

Benchmarking a Pool-Based Execution with GA and PSO Workers on the BBOB Noiseless Testbed

**García Valdez & Merelo
TNM Mexico - UGR Spain**

“The Network is the Computer”

–John Gage, Sun Microsystems



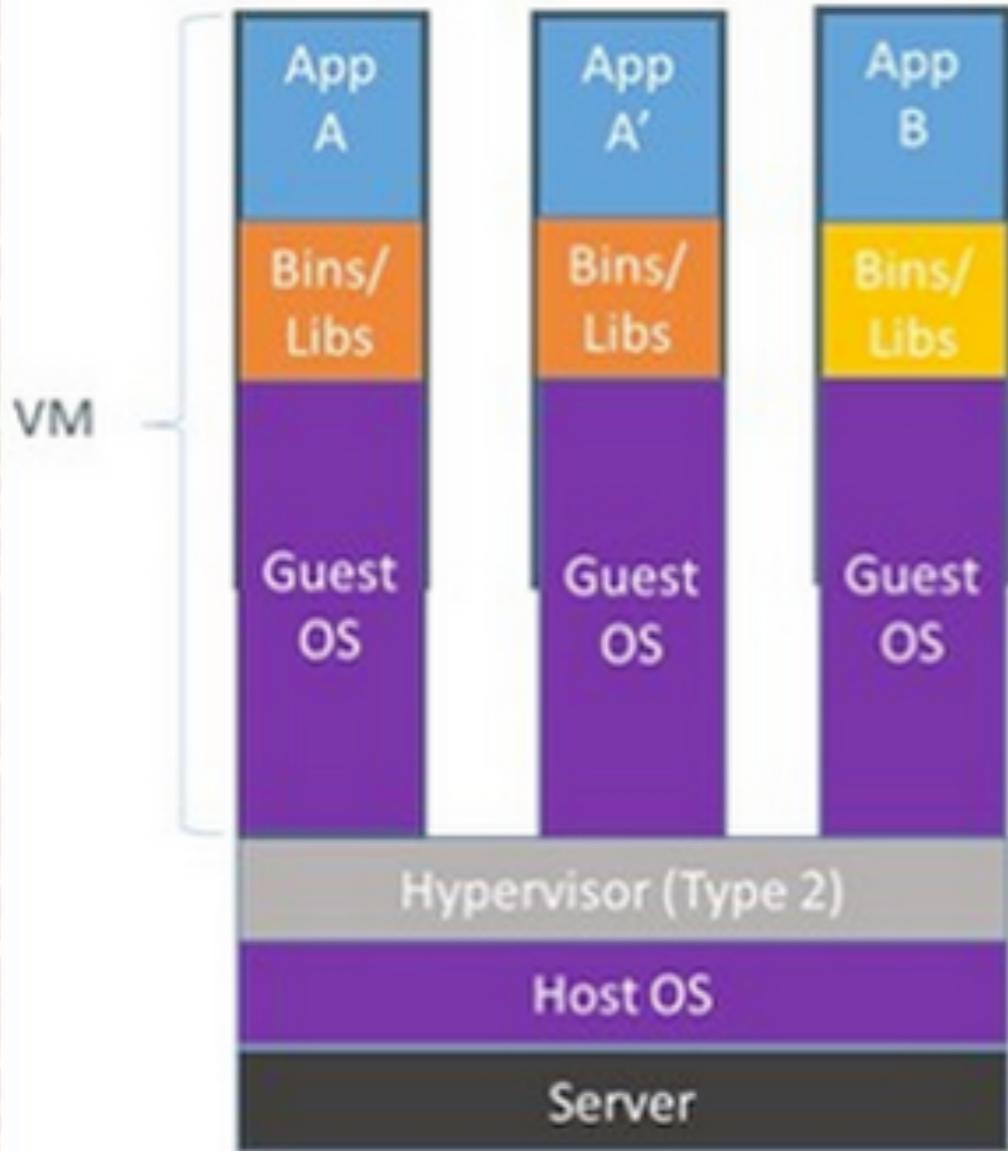
Where do we run EC algorithms?

Available Resources

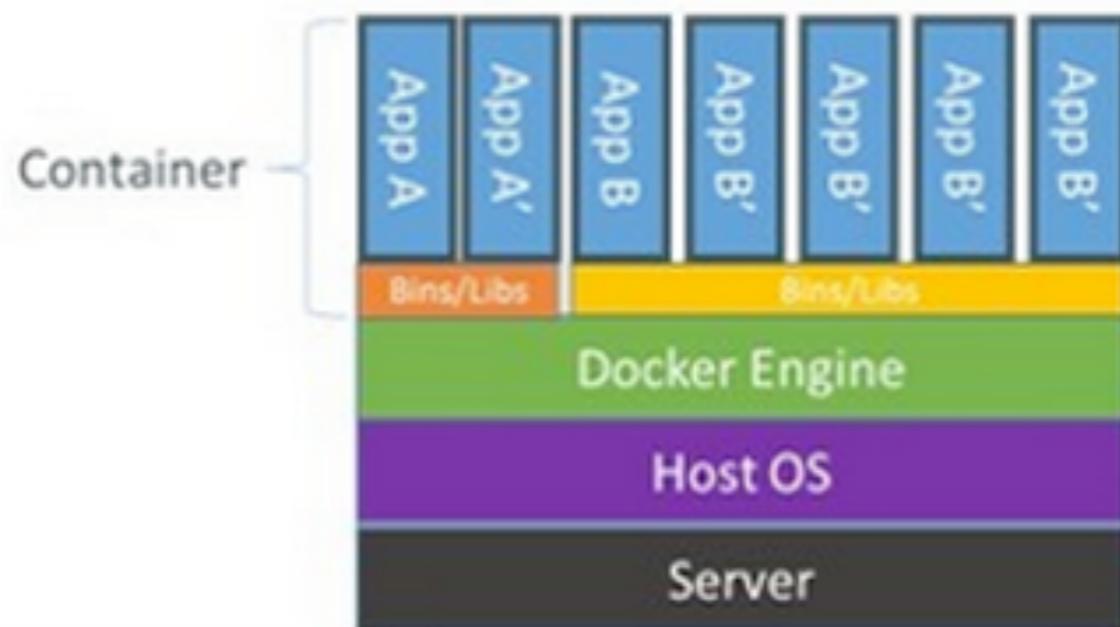


- Laptops
- Desktop
- Grid
- Cloud
- Browsers
- Smart Devices
- Volunteers - Socio Technical System



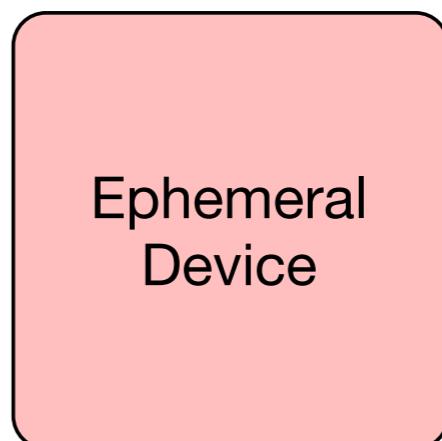
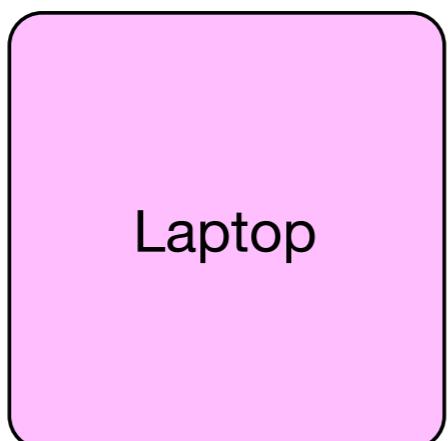
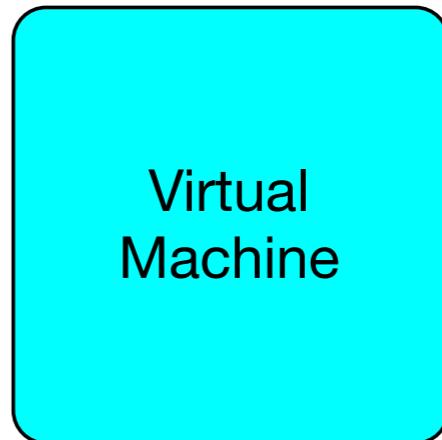
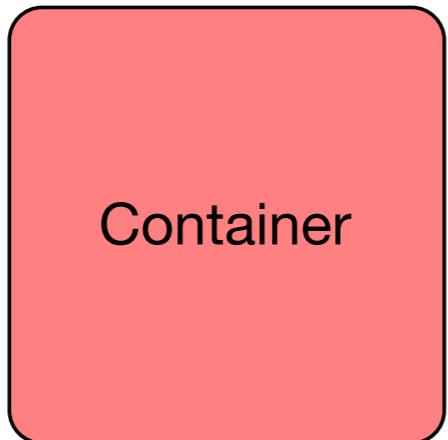


Containers are isolated,
but share OS and, where
appropriate, bins/libraries

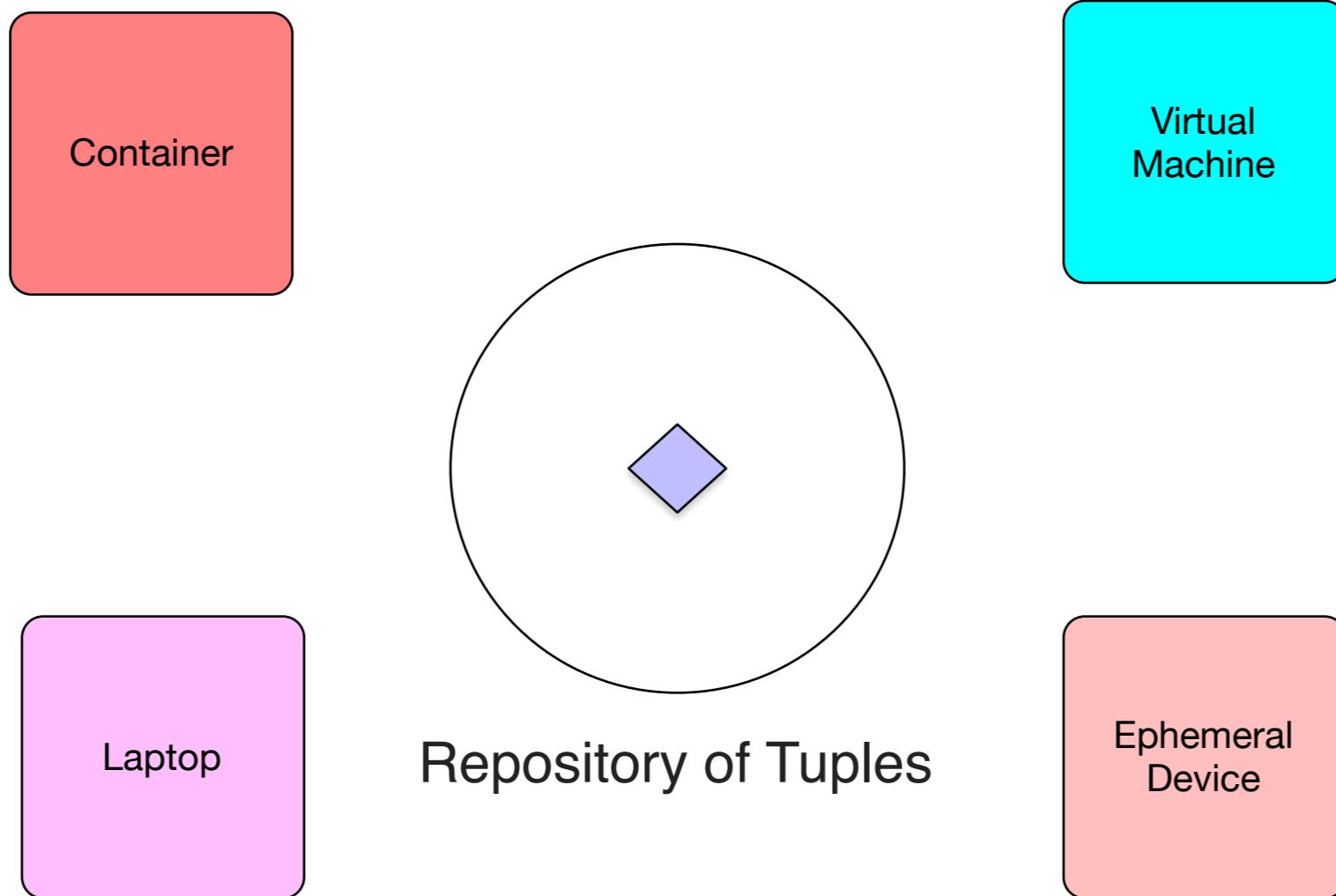


Containers vs Virtual Machines

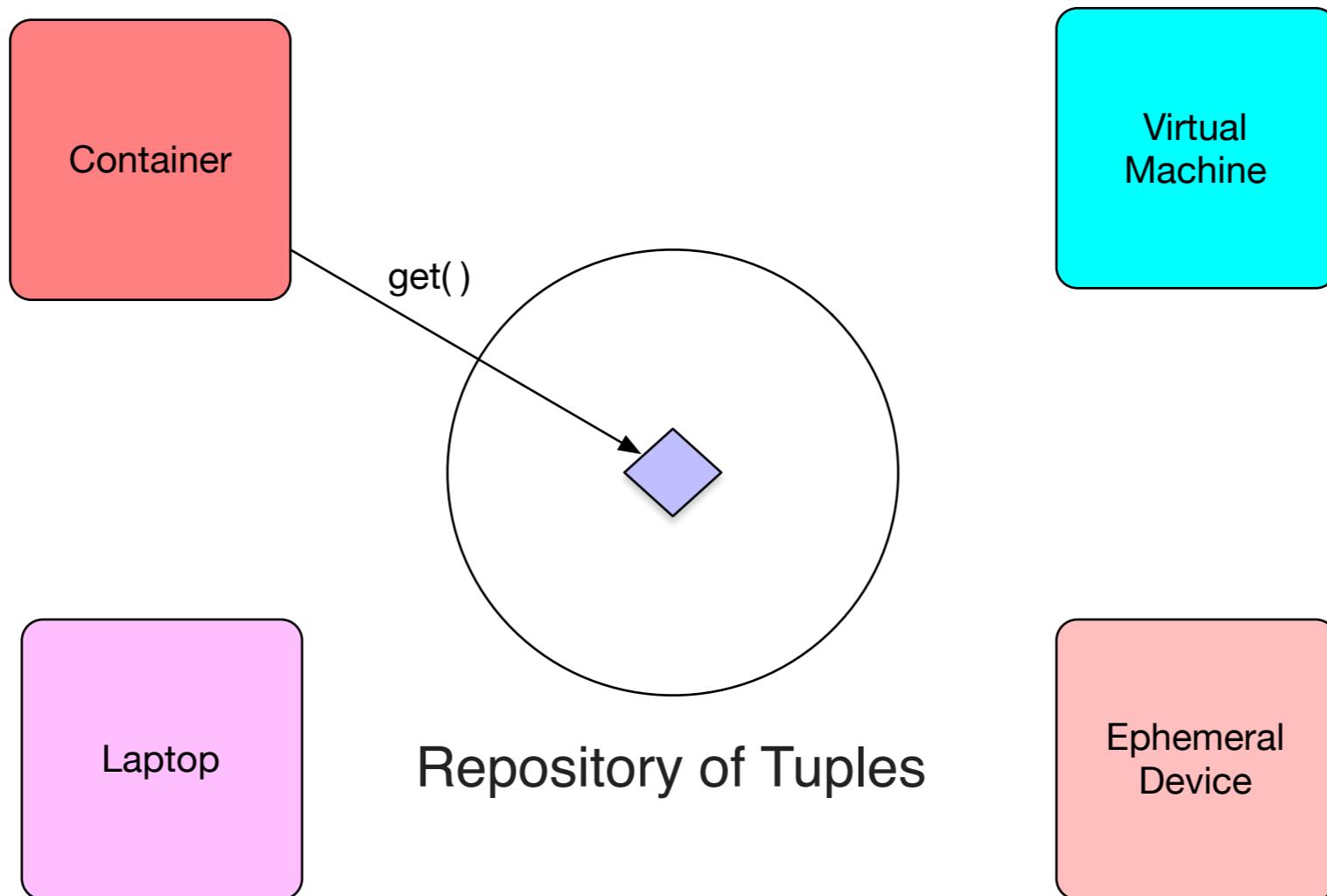
How we can coordinate?



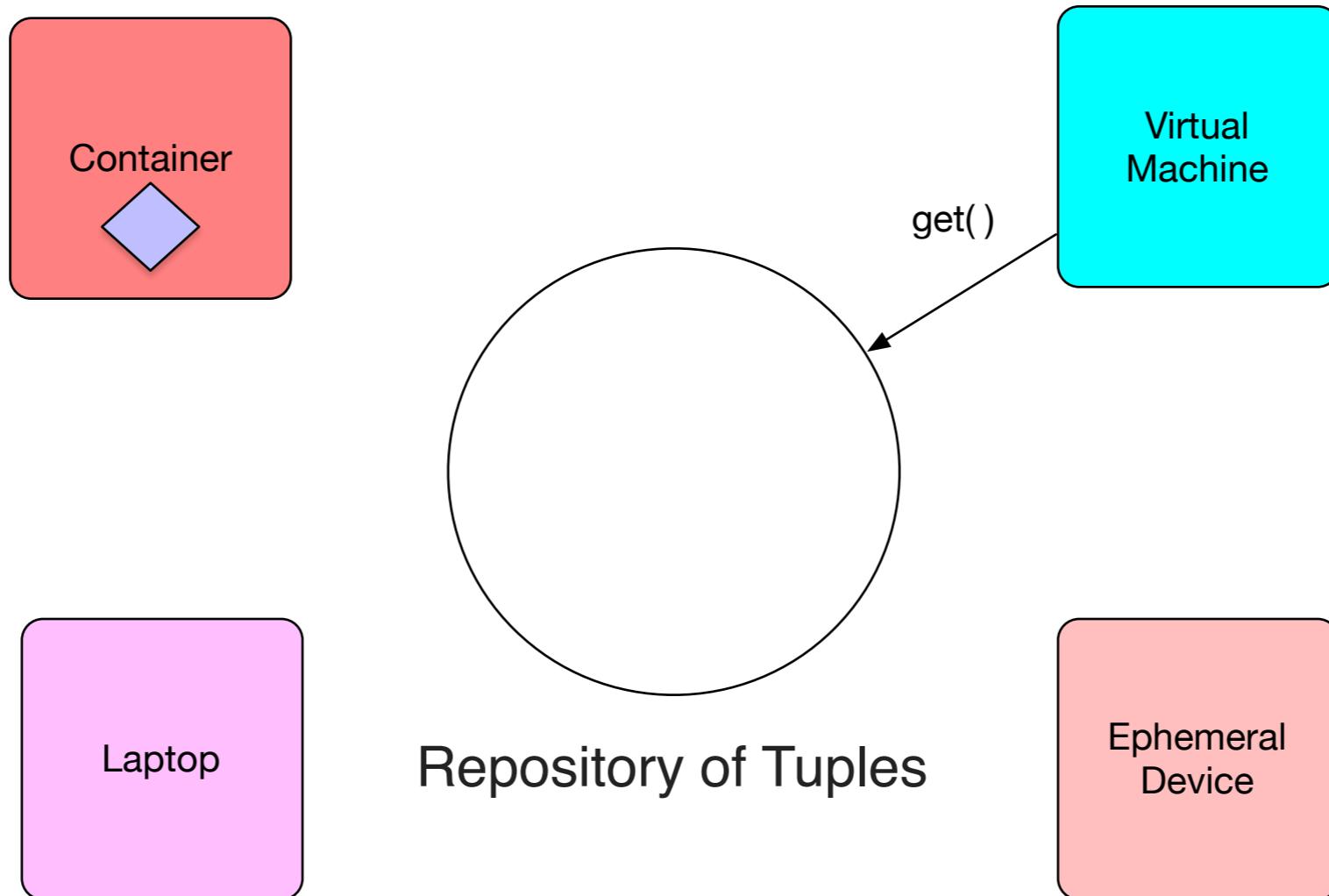
Tuple Space, Pool, Shared Memory.



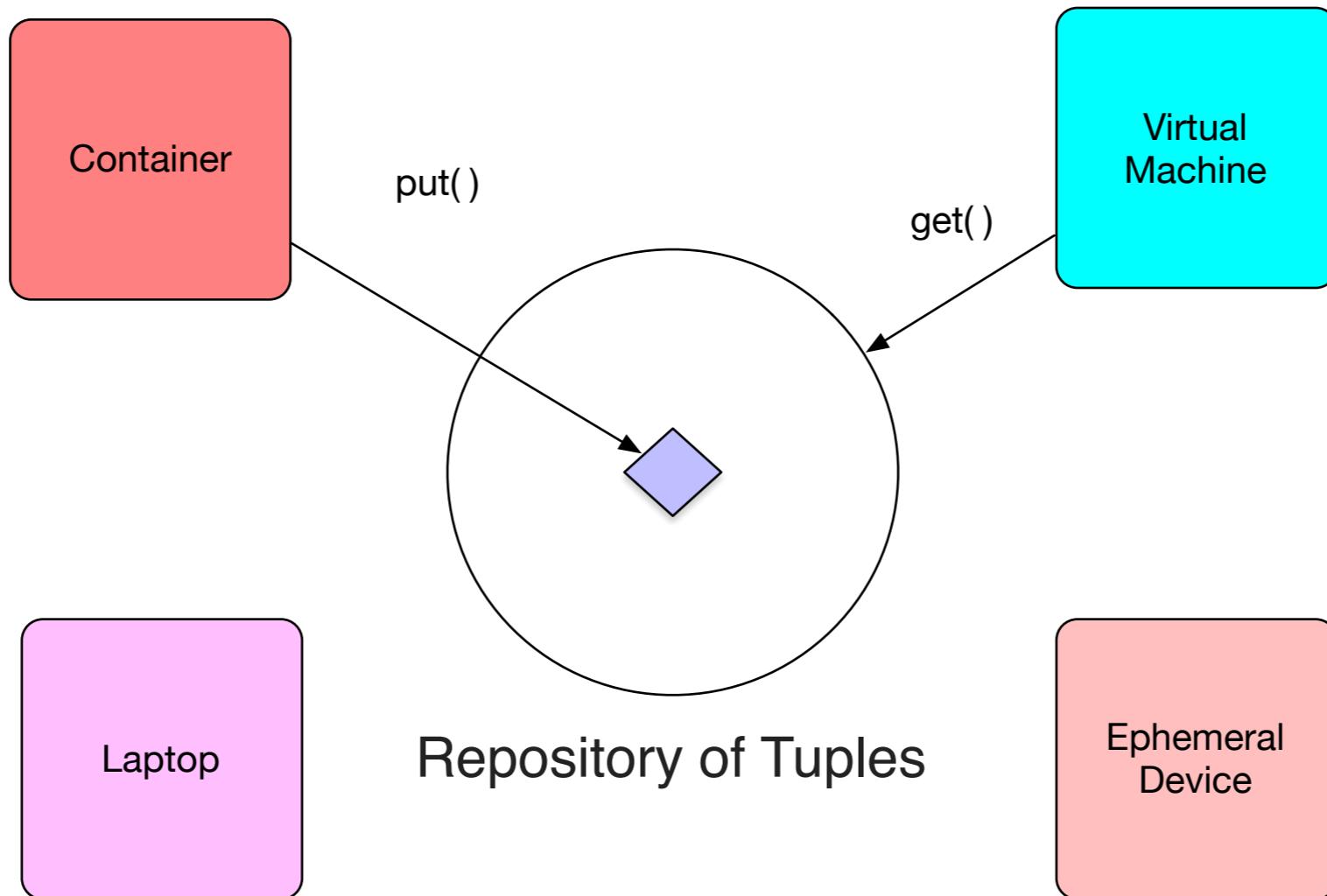
Tuple Space, Pool, Shared Memory.



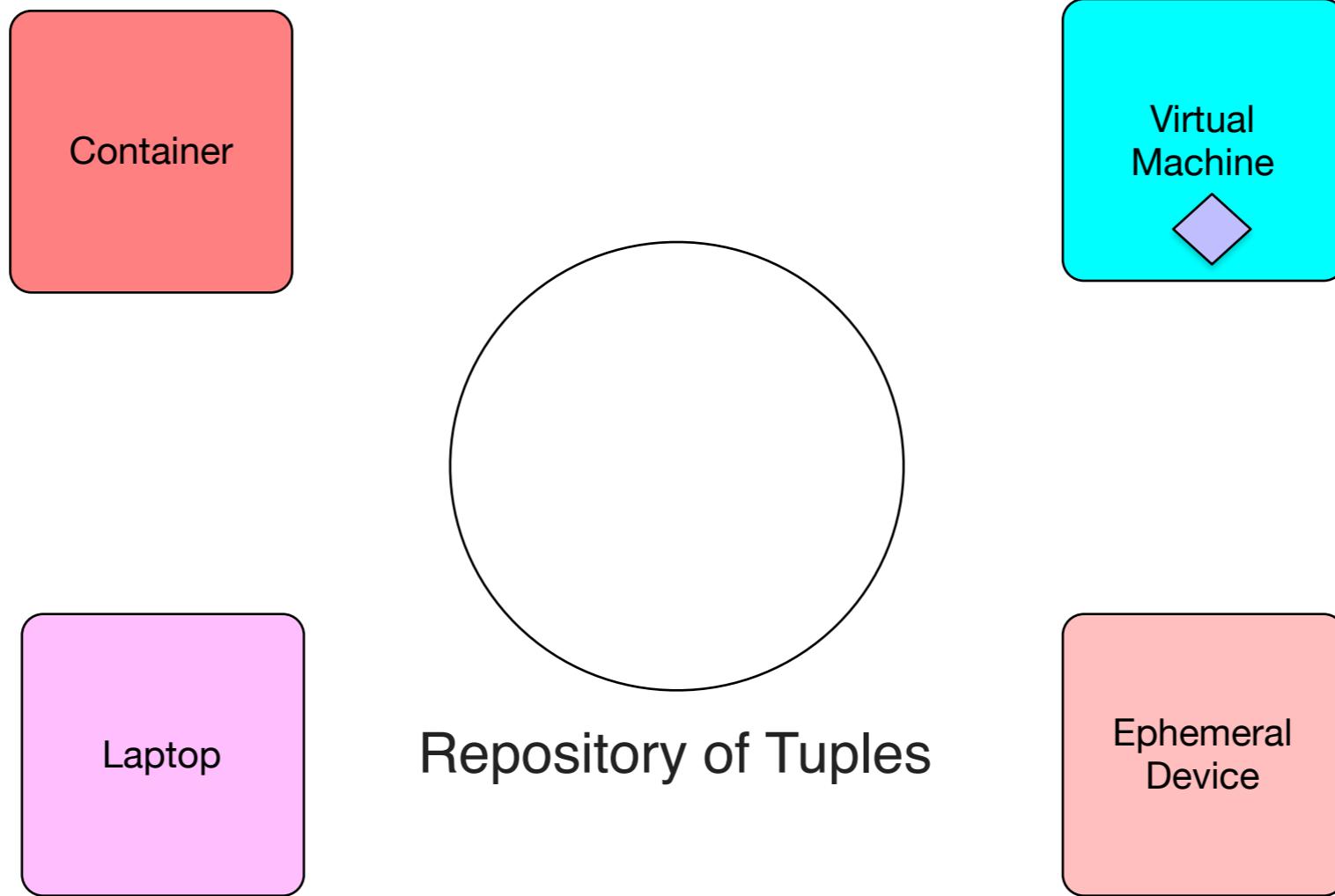
Tuple Space, Pool, Shared Memory.



Tuple Space, Pool, Shared Memory.



Tuple Space, Pool, Shared Memory.



In Pool-Based EAs:

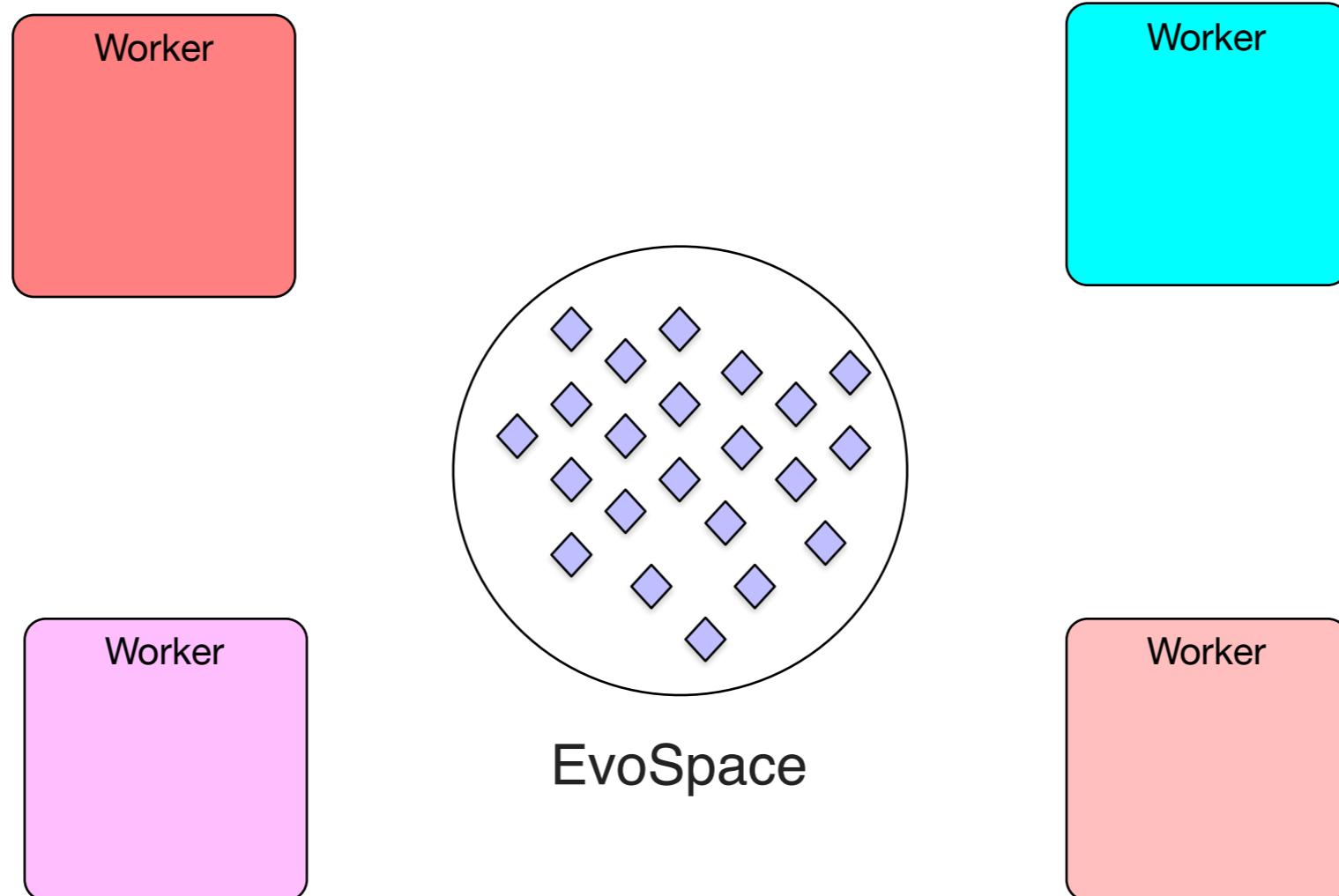
The population repository only receives stateless requests from isolated workers or clients.

An ephemeral collaboration of
computing resources

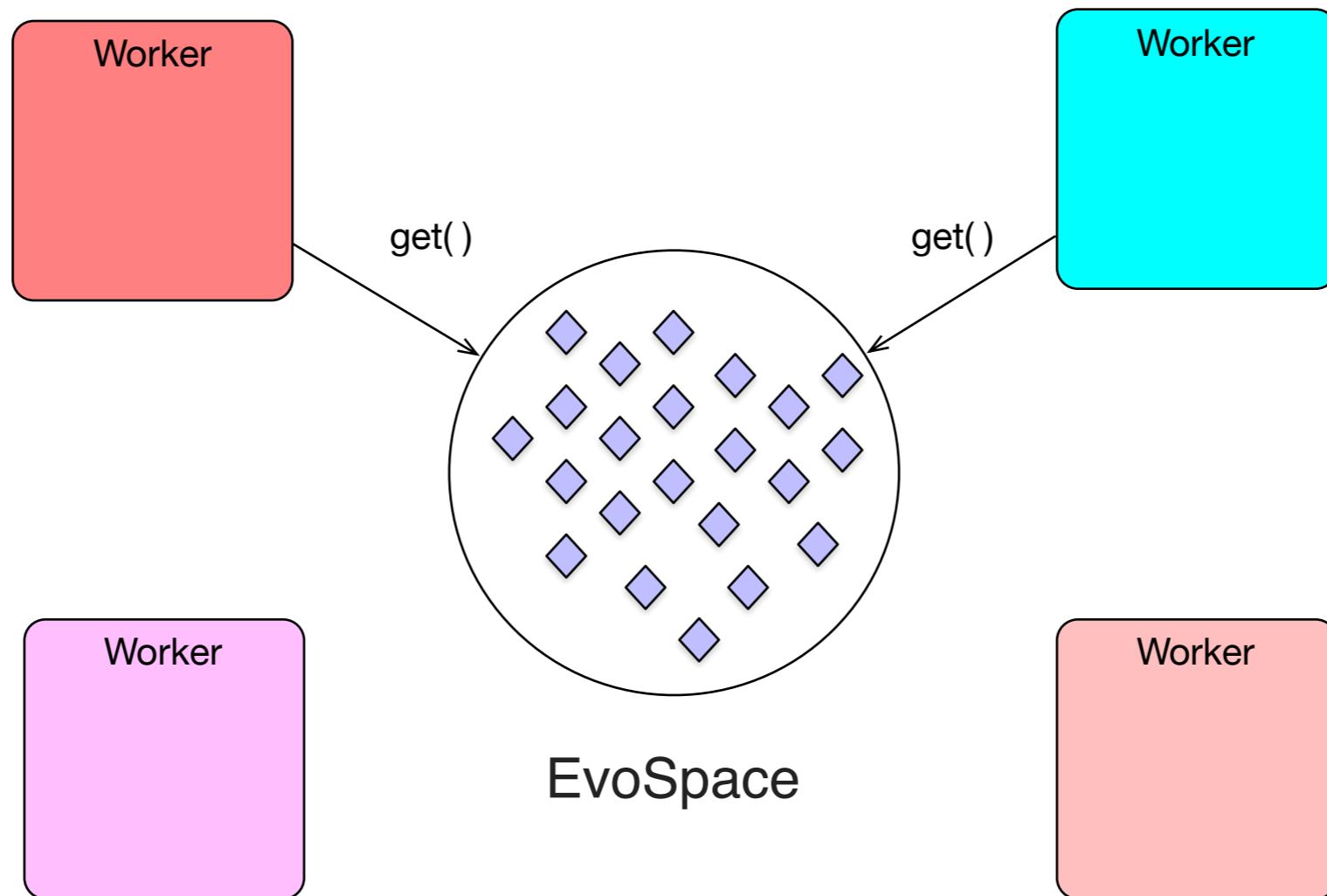
EvoSpace

- 1. The EvoSpace container that stores the evolving population.**
- 2. Remote clients called EvoWorkers, which execute the actual evolutionary process.**

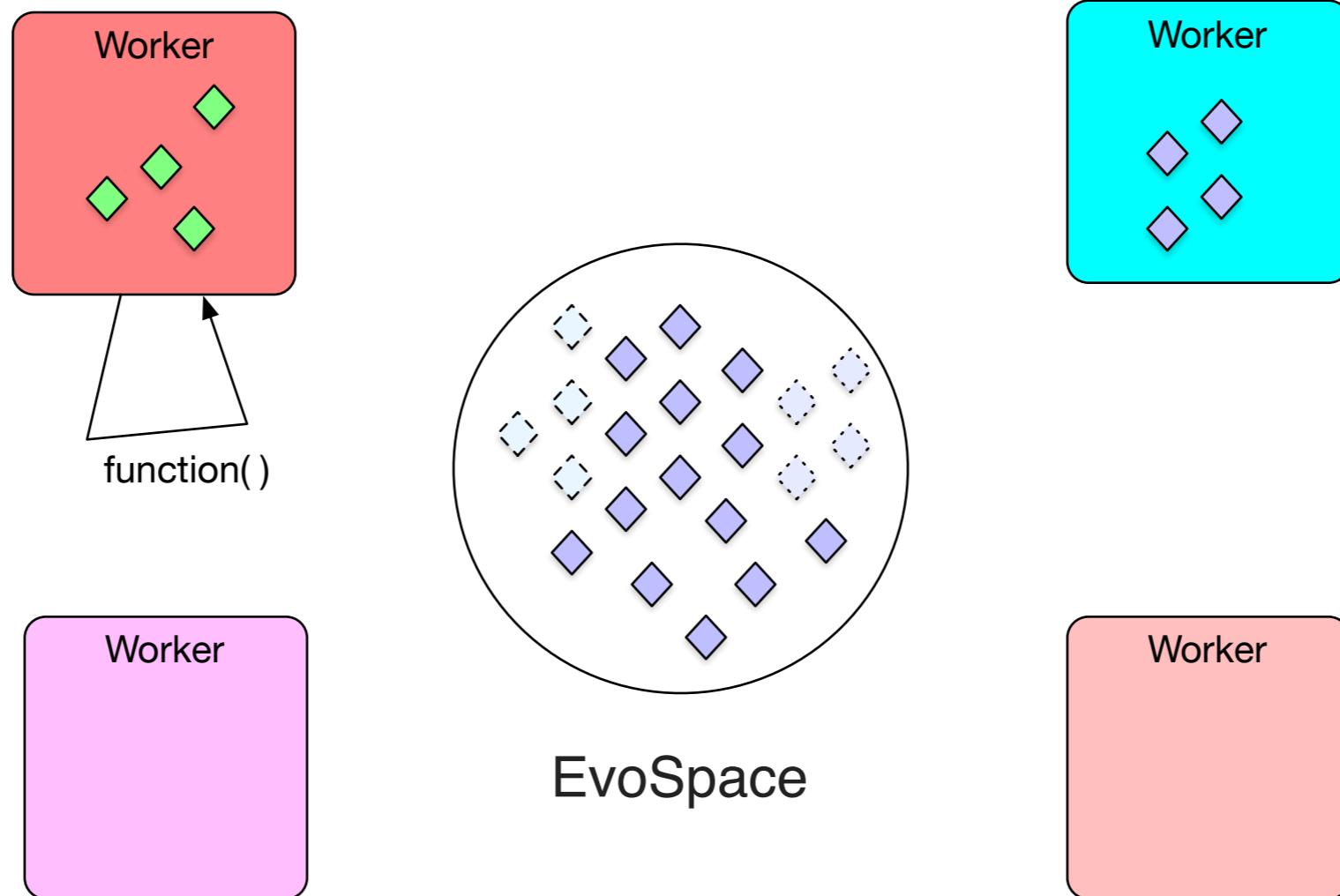
EvoSpace



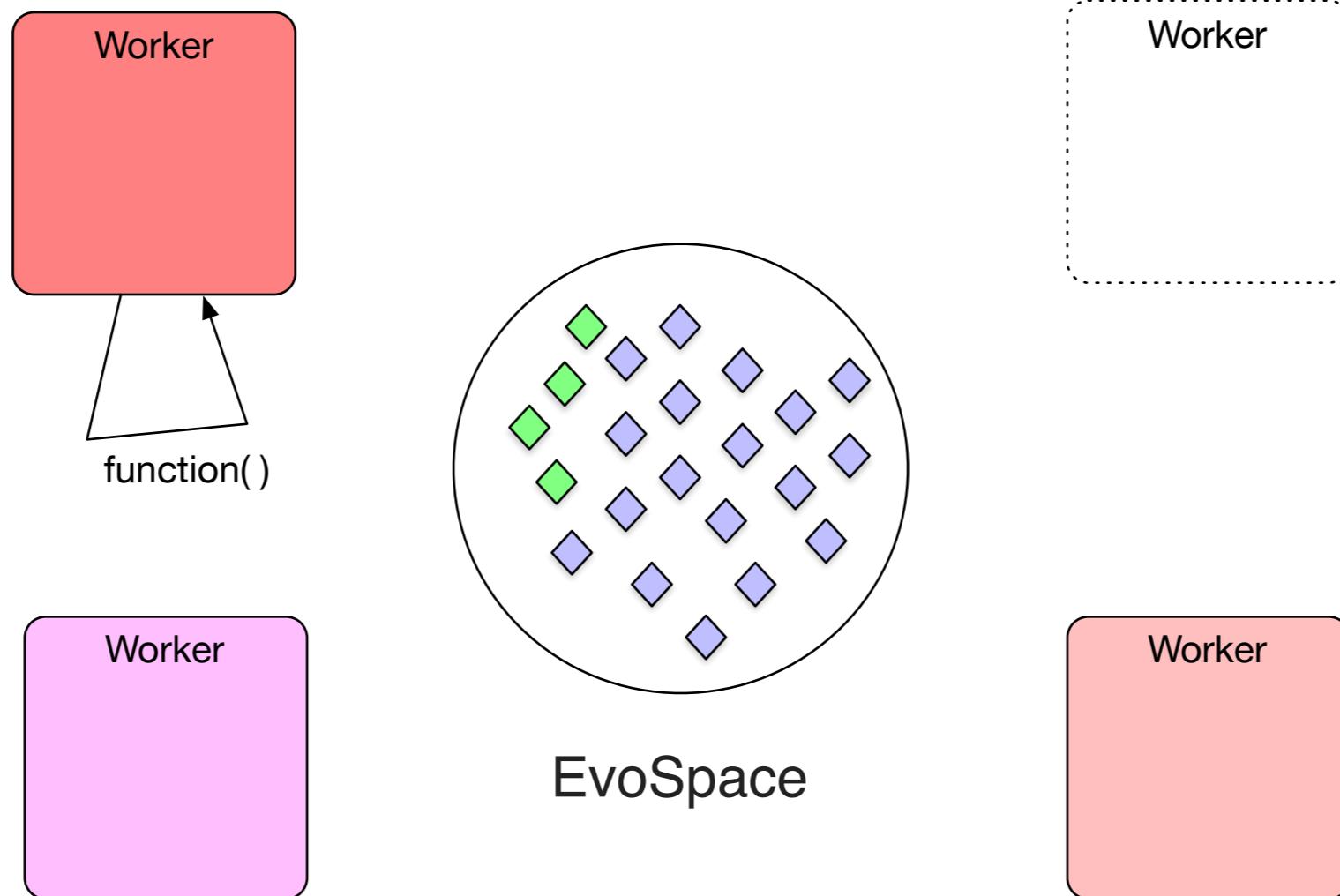
EvoSpace



EvoSpace



EvoSpace



[Show pagesource](#) [Old revisions](#)[Recent changes](#) [Sitemap](#) [Login](#)

Search

Navigation

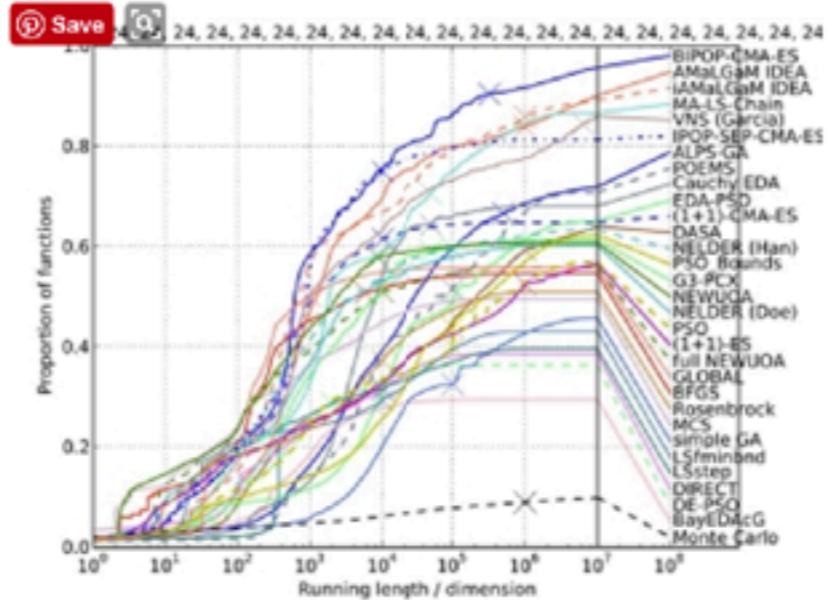
- [Home](#)
- [Documentation](#)
- [download latest old code](#)
- [new code homepage](#)
- [download new code directly](#)
- [BBOB 2017](#)
- [BBOB 2016](#)
- [BBOB 2015 @ GECCO](#)
 - [Algorithms](#)
 - [Results](#)
 - [Schedule](#)
 - [Downloads](#)
- [BBOB 2015 @ CEC](#)
 - [Algorithms](#)
 - [Results](#)
 - [Downloads](#)
- [BBOB 2013](#)
 - [Algorithms](#)
 - [Results](#)
 - [Schedule](#)
 - [Downloads](#)
- [BBOB 2012](#)
 - [Algorithms](#)
 - [Results](#)
 - [Downloads](#)
- [BBOB 2010](#)

COCO (COmparing Continuous Optimisers) is a platform for systematic and sound comparisons of real-parameter global optimisers. COCO provides benchmark function testbeds, experimentation templates which are easy to parallelize, and tools for processing and visualizing data generated by one or several optimizers. The COCO platform has been used for the Black-Box-Optimization-Benchmarking (BBOB) workshops that took place during the GECCO conference in 2009, 2010, 2012, 2013, and in 2015-2017. It was also used at the IEEE Congress on Evolutionary Computation (CEC'2015) in Sendai, Japan. The COCO source code is available at the [downloads](#) page.

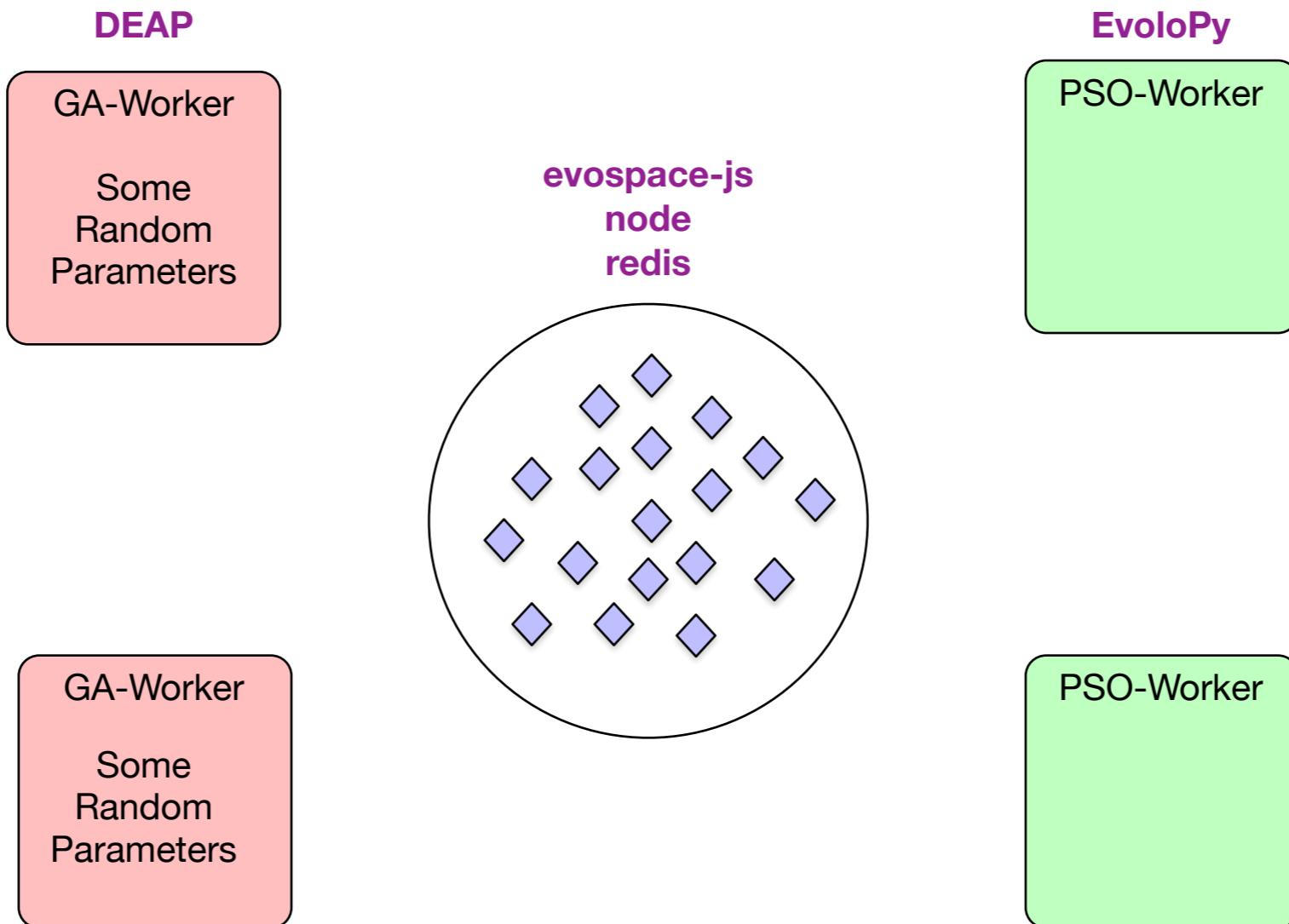
- [Black-Box Optimization Benchmarking \(BBOB\) 2017](#)
- [Black-Box Optimization Benchmarking \(BBOB\) 2016](#)
- [Black-Box Optimization Benchmarking \(BBOB\) 2015](#)
- [CEC'2015 special session on Black-Box Optimization Benchmarking \(CEC-BBOB 2013\)](#)
- [Black-Box Optimization Benchmarking \(BBOB\) 2013](#)
- [Black-Box Optimization Benchmarking \(BBOB\) 2012](#)
- [Black-Box Optimization Benchmarking \(BBOB\) 2010](#)
- [Black-Box Optimization Benchmarking \(BBOB\) 2009](#)
- [Downloads and documentations](#)

To subscribe to (or unsubscribe from) the bbob discussion mailing list follow this link <http://lists.lri.fr/cgi-bin/mailman/listinfo/bbob-discuss>.

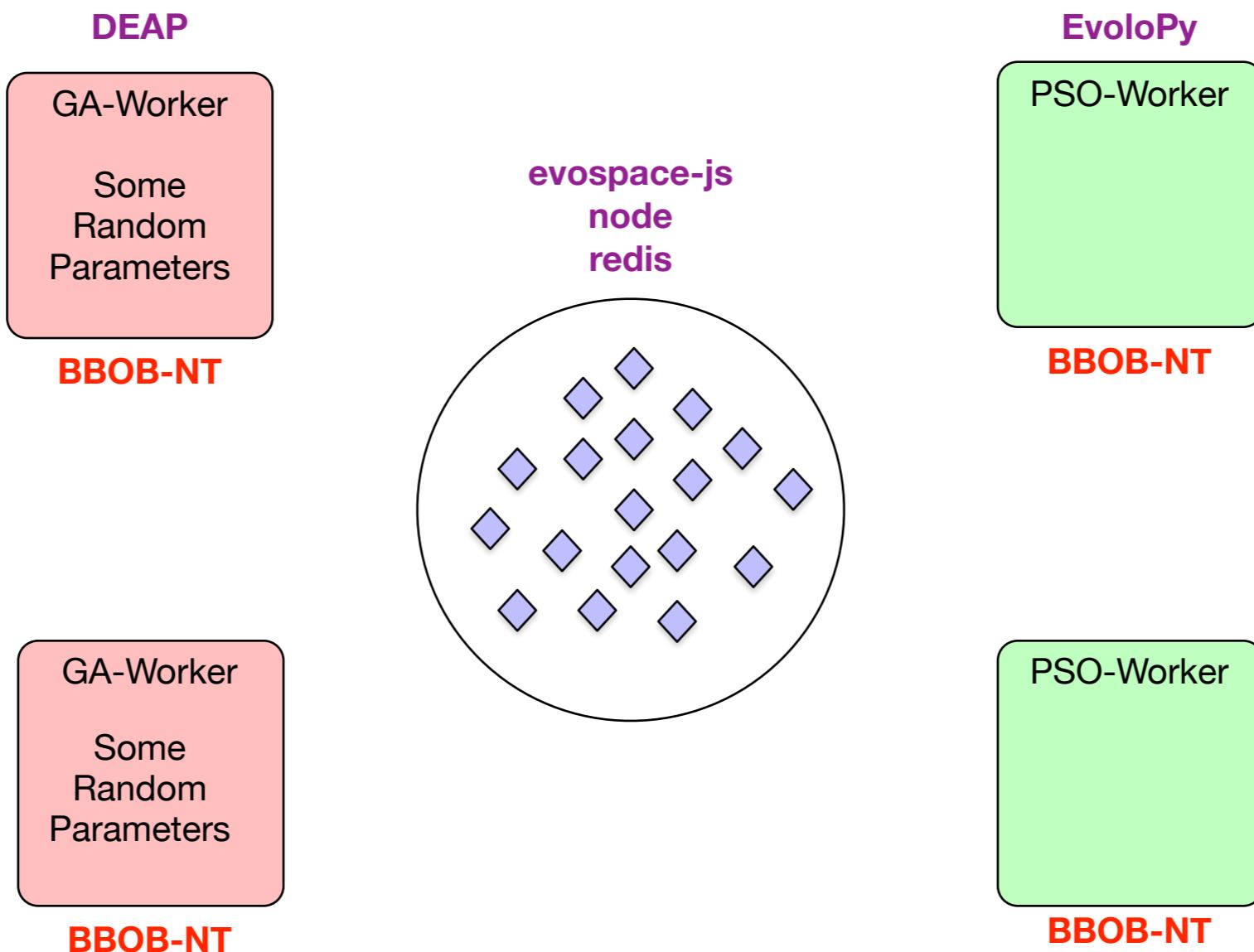
To receive announcements related to the BBOB workshops simply send an email to BBOB team bbob_at_lri.fr with the title "register to BBOB announcement list".

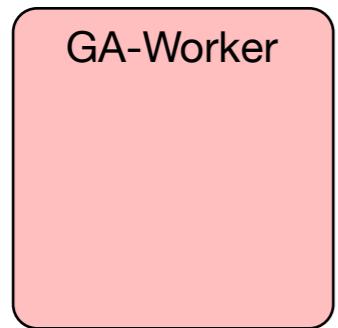


BBOB Noiseless Testbed

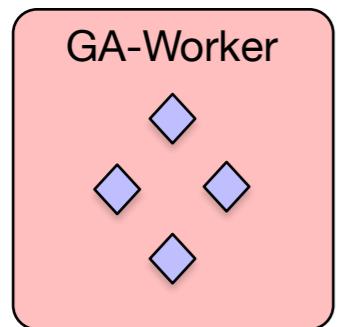


First Problem: BOBB-NT in each Worker?

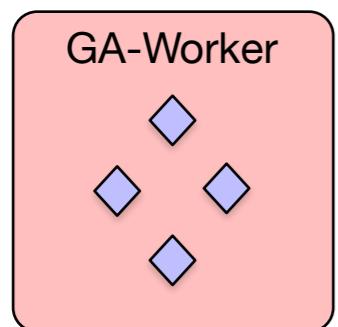




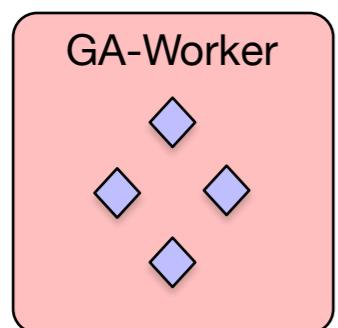
get(4)



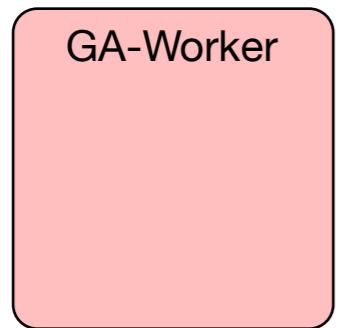
Initialize()
Evaluations: 4



Gen 1
Evaluations: 4



Gen 2
Evaluations: 4



put()

In each put()

GA-Worker

```
put()

[{"evals":  
    [[0, 79.58030098999673, [0.011691798917801144, -0.9514519409206983], 190],  
     [1, 79.5296109817965, [0.16652627034343048, -0.9514519409206983], 96],  
     [2, 79.48747195205547, [0.16652627034343048, -1.1621661556642835], 97],  
     [3, 79.48747195205547, [0.16652627034343048, -1.1621661556642835], 96],  
     [4, 79.48010150324886, [0.24427312353549896, -1.1621661556642835], 94],  
     [10, 79.48010150324886, [0.24427312353549896, -1.1621661556642835], 96],  
     [48, 79.48009316445598, [0.24427312353549896, -1.1522770768590862], 96],  
     [49, 79.48009316445598, [0.24427312353549896, -1.1522770768590862], 94]],  
    "dim": 2, "fopt": 79.48, "algorithm": "GA", "benchmark": 1,  
    "instance": 1, "worker_id": "b234ad9e-30cc-11e7-89ff-0242ac110003",  
    "params": {"selection": "tools.selTournament, tournsize=12",  
               "MUTPB": 0.4736001879432521,  
               "NGEN": 50,  
               "crossover": "cxTwoPoint",  
               "sample_size": 100,  
               "init": "random:[-5,5]",  
               "mutation": "mutGaussian, mu=0, sigma=0.5, indpb=0.05",  
               "CXPB": 0.9181993078352011},  
    "experiment_id": 160}]}
```

Post Processing

- Creates Folder Structure
- Runs the BOBB logging function
- Creates a random solution at the beginning
- Creates .dat and .tdat files

docker_exp.py

```
DATA_ROOT = './experiment_data'
DATA_FOLDER = './experiment_data/' + str(EXPERIMENT_ID)

es_conf = {
    2: {'EVOSPACE_SIZE': 250, 'NGEN': 50, 'SAMPLE_SIZE': 100, 'MAX_SAMPLES': 20, 'PSO': 1, 'GA': 1 },
    3: {'EVOSPACE_SIZE': 250, 'NGEN': 50, 'SAMPLE_SIZE': 100, 'MAX_SAMPLES': 30, 'PSO': 1, 'GA': 1 },
    5: {'EVOSPACE_SIZE': 500, 'NGEN': 50, 'SAMPLE_SIZE': 100, 'MAX_SAMPLES': 25, 'PSO': 2, 'GA': 2 },
    10: {'EVOSPACE_SIZE': 1000, 'NGEN': 50, 'SAMPLE_SIZE': 200, 'MAX_SAMPLES': 25, 'PSO': 2, 'GA': 2 },
    20: {'EVOSPACE_SIZE': 2000, 'NGEN': 50, 'SAMPLE_SIZE': 200, 'MAX_SAMPLES': 25, 'PSO': 4, 'GA': 4 },
    40: {'EVOSPACE_SIZE': 4000, 'NGEN': 50, 'SAMPLE_SIZE': 200, 'MAX_SAMPLES': 25, 'PSO': 8, 'GA': 8 },
}

for function in (1,):
    for dim in (2,3):
        print function,"DIM", dim
        print "instance:",
        for instance in range(1,6)+range(41, 51):
            env = {'FUNCTION': function, 'DIM': dim, 'INSTANCE': instance, 'FEmax': 500000,
                   'EVOSPACE_URL': REDIS_HOST + ':3000/evospace', 'POP_NAME': 'test_pop',
                   'UPPER_BOUND': 5, 'LOWER_BOUND': -5, 'BENCHMARK': True, 'EXPERIMENT_ID': EXPERIMENT_ID,
                   'NGEN': es_conf[dim]['NGEN'], 'SAMPLE_SIZE': es_conf[dim]['SAMPLE_SIZE'],
                   'MAX_SAMPLES': es_conf[dim]['MAX_SAMPLES'], 'BENCHMARK': True}

            info = {'workers': {'GA': {'Number': 1, 'envs': [env]}, 'PSO': {'Number': 1, 'envs': [env]}}}
            evospace.initialize(REDIS_HOST + ':3000/evospace', 'test_pop', env['DIM'], -5, 5, es_conf[dim]['EVOSPACE_SIZE'])

            dw.kill_all()
            dw.remove_all()
            gas = [dw.make_container(env, "python /home/EvoWorker/ga_worker.py %s ") for _ in range(es_conf[dim]['GA'])]
            psos = [dw.make_container(env, "python /home/EvoWorker/pso_worker.py %s ") for _ in range(es_conf[dim]['PSO'])]
            containers = gas+psos

            for c in containers:
                "Starting",c
                c.start()
```

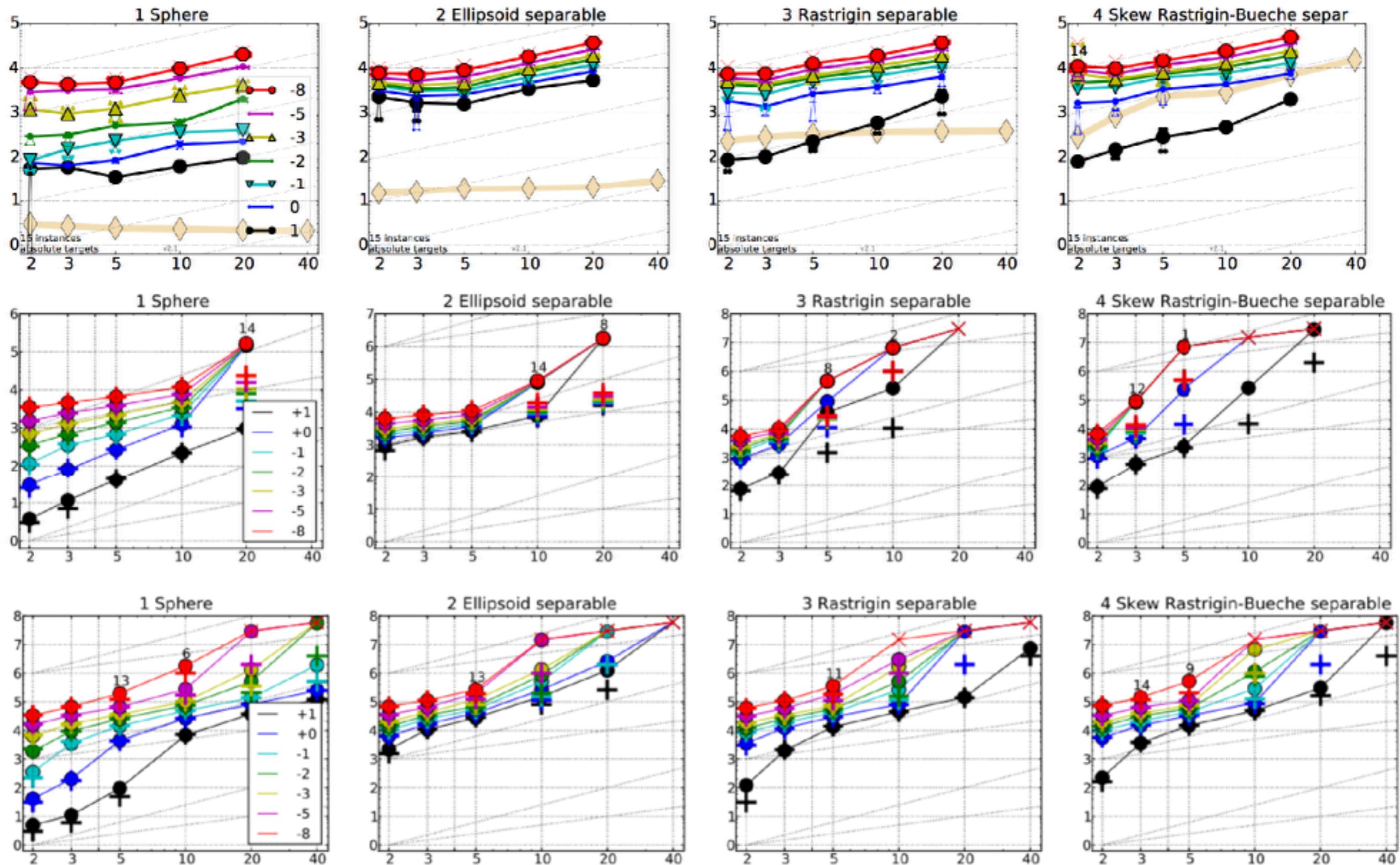
Functions: 1-5 and 41-50 from the 2016 version of the test bed

Maximum: #FE= $10^5 * D$

Table 1: EvoWorker Setup

Dimension	2	3	5	10	20	40
Iterations per Sample	50	50	50	50	50	50
Sample Size	100	100	100	200	200	200
Samples per Worker	20	30	25	25	25	25
PSO Workers	1	1	2	2	4	8
GA Workers	1	1	2	2	4	8

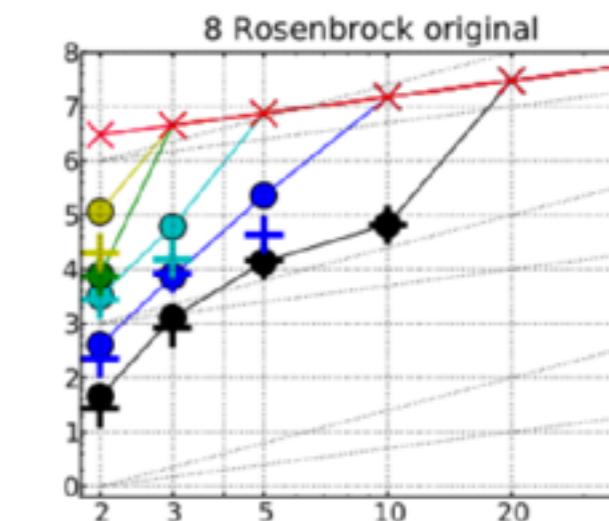
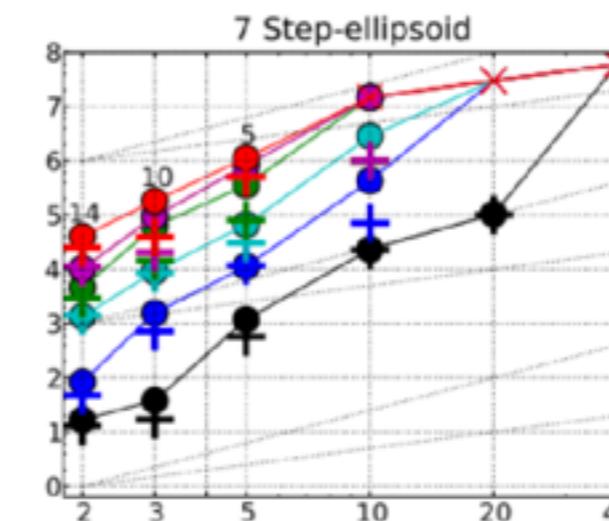
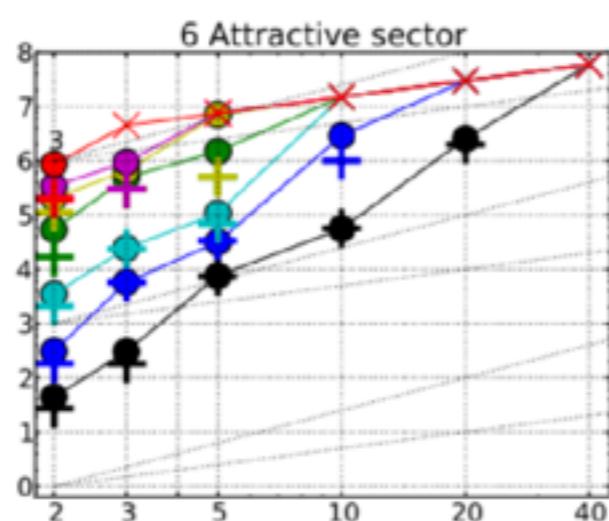
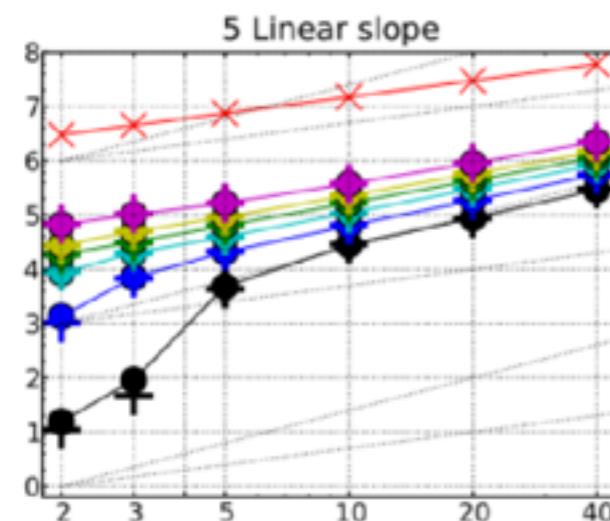
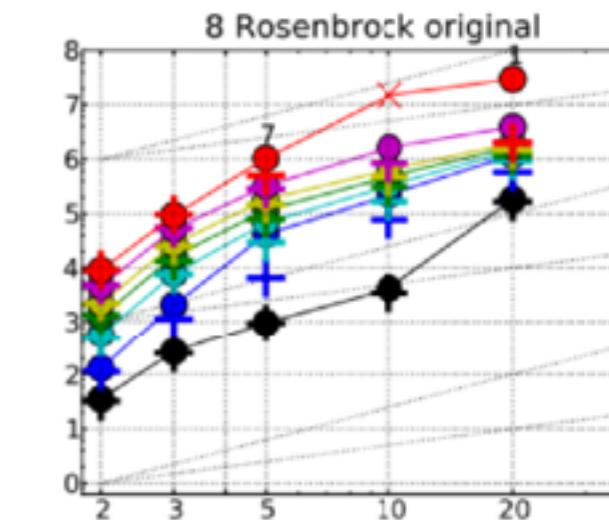
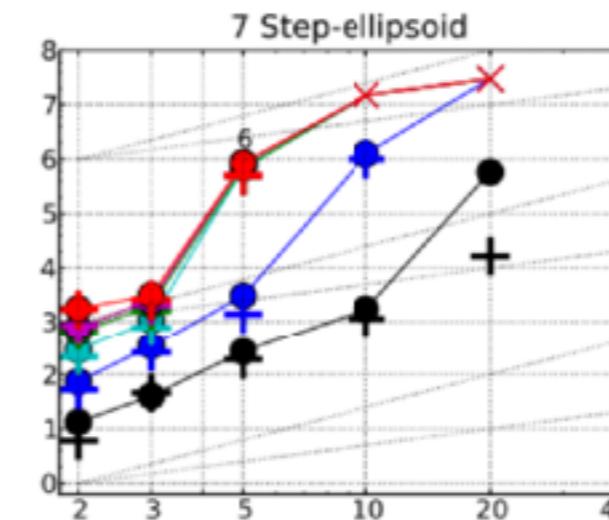
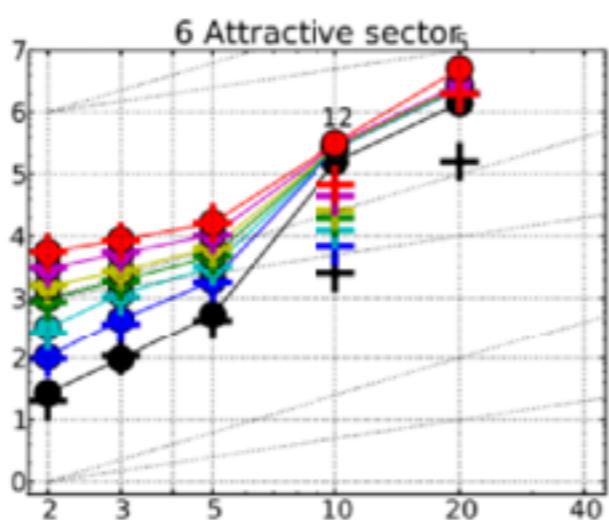
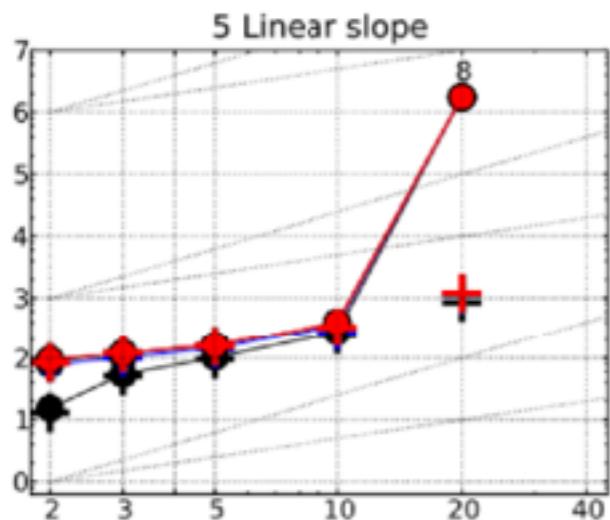
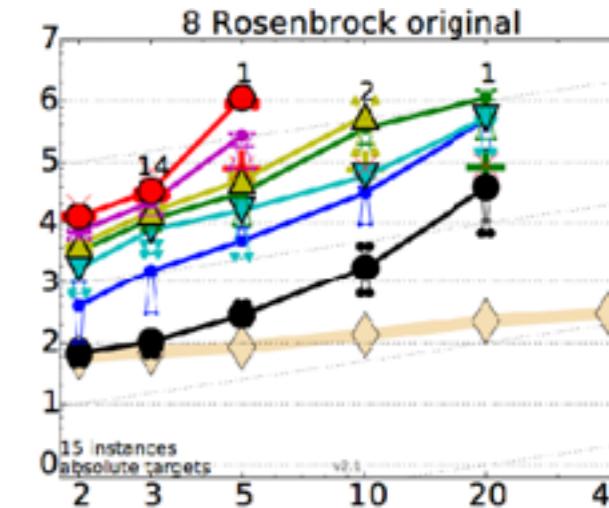
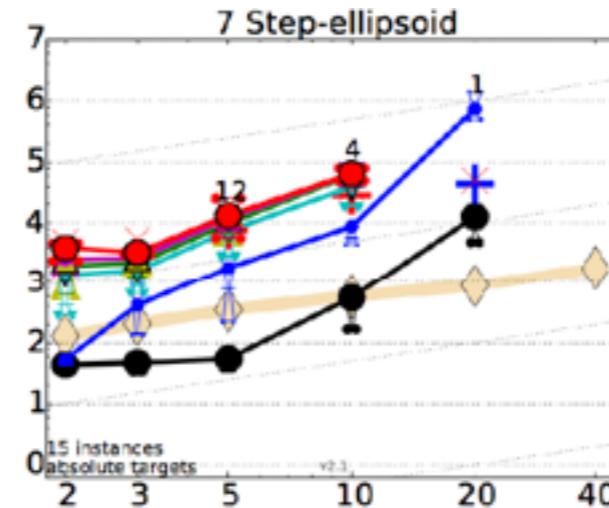
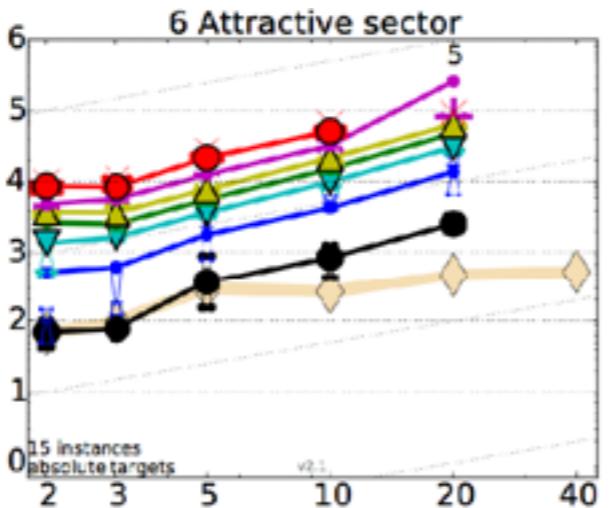
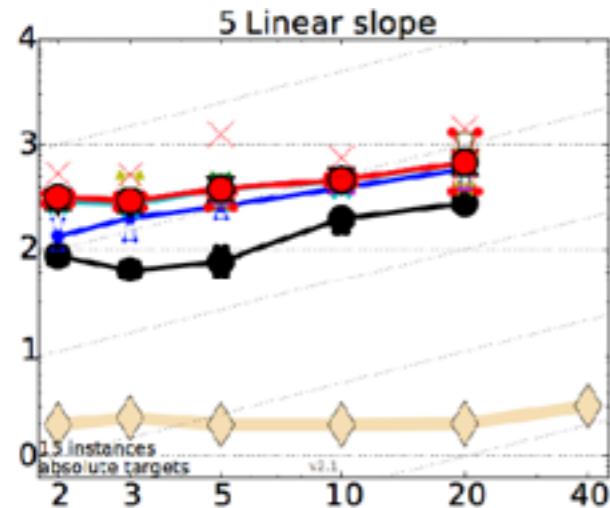
Results



Top: EvoSpace

Middle: PSO El-Abd and Kamel (2009)

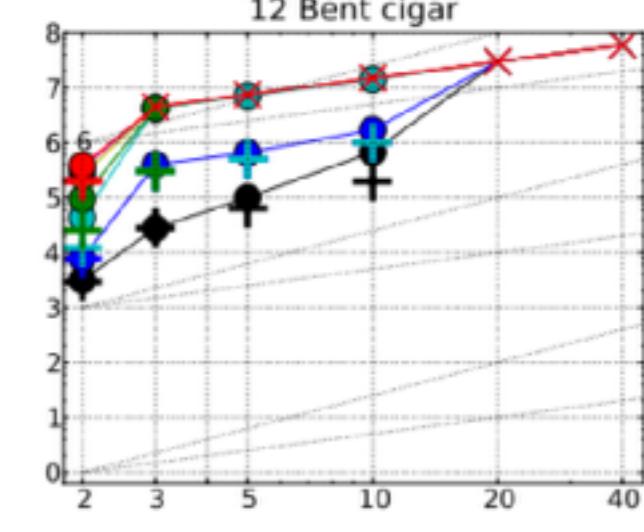
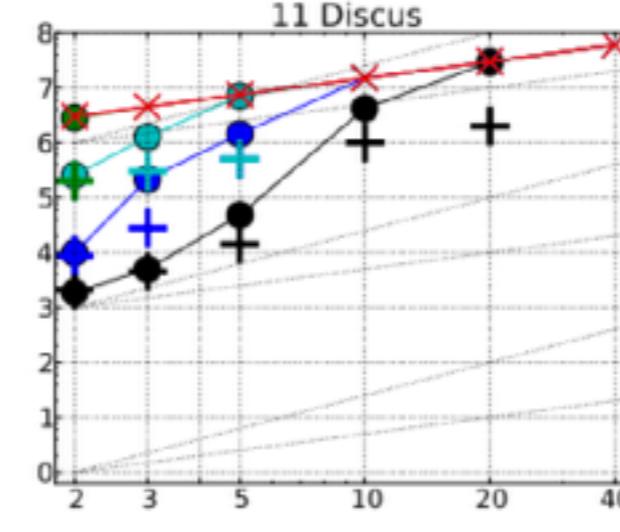
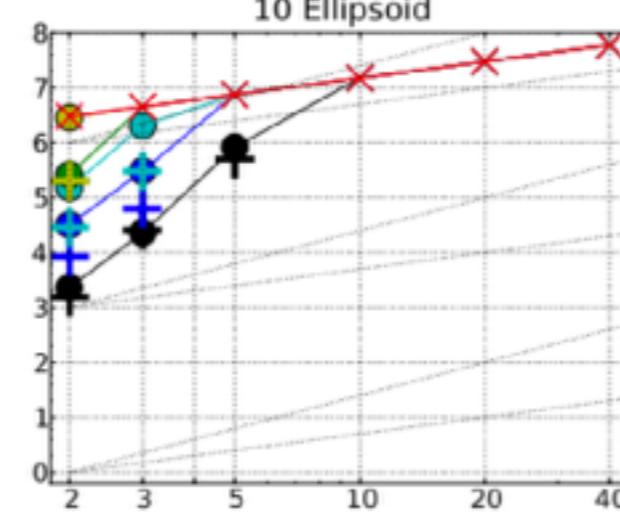
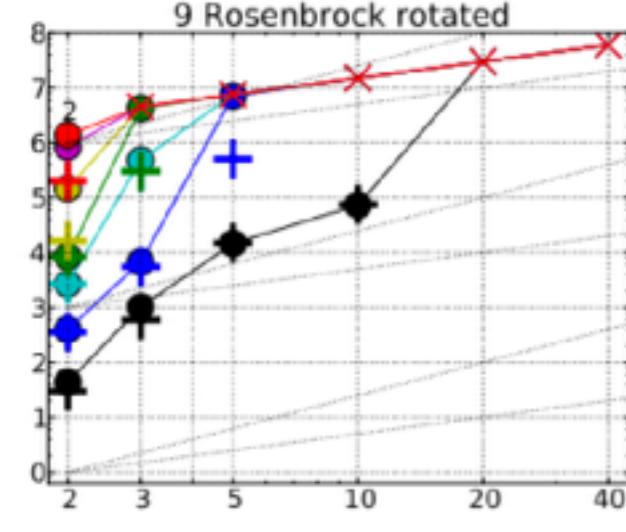
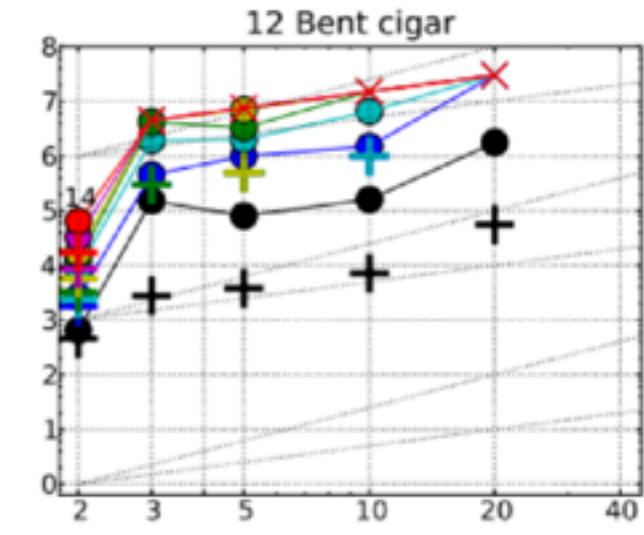
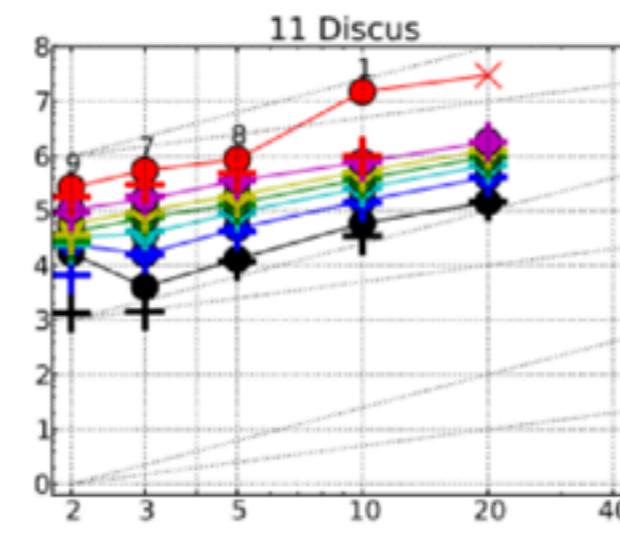
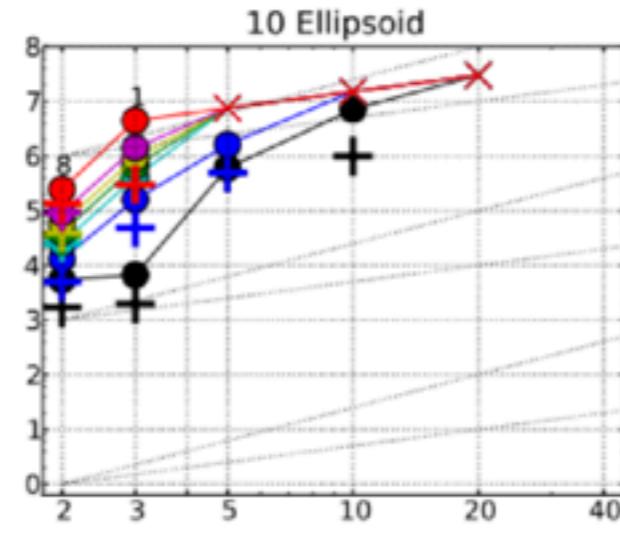
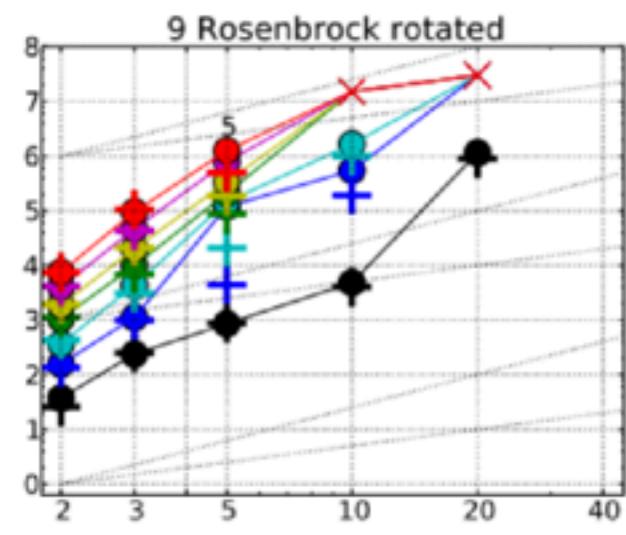
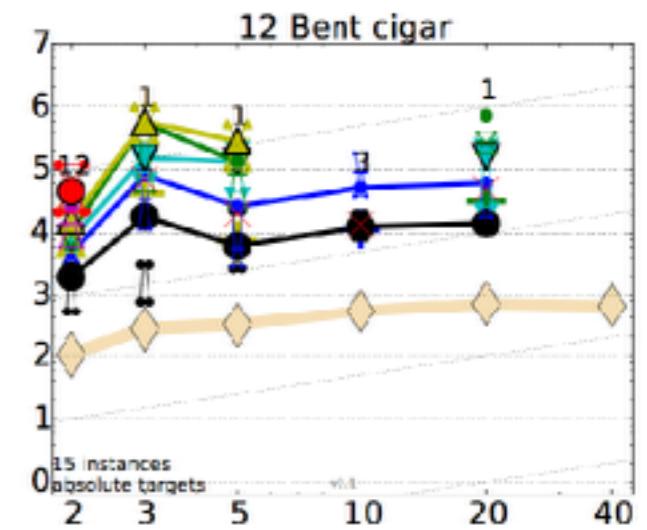
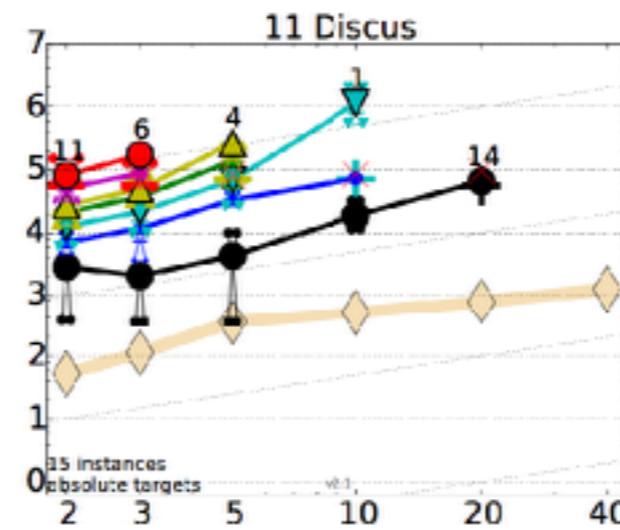
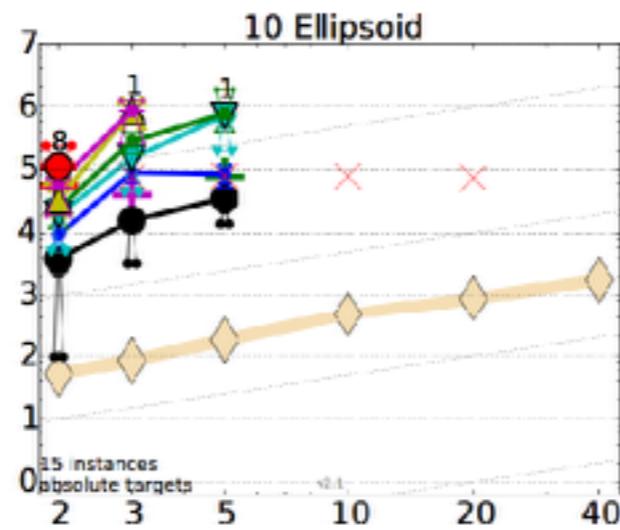
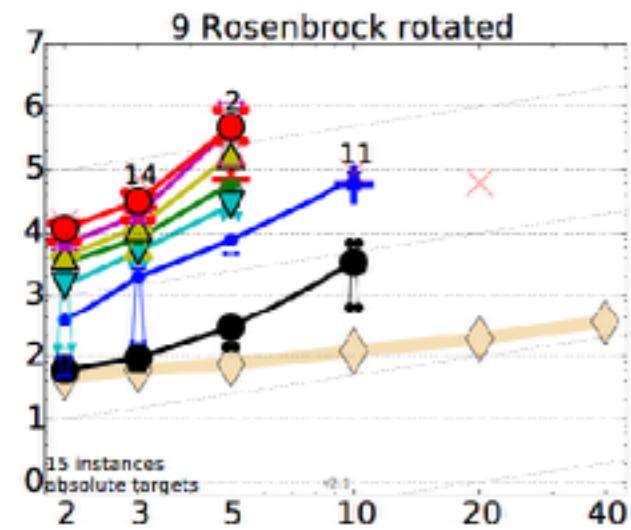
Bottom: GA Nicolau (2009)



Top: EvoSpace

Middle: PSO El-Abd and Kamel (2009)

Bottom: GA Nicolau (2009)



Top: EvoSpace

Middle: PSO El-Abd and Kamel (2009)

Bottom: GA Nicolau (2009)

Discussion

- Reaches the most difficult targets on separable functions (1-5).
- Scales well to higher dimensions.
- Results are competitive when compared to other nature-inspired algorithms.
- With harder problems, for instance functions 13 and 24, the algorithm is not capable of reaching the most difficult targets.

Discussion

- An asynchronous execution of population-based optimization algorithms following a Pool-based approach is possible and easy to achieve.
- Results, however, are still preliminary and further tuning of the parameters could potentially yield better results.

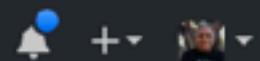
Future Work

- Experiment with other population based algorithms, with better performance in the COCO repository,
- Analyze the data provided by each Worker.
- Explore with Random Parameters in PSO.
- Docker Compose, Dockerfile. (COCO, evospace-js,EvoWorker).
- Upgrade to COCO 2.0 Functions.



This repository Search

Pull requests Issues Marketplace Gist



numbbo / coco

Watch ▾

16

Unstar

42

Fork

25

Code

Issues 137

Pull requests 0

Projects 12

Insights ▾

Numerical Black-Box Optimization Benchmarking Framework <http://coco.gforge.inria.fr/>

16,007 commits

11 branches

31 releases

15 contributors

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

 brockho committed on GitHub Merge pull request #1352 from numbbo/development	...	Latest commit 4b1497a on Apr 20
 code-experiments	A little more verbose error message when suite regression test fails	3 months ago
 code-postprocessing	Hashes are back on the plots.	3 months ago
 code-preprocessing	Fixed preprocessing to work correctly with the extended biobjective s...	5 months ago
 howtos	Update create-a-suite-howto.md	6 months ago
 .clang-format	raising an error in bbob2009_logger.c when best_value is NULL. Plus s...	2 years ago
 .hgignore	raising an error in bbob2009_logger.c when best_value is NULL. Plus s...	2 years ago
 AUTHORS	small correction in AUTHORS	a year ago

Thanks

Thank You

- <https://github.com/mariosky/evospace-js>
- <https://github.com/mariosky/EvoWorker>
- JJ Merelo, @jjmerelo, jmerelo@ugr.es
- Mario García-Valdez, @mariogarcia, mario@tectijuana.edu.mx