A Tutorial on Black–Box Optimization

Andrea Cassioli

[1] LIX - ECOLE POLYTECHNIQUE (FR)
cassioli@lix.polytechinique.fr

25/04/2013

Black–Box Function

Informal Definition

Black-Box Function

A function $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ for which the analytic form is not known.

Black–Box Function

Informal Definition

Black-Box Function

A function $f(x) : \mathbb{R}^n \to \mathbb{R}$ for which the analytic form is not known.

Typically a black-box function can be evaluated to obtain:

- value;
- definitnesss;
- (approximate)gradient

Black–Box Function

Informal Definition

Black-Box Function

A function $f(x) : \mathbb{R}^n \to \mathbb{R}$ for which the analytic form is not known.

Typically a black-box function can be evaluated to obtain:

- value;
- definitnesss;
- (approximate)gradient

Warning!

A black-box function is not necessarily a ”nasty” one, it might be smooth and defined every where and even convex...

Global optimization of Black–Box functions

Informal Definition

$$
\begin{aligned}
&\min_x f(x) \\
&g_j(x) \leq 0 && j = 1 \dots k \\
&x_i \in [l_i, u_i] \subseteq [-\infty, +\infty] && i = 1 \dots n
\end{aligned}
$$

Black-Box Optimization Problem

An optimization model in which at least for a function is a black-box function.

Global optimization of Black–Box functions

Examples

- Legacy code: no access to what is inside a library and/or an executable
- Numerical code involving PDE's, integrals...
- real-life experiments: crash tests, chemical reactions, etc...

Observation on Black–Box functions

A critical issue: computational cost

How much does a function evaluation ”cost”?

With *cost* we mean any measure of the resources needed to evaluate the function (convertible somehow to money).

Observation on Black–Box functions

A critical issue: computational cost

How much does a function evaluation ”cost”?

With *cost* we mean any measure of the resources needed to evaluate the function (convertible somehow to money).

- *cheap function:* it can be evaluated thousands of times
- *costly function:* it can be evaluated few times times (typically $\ll 200$)

Black–Box Optimization
Main tools – Sampling

For cheap black-box functions we can sample the feasible set:

- randomly
- with experiment design (as Latin Hypercube)
- deterministic

We may want to sample around an available point (intensification) or just everywhere on the feasible set (exploration).

Surrogate model

A mathematical data-driven model that mimic the behavior of another model *as closely as possible* while being computationally cheap(er) to evaluate. (Wikipedia)

Surrogate model

A mathematical data-driven model that mimic the behavior of another model *as closely as possible* while being computationally cheap(er) to evaluate. (Wikipedia)

Accuracy is often required only in some regions of the feasible set....

Large amount of function evaluation allows for gradient approximation (forward/centered):

$$\left.\frac{\partial f(x)}{\partial x}\right|_{\bar{x}} \approx \frac{f(\bar{x}) - f(\bar{x}+\delta)}{\delta}$$

- steepest-descent;
- Quasi-Newton methods (L)-BFGS (see the work of Overton for instance);
- the *implicit filtering* methods of Kelley.

Direct Search Algorithms

Main idea

A broad family of algorithms built on a simple idea: given a point $\bar{x}$ and a finite set of directions $D(\bar{x})$ such that

$$\exists d \in D(\bar{x}), d^T \nabla f(\bar{x}) \leq 0,$$

then there exists an $\alpha > 0$ small enough such that

$$f(\bar{x}) \geq f(\bar{x} + \alpha d).$$

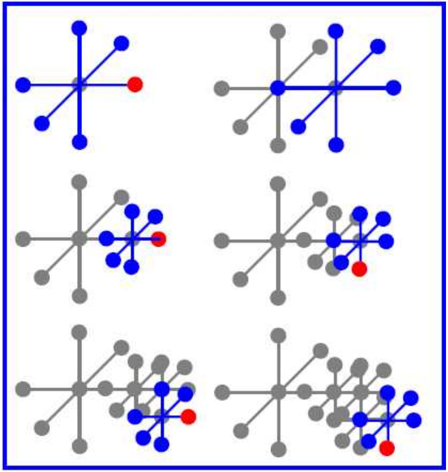
Direct Search Algorithms

Main idea

Input: x_0, α
$f^\star \leftarrow f(x_0)$;
$k \leftarrow 0$;
while *stopping criterion* **do**
    $f_k^\star \leftarrow \min_{d \in D(x_k)} f(x_k + \alpha d)$;
    $x_k^\star \leftarrow \arg\min_{d \in D(x_k)} f(x_k + \alpha d)$;
    if $f_k^\star < f^\star$ **then**
        $x_{k+1} \leftarrow x_k^\star$
    else
        update (shrink) α
    end
    $k \leftarrow k + 1$;
end

Direct Search Algorithms

A *compass search* example



Pattern-Search

Observations

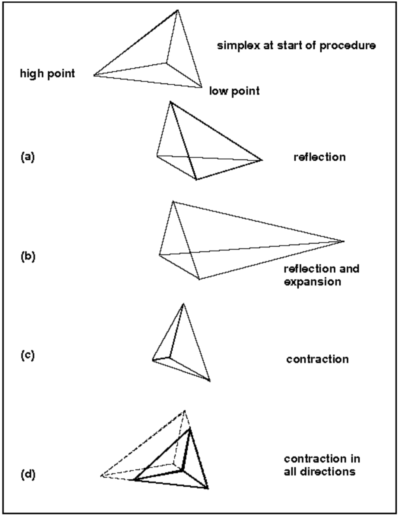
There is an endless number of variants on:

1. how to define D
2. how to select α
3. how to deal with constraints

Kolda et al. *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM review, 2003

Cheap Black–Box Functions

Nelder-Mead Simplex Method



Cheap Black–Box Functions

DIRECT (DIviding RECTangles)

A theoretically sound method for box-constrained problems based on feasible set sequential partition.

Jones et al. *Lipschitzian optimization without the Lipschitz constant*, JOTA 1993

Cheap Black–Box Functions

Global Optimization

Any algorithm based only on function evaluation *might* work:

1. Genetic Algorithms
2. Particle-Swarm
3. Differential Evolutions
4. Variable-Neighborhood Search
5. ....

Cheap Black–Box Functions

Global Optimization

Any algorithm based only on function evaluation *might* work:

1. Genetic Algorithms
2. Particle-Swarm
3. Differential Evolutions
4. Variable-Neighborhood Search
5. ....

Easy to implement and to parallelize, almost no convergence theory and in general quite poor performances.

Cheap Black–Box Functions

Hybrid Approaches

To balance the global/local phases, use a two-phase approach:

1. use a GO algorithm to generate a new set of points (exploration)
2. start local searches from some of them

Cheap Black–Box Functions

Hybrid Approaches

To balance the global/local phases, use a two-phase approach:

1. use a GO algorithm to generate a new set of points (exploration)
2. start local searches from some of them

It can be very effective but more complex to implement and tune.

- Cassioli et al., *A global optimization method for the design of space trajectories*, COAP 2011
- Vicente et al. on `http://www.norg.uminho.pt/aivaz/pswarm/`

Costly Black–Box Functions

Main ideas

We need to minimize the number of function evaluations to accomplish our task...but how?

Costly Black–Box Functions

Main ideas

We need to minimize the number of function evaluations to accomplish our task...but how? Let assume that:

1. in a set S of points $f(\cdot)$ has been evaluated
2. a surrogate model $s(\cdot|f(\cdot), S)$ can be defined "*easily*"

Costly Black–Box Functions
Main ideas

We need to minimize the number of function evaluations to accomplish our task...but how? Let assume that:

1. in a set S of points $f(\cdot)$ has been evaluated
2. a surrogate model $s(\cdot|f(\cdot), S)$ can be defined "*easily*"

We want to choose a point $x \notin S$ that maximize the expected result on the evaluation of $f(x)$.

Costly Black–Box Functions

The merit function

To carefully select the next point in which evaluate $f(x)$ we use a merit function $\mu(\cdot)$ such that:

1. it's cheap to evaluate and possibly to create
2. enjoy all desirable properties for GO
3. depends on $f(x)$ and S

We solve:

$$
\begin{aligned}
& \max \mu(x|f, S) & \\
& g_j(x) \le 0 & j = 1 \dots k \\
& x_j \in [l_i, u_i] \subseteq [-\infty, +\infty] & i = 1 \dots n
\end{aligned}
$$

Costly Black–Box Functions

The merit function

- the merit function can change over the iterations (exploration vs. intensification)
- it can be the surrogate model itself
- it might require the global optimum of the surrogate

The optimization of $\mu(\cdot)$ should be easy and fast compare to the evaluation of $f(\cdot)$.

Costly Black–Box Functions

Template Algorithm

$k \leftarrow 0$;

compose S^0;

while *stopping criterion* **do**

  build $s(\cdot|f, S^k)$;

  $x^k \leftarrow \arg\max_x \mu(x, s(\cdot|f, S^k))$;

  $S^{k+1} = S^k \cup (x^k, f(x^k))$;

  $k \leftarrow k + 1$;

end

Costly Black–Box Functions

Surrogate models

- EGO D.Jones et al. *Efficient global optimization of expensive black-box functions*, JOGO 1998
- Kriging
- RBF: use radial basis function and polynomials, no statistic assumptions
  Gutmann *A radial basis function method for global optimization.*, JOGO 2001
  Holmström et al., *An adaptive radial basis algorithm (ARBF) for expensive black-box global optimization.*, JOGO 2008
- SVM: very similar to RBF
  Suykens *Nonlinear modelling and support vector machines.* IMTC 2001. Proc. IEEE, 2001

Costly Black–Box Functions
Extensions

- choice of different surrogate models based on the iteraton
- combining linealry different surrogate models
- Trust-Region approaches: instead of S, people use a subset $\bar{S}$ around the best/current iterate, to improve refinement and exploit good starting points.
- Interpolation vs. Approximation

See the endless series of paper of Regis and Shoemaker.

Costly Black–Box Functions
Observations

1. most algorithms has convergence properties under mild assumption, but in practice this does not really matter!
2. it is diffucult to asses performance and to compare algorithms
3. still an active field of research, especially the RBF approach

That's all...

questions?

THANK YOU!