HNCO
Influence of the learning rate on the performance of PBIL

August 7, 2020

Abstract

PBIL is applied many times to the same collection of fitness functions (bit vector size $n = 100$), each time with a different learning rate taken from a finite set of values. All learning rates are ranked according to their median fitness over 20 independent runs, first for each fitness function, then across the entire collection of fitness functions. The mean and standard deviation of fitness are also plotted as a function of the learning rate.

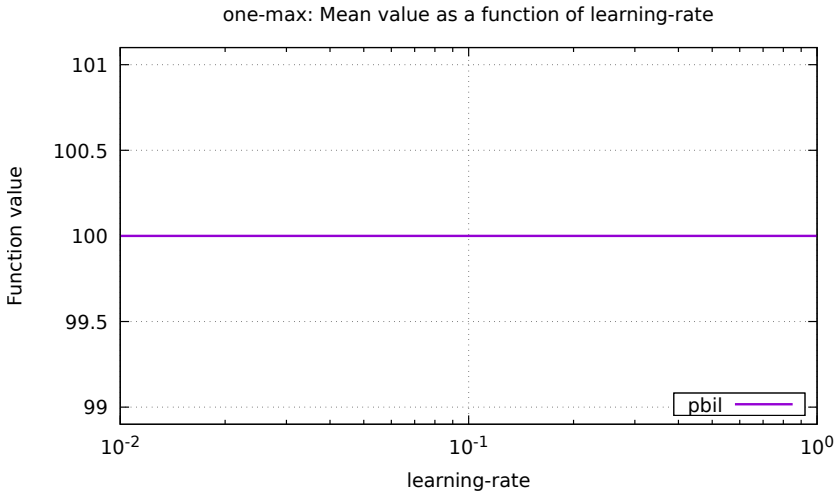
Contents

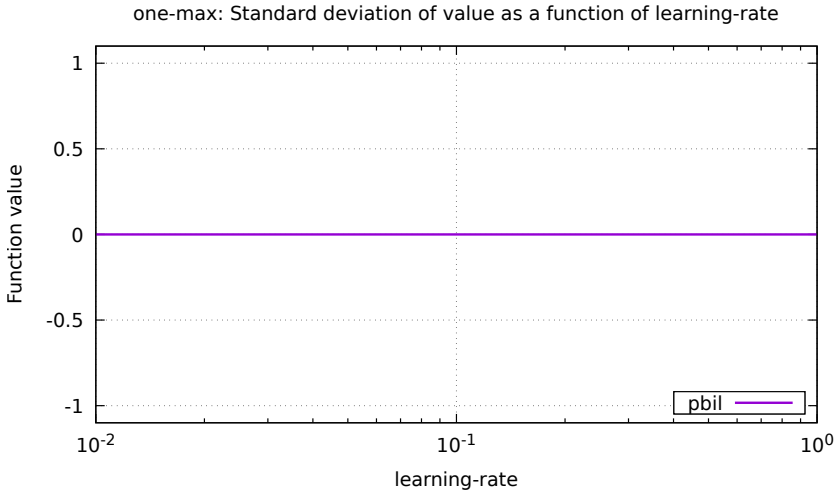
1 Rankings

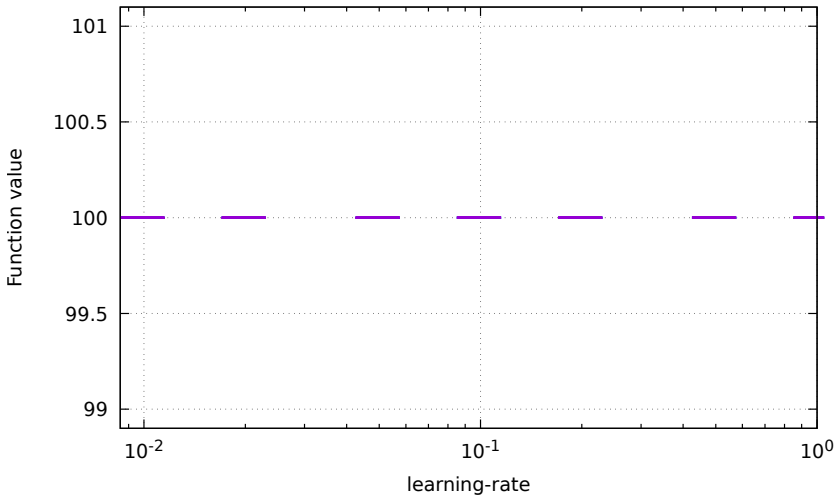
algorithm	learning-rate	rank distribution						
		1	2	3	4	5	6	7
pbil	1	6	1	0	0	1	1	0
pbil	0.01	5	3	0	0	0	1	0
pbil	0.02	3	1	2	0	1	1	1
pbil	0.05	3	0	0	2	2	2	0
pbil	0.5	2	1	3	0	1	2	0
pbil	0.2	2	0	1	1	1	0	4
pbil	0.1	2	0	0	4	1	1	1

2 Function one-max

algorithm	learning-rate	function value					
		min	Q_1	med.	Q_3	max	rk
pbil	0.01	100	100	100	100	100	1
pbil	0.02	100	100	100	100	100	1
pbil	0.05	100	100	100	100	100	1
pbil	0.1	100	100	100	100	100	1
pbil	0.2	100	100	100	100	100	1
pbil	0.5	100	100	100	100	100	1
pbil	1	100	100	100	100	100	1

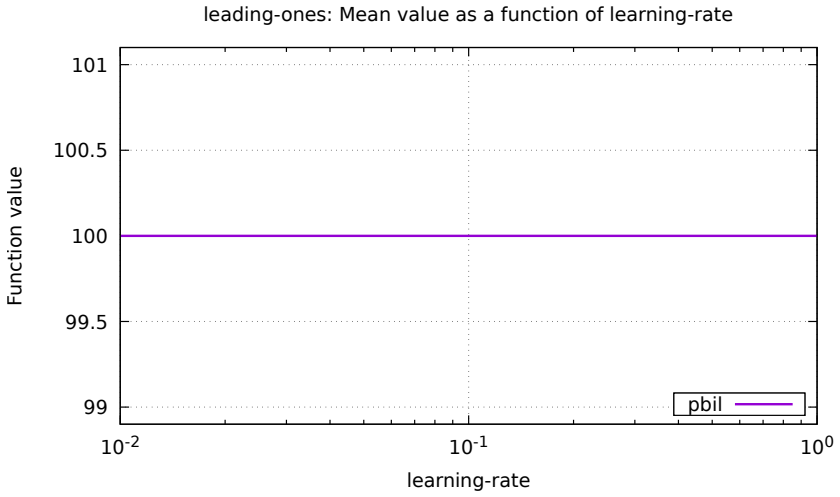


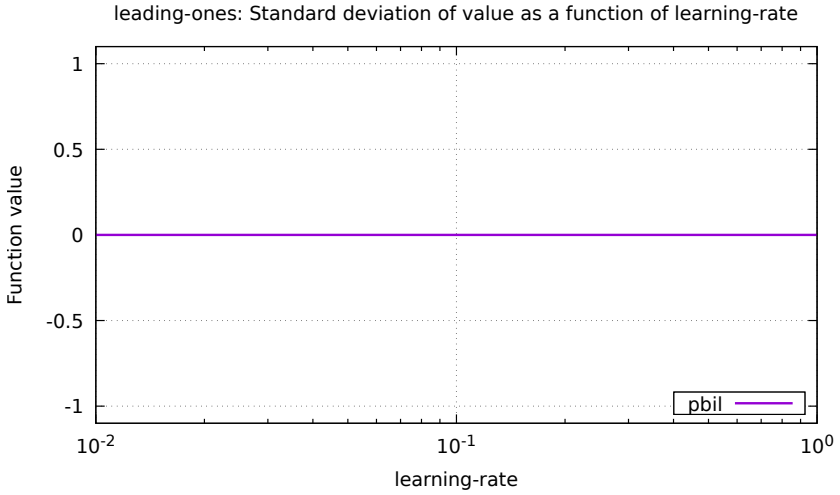


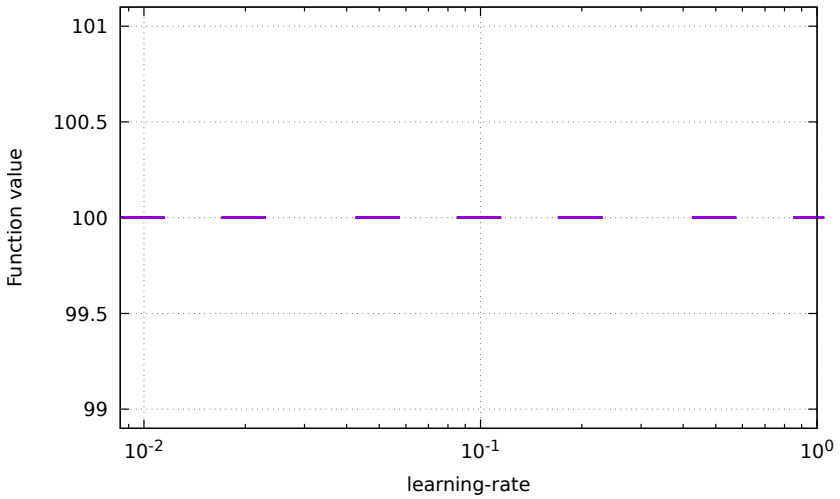


3 Function leading-ones

algorithm	learning-rate	function value					
		min	Q_1	med.	Q_3	max	rk
pbil	0.01	100	100	100	100	100	1
pbil	0.02	100	100	100	100	100	1
pbil	0.05	100	100	100	100	100	1
pbil	0.1	100	100	100	100	100	1
pbil	0.2	100	100	100	100	100	1
pbil	0.5	100	100	100	100	100	1
pbil	1	100	100	100	100	100	1

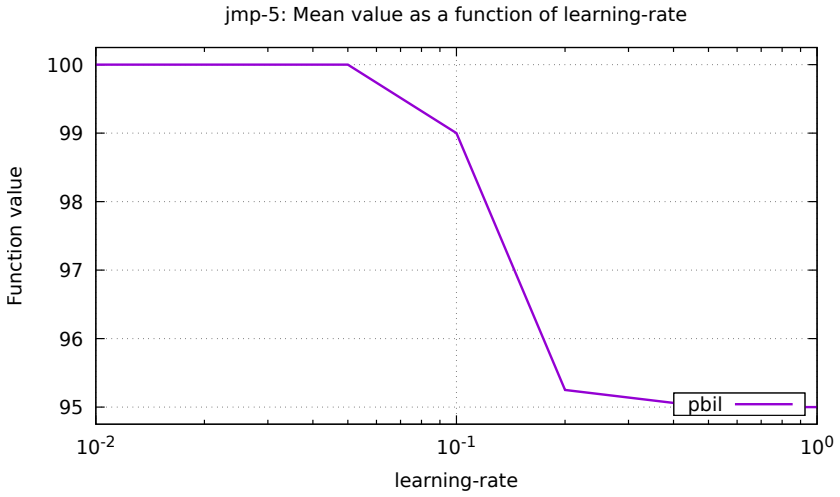


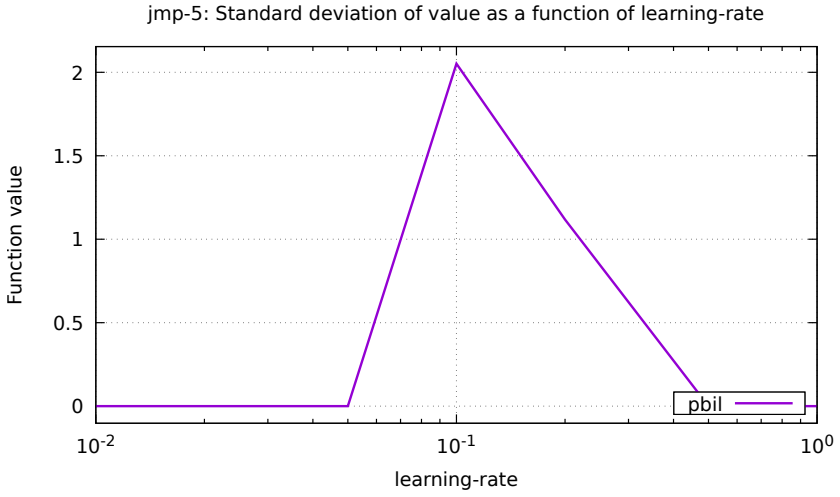


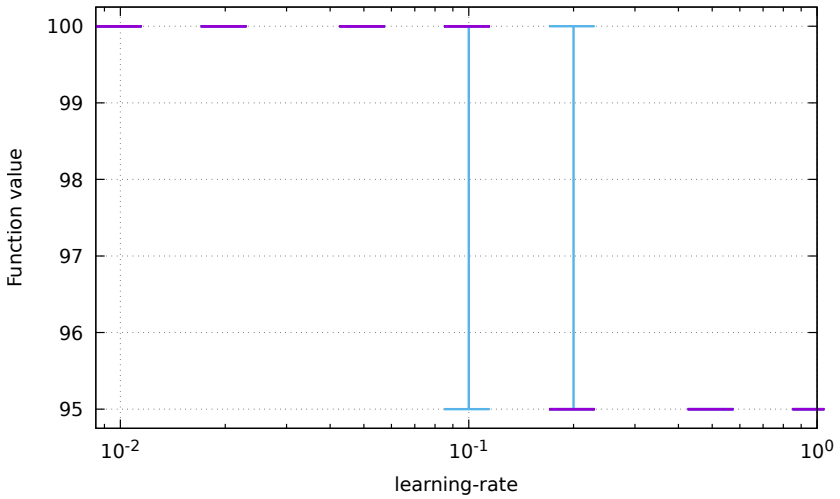


4 Function jmp-5

algorithm	learning-rate	function value					
		min	Q_1	med.	Q_3	max	rk
pbil	0.01	100	100	100	100	100	1
pbil	0.02	100	100	100	100	100	1
pbil	0.05	100	100	100	100	100	1
pbil	0.1	95	100	100	100	100	4
pbil	0.2	95	95	95	95	100	5
pbil	0.5	95	95	95	95	95	6
pbil	1	95	95	95	95	95	6

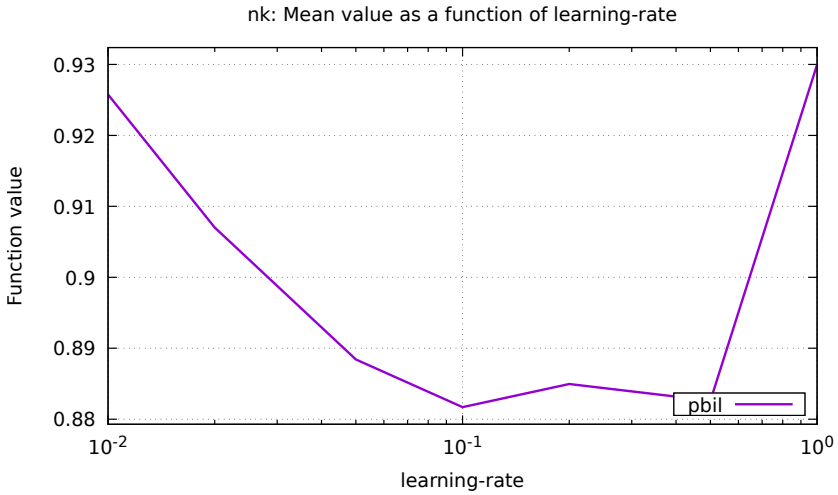


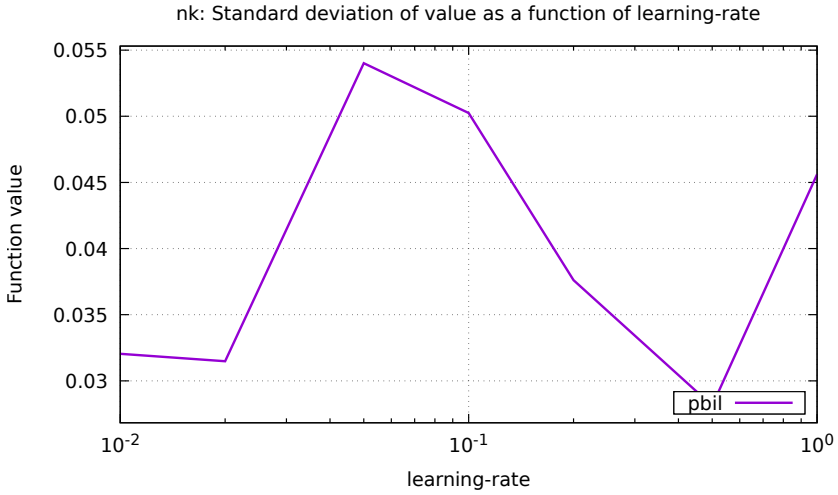


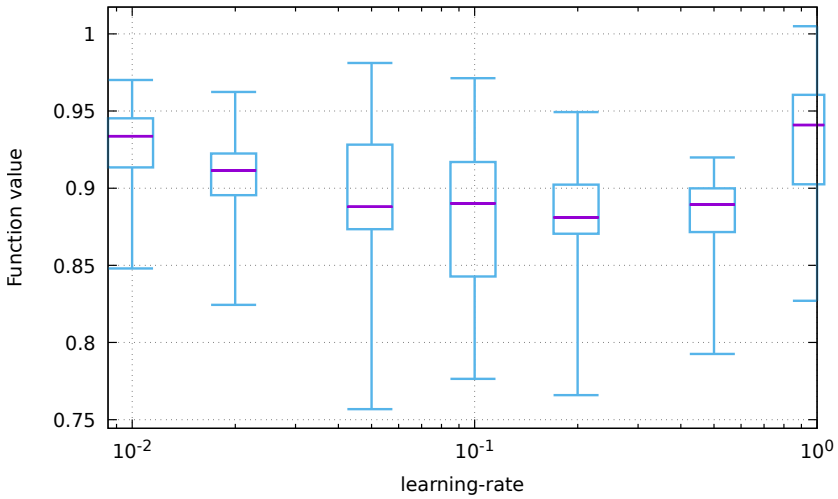


5 Function nk

algorithm	learning-rate	function value					
		min	Q_1	med.	Q_3	max	rk
pbil	0.01	0.85	0.91	0.93	0.95	0.97	2
pbil	0.02	0.82	0.90	0.91	0.92	0.96	3
pbil	0.05	0.76	0.87	0.89	0.93	0.98	6
pbil	0.1	0.78	0.84	0.89	0.92	0.97	4
pbil	0.2	0.77	0.87	0.88	0.90	0.95	7
pbil	0.5	0.79	0.87	0.89	0.90	0.92	5
pbil	1	0.83	0.90	0.94	0.96	1.00	1

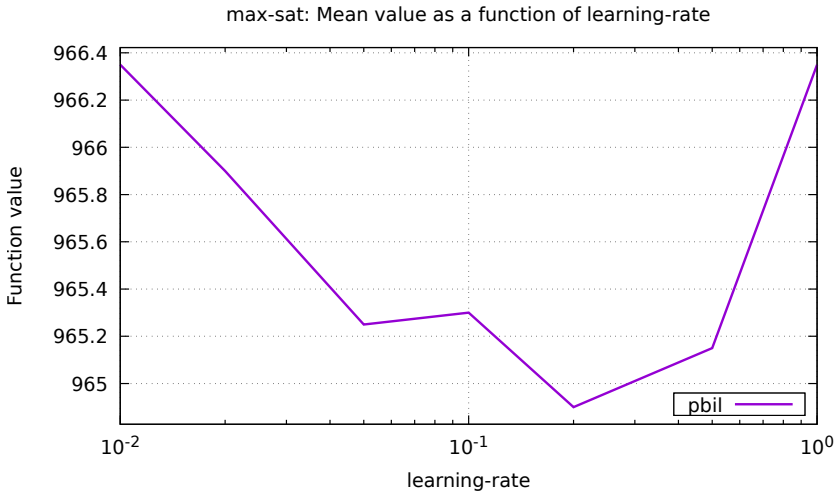


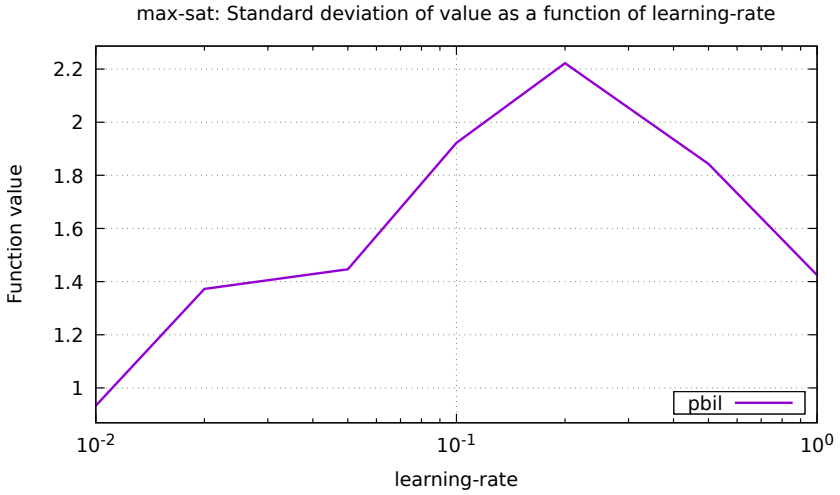


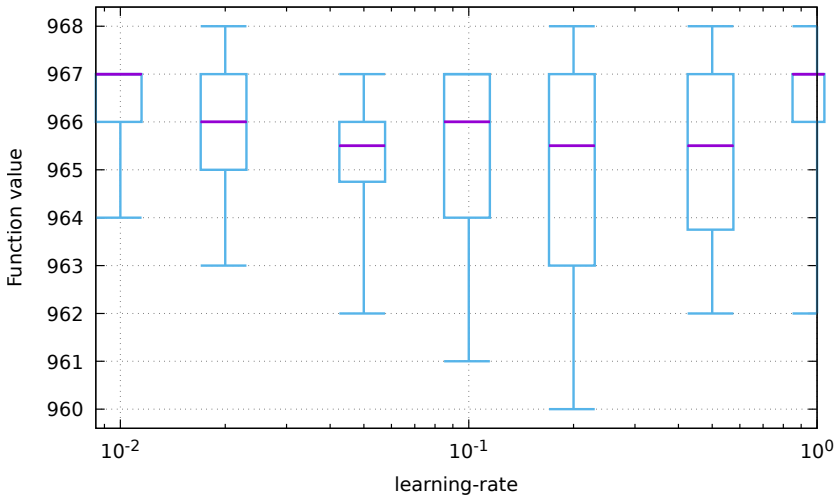


6 Function max-sat

algorithm	learning-rate	function value					
		min	Q_1	med.	Q_3	max	rk
pbil	0.01	964	966	967	967	967	1
pbil	0.02	963	965	966	967	968	3
pbil	0.05	962	965	966	966	967	5
pbil	0.1	961	964	966	967	967	4
pbil	0.2	960	963	966	967	968	7
pbil	0.5	962	964	966	967	968	6
pbil	1	962	966	967	967	968	2

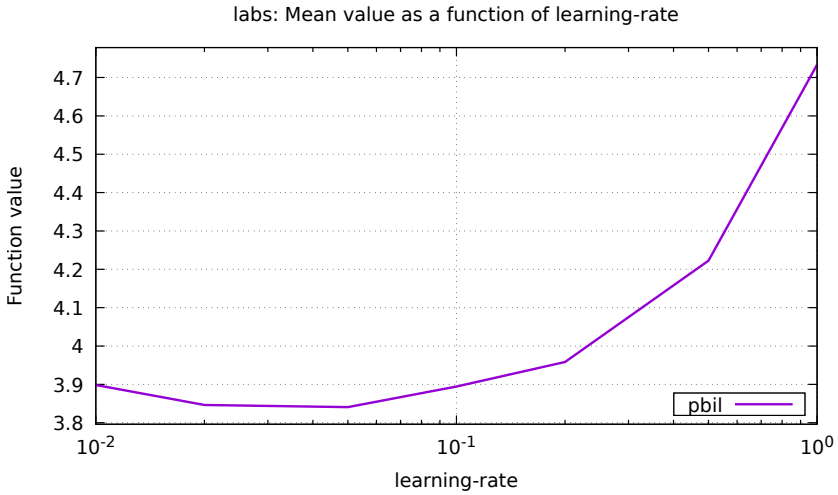


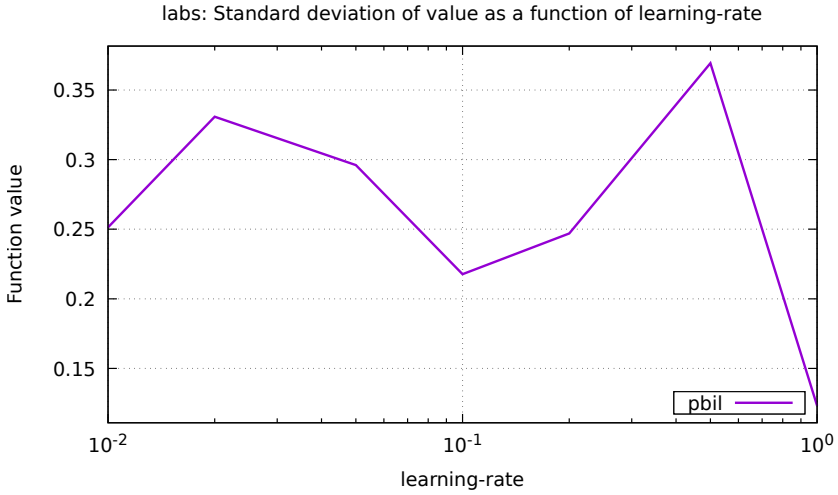


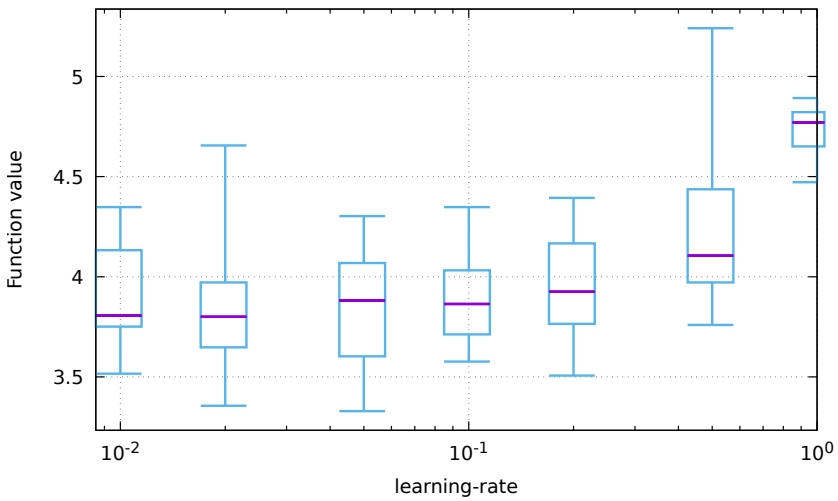


7 Function labs

algorithm	learning-rate	function value					
		min	Q_1	med.	Q_3	max	rk
pbil	0.01	3.52	3.75	3.81	4.13	4.35	6
pbil	0.02	3.36	3.65	3.80	3.97	4.66	7
pbil	0.05	3.33	3.60	3.88	4.07	4.30	4
pbil	0.1	3.58	3.71	3.86	4.03	4.35	5
pbil	0.2	3.51	3.77	3.92	4.17	4.39	3
pbil	0.5	3.76	3.97	4.11	4.44	5.24	2
pbil	1	4.47	4.65	4.77	4.82	4.89	1

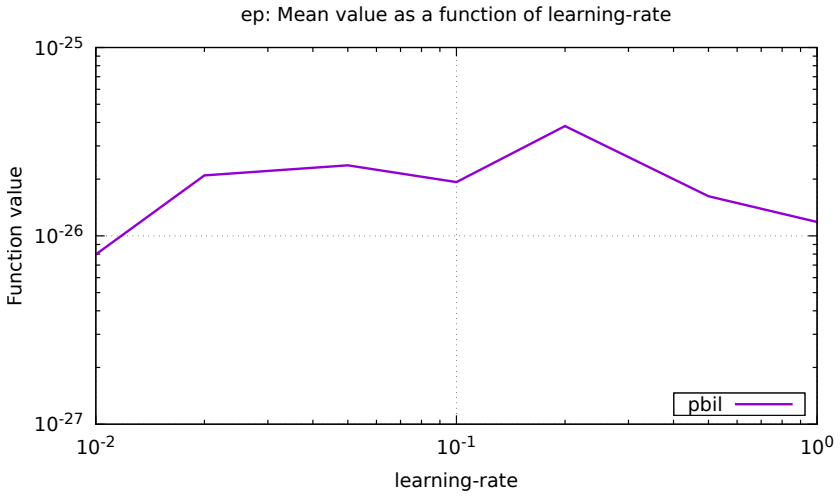


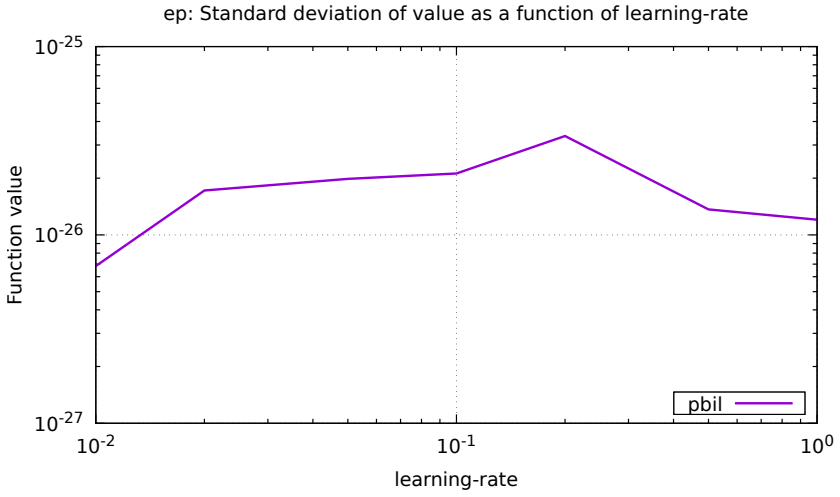


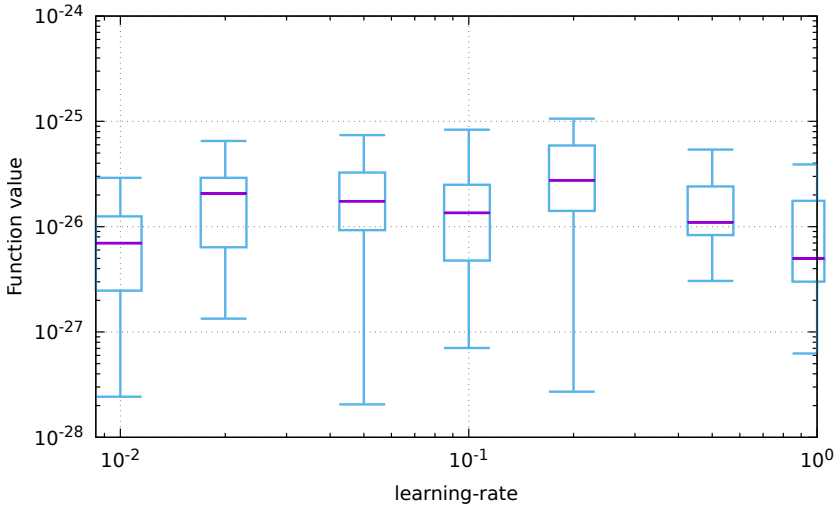


8 Function ep

algorithm	learning-rate	function value					
		min	Q_1	med.	Q_3	max	rk
pbil	0.01	2.4×10^{-28}	2.5×10^{-27}	7.0×10^{-27}	1.3×10^{-26}	2.9×10^{-26}	2
pbil	0.02	1.3×10^{-27}	6.4×10^{-27}	2.1×10^{-26}	2.9×10^{-26}	6.5×10^{-26}	6
pbil	0.05	2.1×10^{-28}	9.3×10^{-27}	1.7×10^{-26}	3.3×10^{-26}	7.4×10^{-26}	5
pbil	0.1	7.0×10^{-28}	4.8×10^{-27}	1.4×10^{-26}	2.5×10^{-26}	8.3×10^{-26}	4
pbil	0.2	2.7×10^{-28}	1.4×10^{-26}	2.8×10^{-26}	5.9×10^{-26}	1.1×10^{-25}	7
pbil	0.5	3.1×10^{-27}	8.3×10^{-27}	1.1×10^{-26}	2.4×10^{-26}	5.4×10^{-26}	3
pbil	1	6.2×10^{-28}	3.0×10^{-27}	5.0×10^{-27}	1.8×10^{-26}	3.9×10^{-26}	1

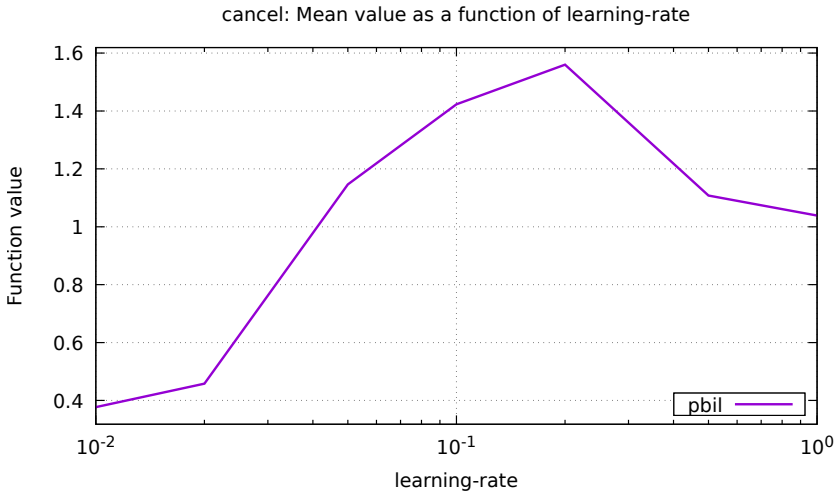


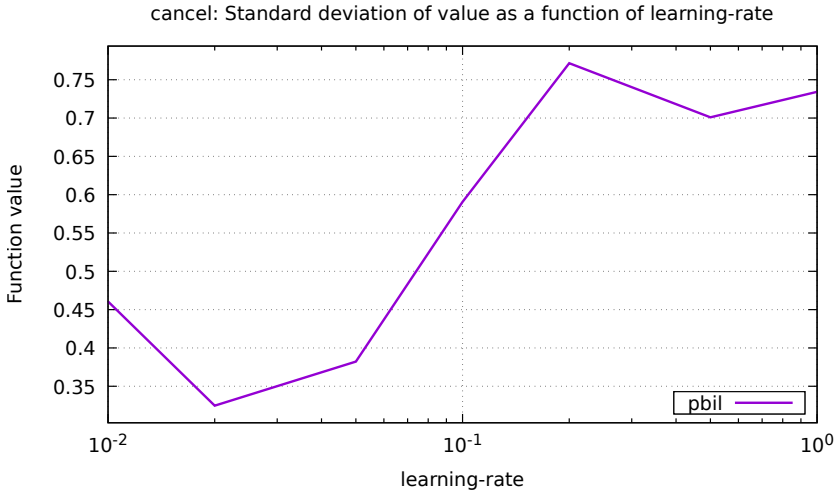


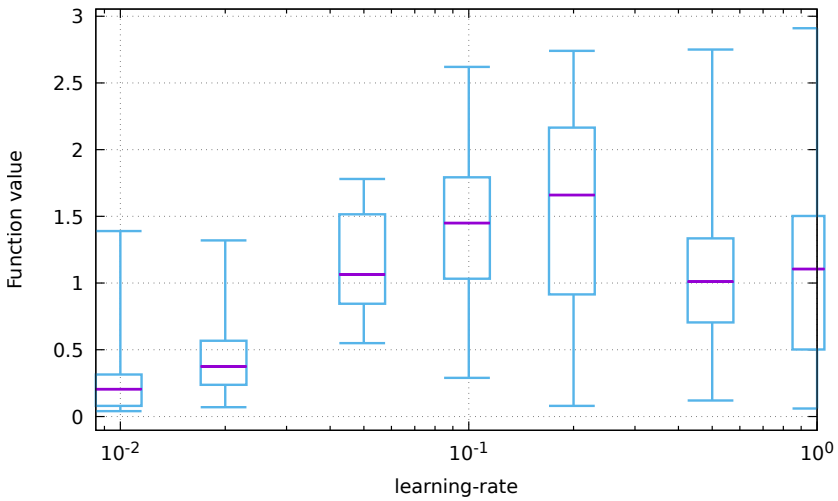


9 Function cancel

algorithm	learning-rate	function value					
		min	Q_1	med.	Q_3	max	rk
pbil	0.01	0.04	0.08	0.21	0.32	1.39	1
pbil	0.02	0.07	0.24	0.38	0.57	1.32	2
pbil	0.05	0.55	0.85	1.07	1.52	1.78	4
pbil	0.1	0.29	1.03	1.45	1.79	2.62	6
pbil	0.2	0.08	0.92	1.66	2.17	2.74	7
pbil	0.5	0.12	0.71	1.01	1.34	2.75	3
pbil	1	0.06	0.50	1.11	1.50	2.91	5

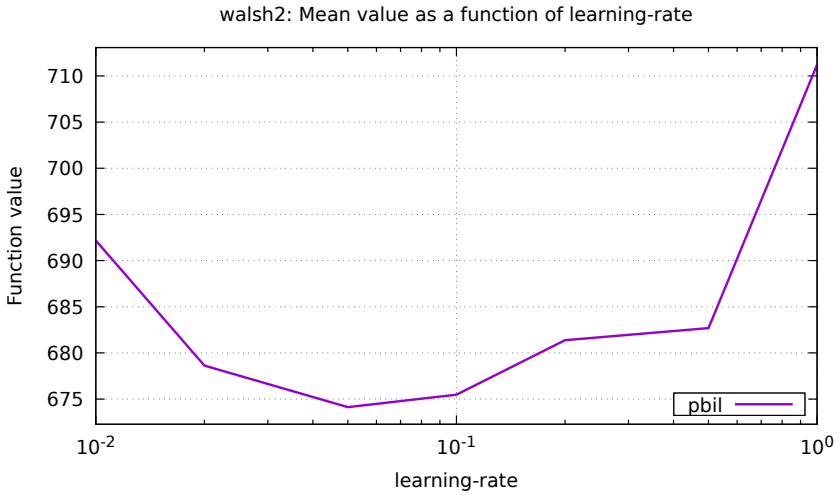


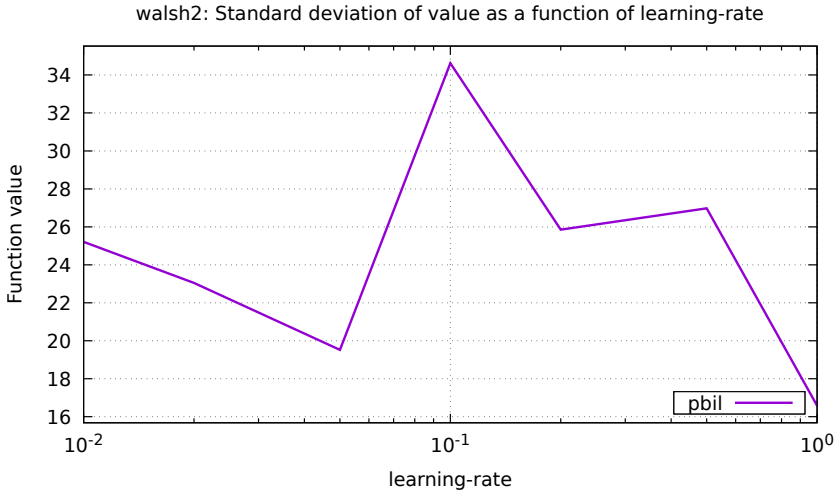


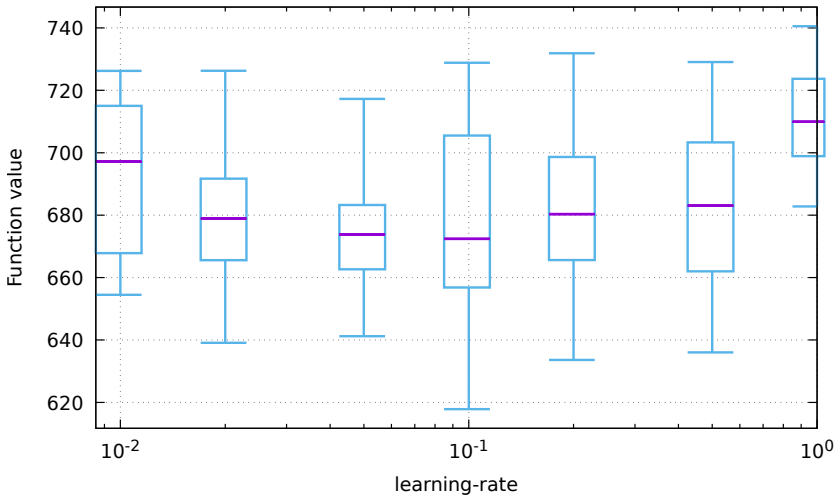


10 Function walsh2

algorithm	learning-rate	function value					
		min	Q_1	med.	Q_3	max	rk
pbil	0.01	654.47	667.82	697.14	715.02	726.23	2
pbil	0.02	639.08	665.57	678.92	691.71	726.29	5
pbil	0.05	641.21	662.65	673.83	683.24	717.26	6
pbil	0.1	617.83	656.83	672.39	705.50	728.83	7
pbil	0.2	633.62	665.62	680.33	698.63	731.88	4
pbil	0.5	636.03	662.00	683.02	703.33	729.08	3
pbil	1	682.81	698.91	709.93	723.70	740.55	1







A Plan


```
{
    "exec": "hnco",
    "opt": "--print-results --map 1 --map-random -s 100",
    "budget": 200000,
    "num_runs": 20,
    "parallel": true,
    "parameter": {
        "id": "learning-rate",
        "values": [ 1e-2, 2e-2, 5e-2, 1e-1, 2e-1, 5e-1, 1 ]
    },
    "graphics": {
        "logscale": true,
        "candlesticks": {
            "boxwidth": "$1 * 0.3"
        }
    },
    "functions": [
        {
            "id": "one-max",
            "opt": "-F 0 --stop-on-maximum",
            "rounding": {
                "value": { "before": 3, "after": 0 },
                "time": { "before": 1, "after": 2 } }
        },
        {
            "id": "leading-ones",
            "opt": "-F 10 --stop-on-maximum",
            "rounding": {
                "value": { "before": 3, "after": 0 },
                "time": { "before": 1, "after": 2 } }
        },
        {
            "id": "jmp-5",
            "opt": "-F 30 --stop-on-maximum -t 5",
            "rounding": {
                "value": { "before": 3, "after": 0 },
                "time": { "before": 1, "after": 2 } }
        },
        {
            "id": "nk",
            "opt": "-F 60 -p instances/nk.100.4",
            "rounding": {
                "value": { "before": 1, "after": 2 },
                "time": { "before": 1, "after": 2 } }
        },
        {
            "id": "max-sat",
            "opt": "-F 70 -p instances/ms.100.3.1000",
            "rounding": {
                "value": { "before": 3, "after": 0 },
                "time": { "before": 1, "after": 2 } }
        },
        {
            "id": "labs",
            "opt": "-F 81",
            "rounding": {
                "value": { "before": 1, "after": 2 },
                "time": { "before": 1, "after": 2 } }
        },
        {
            "id": "ep",
```

```
            "opt": "-F 90 -p instances/ep.100",
            "reverse": true,
            "logscale": true,
            "rounding": {
                "value": { "before": 1, "after": 1 },
                "time": { "before": 1, "after": 2 } }
        },
        {
            "id": "cancel",
            "opt": "-F 100 -s 99",
            "reverse": true,
            "rounding": {
                "value": { "before": 1, "after": 2 },
                "time": { "before": 1, "after": 2 } }
        },
        {
            "id": "walsh2",
            "opt": "-F 162 -p instances/walsh2.100",
            "rounding": {
                "value": { "before": 3, "after": 2 },
                "time": { "before": 1, "after": 2 } }
        }
    ],
    "algorithms": [
        {
            "id": "pbil",
            "opt": "-A 500 -x 10 -y 1"
        }
    ]
}
```

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
# fp_expression = (1-x)^2+100*(y-x^2)^2
# fp_lower_bound = -2
# fp_num_bits = 8
# fp_upper_bound = 2
# 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_rate = 1
# neighborhood = 0
# neighborhood_iterator = 0
# noise_stddev = 1
# num_iterations = 0
# num_threads = 1
# path = function.txt
# pn_mutation_rate = 1
# pn_neighborhood = 0
# pn_radius = 2
# population_size = 10
# pv_log_num_components = 5
# radius = 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 = 0.15
# Generated from hnco.json
```