Analysis of Optimization Algorithms Using Shubert Function

Berkay Ahi
Dept. of Mathematical Engineering
Yildiz Technical University
Istanbul, Turkey
berkay.ahi@std.yildiz.edu.tr

Lect. PhD. Gökhan Göksu
Dept. of Mathematical Engineering
Yildiz Technical University
Istanbul, Turkey
gokhan.goksu@yildiz.edu.tr

Assoc. Prof. Hale Köçken
Dept. of Mathematical Engineering
Yildiz Technical University
Istanbul, Turkey
hgonce@yildiz.edu.tr

***Abstract*— There are several test functions used to measure the efficiency of optimization algorithms. In this paper, the performance of five optimization algorithms—Newton-Raphson, Hestenes-Stiefel, Polak-Ribiere, Fletcher-Reeves, and Dai-Yuan—is compared on the multimodal and periodic Shubert test function. Performance measures such as execution time, number of iterations to convergence, and accuracy of the global minima found were used for comparison. This study provides insights into the capabilities and limitations of each algorithm when applied to complex optimization problems.**

Keywords—optimization, benchmark functions, Shubert function, conjugate, gradient, MATLAB

I. Introduction

Optimization algorithms are fundamental tools in numerous scientific and engineering disciplines, and require a thorough evaluation of their reliability, validity, and efficiency [1]. Key indicators of an effective optimization algorithm are its capacity to generate precise approximations while maintaining a reasonable number of iterations and execution time. A well-known method for assessing optimization algorithms is to benchmark their performance against a set of established test functions [2]. This approach provides a comprehensive evaluation of how algorithms handle different types of optimization problems, including unimodal, multimodal, regular, irregular, separable, non-separable, and multi-dimensional problems. Such benchmarking is crucial for identifying the strengths and limitations of various algorithms under different scenarios.

In this paper, the performance of five different optimization algorithms, Newton-Raphson, Hestenes-Stiefel, Polak-Ribiere, Fletcher-Reeves, and Dai-Yuan [3] is evaluated. These algorithms are applied to the two and three dimensional Shubert function, a test function known for its highly complex, periodic, and multimodal nature. The Shubert function poses significant challenges due to its landscape, which includes numerous local and global minima. This study evaluates these algorithms based on several performance metrics, including execution time, the number of iterations to convergence, and the accuracy of the found global minima. This benchmarking study aims to provide insights into the practical applicability of each algorithm when dealing with complex optimization problems.

II. Shubert Problem

The generalized Shubert function is defined for an n n-dimensional input vector $x = (x_1, x_2, \dots, x_n)$. It is a commonly used test function in optimization and numerical analysis due to its complex landscape with several local and global minima. The function is defined as follows,

$$\min_x f(x) = \prod_{i=1}^{n}\left(\sum_{j=1}^{5} jcos((j+1)x_i + j)\right) \tag{1}$$

$$-10 \le x_i \le 10, where\ i \in \{1,2,\dots,n\} \tag{2}$$

In optimization and numerical analysis, the choice of dimensionality n plays a crucial role in the complexity and behavior of test functions. For the Shubert function, n = 2 is commonly used for several reasons. By choosing n = 2 the function can be easily visualized in a two-dimensional or three-dimensional plot. This allows researchers to gain intuitive insights into the landscape of the function, including the locations and characteristics of its minima. Even in two dimensions, the Shubert function exhibits a highly complex landscape with several local and global minima. This complexity is sufficient to test and evaluate the performance of optimization algorithms. In addition, using n = 2 makes it easier to compare results across different studies that have been done by using Shubert function. The two-dimensional Shubert function is a well-known benchmark problem in optimization, and many algorithms have been tested against it. Considering these reasons, n = 2 was chosen in our study to test several optimization algorithms. In addition to this, tests were also performed for n=3, i.e. for the three-dimensional Shubert function.

A. *Two-Dimensional Shubert Function*

As explained above, our tests are performed for n equals two. The two-dimensional Shubert function can be seen in below,

$$f(x) = \left(\sum_{i=1}^{5} icos\big((i+1)x_1 + i\big)\right)\left(\sum_{i=1}^{5} icos\big((i+1)x_2 + i\big)\right) \tag{3}$$

While the number of local minima for generalized Shubert problem for a given n is unknown, but for n = 2, the function has around 800 local minima, 18 of which are global with,

$$f(x) = -186.7309.$$

All two dimensional global minimizers are listed in the Table 1,

TABLE I.
GLOBAL MINIMIZERS OF THE SHUBERT FUNCTION

$\boldsymbol{f(x_1, x_2) = -186.7309}$		
(-7.7083, -7.0835)	(-7.0835, -1.4251)	(-1.4251, 5.4829)
(-7.7083, -0.8003)	(-7.0835, 4.8581)	(-0.8003, 4.8581)
(-7.7083, 5.4829)	(-1.4251, -0.8003)	(4.8581, 5.4829)

As it can be seen from the Table 1, the Shubert function has 9 pairs of global minimizers in 2 dimensions. Also, to illustrate the complexity of the Shubert function, an initial contour plot of the function is presented in Figure 1. This plot visually represents the periodic and multimodal nature of the Shubert function, highlighting the challenging landscape that optimization algorithms must navigate.

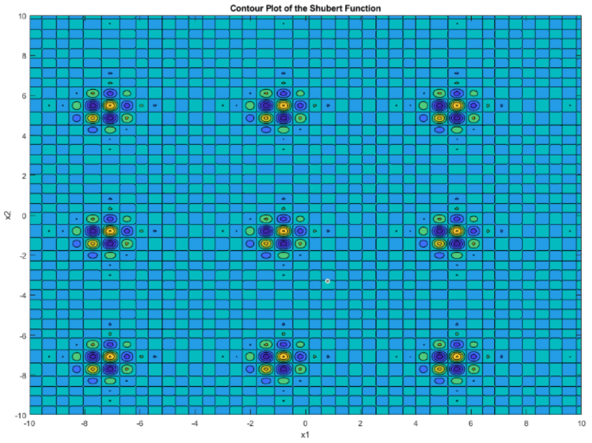


Fig. 1. Contour Plot of the Two-Dimensional Shubert Function.

Following this, the contour plot with local and global minima is shown at Figure 2. Local minima are marked with red circles, and global minima with green stars. This plot clearly depicts the numerous local minima and the fewer global minima, demonstrating the difficulty in finding optimal solutions.

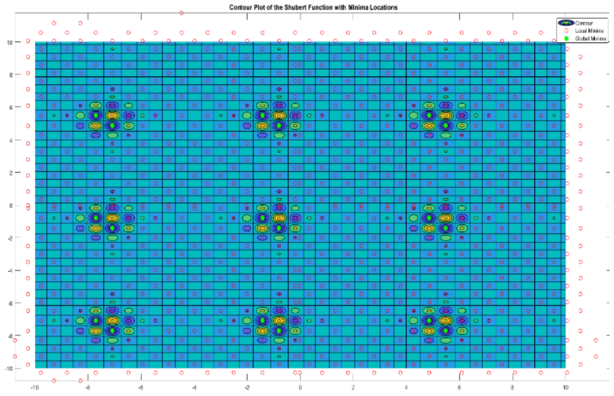


Fig. 2. Contour Plot of the Shubert Function with Minima Locations.

By comparing these two figures, one can understand the potential problems in optimizing the Shubert function and the importance of evaluating the performance of various optimization algorithms in this context.

B. Three-Dimensional Shubert Function

The Three-Dimensional Shubert function is a highly complex and multimodal trigonometric function, known for its challenging optimization landscape. The function can be expressed as,

$$\min_{x} f(x) = \prod_{i=1}^{3}\left(\sum_{j=1}^{5} jcos((j+1)x_i + j)\right) \qquad (4)$$

$$-10 \leq x_i \leq 10, where\ i \in \{1,2,3\} \qquad (5)$$

In addition to being complex and multimodal, the Shubert function is also periodic. It contains a vast number of local minima, making it an tough candidate for evaluating the robustness of optimization algorithms. Given its properties, the Shubert function challenges algorithms to distinguish between local and global minima effectively, providing insights into their convergence behavior and sensitivity to initial conditions.

III. Implementation

The algorithms and benchmarking code were implemented using MATLAB. Initial points were selected from a uniform distribution $x_{1,2} \sim U_2(-10,10)$ within the problem domain, using MATLAB's rand() function. The error bound was set at $\epsilon = 10^{-4}$, with the stopping criterion defined as $||\nabla f(x)|| \leq \epsilon$. This ensures that the optimization process continues until the gradient norm falls below the specified threshold, indicating convergence to a local minimum.

The Shubert function's landscape exhibits a highly periodic and multi-modal nature, as illustrated in the provided potential landscape at Fig 2.

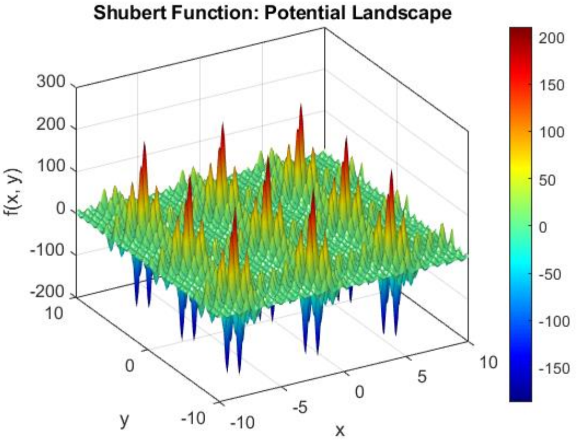


Fig. 3. Shubert Function’s 3D Surface Plot.

Given this periodicity, it is essential to choose an appropriate modulo operation to keep the iterations within the specified bounds. By using mod 10, we ensure that the x_i values stay within the interval [-10,10], aligning with the natural periodicity of the Shubert function. This choice preserves the function's characteristics, facilitating more effective optimization by leveraging its periodic nature. Using a smaller modulo value, such as 5, could truncate or distort the function's landscape, making it less suitable for optimization purposes. Thus, using mod 10 is the preferred approach for maintaining the integrity of the Shubert function's periodic landscape during optimization. If this approach was not applied, the x_1 and x_2 values in all optimization algorithms except the Newton-Raphson algorithm would go out of the interval after a few iterations, showing how much the periodicity affected the contour plots. Therefore, in order to obtain more effective results in visualizations, such an approach was needed due to the periodic nature of the function. After all these conditions were coded in MATLAB, the optimization algorithms were run with random initial points and the results were analyzed in the section four.

IV. Result

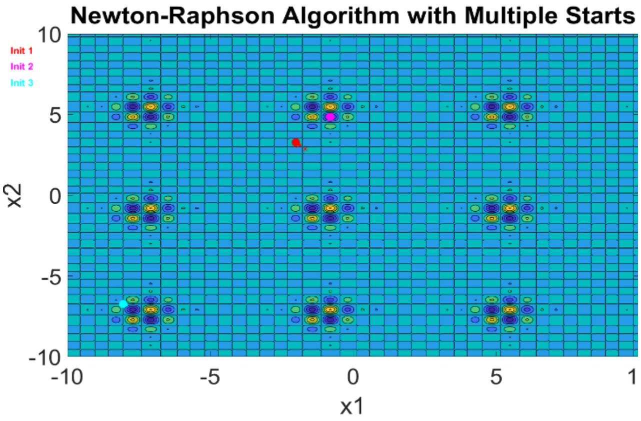
In this section, the algorithms are run for two-dimensional and three-dimensional Shubert functions. The results obtained with three different random initial points are plotted on the contour plot for the two-dimensional Shubert function as shown in Figure 3. The plots for the first initial point are plotted in red, the plots for the second are plotted in pink and the plots for the third are plotted in blue. The minimum points found are marked on the figure with a big circle. The figures presented in this section illustrate the iterative steps taken by five different optimization algorithms—Newton-Raphson, Hestenes-Stiefel, Polak-Ribiere, Fletcher-Reeves, and Dai-Yuan—when applied to the two-dimensional Shubert function. Each figure shows the convergence paths from three different initial points, highlighting the algorithms' ability to navigate the function's complex landscape.

The Newton-Raphson algorithm converges quickly but is highly sensitive to initial conditions, often finding local minima rather than the global minimum. The Hestenes-Stiefel algorithm demonstrates better robustness compared to Newton-Raphson, effectively moving towards the global minimum in many cases.

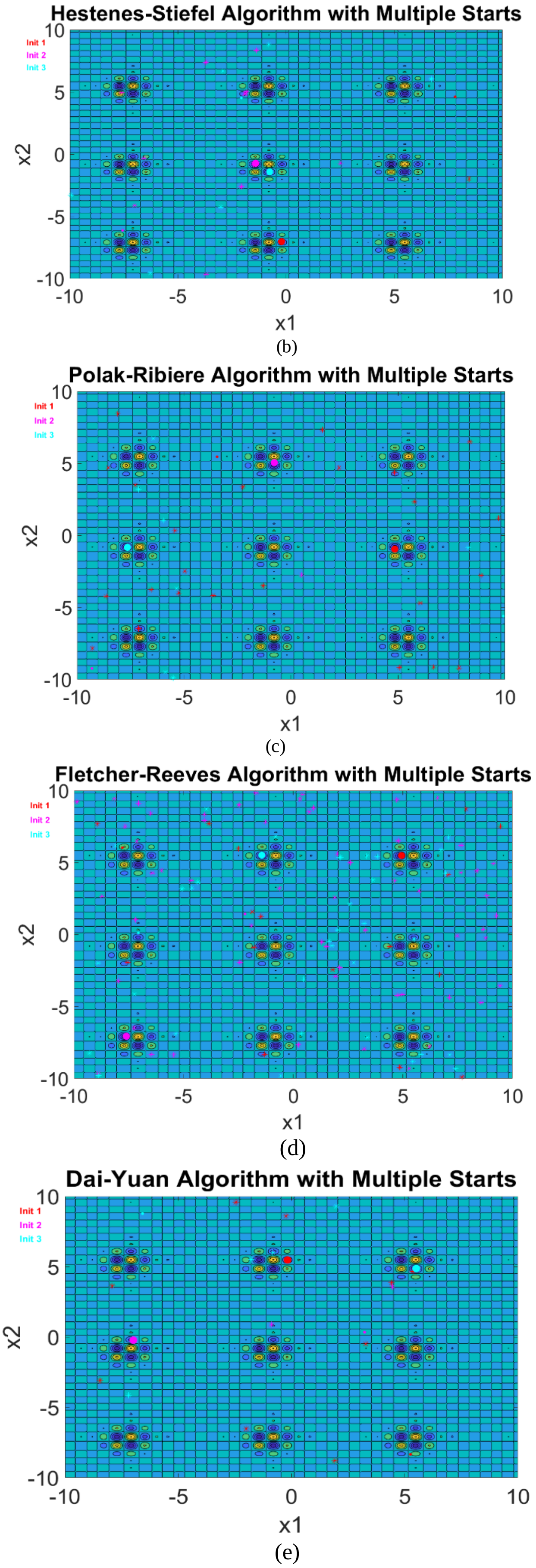
The Polak-Ribiere algorithm, another conjugate gradient method variant, shows balanced performance similar to Hestenes-Stiefel but requires more iterations. The Fletcher-Reeves algorithm, like Polak-Ribiere and Hestenes-Stiefel, uses conjugate gradients, and its convergence behavior highlights its strength in finding minimum points.

The Dai-Yuan algorithm, a variant of the conjugate gradient method with a different formula for the β parameter, shows effective and robust performance in finding the global minima. Compared to other algorithms, Dai-Yuan performs similarly to Hestenes-Stiefel but is slightly more effective. This comparison highlights the algorithm's strengths and limitations, demonstrating how changes in the β parameter can influence efficiency and robustness.

These figures help compare how well each algorithm performs in finding the minimum points. The visualizations highlight the strengths and weaknesses of each algorithm. The different convergence paths and number of steps taken by each algorithm show the trade-offs in their design and parameter choices. Most algorithms, especially Newton-Raphson, face difficulties in finding global minima due to their sensitivity to the starting points, and instead usually converge to local minima. This contrast shows the importance of initial conditions in optimization algorithms. Even if the Polak-Ribiere and Fletcher-Reeves algorithms start with a bad initial point, they are able to produce more optimized results thanks to the steps they take afterwards. In another study we conducted, all these algorithms were run 100 times and the results obtained were visualized with data analysis techniques and the performance of the algorithms was interpreted.



(a)



(b)

(c)

(d)

(e)

Fig. 4. Contour plots showing optimization algorithms with multiple starts: (a) Newton-Raphson, (b) Hestenes-Stiefel, (c) Polak-Ribiere, (d) Fletcher-Reeves, and (e) Dai-Yuan.

A. *Three-Dimensional Shubert Function Results*

The results of running the three-dimensional Shubert function using four different optimization algorithms—Newton-Raphson, Hestenes-Stiefel, Polak-Ribiere, and Fletcher-Reeves—are detailed in Table 2. This table highlights the challenges faced when optimizing the Shubert function, which is known for its highly complex landscape with numerous local and global minima.

The Newton-Raphson algorithm, while efficient in terms of the number of iterations and time taken, consistently converges to local minima, failing to find the global minimum in all runs. This algorithm's reliance on the Hessian matrix and its sensitivity to initial points contribute to its tendency to get stuck in local minima.

The Hestenes-Stiefel algorithm shows varied performance. In some instances, it comes close to the global minimum, but in other cases, it converges to suboptimal solutions. This variability indicates the algorithm's volatility and sensitivity to initial points. The algorithm's efficiency in terms of iterations is offset by its inconsistency in finding the global minimum.

The Polak-Ribiere algorithm also demonstrates similar challenges. Despite taking a higher number of iterations in some cases, it frequently converges to local minima, reflecting its difficulty in navigating the Shubert function's complex landscape. This algorithm's performance further underscores the importance of initial point selection in achieving optimal results.

The Fletcher-Reeves algorithm exhibits the highest variability among the conjugate gradient methods. Its performance ranges from converging to near-global minima to significantly suboptimal points, with substantial variations in the number of iterations and time taken. This inconsistency highlights the difficulty in solving the Shubert function using this method.

Overall, these results illustrate the inherent difficulties in optimizing the Shubert function with conjugate direction algorithms. The function's multiple local and global minima make it challenging to consistently find the global minimum. The algorithms become more volatile, may require more time, and are highly sensitive to initial points, demonstrating the need for robust optimization strategies that can effectively handle such complex functions.

TABLE 2.
THREE-DIMENSIONAL SHUBERT FUNCTION ALGORITHM PERFORMANCES

Algo.	**Time**	**Iteration**	**(x1, x2, x3)**	**F(x1, x2, x3)**	**Error**
N-R	2.2493	11.0000	(7.703360, 8.265737, 5.923677)	0.0000	2508.8326
N-R	1.4266	11.0000	(-8.025754, -4.762576, -3.292863)	0.0000	2508.8326
N-R	1.6621	12.0000	(3.594559, -7.268937, 4.424550)	0.0000	2508.8326
H-S	0.6235	7.0000	(−7.864763, 3.075147, −0.116521)	-654.8666	1853.9660
H-S	4.6606	50.0000	(5.581034, 4.300742, 8.074411)	-2493.7944	15.0382
H-S	5.2635	59.0000	(7.818450, −3.316739, 3.974917)	-193.2449	2315.5877
P-R	1.0663	11.0000	(−6.043803, −9.389181, 4.881485)	-429.0553	2079.7772
P-R	9.0966	101.0000	(0.000449, −0.401557, 8.094445)	-307.0000	2201.8325
P-R	1.8830	22.0000	(2.197333, 2.353328, 7.188846)	-527.7914	1981.0412
F-R	6.2527	71.0000	(6.109788, 1.534430, −6.341551)	-435.2024	2073.6302
F-R	0.8466	10.0000	(−5.201360, 7.730239, −9.426517)	-1345.3337	1163.4989
F-R	7.8778	90.0000	(−0.201972, −6.641457, 9.573613)	-1981.5868	527.2458

$^{\text{N-R}}$ Newton-Raphson, $^{\text{H-S}}$ Hestenes-Stiefel, $^{\text{P-R}}$ Polak- Ribiere, $^{\text{F-R}}$ Fletcher-Reeves

V. Conclusion

Each algorithm has strengths and weaknesses when applied to the Shubert function. Newton-Raphson's sensitivity to initial conditions and dependence on Hessian accuracy make it prone to local minima convergence. Conjugate gradient methods like Hestenes-Stiefel and Polak-Ribiere offer better robustness but require more iterations and careful tuning to avoid non-optimal solutions. Fletcher-Reeves, although simple to implement and robust against poor initial guesses, can also be slower to converge and is similarly prone to getting stuck in local minima. The Dai-Yuan algorithm, another variant of the conjugate gradient method, showed slightly better performance than Hestenes-Stiefel but still faced challenges due to the Shubert function's complex landscape. The challenging multimodal and periodic nature of the Shubert function highlights the need for hybrid or advanced algorithms capable of escaping local minima to find the global optimum effectively. In this study, each of the five algorithms was run 100 times with different random initial points to thoroughly evaluate their performance. The results are summarized and analyzed through several figures like distribution of F values at optimized points, distribution of elapsed time for each algorithm, and total steps taken by each algorithm to converge. Also, all of the results of this experiment are recorded in Table 3. These analyses provide a comprehensive understanding of each algorithm's performance, demonstrating their efficacy and limitations in handling the Shubert function. The extensive evaluation underscores the need for further refinement and the potential benefits of integrating advanced optimization techniques to improve convergence reliability. In addition, to avoid converging to local minima, a well-thought-out initial point selection strategy can be effective.

TABLE 3.
SUMMARIZED BENCHMARK DATA

Algo.	Time			Iteration			f(x)			Error		
	mean	min	max	mean	min	max	mean	min	max	mean	min	max
N-R	0.4105	0.1755	1.3943	5.4000	3.0000	10.0000	0.3711	-186.7309	165.6597	186.9436	0.1617	352.2289
H-S	0.9607	0.0949	4.8387	15.5700	2.0000	79.0000	-117.1758	-185.9021	-1.2791	69.3934	0.6671	185.2902
P-R	3.6788	0.1493	26.5276	57.4800	3.0000	418.0000	-124.4482	-186.3613	-32.7621	62.1210	0.2079	153.8071
F-R	1.6141	0.0920	13.2328	26.9500	2.0000	220.0000	-124.0948	-185.2107	-35.4163	62.4744	1.3585	151.1529
D-Y	0.8674	0.1466	3.5322	14.1100	3.0000	55.0000	-119.9702	-186.3602	-10.5017	66.5990	0.2090	176.0675

- Due to the periodic nature of the Shubert function, the number of steps required to find the minimum point varies a lot with respect to the initial point, so there is no exact number of steps required to find the minimum. However, the average number of iterations required can be seen in Table 3.
- The execution times for the algorithms vary significantly based on their computational complexity and the number of iterations required. Newton-Raphson has the lowest mean execution time due to fewer steps but can be variable because it often gets stuck in local minima. Hestenes-Stiefel and Dai-Yuan show similar performance, Fletcher-Reeves has the highest mean and variability in execution time, reflecting its computational cost and iteration requirements. As a result, since algorithms are very sensitive to the initial point, the volatility in these execution times does not seem unreasonable.
- In the case of the Shubert function, convergence is highly dependent on the initial point. There are 9 pairs of global minimizers in the two-dimensional Shubert function, if an initial point is chosen far from the global minimizer points, the algorithm finds undesirable local optima or saddle points instead of global minima. As can be seen from Table 3, although Polak-Ribiere and Fletcher-Reeves find relatively more optimal minimum points, they are still initial-point sensitive, but not as sensitive as Newton-Raphson.
- The trade-off between execution time and convergence due to initial conditions arises because optimization algorithms need to balance efficiency and accuracy. Newton-Raphson, while fast, is highly sensitive to initial points, often getting stuck in local minima. Conjugate gradient methods like Polak-Ribiere and Fletcher-Reeves are less sensitive to initial conditions but require more iterations and thus more time to find the global minimum. This increased time allows these algorithms to better navigate the function's landscape and avoid local minima. As a result, the choice of initial points significantly impacts both the convergence behavior and execution time of these algorithms.
- Changing the stopping criterion and absolute error bound significantly impacts the number of steps and execution times. A lower error bound results in more iterations and longer execution times, as the algorithm continues to refine the solution. Similarly, a stopping criterion that investigates smaller changes or minimizations will also result in more steps, aiming for a more accurate result.

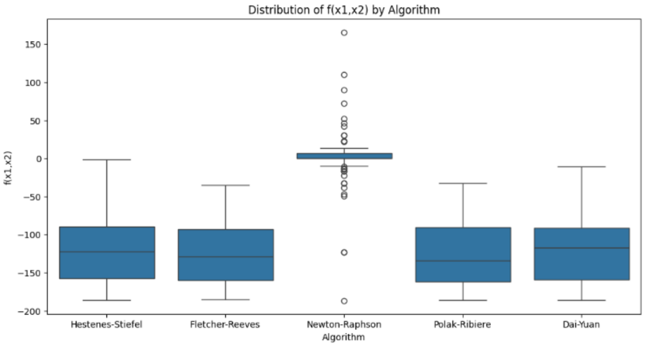


Fig 5: Distribution of F values at optimized point.

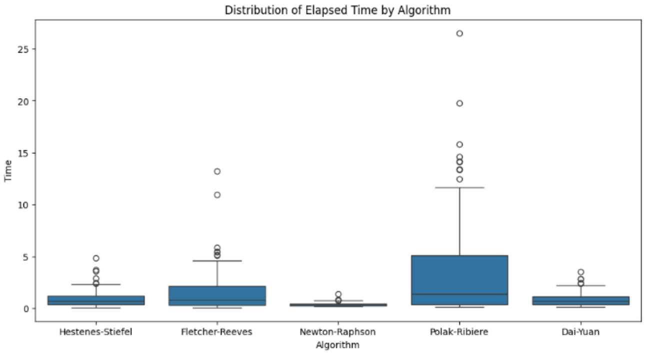


Fig 6: Distribution of elapsed time by each algorithm.

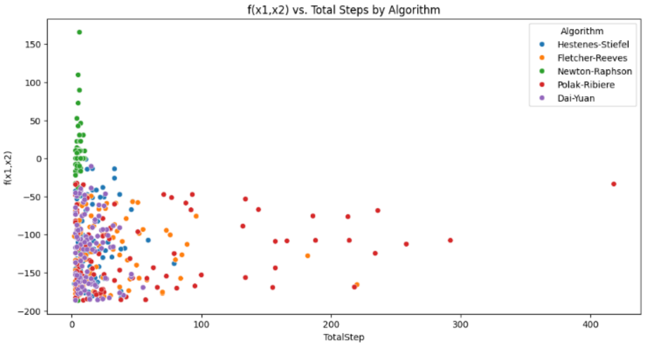


Fig 7: Total steps by each algorithm.

REFERENCES

[1] J. Yuan, H. -L. Liu, Y. -S. Ong and Z. He, "Indicator-Based Evolutionary Algorithm for Solving Constrained Multiobjective Optimization Problems," in IEEE Transactions on Evolutionary Computation, vol. 26, no. 2, pp. 379-391, April 2022

[2] M. M. Ali, C. Khompatraporn, and Z. B. Zabinsky, “A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems,” Journal of Global Optimization, vol. 31, no. 4, pp. 635–672, Apr. 2005

[3] D. Feng, M. Sun, and X. Wang, “A family of conjugate gradient methods for large-scale nonlinear equations,” Journal of inequalities and applications, vol. 2017, no. 1, Sep. 2017

[4] https://github.com/berkayahi/shubert-function-optimization