HNCO
Influence of the learning rate
on the performance of PBIL

March 23, 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.
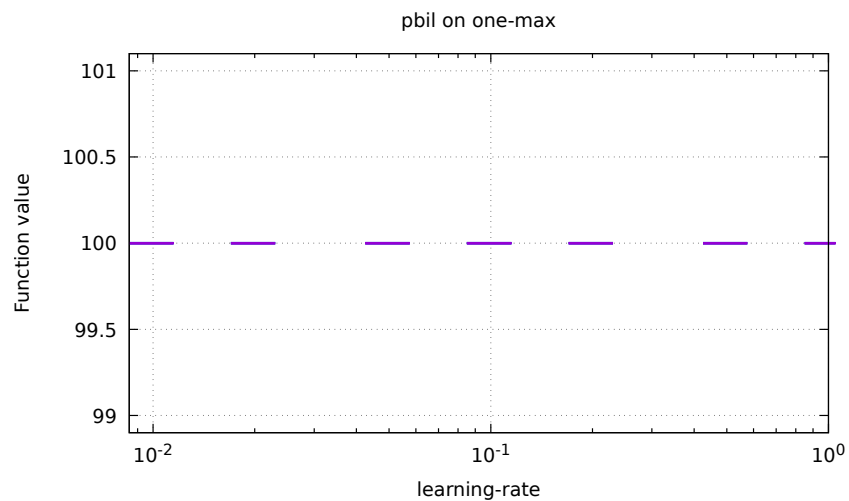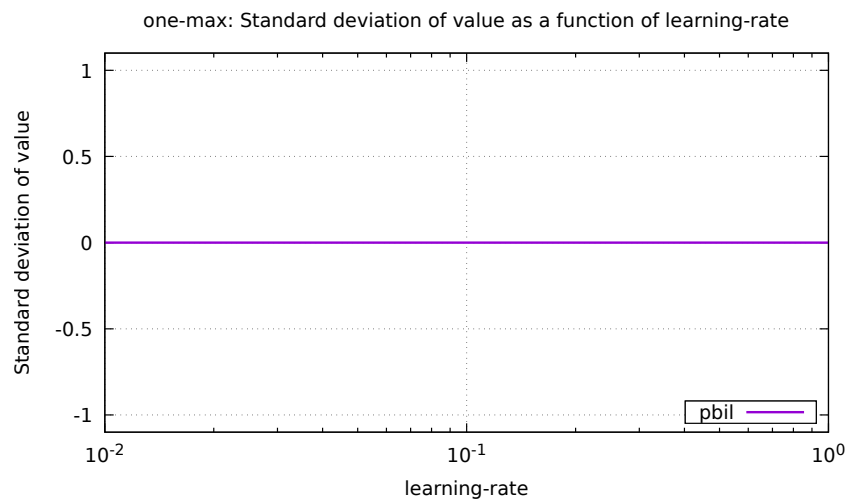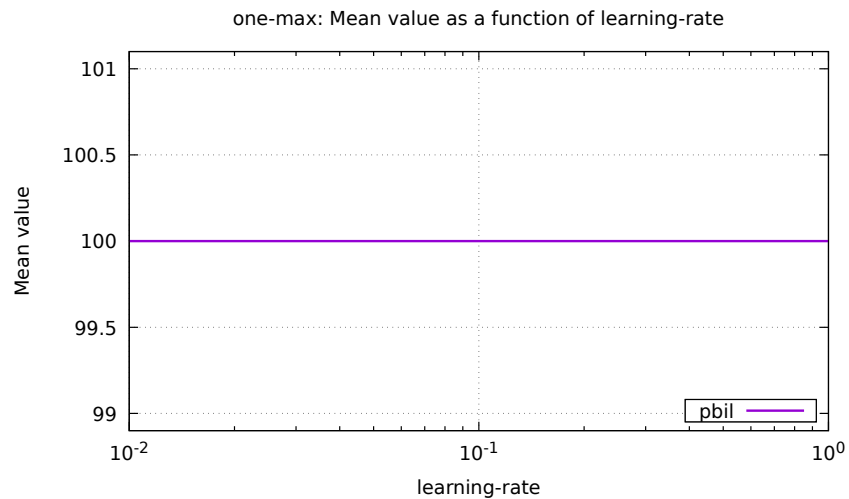
Contents

1 Rankings

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

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

one-max: Mean value as a function of learning-rate

one-max: Standard deviation of value as a function of learning-rate

pbil on one-max

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

leading-ones: Mean value as a function of learning-rate

leading-ones: Standard deviation of value as a function of learning-rate

pbil on leading-ones

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

jmp-5: Mean value as a function of learning-rate

jmp-5: Standard deviation of value as a function of learning-rate

pbil on jmp-5
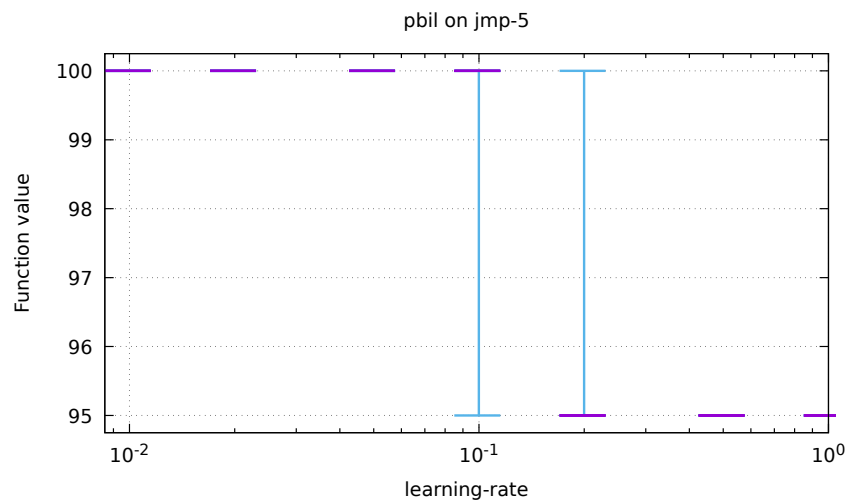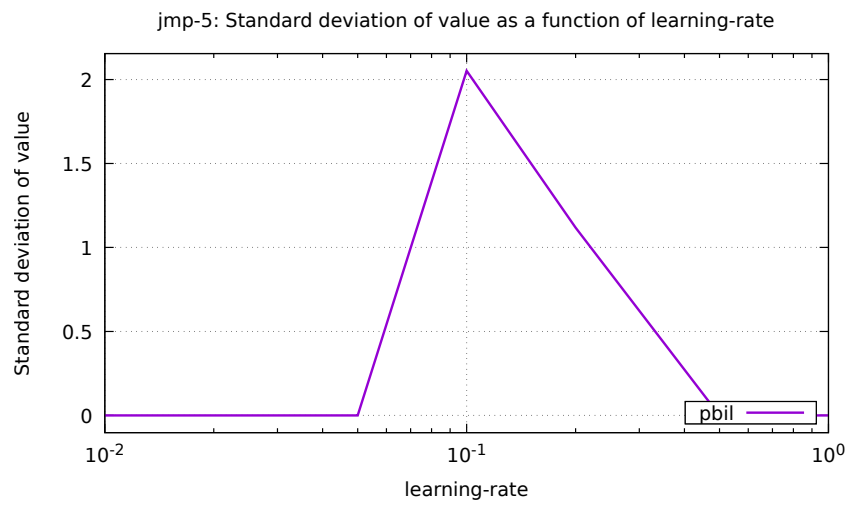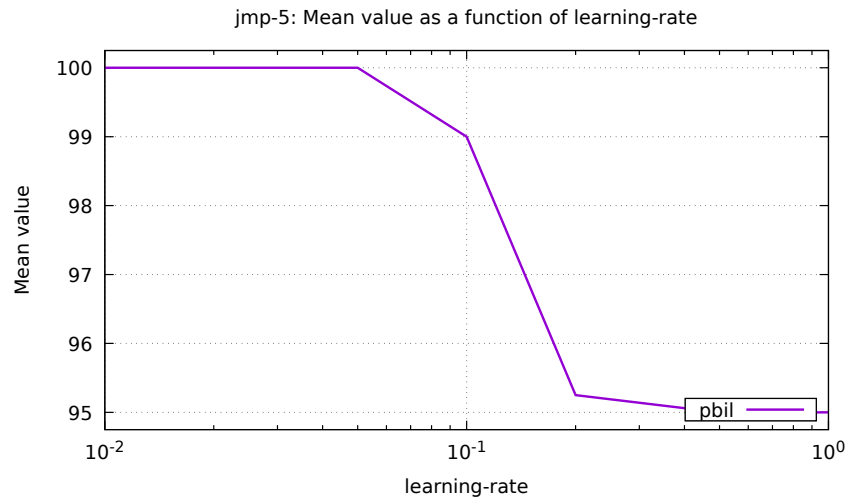
5   Function nk

algorithm	learning-rate	function value					
		min	Q_1	med.	Q_3	max	rk
pbil	0.01	0.89	0.90	0.93	0.94	0.99	2
pbil	0.02	0.86	0.90	0.93	0.94	0.96	3
pbil	0.05	0.82	0.86	0.88	0.93	0.95	6
pbil	0.1	0.81	0.85	0.87	0.90	0.97	7
pbil	0.2	0.80	0.85	0.88	0.92	0.96	5
pbil	0.5	0.79	0.86	0.90	0.94	0.99	4
pbil	1	0.90	0.93	0.94	0.95	0.98	1
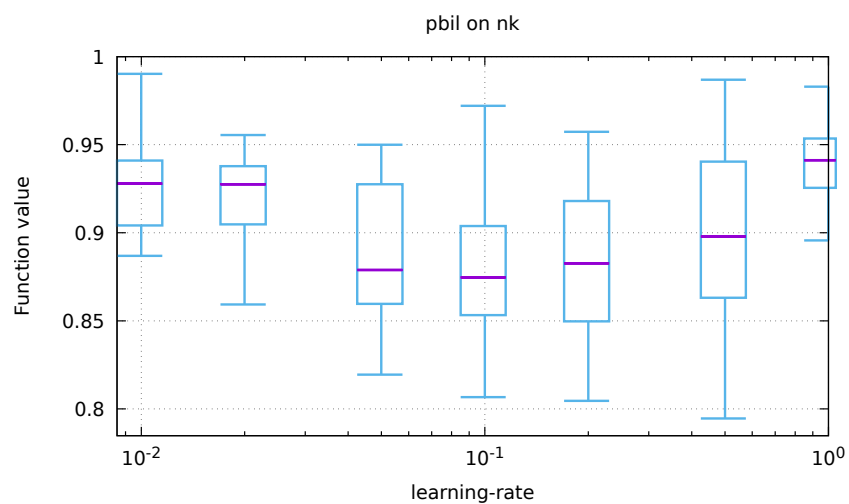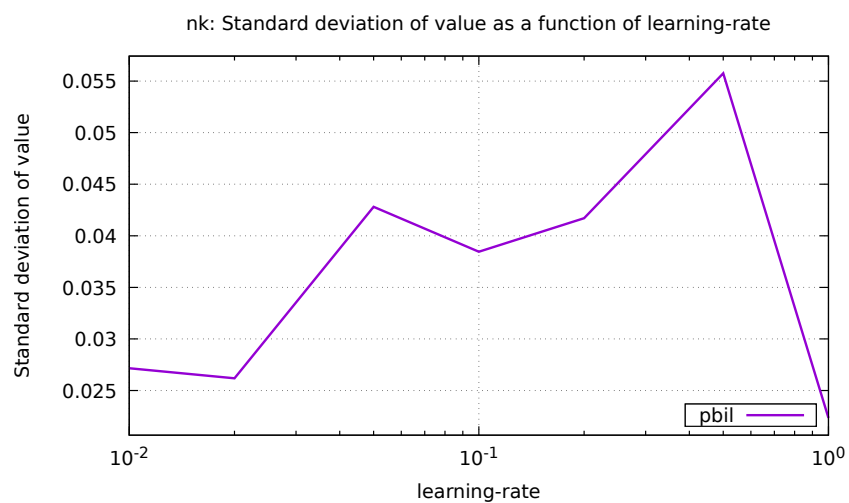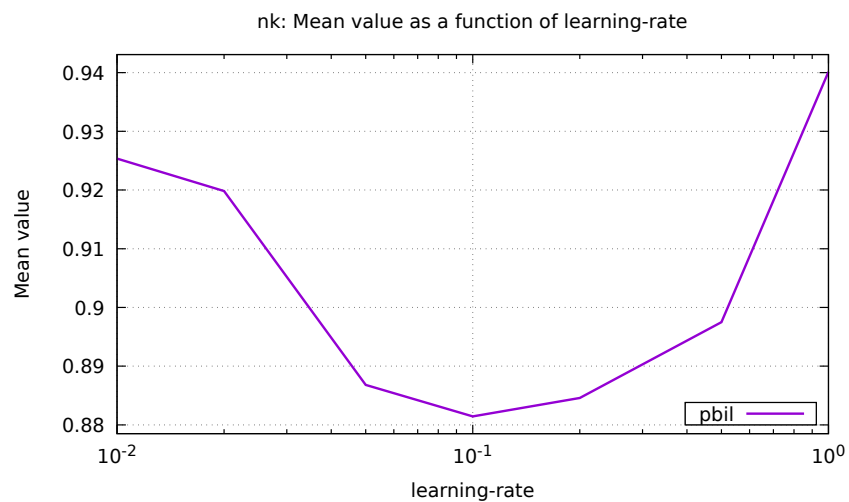
6 Function max-sat

algorithm	learning-rate	function value					
		min	Q_1	med.	Q_3	max	rk
pbil	0.01	963	965	967	967	967	2
pbil	0.02	960	964	965	967	967	5
pbil	0.05	960	963	965	966	967	6
pbil	0.1	959	963	965	967	967	7
pbil	0.2	958	964	966	967	968	3
pbil	0.5	961	964	965	967	968	4
pbil	1	963	966	967	967	968	1

max-sat: Mean value as a function of learning-rate

max-sat: Standard deviation of value as a function of learning-rate

pbil on max-sat

6
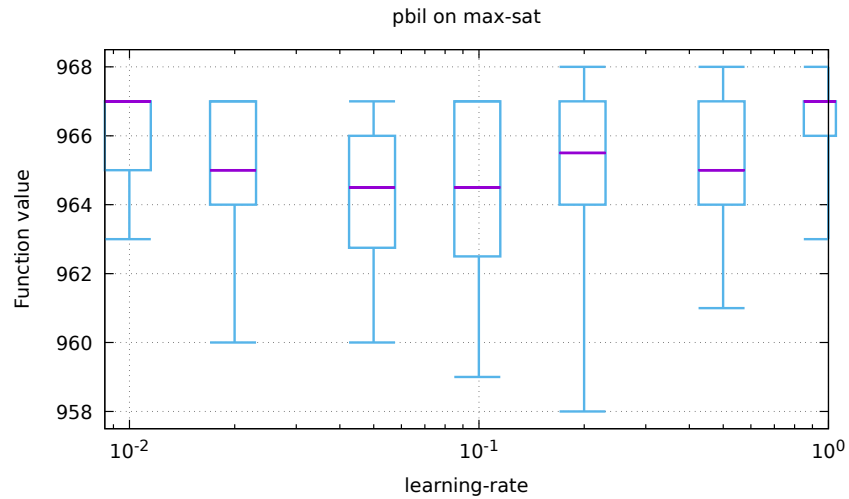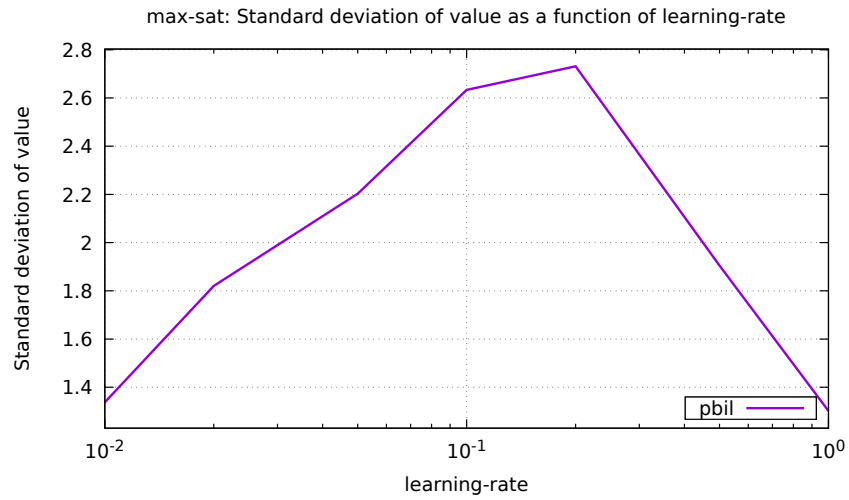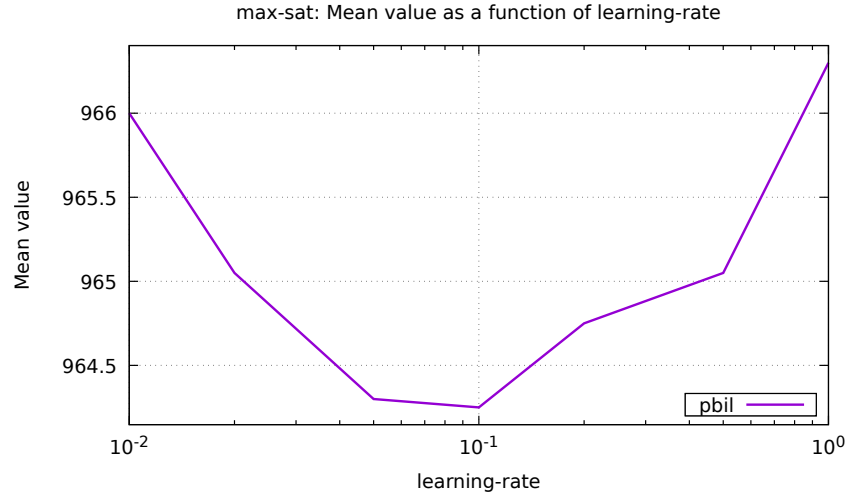
7 Function labs

algorithm	learning-rate	function value					
		min	Q_1	med.	Q_3	max	rk
pbil	0.01	3.40	3.73	3.82	3.97	4.17	7
pbil	0.02	3.32	3.75	3.91	4.11	4.24	4
pbil	0.05	3.29	3.66	3.89	4.01	4.54	6
pbil	0.1	3.39	3.73	3.91	4.10	4.46	5
pbil	0.2	3.48	3.63	3.98	4.15	4.54	3
pbil	0.5	3.79	4.09	4.17	4.40	4.82	2
pbil	1	4.50	4.69	4.80	4.99	5.47	1

labs: Mean value as a function of learning-rate

labs: Standard deviation of value as a function of learning-rate

pbil on labs
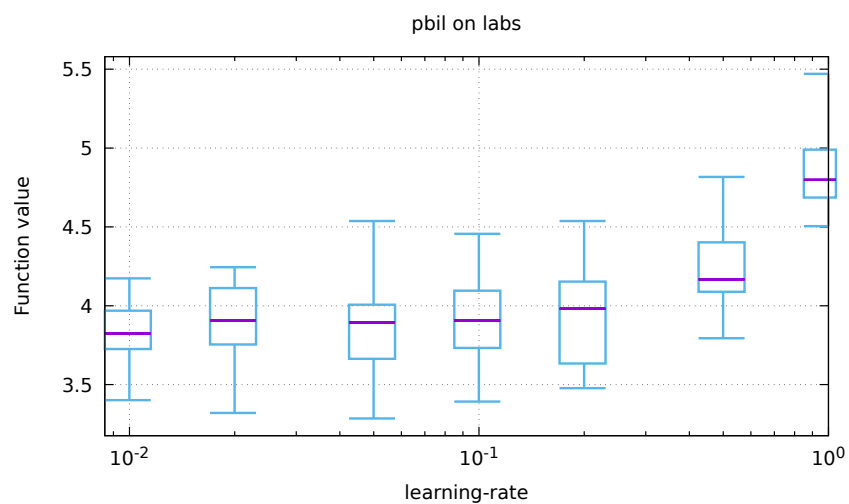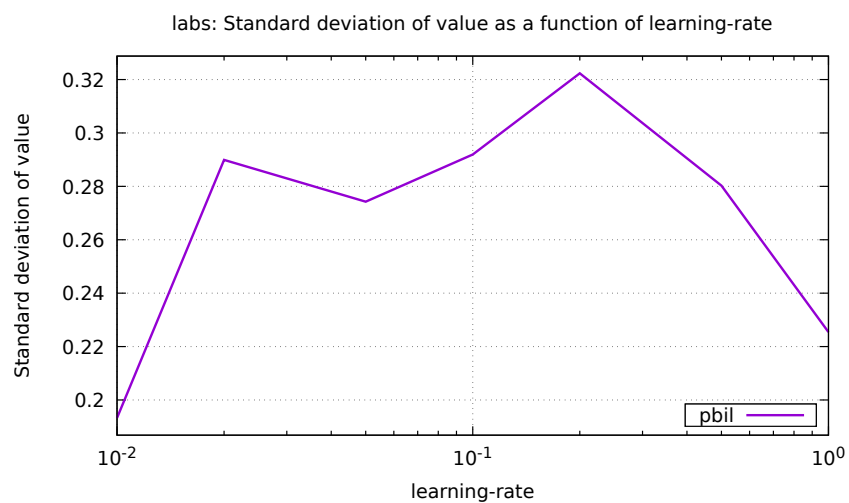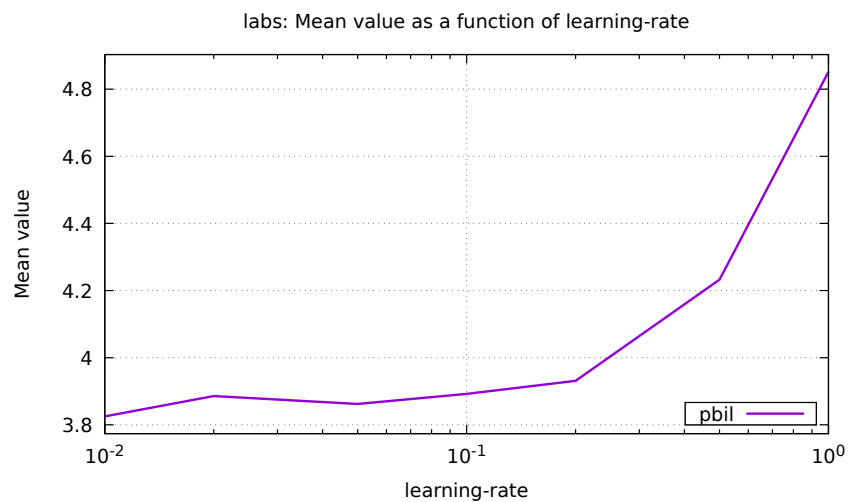
8 Function ep

algorithm	learning-rate	function value					
		min	Q_1	med.	Q_3	max	rk
pbil	0.01	7.5×10^{-28}	3.3×10^{-27}	7.7×10^{-27}	1.2×10^{-26}	2.3×10^{-26}	1
pbil	0.02	2.5×10^{-27}	7.8×10^{-27}	1.4×10^{-26}	3.4×10^{-26}	5.9×10^{-26}	4
pbil	0.05	8.2×10^{-27}	1.8×10^{-26}	2.8×10^{-26}	4.6×10^{-26}	1.3×10^{-25}	5
pbil	0.1	3.0×10^{-27}	1.3×10^{-26}	3.4×10^{-26}	6.4×10^{-26}	1.3×10^{-25}	7
pbil	0.2	6.5×10^{-27}	1.2×10^{-26}	3.4×10^{-26}	5.5×10^{-26}	1.7×10^{-25}	6
pbil	0.5	3.5×10^{-28}	3.7×10^{-27}	8.2×10^{-27}	2.4×10^{-26}	4.0×10^{-26}	3
pbil	1	1.1×10^{-28}	2.9×10^{-27}	8.0×10^{-27}	1.2×10^{-26}	2.4×10^{-26}	2

ep: Mean value as a function of learning-rate

ep: Standard deviation of value as a function of learning-rate

pbil on ep
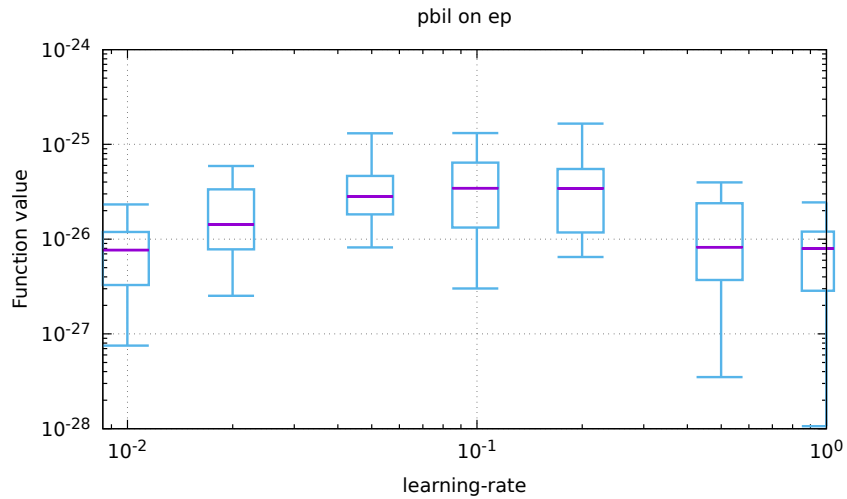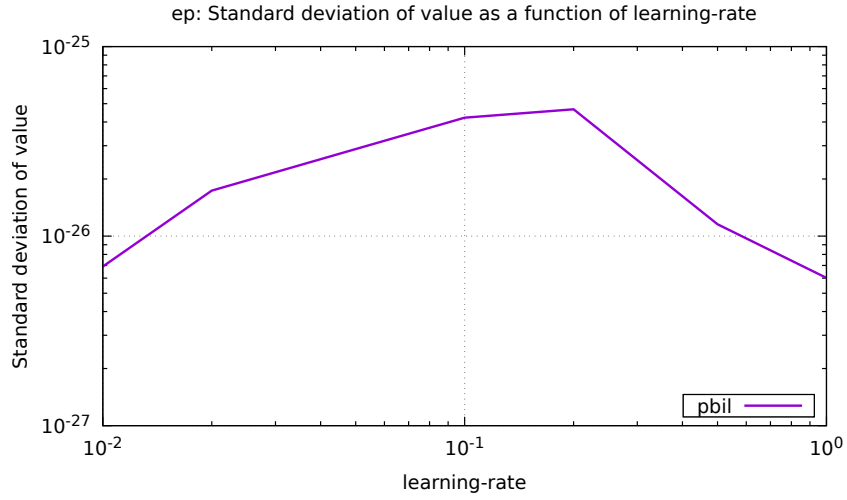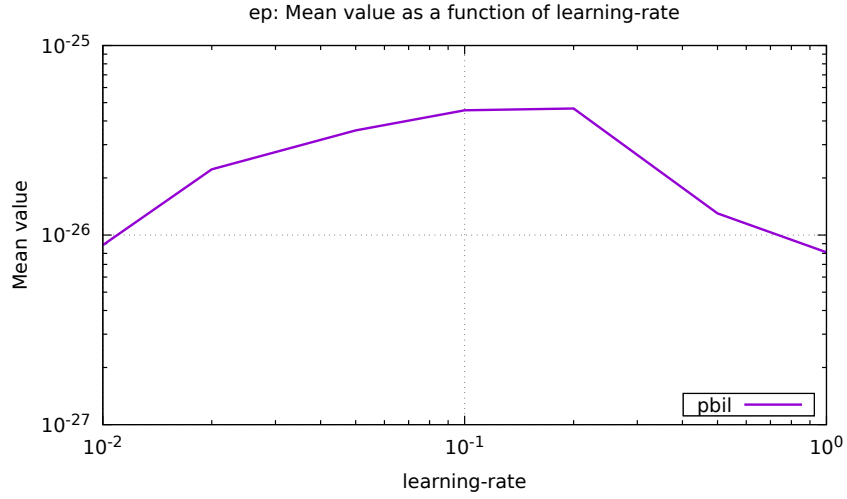
9 Function cancel

algorithm	learning-rate	function value					
		min	Q_1	med.	Q_3	max	rk
pbil	0.01	0.05	0.08	0.20	0.51	1.48	1
pbil	0.02	0.10	0.15	0.31	0.58	1.34	2
pbil	0.05	0.34	1.04	1.48	2.01	2.39	7
pbil	0.1	0.48	0.83	1.20	1.42	3.59	6
pbil	0.2	0.20	0.59	1.03	1.67	2.95	5
pbil	0.5	0.10	0.43	0.79	1.52	2.40	4
pbil	1	0.06	0.14	0.41	1.45	3.25	3

cancel: Mean value as a function of learning-rate

cancel: Standard deviation of value as a function of learning-rate

pbil on cancel
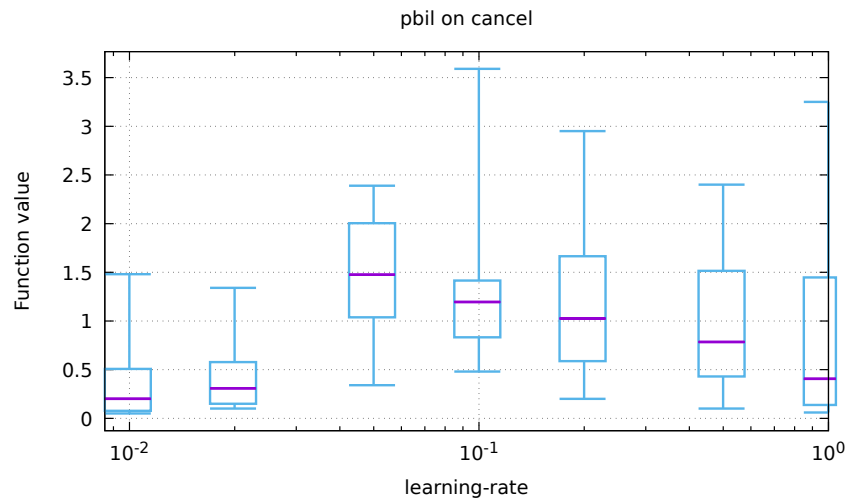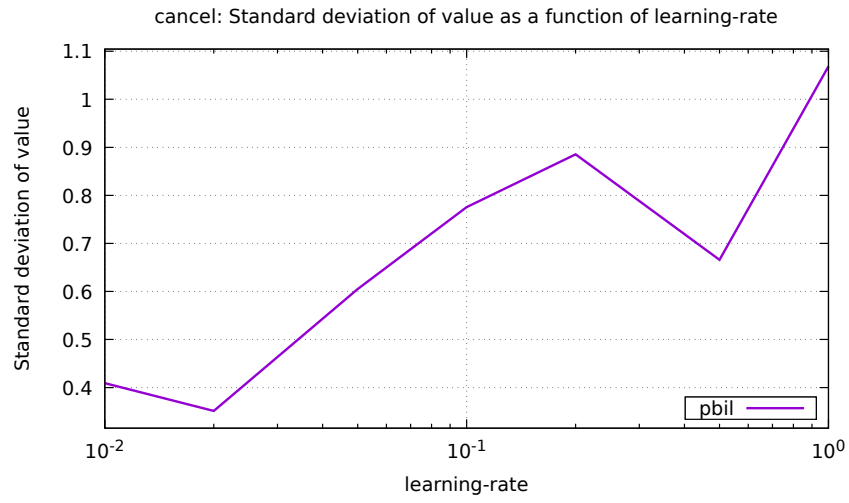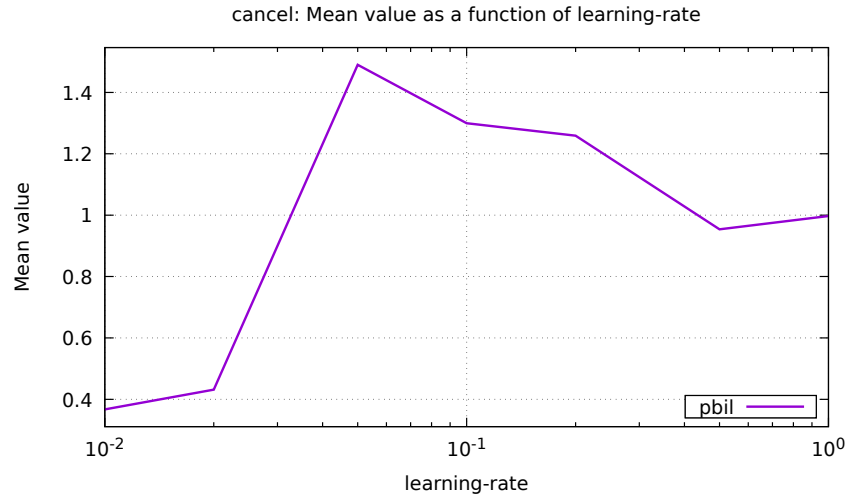
10 Function walsh2

algorithm	learning-rate	function value					
		min	Q_1	med.	Q_3	max	rk
pbil	0.01	661.60	675.90	685.98	712.26	731.39	3
pbil	0.02	599.61	648.90	673.53	701.13	724.80	5
pbil	0.05	608.14	643.46	658.76	682.20	704.60	7
pbil	0.1	606.67	649.19	671.45	689.03	706.98	6
pbil	0.2	600.58	656.55	673.99	697.86	733.57	4
pbil	0.5	624.80	669.68	686.22	711.92	740.55	2
pbil	1	681.90	704.91	722.86	728.62	740.55	1

walsh2: Mean value as a function of learning-rate

walsh2: Standard deviation of value as a function of learning-rate

pbil on walsh2
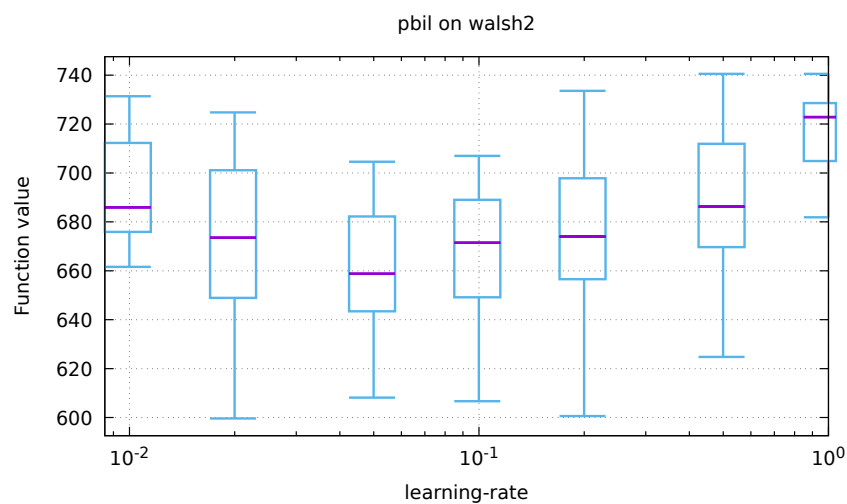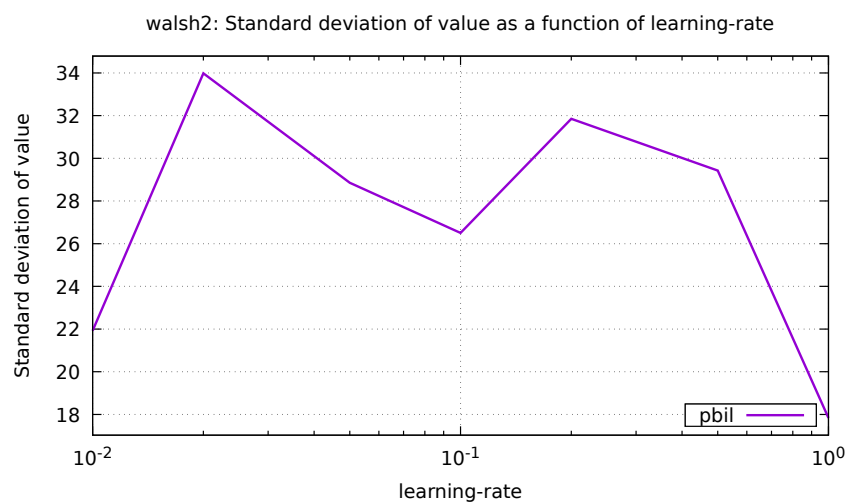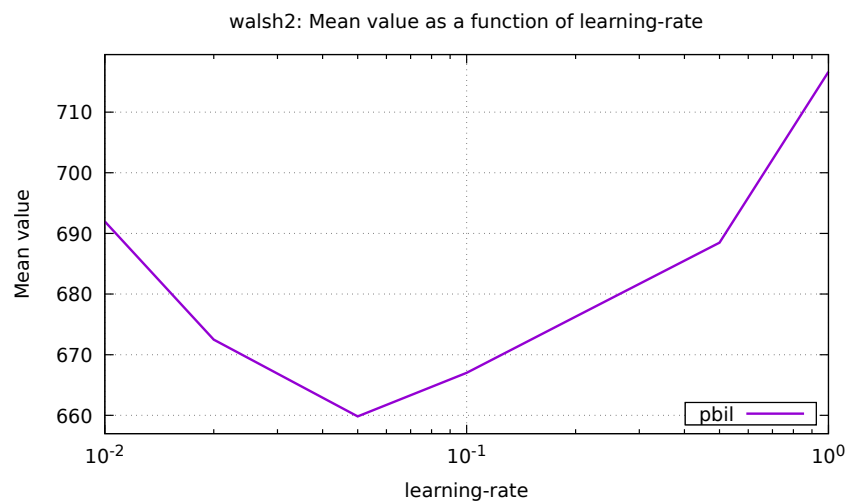
A Plan

```json
{
    "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",
```

```json
            "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
# ea_lambda = 100
# ea_mu = 10
# 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 = nopath
# mutation_probability = 1
# neighborhood = 0
# neighborhood_iterator = 0
```

```
# noise_stddev = 1
# num_iterations = 0
# num_threads = 1
# path = nopath
# pn_mutation_probability = 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
# target = 100
# print_defaults
# last_parameter
# exec_name = hnco
# version = 0.11
# Generated from hnco.json
```