solar: A solar thermal power plant simulator for blackbox optimization benchmarking

Sébastien Le Digabel

GERAD
GROUP FOR RESEARCH IN DECISION ANALYSIS

POLYTECHNIQUE MONTRÉAL
TECHNOLOGICAL UNIVERSITY

SIAM CSE23, 2023-03-02

Presentation outline

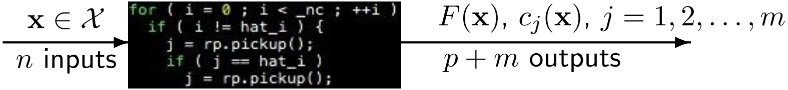
Contributors

- This work is based on the MSc thesis of [Lemyre Garneau, 2015]
- The other contributors are
  - Charles Audet
  - Miguel Diago
  - Aïmen Gheribi
  - Mona Jeunehomme
  - Xavier Lebeuf
  - Viviane Rochon Montplaisir
  - Bastien Talgorn
  - Nicolau Andres Thio
  - Christophe Tribes

Context: Blackbox Optimization (BBO)

$$\min_{\mathbf{x}\in\mathcal{X}} \quad F(\mathbf{x}) \text{ s.t. } \mathbf{x}\in\Omega=\{\mathbf{x}\in\mathcal{X} : c_j(\mathbf{x})\leq 0, j=1,2,\ldots,m\}$$

$\mathcal{X}$ is a n-dimensional space, F can have $p=1$ or $p=2$ components, and the evaluations of F and the c_j's are provided by a blackbox:



- Each call to the blackbox may be expensive
- The evaluation can fail
- Sometimes $F(\mathbf{x})\neq F(\mathbf{x})$
- Derivatives are not available and cannot be approximated

Objectives of this work

Provide a realistic application for “true” BBO benchmarking, that

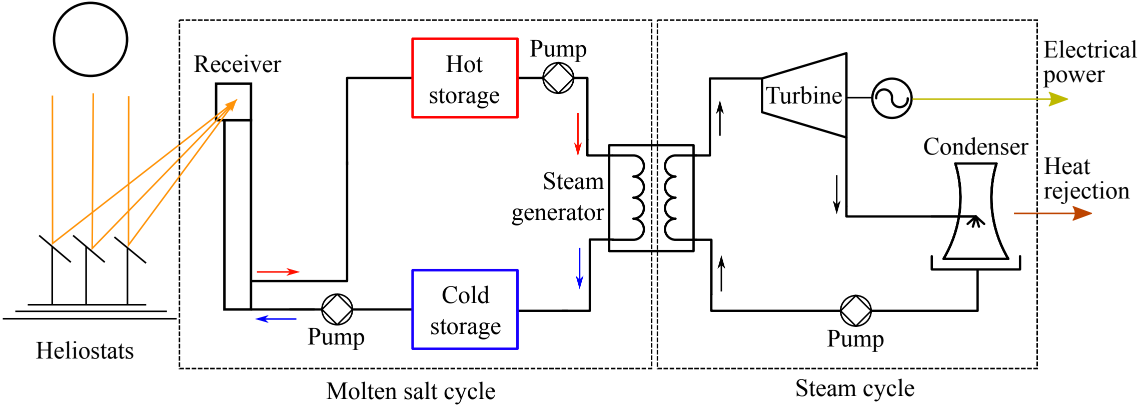
- is easy to install (stand-alone, standard code)
- is multiplatform
- allows to reproduce results
- includes many options allowing to
  - test different aspects of BBO such as
    - time-consuming evaluations
    - discrete/categorical variables
    - constraints handling
    - noise in the blackbox outputs
    - static surrogates
    - multiobjective optimization
  - propose sets of instances to draw performance/data profiles

CSP tower plant with molten salt thermal energy storage

- A large number of mirrors (heliostats) reflects solar radiation on a receiver at the top of a tower
- The heat collected from the concentrated solar flux is removed from the receiver by a stream of molten salt
- Hot molten salt is then used to feed thermal power to a conventional power block
- The photo shows the Thémis CSP power plant, the first built with this design

Source: https://commons.wikimedia.org/wiki/File:Themis_2.jpg

System dynamics



The solar code is

- a command-line application
- the “natural heir” of our STYRENE simulator [Audet et al., 2008]
- publicly available at https://github.com/bbopt/solar under the GNU Lesser General Public License
- a relatively simple code in standard C++ ($\simeq$15k lines of codes)
- stand-alone: no external library to install
- multi-platform: C++ compilator is the only requirement

Ten instances

Instance	# of variables			# of obj.	# of constraints			# of stoch. outputs	Static
	cont.	discr. (cat.)	n	p	simu.	a priori (lin.)	m	(obj. or constr.)	surrogate
solar1	8	1 (0)	9	1	2	3 (2)	5	1	no
solar2[1]	12	2 (0)	14	1	9	4 (2)	13	3	yes
solar3	17	3 (1)	20	1	8	5 (3)	13	5	yes
solar4	22	7 (1)	29	1	9	7 (5)	16	6	yes
solar5	14	6 (1)	20	1	8	4 (3)	12	0	no
solar6	5	0 (0)	5	1	6	0 (0)	6	0	no
solar7	6	1 (0)	7	1	4	2 (1)	6	3	yes
solar8	11	2 (0)	13	2	4	5 (3)	9	3	yes
solar9	22	7 (1)	29	2	10	7 (5)	17	6	yes
solar10[2]	5	0 (0)	5	1	0	0 (0)	0	0	yes

[1] analytic objective
[2] unconstrained

Types of variables

$$\min_{\mathbf{x}\in\mathcal{X}} \quad F(\mathbf{x}) \text{ s.t. } \mathbf{x} \in \Omega = \{\mathbf{x} \in \mathcal{X} : c_j(\mathbf{x}) \leq 0, j = 1, 2, \ldots, m\}$$

- The n variables are described by the set $\mathcal{X}$. They can be continuous or discrete
- $\mathcal{X}$ includes bounds on most of the variables
- There are 29 possible variables. Each instance considers a subset of these variables. solar4 and solar9 consider all $n = 29$ variables
- The solar6 and solar10 instances have no discrete variables. In these cases $\mathcal{X} \subset \mathbb{R}^5$
- One of the discrete variable (the type of turbine) is categorical. solar considers it as an integer in $\{1, 2, \ldots, 8\}$

Types of constraints

$$\min_{\mathbf{x}\in\mathcal{X}} \quad F(\mathbf{x}) \text{ s.t. } \mathbf{x} \in \Omega = \{\mathbf{x} \in \mathcal{X} : c_j(\mathbf{x}) \leq 0, j = 1, 2, \ldots, m\}$$

- $\mathcal{X}$ describes bounds on the variables and the discrete nature of some of the variables. These constraints are unrelaxable
- The m constraints in Ω may be a priori or simulation constraints
- A priori constraints are also unrelaxable. In case of violation, the solar executable returns a flag to indicate a potential solver not to count the evaluation
- Most of the a priori constraints are linear
- Simulation constraints are relaxable
- Presence of hidden constraints
- All constraints (except the hidden ones) are quantifiable

- There are 18 possible constraints. Each instance considers a subset of these constraints, for a maximum of $m = 17$ constraints in solar9

Getting started with solar

- Get the code at https://github.com/bbopt/solar and compile
- Command-line program that takes as arguments
  - a problem id (or instance number) in $\{1, 2, \ldots, 10\}$
  - the name of a file containing the coefficients of a point $\mathbf{x}$

  and displays the values of $F(\mathbf{x})$ and the $c_j(\mathbf{x})$'s
- Example: `> solar 7 x.txt` displays `f c1 c2 ... c6`
  (objective and six constraints)
- Simply executing `> solar` will guide the user and display the options, including a complete inline help with `> solar -help`

Check the solar installation

```
> solar -check
```

Mac:
Core i9: 659s
M1 Pro: 451s
M1 Max: 444s
M2 Max: 393s

Windows:
Core i7: 2,684s

Linux:
AMD EPYC: 1,284s

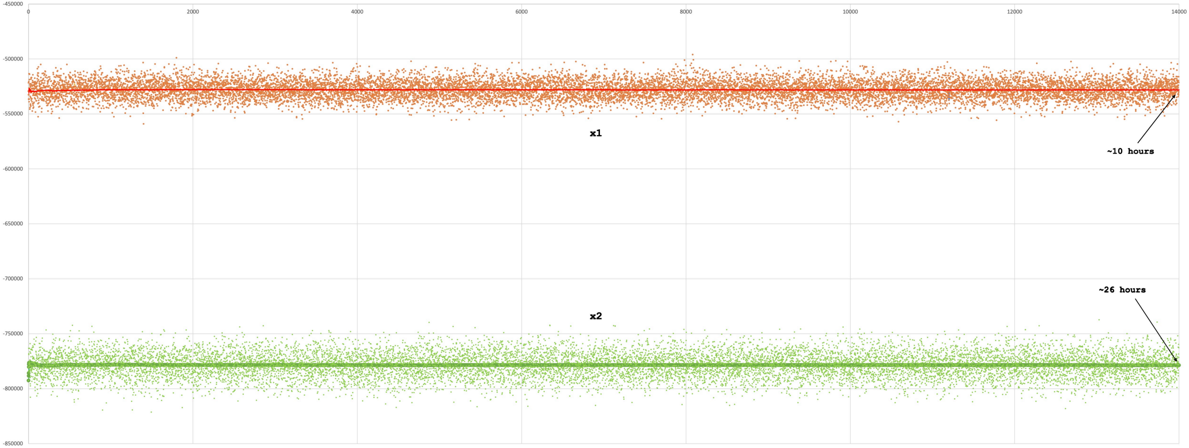
```
[[12:34:11] [~/Desktop] > ./solar -check

Validation tests (can take several minutes):

        RNG test  ( 1/ 2) ...... Ok     Time: CPU=8.8e-05       real=0
        RNG test  ( 2/ 2) ...... Ok     Time: CPU=9e-06         real=0
        Eval test ( 1/26) ...... Ok     Time: CPU=0.090865      real=0
        Eval test ( 2/26) ...... Ok     Time: CPU=0.164074      real=0
        Eval test ( 3/26) ...... Ok     Time: CPU=8.55466       real=9
        Eval test ( 4/26) ...... Ok     Time: CPU=14.3939       real=14
        Eval test ( 5/26) ...... Ok     Time: CPU=12.444        real=12
        Eval test ( 6/26) ...... Ok     Time: CPU=1.67694       real=2
        Eval test ( 7/26) ...... Ok     Time: CPU=1.714         real=2
        Eval test ( 8/26) ...... Ok     Time: CPU=0.000297      real=0
        Eval test ( 9/26) ...... Ok     Time: CPU=1.8335        real=2
        Eval test (10/26) ...... Ok     Time: CPU=16.9975       real=17
        Eval test (11/26) ...... Ok     Time: CPU=0.088462      real=0
        Eval test (12/26) ...... Ok     Time: CPU=1.76882       real=2
        Eval test (13/26) ...... Ok     Time: CPU=2.03457       real=2
        Eval test (14/26) ...... Ok     Time: CPU=57.289        real=57
        Eval test (15/26) ...... Ok     Time: CPU=76.4028       real=76
        Eval test (16/26) ...... Ok     Time: CPU=2.17247       real=2
        Eval test (17/26) ...... Ok     Time: CPU=50.1873       real=51
        Eval test (18/26) ...... Ok     Time: CPU=50.3843       real=50
        Eval test (19/26) ...... Ok     Time: CPU=50.3955       real=50
        Eval test (20/26) ...... Ok     Time: CPU=3.31858       real=4
        Eval test (21/26) ...... Ok     Time: CPU=3.21749       real=3
        Eval test (22/26) ...... Ok     Time: CPU=5.77947       real=6
        Eval test (23/26) ...... Ok     Time: CPU=0.003279      real=0
        Eval test (24/26) ...... Ok     Time: CPU=3.86108       real=4
        Eval test (25/26) ...... Ok     Time: CPU=2.24941       real=2
        Eval test (26/26) ...... Ok     Time: CPU=25.7252       real=26

This version of SOLAR is valid

CPU time : 392.748s
Real time: 393s
```

Stochasticity and replications

- Stochasticity is due to the Monte Carlo simulation for the heliostats field
- Random seed is set to the same value by default: This corresponds to a deterministic blackbox
- Use the option `-seed` to change the random seed
- The option `-seed=diff` makes the blackbox stochastic
- The option `-rep` executes several simulations and outputs average values
- A high number of replications will tend to decrease stochasticity but will lead to expensive evaluations (which is great in BBO benchmarking)

Illustration of replications for the objective of solar1

`> solar 1 -rep=14000 x1.txt` and `> solar 1 -rep=14000 x2.txt`



Multi-fidelity

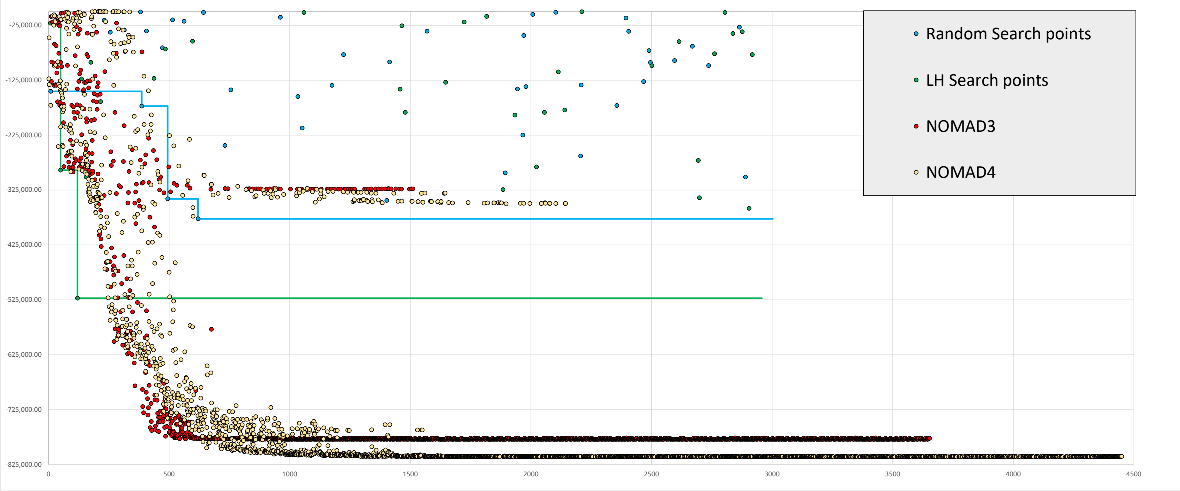
- The option `-fid` with a value in $]0;1]$ changes the fidelity of the simulator
- Each different value of this option generates a static surrogate
- `-fid=1` corresponds to the "true" blackbox (called the truth)
- This option allows to consider multi-fidelity metamodels or variable precision static surrogates
- Note that using the `-rep` option also allows to consider such surrogates when the truth is considered to be obtained with high number of replications

Feasibility with sampling and NOMAD

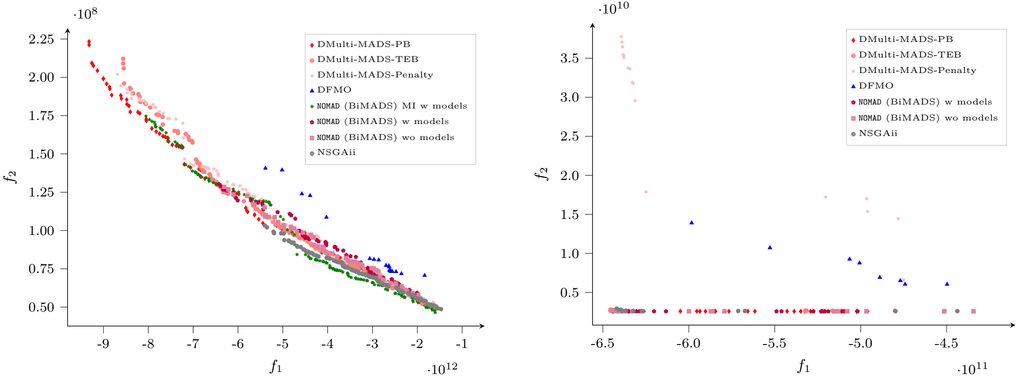
Instance	LH search (10k points)		NOMAD3		
	satisf. ap constr.	feas. pts	satisf. ap constr.	feas. pts	number of eval.
solar1	30%	0.35%	96%	74%	3,792
solar2	0%	0%	97%	0%	1,635
solar3	0.49%	0%	99%	9%	30,525
solar4	0%	0%	83%	0%	44,303
solar5	0%	0%	83%	59%	3,405
solar6	90%	5%	99%	0%	3,539
solar7	2%	1%	74%	72%	2,224
solar8	1%	0.03%			
solar9	1%	0%			

there has been no violation of hidden constraints during the construction of this table

Optimization on solar1



Biobjective optimization (by L. Salomon)



Pareto front approximations for solar8 (left) and solar9 (right) with different solvers with a budget of 5K evaluations. Taken from [Bigeon et al., 2022]

References I

Audet, C., Béchard, V., and Le Digabel, S. (2008).
Nonsmooth optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search.
Journal of Global Optimization, 41(2):299–318.

Bigeon, J., Le Digabel, S., and Salomon, L. (2022).
Handling of constraints in multiobjective blackbox optimization.
Technical Report G-2022-10, Les cahiers du GERAD.

Le Digabel, S. and Wild, S. (2015).
A Taxonomy of Constraints in Simulation-Based Optimization.
Technical Report G-2015-57, Les cahiers du GERAD.

Lemyre Garneau, M. (2015).
Modelling of a solar thermal power plant for benchmarking blackbox optimization solvers.
Master's thesis, Polytechnique Montréal.