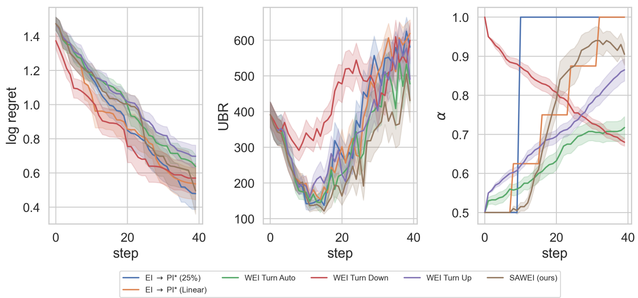

Figure 8: BBOB Function 4

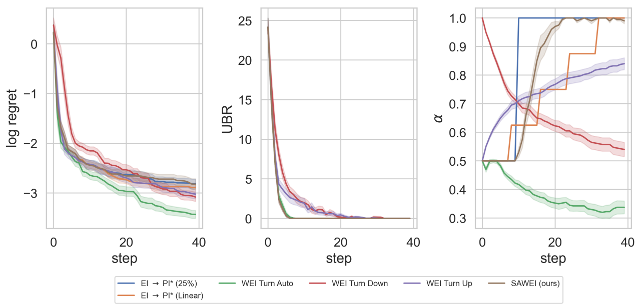

Figure 9: BBOB Function 5

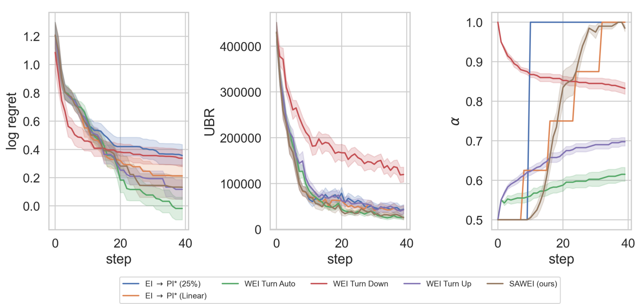

Figure 10: BBOB Function 6

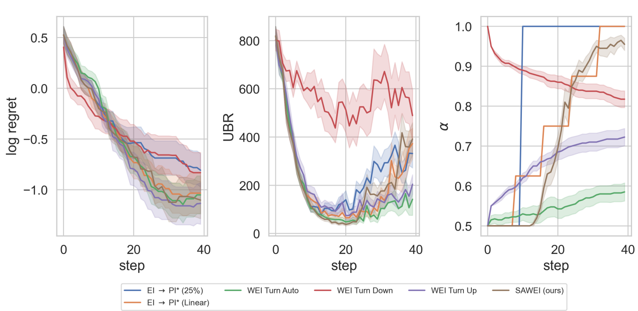

Figure 11: BBOB Function 7

C PERFORMANCE OVER TIME FOR ALL BBOB FUNCTIONS

Here we plot the log regret over time with UBR and α of selected schedules. If we have a closer look at highly multi-modal functions with weak global structure we see that often the UBR *increases*. This might be due that the approximated landscapes changes a lot after each new observed point.

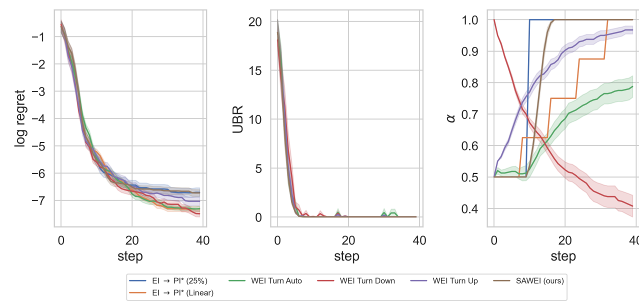

Figure 5: BBOB Function 1

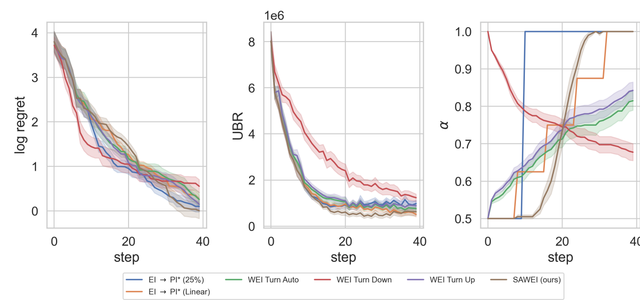

Figure 6: BBOB Function 2

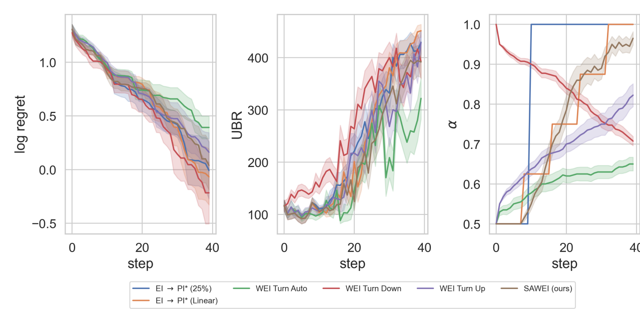

Figure 7: BBOB Function 3

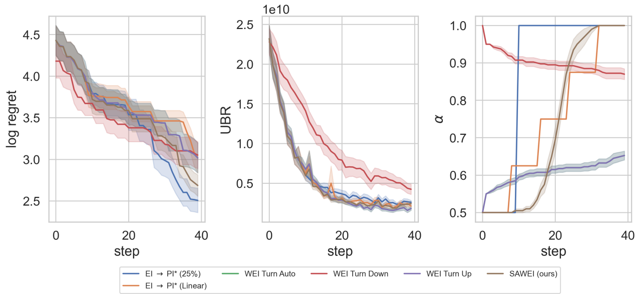

Figure 16: BBOB Function 12

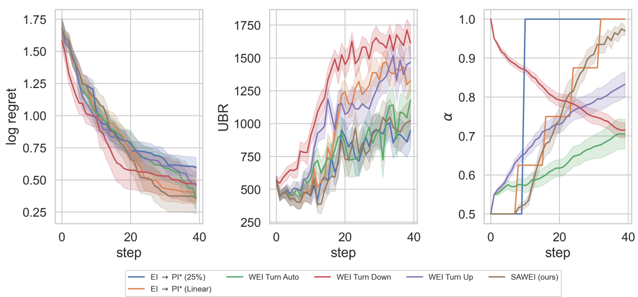

Figure 17: BBOB Function 13

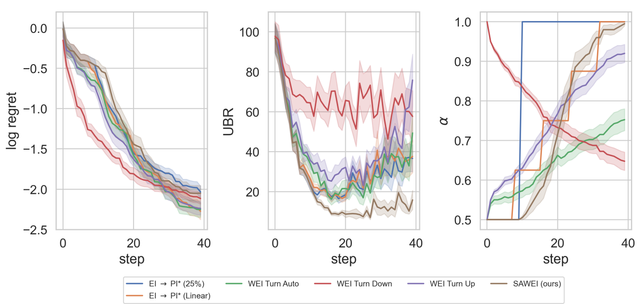

Figure 18: BBOB Function 14

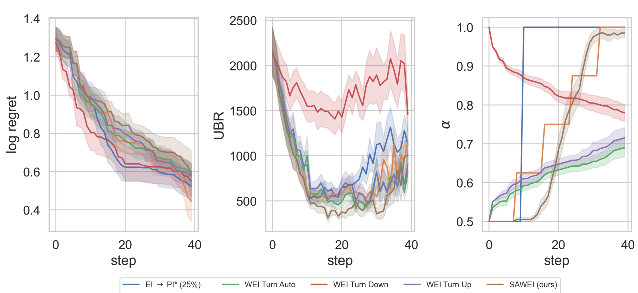

Figure 19: BBOB Function 15

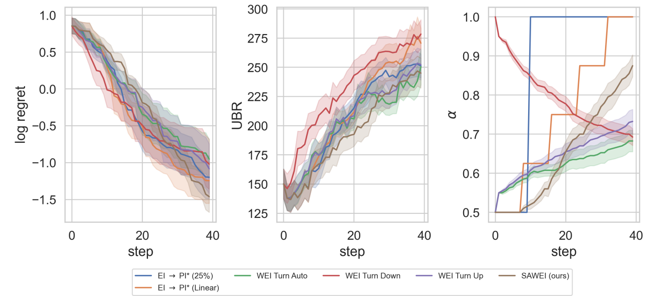

Figure 20: BBOB Function 16

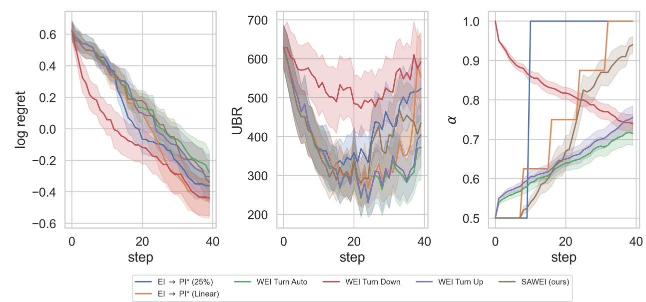

Figure 21: BBOB Function 17

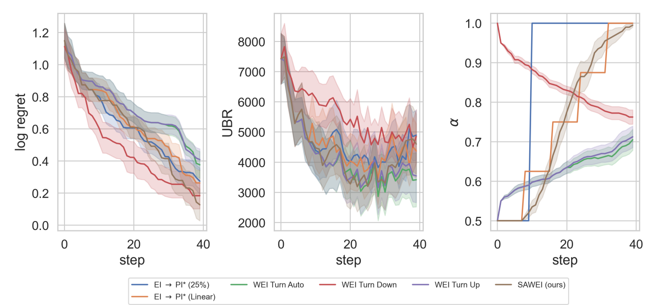

Figure 22: BBOB Function 18

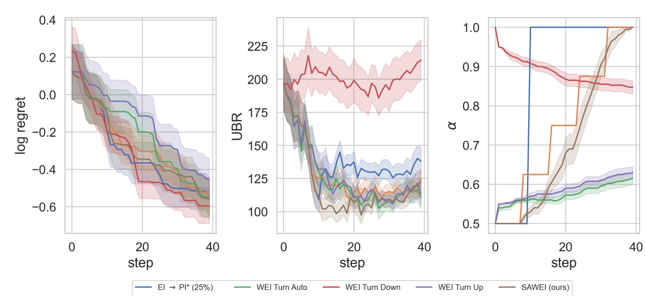

Figure 23: BBOB Function 19

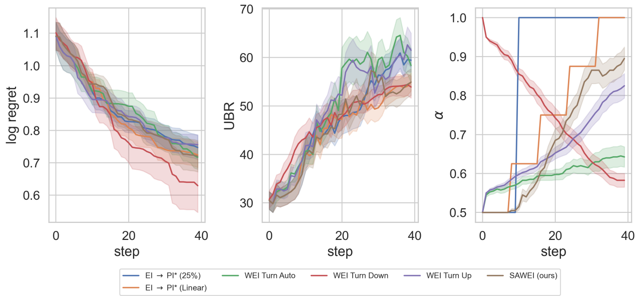

Figure 28: BBOB Function 24