Optimization: Methods and Techniques

Yichen ZHANG

University of Waterloo

yichen.zhang@uwaterloo.ca

February 25, 2015

Overview

Trust-Region, Simulated Annealing and Smoothing Techniques

Trust-Region Method

- A trust-region method defines a region around the current point within which the model is trusted to be an adequate representation of the objective function, and then solve the minimization problem of the model in this region.
- Taylor's Theorem underpins most unconstrained minimization methods, and this is certainly true for trust region methods. If f is twice continuously differentiable then at $x_k \in \mathbb{R}^n$

$$f(x_k + s) = f(x_k) + \nabla f_k^T s + \frac{1}{2} s^T \nabla^2 f_k s + o(||s||^2) \tag{1}$$

Algorithm 1 Trust Region Method (TRM)

1. Solve TRS for s_k
2. Adjust Δ_k
3. Update x

The trust region subproblem at x_k is:

$$min\left\{\nabla f_k^T s + \frac{1}{2}s^T \nabla^2 f_k s : ||s|| \leq \Delta_k\right\} \tag{2}$$

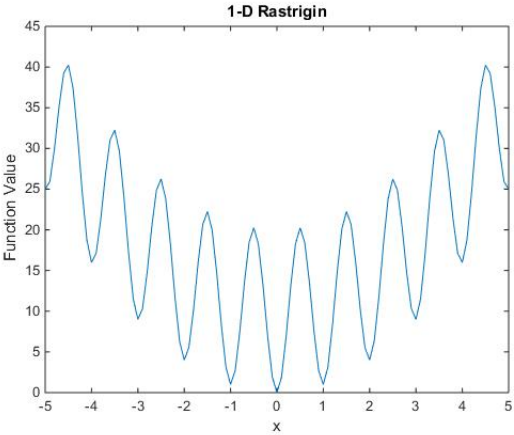


Figure: 1-D Rastrigin. $f = 10 + x^2 - 10\cos(2\pi x)$

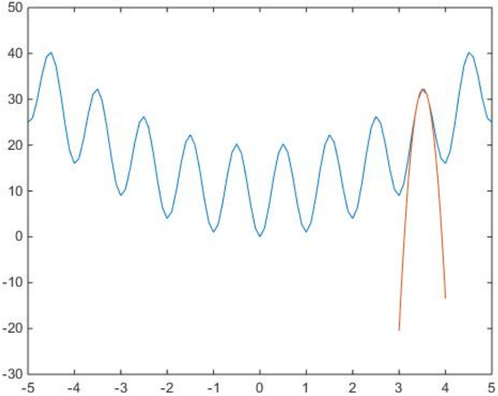


Figure: TRM, First Step, $x_0 = 3.5$, $\Delta = 0.5$

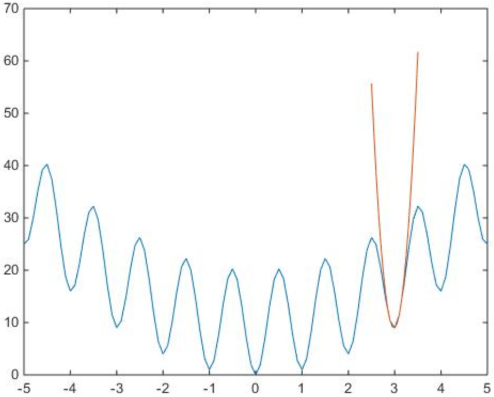


Figure: TRM, Second Step, $x_1 = 3$, $\Delta = 0.5$

Summary

- Trust-Region method can find the (local) optimum in a few steps and a short time.
- It will probably be a local optimum.
- We need the gradient and Hessian.

Simulated Annealing

- Unlike the traditional iteration algorithm which only accept the downhill move, simulated annealing allows perturbation to move uphill in a certain way.
- The advantage to accept uphill move is we may escape from the local optima and find a better answer. Traditional algorithm for solving optimization problem may be trapped in the region near the start point and can not escape from it.

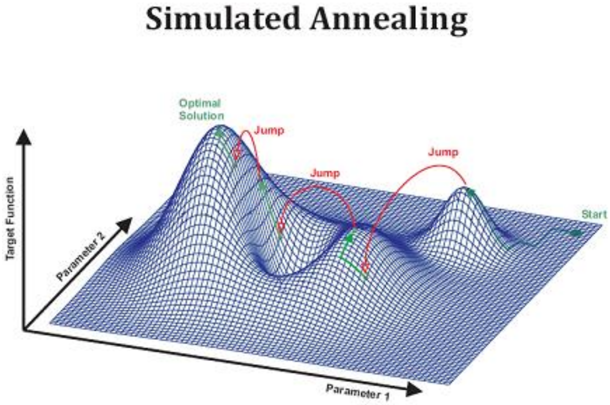


Figure: Simulated Annealing

Algorithm 2 Simulated Annealing at Temperature T

$M =$ number of moves to attempt.
for $m = 1$ to M **do**
  Generate a new neighbouring solution, evaluate f_{new}.
  if $f_{new} < f_{old}$ **then**
    (downhill move: accept it)
    Accept this new solution.
  else
    (uphill move: accept maybe)
    Accept with probability $P(T)$.
  end if
end for

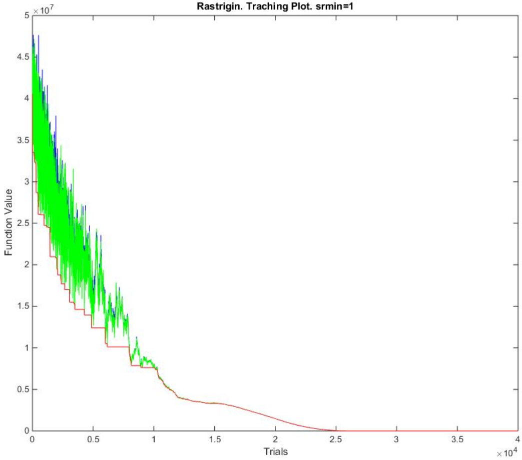


Figure: Simulated Annealing

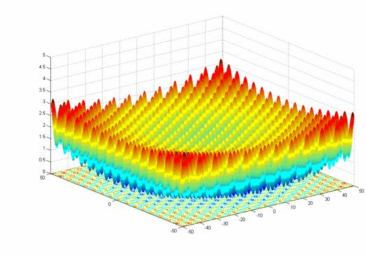
- The simulated annealing procedure is the first phase. After that we use trust-region or some other local search technique with start point(s) from the first phase.

Summary

- Theoretically, simulated annealing is a global optimum search technique and it can find the global optimum in an (infinite) time.
- It does not require any gradient information, just the function value.
- The number of evaluation of $f(x)$ may be very huge.
- All the parameters affect the performance.

Reasons for smoothing

Sometimes the function is very nasty and has so many local optima, so it is very difficult for Trust-Region or Simulated Annealing to find the global one. The smoothing technique can help simulated annealing to find the global optimum more efficiently.



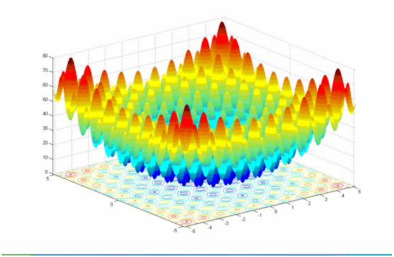


Figure: Griewank Rastrigin

2 Ways for Smoothing

- $\bar{f}(x) = f(x) + \frac{1}{6}\Delta^2 trace(H)$ — Trace-Smooth
- $\bar{f}(x) = f(x) + \lambda||x - x_*||_2^2$ — λ-Smooth

Where H is the Hessian matrix and x_* is the global optimum we guess
Δ and λ are defined by user

Derivation of formula $\bar{f}(x) = f(x) + \frac{1}{6}\Delta^2 trace(H)$

Let f be an objective function and Δ be a positive number. The average value of f over a regular Δ-box $Box(x)$ centred at x with sides $[x_i - \Delta, x_i + \Delta]$ is:

$$\bar{f}(x) = \frac{1}{(2 * \Delta)^n} \int_{Box(x)} f(x) dx_1 ... dx_n \tag{3}$$

The formula above is too expensive to compute when n is large or function f is difficult to compute. However, by approximating f using quadratic Taylor series expansion

$$f(x+s) \cong f(x) + g^T s + \frac{1}{2} s^T H s \equiv q(x) \tag{4}$$

where $g = \nabla f(x), H = \nabla^2 f(x)$, (3) can be approximated as

$$\bar{f}(x) \cong \bar{q}(x) = f(x) + \frac{1}{(2 * \Delta)^n} \int_{\forall i, |s_i| \leq \Delta} (g^T s + \frac{1}{2} s^T H s) ds_1 ... ds_n \tag{5}$$

Continued

Since $g^T s + \frac{1}{2} s^T H s = \sum_i g_i s_i + \frac{1}{2} \sum_i \sum_j s_i s_j h_{ij}$.
Interchanging the order of summation and integration of the above formula yields:

$$\bar{f}(x) = f(x) + \frac{1}{6} \Delta^2 \cdot trace(H) \tag{6}$$

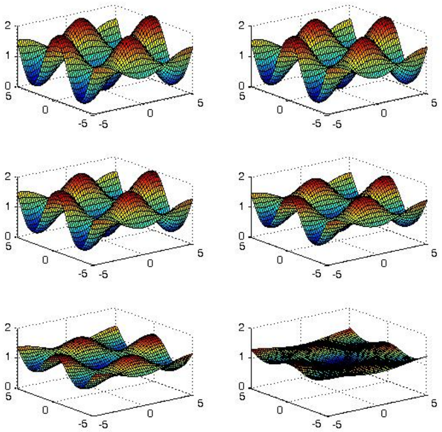


Figure: Griewank. $f(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n}\left(\frac{x_i}{\sqrt{i}}\right) + 1$

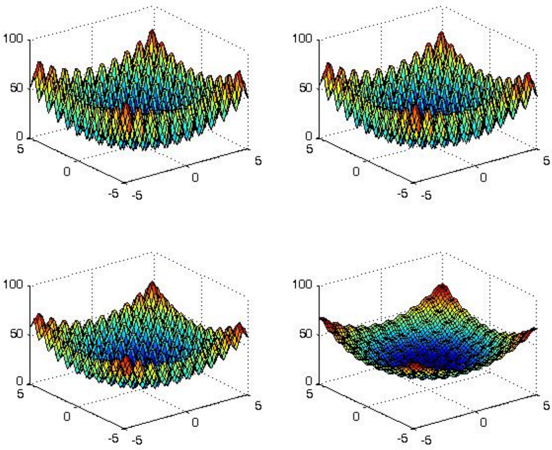


Figure: Rastrigin. $f(x) = 10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i))$

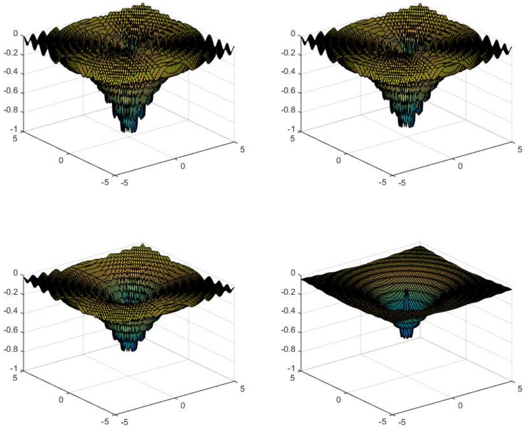
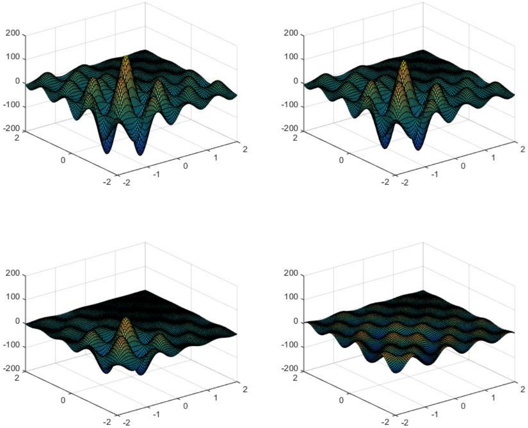


Figure: Drop Wave. $f(x) = -\frac{1+\cos(12\sqrt{x_1^2+x_2^2})}{0.5(x_1^2+x_2^2)+2}$



$$\text{Shubert.} f(x) = (\textstyle\sum_{i=1}^{5} i\cos((i+1)x_1+i))(\sum_{i=1}^{5} i\cos((i+1)x_2+i))$$

Simulated Annealing Combined with Smoothing

- We have Δ-sequence or λ-sequence which contains several elements and the last one is 0.
- For each element of the smooth sequence, we run through all the temperatures using simulated annealing

Trust-Region Combined with Simulated Annealing and Smoothing

- Traditional trust region only accept a point when the new point's value is less than current point. The main idea of trust-region combined with simulated annealing is that we accept a point when it decreases the function value *OR* it increases the function value with a probability.
- Also, we can combine the Trust-Region with the smoothing technique

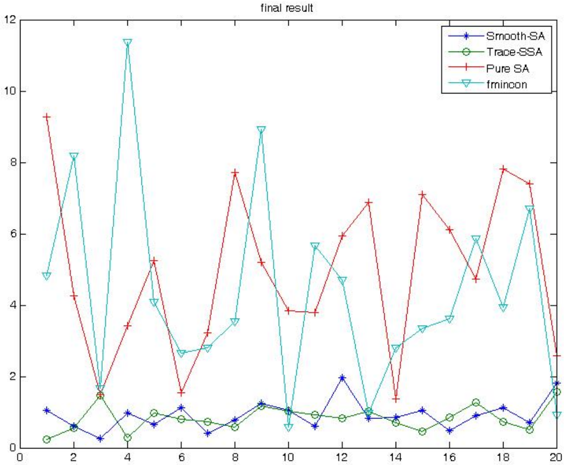


Figure: Griewank $n = 10$

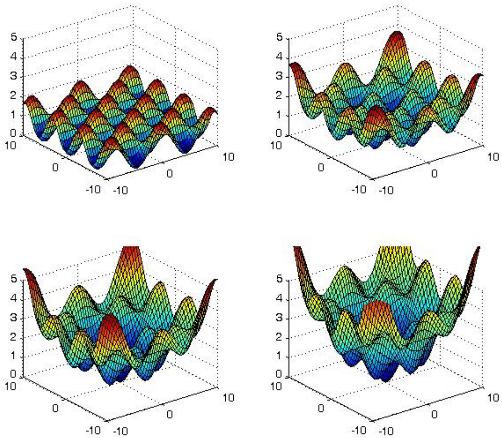
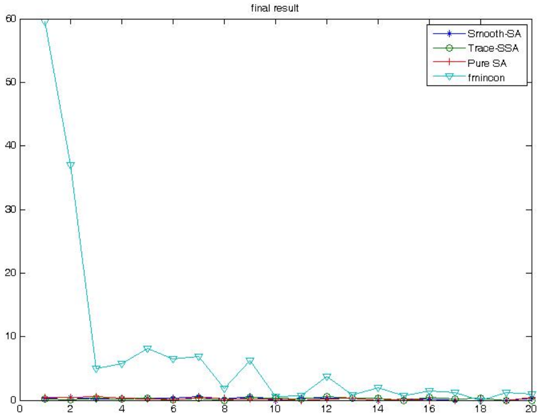


Figure: Griewank in different shape



As n goes bigger, the function become more and more flattened

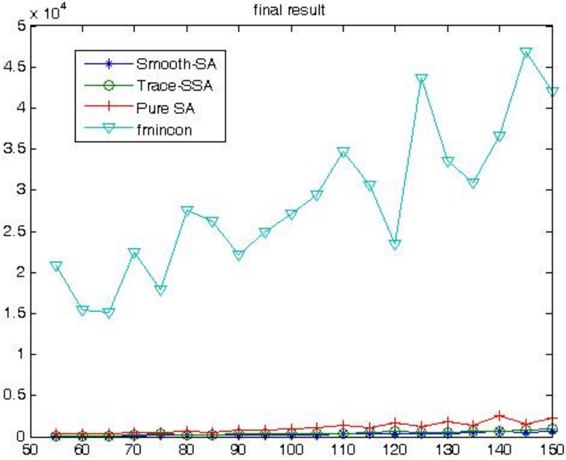


Figure: Rastrigin. Compared with fmincon

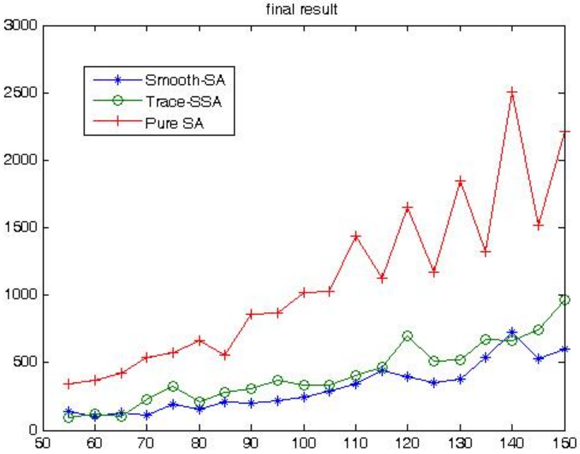


Figure: Rastrigin. Remove fmincon

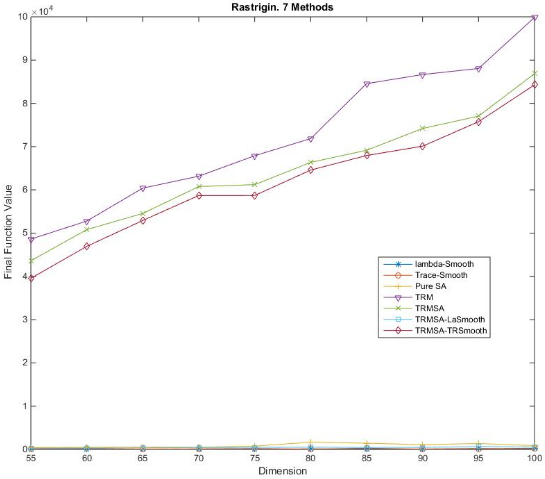


Figure: Rastrigin. 7 Methods

For $Trace - Smooth\ \bar{f}(x) = f(x) + \frac{1}{6}\Delta^2 trace(H)$, since $trace(H) = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}$. So we have:

$$\bar{g} = g + \frac{1}{6}\Delta^2 \frac{\partial^3 f}{\partial x^3}, \qquad \bar{H} = H + \frac{1}{6}\Delta^2 diag(\frac{\partial^4 f}{\partial x^4}) \tag{7}$$

Where g and H is the gradient and Hessian matrix of original function $f(x)$ and

$$\frac{\partial^3 f}{\partial x^3} = \begin{bmatrix} \frac{\partial^3 f}{\partial x_1^3} \\ \frac{\partial^3 f}{\partial x_2^3} \\ \cdots \\ \frac{\partial^3 f}{\partial x_n^3} \end{bmatrix}, diag(\frac{\partial^4 f}{\partial x^4}) = \begin{bmatrix} \frac{\partial^4 f}{\partial x_1^4} & 0 & \cdots & 0 \\ 0 & \frac{\partial^4 f}{\partial x_2^4} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\partial^4 f}{\partial x_n^4} \end{bmatrix} \tag{8}$$

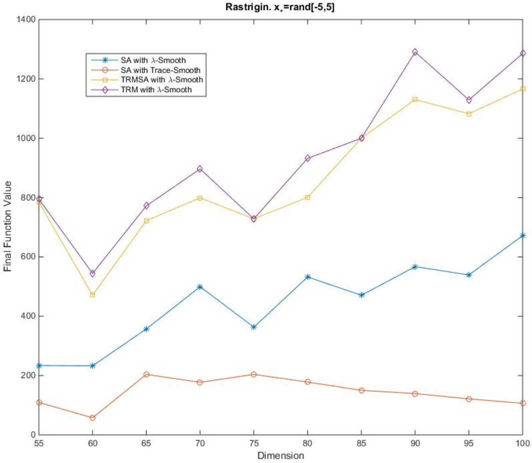


Figure: Rastrigin. 4 Modified Methods

Number of Evaluation of Function Value and Hessian Matrix

Dim	SAwithLa	SAwithTr	TRMSAwithLa	TRMwithLa
55	80412 0	80414 80412	3576 3575	627 626
60	80415 0	80412 80410	3152 3151	1116 1115
65	80412 0	80413 80411	3149 3148	1597 1596
70	80414 0	80418 80416	3528 3527	647 646
75	80416 0	80415 80413	3079 3078	1110 1109
80	80415 0	80416 80414	4533 4532	610 609
85	80413 0	80413 80411	3189 3188	679 678
90	80416 0	80415 80413	3522 3521	664 663
95	80418 0	80415 80413	3565 3564	671 670
100	80418 0	80417 80415	3287 3286	1605 1604

Derivative Free Optimization

Motivation

- In some cases, we can not get the derivative information and it takes very long to evaluate the function value. So it's impossible to use finite difference to evaluate the first and second derivative.
- Since the function is very expensive to compute, it is not ideal to use simulated annealing to optimize.

Main Idea of DFO

- We have a bunch of points and their function value to start. Use these points to build a model to approximate the original function, use this model to solve the optimization problem. We can update the model while solving the problem once we have more information about the function.
- After we have the model, we can use trust-region or other methods to solve the problem.

Choice of Model

- There are many types of model we can choose to approximate the original function. We test two groups of them: Lagrange Polynomial Interpolation(LPI) and Radial Basis Function(RBF).

Lagrange Polynomial Interpolation

- Linear Model:

$$L(x) = \sum_{i=1}^{n} a_i x_i + c \tag{9}$$

- Quadratic Model:

$$L(x) = \sum_{i=1}^{n} \sum_{j=i}^{n} b_{ij} x_i x_j + \sum_{i=1}^{n} a_i x_i + c \tag{10}$$

Lagrange Polynomial Interpolation

- We use m points: $x_1, x_2, ..., x_m$ and $f_1, f_2, ..., f_m$ to build the model $L(x)$.
- $L(x)$ should satisfy $L(x_i) = f_i,\ i = 1, ..., m$.
- In order to make the model unique, we need $n + 1$ points to build the linear model and $\frac{(n+1)n}{2} + 1$ points to build the quadratic model.

Radial Basis Function

- The model has the form:

$$R_m(x) = \sum_{i=1}^{m} \lambda_i \phi(||x - x_i||) + p(x) \tag{11}$$

- And we have different choices for $\phi(r)$ and $p(x)$:

RBF	$\phi(r) > 0$	$p(x)$
cubic	r^3	$b^T \cdot x + a$
linear	r	a
multiquadric	$(r^2 + \gamma^2)^{\frac{1}{2}}, \gamma > 0$	a
Gaussian	$exp(-\gamma r^2), \gamma > 0$	$\{0\}$

To get the parameters λ_i, b, a we need to solve the linear equations:

$$\begin{pmatrix} \Phi & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix} \tag{12}$$

where Φ is the $m \times m$ matrix with $\Phi_{ij} = \phi(||x_i - x_j||)$ and

$$P = \begin{pmatrix} x_1^T & 1 \\ x_2^T & 1 \\ \vdots & \vdots \\ x_m^T & 1 \end{pmatrix}, \lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{pmatrix}, c = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \\ a \end{pmatrix}, F = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_m) \end{pmatrix}. \tag{13}$$

Radial Basis Function

- Notice that if we use a linear model for $p(x)$ and $rank(P) = n + 1$, then the system has a unique solution.
- Otherwise if we just use $p(x) = a$ or $p(x) = 0$, then the system has a unique solution no matter how many points we have.
- Also, it has different ϕ to choose.

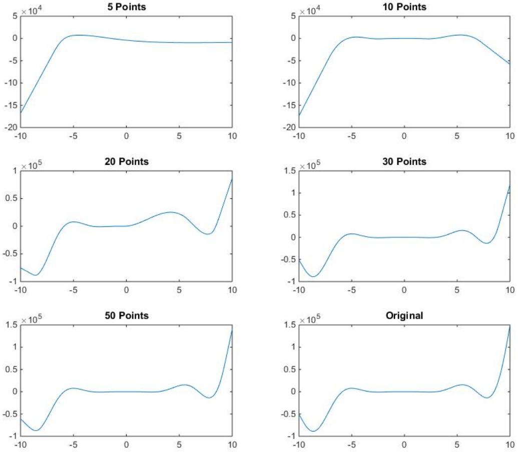
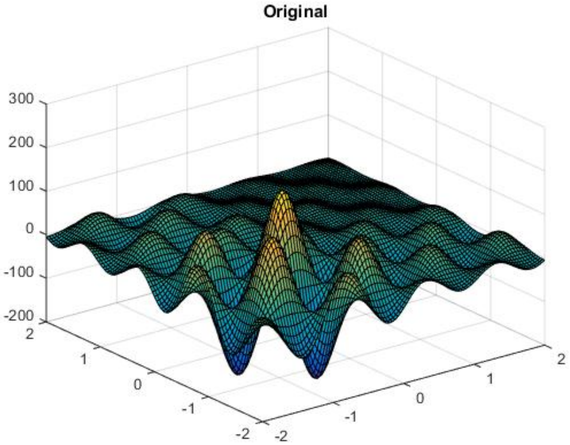
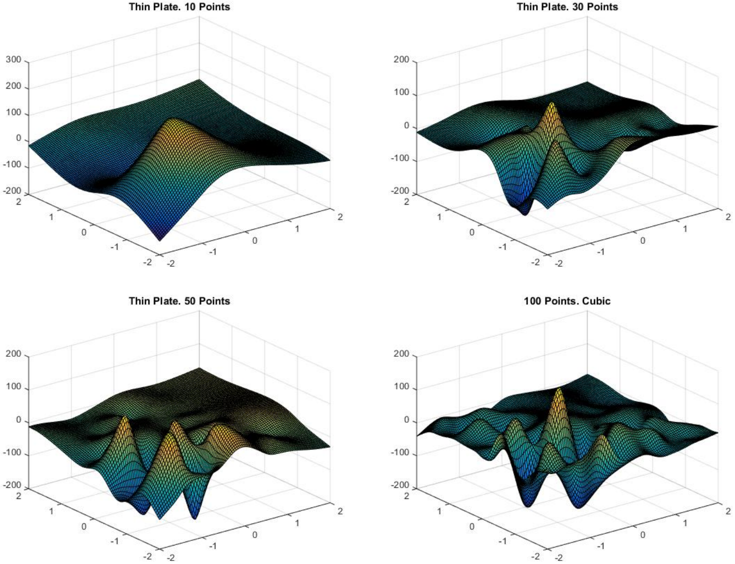


Figure: Radial Basis Function Approximation



$$\text{Shubert}.f(x) = \left(\sum_{i=1}^{5} i\cos((i+1)x_1+i)\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_2+i)\right)$$


Thin Plate. 10 Points
300
200
100
0
-100
-200
2 1 0 -1 -2
-2 -1 0 1 2
Thin Plate. 30 Points
200
100
0
-100
-200
2 1 0 -1 -2
-2 -1 0 1 2
Thin Plate. 50 Points
200
100
0
-100
-200
2 1 0 -1 -2
-2 -1 0 1 2
100 Points. Cubic
200
100
0
-100
-200
2 1 0 -1 -2
-2 -1 0 1 2

Radial Basis Function

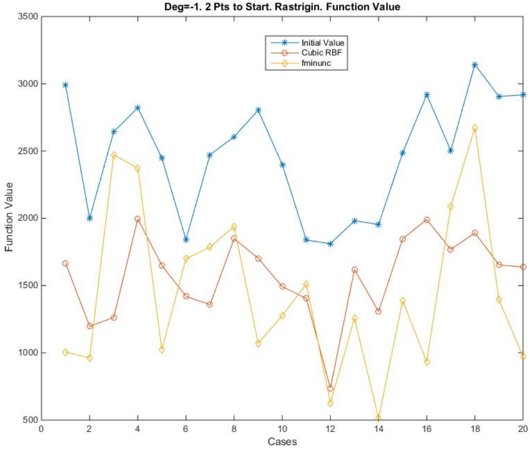
- Once we have built the model $R_m(x)$, we can easily compute the derivative:

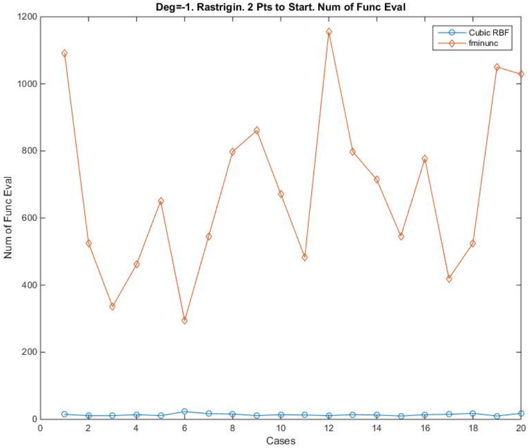
$$R'_m(x) = \sum_{i=1}^{m} \lambda_i \phi'(||z_i||) \frac{z_i}{||z_i||} + p'(x) \tag{14}$$

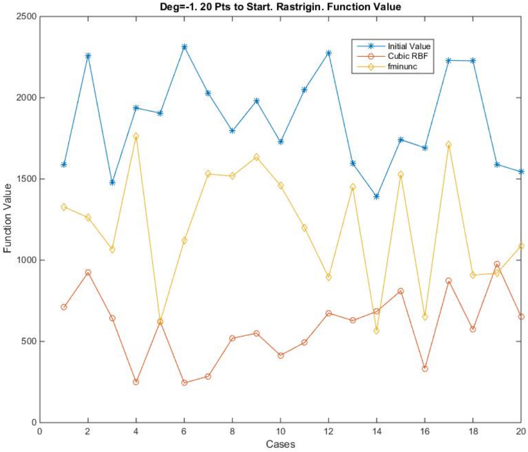
and

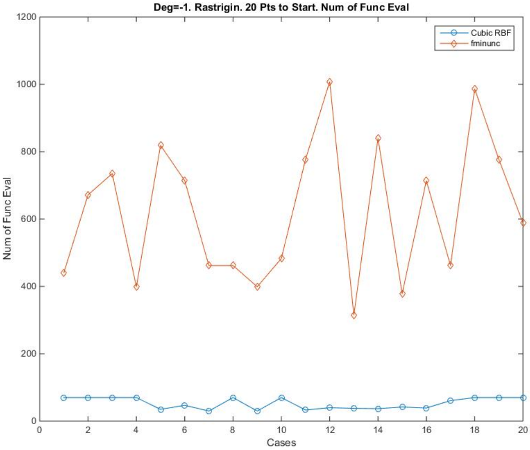
$$R''_m(x) = \sum_{i=1}^{m} \lambda_i \left[\frac{\phi'(||z_i||)}{||z_i||} I_n + \{\phi''(||z_i||) - \frac{\phi'(||z_i||)}{||z_i||}\} \frac{(z_i)(z_i)^T}{||z_i||^2} \right] \tag{15}$$

$z_i = x - x_i$


Deg=-1. 2 Pts to Start. Rastrigin. Function Value
Initial Value
Cubic RBF
fminunc
Function Value
Cases
3500
3000
2500
2000
1500
1000
500
0
2
4
6
8
10
12
14
16
18
20


Deg=-1. Rastrigin. 2 Pts to Start. Num of Func Eval
Cubic RBF
fminunc
Num of Func Eval
Cases
0
200
400
600
800
1000
1200
0
2
4
6
8
10
12
14
16
18
20


Deg=-1. 20 Pts to Start. Rastrigin. Function Value
Initial Value
Cubic RBF
fminunc
Function Value
Cases
2500
2000
1500
1000
500
0
0
2
4
6
8
10
12
14
16
18
20


Deg=-1. Rastrigin. 20 Pts to Start. Num of Func Eval
Cubic RBF
fminunc
Num of Func Eval
Cases
0
200
400
600
800
1000
1200
2
4
6
8
10
12
14
16
18
20

Summary

TRM, SA, DFO and Smoothing

- Each method has its own advantages and disadvantages.
- Our new methods work more efficiently than the traditional ways.

Challenge

- Δ for Trace-Smoothing is really important for the method, but it's a little difficult to find it perfect cause it depends on the problem.
- The λ smooth technique can work well in the condition that we know where the global optimum locates.
- If the search region is quite big, the RBF may not approximate the original function very well just use several points.

Further work

- Need more examples and test problems.
- How to decide the parameters and the models.

Reference

- http://www.sfu.ca/~ssurjano/optimization.html
- http://www.mcs.anl.gov/~wild/orbit/
- https://courses.cit.cornell.edu/jmueller/
- ORBIT: Optimization by Radial Basis Function Interpolation in Trust-Regions
- Introduction to Derivative-Free Optimization

Thanks