UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

School of Computer Science and Applied Mathematics

WITS UNIVERSITY

Optimisation II
Assignment

5 September 2016

Instructions

- Read all the instructions carefully.
- The assignment is broken down into four mark blocks which can be found on the last page.
- Bonus marks will be awarded for implementing something additional so long as it is merit worthy.
- It is strictly individual work, although you can discuss the algorithm with your peers to ensure that you understand it.
- Specifics of the Hand-in can be found under the assignment section.
- It should not take too long, an average implementation should be around 50 to 100 lines of code.
- Any dubious looking code not expected to produce “claimed" results will be heavily penalised.

SECTION 1

Downhill Simplex Method of Nelder and Mead

A direct search method for the unconstrained optimisation problem is the Downhill simplex method developed by Nelder and Mead (1965). It does not make an assumption on the cost function to minimise. Importantly, the function in question does not need to satisfy any condition of differentiability unlike other methods. It makes use of simplices, or polytopes in given dimension $n+1$. For example, in 2 dimensions, the simplex is a polytope of 3 vertices (triangle). In 3 dimensional space it forms a tetrahedron.

The method starts from an initial simplex. Subsequent steps of the method consist of updating the simplex where it defines:

$\bar{x}^h$ is the vertex with highest function value,

$\bar{x}^s$ is the vertex with second highest function value,

$\bar{x}^l$ is the vertex with lowest function value,

$\bar{G}$ is the centroid of all the vertices except $\bar{x}^h$, ie. the centroid of n points out of $n+1$:

$$\bar{G} = \frac{1}{n} \sum_{j=1, j\neq h}^{n+1} \bar{x}^j \tag{1}$$

Let $y = f(\bar{x})$ and $y^h = f(\bar{x}^h)$ then the algorithm suggested by Nelder and Mead is as follows:

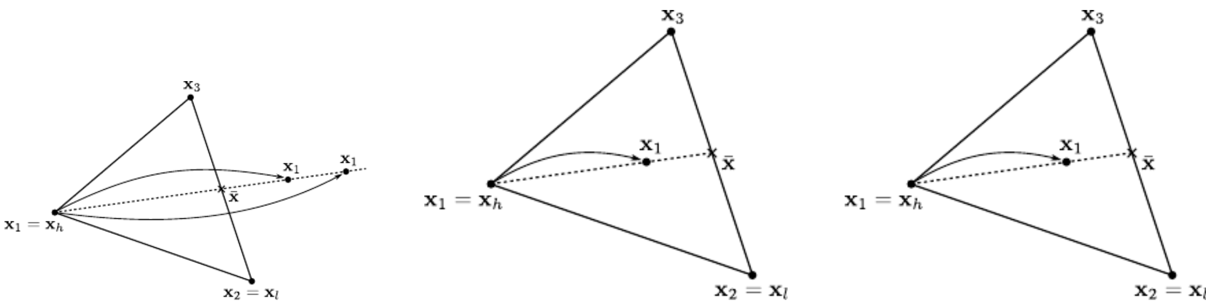
Data: Input vertices of initial simplex and evaluated $f(\bar{x})$
Result: The $\bar{x}^*$, i.e. the local minimum of function f
while STOP-CRIT == FALSE **and** $k < k_{max}$ **do**
    Reflection: Reflect $\bar{x}^h$ using a reflection factor of $\alpha > 0$, i.e. find $\bar{x}'$;
    $\bar{x}' = (1+\alpha)\bar{G} - \alpha\bar{x}^h$;
    if $y^l \leq y' \leq y^s$ **then**
        Replace $\bar{x}^h$ by $\bar{x}'$;
        $\bar{x}' = (1+\alpha)\bar{G} - \alpha\bar{x}^h$;
    end
    if $y' < y^l$ **then**
        Expand the simplex, i.e. find x'';
        $\bar{x''} = \gamma\bar{x}' + (1-\gamma)\bar{G}$;
        if $y'' < y^l$ **then**
            Replace $\bar{x}^h$ with $\bar{x}''$;
        else if $y'' \geq y^l$ **then**
            Replace $\bar{x}^h$ by $\bar{x}'$;
    else if $y' > y^s$ **then**
        Contract the simplex by contraction factor;
        if $y' < y^h$ **then**
            Find $\bar{x}''$ such that;
            $\bar{x}'' = \beta\bar{x}' + (1-\beta)\bar{G}$;
        else if $y' \geq y^h$ **then**
            Find $\bar{x}''$ such that;
            $\bar{x}'' = \beta\bar{x}^h + (1-\beta)\bar{G}$;
        if $y'' < y^h$ **and** $y'' < y'$ **then**
            Replace $\bar{x}^h$ by $\bar{x}''$;
        end
        if $y'' \geq y^h$ **and** $y'' > y'$ **then**
            Reduce the size of the simplex by halving the distances of $\bar{x}^l$;
        end
    $k = k+1$;
end
Return $\bar{x}^* = \bar{x}^l$

Algorithm 1: Downhill simplex of Nelder and Mead (1965)

The typical values for the above factors are $\alpha = 1$, $\gamma = 2$ and $\beta = 0.5$. The stopping critera to use is definied by:

$$\sqrt{\frac{1}{n+1}\sum_{i=0}^{n}\left(f(x_i) - \overline{f(x_i)}\right)^2} \leq \epsilon \tag{2}$$

To help you understand the above algorithm Figure 1 is included. A common practice to generate the



(a) Relfection and expansion (b) Contraction (c) Multiple contraction

Figure 1: Illustration of operations in two dimensions

initial remaining simplex vertices is to make use of $\bar{x}_0 + \bar{e}_i b$, where $\bar{e}_i$ is the unit vector in the direction of the x_i coordinate and b an edge length. Assume a value of 0.1 for b.

SECTION 2

Assignment

Using MATLAB, write a program that implements the Downhill simplex method. You must then use this on the Rosenbrock function:

$$f(\bar{x}) = (1 - x_1)^2 + 10(x_2 - x_1^2)^2, \tag{3}$$

Run your code on following starting points: `xt = [-1.20, 1.00; -0.23, 1.26; -0.94, 1.97]`. That is, each row of the matrix is an initial starting point.

As a result you should obtain the following information for each case:

- The coordinates of the minimum.
- The function value at the minimum.
- The number of iterations required for the algorithm in the given case.

You should also produce two plots per starting point. Figure 2 illustrates the plots required for ONE arbitrary starting point. Once you have completed the assignment, use the publish button in the editor of MATLAB and write your assignment to PDF. This is to be submitted on SAKAI under the *Assignment 2* tab.

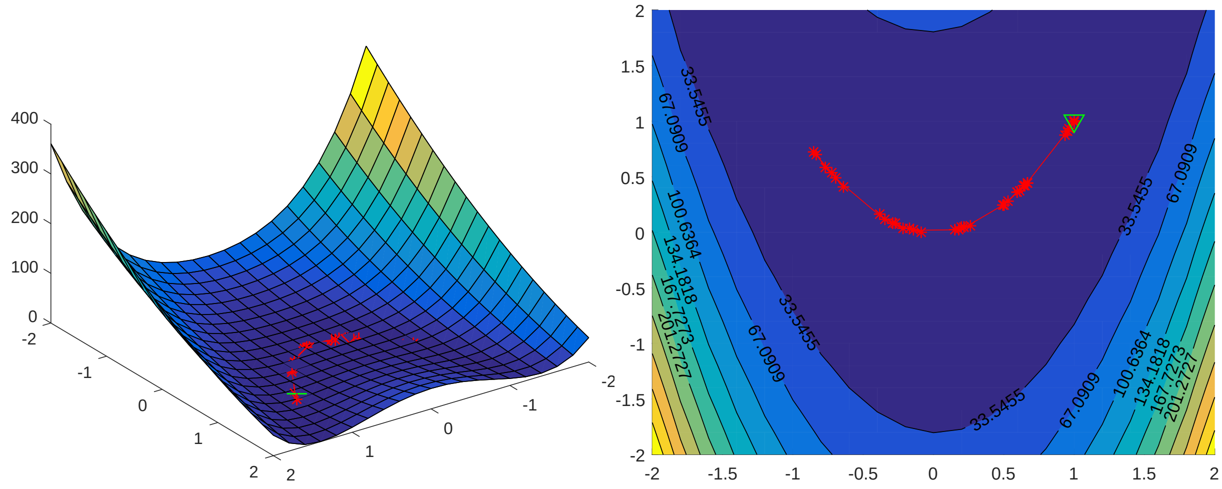


Figure 2: A Expected Sample Output

The mark breakdown for the assignment is the following:

30% For code that runs and implements the method - no figures produced.

30% Correct answers (values, iterations etc.) for each starting point - 10% for each case .

20% Correct 3D and contour plot for the function.

20% The tracking of the best vertex per iteration imposed on both plots for each case. See Figure 2.

Note: No MATLAB published files will be marked, this will lead to a zero mark. All code is to be included in the published file. No code will result in a zero mark. Copying will result in a zero mark and reporting to the school for disciplinary hearing (Turnitin will be used). You will receive your mark through SAKAI once the tutors have marked them.