International Journal of Scientific Research and Management (IJSRM)
||Volume||14||Issue||02||Pages||690-698||2026||
Website: https://ijsrm.net ISSN (e): 2321-3418
DOI: 10.18535/ijsrm/v14i02.m01

Automated Bayesian Optimization for Parameter Tuning of DE and PSO in Constrained Engineering Design under a Fixed Evaluation Budget

Aurora Simoni[1], Blerina Boçi[2,*], Klodiana Bani[3]

[1] Department of Applied Mathematics, Faculty of Natural Science, University of Tirana, Albania

[2] Department of Mathematics, Faculty of Information Technology, University 'Aleksandër Moisiu', Durrës, Albania

[3] Department of Applied Mathematics, Faculty of Natural Science, University of Tirana, Albania

Abstract
Global optimization problems in engineering often require robust algorithmic solutions due to nonconvex and constrained objectives. This study presents a Bayesian Optimization framework using Gaussian process surrogates to tune the parameters of Differential Evolution (DE) and Particle Swarm Optimization (PSO) under a fixed evaluation budget. The approach is tested on three classical constrained design problems—the welded beam, pressure vessel, and compression spring—using a penalty formulation for constraints and early stopping for efficiency. SMBO-tuned configurations outperform or match default settings, achieving improved solution quality and stability across multiple independent trials. These results support the role of automated surrogate-based tuning in enhancing metaheuristic reliability for constrained engineering applications.

Keywords: DE, PSO, Bayesian optimization, parameter tuning, constrained design

1. Introduction

Global optimization methods are often required in engineering design, since many real-world problems are nonlinear, complex, and constrained [1, 2]. Within this context, metaheuristic algorithms have become increasingly important, with two of the most widely studied approaches being differential evolution (DE) and particle swarm optimization (PSO). DE [3] is an evolutionary algorithm characterized by relatively few control parameters, whereas PSO [4] is a swarm intelligence method that incorporates additional parameters but offers greater flexibility. Both algorithms are widely adopted because they cope effectively with nonconvex, black-box, and constrained objectives, making them powerful tools for global optimization [5]. However, their performance is highly sensitive to parameter settings such as population or particle size, iteration budgets, and control coefficients, whose default values seldom generalize well across problems.

This motivates the use of sequential model-based optimization (SMBO) [6], a Bayesian Optimization framework often implemented with Gaussian process surrogates [7], to systematically optimize the parameters of DE and PSO. The idea of surrogate-based global optimization originates with Bayesian formulations [8] and was first practically realized in the efficient global optimization (EGO) algorithm, which combined Gaussian processes with the expected improvement criterion [9]. Building on this foundation, the sequential parameter optimization (SPO) methodology integrated design of experiments, surrogate modelling, and sequential sampling to tune optimization algorithms [10]. Subsequent work extended SPO into variants such as SPO+ [11], while the SPOT toolbox [12] provided a practical framework for automated and interactive tuning. Finally, SMBO was formalized for general algorithm configuration, with SMAC addressing categorical parameters and multiple-instance settings [6]. Beyond algorithm configuration, Bayesian Optimization has also gained wide adoption in machine learning for hyperparameter tuning. Our earlier research employed Bayesian Optimization (BO) to improve regression models for salary prediction, confirming its effectiveness in achieving higher predictive accuracy than default baselines [13]. Building on these findings, the current study extends the application of Bayesian Optimization from machine learning hyperparameter tuning to optimizing the control parameters of DE and PSO in constrained engineering design applications.

In this work, we adopt a Gaussian process-based SMBO (GP-SMBO) approach [14]. GP-SMBO is not applied directly to the engineering design problems but operates at the meta-level [15], tuning the

parameters of DE and PSO to improve their effectiveness on constrained benchmarks such as the welded beam (WBD), pressure vessel (PVD), and compression spring (CSD) problems.

The main contributions of this study are fourfold. First, we design a budget-matched sequential model-based optimization (SMBO) tuning loop that systematically explores parameter configurations while enforcing equal effective objective evaluations to ensure fair comparison. Second, we employ a simple penalty formulation for constraints, allowing differential evolution (DE) and particle swarm optimization (PSO) to operate in their standard forms without specialized feasibility operators. Third, we introduce lightweight early-stopping criteria based on tolerance and patience thresholds, improving computational efficiency without compromising interpretability or reproducibility. Finally, we conduct a statistically grounded evaluation across multiple random seeds, applying nonparametric significance tests and descriptive summary statistics to assess stability and robustness. Collectively, these contributions establish a disciplined and reproducible SMBO framework. This framework enables the systematic tuning and fair comparison of DE and PSO in constrained engineering design problems under fixed computational budgets.

2. Literature Review

The use of population-based methods such as differential evolution (DE) and particle swarm optimization (PSO) in global optimization has become widespread, and both algorithms have been extensively applied to engineering and scientific problems over the last twenty-five years. Over the past two decades, DE has been extended through self-adaptive and strategy-adaptive variants that dynamically adjust control parameters during the run, improving robustness across problem classes [16, 17, 18]. Recently, DE has remained one of the most competitive metaheuristics for global optimization, with numerous variants applied to benchmark functions and engineering design problems. These advances also encompass hybridization strategies, where DE is combined with other metaheuristics or learning-based approaches to improve robustness, scalability, and convergence behavior [19, 20, 21].

In parallel, numerous modifications and hybridizations of PSO have been introduced to overcome premature convergence and enhance performance, including adaptive inertia weights and the constriction factor approach [22, 23], as well as hybridization methods designed to solve nonconvex problems [24] and integration strategies that combine PSO with other metaheuristics [25]. Beyond these advances, PSO continues to demonstrate adaptability and effectiveness across diverse fields, including engineering design, energy systems, and healthcare [26, 27]. Comparative analyses further highlight their complementary strengths, with DE often achieving superior solution accuracy and robustness, while PSO offers faster convergence and simpler implementation [28].

While numerous modifications of DE and PSO have been proposed, their performance remains highly sensitive to parameter settings such as population size, mutation or acceleration coefficients, and iteration budgets. Selecting these parameters appropriately is challenging, as default values rarely generalize well across different problems. Beyond pure Bayesian Optimization, hybrid approaches have been developed that integrate Gaussian processes (GP) with population-based methods. For instance, GP has been combined with PSO to approximate the fitness landscape and accelerate convergence while reducing the number of expensive function evaluations [29]. Similarly, GP has been integrated with differential evolution within an SMBO framework, where DE serves as a meta-optimizer for acquisition functions in complex or non-continuous parameter spaces [30]. Related work in AutoML has also demonstrated hybrid strategies, where evolutionary algorithms such as GA, DE, and CMA-ES are used to optimize BO acquisition functions for parameter search [31].

Earlier, Bayesian Optimization had already been explored for evolutionary algorithm parameter tuning [32], where an offline BO-based tuning strategy employing a hybrid kernel and an evolutionary algorithm was proposed, demonstrating improved performance on flow-shop scheduling benchmarks. Subsequent studies have proposed SMBO-based frameworks for tuning metaheuristics, such as the MetaTuner approach, which employs sequential optimization of perturbed regression models to identify effective parameter settings and assess parameter relevance [33]. Hybrid Bayesian–evolutionary optimization strategies have been introduced, demonstrating the benefits of combining BO with evolutionary search [34]. Further, [35] applied a Bayesian Optimization approach to tune genetic algorithm parameters for practical-oriented pickup and delivery problems, demonstrating that BO can effectively enhance evolutionary algorithm performance in real-world applications. More recently, this line of work was extended to a grouping genetic algorithm, where BO-based configuration improved solution quality in multi-depot pickup and delivery problems with time windows [36].

Their results showed significant improvements in global optimization performance, highlighting the continuing trend of combining surrogate modelling, hybridization, and adaptive strategies to strengthen evolutionary algorithms. The most recent applied research has demonstrated the integration of Bayesian Optimization into PSO to dynamically refine its hyperparameters in neural network fault diagnosis tasks [37].

Building on previous research, this work extends the sequential model-based optimization (SMBO)

framework by integrating Gaussian process surrogates to configure DE and PSO parameters for constrained engineering design problems under a fixed evaluation budget.

3. Methodology

3.1. Benchmark Design Problems

The effectiveness of sequential model-based optimization (SMBO) for parameter tuning of DE and PSO is evaluated using three classical constrained engineering design problems widely recognized as benchmarks in the optimization literature [2]. The welded beam design problem seeks to minimize fabrication cost while satisfying structural constraints on shear stress, bending stress, buckling load, and deflection. The pressure vessel design problem aims to reduce the total cost of materials, forming, and welding, subject to nonlinear constraints on thickness, volume, and stress. The compression spring design problem focuses on minimizing the spring's weight while meeting restrictions related to shear stress, surge frequency, and deflection. Collectively, these benchmarks reflect the nonlinear and highly constrained nature of real-world engineering design tasks and are widely employed in studies of constraint-handling techniques, including self-adaptive penalty formulations [38, 39].

For a constrained optimization problem:

$$Minimize\ f(x), x \in \mathbb{R}^n \tag{1}$$

$$Subject\ to: g_i(x) \leq 0, i = 1, 2, \dots m \tag{2}$$

$$h_j(x) = 0, j = 1, 2, \dots p \tag{3}$$

The penalized fitness is expressed as:

$$F(x) = f(x) + \sum_{i=1}^{m} \alpha_i \max(0, g_i(x))^2 + \sum_{j=1}^{p} \beta_j h_j(x)^2 \tag{4}$$

where $f(x)$ is the objective function, $g_i(x)$ are the inequality constraints, $h_j(x)$ are the equality constraints, α_i, β_j are the penalty coefficients that control the severity of constraint violations.

The constraints in the welded beam, pressure vessel, and compression spring design problems are handled using a static penalty method. Specifically, constraint violations are squared and scaled by a fixed coefficient ($\alpha = 10^6$). Penalty functions have long been employed in evolutionary optimization, starting from early quadratic formulations [40] and later reviewed extensively in surveys of constraint-handling methods [41].

3.2. Metaheuristic Algorithms

Differential evolution evolves a population of candidate solutions through mutation, crossover and selection, controlled primarily by the population size (N_p), mutation factor (F) and crossover rate (C_r). In our implementation, the DE/rand/1/bin strategy was used [16].

A ring-topology PSO with velocity clamping was adopted, where particle velocities were limited to a fraction of the search span to prevent divergence [42]. Its key parameters include the number of particles (N), inertia weight (w) and acceleration coefficients (c_1, c_2), complemented by a velocity clamp factor. The role of velocity clamping has been widely studied in the PSO literature [43, 44] and more recently extended through adaptive strategies such as PSO-SAVL [45].

To ensure efficient convergence, termination was enforced using a MaxDist criterion for DE and a MaxDistQuick criterion for PSO. For differential evolution, the MaxDist criterion terminates the run once the maximum distance between all individuals and the best vector falls below a small threshold, indicating complete population convergence.

For Particle Swarm Optimization, the MaxDistQuick criterion applies the same principle but only to the best p % of particles, efficiently detecting convergence when most of the swarm has stabilized while a few remain unsettled [46].

Both algorithms were executed under comparable computational conditions to ensure reproducibility and fairness in subsequent comparisons. The parameter ranges explored for DE and PSO during the sequential model-based optimization process are summarized in Table 1. These intervals were selected based on established practices in the literature and preliminary sensitivity analyses, ensuring that the search space captured both commonly used and empirically effective configurations.

Table 1. Parameter settings and ranges for DE and PSO algorithms

Algorithm	Parameter	Value
DE	Population size N_p	$40 \leq N_p \leq 100$
	Mutation factor F	$0.30 \leq F \leq 0.95$
	Crossover rate C_r	$0.40 \leq C_r \leq 0.99$
	Stopping criterion	MaxDist

PSO	Number of particles	$50 \leq N \leq 100$
	Inertia weight (w)	$0.70 \leq w \leq 0.90$
	Cognitive coefficient c_1	$0.90 \leq c_1 \leq 1.81$
	Social coefficient c_2	$1.40 \leq c_2 \leq 2.10$
	Velocity clamp factor v_{clamp}	$0.18 \leq v_{clamp} \leq 0.30$
	Stopping criterion	MaxDistQuick
Both	Evaluation budget	40,000 function evaluations

3.3. Sequential Model-Based Optimization

A sequential model-based optimization (SMBO) framework was employed [6, 14], in which Gaussian process surrogates with Matern and radial basis function (RBF) kernels modelled algorithm performance, and the acquisition function sequentially proposed new control-parameter configurations under a fixed evaluation budget. Both Matern (ν = 2.5) and RBF (squared exponential) kernels were considered [47]. At each stage, the surrogate adaptively selected the kernel that best fits the observed data, with a constant kernel serving as a scaling factor and a white-noise kernel accounting for stochastic perturbations. The acquisition function *gp_hedge* dynamically balanced exploration and exploitation by combining expected improvement (EI), probability of improvement (PI), and upper confidence bound (UCB). Initialization relied on a Sobol quasi-random design to ensure broad coverage of the parameter space. Within each Bayesian iteration, the Gaussian-process surrogate was updated sequentially using newly evaluated configurations until no improvement exceeding a tolerance of 10^{-6} was observed for 40 consecutive iterations, triggering early termination.

This perspective aligns with the fixed-budget view in algorithm configuration [6], as well as [33], who explicitly evaluate their SMBO-based MetaTuner framework under a fixed evaluation budget.

3.4 Statistical Evaluation

Both DE and PSO, in their default configurations, were executed under a fixed evaluation budget of 40,000 function evaluations to ensure that all comparisons across algorithms and tuning strategies were conducted under identical computational resources. In the case of SMBO, early stopping criteria sometimes resulted in runs completing slightly below the maximum budget, since further evaluations were unlikely to improve performance. This design prevents redundant fitness calls while preserving fairness across experimental conditions.

For each benchmark problem, 20 independent runs were performed using both default and SMBO-tuned configurations. Performance was summarized by the mean and standard deviation of the best-found fitness values Table 4.

To assess whether SMBO-tuned configurations achieved statistically significant improvements over the default settings, the Wilcoxon signed-rank test [48, 49] was applied to paired best-fitness values obtained from independent runs. The null hypothesis (H_0) stated that no significant difference exists between the compared algorithms, while the alternative hypothesis (H_a) assumed a significant performance difference. The resulting p-values quantify the evidence against H_0 −smaller values indicate stronger evidence that the null may not hold. Although a 5% significance level ($p < 0.05$) is conventionally adopted, the results are interpreted in terms of the p-values themselves, providing direct statistical evidence. The Wilcoxon test is widely recommended for evaluating stochastic optimization algorithms [50, 51].

4. Results and Discussion

The performance of DE and PSO under default configurations across the three constrained engineering design problems is presented in Table 2. The evaluation budget was progressively increased in three stages, from 30,000 to 50,000 evaluations. Convergence was not yet achieved in the initial stage, whereas the latter two yielded identical objective values, confirming that convergence was reached at 40,000 evaluations.

For WBD, DE achieved a better objective value (1.724854 vs 1.724927) compared to PSO, both under the fixed evaluation budget. On PVD, DE slightly outperformed PSO (5870.124 vs. 5887.886), while on CSD, DE achieved a slightly lower objective value (0.012665) compared to PSO (0.012666), indicating that both algorithms converged to essentially identical solutions. These results suggest that DE typically produces more stable solutions than PSO in WBD and PVD, whereas both methods exhibit equivalent performance in CSD.

Table 2. Best results obtained with default DE and PSO

Problem	Algorithm	Best value	Parameters
WBD	DE	1.724854	$N_p = 100, gen = 399,$ $F = 0.8, C_r = 0.9$
	PSO	1.724927	$N = 100, iter = 399$ $w = 0.7298,$ $c_1 = c_2 = 1.49618$ $v_{clamp} = 0.2$
PVD	DE	5870.124	$N_p = 100, gen = 399,$ $F = 0.8, C_r = 0.9$
	PSO	5887.886	$N = 100, iter = 399,$ $w = 0.7298,$ $c_1 = c_2 = 1.49618$ $v_{clamp} = 0.2$
CSD	DE	0.012665	$N_p = 100, gen = 399,$ $F = 0.8, C_r = 0.9$
	PSO	0.012666	$N = 100, iter = 399$ $w = 0.7298$ $c_1 = c_2 = 1.49618,$ $v_{clamp} = 0.2$

These results show that even without tuning, both DE and PSO are capable of achieving competitive solutions under equal computational budgets. Similar findings have been reported in the literature, where a variety of metaheuristic algorithms, including Levy-flight, hybrid CPSOGSA, PSO variants, and other bio-inspired methods, have been successfully applied to classical benchmark design problems [52, 53, 54, 55, 56, 57].

On the WBD problem, both SMBO-DE and SMBO-PSO, using the Matern kernel, reached the best-known objective value with slightly fewer effective evaluations than the full computational budget Table 3.

Table 3. SMBO-tuned metaheuristic configurations WBD, PVD, CSD problems

Algorithm	WBD	PVD	CSD
SMBO-DE	Matern kernel	RBF kernel	Matern kernel
Best value	1.724853	5870.124	0.012665
Best parameters	$N_p = 51,$ $gen = 783$ $F = 0.6450$ $C_r = 0.5253$	$N_p = 67$ $gen = 596$ $F = 0.6360$ $C_r = 0.5416$	$N_p = 69$ $gen = 578$ $F = 0.8462$ $C_r = 0.9824$
Effective evaluations	39,984	39,999	39,951
SMBO-PSO	Matern kernel	RBF kernel	RBF kernel
Best value	1.724852	5885.343	0.012666
Best parameters	$N = 50,$ $iters = 799$ $w = 0.7406$ $c_1 = 0.9039$ $c_2 = 1.8394$ $v_{clamp} = 0.2986$	$N = 50$ $iter = 799$ $w = 0.7571$ $c_1 = 1.1405$ $c_2 = 1.6460$ $v_{clamp} = 0.2251$	$N = 87$ $iter = 458$ $w = 0.7302$ $c_1 = 1.5624$ $c_2 = 1.8244$ $v_{clamp} = 0.1861$

Effective evaluations	40,000	40,000	39,933

For the PVD, SMBO-DE (RBF kernel) achieved a marginal improvement (**5870.124**) over the default DE, while SMBO-PSO (RBF kernel) also improved relative to the PSO default (**5885.343**). For the CSD problem, SMBO-DE (Matern kernel) achieved a best objective value of **0.012665**, identical to the value obtained by the default DE. Although the improvement achieved by SMBO-PSO in the CSD problem appears numerically insignificant, the tuned configuration demonstrated greater computational efficiency and stability. Overall, the SMBO framework consistently identified competitive or improved solutions through optimized parameter settings, such as reduced population or particle sizes and adapted control coefficients

The effective evaluation counts for SMBO-based runs are occasionally slightly below the fixed budget Table 3. This occurs because the SMBO framework applies early stopping once convergence is reached, thereby avoiding redundant evaluations. Importantly, this does not affect fairness, since all algorithms operated under the same nominal budget of 40,000 function evaluations.

To evaluate the effectiveness and robustness of the SMBO-tuned models, a statistical analysis was conducted over 20 independent runs for each algorithm–problem combination. On the WBD problem, SMBO-DE achieved a slightly better mean and reduced variance compared to the default DE, while SMBO-PSO produced a substantial reduction in variance, indicating more stable convergence behavior across runs, as shown in Table 4.

Table 4. Statistical results across 20 independent runs

Problem	Algorithm	Mean	Std
WBD	DE	1.724854	8.23E−07
	SMBO-DE	1.724852	5.63E−07
	PSO	1.725040	2.36E−04
	SMBO-PSO	1.724852	1.10E−06
PVD	DE	5870.124	1.58E−05
	SMBO-DE	5870.124	4.00E−06
	PSO	5885.826	5.81E−01
	SMBO-PSO	5885.334	6.11E−03
CSD	DE	0.0126652	8.88E−11
	SMBO-DE	0.0126652	1.16E−15
	PSO	0.0126731	1.58E−05
	SMBO-PSO	0.0126718	1.36E−05

For the PVD problem, both SMBO-DE and SMBO-PSO outperformed their default counterparts, achieving lower mean objective values and markedly reduced variance, with SMBO-PSO exhibiting a particularly strong improvement in stability. In the CSD problem, the SMBO-DE results remained marginally better than those of the default DE, while SMBO-PSO again exhibited lower variability compared to its untuned counterpart Table 4. Overall, the SMBO framework consistently maintained or improved the mean objective values and substantially reduced variance, particularly for PSO, confirming its effectiveness in enhancing robustness, stability, and reproducibility across optimization runs.

The results of the Wilcoxon signed rank test were obtained from 20 independent runs conducted under a fixed evaluation budget Table 5. This non-parametric test was used to assess whether the performance differences between the default and SMBO-tuned algorithms were statistically significant.

Table 5. Wilcoxon signed-rank test p -values 20 runs, fixed budget

Problem	DE vs. SMBO-DE (p -value)	PSO vs. SMBO-PSO (p -value)
WBD	1.68E−04	1.91E−06
PVD	4.77E−05	3.81E−06
CSD	1.91E−06	5.96E−01

For the WBD problem, both DE and PSO exhibited highly significant improvements when tuned with SMBO ($p < 0.001$). In the PVD problem, the results were even more decisive, with both SMBO-DE and

SMBO-PSO clearly outperforming their default counterparts ($p < 0.00001$). For the CSD problem, only SMBO-DE showed a statistically significant improvement ($p < 0.01$), while the difference between PSO and SMBO-PSO was not significant ($p = 0.596$), reflecting their nearly identical performance Table 5.

Although SMBO-DE and DE achieved nearly identical mean and variance values in the CSD problem, the Wilcoxon signed-rank test indicated statistical significance ($p < 0.01$). This outcome can be attributed to the test's sensitivity to consistent directional improvements across runs, even when numerical differences are extremely small. The result suggests that SMBO-DE produced slightly more reliable convergence behavior over multiple trials. Overall, the Wilcoxon analysis confirms that SMBO provides statistically significant improvements in most cases, particularly for WBD and PVD, whereas CSD remains a more challenging problem in which gains are less pronounced.

5. Conclusion

This study implemented an automated Bayesian optimization framework, employing sequential model-based optimization (SMBO) with Gaussian process surrogates, to systematically tune the control parameters of differential evolution (DE) and particle swarm optimization (PSO) in three classical constrained engineering design benchmarks, namely the welded beam, the pressure vessel, and the tension or compression spring design problems, under a fixed computational budget. The metaheuristic models were maintained in their standard form without introducing algorithmic modifications, while SMBO was used specifically to optimize their parameter settings automatically.

The results demonstrated that SMBO consistently improved or maintained the quality of solutions while reducing variance across multiple independent runs. These effects were most pronounced for PSO, where the variance decreased by one to two orders of magnitude, indicating a substantial gain in convergence stability and robustness. For DE, SMBO achieved smaller but consistent improvements, and the Wilcoxon signed rank test confirmed statistically significant differences in most cases. In the more challenging compression spring problem, the SMBO tuned DE produced results nearly identical to those of the default DE but showed greater consistency across runs. In contrast, the SMBO tuned PSO maintained comparable performance with no statistically significant difference.

Overall, the findings support the idea that SMBO provides a reliable and automated strategy for parameter tuning, grounded in statistical modelling rather than manual experimentation. By integrating Bayesian Optimization into the tuning process, this approach enhances both solution reliability and reproducibility in the application of DE and PSO to constrained engineering design.

Although the three benchmark problems examined, namely the welded beam, pressure vessel, and compression spring, represent simplified design formulations, they capture essential characteristics of real-world engineering tasks. The observed improvements obtained through SMBO tuned configurations of DE and PSO demonstrate the potential for broader adoption of such meta optimization frameworks in practical mechanical and industrial design problems, where systematic parameter tuning is essential for achieving reliable and cost-effective solutions.

6. Future Work

Future research will focus on applying the SMBO-DE and SMBO-PSO frameworks to real world problems that are nonlinear and constrained. Such problems often arise in practical domains such as energy, transportation, and manufacturing, where objectives are complex and multiple limitations must be satisfied. Extending the proposed approach to these realistic applications will demonstrate its capability to enhance solution quality, stability, and robustness in real decision-making environments.

Reference

1. Arora, J. S. (2011). *Introduction to Optimum Design*. Academic Press.
2. Rao, S. S. (2019). *Engineering Optimization: Theory and Practice*. John Wiley & Sons.
3. Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, *11*(4), 341-359.
4. Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, vol. 4, pp. 1942-1948. IEEE.
5. Parouha, R. P., & Verma, P. (2022). A systematic overview of developments in differential evolution and particle swarm optimization with their advanced suggestion. *Applied Intelligence*, *52*(12), 10448–10492. https://doi.org/10.1007/s10489-021-02557-4
6. Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2010). Sequential model-based optimization for general algorithm configuration (extended version). *Technical Report TR-2010–10, University of British Columbia, Computer Science, Tech. Rep.*
7. Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.
8. Mockus, J., Tiesis, V., & Žilinskas, A. (1978). The application of Bayesian methods for seeking the extremum. In L. C. W. Dixon & G. P. Szegö (Eds.), *Towards Global Optimization 2* (pp. 117–129). North-Holland.
9. Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-

box functions. *Journal of Global Optimization, 13*(4), 455–492. https://doi.org/10.1023/A:1008306431147
10. Bartz-Beielstein, T., Lasarczyk, C., & Preuss, M. (2005). Sequential parameter optimization. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC 2005)* (pp. 773–780). IEEE. https://doi.org/10.1109/CEC.2005.1554721
11. Hutter, F., Hoos, H. H., Leyton-Brown, K., & Murphy, K. P. (2009). An experimental investigation of model-based parameter optimization: SPO and beyond. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO '09)* (pp. 271–278). ACM. https://doi.org/10.1145/1569901.1569933
12. Bartz-Beielstein, T. (2010). Sequential parameter optimization—an annotated bibliography. *Update*.
13. Boçi, B., & Simoni, A. (2025). Refining wage predictions with machine learning and Bayesian optimization. *Mathematics and Statistics, 13*(5), 279–296. https://doi.org/10.13189/ms.2025.130503
14. Bergstra, J., Komer, B., Eliasmith, C., & Warde-Farley, D. (2014). Preliminary evaluation of Hyperopt algorithms on HPOLib. *Proceedings of the ICML 2014 AutoML Workshop*.
15. Eiben, A. E., & Smit, S. K. (2011). *Parameter tuning for configuring and analyzing evolutionary algorithms*. Swarm and Evolutionary Computation, 1(1), 19–31.
16. Brest, J., Greiner, S., Bošković, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation, 10*(6), 646–657. https://doi.org/10.1109/TEVC.2006.87213
17. Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation, 13*(2), 398–417. https://doi.org/10.1109/TEVC.2008.927706
18. Zamuda, A., & Brest, J. (2015). Self-adaptive control parameters' randomization frequency and propagations in differential evolution. *Swarm and Evolutionary Computation, 25*, 72–99. https://doi.org/10.1016/j.swevo.2015.01.004
19. Das, S., Mullick, S. S., & Suganthan, P. N. (2016). Recent advances in differential evolution: An updated survey. *Swarm and Evolutionary Computation, 27*, 1–30. https://doi.org/10.1016/j.swevo.2016.01.004
20. Ahmad, M. F., Isa, N. A. M., Lim, W. H., & Ang, K. M. (2022). Differential evolution: A recent review based on state-of-the-art works. *Alexandria Engineering Journal, 61*(10), 3831–3872. https://doi.org/10.1016/j.aej.2021.09.013
21. Gianni, A. M., Tsoulos, I. G., Charilogis, V., & Kyrou, G. (2025). Enhancing Differential Evolution: A Dual Mutation Strategy with Majority Dimension Voting and New Stopping Criteria. *Symmetry, 17*(6), 844. https://doi.org/10.3390/sym17060844
22. Shi, Y., & Eberhart, R. C. (1998). A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings* (pp. 69–73). IEEE. https://doi.org/10.1109/ICEC.1998.69914
23. Clerc, M., & Kennedy, J. (2002). The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation, 6*(1), 58–73. https://doi.org/10.1109/4235.985692
24. Ganesan, T., Vasant, P., & Elamvazuthy, I. (2012). A hybrid PSO approach for solving non-convex optimization problems. *Archives of Control Sciences, 22*(1), 87-105.
25. Yun, Y., Gen, M., & Erdene, T. N. (2023). Applying GA-PSO-TLBO approach to engineering optimization problems. *Math. Biosci. Eng, 20*(1), 552-571. doi: 10.3934/mbe.2023025
26. Bonyadi, M. R., & Michalewicz, Z. (2017). Particle swarm optimization for single objective continuous space problems: a review. *Evolutionary computation*, *25*(1), 1-54.
27. Abualigah, L. (2025). Particle swarm optimization: Advances, applications, and experimental insights. *Computers, Materials & Continua, 82*(2), 1539–1556. https://doi.org/10.32604/cmc.2025.060765
28. Piotrowski, A. P., Napiorkowski, J. J., & Piotrowska, A. E. (2023). Particle swarm optimization or differential evolution—A comparison. *Engineering Applications of Artificial Intelligence, 121*, 106008. https://doi.org/10.1016/j.engappai.2023.106008
29. Zhang, Y., Li, H., Bao, E., Zhang, L., & Yu, A. (2019). A hybrid global optimization algorithm based on particle swarm optimization and Gaussian process. *International Journal of Computational Intelligence Systems, 12*(2), 1270–1281. https://doi.org/10.2991/ijcis.d.191101.004
30. Klus, J., Grunt, P., & Dobrovolný, M. (2021). Hyper-optimization with Gaussian process and differential evolution algorithm. *Neural Information Processing Systems (NeurIPS) BlackBox Optimization Challenge 2020*. arXiv:2101.10625. https://arxiv.org/abs/2101.10625
31. Vincent, A. M., & Jidesh, P. (2023). An improved hyperparameter optimization framework for AutoML systems using evolutionary algorithms. *Sci Rep 13*, 4737.
32. Roman, I., Ceberio, J., Mendiburu, A., & Lozano, J. A. (2016, July). Bayesian optimization for parameter tuning in evolutionary algorithms. In *2016 IEEE Congress on Evolutionary Computation (CEC)* pp. 4839-4845. IEEE.
33. Trindade, Á. R., & Campelo, F. (2019). Tuning metaheuristics by sequential optimisation of regression models. *Applied Soft Computing, 85*, 105829. https://doi.org/10.1016/j.asoc.2019.105829
34. Atkinson, J., Müller-Bady, R., & Kappes, M. (2020, July). Hybrid bayesian evolutionary optimization for hyperparameter tuning. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion* (pp. 225-226).
35. Ruther, C., & Rieck, J. (2021). A Bayesian optimization approach for tuning a genetic algorithm solving practical-oriented pickup and delivery problems. *IEEE Transactions on Automation Science and Engineering*, 1-12.
36. Rüther, C., & Rieck, J. (2024). A Bayesian optimization approach for tuning a grouping genetic algorithm for solving practically oriented pickup and delivery problems. *Logistics, 8*(1), 14.
37. Soares, R. C., Silva, J. C., Lucena Junior, J. A. de, Lima Filho, A. C., Ramos, J. G. G. de S., & Brito,

A. V. (2025). Integration of Bayesian optimization into hyperparameter tuning of the particle swarm optimization algorithm to enhance neural networks in bearing failure classification. *Engineering Applications of Artificial Intelligence*, *135*, 107252. https://doi.org/10.1016/j.measurement.2024.115829

38. Coello, C. A. C. (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, *41*(2), 113-127.
39. Coello, C. A. C. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering*, *191*(11-12), 1245-1287. https://doi.org/10.1016/S0045-7825(01)00323-1
40. Homaifar, A., Qi, C. X., & Lai, S. H. (1994). Constrained optimization via genetic algorithms. *Simulation*, *62*(4), 242-253.
41. Mezura-Montes, E., & Coello Coello, C. A. (2011). Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation*, *1*(4), 173–194. https://doi.org/10.1016/j.swevo.2011.10.001
42. Li, X. (2010). Niching without niching parameters: Particle swarm optimization using a ring topology. *IEEE Transactions on Evolutionary Computation*, *14*(1), 150–169. https://doi.org/10.1109/TEVC.2009.2017514
43. Pluháček, M., Šenkeřík, R., Viktorin, A., & Kadavý, T. (2018). Study on velocity clamping in PSO using CEC13 benchmark. *Proceedings-European Council for Modelling and Simulation, ECMS*.
44. Lim, S. P., Hoon, H., & Song, W. H. (2020, May). A comparative study on different parameter factors and velocity clamping for particle swarm optimisation. In *IOP Conference Series: Materials Science and Engineering* (Vol. 864, No. 1, p. 012068). IOP Publishing.
45. Li, X., Mao, K., Lin, F., & Zhang, X. (2021). Particle swarm optimization with state-based adaptive velocity limit strategy. *Neurocomputing*, *447*, 64-79.
46. Zielinski, K., & Laur, R. (2008). Stopping criteria for differential evolution in constrained single-objective optimization. In *Advances in differential evolution* (pp. 111-138). Berlin, Heidelberg: Springer Berlin Heidelberg.
47. Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, *25*.
48. Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, *1*(6), 80–83.
49. Hollander, M., Wolfe, D. A., & Chicken, E. (2013). *Nonparametric statistical methods*. John Wiley & Sons.
50. Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*(1), 3-18.
51. Li, J., & Sun, K. (2023). Pressure vessel design problem using improved Gray Wolf optimizer based on Cauchy distribution. *Applied Sciences*, *13*(22), 12290.
52. Houssein, E. H., Saad, M. R., Hashim, F. A., Shaban, H., & Hassaballah, M. (2020). Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, *94*, 103731.
53. Rather, S. A., & Bala, P. S. (2021). Application of constriction coefficient-based particle swarm optimisation and gravitational search algorithm for solving practical engineering design problems. *International Journal of Bio-Inspired Computation*, *17*(4), 246-259.
54. El-Shorbagy, M. A., & El-Refaey, A. M. (2022). A hybrid genetic–firefly algorithm for engineering design problems. *Journal of Computational Design and Engineering*, *9*(2), 706-730.
55. Shami, T. M., Mirjalili, S., Al-Eryani, Y., Daoudi, K., Izadi, S., & Abualigah, L. (2023). Velocity pausing particle swarm optimization: a novel variant for global optimization. *Neural Computing and Applications*, *35*(12), 9193-9223.
56. Qiao, J., Wang, G., Yang, Z., Luo, X., Chen, J., Li, K., & Liu, P. (2024). A hybrid particle swarm optimization algorithm for solving engineering problem. *Scientific Reports*, *14*(1), 8357.
57. Xia, H., Ke, Y., Liao, R., & Zhang, H. (2025). Fractional order dung beetle optimizer with reduction factor for global optimization and industrial engineering optimization problems. *Artificial Intelligence Review*, *58*(10), 308.