A Visualizations of the Optimization Landscape of $2D$ HPO problems

In Figure 1, we visualize the optimization landscape of the $2D$ HPO problems. We use the same LHS initial design that has been used for the calculation of ELA features and train a Gaussian process with a Matérn 5/2 kernel to be able to interpolate between points and plot the predictions of the Gaussian process. Colors are based on performance quantiles with a dark green representing the 0.1% quantile of best points with respect to the logloss, yellow representing the 7% quantile and white representing the 100% quantile. The red point depicts the empirical minimum.

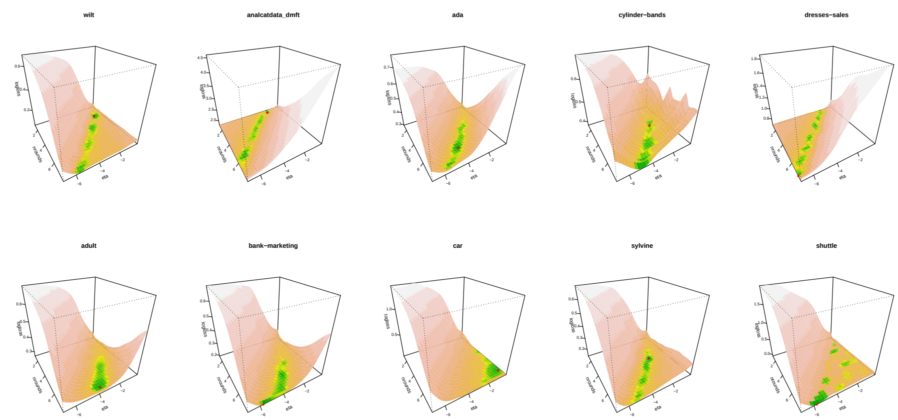

Fig. 1: Optimization landscapes of $2D$ HPO problems.

B Optimizer Details

All optimizers are implemented within the `mlr3` ecosystem [4], either in `bbotk` [1], or `mlr3mbo` [7]. `CMAES` relies on `adagio::pureCMAES()` from the `adagio` package [2]. `GENSA` relies on `GenSA::GenSA()` from the `GenSA` package [8]. `Grid` and `Random` are directly provided by `bbotk`. `MBO` is used as implemented in the `coco` branch of `mlr3mbo` (following the configuration presented in [5]) and uses a Gaussian process (Matérn 5/2 kernel) as surrogate model and expected improvement as acquisition function which is optimized by a global local search (a procedure starting with a random search followed by L-BFGS-B [3] with the current best point as the starting point, which is repeated multiple times). We use an initial design of size $4D$ constructed by sampling uniformly at random.

C Nearest BBOB Neighbors

In Table 1, we list the nearest BBOB neighbors of our HPO problems obtained by minimizing the Euclidean distance over the first two principal component scores in ELA feature space.

Table 1: Nearest BBOB neighbors to HPO problems in ELA feature space (two principle components).

HPO	Nearest BBOB	HPO	Nearest BBOB
wilt_2	2_1_2	adult_2	4_4_2
wilt_3	7_3_3	adult_3	5_5_3
wilt_5	3_3_5	adult_5	3_3_5
analcatdata_dmft_2	11_4_2	bank-marketing_2	15_1_2
analcatdata_dmft_3	7_5_5	bank-marketing_3	13_4_3
analcatdata_dmft_5	2_1_5	bank-marketing_5	5_3_5
ada_2	17_4_3	car_2	14_2_2
ada_3	17_5_5	car_3	13_5_3
ada_5	10_5_5	car_5	17_1_5
cylinder-bands_2	7_1_2	sylvine_2	2_1_2
cylinder-bands_3	10_4_3	sylvine_3	7_2_3
cylinder-bands_5	14_1_5	sylvine_5	7_3_5
dresses-sales_2	19_4_3	shuttle_2	6_5_2
dresses-sales_3	7_5_5	shuttle_3	8_2_3
dresses-sales_5	2_4_3	shuttle_5	2_3_5

D Configuring `GENSA` for HPO

We observed that `GENSA` shows lacklustre performance on HPO problems compared to BBOB problems. `GENSA` relies on `GenSA::GenSA()` from the `GenSA` package [8] and allows for setting several hyperparameters affecting its generalized simulated annealing routine, e.g., `temperature`, `visiting.param` and `acceptance.param`. However, [8] state that the optimizer showed robust performance with the current default values. Nevertheless, two additional parameters, `smooth` (default is `TRUE`) and `simple.function` (default is `FALSE`) are available which appear to be relevant in our context. `smooth` switches the local search algorithm within the generalized simulated annealing routine from L-BFGS-B [3] to Nelder-Mead [6] (which may work better if the objective function has only few places where numerical derivatives can be computed). `simple.function` impacts the number of local search steps performed when the best energy value is not updated after several iterations within the generalized simulated annealing routine. In our experiments, `GENSA` was employed with its default parameters on both the BBOB and HPO problems. To investigate the effect of the parameters `smooth` and `simple.function`, we ran a small ablation study, comparing four `GENSA` variants with respect to these parameters in a full factorial design (`GENSAv1`: `smooth = TRUE` and `simple.function = TRUE`; `GENSAv2`: `smooth = FALSE` and `simple.function` = `TRUE`; `GENSAv3`: `smooth = TRUE` and `simple.function = FALSE` - the default version as used in our main experiments; `GENSAv4`: `smooth = FALSE` and `simple.function = FALSE`). As HPO problems we selected three $2D$ problems on which `GENSA` initially showed very poor performance: `wilt_2`, `ada_2`, and `car_2`. We visualize the anytime performance in Figure 2. We observe that setting `smooth = FALSE` (`GENSAv2` and `GENSAv4`) results in substantially better performance while the effect of `simple.function` varies between the benchmark problems (`GENSAv2` outperforming `GENSAv4` on `wilt_2` and `car_2` but not on `ada_2`). With these modifications, the performance of `GENSA` is competitive and closer to the performance observed on the BBOB problems. Future work should further investigate the impact of these parameters on the performance of `GENSA`.

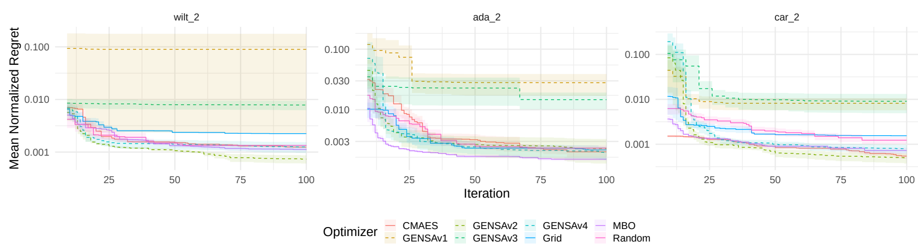

Fig. 2: Anytime mean normalized regret of optimizers and different `GENSA` variants. The x-axis starts after 8% of the optimization budget has been used (initial `MBO` design). Ribbons represent standard errors. 10 replications.

References

1. Becker, M., Richter, J., Lang, M., Bischl, B., Binder, M.: bbotk: Black-Box Optimization Toolkit, https://bbotk.mlr-org.com, https://github.com/mlr-org/bbotk
2. Borchers, H.W.: adagio: Discrete and Global Optimization Routines (2021), https://CRAN.R-project.org/package=adagio, R package version 0.8.4
3. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM Journal of Scientific Computing **16**(5), 1190–1208 (1995)
4. Lang, M., Binder, M., Richter, J., Schratz, P., Pfisterer, F., Coors, S., Au, Q., Casalicchio, G., Kotthoff, L., Bischl, B.: mlr3: A modern object-oriented machine learning framework in R. Journal of Open Source Software **4**(44), 1903 (2019)
5. Le Riche, R., Picheny, V.: Revisiting Bayesian optimization in the light of the COCO benchmark. Structural and Multidisciplinary Optimization **64**(5), 3063–3087 (2021)

6. Nelder, J.A., Mead, R.: A simplex method for function minimization. The Computer Journal **7**(4), 308–313 (1965)
7. Schneider, L., Richter, J., Becker, M., Lang, M., Bischl, B., Binder, M., Moosbauer, J.: mlr3mbo: Flexible Bayesian Optimization in R (2022), https://mlr3mbo.mlr-org.com, https://github.com/mlr-org/mlr3mbo
8. Xiang, Y., Gubian, S., Suomela, B., Hoeng, J.: Generalized simulated annealing for global optimization: The GenSA package. The R Journal **5**(1), 13–28 (2013)