Solving Noisy Optimization Problems by a Quadratic model

D. Juárez-Romero, A. Mosso, Marco Antonio Cruz-Chávez, L.L. Castro-Gómez, G. Flores

Cento de Investigación en Ingeniería y Ciencias Applicadas, Av. Universidad 1000, C.P. 62209, MEXICO e-mail:djuarezr7@gmail.com

Abstract. We propose a set of safeguards to solve optimization problems with Derivative-Free optimization when the function has noise. The safeguards are related to the construction of a quadratic model, and the tests of acceptance of the step size. These safeguards consider noise in the measurements, thus as a threshold tolerance criteria to test the model approximation, and the descendent direction, this threshold considers the value of the standard deviation was used. Also some indexes to monitor the iteration progress are suggested. The results are exemplified with a 2 variable application. These results show the improvement in robustness of the method using a quadratic model.

1 INTRODUCTION

To solve a continuous optimization pro em, the derivatives required in the Taylor approximation can be obtained numerically, analytically, or by automatic differentiation [Griewank91] but when the function is not continuous, the evaluation of first and second derivatives causes noise. In some applications there is inaccuracy in the function evaluation. If a model is not exact, the evaluation of derivatives worsen. Some types of these applications are:

1. The functions that use nested iterative methods.
2. The functions with smooth and non-smooth parts.
3. The functions which imply noisy measurements
4. The selected norm $||\cdot||$ required in the Objective functions presents a non smooth derivative, for example $max(f(x)), abs(f(x)), min\, max f(x)$

In many applications the objective function has the form $f(x) = h(x) + r(x)$ where $f(x)$ is a nonlinear function $r(x)$ has a different value in every evaluation, or has a random value.

Suitable methods include the simplex-reflection method of Nelder and Mead, pattern-search methods, conjugate method, and evolutive algorithms[Krink04]. Other types of methods are direct methods [AndeFerri00], and methods which construct a local (linear or quadratic) model.

when the derivative is obtained by finite differences

$$\triangledown f(x) = \frac{f(x + \delta x_i) - f(x - \delta x_i)}{2\delta x_i}$$

M.A Cruz-Chávez, J.C. Zavala-Díaz (Eds): CICos 2008, ISBN:978-607-00-0165-9. pp. 45-51, 2008

if the noise dominates in the difference interval $[-\delta x, +\delta x]$, the evaluation of $\bigtriangledown f(x)$ presents little accuracy.

2 MODEL BASED METHODS

Most of the methods compute steps by minimizing a quadratic model of the objective function, $f(x)$. When derivatives are not available, we may construct a quadratic model of the objective function that interpolates $f(x)$ at a set of appropriately chosen sample points. Since such a model is usually non-convex, model-based methods use a trust region to compute the step [Nocedal00].

Suppose that at current iterate x_k we have a table of sample points x^l, y^l. We assume that x^k is an element of this set and that no point in this set has a lower function value than x^k. We wish to construct a quadratic model of the form

$$m_k(x_k + \delta x) = c + g^T \delta x + \frac{1}{2}\delta x^T G \delta x.$$

where

$$\delta x^l = x^l - x^0$$

We evaluate the scalar c, the vector g, and the symmetric matrix G by imposing the interpolation conditions

$m_k(x^l) = f(x^l), \qquad l = 1, 2, .., q$ interpolation

$G_{ij} = G_{ji}$ symmetry

Since there are $q = \frac{1}{2}(n+1)(n+2)$ coefficients in this model the interpolation conditions determine m_k uniquely when q points are used

Once m_k has been formed, we compute a step Δx by approximately solving the trust region problem.

$$\min_{\Delta x} m_k(x_k + \Delta x)$$

$$subject \quad to ||s\Delta x|| \leq D$$

where s is a scaling vector.

When the new point is obtained $x = x^0 + \Delta x$, then the worst point is replaced

$$m_{max} l = max(y^l)$$

The coordinates and the value of this point are replaced by (x^0, y^0)

2.1 Improvements

In the development of the model we assume that function $f(x)$ can be described by a noisy quadratic model where the noise is small compared with the function value.

Recursive Least Squares Since the information contained in the g vector and G matrix is not fully reliable, to assure convergence we use several safeguards:

0. We can use a a larger number of points to reduce the effect of noise. Thus a linear model can be constructed with $p > q$ points to fit n parameters, θ as $X^T X\theta = X^T y$. The sample points were selected randomly.

Also for new observations $w^T\theta = \overline{y}$, the updated value of the parameters can be obtained using recursive least squares [Bjorck96] :

$$(A^T A + ww^T)(\overline{\theta} - \theta) = (\overline{y} - w^T\theta)w$$

Evaluation of Trust Region When the constrains are imposed on the optimization, the equations produced by the Lagrage multipliers are:

$$\frac{\partial L}{\partial \Delta x} = g + \Delta x^T G + \lambda s \Delta x^T I = 0$$

and for the Eucledian norm:

$$\frac{\partial L}{\partial \lambda} = (s\Delta x)^2 - D = 0$$

for nonzero values of λ (when the constraint is active) the solution for Δx is

$$\Delta x = (G + 2\lambda s^2 I)^{-1} g$$

1. We test the eigenvalues of the matrix G

$$eig(H) = \lambda_i > 0 \forall i$$

2. We test the curvature

$$\Delta x^T G \Delta x < 0$$

3. We test the prediction capability of the model, which now considers noise

$$\rho^- = \frac{f(x_k) - f(x_k - \Delta x) - \tau}{m_k(x_k) - m_k(x_k + \Delta x)}$$

$$\rho^+ = \frac{f(x_k) - f(x_k + \Delta x) + \tau}{m_k(x_k) - m_k(x_k + \Delta x)}$$

also $\rho_{Small} = min(\rho^+, \rho^-)$, $\rho_{Big} = max(\rho^+, \rho^-)$

if $\rho_{min} < \rho_{small}$, and $\rho_{Big} < \rho_{max}$ the model approximation is accepted otherwise the Δx is truncated.

τ is a threshold value which depends on the standard deviation of the model, $\sigma(x)$, this variable can be approximated from analysis of the quality of the correlation used:

$$\sigma^2(x) = \frac{y^T y - m(x)^T m(x)}{p - n}$$

4. We also test the descent condition.

$$g^T \delta x + \frac{1}{2} \delta x^T G \delta x < 0$$

The descent direction is maintained, except for the prescence of noise

$$f(x + \Delta x) \leq f(x) + \tau$$

6. As the iteration progresses, the neighboring nodes are closer, but the overall knowledge of the function is poorer. Thus as a updating strategy for the replaced node x_l, considers to retain some knowledge about the previous points.

$$x^l = x^0 \phi + x_l * (1. - \phi)$$

here $0 \leq \phi \leq 1$ Then $y^l(x^l)$ is evaluated

parameters of the quadratic model: $\rho_{max} = 0.7 \rho_{max} = 1.2$ $\Delta = 0.1$ $\phi = 0.9$ $X^0 = [0.953.86]$

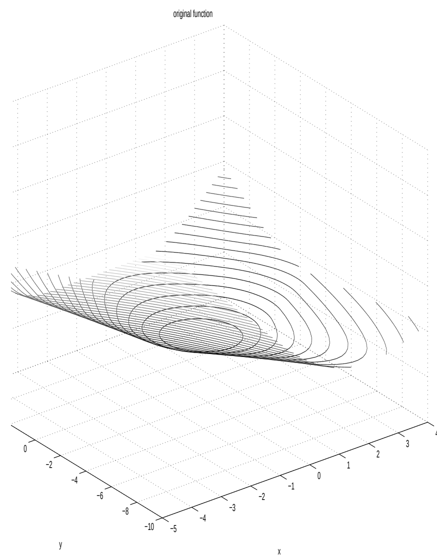

Fig. 1. Surface of the optimization

3 RESULTS

We present the results for the test function:

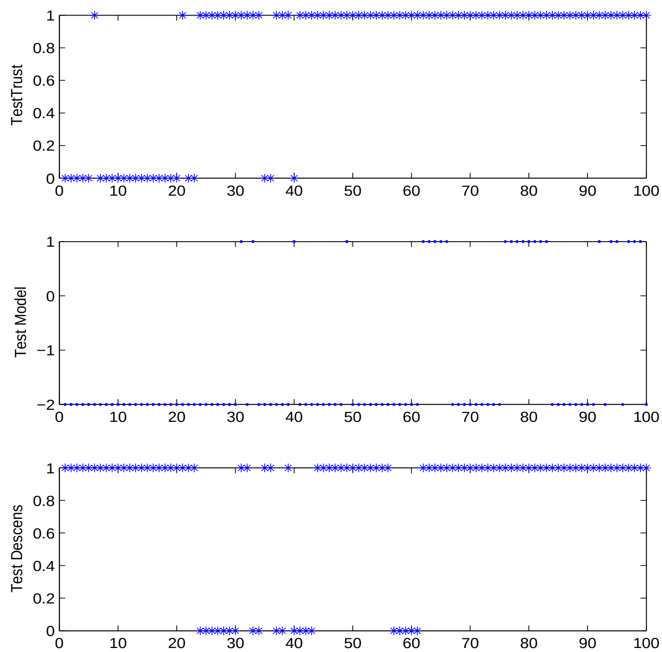

Fig. 2. Surface response of the test function

$$\Phi_0(\xi) = \begin{cases} 1-\xi & \text{if } \xi < (1-\beta) \\ \xi - 1 & \text{if } \xi > (1+\beta) \\ 0.5\frac{(\xi-1)(\xi-1)}{\beta} + 0.5 * \beta & \text{otherwise} \end{cases}$$

also

$$\Phi_N(\xi) = \frac{2.(1. - \beta) sin(\alpha 2\pi\xi)}{\alpha * \pi}$$

then

$$\Phi(\xi) = \Phi_0(\xi) + \Phi_N(\xi);$$

For two variables $\Phi(x_1, x_2) = \Phi(x_1) + \omega_{x2}\Phi(x_2) + \omega_{x12}\Phi(x_1) * \Phi(x_2)$. The contour of this function appears in figure 1.

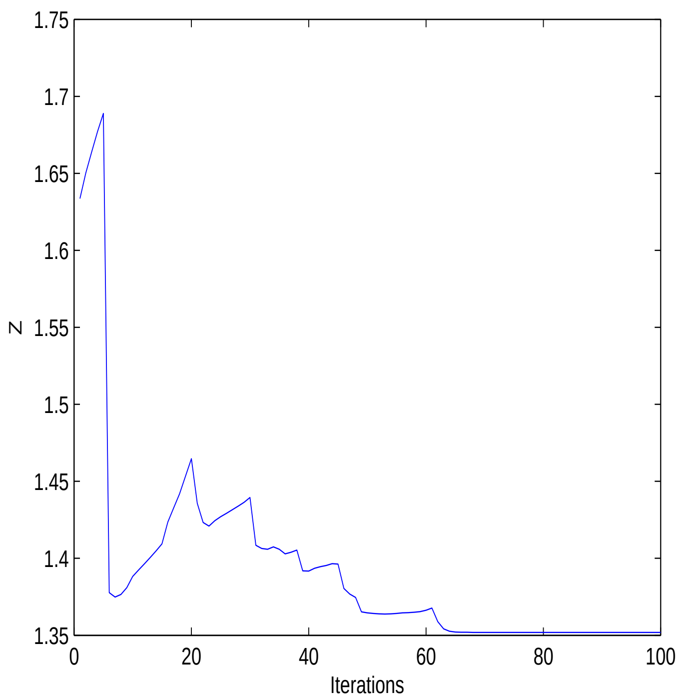

Fig. 3. Function value vs. iterations

Figure 2 shows the monitoring variables. The index of Trust region (upper plot) shows that initially the trust criteria was not satisfied, thus the trust region iteratio was carried out. The index of model approximation (middle plot) indicates that most of the iterations the model approximation was inaccurate thus the step was truncated twice every iteration. Without this reduction the descent condition could not be achieved. The index of descent indicates that in most of the iterations the descent direction was obtained.

Figure 3 shows that initially the reduction is fast, but as the iteration point has decreased its value, the noise dominates and the progress becomes slow and oscillatory. However the safeguards used assure the convergence close to the optimum in about 60 iterations. After that there is little improvement. If Matlab function `fminunc` was used the effectiveness depends on the starting value.

$q = 18$ Data points were selected randomly.

4 CONCLUSIONS

A derivative-free derivative method was build to solve noisy functions. The method consider several type of monitors and safeguards to improve convergence. This model considers the standard deviation of the model to test the iteration progress. From initial set of coordinates and function values is constructed this model and the standard deviation. As the iteration progress the worst values are replaced, then the model becomes a local approximation when the number of iterations is grater than the number of samples, thus more sensitive to noise, thus it is recommended to re-sample.

- Quadratic model requires suitable starting points. The quality of the solution depends on the and criteria used to construct the model, and the positions of the starting points. As an alternative we use random sampling.
- A set of testing step have been used to improve convergence of this method. The advantage of this set of test is that some of them can be exchanged to improve the robustness of the methodology.
- We have used some indices to diagnose progress of the solution

References

[Nocedal00] Nocedal J., S. J. Wright "Numerical Optimization ", Springer Series in Operations research. Springer-Verlag (2000)

[Griewank91] Griewank A. and G. F. Corliss "Automatic differentiation of algorithms: theory, implementation, and application" SIAM (1991).

[Krink04] Krink , T., B. Filipie, G.B. Fogel "Noisy Optimization Problems - A Particular challenge for Differential Evolution (2004) ,Sixth Congress on Evolutionary Computation. CEC-2004. IEEE Press, Piscataway NJ: pp. 332-339.

[Bjorck96] Björck A. "Numerical Methods for Least Squares Problems", SIAM (1996)

[AndeFerri00] Anderson E. J. , Michael C Ferris, "A Direct Search Algorithm for Optimization with Noisy Function Evaluations" SIAM J. on Optimization, Vol. 11, **No. 3**. (2000), pp. 837-857.