VERKFRÆÐILEGAR BESTUNARAÐFERÐIR

Team 2 - Day 5

T-423-ENOP

Arnar Gylfi Haraldsson
`arnarh23@ru.is`

Hafþór Árni Hermannsson
`hafthorh20@ru.is`

Ragnheiður Gná Gústafsdóttir
`ragnheidurg@ru.is`

May 7, 2024

Exercise 1 - Pattern Search Algorithm

Description

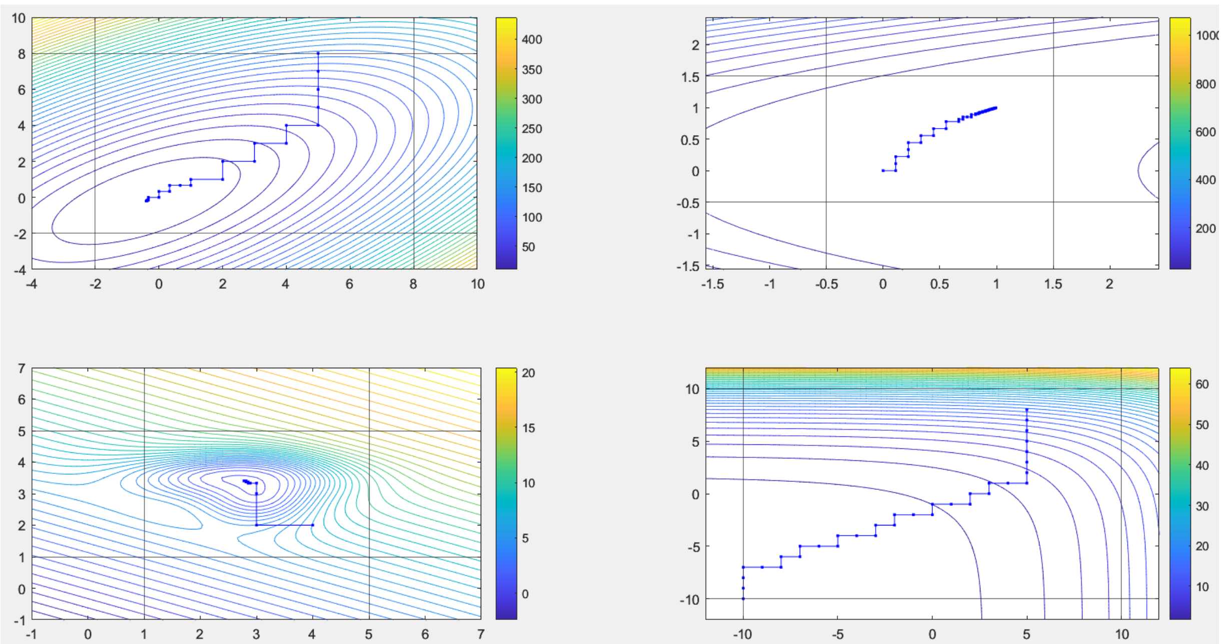
Develop and implement a simple pattern search algorithm that seeks for a minimum of a function f on a rectangular grid of variable size. Visualize the operation of the algorithm for the case when f is a function of two variables.

Test your function using the following cases:

1. $f(x,y) = 2x^2 + 3y^2 - 3xy + x$, where $-2 \le x, y \le 8$, starting point $[5\ 8]^T$
2. $f(x,y) = (1-x)^2 + 5(x-y^2)^2$, where $-0.5 \le x, y \le 1.5$, starting point $[0\ 0]^T$
3. $f(x,y) = (x+2y)\cdot(1-0.9\cdot\exp(-0.3\cdot(x-2.5)^2 - 2\cdot(y-3.5)^2))\cdot(1-0.9\cdot\exp(-(x-3)^2-(y-3)^2))$, where $1 \le x, y \le 5$, starting point $[4\ 2]^T$
4. $f(x,y) = \exp(x/5) + \exp(y/3)$, where $-10 \le x, y \le 10$, starting point $[5\ 8]^T$

Solution

We define a rectangular grid with an initial size and check the function values for each of the four adjacent points to an initial guess (x_0, y_0). If a point is within defined boundaries and yields a function value lesser than the initial guess we accept it and repeat the process. If no point is found within the boundary that yields a lower function value we decrease the grid size by a factor of three and repeat. This process yields the following figures, functions 1,2,3 and 4 going from left to right



Exercise 2 - Smarter Pattern Search Algorithm

Description

Develop and implement an improved pattern search algorithm that seeks for a minimum of a function f on a rectangular grid of variable size. This version incorporates additional moves, akin to line search steps but still restricted to the grid. Visualize the operation of the algorithm for the case when f is a function of two variables.

Test your function using the following cases:

1. $f(x, y) = 2x^2 + 3y^2 - 3xy + x$, where $-2 \leq x, y \leq 8$, starting point $[5\ 8]^T$
2. $f(x, y) = (1-x)^2 + 5(x - y^2)^2$, where $-0.5 \leq x, y \leq 1.5$, starting point $[0\ 0]^T$
3. $f(x, y) = (x+2y) \cdot (1 - 0.9 \cdot \exp(-0.3 \cdot (x-2.5)^2 - 2 \cdot (y-3.5)^2)) \cdot (1 - 0.9 \cdot \exp(-(x-3)^2 - (y-3)^2))$, where $1 \leq x, y \leq 5$, starting point $[4\ 2]^T$
4. $f(x, y) = \exp(x/5) + \exp(y/3)$, where $-10 \leq x, y \leq 10$, starting point $[5\ 8]^T$

Functionality

It operates by systematically exploring neighboring points around a current point in a grid-like fashion, this time also allowing for linearly dependent direction vector (dependent to the previous direction vector) in the 2D space to be considered. Here are the key features:

- **Initialization:** Starts from a user-defined starting point and initializes variables such as the grid size and the minimum function value found.
- **Exploration:** Iteratively explores neighboring points by adjusting both dimensions simultaneously within a defined grid size.
- **Bound Checking:** Ensures that the exploration does not go beyond the user-specified bounds of the problem.
- **Checking for new minima:** Checks if a new local minima has been found and updates the minima if it has been found, else it continues.
- **Termination:** The algorithm terminates either when a maximum number of iterations is reached or a specified tolerance level is achieved indicating convergence.

Results

Case	Iterations	Improvements	$x_{\min}$		$f_{\min}$
1	13	21	-0.4036	-0.1945	-0.1998
2	8	2	1	1	0
3	14	57	2.7798	3.4064	0.3567
4	9	51	-10.0000	-9.9982	0.1710

Table 1: Results from the smarter pattern search for each of the four cases.

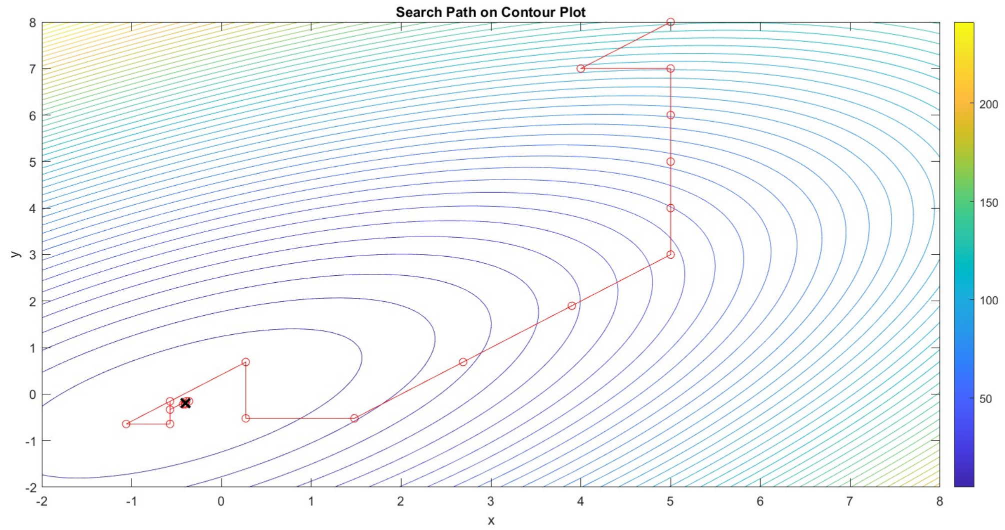


Figure 1: 2D visualization for the smarter pattern search for case 1

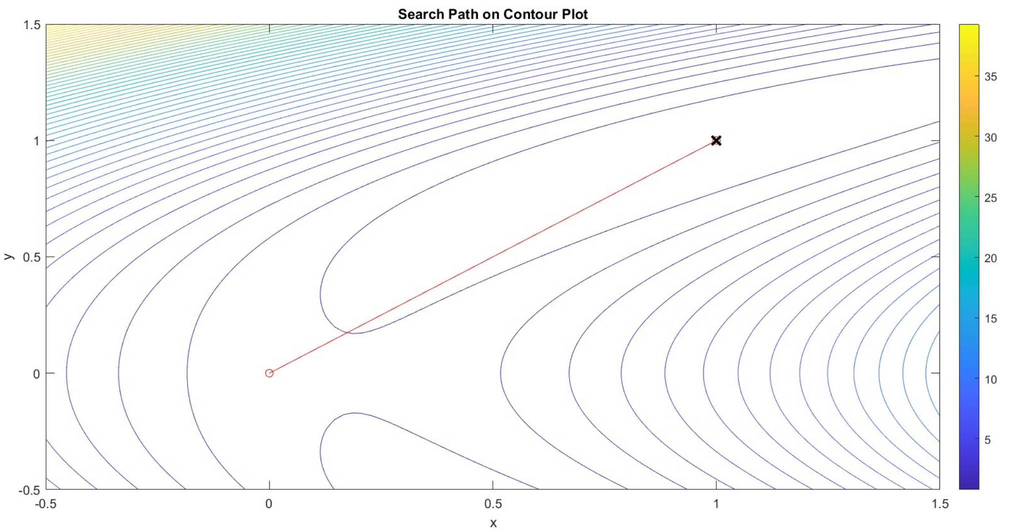


Figure 2: 2D visualization for the smarter pattern search for case 2

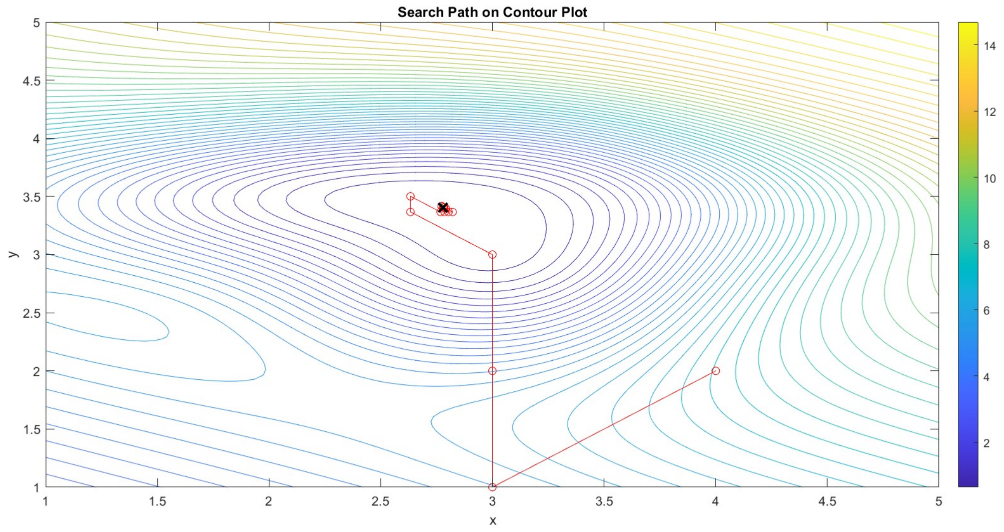


Figure 3: 2D visualization for the smarter pattern search for case 3

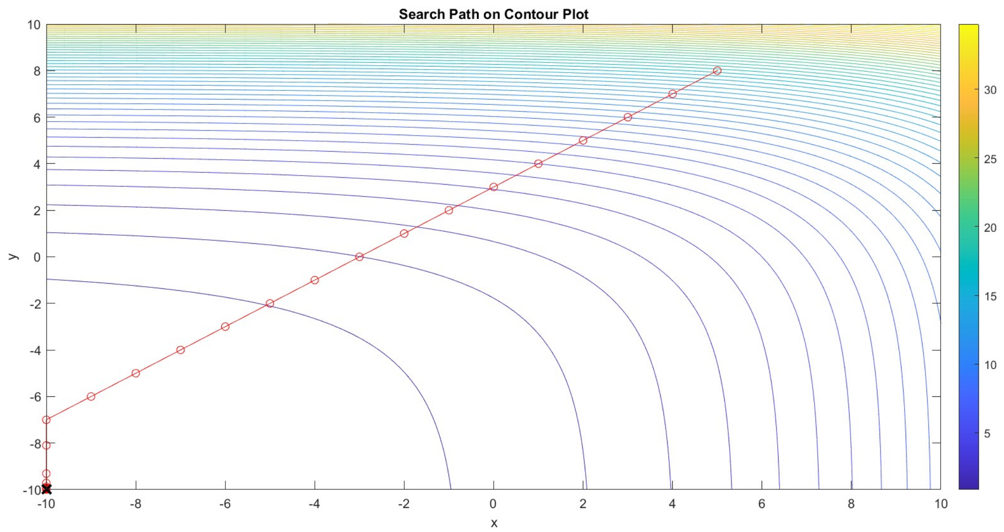


Figure 4: 2D visualization for the smarter pattern search for case 4

Exercise 3 - 3D Visualization of Pattern Search Algorithm

Description

Develop a more extensive visualization of the pattern search algorithm from Exercise 1. This should include contour plots as well as 3D surface plots indicating the minimum to be found. Include one regular and one zoomed-in 3D surface plot to provide detailed insight into the algorithm's performance at finding the minimum.

Result

"Placeholder figures"

Case 1

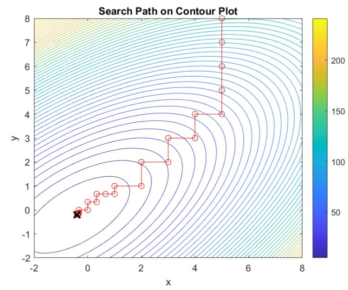


Figure 5: 2d contour plot for case 1

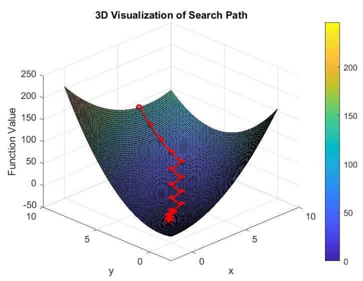


Figure 6: 3D plot for case 1

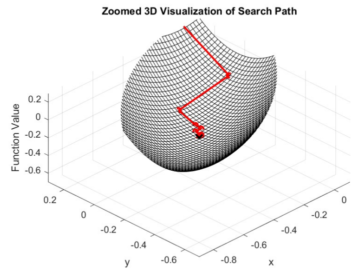


Figure 7: Zoomed in 3D plot for case 1

Case 2

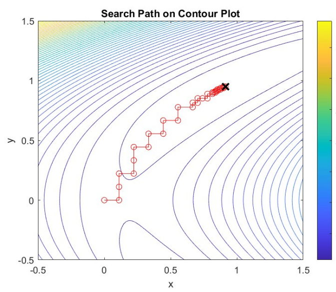


Figure 8: 2d contour plot for case 2

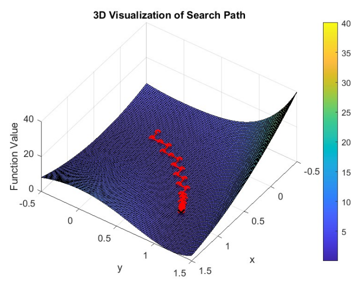


Figure 9: 3D plot for case 2

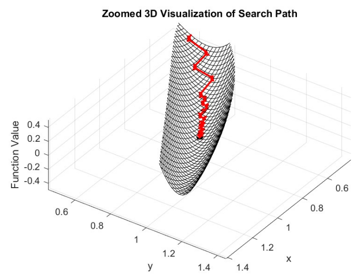


Figure 10: Zoomed in 3D plot for case 2

Case 3

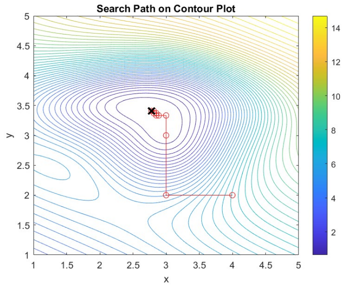


Figure 11: 2d contour plot for case 3

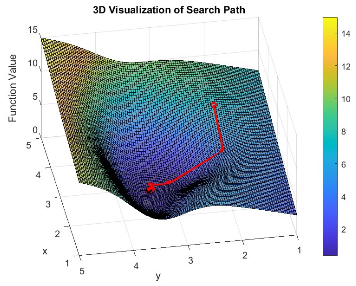


Figure 12: 3D plot for case 3

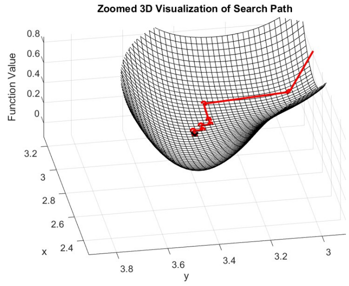


Figure 13: Zoomed in 3D plot for case 3

Case 4

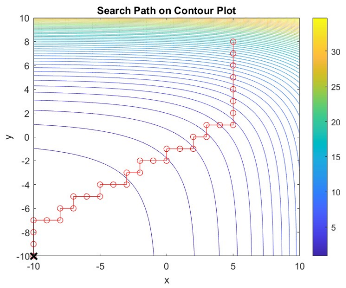


Figure 14: 2d contour plot for case 4

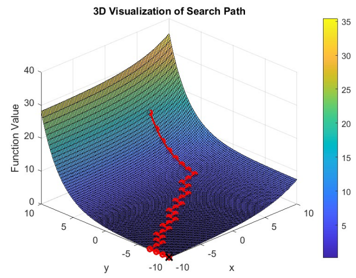


Figure 15: 3D plot for case 4

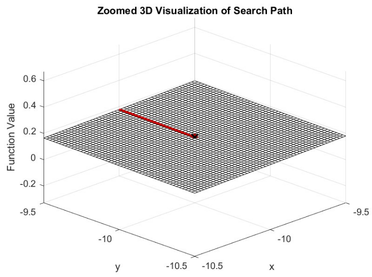


Figure 16: Zoomed in 3D plot for case 4

Exercise 4 - Hooke-Jeeves Algorithm

Description

The famous Hooke- Jeeves algorithm consists of a sequence of exploratory moves about a base point which, if successful, are followed by pattern moves. Summary:

- The search starts on a rectangular grid of size D.
- Test the neighbors of the current point $x^{(i)}$ sequentially and immediately move if a better function value is found (opportunistic procedure).
- After considering all variables, if a better point $x^{(i+1)}$ is found, perform a line search along the direction $h = x^{(i+1)} - x^{(i)}$.
- If the line search yields an even better point, update $x^{(i+1)}$ accordingly.
- If the neighbor search fails, reduce the grid size (e.g., by three times).
- Continue until the grid size falls below the assumed tolerance threshold.

Test your function using the following cases:

1. $f(x, y) = 2x^2 + 3y^2 - 3xy + x$, where $-2 \le x, y \le 8$, starting point $[5\,8]^T$.
2. $f(x, y) = (1 - x)^2 + 5(x - y^2)^2$, where $-0.5 \le x, y \le 1.5$, starting point $[0\,0]^T$.
3. $f(x, y) = (x + 2y) \cdot (1 - 0.9 \cdot \exp(-0.3 \cdot (x - 2.5)^2 - 2 \cdot (y - 3.5)^2)) \cdot (1 - 0.9 \cdot \exp(-(x - 3)^2 - (y - 3)^2))$, where $1 \le x, y \le 5$, starting point $[4\,2]^T$.
4. $f(x, y) = \exp(x/5) + \exp(y/3)$, where $-10 \le x, y \le 10$, starting point $[5\,8]^T$.

Implementation

The "HookeJeeves.m" function takes in the objective function, bounds, starting point, maximum number of iterations, and tolerance. It initiates an opportunistic search on a grid, testing neighboring points sequentially for better function values. Unlike the algorithm developed in exercise 2,the Hooke-Jeeves method adjusts the exploration point along one dimension at a time, and may adjust its step size based on the results. It checks if moving along a dimension improves the solution, and if so, continues in that direction. If unsuccessful, it reduces the grid size iteratively until convergence is achieved.

Results

Case	Iterations	Improvements	$x_{\min}$	$f_{\min}$
1	15	22	$\begin{bmatrix}-0.3951 & -0.1975\end{bmatrix}$	-0.2000
2	6	11	$\begin{bmatrix}1.0000 & 1.0000\end{bmatrix}$	2.9582e-31
3	10	9	$\begin{bmatrix}2.7778 & 3.4074\end{bmatrix}$	0.3568
4	7	20	$\begin{bmatrix}-10 & -10\end{bmatrix}$	0.1710

Table 2: Results from the Hooke Jeeves algorithm for each of the four cases.

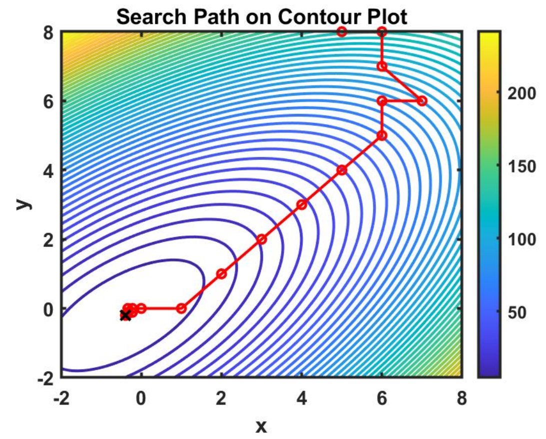


Figure 17: Visualization for test problem 1

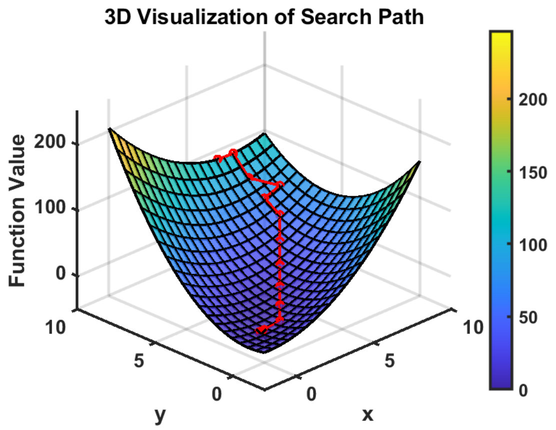


Figure 18: Visualization for test problem 1 3D

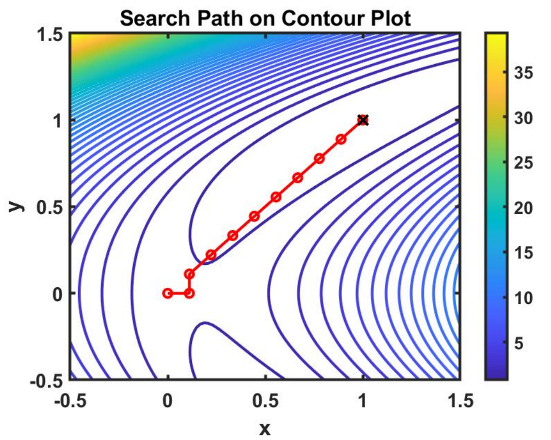


Figure 19: Visualization for test problem 2

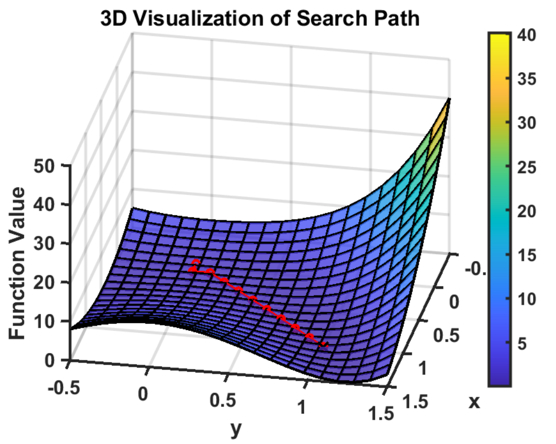


Figure 20: Visualization for test problem 2 3D

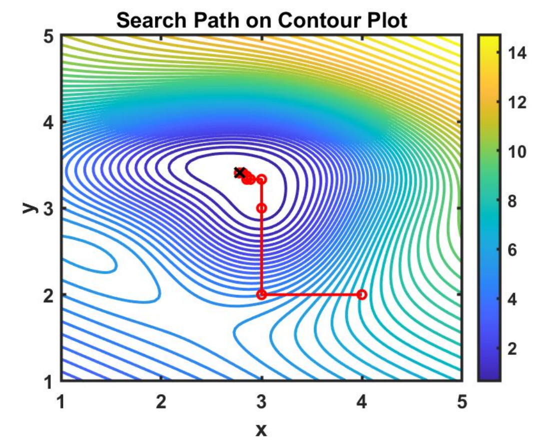


Figure 21: Visualization for test problem 3

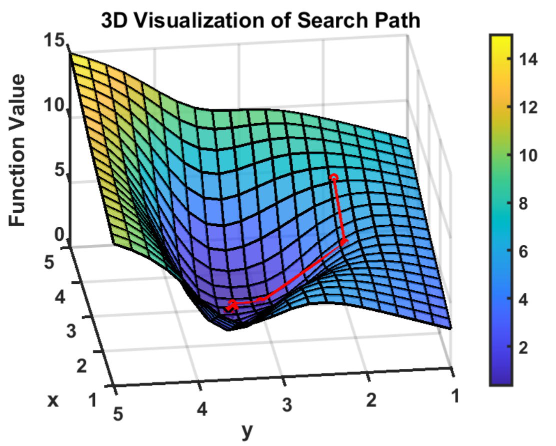


Figure 22: Visualization for test problem 3 3D

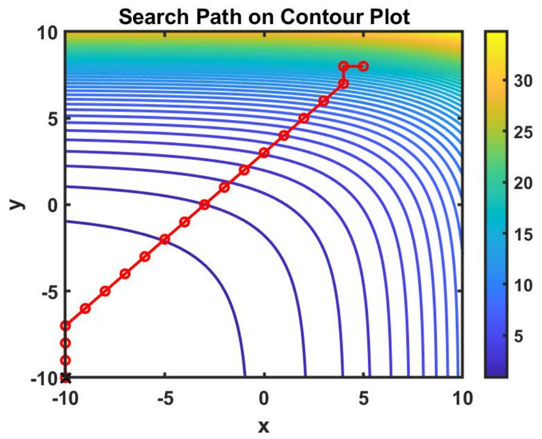


Figure 23: Visualization for test problem 4

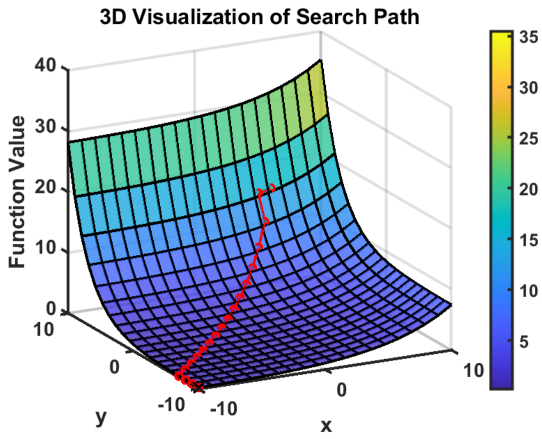


Figure 24: Visualization for test problem 4 3D