Delaunay-based Derivative-free Optimization via Global Surrogate

Pooriya Beyhaghi, Daniele Cavaglieri and Thomas Bewley

UC San Diego

May 23, 2014

Delaunay-based Derivative-free Optimization via Global Surrogate

Pooriya Beyhaghi, Daniele Cavaglieri and Thomas Bewley

UC San Diego

May 23, 2014

Outline

- Introduction
- Δ-Dogs for problems with a linear constraints.
- Δ-Dogs for problems with a general convex constraints.
- Minimizing the cost function that is derived by the infinite time-averaged.
- Conclusion.

Properties of the Derivative free Algorithms

Advantages

- Does not need any information about the derivative.
- Can handle problems with noisy or inaccurate cost function evaluations.
- Capability of the global Search

Disadvantages

- High computational cost with respect to the dimension of the problem.
- Slow speed of convergence.

General classification of the Derivative free methods

- Direct methods
- Nelder-Mead method
- Response surface methods
- Branch and bound algorithms
- Bayesian approaches
- Adaptive search algorithms
- Hybrid methods

General implementation of the response surface methods

- Design a model (interpolation) for the cost function based on the current data points.
- Find the most promising points for the global minimum based on the model.
- Calculate the cost function evaluation at the new data point.
- Add it to the data set, continue the algorithm until the global (local) minimum is found.

Optimization base on the kriging interpolations

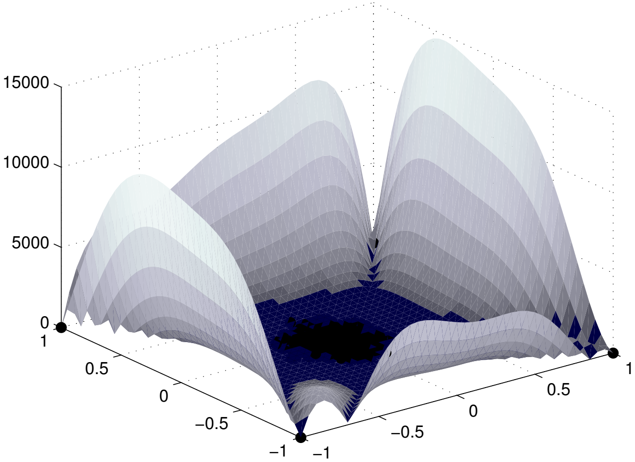
Advantages

- Have an estimation for both the cost function and its uncertainty at each feasible point.
- Can handle scattered data.
- Could be extended to high dimensional problems

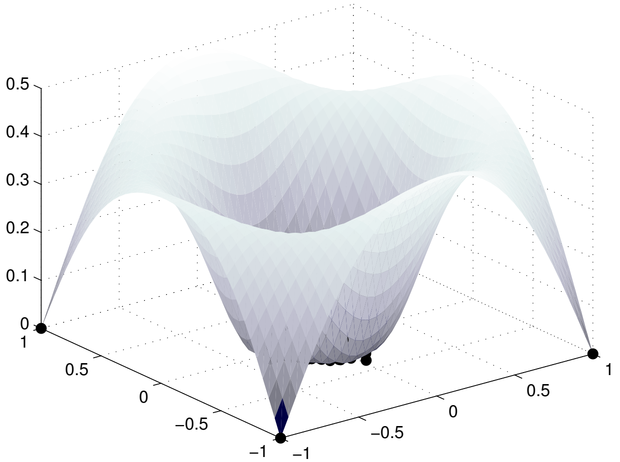
Disadvantages

- Has the numerical inaccuracy when the data points are clusttred in some region of the domain.
- Finding parameters of the Kriging interpolation is a hard non-convex subproblem.
- Minimizing the search function at each step is a non-smooth, non-convex optimization algorithm.

Performance of the Kriging for an illposed example



Performance of the polyharmonic spline for an illposed example



Initialization of the algorithm for problems with bounded Linear constraints

- In order to initialize the algorithm, a set of data points is needed that its convex hull is the feasible domain.
- The minimal subset of the feasible domain that its convex hull is our constraint is the set of vertices.
- There are some algorithms to find all vertices of a linear constrained problem.
- The box constraint is a special case which the corners are these vertices.

The Optimization Algorithm for the problems with linear constraints

- 1. Find all vertices of the feasible domain.

The Optimization Algorithm for the problems with linear constraints

- 1. Find all vertices of the feasible domain.
- 2. Define a set of initial evaluation points and add the vertices to it.

The Optimization Algorithm for the problems with linear constraints

- 1. Find all vertices of the feasible domain.
- 2. Define a set of initial evaluation points and add the vertices to it.
- 3. Calculate an interpolating function $p(x)$ among the set of evaluation points.

The Optimization Algorithm for the problems with linear constraints

- 1. Find all vertices of the feasible domain.
- 2. Define a set of initial evaluation points and add the vertices to it.
- 3. Calculate an interpolating function $p(x)$ among the set of evaluation points.
- 4. Perform a Delaunay triangulation among the points.

The Global Optimization Algorithm for the problems with linear constraints

- 5. For each simplex S_i
    - Calculate its circumcenter x_C and the circumradius R.

The Global Optimization Algorithm for the problems with linear constraints

- 5. For each simplex S_i
  - Calculate its circumcenter x_C and the circumradius R.
  - Define an error function $e_i(x)$ on each simplex such that $e_i(x) = R^2 - (x - x_C)^T(x - x_C)$.

The Global Optimization Algorithm for the problems with linear constraints

- 5. For each simplex S_i
  - Calculate its circumcenter x_C and the circumradius R.
  - Define an error function $e_i(x)$ on each simplex such that $e_i(x) = R^2 - (x - x_C)^T(x - x_C)$.
  - Define a search function in this simplex as $c_i(x) = p(x) - K\, e_i(x)$.

The Global Optimization Algorithm for the problems with linear constraints

- 5. For each simplex S_i
  - Calculate its circumcenter x_C and the circumradius R.
  - Define an error function $e_i(x)$ on each simplex such that $e_i(x) = R^2 - (x - x_C)^T(x - x_C)$.
  - Define a search function in this simplex as $c_i(x) = p(x) - K\, e_i(x)$.
  - Minimize the search function $c_i(x) = p(x) - K\, e_i(x)$ in this simplex.

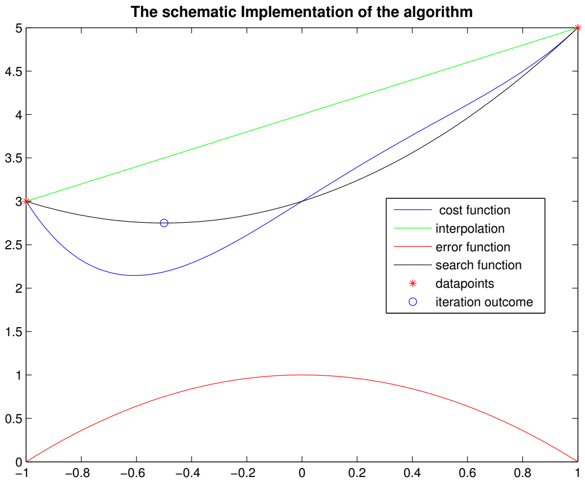
The Global Optimization Algorithm for the problems with linear constraints

- 5. For each simplex S_i
  - Calculate its circumcenter x_C and the circumradius R.
  - Define an error function $e_i(x)$ on each simplex such that $e_i(x) = R^2 - (x - x_C)^T(x - x_C)$.
  - Define a search function in this simplex as $c_i(x) = p(x) - K\, e_i(x)$.
  - Minimize the search function $c_i(x) = p(x) - K\, e_i(x)$ in this simplex.
- 6. Take the minimum of the result of the minimization performed in each simplex and add it to the set of evaluation points.

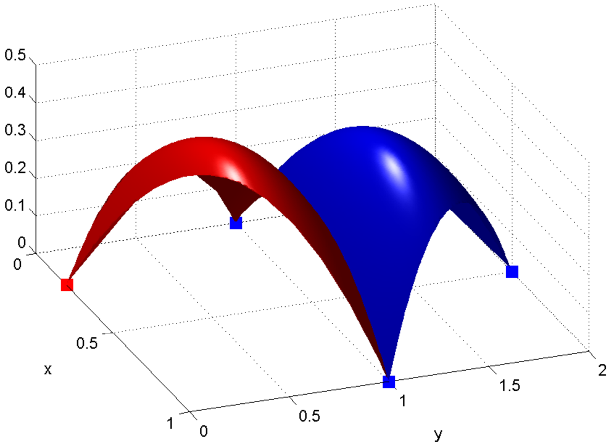
The Global Optimization Algorithm for the problems with linear constraints

- 5. For each simplex S_i
  - Calculate its circumcenter x_C and the circumradius R.
  - Define an error function $e_i(x)$ on each simplex such that $e_i(x) = R^2 - (x - x_C)^T(x - x_C)$.
  - Define a search function in this simplex as $c_i(x) = p(x) - K\, e_i(x)$.
  - Minimize the search function $c_i(x) = p(x) - K\, e_i(x)$ in this simplex.
- 6. Take the minimum of the result of the minimization performed in each simplex and add it to the set of evaluation points.
- 7. Repeat steps 3 to 6 until convergence.

Schematic implementation of the algorithm



Error function plots in 2 dimension



Minimizing the search function

- The search function $p(x) - Ke(x)$ has to be minimized in each simplex.
- A good initial estimate for the value of the minizer of this search function is derived by replacing $p(x)$ with the linear interpolation.
- For interpolation based on the radial basis functions, the gradient and Hessian of the search function is derived analytically; thus, the search function can be minimized by using the Newton method.
- If the linear constraints of the above optimization problems be relaxed with the whole feasible domian; the global minimizer of the search function is not changed.

Convergence Result

- The above algorithm will converge to the global minimum, if there is a K that for all steps of the algorithm, there is a point $\tilde{x}$ which

$$p_n(\tilde{x}) - K\ e_n(\tilde{x}) \leq f(x^*), \tag{1}$$

where $f(x^*)$, $p_n(x)$ and $e_n(x)$ are the global minimum, interpolating function and uncertainty functions at step n respectively.

- The above equation is true; if we have:

$$K \geq \lambda_{\max}(\nabla^2 f(x) - \nabla^2 p_n(x))/2, \tag{2}$$

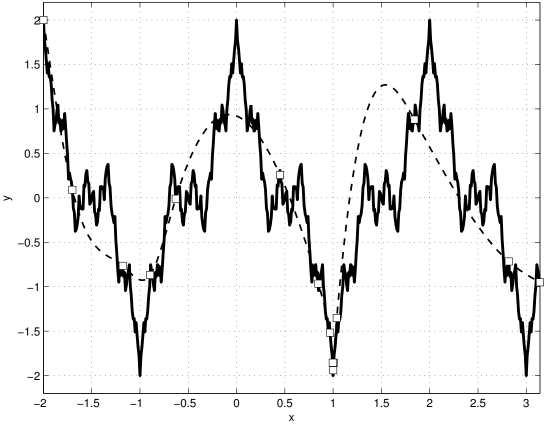
for all steps of the algorithm.

Choose the optimal value for K

- If we have a lower bound for the global minimum (y_0), we could minimize $\frac{p(x)-y_0}{e(x)}$ instead of the above search function.
- If y_0 is the global minimum; the second method is equivalent to the optimal choice for the tuning parameter K.
- This new approach will converge to the global minimum even if the search function is not globally minimized at each step.

Results

$$f(x) = \sum_{i=0}^{N} \frac{1}{2^i} \cos(3^i \pi x), \;\; N = 300;$$



Results

$$f(x,\ y) = x^2 + y^2$$

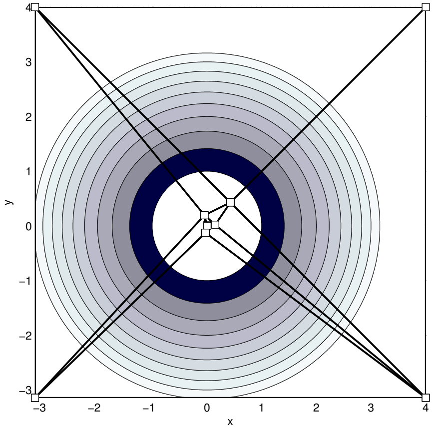


Figure : parabola function

Results

$$f(x,y) = -x sin(\sqrt{|x|}) - y sin(\sqrt{|y|})$$

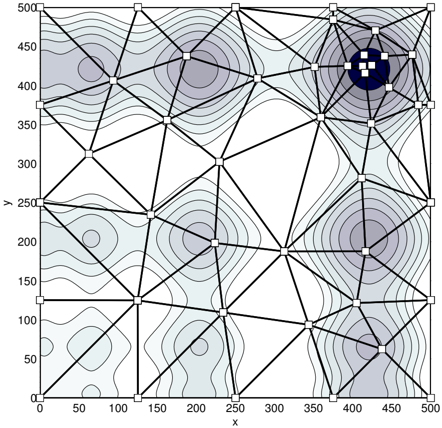


Figure : Schwefel function

Results

$$f(x) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$$

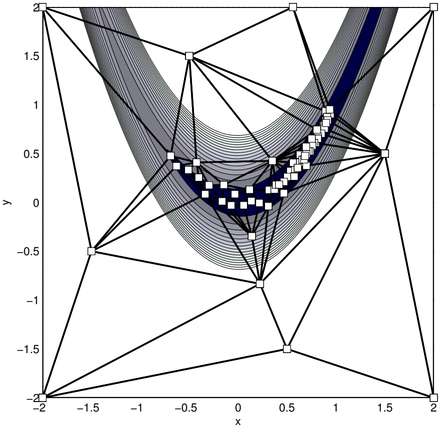


Figure : Rosenbrock function

Perturbed Rosenbrock function

$$f_P(x) = f(x) + \frac{10}{\pi N} sin(N\pi x_1)^2 sin(N\pi x_2)^2 \quad N \to \infty$$

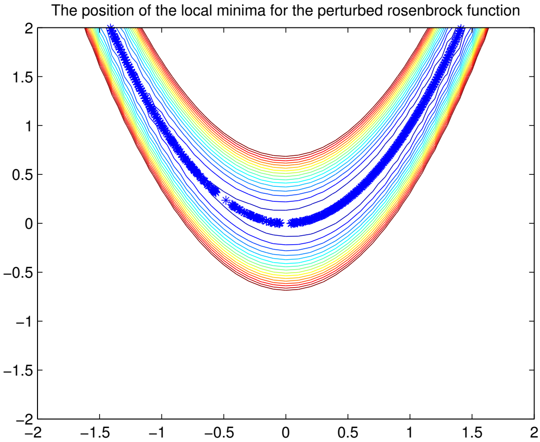


Figure : Rosenbrock function

Generalization of the algorithm for problems with convex constraints

- Above algorithm in restricted to convex hull of the available evaluation points.
- The modification that solves above problems is to project the a search point at each step to the feasible boundary if the search point is out (or on the boundary) of the this convex hull from an interior points.
- We proved the convergence of the above algorithm if the feasible boundary is smooth or the global minizer is an interior point.
- This algorithm is efficient if the feasibility check is a cheap process.
- The new modified algorithms needs $d+1$ initial points instead of 2^d points.

Results

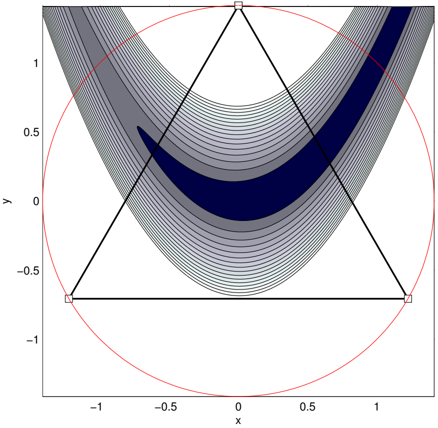


Figure : Rosenbrock function in a circle

Results

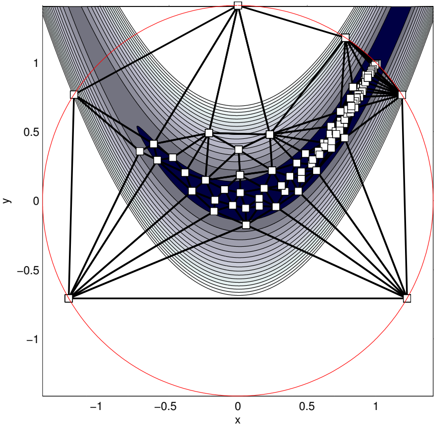


Figure : Rosenbrock function in a circle

Minimizing the long time averaged statistics

The objective function that has been considered is as follow:

$$\min_{x} \lim_{T \to \infty} \frac{1}{T} \int_0^T F(x,t)dt \tag{3}$$

Assumptions

- F(x,t) is the only accessible value that is derived with a simulation or an experiment.
- F(x,t) is a stationary process.
- Above function is a non-convex function.
- The dimension of the design parameters is small.

Construct the model for the problem

The mathematical model that is designed for the above problem is

$$F(x,t) = f(x) + v(x,t) \tag{4}$$

$$v(x,t) = \mathcal{N}(0,\sigma^2) \tag{5}$$

$$g(x,N) = \frac{1}{N}\sum_{i=1}^{N} F(x, i\Delta t) \tag{6}$$

- The computational cost of calculating $g(x,N)$ is proportional to N.
- $g(x,N)$ is an approximate for $f(x)$ that is more accurate as N increased.
- An estimation for the error of $g(x,N)$ has been derived based on the long-memory process theory.

The process of the algorithm for Weierstrass test function

The outcome of the algorithm for Weierstrass test function

Conclusions

- A new optimization algorithm has been developed that found the global minimum with a minimum number of function evaluations.
- Any smooth interpolating function can be used in this algorithm.
- This algorithm is not sensitive to the noisy or inaccurate cost function evaluations.
- The global minimum can be approximate pretty fast, yet the speed of the convergence for the algorithm is slow.
- This method can be combined a local method to develop a fast converging algorithm.
- This algorithm can deal with problems with general convex constraints.
- A new method that uses Δ-Dogs has been developed which minimized the simulated based optimization problems in which the cost function evaluations are derived from infinite-time-average statistics.