HNCO
Empirical cumulative distribution functions of the runtime of various black box optimization algorithms

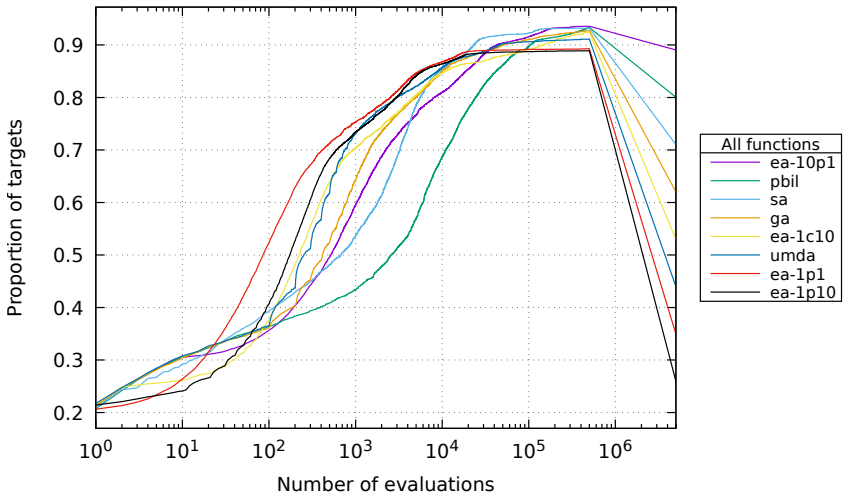
May 18, 2020

Abstract

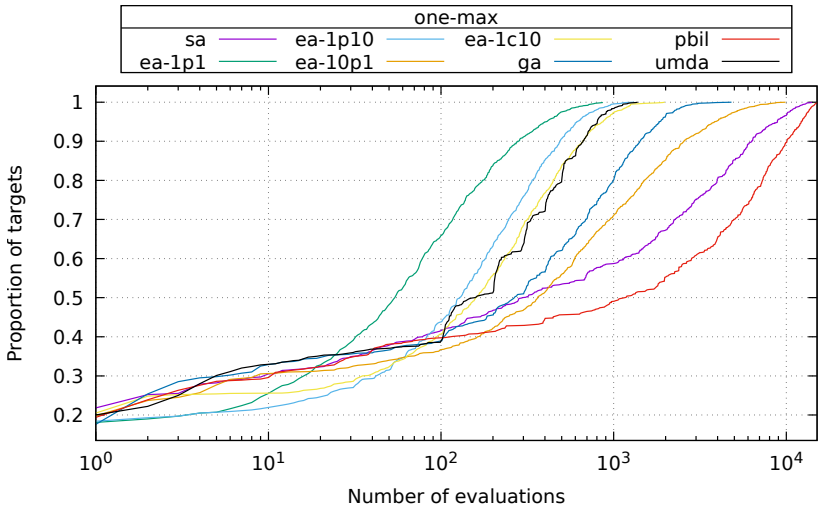
We partly follow the experimental procedure of the COCO framework for the performance assessment of black box optimization algorithms Hansen et al. [2016]. Each algorithm is run independently 20 times on each objective (or fitness) function. The dimension is fixed at $n = 100$. Then 50 equally spaced targets are computed for each objective function. For each algorithm and each function we compute the empirical cumulative distribution function (ECDF) of the runtime, that is the proportion of targets reached as a function of the number of evaluations over all 20 runs. We also compute the global ECDF which takes into account the targets of all functions. The results are listed by function. For clarity reasons only 8 algorithms (hence 8 colors) are included in the study. It should be noted that the linear scale of targets does not fit the function EqualProducts.

Contents

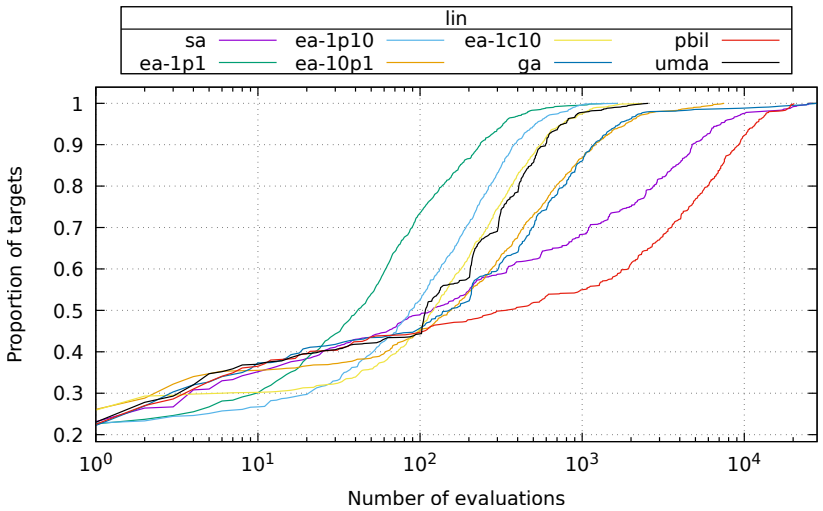
1 All Functions



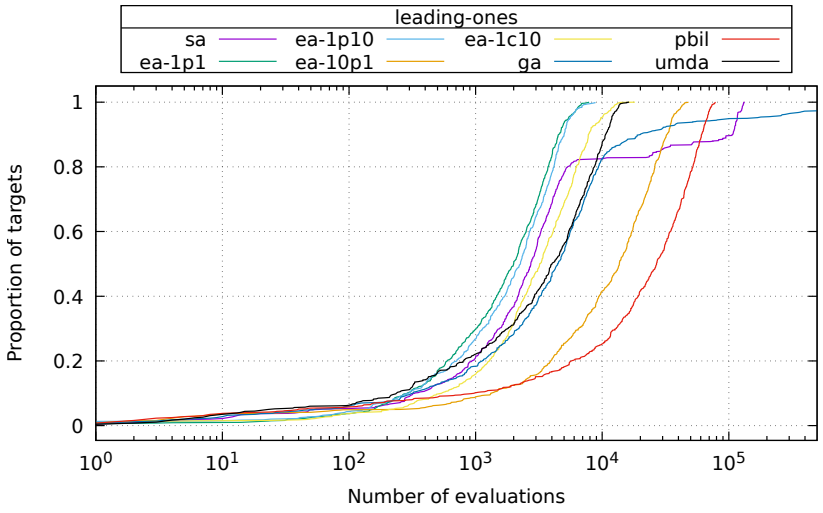
2 Function one-max



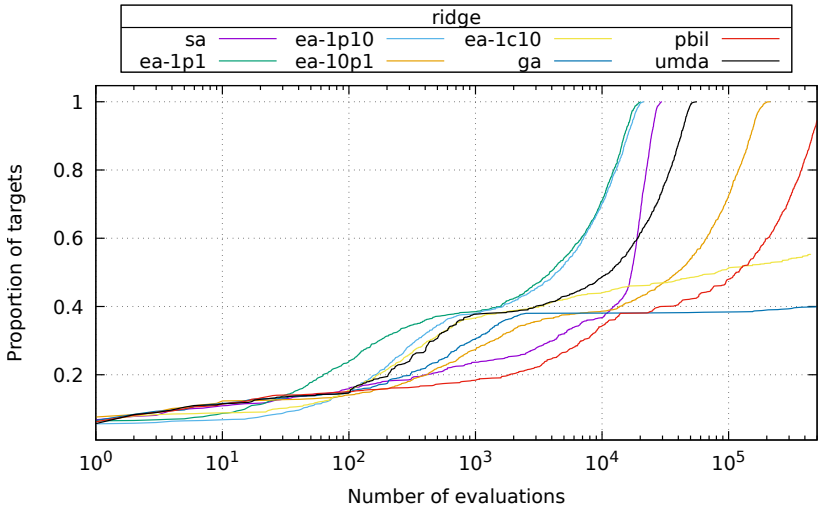
3 Function lin



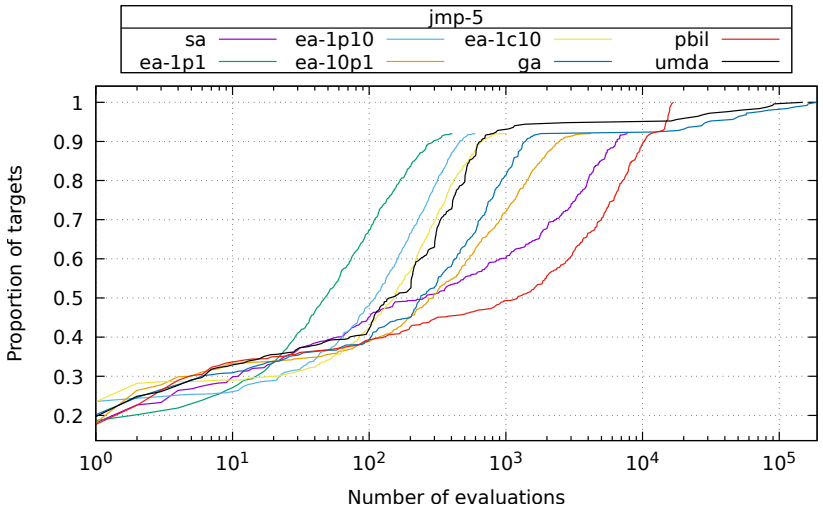
4 Function leading-ones



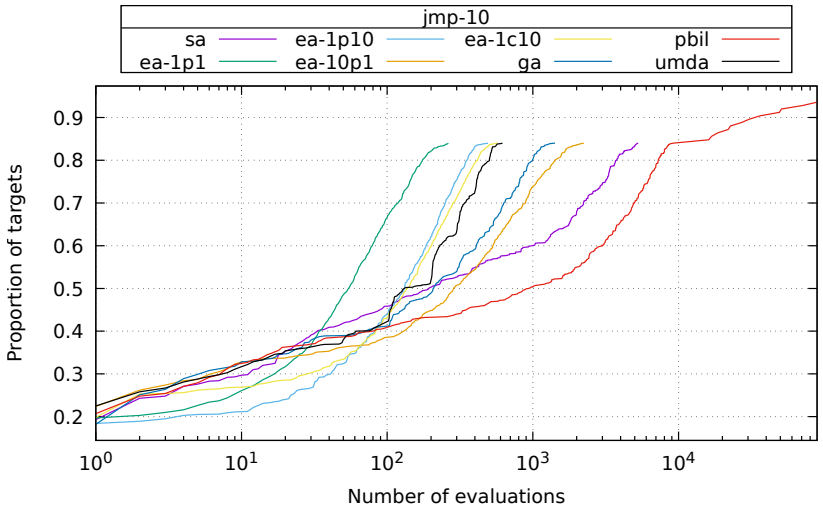
5 Function ridge



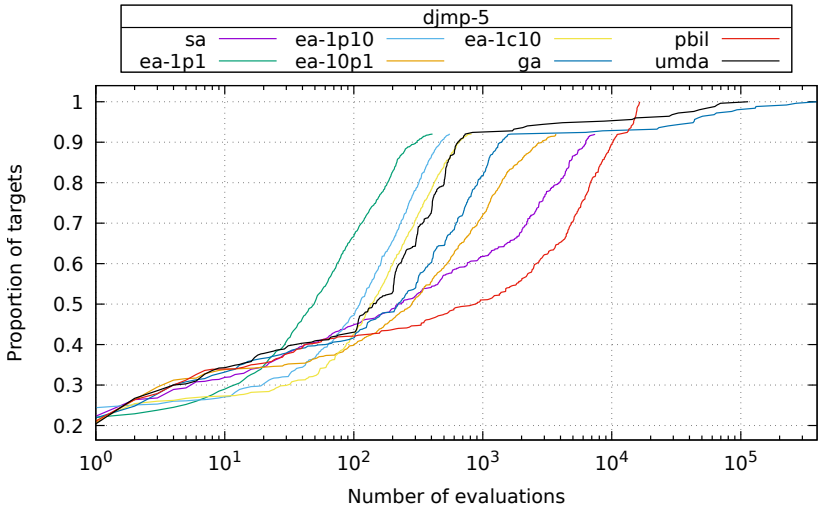
6 Function jmp-5



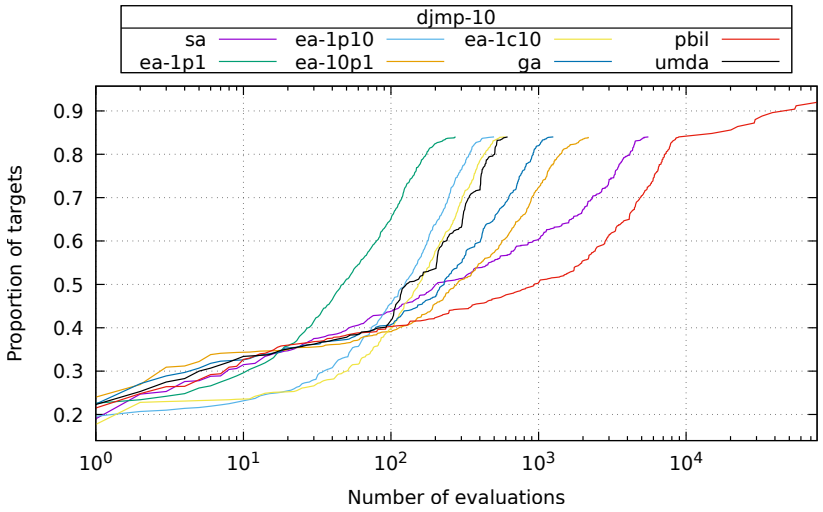
7 Function jmp-10



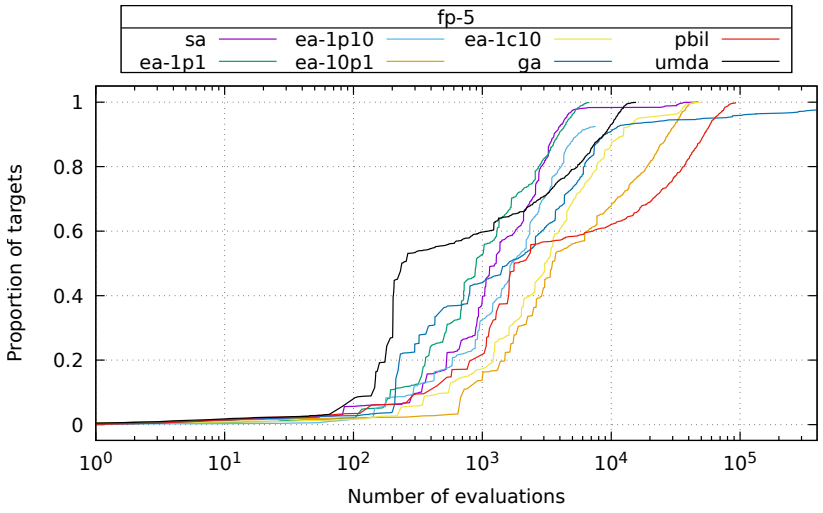
8 Function djmp-5



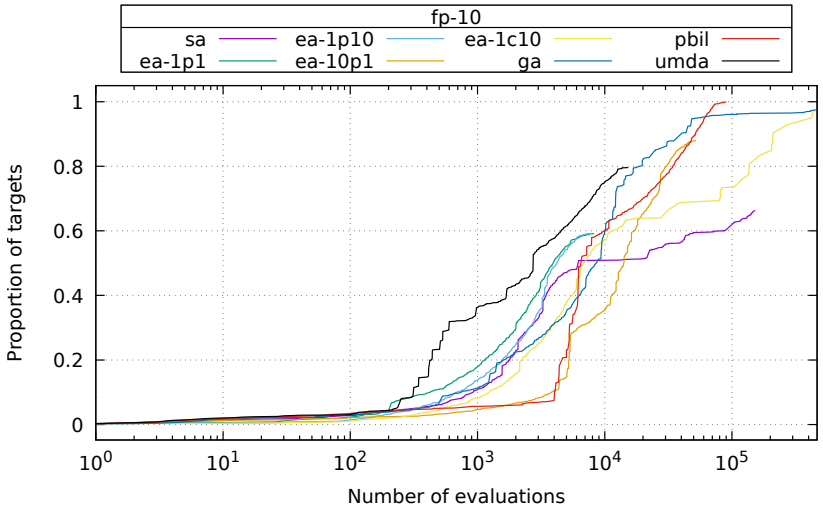
9 Function djmp-10



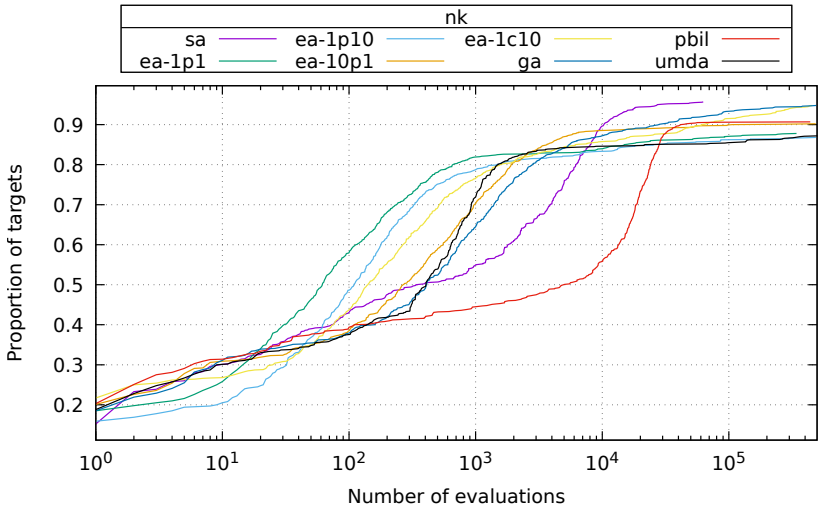
10 Function fp-5



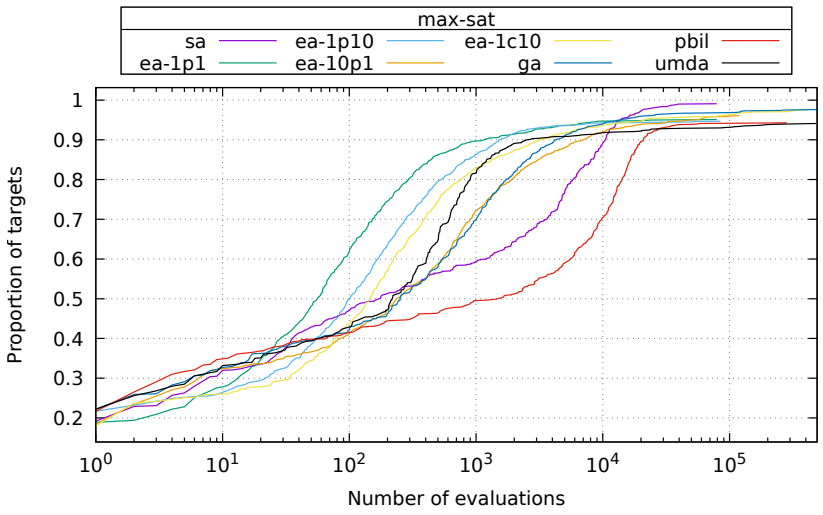
11 Function fp-10



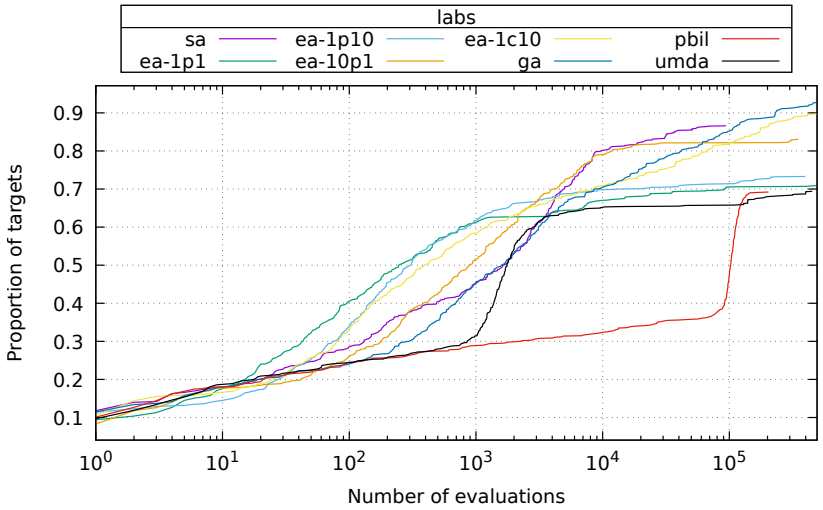
12 Function nk



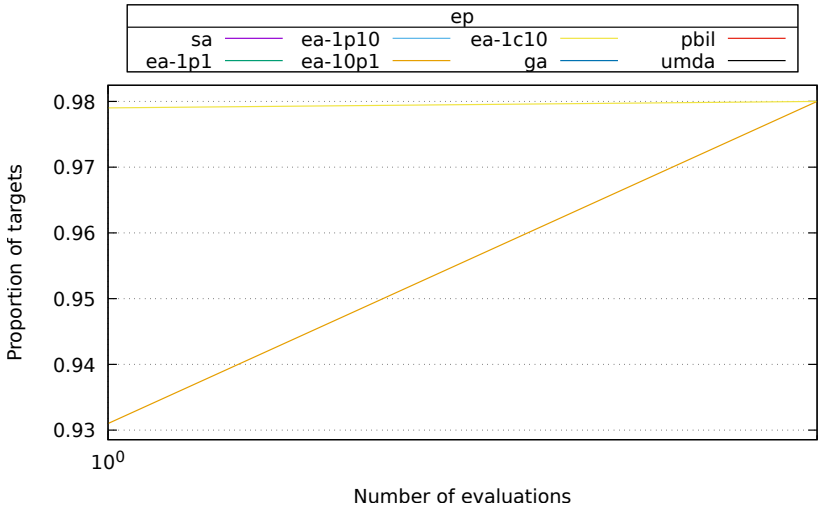
13 Function max-sat



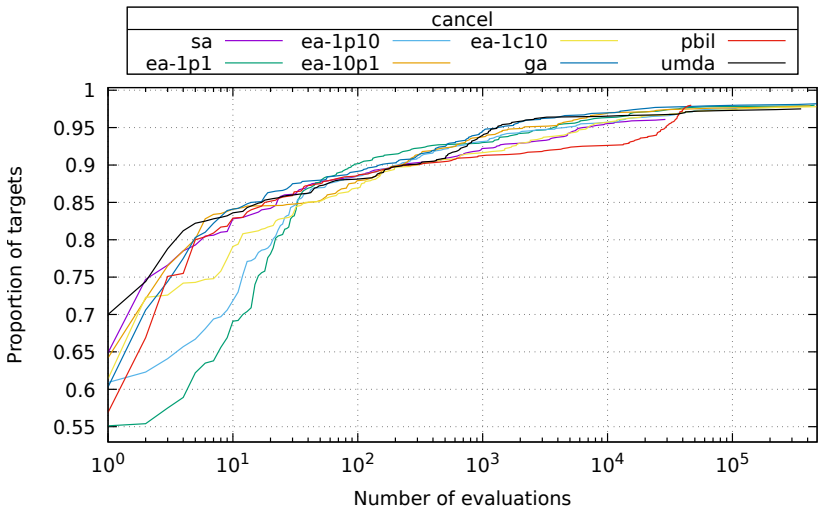
14 Function labs



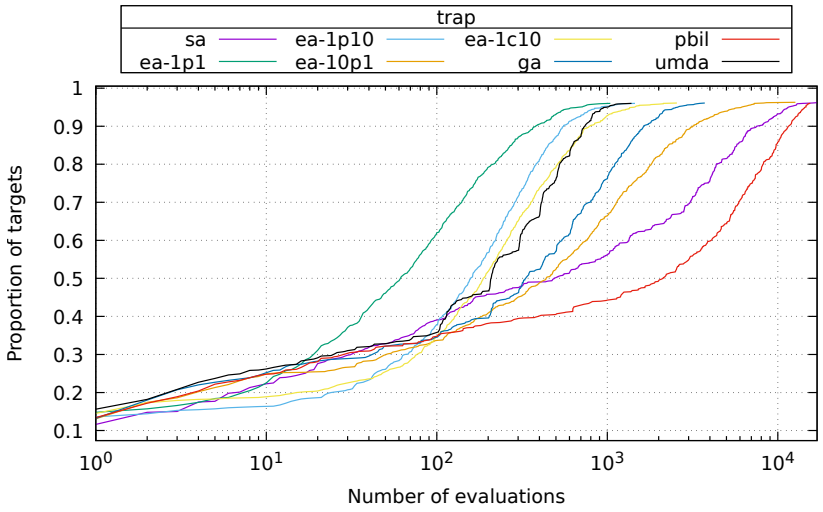
15 Function ep



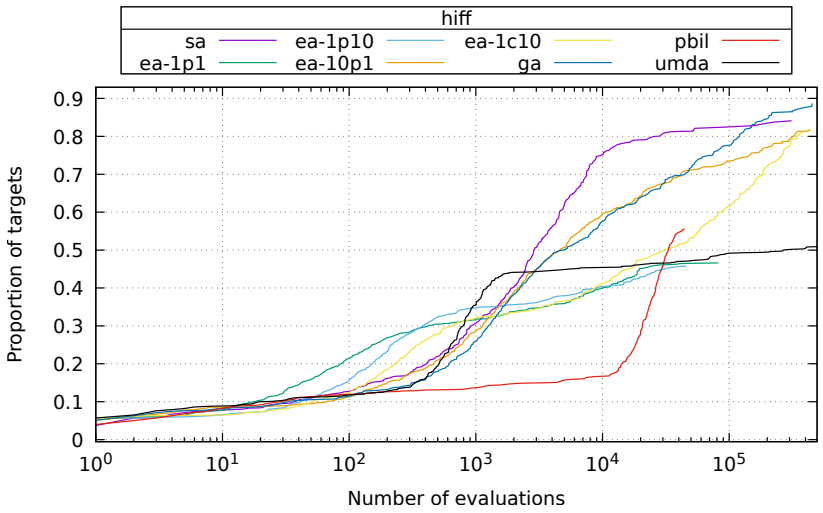
16 Function cancel



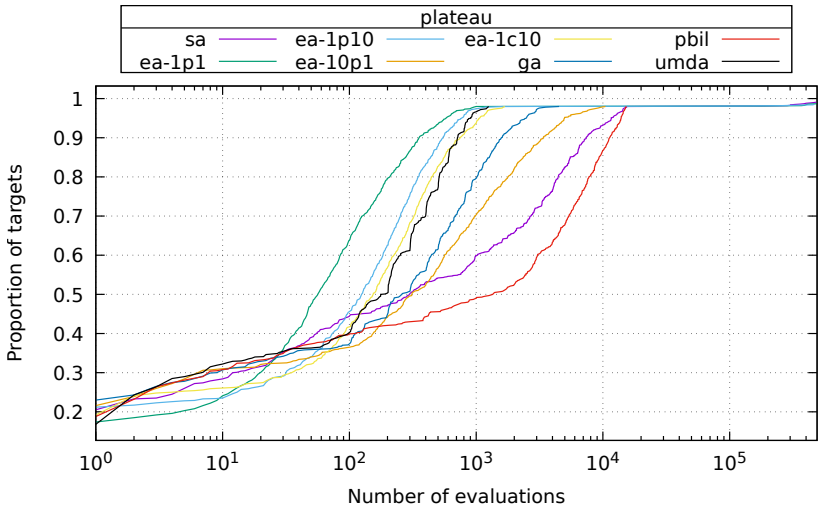
17 Function trap



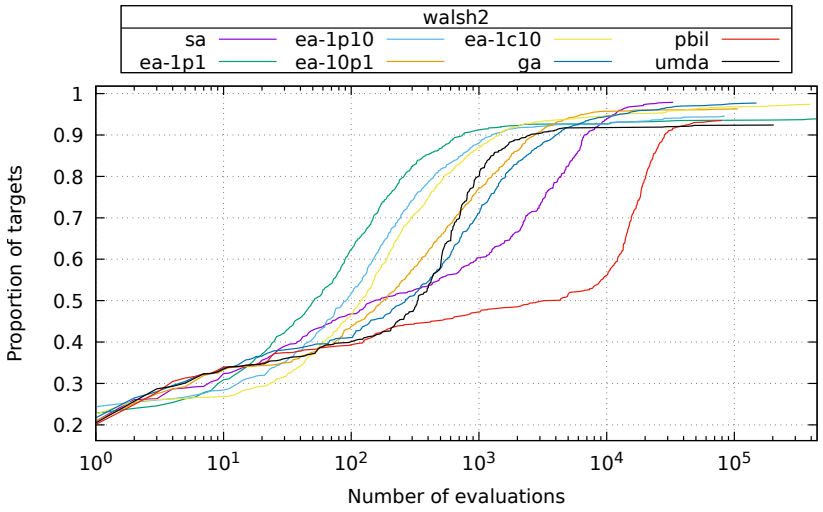
18 Function hiff



19 Function plateau



20 Function walsh2



References

Nikolaus Hansen, Anne Auger, Dimo Brockhoff, Dejan Tusar, and Tea Tusar. COCO: performance assessment. *CoRR*, abs/1605.03560, 2016. URL `http://arxiv.org/abs/1605.03560`.

A Plan


```
{
    "exec": "hnco",
    "opt": "--log-improvement --map 1 --map-random -s 100",
    "budget": 500000,
    "num_runs": 20,
    "num_targets": 50,
    "parallel": true,
    "functions": [
        {
            "id": "one-max",
            "opt": "-F 0 --stop-on-maximum"
        },
        {
            "id": "lin",
            "opt": "-F 1 -p instances/lin.100"
```

```
},
{
    "id": "leading-ones",
    "opt": "-F 10 --stop-on-maximum"
},
{
    "id": "ridge",
    "opt": "-F 11 --stop-on-maximum"
},
{
    "id": "jmp-5",
    "opt": "-F 30 --stop-on-maximum -t 5"
},
{
    "id": "jmp-10",
    "opt": "-F 30 --stop-on-maximum -t 10"
},
{
    "id": "djmp-5",
    "opt": "-F 31 --stop-on-maximum -t 5"
},
{
    "id": "djmp-10",
    "opt": "-F 31 --stop-on-maximum -t 10"
},
{
    "id": "fp-5",
    "opt": "-F 40 --stop-on-maximum -t 5"
},
{
    "id": "fp-10",
    "opt": "-F 40 --stop-on-maximum -t 10"
},
{
    "id": "nk",
    "opt": "-F 60 -p instances/nk.100.4"
},
{
    "id": "max-sat",
    "opt": "-F 70 -p instances/ms.100.3.1000"
},
{
    "id": "labs",
    "opt": "-F 81"
},
{
    "id": "ep",
    "opt": "-F 90 -p instances/ep.100",
    "reverse": true,
    "logscale": true
},
{
    "id": "cancel",
    "opt": "-F 100 -s 99",
    "reverse": true
},
{
    "id": "trap",
    "opt": "-F 110 --stop-on-maximum --fn-num-traps 10"
},
{
    "id": "hiff",
    "opt": "-F 120 --stop-on-maximum -s 128"
```

```
        },
        {
            "id": "plateau",
            "opt": "-F 130 --stop-on-maximum"
        },
        {
            "id": "walsh2",
            "opt": "-F 162 -p instances/walsh2.100"
        }
    ],
    "algorithms": [
        {
            "id": "sa",
            "opt": "-A 200 --sa-beta-ratio 1.05 --sa-num-trials 10"
        },
        {
            "id": "ea-1p1",
            "opt": "-A 300"
        },
        {
            "id": "ea-1p10",
            "opt": "-A 310 --ea-mu 1 --ea-lambda 10"
        },
        {
            "id": "ea-10p1",
            "opt": "-A 310 --ea-mu 10 --ea-lambda 1"
        },
        {
            "id": "ea-1c10",
            "opt": "-A 320 --ea-mu 1 --ea-lambda 10 --allow-stay"
        },
        {
            "id": "ga",
            "opt": "-A 400 --ea-mu 100"
        },
        {
            "id": "pbil",
            "opt": "-A 500 -r 5e-3"
        },
        {
            "id": "umda",
            "opt": "-A 600 -x 100 -y 10"
        }
    ]
}
```

B Default parameters

```
# algorithm = 100
# bm_mc_reset_strategy = 1
# bm_num_gs_cycles = 1
# bm_num_gs_steps = 100
# bm_sampling = 1
# budget = 10000
# bv_size = 100
# description_path = description.txt
# ea_lambda = 100
# ea_mu = 10
# expression = x
# fn_name = noname
# fn_num_traps = 10
# fn_prefix_length = 2
```

```
# fn_threshold = 10
# function = 0
# ga_crossover_bias = 0.5
# ga_crossover_probability = 0.5
# ga_tournament_size = 10
# hea_bit_herding = 0
# hea_num_seq_updates = 100
# hea_reset_period = 0
# hea_sampling_method = 0
# hea_weight = 1
# learning_rate = 0.001
# map = 0
# map_input_size = 100
# map_path = map.txt
# map_ts_length = 10
# map_ts_sampling_mode = 0
# mutation_probability = 1
# neighborhood = 0
# neighborhood_iterator = 0
# noise_stddev = 1
# num_iterations = 0
# num_threads = 1
# path = function.txt
# pn_mutation_probability = 1
# pn_neighborhood = 0
# pn_radius = 2
# population_size = 10
# pv_log_num_components = 5
# radius = 2
# real_expression = (1-x)^2+100*(y-x^2)^2
# real_lower_bound = -2
# real_num_bits = 8
# real_upper_bound = 2
# results_path = results.json
# rls_patience = 50
# sa_beta_ratio = 1.2
# sa_initial_acceptance_probability = 0.6
# sa_num_transitions = 50
# sa_num_trials = 100
# seed = 0
# selection_size = 1
# solution_path = solution.txt
# target = 100
# print_defaults
# last_parameter
# exec_name = hnco
# version = 0.14
# Generated from hnco.json
```