COCO code structure and adding a new suite

Tea Tušar
Inria Lille – Nord Europe

Paris, 26. 5. 2016

Source files (C89)

- Folder **src** contains:
  - 57 source files (*.c)
  - 3 header files (coco.h, coco_internal.h, coco_platform.h)
- After amalgamation:
  - coco.c
  - coco.h
- Amalgamated files copied to **build/<language>** folders

coco.h

- API (application programming interface) – declares all public functions, constants and variables
- Provides interfaces to:
  - Problem suites
  - Problems
  - Observers
  - Archives
- Declares some other useful functions (memory handling, random numbers etc.)

coco.h

- Simplified usage example:

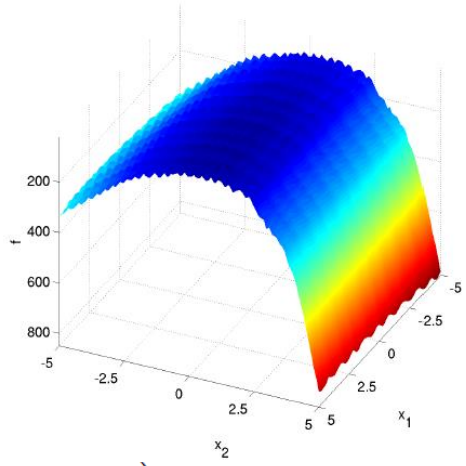
```
coco_suite_t *suite = coco_suite("bbob-biobj", "year: 2016", "");
coco_observer_t *observer = coco_observer("bbob-biobj", "");
coco_problem_t *problem;

while ((problem = coco_suite_get_next_problem(suite, observer))
    != NULL) {
    my_optimizer(...);
}

coco_observer_free(observer);
coco_suite_free(suite);
```

Source file organization

- General:
  - coco_*.c
- Functions and problems:
  - f_*.c
- Transformations:
  - transform_vars_*.c
  - transform_obj_*.c



Rastrigin Function

$$f_3(\mathbf{x}) = 10\left(D - \sum_{i=1}^{D}\cos(2\pi z_i)\right) + \|\mathbf{z}\|^2 + f_{\text{opt}}$$

$$\mathbf{z} = \Lambda^{10} T_{\text{asy}}^{0.2}(T_{\text{osz}}(\mathbf{x} - \mathbf{x}^{\text{opt}}))$$

f_rastrigin_bbob_problem_allocate:

1. problem = f_rastrigin_allocate(dimension)
2. problem = transform_vars_conditioning(problem, 10.0)
3. problem = transform_vars_asymmetric(problem, 0.2)
4. problem = transform_vars_oscillate(problem)
5. problem = transform_vars_shift(problem, xopt, 0)
6. problem = transform_obj_shift(problem, fopt)

Source file organization

- General:
  - coco_*.c
- Functions and problems:
  - f_*.c
- Transformations:
  - transform_vars_*.c
  - transform_obj_*.c
- Problem suites:
  - suite_*.c
- Observers:
  - observer_*.c
- Loggers:
  - logger_*.c
- Other:
  - large_scale_*.c
  - mo_*.c

Existing problem suites and observers

- Problem suites:
  - bbob
  - bbob-biobj
  - bbob-largescale
  - toy
- Observers:
  - bbob
  - bbob-biobj
  - toy

Writing a new problem suite

- Using existing problems:
  - Construct the suite
- Using new problems:
  - Implement new problems (functions and transformations)
  - Construct the suite
- Possibly implement a new observer/logger

Implement a new **cool** problem

- Implement new **cool** function:
  - y = **f_cool_raw**(x, dim)
  - **f_cool_evaluate**(problem, x, y)
  - problem = **f_cool_allocate**(dim)

- Implement new **hot** transformation of variables:
  - **transform_vars_hot_evaluate**(problem, x, y)
  - problem = **transform_vars_hot**(inner_problem, parameters)

Implement a new **cool** problem

- Implement new **green** objective transformation:
  - **transform_obj_green_evaluate**(problem, x, y)
  - problem = **transform_obj_green**(inner_problem, parameters)

- **f_cool_problem_allocate**(function, dim, instance):
  - if (function == 1) {

    problem = **f_cool_allocate**(dim)

    problem = **transform_vars_hot**(problem, instance, 2.5)

    problem = **transform_obj_green**(problem, instance, 2.0, 3.0)

    } else ...
  - set ID, name, ...

Construct a new **sunny** suite

- **suite_sunny_initialize**()
  - Suite name
  - Number of available functions
  - Available dimensions
  - Default instances
- problem = **suite_sunny_get_problem**(function, dim, instance)
  - Return the right problem for the given parameters
- Add **sunny** to COCO problem suites