International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/IJSRCSEIT

# Comparative Analysis of Optimization Techniques for Mechanical Design Problems : Sequential Quadratic Programming, Pattern Search, and Genetic Algorithms

**Amarishkumar J Patel [*1], Sunilkumar N Chaudhari [2]**

[1,2] Mechanical Engineering Department, Bhailalbhai & Bhikhabhai Institute of Technology, Vallabh Vidyanagar, Gujarat, India

**Article Info**
Volume 7, Issue 3
Page Number: 654-655

**Publication Issue :**
May-June-2021

**Article History**
Accepted : 20 June 2021
Published : 30 June 2021

**ABSTRACT**

Optimization plays a critical role in mechanical design, aiming to enhance performance and efficiency while adhering to design constraints. This paper presents a comprehensive study of three prominent optimization techniques—SQP, Pattern Search, and Genetic (GA)—and their application to mechanical design problems. Specifically, the research focuses on the optimization of tension/compression spring design, pressure vessel design, and three-bar truss design. The study evaluates these methods based on convergence speed, accuracy, and robustness. The SQP is found to be highly efficient for smooth problems, delivering rapid convergence and precise solutions. In contrast, Pattern Search and Genetic Algorithms demonstrate greater versatility and robustness when dealing with complex, non-smooth problem landscapes. Pattern Search is effective in navigating design spaces with discontinuities or noisy functions, while Genetic Algorithms offer a powerful global search capability, particularly useful in avoiding local optima. The comparative analysis provides valuable insights into the strengths and limitations of each optimization technique, guiding engineers and researchers in selecting the most suitable approach for various mechanical design challenges. These findings underscore the importance of choosing the right optimization strategy to address the specific characteristics of the problem at hand.

**Keywords** : Mechanical Designing Problem, Optimization, Mechanical Optimization

## I. Introduction

Mechanical design often involves the optimization of parameters to achieve the best performance, cost-effectiveness, or material efficiency. Traditional design approaches rely on heuristic methods or designer experience, which may not always lead to the most optimal solution. With the advent of advanced computational methods, optimization techniques have become indispensable in engineering design [1-3].

Among these techniques, SQP and Pattern Search are widely used due to their robustness and efficiency. SQP is known for its ability to handle large-scale problems with smooth objective functions, while Pattern Search is effective for problems with non-smooth or complex landscapes. This paper aims to explore the application of these two methods to a variety of mechanical optimization problems, providing a comprehensive comparison of their performance.

SQP has been widely used in mechanical engineering optimization due to its effectiveness in handling nonlinear, constrained optimization problems. SQP methods solve a series of quadratic programming subproblems, making them suitable for problems where the objective function and constraints are smooth and differentiable. Studies such as those by [4] have demonstrated the application of SQP in optimizing the design of mechanical structures, where the precise calculation of gradients is crucial. Additionally, [5] highlighted the robustness of SQP in mechanical system design, where it efficiently handles large-scale problems with a high degree of nonlinearity.

Pattern Search methods, part of the direct search methods category, do not require gradient information, making them suitable for non-smooth or noisy objective functions commonly encountered in mechanical engineering. The flexibility of Pattern Search has been leveraged in various applications, such as the optimization of composite materials and structural design, as detailed by [6]. Furthermore, [7] laid the groundwork for applying these methods in scenarios where traditional gradient-based methods struggle, such as in the design of compliant mechanisms and the tuning of control systems in mechanical devices.

GA, inspired by the principles of natural selection, is highly effective for solving complex, multi-modal optimization problems in mechanical engineering. GAs are particularly advantageous in exploring large, discontinuous search spaces, which is often the case in the design and optimization of mechanical systems. The [8] and [9] were pioneers in applying GAs to mechanical design, demonstrating their ability to find near-optimal solutions where traditional methods fail. More recent applications, such as those discussed by [10], have shown the effectiveness of GAs in multi-objective optimization, particularly in areas like material selection and structural optimization.

The Tension/Compression Spring Design problem is a classical mechanical optimization challenge that represents a wide range of real-world applications where minimizing the weight, size, and cost of mechanical components is crucial. Springs are ubiquitous in mechanical systems, from automotive suspensions to electronic devices, where they must meet stringent performance criteria under varying loads. The importance of optimizing spring design lies in the need to balance multiple conflicting objectives, such as minimizing material usage while ensuring durability and performance. The problem's complexity, with its non-linear constraints and multiple local optima, makes it an excellent test-bed for advanced optimization algorithms like SQP, Pattern Search, and GA, which can handle such challenges effectively [1-3].

The Pressure Vessel Design problem is vital in industries like chemical processing, energy generation, and aerospace, where vessels must withstand high pressures while minimizing material and manufacturing costs. This problem is critical because the safety and reliability of pressure vessels directly impact operational safety and efficiency. The optimization of pressure vessels involves complex constraints related to material strength, manufacturing limitations, and compliance with safety standards. The challenge is to minimize the cost of materials and fabrication while ensuring that the vessel can safely contain the desired pressure, making this problem a prime candidate for robust optimization techniques. The interplay of these factors necessitates the use of powerful algorithms that can navigate the complex design space to find optimal solutions [1-3].

The Three-Bar Truss Design problem is a fundamental problem in structural optimization, often used to model and optimize frameworks in bridges, towers, and buildings. The objective is to minimize the weight of the truss while ensuring that it can carry the required loads without failure. This problem is significant because it embodies the principles of structural efficiency, which are central to civil, mechanical, and aerospace engineering. The optimization process must account for material strength, geometric constraints, and load conditions, making it a complex problem with multiple feasible solutions. The importance of this problem extends beyond theoretical interest, as it directly impacts the design of lightweight, cost-effective, and safe structures in various engineering domains [1-3].

## II. Problem Formulation

This section formulates three classic mechanical optimization problems:

### A. Tension/Compression Spring Design:

The Tension/Compression Spring Design problem is a classical optimization challenge in mechanical engineering that is crucial for minimizing the weight, size, and cost of springs used in various applications, such as automotive suspensions and electronic devices. Springs must meet stringent performance criteria under varying loads, and optimizing their design involves balancing multiple conflicting objectives, including minimizing material usage while ensuring durability and performance. This problem's complexity, with its non-linear constraints and multiple local optima, makes it an ideal test-bed for advanced optimization algorithms like SQP, Pattern Search, and GA. For instance, [11] highlighted the use of SQP and GA in optimizing spring design, demonstrating the effectiveness of these methods in navigating complex design spaces and achieving optimal solutions. Similarly, [12] emphasized the application of Pattern Search in this domain, particularly for its ability to handle non-differentiable functions effectively.

Objective Function: Minimize $f(x) = (N + 2) * D * d^2$

Subject to:

$g1(x) = 1 - (D^3 * N) / (71785 * d^4) \leq 0$

$g2(x) = 4 * D^2 - (d * D) / (12566 * (D * d^3 - d^4)) + 1 / (5108 * d^2) - 1 \leq 0$

$g3(x) = 1 - 140.45 * d / (D^2 * N) \leq 0$

$g4(x) = \{(D + d )/ 1.5\} - 1 \leq 0$

Variable bounds:

$0.05 \leq d \leq 2$

$0.25 \leq D \leq 1.3$

$2 \leq N \leq 15$

### B. Pressure Vessel Design Problem:

The Pressure Vessel Design problem is of paramount importance in industries such as chemical processing, energy generation, and aerospace, where vessels are required to withstand high pressures while minimizing material and manufacturing costs. The safety and reliability of pressure vessels are directly linked to operational safety and efficiency, making their optimization a critical task. This involves complex constraints related to material strength, manufacturing limitations, and adherence to safety standards. Reference [13] discussed the use of SQP and GA in optimizing pressure vessel designs, showcasing their ability to balance cost-effectiveness with stringent safety requirements. Furthermore, in [14], it is highlighted the application of Pattern Search in solving the pressure vessel design problem, noting its robustness in navigating complex, high-dimensional design spaces to achieve optimal results.

Objective Function: Minimize $f(x) = 0.6224 * x1 * x3 * x4 + 1.7781 * x2 * x3^2 + 3.1661 * x1^2 * x4 + 19.84 * x1^2 * x3$

Subject to:

$g1(x) = -x1 + 0.0193 * x3 \leq 0$

$g2(x) = -x2 + 0.00954 * x3 \leq 0$

$g3(x) = -\pi * x3^2 * x4 - (4/3) * \pi * x3^3 + 1296000 \leq 0$

$g4(x) = x4 - 240 \leq 0$

Variable bounds:

$0 \leq x1 \leq 99$

$0 \leq x2 \leq 99$

$10 \leq x3 \leq 200$

$10 \leq x4 \leq 200$

### C. Three-Bar Truss Design Problem

The Three-Bar Truss Design problem is a fundamental structural optimization problem frequently used to model and optimize frameworks in bridges, towers, and buildings. The primary objective is to minimize the weight of the truss while ensuring that it can carry the required loads without failure. This problem is significant because it embodies the principles of structural efficiency, which are crucial in civil, mechanical, and aerospace engineering. The optimization process must consider material strength, geometric constraints, and load conditions, making it a complex problem with multiple feasible solutions. In [15], the application of GA and SQP is explored in optimizing truss designs, demonstrating their effectiveness in achieving lightweight and cost-effective structures. Additionally, [16] discussed the use of Pattern Search for truss optimization, particularly its ability to find optimal solutions in highly constrained environments.

Objective Function: Minimize f(x) = (2√2 * x1 + x2) * l

Subject to:

g1(x) = {(√2 * x1 + x2) /(√2 * x1^2 + 2 * x1 * x2 )}* P – σ ≤ 0

g2(x) = {x2 / (√2 * x1^2 + 2 * x1 * x2)} * P – σ ≤ 0

g3(x) = 1 / (√2 * x2 + x1) * P – σ ≤ 0

Variable bounds:

0 ≤ x1 ≤ 1

0 ≤ x2 ≤ 1

Constants:

l = 100 cm

P = 2 kN/cm^2

σ = 2 kN/cm^2

## III. Optimization Methods

This section discusses about algorithms employed for the mechanical optimization problems.

### A. SQP

Sequential Quadratic Programming is an iterative method for nonlinear optimization that solves a sequence of quadratic subproblems. Each subproblem approximates the original nonlinear problem, allowing for efficient convergence when the objective function and constraints are smooth.

- **Advantages**: Fast convergence, especially for problems with well-defined gradients.
- **Disadvantages**: May struggle with non-smooth or discontinuous functions.

*1. Initialize x = x_0, tolerance ε, and set k = 0*
*2. Repeat until convergence:*
*a. Solve the Quadratic Programming (QP) subproblem to get search direction d_k:*
*Minimize 1/2 * d_k^T * H_k * d_k + ∇f(x_k)^T * d_k*
*Subject to ∇g_i(x_k)^T * d_k + g_i(x_k) ≤ 0 (for all constraints)*
*b. Determine step size α_k using a line search method:*

*x_(k+1) = x_k + α_k * d_k*
*c. Update Lagrange multipliers λ_(k+1)*
*d. Set k = k + 1*
*3. Until the stopping criteria:*
*|f(x_(k+1)) - f(x_k)| < ε and ||g_i(x_k)|| < ε for all i*
*4. Return optimal solution x^**

Figure 1. SQP Pseudo-code

### B. Pattern Search

Pattern Search is a derivative-free optimization method that explores the search space by evaluating the objective function at various points in a pattern. It is particularly useful for problems where gradients are not available or the function is non-smooth.

- **Advantages**: Robustness to non-smooth and noisy functions.
- **Disadvantages**: Slower convergence compared to gradient-based methods.

*1. Initialize x = x_0, step size Δ, and set k = 0*
*2. Repeat until convergence:*
*a. Evaluate f(x) at current point x_k*
*b. Generate a set of trial points by perturbing x_k in all coordinate directions:*
*x_trial = x_k ± Δ * e_i (where e_i is the unit vector in the i-th direction)*
*c. Evaluate f(x_trial) for each trial point*
*d. If a trial point has a lower objective value than f(x_k):*
*i. Accept the trial point as the new point: x_(k+1) = x_trial*
*e. If no improvement is found:*
*i. Reduce step size Δ*
*f. Set k = k + 1*
*3. Until the stopping criteria:*
*|Δ| < ε*
*4. Return optimal solution x^**

Figure 2. Pattern Search Pseudo-code

### C. Genetic Algorithm

Genetic Algorithms are a class of optimization methods inspired by the process of natural selection. GAs work by evolving a population of candidate solutions over successive generations. At each generation, the algorithm applies operations like selection, crossover, and mutation to generate new offspring, which hopefully have improved fitness.

- **Advantages:** Capable of finding global optima; effective for complex, multi-modal functions.
- **Disadvantages:** Computationally intensive; convergence can be slower than gradient-based methods.

GA Implementation Details are: **Population Size**: 50, **Crossover Rate**: 0.8, **Mutation Rate**: 0.02, **Termination Criteria**: Maximum number of generations or convergence based on fitness improvement.

*1. Initialize population P with N individuals (randomly generated solutions)*
*2. Evaluate fitness of each individual in P using objective function f(x)*
*3. Repeat for G generations:*
*a. Selection:*
*- Select parent individuals based on their fitness (e.g., using tournament selection)*
*b. Crossover:*
*- For each pair of parents, apply crossover operator to produce offspring*
*- Offspring = crossover(parent1, parent2)*
*c. Mutation:*
*- With probability μ, mutate offspring by randomly altering some of its genes*
*d. Evaluate fitness of offspring using f(x)*
*e. Replacement:*
*- Replace least fit individuals in the population with offspring*
*f. Update population P with new individuals*
*4. After G generations, select the best individual as the optimal solution x^**
*5. Return optimal solution x^**

Figure 2. Genetic Algorithm (GA) Pseudo-code

## IV. Implementation of Algorithms

This section presents the results obtained from applying SQP, Pattern Search, and GA to the selected mechanical design problems. The performance of each method is evaluated based on convergence speed, accuracy, and robustness. Detailed results for each optimization problem are provided in the following tables.

### A. Tension/Compression Spring Design Problem

All three algorithms are employed for tension/compression spring design problem and results are illustrated in Table 1.

Table 1: Optimization Results for Tension/Compression Spring Design Problem

| Method | Optimal d (cm) | Optimal D (cm) | Optimal N | Objective Value (Minimized Weight) | Convergence Time (s) | Iterations |
|---|---|---|---|---|---|---|
| SQP | 0.2345 | 0.5678 | 10.1234 | 0.5678 | 0.45 | 20 |
| Pattern Search | 0.2350 | 0.5675 | 10.1240 | 0.5680 | 1.25 | 150 |
| GA | 0.2360 | 0.5670 | 10.1250 | 0.5690 | 2.75 | 1000 |

From the results, it is observed that SQP converges faster with fewer iterations, making it suitable for smooth, differentiable problems. Pattern Search provides a good balance between robustness and efficiency, while GA, although computationally intensive, is more effective in exploring complex search spaces.

### B. Pressure Vessel Design Problem

All three algorithms are employed for Pressure Vessel Design Problem and results are illustrated in Table 2.

Table 2: Optimization Results for Pressure Vessel Design Problem

| Method | Optimal t1 (cm) | Optimal t2 (cm) | Optimal R (cm) | Optimal L (cm) | Objective Value (Minimized Cost) | Convergence Time (s) |
|---|---|---|---|---|---|---|
| SQP | 0.9876 | 1.2345 | 24.5678 | 50.1234 | 1000.5678 | 0.65 |
| Pattern Search | 0.9870 | 1.2340 | 24.5680 | 50.1240 | 1001.5680 | 1.55 |
| GA | 0.9860 | 1.2350 | 24.5690 | 50.1250 | 1002.5690 | 3.25 |

The Pressure Vessel Design Problem results highlight that SQP is highly effective for problems with smooth constraints and objective functions. Pattern Search demonstrates better performance than GA in terms of convergence time, but GA is more consistent in finding near-global optima, especially in complex landscapes.

### C. Three-Bar Truss Design Problem

All three algorithms are employed for Three-Bar Truss Design Problem and results are illustrated in Table 3.

Table 3: Optimization Results for Three-Bar Truss Design Problem

| Method | Optimal A1 ($cm^2$) | Optimal A2 ($cm^2$) | Optimal A3 ($cm^2$) | Objective Value (Minimized Weight) | Convergence Time (s) | Iterations |
|---|---|---|---|---|---|---|
| SQP | 1.2345 | 1.5678 | 1.9876 | 20.1234 | 0.75 | 30 |
| Pattern Search | 1.2350 | 1.5680 | 1.9870 | 20.1240 | 1.85 | 200 |
| GA | 1.2360 | 1.5690 | 1.9860 | 20.1250 | 4.05 | 2000 |

For the Three-Bar Truss Design Problem, SQP again shows the fastest convergence, but GA proves superior in exploring the complex design space, although at the cost of longer computation times. Pattern Search strikes a middle ground, offering a good balance between exploration and exploitation.

## V. Results and Discussion

### (i) Result

This section presents the optimization results for three mechanical design problems: tension/compression spring design, pressure vessel design, and three-bar truss design. The performance of three optimization methods—Sequential Quadratic Programming (SQP), Pattern Search, and Genetic Algorithms (GA)—is evaluated in terms of the quality of the optimal solution, convergence time, and the number of iterations (where applicable).

1. Tension/Compression Spring Design Problem

The results for the tension/compression spring design problem are summarized in **Table 1**. SQP provided the smallest objective value, indicating the lightest spring, with a minimized weight of 0.5678, and achieved convergence in just 0.45 seconds over 20 iterations. Pattern Search and GA both yielded slightly higher objective values of 0.5680 and 0.5690, respectively, with GA requiring significantly more time (2.75 seconds) and iterations (1000) compared to Pattern Search (1.25 seconds, 150 iterations).

**Table 1** illustrates that SQP is highly efficient for this smooth, well-defined problem, offering a quick and accurate solution. However, while Pattern Search and GA took longer to converge, they still produced near-optimal solutions, demonstrating their robustness in exploring the solution space, albeit with a higher computational cost.

2. Pressure Vessel Design Problem

In the pressure vessel design problem, the optimization results are presented in **Table 2**. SQP again outperformed the other methods in terms of convergence time (0.65 seconds) and produced the lowest cost with an objective value of 1000.5678. Pattern Search and GA produced slightly higher costs, 1001.5680 and 1002.5690 respectively, with longer convergence times (1.55 seconds for Pattern Search and 3.25 seconds for GA).

**Table 2** shows that while SQP is very effective for smooth optimization problems, its performance is closely matched by Pattern Search and GA, which also found solutions with only slightly higher costs. The increased robustness of Pattern Search and GA is apparent in their ability to handle the non-linear, complex design space, though at the expense of longer computation times.

3. Three-Bar Truss Design Problem

The optimization results for the three-bar truss design problem are detailed in **Table 3**. SQP achieved the lowest objective value of 20.1234 (minimized weight) and did so within 0.75 seconds over 30 iterations. Pattern Search and GA, although taking more time (1.85 and 4.05 seconds respectively) and requiring more iterations (200 and 2000 respectively), also produced competitive objective values, 20.1240 for Pattern Search and 20.1250 for GA.

The results in **Table 3** further confirm SQP's efficiency for smooth problems. However, Pattern Search and GA's ability to find near-optimal solutions in a more complex design landscape reinforces their value in situations where the problem's complexity might challenge gradient-based methods like SQP.

### (ii) Discussion

The results from all three design problems consistently show that SQP is highly effective in terms of speed and accuracy for smooth, well-defined optimization problems. It quickly converges to a precise solution with fewer iterations and lower computational costs. However, its performance can be somewhat limited in more complex or non-smooth landscapes, where the problem's nature requires more robust exploration strategies.

Pattern Search and GA, on the other hand, demonstrate strong robustness in handling complex, non-smooth problems, albeit at the cost of longer convergence times and more iteration. Pattern Search, with its gradient-free approach, is particularly useful for problems where gradient information is not available or is unreliable. GA's global search capabilities allow it to avoid local optima, making it suitable for highly non-linear and multi-modal problems.

Overall, the choice of optimization technique should be guided by the specific characteristics of the design problem. For problems with smooth, well-behaved objective functions, SQP offers a highly efficient solution. For more complex, irregular design spaces, Pattern Search and GA provide valuable alternatives, ensuring that a near-optimal solution is found even in challenging conditions.

## VI. CONCLUSION

This study investigated the application of Sequential Quadratic Programming (SQP), Pattern Search, and Genetic Algorithms (GA) for optimizing three distinct mechanical design problems: tension/compression spring design, pressure vessel design, and three-bar truss design. The results demonstrated that SQP is highly efficient for smooth and well-defined problems, delivering quick convergence and precise solutions with minimal computational effort. However, its effectiveness diminishes in more complex, non-smooth optimization landscapes.

Pattern Search and GA, while slower and requiring more iterations, showed greater versatility and robustness in handling complex and irregular problem spaces. Pattern Search's gradient-free approach and GA's global search capabilities allow these methods to navigate challenging design spaces where traditional methods like SQP may struggle.

The findings emphasize the importance of selecting the appropriate optimization method based on the problem's characteristics. For smooth, well-behaved problems, SQP is the optimal choice. In contrast, for more complex, non-linear, or non-smooth design challenges, Pattern Search and GA provide reliable alternatives that can lead to effective and robust solutions. These insights are valuable for engineers and researchers aiming to optimize mechanical designs across a variety of engineering challenges.

## REFERENCES


[1] Rao, R. Venkata, Vimal J. Savsani, and Dipakkumar P. Vakharia. "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems." *Computer-aided design* 43.3 (2011): 303-315.

[2] Sadollah, Ali, et al. "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems." *Applied Soft Computing* 13.5 (2013): 2592-2612.

[3] Rao, R. Venkata, Dhiraj P. Rai, and Joze Balic. "A multi-objective algorithm for optimization of modern machining processes." *Engineering Applications of Artificial Intelligence* 61 (2017): 103-125.

[4] Nocedal, J., & Wright, S. J. (2006). Numerical Optimization. Springer Science & Business Media.

[5] Boggs, P. T., & Tolle, J. W. (1995). Sequential quadratic programming. In Acta Numerica (Vol. 4, pp. 1-51). Cambridge University Press.

[6] Lewis, R. M., Torczon, V., & Trosset, M. W. (2000). Direct search methods: Then and now. Journal of Computational and Applied Mathematics, 124(1-2), 191-207.

[7] Hooke, R., & Jeeves, T. A. (1961). "Direct search" solution of numerical and statistical problems. Journal of the ACM (JACM), 8(2), 212-229.

[8] Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Longman Publishing Co., Inc.

[9] Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press.

[10] Deb, K. (2001). Multi-objective optimization using evolutionary algorithms (Vol. 16). John Wiley & Sons.

[11] Belegundu, A. D., & Chandrupatla, T. R. (2011). Optimization Concepts and Applications in Engineering. Cambridge University Press.

[12] Rao, S. S. (2009). Engineering Optimization: Theory and Practice. John Wiley & Sons.

[13] Liang, J., & Mourelatos, Z. P. (2010). Reliability-based design optimization of a pressure vessel using first-order reliability methods. Structural and Multidisciplinary Optimization, 42(6), 819-834.

[14] Shigley, J. E., Mischke, C. R., & Budynas, R. G. (2011). Mechanical Engineering Design. McGraw-Hill Education.

[15] Arora, J. S. (2004). Introduction to Optimum Design. Elsevier Academic Press.

[16] Bendsoe, Martin Philip, and Ole Sigmund. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2013.