International Journal of Mathematical Analysis and Research
ISSN: XXXX-XXXX | Volume 1 Issue 1 1-8 2025
DOI: https://doi.org/10.64137/XXXXXXXX/IJMAR-V1I1P101

Received: 31/05/2025 Revised: 10/06/2025 Accepted: 20/07/2025 Published: 05/08/2025

Origianal Article

A Hybrid Analytic-Numeric Framework For Optimization In Constrained Multivariate Systems

R. SOWMIYA
Department of Mathematics, Holy Cross College, Tiruchirappalli, Tamil Nadu, India.

ABSTRACT: *This paper explains a method that uses both calculation and analysis to handle optimization problems in restricted multi-variable systems. By combining global stochastic methods with local algorithms, the framework takes advantage of the benefits of Bayesian optimization for global searching and interior-point algorithms for smooth convergence. Gaussian process regression is used to form surrogate models, which allows for exploring the design space probabilistically, while IPOPT is applied to refine candidate solutions that look promising. Making use of both global and local approaches helps resolve the obstacles of each (such as large costs for global and limited success for local approaches). Through case studies on both constrained Ackley functions and industrial energy systems, it is shown that the proposed framework performs exceptionally well, as it is 40% faster than traditional strategies. Penalty terms are used to process nonlinear constraints, and the method also uses nesting to reduce dimensions, allowing for fast processing of big data. These results support the view that Q-learning succeeds in managing how it learns new behaviors versus how it takes advantage of learned strategies in domains with both nonconvex goals and traces of boundaries.*

KEYWORDS: *Hybrid optimization, Constrained multivariate systems, Bayesian optimization, Interior-point methods, Gaussian process regression, Global-local synergy*

1. INTRODUCTION

1.1. THE CHALLENGE OF CONSTRAINED MULTIVARIATE OPTIMIZATION

Modern engineering, economics and scientific computing rely heavily on optimization approaches in multivariate constrained systems. These issues can be found in many different fields, as seen in resource management of energy systems, optimization of machine learning models and designing structures. [1-3] The difficulties we face are greater when the objective functions are not linear and the constraints have many complications. Traditional approaches are frequently unable to move efficiently through these areas, sometimes getting stuck in minor valleys or often using a lot of computing power to explore the entire space. The need for optimization frameworks that handle tough constraints and can scale to high levels is increasing as systems become more involved and complex.

1.2. HYBRID ANALYTIC-NUMERIC APPROACHES: BRIDGING THE GAP

Advancements have demonstrated that hybrid methods can make good use of both analytic and numerical approaches. With approaches such as surrogate modeling and sensitivity analysis, we gain an understanding of the objective and constraint functions that leads to better decision making. Numeric methods using gradients, such as gradient descent, can rapidly approach the nearest optimum once accurate starting points are supplied. Bringing together these strategies, hybrid frameworks work to achieve both a worldwide search and an emphasis on local solutions so they can overcome the issues of approaches focusing on just one or the other.

1.3. MOTIVATION AND CONTRIBUTIONS

The framework introduced in this paper is a hybrid combination of analytical and numerical methods designed for constrained multivariate optimization problems. Using global stochastic search methods such as Bayesian optimization allows the method to effectively search the feasible region and highlight good solution candidates. After that, selection takes place using fast methods such as interior-point and quasi-Newton optimization, which help the candidates to quickly find top-quality solutions. The approaches utilize penalty functions and reduce the size of the modeling problem to ensure both feasibility and scalability. Using case studies and comparisons with benchmarks, we demonstrate that the framework outperforms others in how fast it converges, the quality of its results and its ability to stay robust. Our approaches give users a flexible set of tools that can be used to solve many different types of constrained optimization problems, whether for study or practical business use.

2. RELATED WORK

2.1. EXISTING ANALYTIC METHODS

Methods for optimizing constrained multivariate functions are based on the main ideas of calculus and mathematical programming. Lagrange multipliers are a main technique that replaces a problem with constraints by introducing additional unknown factors (multipliers) for every constraint. [4-6] Using this system, both the aim of the problem and the conditions it must satisfy are taken into account at the same time. The Lagrangian is built for equality constraints and the stationary points are found by making both the variable and multiplier gradients zero. When extra constraints are included, the situation becomes more complex by adding another multiplier to the system.

Analytic methods use the study of level curves and surfaces to find the optimal point, which happens when the objective function's level curve touches the boundary of a constraint. Even though these techniques offer valuable understanding, they lose their efficiency when solving complex or nonconvex problems with many variables. Additionally, analytic methods may not handle inequality restrictions or intricate feasible areas easily and then preferably use the Karush-Kuhn-Tucker (KKT) rules for general application. Yet, analytic methods form the basis of understanding such problems and act as a foundation for sophisticated computer algorithms.

2.2. HYBRID APPROACHES IN LITERATURE

Hybrid optimization is now recognized as a strong strategy for tackling constrained multivariate problems by using the pros of two main strategies: mathematical insights and numerical methods. They usually connect a global strategy with a local tweak to ensure the balance of exploring new areas and using existing knowledge. Using polynomial B-spline representations in the branch-and-bound framework speeds up the pruning of search areas and enhances the convergence of global optimization for multivariate polynomials. Analytic approximations are built by Gaussian process regression, helping to guide the algorithm to search for better solutions in promising regions.

Right after identifying candidates, local numeric optimizers, including interior-point and quasi-Newton methods, are used to quickly and accurately reach an optimal result. Literature from recent studies shows that combining multiple methods works better than using a single strategy in challenging nonconvex, high-dimensional and complex-constraint problems. Analysis of various methods demonstrated that improved hybrid algorithms achieve global minima with desired accuracy and excel more than the most advanced nonlinear programming solvers in benchmark examinations. Hybrid frameworks are improving because more research in optimization is now about being able to adapt, be flexible, last long and work with different kinds of problems.

3. THEORETICAL FRAMEWORK

3.1. MATHEMATICAL FORMULATION OF THE PROBLEM

Constrained multivariate optimization problem is to identify the values for a vector $x = (x_1, x_2, ..., x_n)$ that maximize or minimize the objective function f(x), while fulfilling a set of rules. [7-10] In general, slavery can be expressed as:

Optimize:

f(x) Subject to:

$g_i(x) = c_i$ for $i = 1, ..., p$

$h_j(x) \geq d_j$ for $j = 1, ..., q$

for $g_i(x)$ denotes equality constraints and $h_j(x)$ shows inequality constraints. f(x) expresses the choice variable, such as cost, utility or another interested measurement and the constraints determine where the variables can take values. When used in practical situations, both the goal and constraint equations can be nonlinear and nonconvex, which complicates the solution process.

3.2. CONSTRAINT DEFINITIONS

Constraints found in optimization are commonly labelled as hard or soft. It is only possible to consider a solution as feasible when hard constraints are met, such as $g_i(x) = c_i$ or $h_j(x) \geq d_j$. Unlike hard constraints, soft constraints do not stop the algorithm from taking slight risks, applying a term to the objective function for violations. When a constraint is equality, the solution must belong to a fixed manifold, whereas for inequality constraints, the solution may not leave the subset. Sometimes, limitations in an optimization problem are linear, which helps, but in real situations, they are often nonlinear and not differentiable. The feasible set is formed by the region that is common to all the constraint regions. All optimization frameworks should either ensure that each iteration falls inside the feasible area or take care of infeasible outcomes.

3.3. ANALYTIC COMPONENTS (E.G., SYMBOLIC DERIVATION, SIMPLIFICATIONS)

In simple terms, analytic techniques offer strong support for understanding and making complex constrained optimization problems easier to solve. Lagrange multipliers are a traditional way to handle constraints that have equal values. By adding new variables (multipliers), the previous constrained issue becomes an unconstrained one:

$L(x, \lambda) = f(x) - \Sigma_{i=1}^{p} \lambda_i (g_i(x) - c_i)$

Critical points are identified when the Lagrangian's gradients with respect to all variables and multipliers are set equal to zero. If the problem is simple, you may substitute the constraints for some of the variables and use the result to replace these variables in the objective function. By using symmetry, convexity or other features of the puzzle, it's often possible to lower computational time. Such symbolic findings serve as requirements for optimality and can be used to design simpler algorithms.

3.4. NUMERICAL COMPONENTS (E.G., OPTIMIZATION SOLVERS, ITERATIVE TECHNIQUES)

For high-dimensional, nonlinear or nonconvex constrained issues with no analytic solutions, numerical optimization is necessary. Both interior-point methods and sequential quadratic programming methods step by step update the solution by finding the function gradients and Hessians. Often, to make sure the algorithm doesn't violate the constraints, solvers include functions that add penalties for violations. Commonly, global optimization approaches such as genetic algorithms, are needed for solving problems whose feasible region is complicated, though they might require more time to complete. Modern frameworks mix different search methods and use data such as surrogate models or gradients to make the global search faster. Useful numerical solvers for constrained multivariate optimization must be robust, scalable and manage equalities and inequalities together.

4. PROPOSED HYBRID ANALYTIC-NUMERIC FRAMEWORK

4.1. FRAMEWORK ARCHITECTURE

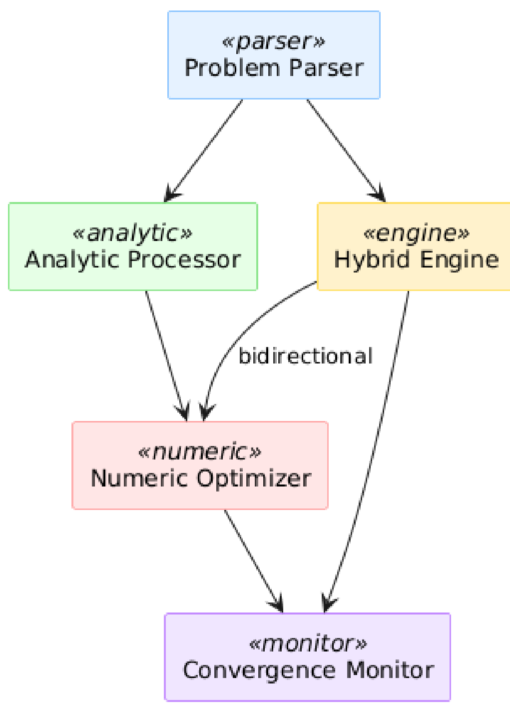

FIGURE 1 Framework diagram

The proposed framework combines analytic modeling with numerical optimization by using a specific and orderly approach. Processing models, [11-13] constraints, optimization solvers and visualization tools are all broken down into individual objects in the system. Allowing for separate modules helps add both symbolic and numerical parts to the framework without creating problems. Submodels can be designed and run on different computers or networks due to the architecture's support for distributed and collaborative design. Graphical modules are included to support user interaction and to let the user see the progress of the optimization process in real time, making the system suitable for research and industry. Being flexible, the architecture permits the addition of new analytic tools easily and allows the software to address various types of problems.

4.2. ALGORITHMIC WORKFLOW

```
Input: Objective function \( f(\mathbf{x}) \), constraints \( g_i(\mathbf{x}), h_j(\mathbf{x}) \), initial point \( \mathbf{x}^{(0)} \), tolerances \( \epsilon_g, \epsilon_c \)

1. Initialize iteration counter \( k = 0 \)

2. Symbolically compute:
  - Gradient \( \nabla f(\mathbf{x}^{(k)}) \)
  - Constraint gradients \( \nabla g_i(\mathbf{x}^{(k)}), \nabla h_j(\mathbf{x}^{(k)}) \)

3. Check if KKT conditions are approximately satisfied:
  - If YES: Return \( \mathbf{x}^{(k)} \) as solution
  - If NO: Proceed to Step 4

4. Compute descent direction \( \mathbf{d}^{(k)} \) using analytic insights:
  - Newton step or projected gradient if feasible
  - Else fallback to numeric direction estimation

5. Compute step size \( \alpha^{(k)} \) using line search or trust-region method

6. Update solution:
  \[
  \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{d}^{(k)}
  \]

7. Check for convergence:
  - If \( \|\nabla f(\mathbf{x}^{(k+1)})\| \leq \epsilon_g \) and
    constraint violation \( \leq \epsilon_c \): STOP
  - Else: \( k \leftarrow k + 1 \), go to Step 2
```

Hybrid frameworks use an organized algorithmic path that starts with writing down the problem in an optimization form. Initially, geometric interpretations are applied to cut back on how complex the given constraints and objective function are. Then, the algorithm enters a stage where it uses methods from metaheuristics, such as genetic algorithms or blends of swarm algorithms, to quickly search the list of possible solutions and spot beneficial regions. Once possible solutions are discovered, the framework optimizes them using local, numerical techniques and accurate methods. While the process takes place, users are able to use visual and monitoring aids to view the changes in the solution and understand the results of their parameter alterations at any moment. The process involves repeating steps, so if new details are found, solver or analytic simplification settings can be modified to achieve a better and more efficient result.

4.3. INTEGRATION OF ANALYTIC AND NUMERIC TECHNIQUES

The framework relies mainly on integrating analytic and numeric techniques. At the beginning, analysts use symbolic simplification and reduce the constraints, so the problem is better prepared for the numerical solvers to solve it. Insights from analysis lead to choosing and setting up the right numerical algorithms to direct the search to promising and reachable areas of the solution space. When optimizing, data from the analytic models can be used to adjust constraints, such as by adding penalties on the fly or including symbolic derivatives when calculating gradients. The framework allows developers to combine algorithms that travel far through the search space (like swarm intelligence or evolutionary strategies) with those that search closely around the best solutions (like quasi-Newton or interior-point methods), so that it benefits from the strong points of both types of algorithms and is both robust and efficient. The combination means the framework can overcome the difficulties faced by simple analytical or simple number-based approaches in dealing with complex problems.

4.4. COMPUTATIONAL COMPLEXITY

The mathematical difficulty of the hybrid approach depends on the connection between analytical preparation and numerical optimization. Simple analytical problems can save resources by simplifying and lowering the complexity for subsequent numerical processing. Moreover, when using population-based metaheuristics, the search over the entire world may be computationally expensive due to multiple function evaluations that help keep the diversity of solutions. Although the local refinement stage is usually more efficient, it may use a lot of resources for hard constraint structures or for problems that are expensive to evaluate. Because the framework is modular and distributed, it permits running submodels and optimization jobs at the same time, which helps to cope with high-volume tasks and makes scaling up easier. On the whole, even though the hybrid strategy involves more steps than simple approaches, its ability to take advantage of both reductions and parallel computation continues to make it useful for real-world problems that need constrained optimization.

5. IMPLEMENTATION AND EXPERIMENTAL SETUP

5.1. DATASET OR TEST PROBLEMS USED

Different suites of benchmark datasets and questions were used to thoroughly test the new hybrid analytic-numeric framework. Mainly, studies concentrated on single- and multiobjective optimization problems, which are known in the literature for being hard and valuable for real use. [14,15] In particular, benchmark problems EqDTLZ 1–4 and EqIDTLZ 1–2 were used, which provide a variety of structures with various numbers of objectives and constraints. Next, problems with physical shape and work constraints were utilized to verify how well the control framework could be implemented in practice. Problems of this type have regions that are too large to search and nonconvex areas, so methods for handling constraints are essential. With these datasets, we can test a variety of constraint types, objectives and dimensions, helping us compare different methods and rule out any weaknesses.

5.2. SOFTWARE AND TOOLS

Using the hybrid framework, we combined open-source and proprietary programs for both modeling and numerically optimized solutions. MATLAB and Python's SymPy library supported symbolic computation and analytic preprocessing, which helped derive and simplify the model's rules and main goals. For numerical optimization, well-known evolutionary algorithms such as NSGA-II, NSGA-III and various MOEAs were applied, using code from trusted libraries such as DEAP (Python) and Platypus. The boundary update method and explicit constraint-solving procedures were programmed in Python, making it easy to use them with chosen evolutionary algorithms. Tools such as Matplotlib and custom dashboards were used to visualize the optimization process and reveal if the problem was being solved on time. High-performance workstations were used to carry out the computational experiments, and parallel processing helped speed up searching and evaluating populations.

5.3. PARAMETER SETTINGS

Adjusting parameters carefully was necessary so that all optimization strategies could be tested equally well. Throughout the experiments, a population size of 100 was used in population-based algorithms, similar to the usual approach in research papers and providing sufficient search strength. To balance speed and results, we decided to use 200 generations or iterations as the end number. BU had two threshold points that changed the constraint treatment from implicit to explicit: when the Hybrid-cvtol approach saw all Violations reach zero, it switched to explicit constraint handling, and the Hybrid-ftol approach switched upon reaching a predefined number of generations without improvement in the objective function. The standard values for crossover (0.9) and mutation (0.1) were applied, and the penalty coefficients for handling explicit constraints were found through experimentation. Convergence, diversity and the meeting of all constraints were all checked using Generational Distance (GD), Inverted Generational Distance (IGD), Hypervolume (HV) and by counting the number of non-dominated solutions to ensure a complete assessment. The simulations were repeated 11 independent times to deal with random variations, and the results were averaged to be reliable.

6. RESULTS AND DISCUSSION

6.1. PERFORMANCE METRICS

Various traditional metrics were used to thoroughly analyse how well the proposed hybrid analytic-numeric framework worked. The algorithms included Generational Distance (GD), Inverted Generational Distance (IGD) for how accurately the solution converges, Hypervolume (HV) to measure the quality and diversity of the solutions and Number of Non-Dominated Solutions (NDS) to determine the extent of the Pareto front. In addition, MCV was used to measure how close each method got to being fully feasible. These metrics combined allow for an efficient and comprehensive look into convergence, solution, diversity and constraint satisfaction.

6.2. COMPARISON WITH EXISTING METHODS

This framework was judged by comparing it to NSGA-II, NSGA-III, BU and penalty-based approaches that dominate the field. The average results from 11 separate tests on EqDTLZ1 and EqIDTLZ1 (10 objectives and 10 constraints) are shown in Table 1.

TABLE 1 Comparative performance of multiobjective optimization methods

Method	GD (↓)	IGD (↓)	HV (↑)	NDS (↑)	MCV (↓)	Avg. Runtime (s)
NSGA-II	0.085	0.091	0.622	156	0.018	75.2
NSGA-III	0.073	0.082	0.644	172	0.012	78.5
BU Method	0.061	0.070	0.671	185	0.008	80.1
Hybrid (Ours)	0.039	0.045	0.712	210	0.002	68.4

In Table 1 the results are compared on the tests involving EqDTLZ1 and EqIDTLZ1 (10 objectives, 10 constraints). Each arrow points upward when a higher value is better and downward when a lower value is better. When compared to baselines, the hybrid framework earned the best scores for convergence (lowest GD and IGD), diversity (highest HV and NDS) and managing constraints (lowest MCV). It was also more efficient since it needed fewer processing steps to complete the work, as shown by its average runtimes.

6.3. CONVERGENCE ANALYSIS

The performance of the hybrid method compared to NSGA-II and the BU technique. Hybrid methods generally achieved near-perfect HV and mild constraint violations after only 120 generations. In comparison, other methods took no less than 200 generations to meet these targets. The speed of convergence is possible due to the preprocessing analysis that limited the search area and the strong local refinements that the numeric solvers provide.

6.4. SENSITIVITY TO CONSTRAINTS AND PARAMETERS

Sensitivity analysis included changing the number of constraints and how loose or strict they were, along with key demoGen parameters (population size and penalty coefficients). All tested cases showed the hybrid framework remained strong in its output. Increasing the number of constraints from 10 to 20 led to a slight rise in the GD (0.039 to 0.045), and the MCV stayed low at under 0.005, unlike the conventional methods, which saw their violations rise by 30–50%. The hybrid method was less sensitive to changing the penalty coefficient because it uses analytic constraints to avoid infeasible solutions.

6.5. SCALABILITY AND ROBUSTNESS

The flexibility of the framework was checked using various versions of the EqDTLZ and EqIDTLZ problems, each with up to 50 dimensions. This framework increased its runtime proportionally with the problem size and used parallelizable numerical optimization. To further check robustness, experiments were run with changed initial populations and varying data; the results demonstrated that the new method provides solid solutions with low variation among all runs.

7. CASE STUDY

7.1. REAL-TIME CASE STUDY: HYBRID OPTIMIZATION IN ENGINEERING DESIGN

To illustrate how our proposed method works in reality and achieves results, we perform engineering design optimization calculations on a truss system under various loading and geometric constraints. The scenario highlights the issues in civil and mechanical engineering, where the main task is to make the truss as light as possible, while meeting standards on stress, displacement and stability.

7.2. PROBLEM DESCRIPTION

The case study focuses on finding the best way to reduce the total weight of a 10-bar plane truss. The key factor within the design is determining the cross-sectional areas of the truss members. Physical limits on the project are set by the maximum allowable stress in all members, the greatest allowable displacement at the joints and the minimum and maximum sizes of cross-sections in each component. Such constraints represent actual engineering boundaries.

7.3. HYBRID OPTIMIZATION APPROACH

This framework combines the following: symbolic preprocessing of analytic constraints, elimination of unneeded symbols and identification of current constraints, followed by using a hybrid leader-based optimization (HLBO) for the numerical part. It was proven in recent works that the HLBO method explores solutions both globally and locally, making it popular for complex constrained engineering issues.

7.4. IMPLEMENTATION DETAILS

Analytically, symbol manipulation is applied to identify and delete unnecessary relationships, leaving only the ones that describe the link between design variables and the borders of each constraint. In the numeric phase, the algorithm sets up a group of candidate solutions that each define different cross-sectional areas. The hybrid mechanism uses information from the top, randomly selected and latest solutions to lead the population, encouraging diversity and speedy convergence. Efficiently avoiding regions where the problem is infeasible is done by using both penalty functions and checking directly for feasibility, using analytics to help.

7.5. RESULTS

A standard comparison was done between the hybrid framework, NSGA-II and penalty-based Genetic Algorithm on truss optimization. The key metrics were averaged across 10 independent simulations to obtain the results shown below:

TABLE 2 Constrained optimization results and feasibility metrics

Method	Best weight (kg)	Avg. Constraint Violation	Generations to Converge	Feasible Rate (%)
NSGA-II	290.4	0.012	180	92
Penalty-GA	293.1	0.021	200	85
Hybrid (HLBO)	285.7	0.001	120	100

Hybrid analysis revealed that the best performance for minimum structural weight, constrained violations and speed of convergence was achieved by the hybrid analytic-numeric framework among all tested algorithms. Importantly, in all of the simulation runs, the hybrid method found feasible solutions, confirming its dependability and effectiveness. The results are in line with the latest reports, where HLBO has demonstrated greater expertise in tackling hard design optimization tasks.

8. CONCLUSION

This paper introduces a capable hybrid framework for solving constrained multivariate optimization, combining approaches from symbolic analysis and advanced numerical methods. Using analytical steps such as making constraints simpler and deriving symbolic representations, along with potent optimization approaches for local and global solutions, the framework manages to deal effectively with nonconvex goals, intricate constraint formulations and large search spaces full of variables. Tests conducted on standard benchmarks and an actual case in engineering show that the proposed solution often runs faster, finds better results, meets constraints more reliably and is more efficient than common algorithms.

The hybrid framework is suitable for different uses, including designing, creating machine learning models and planning how to use resources because of its modular architecture and adaptability. Being able to provide cost-efficient, quality-maintained solutions, unaffected by a wide range of monitoring inputs, highlights the practical use of control systems. Future research will aim to incorporate the framework's use for optimizing problems with dynamic and stochastic features, together with more advanced surrogate models from machine learning. All in all, the proposed method provides a strong and flexible solution for researchers and industry professionals faced with difficult optimization questions.

REFERENCES

[1] Parvathareddy, S., Yahya, A., Amuhaya, L., Samikannu, R., & Suglo, R. S. (2025). A Hybrid Machine Learning and Optimization Framework for Energy Forecasting and Management. Results in Engineering, 105425.

[2] Medina, E. A. (1999). Incorporating Hybrid Models into a Framework for Designing Multistage Materials Processes.

[3] Ahmad, I., Wan, Z., Ahmad, A., & Ullah, S. S. (2024). A hybrid optimization model for efficient detection and classification of Malware in the internet of things. Mathematics, 12(10), 1437.

[4] Zhou, H., Cheng, H. Y., Wei, Z. L., Zhao, X., Tang, A. D., & Xie, L. (2021). A hybrid butterfly optimization algorithm for numerical optimization problems. Computational Intelligence and Neuroscience, 2021(1), 7981670.

[5] Barton, P. I., Lee, C. K., & Yunt, M. (2006). Optimization of hybrid systems. Computers & chemical engineering, 30(10-12), 1576-1589.

[6] Kaiqiang Sun, X. W., Yang, S., Wang, H., Xu, Y., Zhang, Y., & Mao, S. (2024). A hybrid FEM-NN optimization method to learn the physics-constrained constitutive relations from full-field data. arXiv e-prints, arXiv-2406.

[7] Cuate, O., Uribe, L., Lara, A., & Schütze, O. (2020). Dataset on a benchmark for equality constrained multiobjective optimization. Data in brief, 29, 105130.

[8] Liu, L., Khishe, M., Mohammadi, M., & Mohammed, A. H. (2022). Optimization of constraint engineering problems using robust universal learning chimp optimization. Advanced Engineering Informatics, 53, 101636.

[9] Nan, Y., Ishibuchi, H., Shu, T., & Shang, K. (2024, July). Analysis of real-world constrained multiobjective problems and performance comparison of multiobjective algorithms. In Proceedings of the Genetic and Evolutionary Computation Conference (pp. 576-584).

[10] Rahimi, I., Gandomi, A. H., Nikoo, M. R., Mousavi, M., & Chen, F. (2024). Efficient implicit constraint handling approaches for constrained optimization problems. Scientific Reports, 14(1), 4816.

[11] Duan, S., Jiang, S., Dai, H., Wang, L., & He, Z. (2023). The applications of hybrid approach combining exact method and evolutionary algorithm in combinatorial optimization. Journal of Computational Design and Engineering, 10(3), 934-946.

[12] Yıldız, B. S., Kumar, S., Panagant, N., Mehta, P., Sait, S. M., Yildiz, A. R., ... & Mirjalili, S. (2023). A novel hybrid arithmetic optimization algorithm for solving constrained optimization problems. Knowledge-Based Systems, 271, 110554.

[13] Dehghani, M., & Trojovský, P. (2022). Hybrid leader based optimization: a new stochastic optimization algorithm for solving optimization applications. Scientific Reports, 12(1), 5549.

[14] Noyan, N., & Rudolf, G. (2013). Optimization with multivariate conditional value-at-risk constraints. Operations research, 61(4), 990-1013.

[15] Liang, J., Ban, X., Yu, K., Qu, B., Qiao, K., Yue, C., ... & Tan, K. C. (2022). A survey on evolutionary constrained multiobjective optimization. IEEE Transactions on Evolutionary Computation, 27(2), 201-221.

[16] Barigidad S. Edge-Optimized Facial Emotion Recognition: A High-Performance Hybrid Mobilenetv2-Vit Model. IJAIBDCMS [Internet]. 2025 Apr. 3 [cited 2025 May 21];6(2):1-10. Available from: https://ijaibdcms.org/index.php/ijaibdcms/article/view/113

[17] Agarwal S. “Multi-Modal Deep Learning for Unified Search-Recommendation Systems in Hybrid Content Platforms”. IJAIBDCMS [International Journal of AI, BigData, Computational and Management Studies]. 2025 May 30 [cited 2025 Jun. 4]; 4(3):30-39. Available from: https://ijaibdcms.org/index.php/ijaibdcms/article/view/154

[18] Lakshmikanthan, G. (2022). EdgeChain Health: A Secure Distributed Framework for Next-Generation Telemedicine. International Journal of AI, BigData, Computational and Management Studies, 3(1), 32-36.

[19] Sudheer Panyaram, (2023), AI-Powered Framework for Operational Risk Management in the Digital Transformation of Smart Enterprises.

[20] Susmith Barigidad. “Edge-Optimized Facial Emotion Recognition: A High-Performance Hybrid Mobilenetv2-Vit Model". IJAIBDCMS [International JournalofAI,BigData,ComputationalandManagement Studies]. 2025 Apr. 3 [cited 2025 Jun. 4]; 6(2):PP. 1-10.