Report NO4LSP

Chiodo Martina - 343310
Vigè Sophie - 339268

INTRODUZIONE

PROBLEMA 64

PROBLEMA 75

PROBLEMA 76

Model

The function described in this problem is the following

$$F(\mathbf{x}) = \frac{1}{2} \sum_{k=1}^{n} f_k^2(x)$$

$$f_k(\mathbf{x}) = x_k - \frac{x_{k+1}^2}{10}, \quad 1 \le k < n$$

$$f_n(\mathbf{x}) = x_n - \frac{x_1^2}{10}$$

where n denotes the dimensionality of the input vector $\mathbf{x}$.

The starting point for the minimization is the vector $\mathbf{x}_0 = [2, 2, \ldots, 2]$.

To be able to say something more about the behaviour of the problem is useful to look at the gradient of the function $F(\mathbf{x})$ and at its Hessian matrix.

$$\nabla F(\mathbf{x}) = \begin{bmatrix} \frac{\partial F}{\partial x_1}(\mathbf{x}) \\ \vdots \\ \frac{\partial F}{\partial x_k}(\mathbf{x}) \\ \vdots \\ \frac{\partial F}{\partial x_n}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} \frac{1}{2} \left[f_n^2 + f_1^2 \right](\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_k} \frac{1}{2} \left[f_{k-1}^2 + f_k^2 \right](\mathbf{x}) \\ \vdots \\ \frac{\partial}{\partial x_n} \frac{1}{2} \left[f_{n-1}^2 + f_n^2 \right](\mathbf{x}) \end{bmatrix} = \begin{bmatrix} -\frac{x_1}{5}\left(x_n - \frac{x_1^2}{10}\right) + \left(x_1 - \frac{x_1^2}{10}\right) \\ \vdots \\ -\frac{x_k}{5}\left(x_{k-1} - \frac{x_k^2}{10}\right) + \left(x_k - \frac{x_{k+1}^2}{10}\right) \\ \vdots \\ -\frac{x_n}{5}\left(x_{n-1} - \frac{x_n^2}{10}\right) + \left(x_n - \frac{x_1^2}{10}\right) \end{bmatrix}$$

Due to the particular structure of the function, the Hessian matrix as a sparse structure, with only 3 diagonals different from zero. The non-zero elements are the following:

$$\frac{\partial^2 F}{\partial x_k^2}(\mathbf{x}) = -\frac{1}{5}x_{k-1} - \frac{3}{50}x_k^2 + 1, \quad 1 < k \le n \qquad \frac{\partial^2 F}{\partial x_1^2}(\mathbf{x}) = -\frac{1}{5}x_n - \frac{3}{50}x_1^2 + 1,$$

$$\frac{\partial^2 F}{\partial x_k \partial x_{k+1}}(\mathbf{x}) = -\frac{1}{5}x_{k+1}, \quad 1 \le k < n \qquad \frac{\partial^2 F}{\partial x_n \partial x_1}(\mathbf{x}) = -\frac{1}{5}x_1$$

$$\frac{\partial^2 F}{\partial x_k \partial x_{k-1}}(\mathbf{x}) = -\frac{1}{5}x_k, \quad 1 < k \le n \qquad \frac{\partial^2 F}{\partial x_1 \partial x_n}(\mathbf{x}) = -\frac{1}{5}x_n$$

We can now easily notice that the gradient of the function is null when all the components of the vector $\mathbf{x}$ are equal to 0. The Hessian matrix is positive definite, so the point $\mathbf{x} = \mathbf{0}$ is a minimum of the function $F(\mathbf{x})$. Because of the definition of the function, 0 is the lowest value that the function can assume, so the minimum is global.

Nealder Mead Method

We now report a table containing some general results obtained by running the Nealder Mead method on the function $F(\mathbf{x})$.

	avg fbest	avg num of iters	avg time of exec (sec)	n failure	avg roc
10	1.0000e-04	2.1864e+02	4.5116e+00	0.0000e+00	NaN
25	0.0000e+00	1.6804e+03	3.1524e+01	0.0000e+00	NaN
50	2.9039e+01	1.4007e+04	2.6917e+02	1.0000e+01	NaN

Figura 1: Resultats obtained by running the symplex method on the function $F(\mathbf{x})$.

First thing we can notice is that for smaller dimensionalities the symplex method is able to find the minimum in a reasonable amount of time, but when the dimensionality becomes higher the method starts failing. From the plot in figure (2), we can see that for most points belonging to $\mathbb{R}^{50}$, the method keeps iterating until the maximum number of iterations is reached without satisfying the stopping criterion. This behaviour can probably be explained by the fact that when the dimensionality increases the starting point is more far from the minimum due to its definiotion, so the method needs to perform more iterations to reach the minimum.
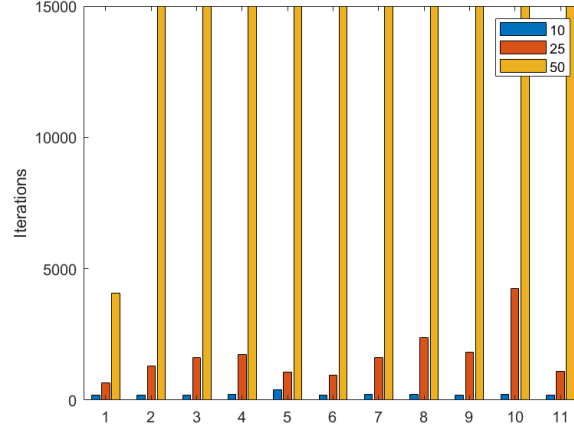
Figura 2: Number of iterations needed by the Nealder Mead method to find the minimum of the function $F(\mathbf{x})$ for each starting point.

Modified Newton Method

We now report a table containing some general results obtained by running the Modified Newton method on the function $F(\mathbf{x})$. We obviously expect the method to perform better than the symplex method because of the exact derivatives used in the computation of the descent direction.

	avg fbest	avg gradf_norm	avg num of iters	avg time of exec (sec)	n failure	avg roc
1000	6.7091e-10	3.6631e-05	4.0000e+00	1.6874e-02	0.0000e+00	1.9391e+00
10000	2.4479e-17	6.9969e-09	4.0000e+00	1.4366e-02	0.0000e+00	2.0053e+00
100000	8.1020e-21	1.2730e-10	5.0000e+00	1.9585e-01	0.0000e+00	2.0026e+00

Figura 3: Resultats obtained by running the Modified Newton Method on the function $F(\mathbf{x})$ using the exact derivatives.

This time, the method always converges to the minimum in very few iterations, even for higher dimensionalities.

CONCLUSIONI