Programming Project Phase II

Optimization of Multivariable Function

By

Tahir Manuel D'Mello

Roll No.180106055

Aman Kumar

Roll No.214103404

Introduction

In phase 1 we have seen various optimization methods which were developed for single variable problems. The limitation of such methods is that they cannot incorporate the problems which are dictated by more than one variable, which are generally the kind of problems we deal in the practical scenario.

Therefore, it is because of this, certain methods are devised which can deal with the problems, those involving multiple variables. However, these methods do make significant use of the Single variable optimization techniques in their algorithm for unidirectional search as we will see later.

Similar to single variable optimization methods, these methods could be categorized as follows:

I. **Direct Search Method:** These methods make use of the function evaluations and single variable optimization techniques to trace out the optima point.e. g, Nelder & Mead Simplex Search Method, Powell's Conjugate Direction Method.

II. **Gradient-Based Methods:** These methods make use of the gradient/Hessian of the function along with the function evaluation and single variable optimization techniques. Generally, they are computationally more expensive as compared to Direct Search Methods. E.g., Cauchy's Steepest Descent Method, Newton's Method, Marquardt's Method, Conjugate Gradient Method.

In this report, we will be discussing about Powell's Conjugate Direction Method.

Powell’s Conjugate Direction Method

It uses history of previous points to create new search directions. Basic idea is to create a set of N Linearly independent search directions and perform a series of unidirectional searches along each of these search directions, starting each time from the previous best point. This procedure guarantees to find the minimum of quadratic function by one pass of N unidirectional searches along each search direction.

Following are the steps involved in algorithm:

Step 1 Choose starting point x (0) and a set of N linearly independent directions;

possibly s(i) = e(i) for i = 1, 2, . . ., N.

Step 2 Minimize along N unidirectional search directions using the previous

minimum point to begin the next search.

Begin with the search direction s (1) and end with s(N).

Thereafter, perform another unidirectional search along s (1).

Step 3 Form a new conjugate direction d using the extended parallel subspace

property.

Step 4 If || d || is small or search directions are linearly dependent, Terminate;

Else replace s(j) = s(j−1) ∀ j = N, N − 1, . . ., 2.

Set s (1) = d/ ||d|| and go to Step 2.

Result and Discussion

In this part, the discussion on results obtained via program on the following questions has been made.

1. Sum square function: $f(x) = \sum_{i=1}^{d} ix^2$

2. Rosenbrock function:$f(x) = \sum_{i=1}^{d-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$

3. Dixon Price function:$f(x) = (x_1 - 1)^2 + \sum_{i=2}^{d} i(2x_i^2 - x_{i-1})^2$

4. Trid function:$\sum_{i=1}^{d}(x_i - 1)^2 - \sum_{i=2}^{d} x_i x_{i-1}$

5. Zakharov function:$\sum_{i=1}^{d} x_i^2 + \left(\sum_{i=1}^{d} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{d} 0.5ix_i\right)^4$

Problem 1:

Number of variables	Initial Guess	Iterations	Converged Value	Function value
5	[-1.2, 5, 3.6, 2, -1]	2	[0, 0, 0, 0, 0]	0
5	[-2, 2, 7, 8, -1]	2	[0, 0, 0, 0, 0]	0
5	[-0.2, 0.08, 0.5, 1, -1]	2	[0, 0, 0, 0, 0]	0

Problem 2:

Number of variables	Initial Guess	Iterations	Converged Value	Function Value
3	[-1.4, 0.6, 1.8]	13	[1, 1, 1]	0
3	[-0.237, 0.064, 0.019]	18	[1, 1, 1]	0
3	[0.177, 0.044, 0.015]	18	[1, 1, 1]	0

Peculiarity observed: With varying initial guess, there may be some points where the convergence is not as per the expectations, however, incorporating the restart feature in the algorithm, the convergence has been successfully done for those points.

Below table shows the restart implementation for one such point:

Number of restarts	Initial Guess	Iterations	Converged Value
0	[0.95, 2, -0.6]	32	[-0.803, 0.656, 0.429]
1	[-0.803, 0.656, 0.429]	24	[0.177, 0.044, 0.015]
2	[0.177, 0.044, 0.015]	18	[1, 1, 1]

Problem 3:

Number of variables	Initial Guess	Iterations	Converged Value	Function value
4	[-1.9, 1.4, -0.4. 8.7]	11	[1, 0.707, 0.594, 0.546]	0
4	[0.988, 0.75, 0.606, 0.566]	7	[1.001, 0.708, 0.594, 0.544]	0
4	[3, -2, -1.5, 6]	22	[0.997, 0.706, 0.595, 0.544]	0

Peculiarity observed: With varying initial guess, there may be some points where the convergence is not as per the expectations, however, incorporating the restart feature in the algorithm, the convergence has been successfully done for those points.

Below table shows the restart implementation for one such point:

Number of restarts	Initial Guess	Iterations	Converged Value
0	[2.8, -1.5, 3.5, -2]	22	[0.967, 0.686, 0.586, -0.576]
1	[0.967, 0.686, 0.586, -0.576]	9	[1, 0.707, 0.595, 0.545]

In this case the, after the 2^{nd} restart the convergence has not changed, it could be because the algorithm has struck some local minima which is a plausible explanation considering the non-linearity of the function.

Problem 4:

Number of variables	Initial Guess	Iterations	Converged Value	Function value
6	[5.23, 10, 5.3, 2.54, 2.1, 7]	2829	[5.545, 9.085, 10.622, 10.151, 7.689, 4.033]	-47.472000.
6	[4.2, -10, 3.3, 7, 5.3, 2]	8517	[6.225, 10.084, 11.843, 11.571, 9.377, 5.046]	-49.370000.
6	[4.7, -11, 4.2, 8, 5, 3]	10529	[5.827, 9.667, 11.423, 11.159, 9.087, 4.68]	-48.950000.

Peculiarity observed: As evident from the Iterations, the function is highly computationally-expensive, even a slight change in the initial guess can affect the iteration counter very much. The theoretical convergence is -50.000 which is likely to fall within a narrow range of domain values for 6 variables. As evident from the table, due to non-linearity of the function, it's possibly getting converged at some nearby local minima.

Problem 5:

Number of variables	Initial Guess	Iterations	Converged Value	Function value
2	[-1.9, 9.78]	4	[0, 0]	0
2	[-3, 6]	4	[0, 0]	0
2	[2, 3]	4	[0, 0]	0