

Article

Multi-Objective Robotics Optimization Using Improved MO-BxR Algorithms

Ravipudi Venkata Rao ^{1,*} , Harishankar Morazha Variam ¹ and Joao Paulo Davim ² ¹ Department of Mechanical Engineering, Sardar Vallabhbhai National Institute of Technology, Surat 395007, India; p24me004@med.svnit.ac.in² Department of Mechanical Engineering, University of Aveiro, Campus Santiago, 3810-193 Aveiro, Portugal; pdavim@ua.pt

* Correspondence: rvr@med.svnit.ac.in

Abstract

Robotics optimization is essential for improving the performance, efficiency, and reliability of robotic systems, especially when dealing with complex engineering problems involving multiple conflicting objectives. Algorithm-specific parameter-free metaheuristic algorithms have gained attention in such applications because they eliminate the need for problem-specific parameter tuning. However, their performance can be further enhanced by improving convergence and maintaining solution diversity in multi-objective optimization. This paper proposes three multi-objective variants—archive, opposition, and self-adaptive multi-population (SAMP)—for the algorithm-specific parameter-free BxR algorithms such as Best–Mean–Random (BMR), Best–Worst–Random (BWR), and Best–Mean–Worst–Random (BMWR). The proposed variants are evaluated on five robotic optimization problems spanning two to six objectives, including Autonomous Underwater Vehicle shape optimization, power line inspection robot design, inverse kinematics of a 4-DOF manipulator, wall-building robot trajectory planning, and optimization of a reconfigurable parallel cutting and grinding mechanism. Their performance is compared with several established multi-objective algorithms using metrics such as GD, IGD, SPC, and HV, supported by rigorous statistical testing involving Friedman tests, Conover post hoc analysis with Holm correction, and Vargha–Delaney A_{12} effect sizes over 30 independent runs. The results show that archive variants achieve the best IGD rank in four of the five case studies and the best HV rank in three of them, with the five-objective trajectory planning problem being the sole exception where SAMP and base BxR variants show improved IGD performance. The base BxR algorithms prove to be strong competitors, consistently outperforming established parameter-dependent methods on IGD across all five problems. The opposition variants do not provide consistent improvement; however, they also do not cause catastrophic degradation, suggesting that refined opposition strategies warrant further investigation. The study demonstrates the effectiveness of the proposed algorithms as practical optimization tools for complex robotic optimization problems.



Academic Editor: Paolo Renna

Received: 24 April 2026

Revised: 17 May 2026

Accepted: 18 May 2026

Published: 21 May 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: robotics optimization; multi-objective optimization; MO-BxR algorithms; performance metrics

1. Introduction

With rapid technological advancements, robotics is finding applications in various fields of engineering, such as construction, inspection, and manufacturing mechanisms.

However, the optimization of such systems is often difficult due to conflicting objectives, constraints, and a large number of design variables. Inverse kinematics of mechanisms is another challenging problem where developing analytical solutions is difficult due to higher motion redundancy. Optimization techniques can be employed to address these challenges effectively. In robotic systems, optimization plays a critical role in addressing challenges such as task allocation among multiple robots, energy-efficient trajectory planning, path planning in obstacle-rich environments, and optimal control of complex mechanical systems. These problems are frequently multi-objective in nature, requiring simultaneous optimization of competing criteria such as minimizing energy consumption while maximizing task completion efficiency, or reducing travel distance while maintaining workload balance. The search spaces associated with such problems are often large, non-convex, and riddled with local optima, making traditional exact methods computationally prohibitive for real-world problem scales. Consequently, metaheuristic optimization approaches have gained significant attention in the robotics community due to their ability to find near-optimal solutions within reasonable computational time without requiring gradient information or strict mathematical formulations of the problem.

Metaheuristic optimization techniques can be employed to effectively explore the search space and identify near-optimal solutions. Some of the popular metaheuristic algorithms include the Genetic Algorithm (GA), Differential Evolution (DE), Covariance Matrix Adaptation Evolution Strategies (CMA-ES), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), and Bacterial Foraging Optimization (BFO). Although these are popular, each algorithm has its own control parameters that must be tuned to achieve good performance for each problem. This imposes a significant tuning overhead, as practitioners must calibrate these parameters for each specific application prior to deployment.

Algorithm-specific parameter-free metaheuristic algorithms such as Rao and Jaya can address this issue. Best-Mean-Random (BMR), Best-Worst-Random (BWR), and Best-Mean-Worst-Random (BMWR), collectively known as BxR algorithms, are some of the latest in this category, introduced by Rao and Davim [1]. While the base equations of these algorithms are algorithm-specific and parameter-free, the overall sequence of operations within the algorithm can be modified or combined with other operations, possibly boosting their efficiency. Rao and Pawar [2] demonstrated the effectiveness of such modified versions of Rao algorithms in their studies. Since BMR, BWR, and BMWR are relatively recent algorithms, their improved multi-objective variants have not yet been systematically developed or evaluated. Therefore, this study introduces three multi-objective variants; archive-based, opposition-based, and self-adaptive multi-population (SAMP)-based. These algorithms are then tested through case studies in the field of robotics to gauge their performance in real engineering applications.

Another problem related to design optimization is the integration of complex calculations/simulations that are slow to run with these optimization algorithms. This is particularly true for kinematics and dynamics calculations related to robots with a higher number of degrees of freedom (DOF), as they often include recursive procedures and solving of higher order matrix systems. Mathematical models that approximate the slow simulations can be used to accelerate the process. Such models are classified as surrogate models. Since the objective of this study is to evaluate recent optimization algorithms, case studies involving surrogate models were intentionally selected.

The true effectiveness of an algorithm can only be assessed when it is applied to a variety of problems with diverse characteristics and complex objectives. The field of robotics is well suited for such evaluations because it encompasses a wide variety of optimization problems.

Tang et al. [3] optimized the shape design of an Autonomous Underwater Vehicle (AUV) using MOPSO-CD, with surrounded volume and resistance force as objectives. Surrogate models were developed using RSM and GEP, with the GEP model proving more accurate. Nabavi et al. [4] optimized a 6-PUS parallel robot configuration using GA, with dexterity, kinetic energy, and workspace index as objectives, identifying a unique architecture that significantly reduces actuator forces.

Rokbani et al. [5] applied a modified PSO algorithm to solve the inverse kinematic problem of a 5-DOF robot, formulating a two-objective problem based on position and orientation errors. Zhao et al. [6] optimized the Magnetic Adsorption Unit (MAU) of a wall-climbing robot using RSM-based objective functions and NSGA-2, achieving 6.49% reduction in vibration amplitude and 9.66% reduction in mass.

Shi et al. [7] developed a many-objective trajectory optimization method for wall-building robots using RSM and NSGA-3, addressing the viscoelastic effect of cement mortar and reducing masonry error from 2.62 mm to 0.11 mm. Liu et al. [8] solved the inverse kinematic problem using a single-objective fitness function targeting positional errors and energy consumption, demonstrating effectiveness through 4, 6, and 7 DOF robot cases.

Yoon et al. [9] optimized quad-copter blade shapes for maximum hovering thrust and minimum power consumption using CFD-based surrogate models and NSGA-2, fabricating and testing seven Pareto-optimal blade designs. Jin et al. [10] optimized a walking device for hybrid power line inspection robots under strong wind conditions using NSPSO, reporting deflection angle reductions of 33.19%, 39.21%, and 13.23% around X, Y, and Z axes respectively.

Duan et al. [11] proposed an improved NSGA-2 (INSGA-2) for multi-objective path planning optimizing safety, length, and smoothness, demonstrating superiority over six algorithms across 16 benchmark instances. Yue et al. [12] developed an improved PSO using chaos theory to optimize controller parameters for trajectory tracking of robotic manipulators, achieving tracking errors within 2%.

Yang et al. [13] integrated simulations, a Chebyshev surrogate model, and dung beetle optimizer for structural optimization of a wall-climbing robot, reducing adhesive module weight by 11.7%. Zeid et al. [14] formulated the design of an In-Pipe Inspection Robot (IPIR) as a multi-objective problem and solved it using NSGA-2, achieving 11 kg payload capacity and 33 mm radial range.

Song et al. [15] optimized obstacle crossing height and driving torque of a tracked robot using NSGA-2 with adaptive genetic strategy (NSGA-2-AGS), which also demonstrated competitive performance on DTLZ and WFG benchmarks. Cui et al. [16] presented a multi-strategy adaptable ant colony optimization (MsAACO) for path planning, outperforming A*, multiple ACO variants, and other established methods across five benchmark environments.

Duan et al. [17] proposed a multi-objective optimal control method for a 6-DOF robotic crusher using gray wolf optimization, achieving energy consumption reductions of up to 36.51% and wear reductions of up to 43.39%. Wang et al. [18] optimized energy consumption and execution time of a flying–walking power line inspection robot using an improved NSGA-II with Cauchy variation and simulated annealing, achieving approximately 5% energy savings.

Wang et al. [19] developed a reconfigurable parallel mechanism for grinding–cutting processes, using RSM surrogate models and MO-PSO to achieve 70.66% workspace expansion and 19.89% load capacity enhancement. Gupta et al. [20] compared NSGA-2, CMOPSO, DCNSGA-3, and CMOEA/D for gait cycle optimization of a 25-DOF humanoid robot, with NSGA-2 emerging as the best option overall. Wang et al. [21] optimized a large-amplitude ultrasonic osteotome for surgical robots using SPEA-2 and NSGA-2, targeting resonance frequency, displacement amplification, and tool tip vibration.

Zhang et al. [22] introduced Collaborative Swarm-Differential Evolution (CSDE) for multi-robot task assignment, outperforming MOPSO, MODE, and MOGA in IGD and HV across benchmark scenarios. You et al. [23] proposed the Fused Multi-Strategy White Shark Optimization (FMWSO) for mobile robot path planning, outperforming WSO, D*, A*, and GWO in path length, smoothness, and computational efficiency. Xing et al. [24] highlighted recent developments in dual-task coordination, compliant interaction control, and interaction-aware motion planning for mobile manipulators operating in constrained environments.

In summary, the literature review confirms the relevance of advanced optimization techniques in the field of robotics. NSGA-2 and its variants [6,9,10,15,20,21], and PSO variants [3,10,12,19] emerged as the most widely used optimization techniques for robotic application. However, several limitations can be identified from the existing literature. First, the majority of studies rely on well-established algorithms such as NSGA2 and PSO variants, which require careful parameter tuning for each specific problem, adding computational overhead and limiting their generalizability across diverse applications. Second, most works evaluate their proposed methods on a single problem type or a narrow class of robotics applications, making it difficult to assess algorithm robustness across varied problem characteristics such as differing numbers of objectives, design variables, and constraint structures. Third, although algorithm-specific parameter-free algorithms like Rao, Jaya, BMR, BWR, and BMWR have shown promise in single-objective settings, their multi-objective extensions remain largely unexplored, particularly in the context of real-world robotics problems. Finally, few studies systematically compare algorithm performance using a comprehensive set of multi-objective metrics (e.g., GD, IGD, spacing, and HV), often relying on only one or two indicators. These gaps highlight the need for developing and rigorously benchmarking new reduced-parameter multi-objective algorithm variants across a diverse set of robotics optimization problems, which forms the scope of the present work.

The objectives of this work are as follows:

- Develop improved multi-objective variants of BxR algorithms (MO-BxR).
- Identify robotic optimization problems, focusing on diversity in problem types.
- Optimize the selected problems using the improved algorithms.
- Conduct a detailed analysis of the performance of these algorithms with the existing algorithms.

2. Proposed Multi-Objective BxR Variants and Evaluation Framework

MO-BxR algorithms developed by Rao and Davim [1] are briefly described below, followed by the improved variants.

2.1. BxR and MO-BxR Algorithms and Their Variants

Like all population-based algorithms, BxR algorithms create the next generation using the current population and compare the two to decide which ones are carried forward to the next iteration. The update rules for BMR, BWR, and BMWR are given in Equations (1)–(3).

Each algorithm employs the same random mutation formula given in Equation (4) to prevent premature convergence to local optima.

$$X_{new} = X_{old} + r_1 \times (X_{best} - T \times X_{mean}) + r_2 \times (X_{best} - X_{random}) \quad (1)$$

$$X_{new} = X_{old} + r_1 \times (X_{best} - T \times X_{random}) - r_2 \times (X_{worst} - X_{random}) \quad (2)$$

$$X_{new} = X_{old} + r_1 \times (X_{best} - T \times X_{mean}) - r_2 \times (X_{worst} - X_{random}) \quad (3)$$

$$\begin{aligned} &\text{If } r_4 > 0.5 \\ &X_{new} = X_U - (X_U - X_L) \times r_3 \end{aligned} \quad (4)$$

where X_{new} is the new solution vector, X_{old} is the current (old) solution vector, X_{best} is the best solution vector, X_{worst} is the worst solution vector, X_{mean} is the mean solution vector of the population, X_{random} is the randomly selected solution vector from the population, r_1 , r_2 , r_3 , and r_4 are the uniformly distributed random numbers in the range $[0, 1]$, T is a discrete variable whose value can be randomly selected as one or two during runtime. X represents the solution vector, while X_U and X_L represent the upper and lower bounds of the design variables, respectively.

For single-objective problems, the best and worst solutions are selected based on the fitness value from the population, depending on the optimization goal (maximization or minimization). In case of multi-objective optimization, non-dominated sorting with crowding-distance is used to select the Pareto set and to decide the solutions that are carried over to the next generation. In this scenario, there is no single best or worst solution, hence a pool of best and worst solutions is selected based on dominance ranking and crowding-distance. At early stages of optimization, there will be a non-dominated set and several dominated sets. Here, the solutions from the non-dominated set are used as best, and the solutions from the most dominated set are selected as worst. At later stages, the population may consist entirely of non-dominated solutions; the selection mechanism then shifts to use solutions with high crowding-distance values as best and low values as worst.

Another important aspect of optimization is constraint handling. When there are no feasible solutions, the solutions with the fewest constraint violations are selected as best and vice versa. This ensures that the algorithm always moves towards a feasible region before analyzing dominance and crowding-distances. During this, the best and worst are selected based on the number of constraint violations.

Now, three variants of the MO-BxR algorithms are proposed and the details are described below, starting with the archive-based variant.

2.1.1. Archive-Based Variants

This variant maintains an external archive of non-dominated solutions. Let A denote the external archive containing non-dominated solutions. Its size is taken as twice the population size and it helps store a larger number of Pareto-optimal solutions that may otherwise be discarded due to population size limits. The external archive is also sorted after each iteration by combining with the non-dominated set from the new population. This additional non-dominated sorting step further refines the solution set and retains only the best solutions in the archive. A flowchart for the external archive is shown in Figure 1.

The size of the archive is an algorithmic choice rather than a parameter to be tuned, and this decision works well as can be seen from the results presented in the following sections.

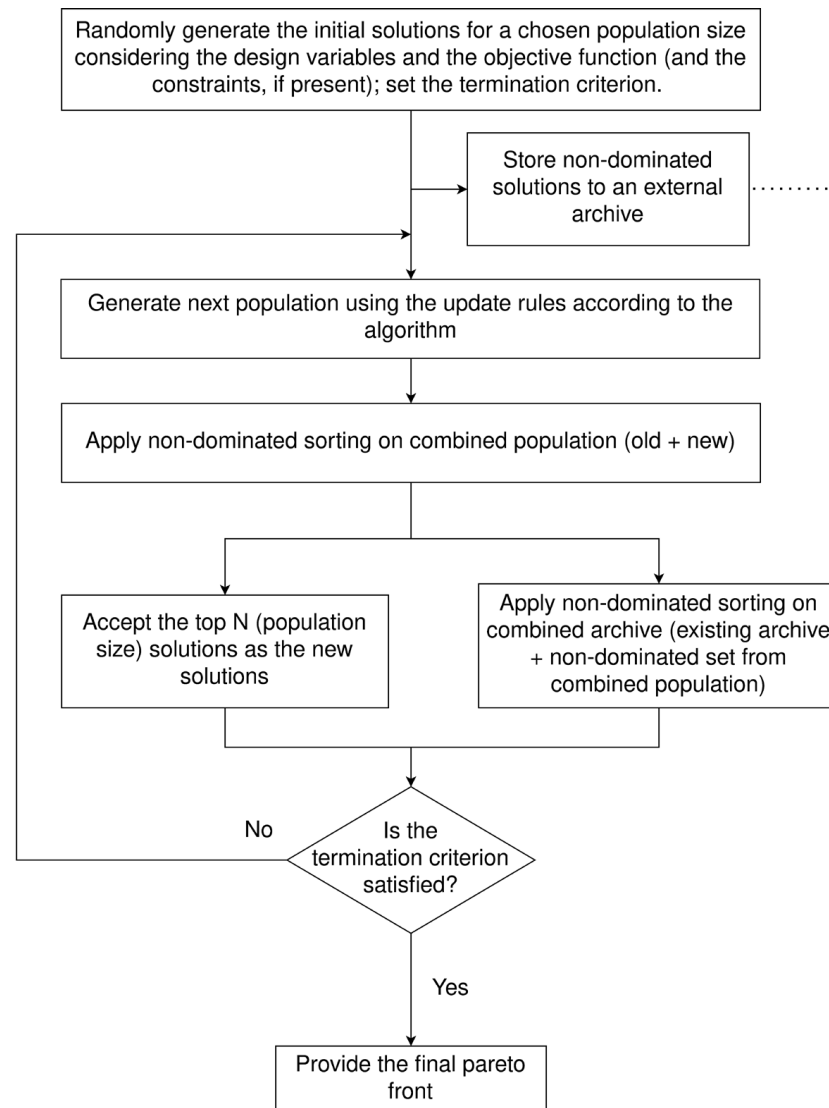


Figure 1. Flowchart for archive-based variants.

2.1.2. Opposition-Based Variants

Rao and Pawar [2] implemented a quasi-opposition approach, where the population is combined with its opposite population before applying the algorithm update rules. In this study, the simple opposition rule is applied, as shown in Equation (5). A parallel approach is implemented, where the algorithm generates a new population using the update rules and an opposite population is generated from the same population. After this, all three populations are combined and non-dominated sorting is carried out.

$$X_{opposite} = X_U + X_L - X_{old} \quad (5)$$

where $X_{opposite}$ is the opposite solution vector corresponding to the current solution vector X_{old} , and X_L and X_U denote the lower and upper bounds of the design variables, respectively. If any component of the generated opposite solution exceeds the prescribed variable bounds, it is clipped to the corresponding lower or upper bound to ensure feasibility of the generated solution vectors throughout the optimization process.

The flowchart for the process is shown in Figure 2.

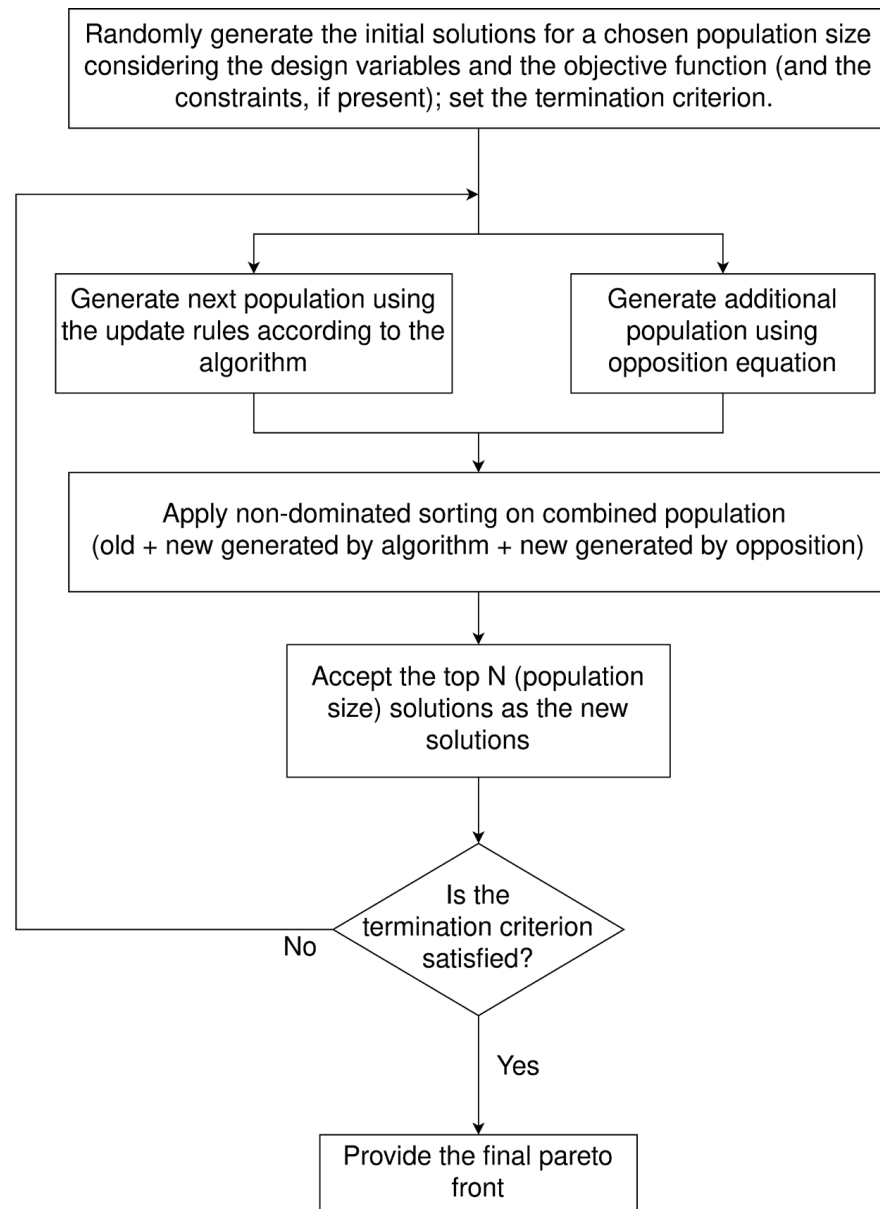


Figure 2. Flowchart for opposition-based variants.

2.1.3. Self-Adaptive Multi-Population-Based Variants

The multi-population method can improve the diversity of solutions by dividing the population to search for solutions in different regions of the search space. The number of sub-populations can be automatically adapted based on the performance of the whole population, using a performance metric such as hypervolume (HV). If the population shows improvement in the selected metric, it means the algorithm is finding better solutions, hence the number of sub-populations is incremented to improve exploration. If the population shows no improvement, this may indicate possible convergence, hence the number of sub-populations can be decreased to promote exploitation of the better regions. The flowchart for the SAMP-based algorithm is shown in Figure 3.

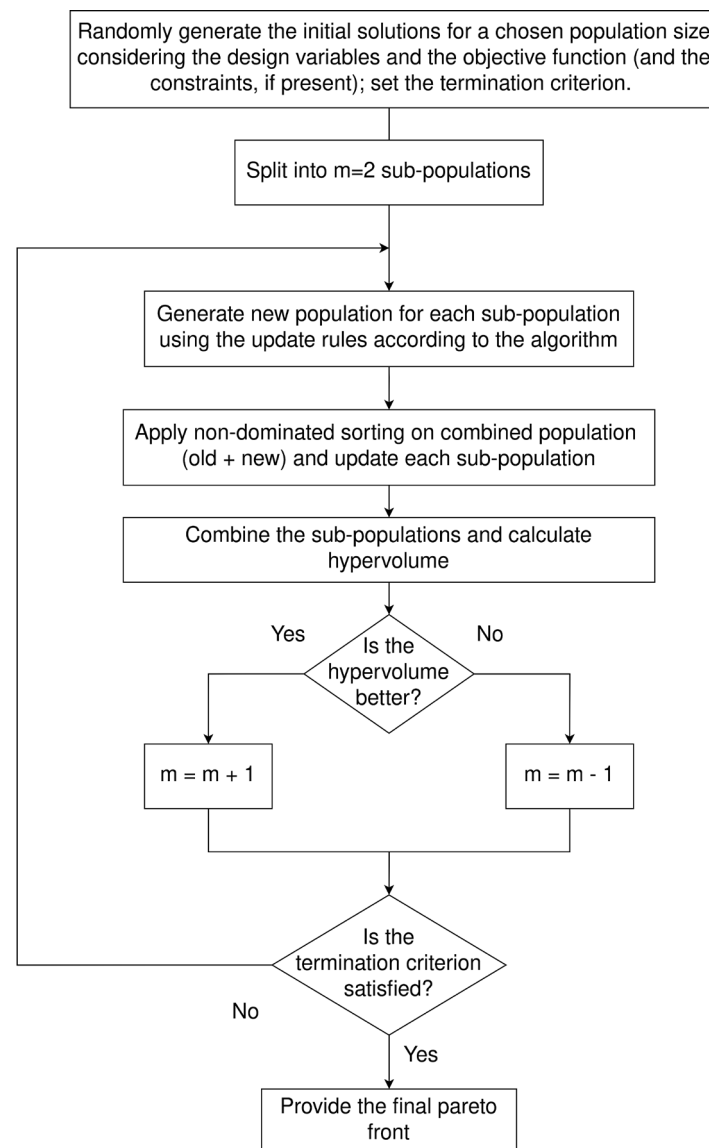


Figure 3. Flowchart for self-adaptive multi-population (SAMP)-based variants.

The original BxR algorithms are termed parameter-free because their update equations do not require algorithm-specific tuning parameters such as crossover probability, mutation rate, inertia weight, acceleration coefficients, or other tunable operators commonly required in conventional metaheuristic algorithms. However, it is important to note that parameters such as population size, function evaluation budget, and termination criterion are common experimental settings required by nearly all population-based optimization algorithms and are not unique to the proposed BxR formulations. In contrast, the enhanced variants introduce certain design-specific parameters, such as archive size in the archive-based variants and adaptation rules in the SAMP-based variants. These should therefore be regarded as design parameters and the proposed archive-based and SAMP algorithms may be considered as the reduced-parameter algorithms.

2.2. Response Surface Methodology

Response Surface Methodology (RSM) is a collection of statistical and mathematical techniques used to model, analyze, and optimize processes where response values are influenced by several inputs. RSM helps in optimization when direct computation or experimentation is expensive by replacing complex, time-consuming evaluations with a simpler

mathematical approximation of the system. Instead of testing every possible combination of variables, RSM uses a carefully designed set of experiments to build a predictive model (usually a quadratic equation) that represents the response surface. This model can then be analyzed quickly to identify optimal conditions with far fewer evaluations, significantly reducing computational cost and time while still providing reliable results.

2.3. Performance Metrics

Each algorithm is evaluated using various performance metrics. Since the problems considered in this study do not have known true Pareto fronts, a reference front is constructed for each problem using the following procedure. The final populations obtained from all 30 runs of the 16 algorithms are iteratively pooled. In each pass, non-dominated sorting is applied, and furthest point sampling retains 1000 well-distributed points while prioritizing extreme solutions along each objective. This process repeats across all 30 runs, ensuring that every algorithm contributes to the reference front, preventing any single algorithm from disproportionately shaping the reference set.

Inverted generational distance (IGD) and HV are adopted as the primary performance indicators. HV is the only known Pareto-compliant unary indicator [25], meaning that if one solution set strictly dominates another in the Pareto sense, HV will always assign a better value to the dominant set. It simultaneously evaluates convergence, diversity, and cardinality [26]. IGD captures both convergence to and coverage of the reference front in a single value, making it one of the most widely used comprehensive indicators, particularly in many-objective settings where HV computation becomes expensive [27]. Together, IGD and HV provide a robust assessment of overall algorithm quality.

Generational distance (GD) and spacing (SPC) serve as complementary diagnostic metrics. GD measures convergence by computing the average distance from each obtained solution to the nearest point on the reference front, but it is insensitive to diversity—an algorithm producing a single well-converged solution repeated many times would achieve an excellent GD value [26]. SPC evaluates the uniformity of spacing between adjacent solutions without regard to convergence or extent. These partial metrics are used to explain observed patterns in the primary IGD and HV results, following the established practice of combining composite and aspect-specific indicators.

While conclusions can be drawn from the mean and standard deviations of these metrics, a more appropriate analysis can be made using statistical tests. These tests can reveal significant differences between the tested algorithms and the practicality of the obtained results. To this end, detailed statistical testing described in the following section is adopted for each case study.

2.4. Statistical Analysis

To rigorously assess performance differences among the 16 algorithms, a non-parametric statistical testing framework is adopted following the guidelines put forward by Demšar [28]. Since the performance metric values obtained from 30 independent runs do not necessarily follow a normal distribution, the Friedman test is employed as an omnibus test to determine whether statistically significant differences exist among the algorithms. The Friedman test ranks the algorithms within each independent run and tests whether the observed rank differences are unlikely under the null hypothesis that all algorithms perform equivalently. When the Friedman test rejects the null hypothesis ($p < 0.05$), the Conover post hoc test with Holm's p -value adjustment, suggested by García et al. [29], is applied for pairwise comparisons. The Holm correction controls the family-wise error rate while offering greater statistical power than the Bonferroni correction, which is important given the 120 pairwise comparisons among 16 algorithms. Additionally, the

Vargha–Delaney A_{12} effect size statistic [30] is computed for each algorithm pair to quantify the practical magnitude of observed differences. The A_{12} value represents the probability that a randomly selected run of algorithm A yields a better result than a randomly selected run of algorithm B, with $A_{12} = 0.5$ indicating no difference.

The results are summarized using average rank diagrams with Holm cliques, following the approach of García et al. [29]. In these diagrams, algorithms are positioned along a horizontal axis according to their average Friedman rank, with the best-ranked algorithm on the left. Horizontal bars (Holm cliques) connect groups of algorithms whose pairwise differences are not statistically significant according to the Conover post hoc test with Holm correction ($p \geq 0.05$). Algorithms not connected by a common bar differ significantly. For each case study, these diagrams are presented for IGD and HV in the main text, and for GD and SPC in Appendix A. The final Pareto front of the top-ranked proposed variant by mean IGD rank, its base algorithm, and remaining algorithms are also used to further elaborate the findings where necessary.

Recent advances in many-objective optimization have emphasized the importance of simultaneously maintaining convergence, diversity, and scalability in high-dimensional objective spaces. Deb and Jain [31] introduced the reference-point-based NSGA-III framework to address the limitations of traditional Pareto-based methods in many-objective problems, while Xue et al. [32] further demonstrated the effectiveness of enhanced evolutionary mechanisms for improving convergence and diversity preservation. In addition, diagnostic benchmarking methodologies have been recommended to provide more reliable and comprehensive evaluation of many-objective optimization algorithms using complementary performance indicators and statistical analysis [33]. Archive maintenance strategies have also been recognized as effective mechanisms for preserving high-quality non-dominated solutions and improving diversity in evolutionary multi-objective optimization [34]. Similarly, opposition-based learning techniques have been widely investigated as mechanisms for improving exploration capability and convergence behavior in metaheuristic algorithms [35,36]. At the same time, Ishibuchi et al. [37] highlighted the difficulties associated with fair performance comparison in multi-objective optimization, particularly due to the differing sensitivities of performance indicators such as IGD and HV. These studies collectively motivate the present work, which investigates archive-based, opposition-based, and self-adaptive multi-population enhancement strategies within the MO-BxR framework for complex robotics optimization problems.

2.5. Comparative Algorithms

Along with the new variants, four well-established algorithms are selected in the present work for comparative analysis: Competitive Multi-Objective Particle Swarm Optimization (CMOPSO), Multi-Objective Particle Swarm Optimization with Crowding-distance (MOPSO-CD), and Non-Dominated Sorting Genetic Algorithms (NSGA2 and NSGA3). NSGA2 and NSGA3 are chosen as they are among the most widely adopted multi-objective optimization algorithms in the robotics literature, as evidenced by the review in Section 1, serving as a reliable benchmark for convergence and diversity. MOPSO-CD and CMOPSO represent the PSO family, which is the other dominant class of algorithms used in robotics optimization. MOPSO-CD employs crowding-distance for archive maintenance, providing a standard PSO-based baseline, while CMOPSO introduces a competitive mechanism that enhances convergence speed, offering a more recent PSO-based comparison. Together, these four algorithms span two major algorithmic paradigms (evolutionary and swarm-based) and provide a robust baseline for evaluating the proposed reduced-parameter MO-BxR variants. These existing algorithms are directly applied using their implementation in the Pymoo library, which is a popular Python library for optimization

introduced by Blank and Deb [38]. MO-BxR and their variants are implemented in Python 3.11.14, and all tests are conducted on a laptop equipped with an AMD Ryzen 5 5600H processor and 16 GB DDR4 RAM.

In the present study, the reference front is generated by pooling the final populations obtained from all 16 algorithms across 30 independent runs. The combined solution set is then processed using iterative non-dominated sorting followed by furthest point sampling to retain 1000 well-distributed solutions, while preserving extreme points along each objective. This procedure is intentionally adopted to reduce the possibility of any single algorithm disproportionately influencing the reference set. Furthermore, the same reference front is consistently used for evaluating all algorithms, ensuring a uniform basis for IGD and GD computation. The inclusion of solutions from all competing algorithms, together with diversity-preserving furthest point sampling, helps improve coverage and spread of the constructed reference front and reduces favoritism toward any particular method. It may be noted that such approximation-based reference front construction is a widely adopted practice in real-world multi-objective optimization studies where analytical Pareto fronts are unavailable, especially for complex engineering problems such as the robotics applications considered in the present work.

3. Case Studies

The effectiveness of MO-BxR algorithms in optimizing selected casting problems was demonstrated by Rao and Davim [1]. In addition, Rao et al. [39] further confirmed their applicability in heat transfer problems. The present work introduces improved variants of these algorithms and applies them to various robotics optimization problems through five different case studies, described in the following sections. Table 1 presents a summary of the robotics optimization problems considered, including the number of objectives, number of variables, surrogate models employed, number of function evaluations performed, and the original algorithms utilized by the researchers.

Table 1. Summary of robotics optimization problems.

Case Study	#Obj	#Var	Surrogate	FE Budget	Original Algo.	Ref.
AUV Shape Optimization	2	4	GEP	80,000	MOPSO-CD	Tang et al. [3]
Inspection Robot	3	6	RSM	20,000	NSPSO	Jin et al. [10]
Inverse Kinematics (4 DOF)	3	4	None *	6000	DMFO	Liu et al. [8]
Trajectory Planning	5	7	RSM	10,000	NSGA-3	Shi et al. [7]
Parallel Mechanism	6	3	RSM	10,000	MO-PSO	Wang et al. [19]

* Direct forward kinematics evaluation; no surrogate model required. FE = function evaluations. All cases use 30 independent runs per algorithm using seeds 1 to 30.

3.1. Shape Optimization of Autonomous Underwater Vehicle

Underwater robotics have seen an increase in interest due to their importance in exploration of marine resources and marine engineering maintenance. Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs) are popular options at present. Though ROVs have a huge impact on underwater operations, they still require mother ship support and manual intervention to complete missions. AUVs on the other hand do not require manual control, and can adapt automatically to complete tasks, yet the complexity of their working environment necessitates optimal design of their shape in order to reduce friction and improve endurance. The objective of this study is to optimize the shape of an AUV to minimize the resistance force (F) and maximize the surrounded

volume (V). F directly influences the efficiency of the vehicle and higher efficiency would allow the vehicle to operate longer, increasing its effectiveness in deep sea exploration. Surrounded volume is the volume of water displaced by the vehicle. It is a measure of how much volume is available inside the vehicle itself. A larger volume can accommodate more equipment for data collection and other operations, but it would increase the surface area which would directly increase resistance. So, both objectives must be balanced.

The AUV optimization problem is highly nonlinear, so it is best to develop and use surrogate models. Tang et al. [3] considered this problem and developed two different surrogate models; one using RSM and another using Gene Expression Programming (GEP). The GEP model developed by Tang et al. [3] is considered as the optimization problem due to its higher accuracy compared to the RSM model. Figure 4 shows the body of the AUV, and Table 2 lists the meaning of each design variable. The design vector is represented as $[x_1, x_2, x_3, x_4]$.

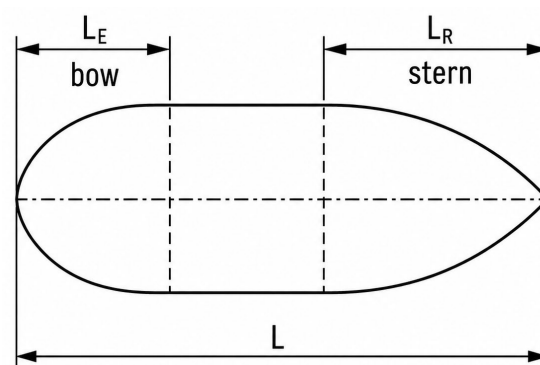


Figure 4. AUV body.

Table 2. AUV design variables [3].

Design Variable	Description
x_1	Ratio of length of bow to total length
x_2	Ratio of length of stern to total length
x_3	Shape factor of bow
x_4	Shape factor of stern

The optimization problem is given in Equation (6) [3].

$$\begin{aligned}
 &\text{Minimise } (f_1 = F, f_2 = \frac{1}{V}) \\
 &\text{s.t. } 0.148 \leq x_1, x_2 \leq 0.285 \\
 &\quad 1.5 \leq x_3 \leq 4 \\
 &\quad 1.5 \leq x_4 \leq 3
 \end{aligned} \tag{6}$$

Since the optimization framework is formulated as a minimization problem, the inverse of surrounded volume ($1/V$) is minimized which is equivalent to maximizing V .

The surrogate models developed by Tang et al. [3] for F and V based on GEP are given in Equations (7) and (8).

$$\begin{aligned}
 F = 10^{\arctan(\arccos(\max[(1.6126 - x_4)1.8852, \tan(x_1)]))} \\
 + \cos(x_{12.0651})x_3 + \frac{1}{x_2} + \max[x_4, -3.0430] \\
 + \arctan(\ln(5.4558 - \tan(\tan(8.4771 + x_4))))^2
 \end{aligned} \tag{7}$$

$$V = \sqrt[3]{\exp \left[\sqrt[3]{\cos(x_3) \left(\frac{1}{-9.5433} - x_2 \right)^2} \right]} + \sqrt[3]{\min \left[\arctan \left(\frac{1}{8.6543 - x_4 - \frac{x_3 + x_1}{2}} \right), \arctan \left(\frac{1}{x_3 + 6.6160} \right) \right]} + \sqrt[3]{\arcsin \left(\left(\max[0.1969, x_1] \right) 0.1174^2 - \min \left[\frac{x_1 + x_2}{2}, x_2 \right] \right)} \quad (8)$$

Tang et al. [3] used 80,000 function evaluations with the MOPSO-CD algorithm to solve this problem. After running the proposed algorithms with the same evaluation budget for 30 independent runs and conducting the statistical tests, the average rank results for IGD and HV with Holm cliques are visualized in Figures 5 and 6.

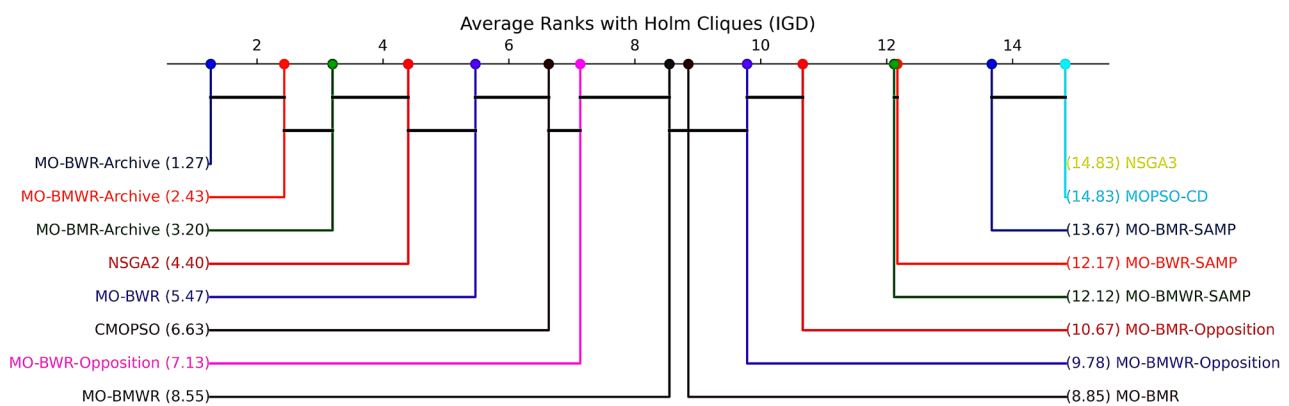


Figure 5. IGD average rank with Holm cliques for AUV shape optimization.

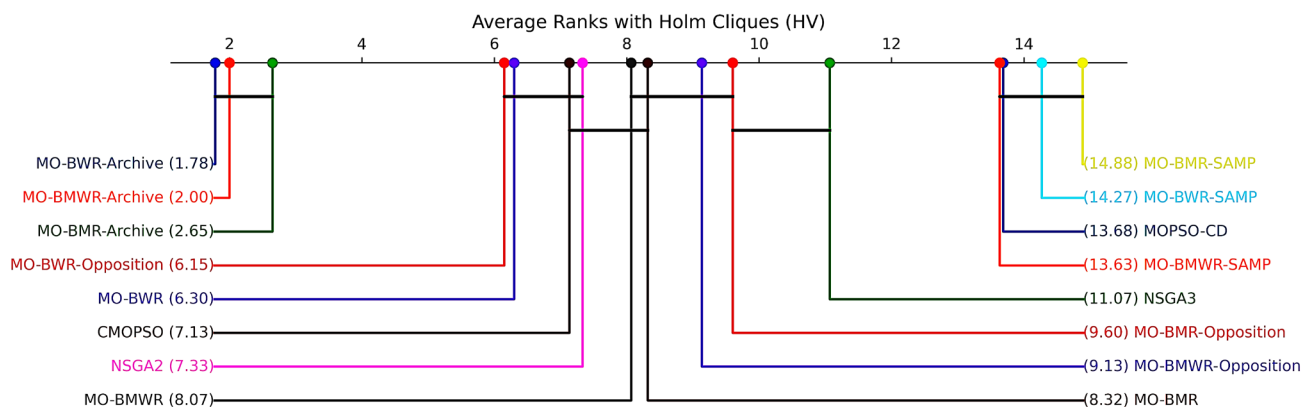


Figure 6. HV average rank with Holm cliques for AUV shape optimization.

Table 3 shows the mean values of the performance metrics for AUV optimization.

MO-BWR-archive ranks first on both IGD (1.3) and HV (1.8), along with other archive variants forming a statistically distinct top tier separated from all other algorithms by a clear gap in the average rank diagrams. The base MO-BWR ranks fifth on IGD (5.5) and fifth on HV (6.3), meaning the archive mechanism alone accounts for a jump of nearly four rank positions. Among established algorithms, NSGA2 (IGD rank 4.4) and CMOPSO (HV rank 7.1) perform reasonably but remain well behind the archive tier. NSGA2 got the lowest mean GD but is statistically comparable to a few other algorithms, including all the archive variants. For further clarity, the pairwise A_{12} values for MO-BWR-archive are listed in Table 4. The pairwise A_{12} values of 1.000 across all comparisons confirm complete dominance—MO-BWR-archive outperformed every competitor in every single run on both metrics. Although such complete separation is uncommon in stochastic optimization, the

consistency of results across 30 independent runs suggests that the observed dominance is not due to random variation but reflects a systematic performance advantage of the proposed method. The AUV optimization problem considered in this study consists of two objectives and four decision variables, together with a comparatively large evaluation budget of 80,000 function evaluations. Under these conditions, most algorithms achieve GD values within a relatively small range, indicating that the Pareto front is comparatively easy to approach. However, the MO-BWR-archive variant is able to retain high-quality non-dominated solutions across all 30 independent runs with very low variation (IGD std = 0.0003 and HV std = 0.0001), whereas several competing algorithms exhibit higher variability and occasional convergence instability.

Table 3. Performance metrics for AUV optimization (mean and standard deviation over 30 runs, sorted by mean IGD).

Algorithm	IGD (Mean)	HV (Mean)	GD (Mean)	SPC (Mean)
MO-BWR-Archive	0.0035 (0.0003)	0.9037 (0.0001)	0.0018 (0.0029)	0.0062 (0.0056)
MO-BMWR-Archive	0.0043 (0.0008)	0.9037 (0.0001)	0.0013 (0.0009)	0.0075 (0.0096)
NSGA2	0.0057 (0.0002)	0.9020 (0.0002)	0.0011 (0.0018)	0.0119 (0.0152)
MO-BWR	0.0060 (0.0004)	0.9021 (0.0002)	0.0026 (0.0029)	0.0107 (0.0132)
CMOPSO	0.0064 (0.0004)	0.9021 (0.0002)	0.0025 (0.0035)	0.0154 (0.0174)
MO-BWR-Opposition	0.0066 (0.0007)	0.9021 (0.0001)	0.0026 (0.0043)	0.0115 (0.0133)
MO-BMWR	0.0072 (0.0009)	0.9020 (0.0003)	0.0036 (0.0030)	0.0097 (0.0072)
MO-BMWR-Opposition	0.0091 (0.0044)	0.9015 (0.0014)	0.0050 (0.0034)	0.0186 (0.0177)
MO-BWR-SAMP	0.0121 (0.0034)	0.8972 (0.0035)	0.0048 (0.0035)	0.0209 (0.0172)
MO-BMWR-SAMP	0.0123 (0.0040)	0.8975 (0.0031)	0.0057 (0.0054)	0.0202 (0.0168)
MO-BMR-Archive	0.0123 (0.0422)	0.8914 (0.0658)	0.0080 (0.0335)	0.0068 (0.0081)
NSGA3	0.0184 (0.0024)	0.9016 (0.0000)	0.0023 (0.0029)	0.0445 (0.0257)
MOPSO-CD	0.0219 (0.0084)	0.8989 (0.0020)	0.0034 (0.0028)	0.0212 (0.0168)
MO-BMR-SAMP	0.0220 (0.0406)	0.8848 (0.0649)	0.0130 (0.0339)	0.0226 (0.0161)
MO-BMR	0.0228 (0.0580)	0.8776 (0.0914)	0.0155 (0.0459)	0.0109 (0.0084)
MO-BMR-Opposition	0.0243 (0.0576)	0.8773 (0.0913)	0.0157 (0.0450)	0.0123 (0.0089)

Values in brackets are the standard deviations.

Table 4. Pairwise A_{12} effect size comparison for MO-BWR-archive in AUV optimization.

Comparison	A_{12} (IGD)	A_{12} (HV)
MO-BWR-Archive vs. NSGA-2	1.000	1.000
MO-BWR-Archive vs. NSGA-3	1.000	1.000
MO-BWR-Archive vs. CMOPSO	1.000	1.000
MO-BWR-Archive vs. MOPSO-CD	1.000	1.000
MO-BWR-Archive vs. MO-BWR	1.000	1.000

Therefore, the observed A_{12} statistic likely reflects the combined influence of low problem dimensionality, a relatively large evaluation budget, and the stability advantage provided by the archive mechanism over more variable competing algorithms, rather than indicating universal superiority across all optimization scenarios.

It is also worth noting that the MO-BMR and MO-BMR-opposition variants exhibit noticeably high standard deviations on this problem (IGD of 0.0228 ± 0.0580 and 0.0243 ± 0.0576 respectively), indicating occasional runs where these algorithms fail to converge properly. The BWR and BMWR families do not show this instability, suggesting that the BMR update rule is less robust on this particular landscape. SAMP variants collapse to the bottom ranks (12–15 on both metrics), suggesting that population splitting is counterproductive in this low-dimensional objective space where the Pareto front is a simple one-dimensional curve.

The Pareto fronts in Figures 7–9 visually confirm these findings: MO-BWR-archive produces a dense, uniformly distributed front spanning the full trade-off range between resistance force and inverse volume, while NSGA2 shows comparable coverage but with noticeably sparser distribution in the knee region.

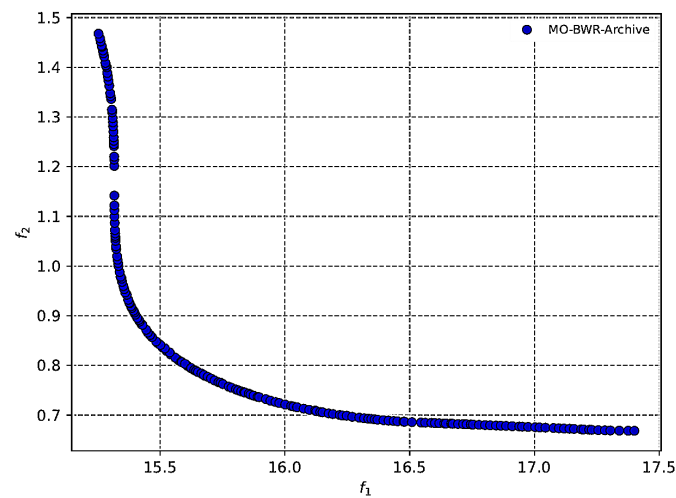


Figure 7. AUV Pareto front: MO-BWR-archive.

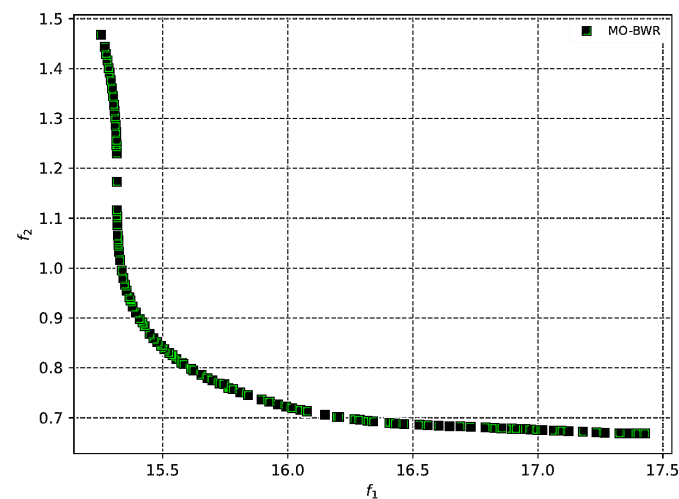


Figure 8. AUV Pareto front: MO-BWR.

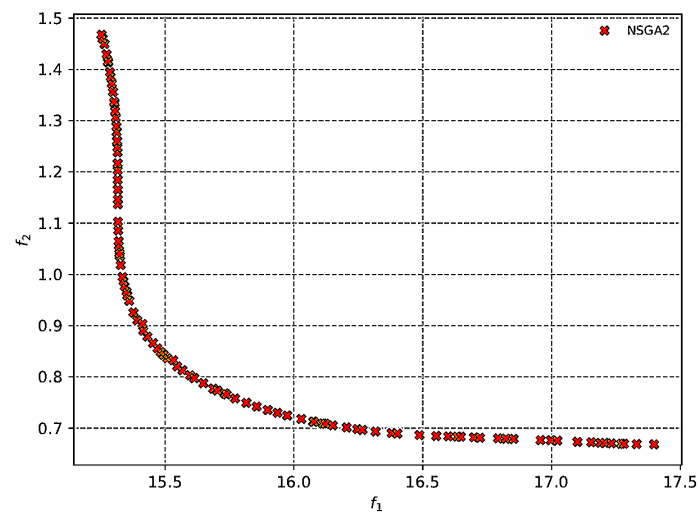


Figure 9. AUV Pareto front: NSGA2.

Table 5 presents representative solutions spanning the full Pareto front alongside the three solutions reported by Tang et al. [3]. The knee-region solution alone dominates all three of Tang et al.'s points, achieving lower resistance (15.574 vs. 15.685) and substantially lower inverse volume (0.810 vs. 0.900) compared to their best individual values on each objective. In total, 31 of the 187 solutions on the final front dominate all three reported solutions simultaneously. The front also extends well beyond the range explored by Tang et al.—the best f_1 solution achieves a resistance of 15.253 (2.8% lower than Tang et al.'s best), while the best f_2 solution reaches an inverse volume of 0.669 (25.7% lower) indicating substantially larger vehicle volume. Notably, design variables x_1 and x_2 remain nearly constant across the front ($x_1 \approx 0.22$, $x_2 = 0.285$), while x_3 (bow shape factor) drives the trade-off between resistance and volume, increasing from 1.50 at the low-resistance extreme to 3.08 at the high-volume extreme.

Table 5. Representative Pareto-optimal solutions of MO-BWR-archive and comparison with Tang et al. [3]. Both objectives are minimized (f_1 = resistance force F , f_2 = inverse volume $1/V$).

Solution	x_1	x_2	x_3	x_4	f_1	f_2
Tang et al. [3]—MOPSO-CD						
Point 1	—	—	—	—	15.685	1.003
Point 2	—	—	—	—	20.436	0.922
Point 3	—	—	—	—	22.123	0.900
MO-BWR-Archive (this study)						
Best f_1	0.2230	0.2850	1.5000	1.8808	15.253	1.468
Knee [†]	0.2230	0.2850	1.8584	1.8810	15.574	0.810
Mid-front	0.2215	0.2850	2.4535	1.8804	16.122	0.709
Best f_2	0.1480	0.2850	3.0812	1.8799	17.401	0.669

[†] Closest to the ideal point in normalized objective space; dominates all three solutions of Tang et al. [3].

3.2. Structural Parameter Optimization of Hybrid Inspection Robot

Power Transmission Line Inspection Robots (PTLIRs) have emerged as a highly promising solution for maintaining overhead power grids. Because transmission lines are continuously exposed to harsh natural environments, they are highly susceptible to corrosion, abrasion, and strand breakage. Regular inspection is essential to prevent power supply disruptions and ensure efficient maintenance practices.

Hybrid PTLIRs combine aerial and ground locomotion for more efficient performance. These robots consist of a flight device—a hexacopter UAV providing lift and mobility—and a walking device that suspends the robot beneath the transmission line for inspection. During operation, the robot ascends from the ground, aligns with the transmission line, and lands on it via the walking device. Once clamped, a motorized walking wheel propels the robot along the line. Since these robots operate at high altitudes on 110 kV transmission lines (towers typically 50–65 m high), they are subjected to significant wind forces making safe operation challenging. The robot considered in this study weighs 38 kg (6.4 kg walking device, 31.6 kg flight device) and has a total height of 1.4 m.

To ensure the safe operation of such devices, Jin et al. [10] proposed a novel walking device design and optimized it using metaheuristic techniques. The walking device employs a single-degree-of-freedom eight-link pressing mechanism consisting of a main walking wheel, main and auxiliary pressing wheels, connecting rods, a spring-loaded swing rod, and a linear motor-driven horizontal bar. The pressing mechanism clamps the robot onto the transmission line and must maintain contact under wind disturbance. Strong wind conditions can induce significant oscillations on the robot, which in turn rotates the robot about all three axes: α around the X-axis, β around the Z-axis, and γ

around the Y-axis. The magnitude of these deflection angles depends on the structural parameters of the pressing mechanism—specifically the lengths L_1, L_2, L_3, L_4, L_5 , and L_7 of the connecting rods, shown in Figure 10. The objective is to minimize deflections around each axis. Equation (9) shows the optimization problem, where f_α , f_β , and f_γ are the deflection angles α , β , and γ respectively.

$$\begin{aligned}
 &\text{Minimize} \quad \{f_1 = f_\alpha, f_2 = f_\beta, f_3 = f_\gamma\} \\
 &\text{s.t.} \quad 130 \text{ mm} \leq L_1 \leq 210 \text{ mm} \\
 &\quad \quad 30 \text{ mm} \leq L_2 \leq 90 \text{ mm} \\
 &\quad \quad 30 \text{ mm} \leq L_3 \leq 90 \text{ mm} \\
 &\quad \quad 40 \text{ mm} \leq L_4 \leq 80 \text{ mm} \\
 &\quad \quad 40 \text{ mm} \leq L_5 \leq 80 \text{ mm} \\
 &\quad \quad 200 \text{ mm} \leq L_7 \leq 250 \text{ mm}
 \end{aligned} \tag{9}$$

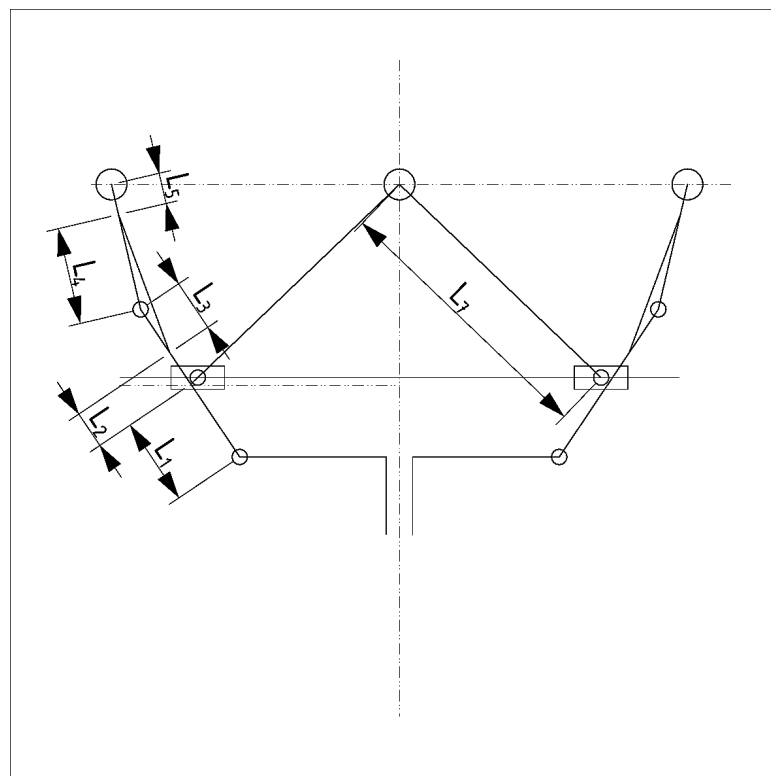


Figure 10. Structural parameters of the walking mechanism.

Equations (10)–(12) are the three polynomial regression response functions derived by Jin et al. [10], using a single pendulum model with force equilibrium analysis under level 7 wind (15.5 m/s).

$$\begin{aligned}
 f_\alpha = & 6.61 - 24.1 \times 10^{-3} L_1 - 18.1 \times 10^{-3} L_2 - 51.7 \times 10^{-3} L_3 \\
 & - 9.5 \times 10^{-3} L_5 + 27.1 \times 10^{-3} L_7 + 2.1 \times 10^{-4} L_1 L_3
 \end{aligned} \tag{10}$$

$$\begin{aligned}
 f_\beta = & 43.85 - 94.3 \times 10^{-3} L_1 - 91.2 \times 10^{-3} L_2 + 92.3 \times 10^{-3} L_3 \\
 & + 110.5 \times 10^{-3} L_4 - 114.1 \times 10^{-3} L_5 - 73.9 \times 10^{-3} L_7 - 2.1 \times 10^{-4} L_4 L_5
 \end{aligned} \tag{11}$$

$$f_\gamma = 0.74 + 10.5 \times 10^{-3} L_1 + 11.4 \times 10^{-3} L_7 + 1.9 \times 10^{-5} L_1 L_7 \tag{12}$$

Jin et al. [10] solved this problem using an improved non-dominated sorting Particle Swarm Optimization (NSPSO) algorithm with a population size of 100 over 200 iterations (20,000 function evaluations). To enable a fair comparison, each algorithm proposed in

this study is applied to the same problem with an identical budget of 20,000 function evaluations. Figure 11 shows the IGD average ranks with Holm cliques and Figure 12 shows the HV average ranks with Holm cliques for HIR optimization.

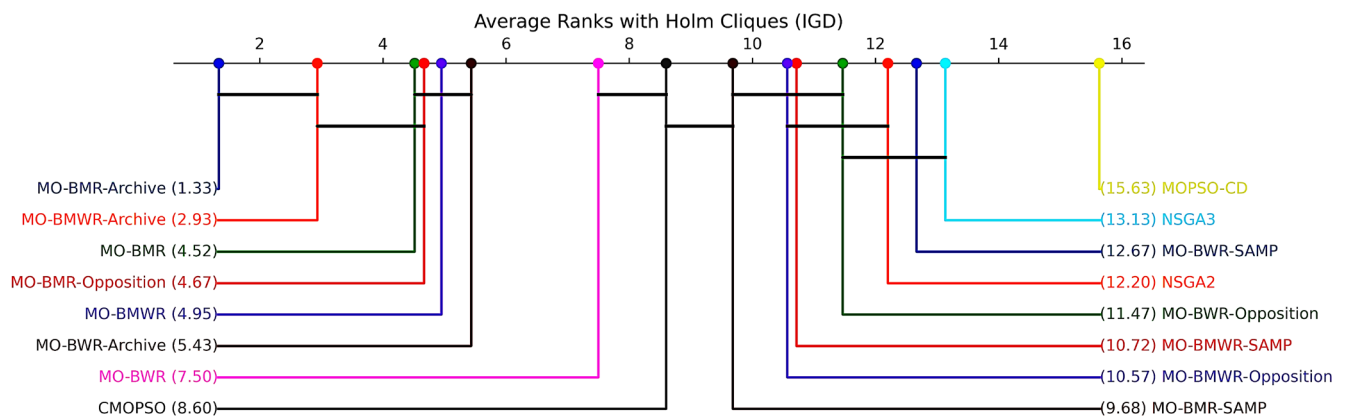


Figure 11. IGD average ranks with Holm cliques for HIR optimization.

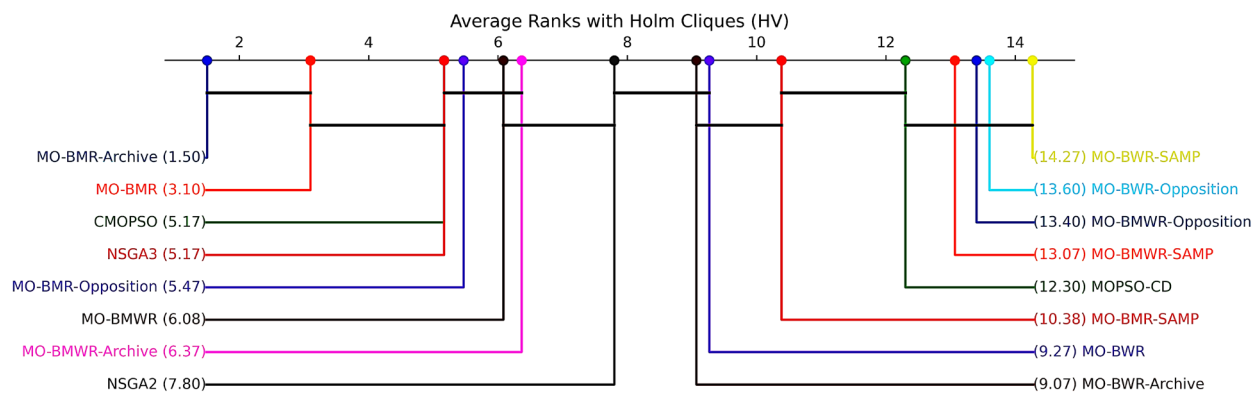


Figure 12. HV average ranks with Holm cliques for HIR optimization.

Table 6 shows the mean values of the performance metrics for HIR structural optimization.

Table 6. Performance metrics for HIR structural optimization.

Algorithm	IGD (Mean)	HV (Mean)	GD (Mean)	SPC (Mean)
MO-BMR-Archive	0.0224 (0.0018)	0.7262 (0.0038)	0.0161 (0.0025)	0.0209 (0.0061)
MO-BMWR-Archive	0.0261 (0.0031)	0.7171 (0.0057)	0.0226 (0.0039)	0.0210 (0.0050)
MO-BMR	0.0285 (0.0020)	0.7221 (0.0019)	0.0142 (0.0012)	0.0220 (0.0014)
MO-BMR-Opposition	0.0289 (0.0026)	0.7180 (0.0072)	0.0201 (0.0082)	0.0267 (0.0065)
MO-BMWR	0.0292 (0.0028)	0.7172 (0.0041)	0.0195 (0.0032)	0.0217 (0.0024)
MO-BWR-Archive	0.0306 (0.0067)	0.7109 (0.0120)	0.0281 (0.0096)	0.0202 (0.0035)
MO-BWR	0.0333 (0.0041)	0.7118 (0.0076)	0.0245 (0.0071)	0.0221 (0.0058)
CMOPSO	0.0349 (0.0036)	0.7193 (0.0020)	0.0156 (0.0011)	0.0138 (0.0026)
MO-BMR-SAMP	0.0368 (0.0049)	0.7091 (0.0089)	0.0289 (0.0205)	0.0324 (0.0065)
MO-BMWR-Opposition	0.0386 (0.0062)	0.6996 (0.0114)	0.0372 (0.0133)	0.0336 (0.0067)
MO-BMWR-SAMP	0.0387 (0.0042)	0.7026 (0.0066)	0.0314 (0.0069)	0.0303 (0.0052)
MO-BWR-Opposition	0.0402 (0.0064)	0.6978 (0.0122)	0.0390 (0.0148)	0.0325 (0.0065)
NSGA2	0.0419 (0.0058)	0.7161 (0.0012)	0.0146 (0.0006)	0.0203 (0.0022)
NSGA3	0.0432 (0.0014)	0.7192 (0.0010)	0.0141 (0.0005)	0.0112 (0.0015)
MO-BWR-SAMP	0.0437 (0.0066)	0.6957 (0.0093)	0.0389 (0.0129)	0.0323 (0.0058)
MOPSO-CD	0.0497 (0.0025)	0.7063 (0.0027)	0.0161 (0.0008)	0.0284 (0.0023)

MO-BMR-archive achieves the best mean IGD (0.0224) and HV (0.7262), ranking first on both metrics (1.3 in IGD and 1.5 in HV) with a substantial margin over the nearest established algorithm. The A_{12} values against all four established algorithms exceed 0.95 on IGD and HV confirming near-certain superiority across runs. A_{12} values also suggest that the archive mechanism provides a consistent performance gain over MO-BMR on this three-objective problem. NSGA3 does show the smallest GD value (0.0141) but it is statistically indistinguishable from MO-BMR in this metric. This also means that NSGA3's solutions converge well to the reference front individually and are uniformly spaced ($SPC = 0.0112$), but their poor IGD confirms that this convergence comes at the cost of coverage—they cluster in narrow regions rather than spanning the full front. It can also be observed that the MO-BMR-archive is very competitive in terms of GD with a value of 0.0161. Table 7 shows the pairwise A_{12} effect size comparison for MO-BMR-archive in hybrid inspection robot optimization.

Table 7. Pairwise A_{12} effect size comparison for MO-BMR-archive in hybrid inspection robot optimization.

Comparison	A_{12} (IGD)	A_{12} (HV)
MO-BMR-Archive vs. NSGA-2	1.000	0.967
MO-BMR-Archive vs. NSGA-3	1.000	0.964
MO-BMR-Archive vs. CMOPSO	0.999	0.953
MO-BMR-Archive vs. MOPSO-CD	1.000	0.999
MO-BMR-Archive vs. MO-BMR	0.972	0.866

The Pareto fronts in Figures 13–15 illustrate that the MO-BMR-archive covers the full three-dimensional trade-off surface densely, while NSGA3 is visibly missing solutions closer to f_1 .

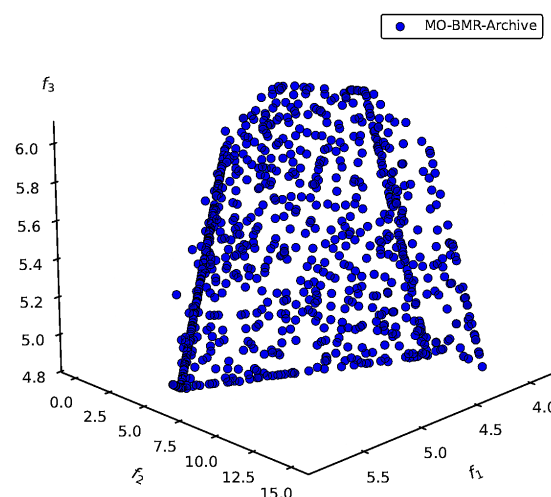


Figure 13. HIR Pareto front: MO-BMR-archive.

The representative solutions in Table 8 reveal the breadth of trade-offs captured by MO-BMR-archive. The solution minimizing f_1 achieves a deflection angle around the X-axis of just 3.90° , a 32% improvement over the best value reported by Jin et al. (5.53°), though at the cost of a higher Y-axis deflection ($f_3 = 6.02^\circ$). The solution minimizing f_2 effectively eliminates Z-axis deflection ($f_\beta = 0.02^\circ$) compared to Jin et al.'s best of 10.81° . The compromise solution provides balanced values for all three objectives and outperforms the corresponding trade-offs reported in the original study. The next section presents the case study of solving the inverse kinematics problem using optimization techniques.

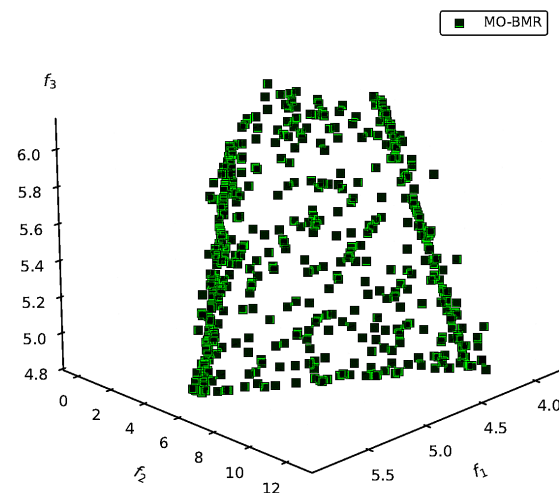


Figure 14. HIR Pareto front: MO-BMR.

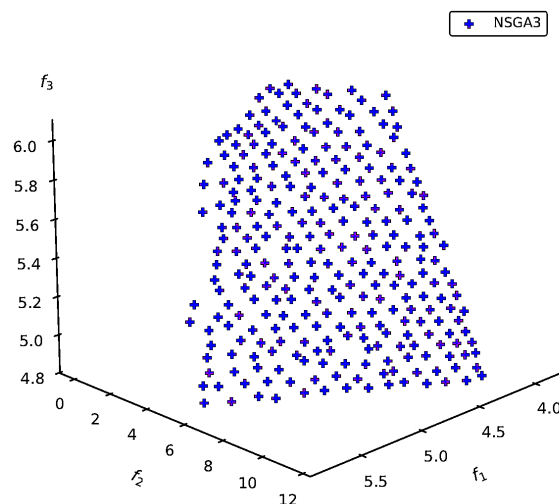


Figure 15. HIR Pareto front: NSGA3.

Table 8. Representative Pareto-optimal solutions of MO-BMR-archive and comparison with Jin et al. [10]. All objectives are minimized.

Solution	L_1	L_2	L_3	L_4	L_5	L_7	f_1	f_2	f_3
Jin et al. [10]—NSPSO									
Group 1	161.15	47.61	78.26	44.52	42.99	209.02	5.72	15.70	5.45
Group 2	161.21	67.91	60.80	64.09	67.57	212.77	5.53	10.81	5.51
Group 3	152.61	39.23	64.06	59.38	64.25	210.03	6.04	14.70	5.35
Group 4	147.32	73.98	53.83	80.00	54.56	208.93	5.75	14.44	5.25
Original	130.00	30.00	40.00	30.00	40.00	250.00	8.35	12.57	5.57
MO-BMR-Archive (this study)									
Best f_1	210.00	90.00	90.00	40.00	80.00	200.00	3.90	3.99	6.02
Best f_2	198.04	85.18	30.00	40.00	80.00	200.00	4.65	0.02	5.85
Best f_3	130.00	90.00	90.00	40.00	80.00	200.00	4.31	11.53	4.88
Compromise	157.99	89.92	73.33	40.00	79.97	200.00	4.48	7.36	5.28

3.3. Solving Inverse Kinematic Problem Using Optimization Techniques

Inverse kinematics is a fundamental problem in robotic manipulation, concerned with determining the joint angles required for the end effector to achieve a desired target position

and orientation in Cartesian space. The forward kinematics of a manipulator—computing end-effector pose from known joint angles—is straightforward and follows directly from the chain of homogeneous transformations along the kinematic links. Inverse kinematics, however, is considerably more complex. The mapping from Cartesian space back to joint space involves coupled nonlinear trigonometric expressions, and for manipulators with higher DOFs, these expressions become highly nonlinear. Multiple joint configurations can place the end effector at the same point, and near certain poses, the forward model loses invertibility altogether due to kinematic singularities. Analytical methods that attempt to directly invert the forward model struggle in these situations, and for many robot geometries, a general closed-form inverse solution does not exist.

Optimization techniques provide an effective alternative in such situations. Instead of inverting the kinematic equations, the idea is to search for joint angles that minimize the mismatch between the resulting end-effector pose and the desired target, evaluated directly by the forward model. The forward model is always available regardless of the robot's geometry, so this approach sidesteps the singularity and invertibility issues entirely. It also makes it easy to incorporate practical considerations, such as joint limits and a preference for smaller joint movements, within the same framework.

Liu et al. [8] took this approach and formulated a single-objective fitness function that accounts for positional accuracy, orientation accuracy, and a compliance term that penalizes large changes in joint angles—the rationale being that smaller joint movements translate to lower energy consumption and smoother motion. The three components were combined into one scalar value using weighting factors. While effective, this raises the usual difficulty with weighted sums: the relative scaling between position error (measured in meters), orientation error (measured in radians), and joint displacement varies across different robot configurations and target points, making the choice of weights somewhat arbitrary and problem-specific.

Rokbani et al. [5] took a different approach, treating the position and orientation errors as separate objectives and solving the problem using a Pareto-based strategy applied to a 5-DOF robotic arm. This avoids the weighting issue altogether and instead returns a set of non-dominated solutions representing different trade-offs—for example, one configuration might achieve near-perfect position accuracy at the cost of a less ideal orientation, while another balances both more evenly. The designer then picks from this set based on what the application actually requires.

In this study, the 4-DOF manipulator model from Liu et al. [8] is combined with the multi-objective formulation of Rokbani et al. [5], and the compliant criterion is added as a third independent objective rather than being folded into a weighted sum. This results in a three-objective optimization problem as shown in Equation (13). Treating the compliant criterion as an independent objective allows the optimizer to explicitly explore configurations where a slightly less accurate pose corresponds to smoother inter-joint transitions—for instance, favoring solutions in which adjacent joints change gradually rather than abruptly, which translates to simpler kinematic configurations.

$$\begin{aligned}
 \text{Minimize } f_o &= \sqrt{(r_x - r'_x)^2 + (r_y - r'_y)^2 + (r_z - r'_z)^2} \\
 \text{Minimize } f_p &= \left| [p_x, p_y, p_z] - [p'_x, p'_y, p'_z] \right| \\
 \text{Minimize, } f_c &= \frac{1}{\sum_{i=1}^n (\theta_i^{\max} - \theta_i^{\min})} \sum_{i=1}^n (\theta_{i+1} - \theta_i) \\
 &\text{s.t. } \theta_i^{\min} \leq \theta_i \leq \theta_i^{\max}, i = 1, \dots, n
 \end{aligned} \tag{13}$$

where f_o is the orientation error, f_p is the position error, f_c is the compliant criteria measuring the normalized sum of angular differences between consecutive joints—configurations with smaller inter-joint variations are preferred as they correspond to smoother, less complex postures. θ_i denotes the angle of the i th joint and n denotes the total number of joints.

The manipulator kinematic model is constructed using the Denavit–Hartenberg (DH) convention, which establishes the relationship between adjacent coordinate frames along the kinematic chain. The homogeneous transformation between two consecutive links is given in Equation (14).

$$A_i^{i-1} = \text{Trans}(0, 0, l_i) \text{Rot}(Z, \chi_i) \text{Trans}(\zeta_i, 0, 0) \text{Rot}(X, \eta_i) \\ = \begin{bmatrix} \cos\chi_i & -\sin\chi_i\cos\eta_i & \sin\chi_i\sin\eta_i & \zeta_i\cos\chi_i \\ \sin\chi_i & \cos\chi_i\cos\eta_i & -\cos\chi_i\sin\eta_i & \zeta_i\sin\chi_i \\ 0 & \sin\eta_i & \cos\eta_i & l_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

For n joints, the final transformation matrix becomes

$$A_0^n = \prod_{i=1}^n A_i^{i-1} = \begin{bmatrix} q_x & w_x & e_x & t_x \\ q_y & w_y & e_y & t_y \\ q_z & w_z & e_z & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & P_{1 \times 3} \\ 0_{1 \times 3} & 1 \end{bmatrix}$$

Details of the 4-DOF model used by Liu et al. [8] is given in Table 9.

Table 9. DH parameters and joint limits for the 4-DOF manipulator.

Joint i	θ_i (rad)	l_i (m)	ζ_i (m)	η_i (rad)	Joint Limits (rad)
1	θ_1	0	0.15	0	$[-1.7453, 1.7453]$
2	θ_2	0	0.20	0	$[-0.8727, 0.8727]$
3	θ_3	0	0	$\pi/2$	$[-1.7453, 1.7453]$
4	θ_4	0	0	0	$[-1.7453, 1.7453]$

The target matrix for optimization from Liu et al. [8] is shown below

$$A_0^4 = \begin{bmatrix} 0.6633 & -0.6671 & 0.3391 & 0.1 \\ 0.2391 & 0.2405 & 0.9407 & 0.3 \\ 0.7091 & 0.7051 & 6.123 \times 10^{-17} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Liu et al. [8] used fixed function evaluations of 6000 to optimize the single-objective variant. While this is smaller compared to the maximum evaluations used in multi-objective optimization, results show that for this particular problem it is sufficient. Table 10 shows the mean values of the performance metrics for the inverse kinematics problem.

Table 11 shows the pairwise A_{12} effect size comparison for MO-BMWR-archive in the inverse kinematics problem. Figure 16 shows the IGD average ranks with Holm cliques and Figure 17 shows the HV average ranks with Holm cliques for HIR optimization.

MO-BMWR-archive leads on both IGD (rank 2.6, mean 0.0399) and HV (rank 3.1, mean 0.8207), with all three archive variants occupying the top three positions on both metrics. The A_{12} against NSGA-2 (0.957 IGD, 0.948 HV) and MOPSO-CD (0.989 IGD, 0.971 HV) indicates strong practical superiority. Against CMOPSO, the effect is moderate (0.714 IGD, 0.833 HV), reflecting CMOPSO's comparatively better performance on this problem among the established algorithms. The base-to-archive improvement is statistically significant with A_{12} of 0.814 on IGD, confirming that archive mechanism provides a real rather than

marginal benefit. All SAMP variants rank at the bottom (13–14), similar to the pattern seen in the AUV case—population splitting does not help on lower-dimensional objective spaces. The Pareto fronts in Figures 18–20 show that the MO-BMWR-archive produces a well-spread front across all three objectives, including solutions in the practically important region where orientation accuracy and compliant cost are simultaneously low.

Table 10. Performance metrics for inverse kinematics problem.

Algorithm	IGD (Mean)	HV (Mean)	GD (Mean)	SPC (Mean)
MO-BMWR-Archive	0.0399 (0.0027)	0.8207 (0.0059)	0.0242 (0.0032)	0.0294 (0.0046)
MO-BWR-Archive	0.0406 (0.0033)	0.8187 (0.0076)	0.0249 (0.0052)	0.0281 (0.0041)
MO-BMR-Archive	0.0435 (0.0048)	0.8168 (0.0086)	0.0268 (0.0044)	0.0341 (0.0066)
MO-BWR	0.0438 (0.0039)	0.8118 (0.0099)	0.0272 (0.0058)	0.0361 (0.0069)
MO-BMWR	0.0446 (0.0052)	0.8123 (0.0104)	0.0288 (0.0069)	0.0360 (0.0070)
MO-BMR	0.0460 (0.0047)	0.8154 (0.0097)	0.0294 (0.0060)	0.0370 (0.0070)
MO-BMWR-Opposition	0.0513 (0.0040)	0.7975 (0.0116)	0.0336 (0.0056)	0.0415 (0.0066)
MO-BWR-Opposition	0.0534 (0.0054)	0.7932 (0.0134)	0.0349 (0.0089)	0.0442 (0.0109)
MO-BMR-Opposition	0.0535 (0.0047)	0.7972 (0.0098)	0.0355 (0.0058)	0.0410 (0.0045)
NSGA3	0.0537 (0.0142)	0.7953 (0.0215)	0.0418 (0.0190)	0.0350 (0.0141)
MOPSO-CD	0.0543 (0.0106)	0.7893 (0.0211)	0.0498 (0.0150)	0.0410 (0.0054)
CMOPSO	0.0559 (0.0205)	0.7864 (0.0331)	0.0515 (0.0262)	0.0289 (0.0061)
NSGA2	0.0582 (0.0187)	0.7915 (0.0261)	0.0476 (0.0194)	0.0306 (0.0035)
MO-BMR-SAMP	0.0678 (0.0076)	0.7812 (0.0136)	0.0451 (0.0102)	0.0509 (0.0106)
MO-BMWR-SAMP	0.0684 (0.0082)	0.7753 (0.0157)	0.0442 (0.0101)	0.0512 (0.0084)
MO-BWR-SAMP	0.0688 (0.0088)	0.7757 (0.0173)	0.0465 (0.0113)	0.0557 (0.0158)

Table 11. Pairwise A_{12} effect size comparison for MO-BMWR-archive in inverse kinematics problem.

Comparison	A_{12} (IGD)	A_{12} (HV)
MO-BMWR-Archive vs. NSGA-2	0.957	0.948
MO-BMWR-Archive vs. NSGA-3	0.873	0.906
MO-BMWR-Archive vs. CMOPSO	0.714	0.833
MO-BMWR-Archive vs. MOPSO-CD	0.989	0.971
MO-BMWR-Archive vs. MO-BMWR	0.814	0.741

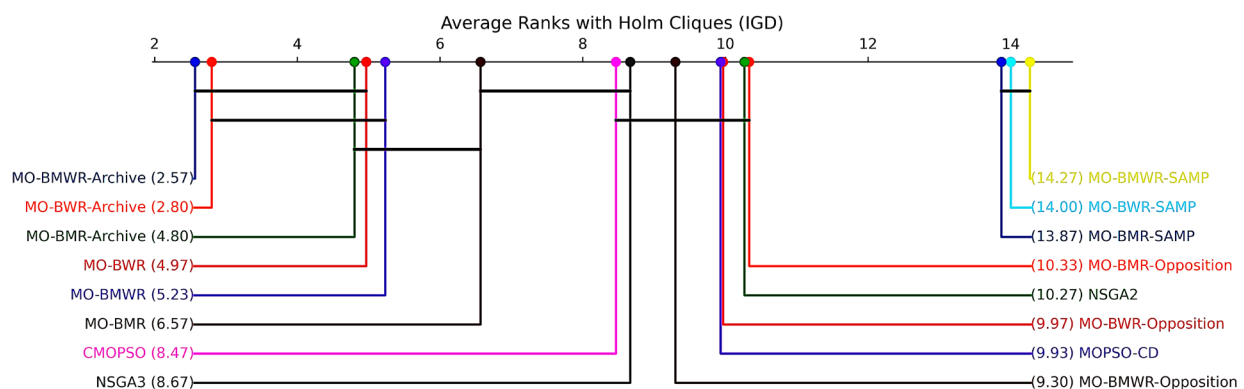


Figure 16. IGD average ranks with Holm cliques for inverse kinematic problem.

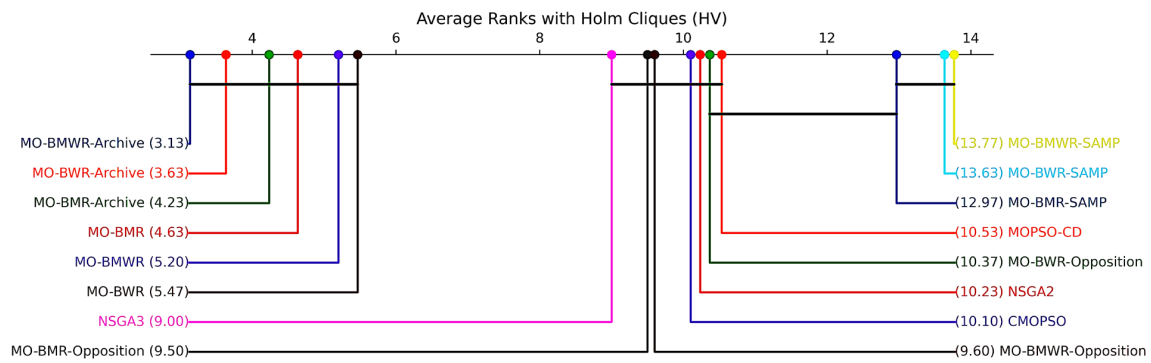


Figure 17. HV average ranks with Holm cliques for inverse kinematic problem.

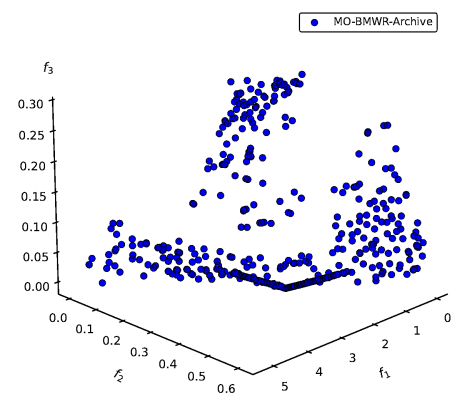


Figure 18. IK Pareto front: MO-BMWR-archive.

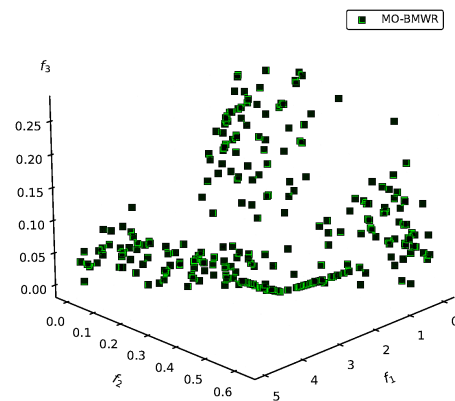


Figure 19. IK Pareto front: MO-BMWR.

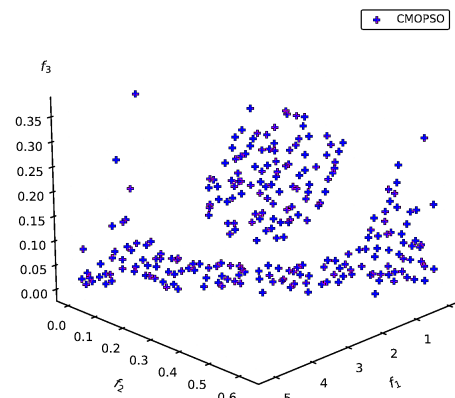


Figure 20. IK Pareto front: CMOPSO.

Table 12 shows the Pareto-optimal solutions of MO-BMWR-archive.

Table 12. Pareto-optimal solutions of MO-BMWR-archive.

θ_1	θ_2	θ_3	θ_4	f_1	f_2	f_3
−0.0001	−0.0000	−0.0001	−0.0001	2.3785	0.3905	0.0000
0.7452	0.8727	0.9212	0.7861	5.0217	0.0017	0.0255
0.7452	0.8727	0.9197	0.7868	5.0204	0.0017	0.0252
−0.3010	−0.8701	−1.1608	−0.3589	0.0251	0.5424	0.1360
−0.3020	−0.8727	−1.1529	−0.3573	0.0281	0.5427	0.1348

3.4. Wall-Building Robot Trajectory Optimization

Robots can be used to replace humans in hazardous and strenuous tasks in many fields. The construction sector is one such area where deployment of robots can reduce workplace injuries, improve efficiency, and convert construction into a high-tech industry. While the advantages are evident, wall-building robots have not yet become mainstream solutions in construction. According to Shi et al. [7], existing wall-building robots treat the process as an ordinary pick and place action, discounting the material properties of cement mortar. This leads to low precision and poor quality; due to which, wall-building robots are not preferred. To address this issue, Shi et al. [7] present a five-objective trajectory planning problem that minimizes the masonry error (f_1), completion time of the trajectory (f_2), energy requirement (f_3), contact force between cement mortar and robotic arm (f_4), and the instability of the trajectory (f_5). All five objectives are to be minimized. Masonry error and the contact force depend on the viscoelastic properties of cement mortar and Shi et al. [7] used Response Surface Methodology (RSM) to model this.

The trajectory is divided into two segments, the ABCD segment following a seventh-degree B-spline trajectory, and the masonry segment DE which follows a uniform linear motion, shown in Figures 21 and 22. Velocity, acceleration and jerk at the start and end of the trajectory are taken as the boundary conditions required to solve the seventh-degree B-spline trajectory. Acceleration and jerk are zero at both trajectory boundaries, but velocity is zero only at the start of the trajectory. The end velocity of each joint is calculated based on the masonry speed required in the DE segment. To uniquely define the trajectory curve while satisfying these boundary conditions and passing through the specified waypoints, Shi et al. [7] used a system that calculates a set of control vertices using the de Boer recursive formula. The time nodes for these vertices are normalized utilizing the cumulative chord length parameterization method. A key mathematical advantage of this B-spline formulation is its convex envelope property, which significantly simplifies the enforcement of the UR5 robot's physical limitations. Rather than evaluating the kinematic state at every infinite point along the continuous path, the strict joint constraints for maximum velocity, acceleration, and jerk are mathematically transformed into direct limits on the control vertices themselves. In the present work, the B-spline implementation of SciPy library is used to generate the ABCD segment trajectory.

Shi et al. [7] used a UR5 robot with six degrees of freedom (DOF) for optimization. In this paper the same robot is considered. The waypoints A, B, and C are fixed and the joint angles at these points are listed in Table 13. The location of D is calculated by first finding the coordinates of the point in robot coordinates using masonry height (given in wall coordinate system) and then using inverse kinematics.

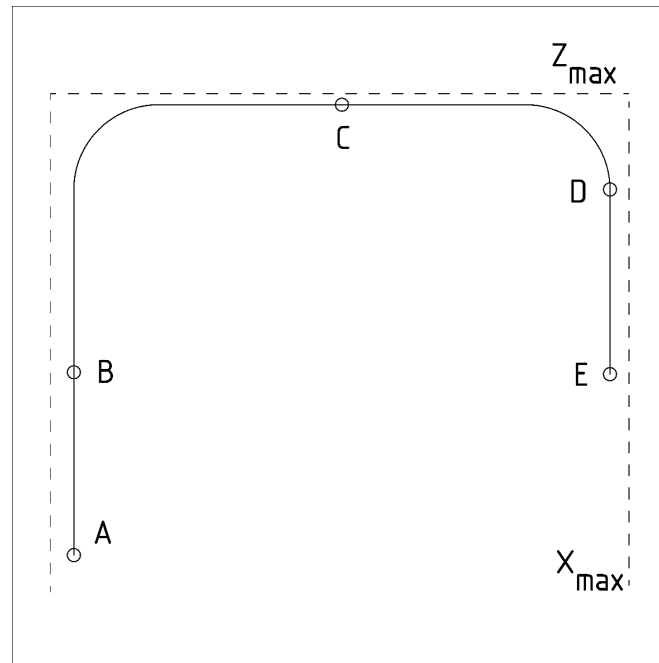


Figure 21. Wall-building trajectory.

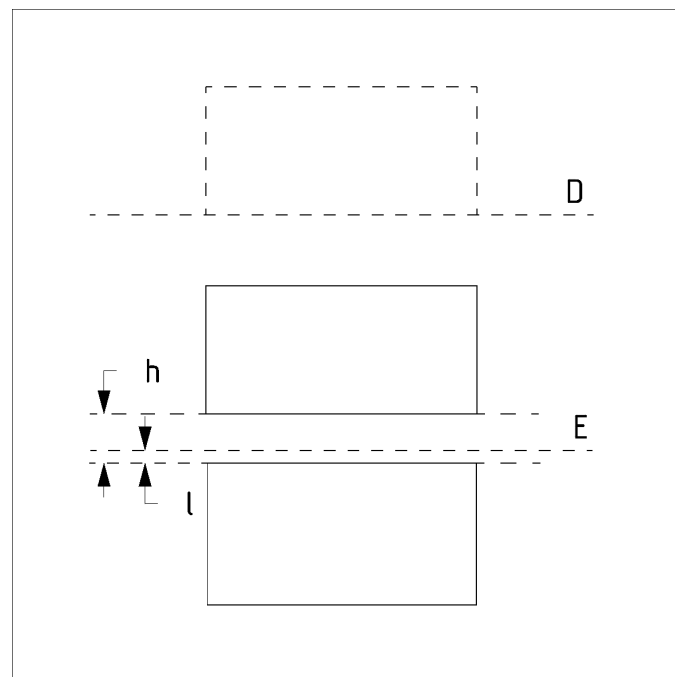


Figure 22. Viscoelastic compensation variables.

Table 13. Joint angles of waypoints [7].

Waypoints	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
A	0	−63.18	133.57	19.61	−90	−90
B	0	−76.33	131.69	34.64	−90	−90
C	0	−70.61	93.77	66.84	−90	−90

Variables associated with the problem and their description are given in Table 14.

Table 14. Variable description.

Variable	Description
x	Masonry Height (mm)
l	Position Compensation (mm)
v	Masonry Speed (m/s)
h	Mortar Thickness (mm)
t	Time (s)
i	Trajectory Segment Index
n	Number of Segments
m	Robot Joint Index
M	Total Number of Joints
v_m	Joint Speed ($^{\circ}/s$)
a_m	Joint Acceleration ($^{\circ}/s^2$)
j_m	Joint Jerk ($^{\circ}/s^3$)
τ_m	Joint Torque (Nm)

The ABCD segment passes through four waypoints with joint angles at A, B, and C fixed as listed in Table 13. The three variables t_1 , t_2 , and t_3 define the arrival times at waypoints B, C, and D respectively, which together with the boundary conditions (zero velocity at A, masonry speed at D, zero acceleration and jerk at both ends) uniquely determine the seventh-degree B-spline trajectory through the equation given by Shi et al. [7]. The four viscoelastic parameters (x , l , h , v) complete the seven design variables.

It can be seen from Table 13 that only three joints (2, 3, and 4) are actively moving, while others are fixed at particular joint angles. Hence g_4 to g_7 are only applied to these three joints and the limits are listed in Table 15.

Table 15. Motion constraints for each joint [7].

Constraint Item	Joint 2	Joint 3	Joint 4
$v_{m,max}$	80	80	60
$a_{m,max}$	150	150	100
$j_{m,max}$	1000	1000	500
$\tau_{m,max}$	120	120	22.4

The objective functions and constraints are given below:

$$f_1 = -6.32 + 0.0176x - 0.419l - 0.0219v + 0.43h \\ -0.000026x^2 - 0.0048l^2 - 0.000059v^2 - 0.00463h^2 \\ +0.000055xl + 0.000003xv - 0.000047xh \\ +0.0011lv + 0.00828lh + 0.000679vh \quad (15)$$

$$f_2 = \sum_{i=0}^{n-1} \Delta t_i = \sum_{i=0}^{n-1} (t_{i+1} - t_i) \quad (16)$$

$$f_3 = \sum_{m=1}^M \sqrt{\frac{1}{T} \int_0^T (a_i)^2 dt} \quad (17)$$

$$f_4 = 6222 + 1.41x - 50.5l - 2.05v + 51.9h \\ -0.00174x^2 + 1.667l^2 + 0.00187v^2 + 0.382h^2 \\ +0.0097xl + 0.0013xv - 0.0285xh \\ +0.1472lv + 4.773lh + 0.2907vh \quad (18)$$

$$f_5 = \sum_{m=1}^M \sqrt{\frac{1}{T} \int_0^T (j_i)^2 dt} \quad (19)$$

$$s.t. \left\{ \begin{array}{l} g_1 : X_{\min} \leq X_i(t) \leq X_{\max}, \\ g_2 : Z_{\min} \leq Z_i(t) \leq Z_{\max} \\ g_3 : |X_i(t) - X_{\text{straight}}| \leq \delta_{\max} \\ g_4 : |v_m(t)| \leq v_{m,\max} \\ g_5 : |a_m(t)| \leq a_{m,\max} \\ g_6 : |j_m(t)| \leq j_{m,\max} \\ g_7 : |\tau_m| \leq \tau_{m,\max} \\ g_8 : 252 \leq x \leq 400 \\ g_9 : 0.1 \leq l \leq 10 \\ g_{10} : 10 \leq v \leq 150 \\ g_{11} : 10 \leq h \leq 30 \\ g_{12} : f_1 > 0 \\ g_{13} : f_4 < 9500 \end{array} \right.$$

Constraints g_1 to g_3 are bounds on the space beyond which the robot should not move; $X_{\min} = Z_{\min} = 0$, $X_{\max} = 0.85m$ and $Z_{\max} = 0.5m$. g_4 to g_7 are the motion constraints of the robot, and g_8 to g_{13} are related to the RSM model for viscoelastic effects.

The magnitudes for v_m , a_m , and j_m are direct results of trajectory planning. τ_m is calculated using the Recursive Newton–Euler Algorithm (RNEA) from pinocchio library (Carpentier et al. [40]).

Finally, the optimization was carried out with a fixed evaluation budget of 10,000 evaluations, the same limit used by Shi et al. [7]. The resulting performance metrics (mean values of 30 runs) are given in Table 16. Figure 23 shows the IGD average rank diagram with Holm cliques and Figure 24 shows the HV average rank diagram with Holm cliques for WBR trajectory optimization.

Table 16. IGD and HV performance for wall-building robot trajectory optimization.

Algorithm	IGD (Mean)	HV (Mean)	GD (Mean)	SPC (Mean)
MO-BMR-SAMP	0.2001 (0.0257)	0.8868 (0.0075)	0.0481 (0.0053)	0.0865 (0.0183)
MO-BWR-SAMP	0.2147 (0.0289)	0.8910 (0.0056)	0.0451 (0.0051)	0.0796 (0.0109)
MO-BMR	0.2147 (0.0345)	0.8889 (0.0088)	0.0431 (0.0060)	0.0863 (0.0160)
MO-BMWR-SAMP	0.2155 (0.0290)	0.8873 (0.0089)	0.0454 (0.0042)	0.0829 (0.0164)
MO-BMWR	0.2181 (0.0244)	0.8863 (0.0098)	0.0438 (0.0069)	0.0837 (0.0146)
NSGA2	0.2276 (0.0583)	0.8250 (0.0301)	0.0534 (0.0039)	0.0503 (0.0077)
MO-BWR	0.2281 (0.0279)	0.8882 (0.0084)	0.0404 (0.0053)	0.0751 (0.0121)
MO-BMR-Archive	0.2450 (0.0355)	0.8647 (0.0370)	0.0482 (0.0055)	0.0870 (0.0223)
MO-BMWR-Archive	0.2534 (0.0359)	0.8804 (0.0169)	0.0459 (0.0051)	0.0763 (0.0178)
MO-BWR-Archive	0.2559 (0.0502)	0.8829 (0.0108)	0.0444 (0.0054)	0.0747 (0.0162)
MO-BWR-Opposition	0.2761 (0.0369)	0.8374 (0.0266)	0.0519 (0.0059)	0.1195 (0.0210)
MO-BMR-Opposition	0.2847 (0.0366)	0.8406 (0.0243)	0.0494 (0.0089)	0.1215 (0.0301)
MO-BMWR-Opposition	0.2858 (0.0363)	0.8291 (0.0453)	0.0542 (0.0080)	0.1179 (0.0192)
MOPSO-CD	0.2922 (0.0285)	0.7535 (0.0275)	0.0572 (0.0047)	0.1239 (0.0236)
NSGA3	0.3345 (0.0351)	0.7734 (0.0305)	0.0481 (0.0027)	0.0953 (0.0272)
CMOPSO	0.3899 (0.0464)	0.7480 (0.0397)	0.0439 (0.0018)	0.0283 (0.0066)

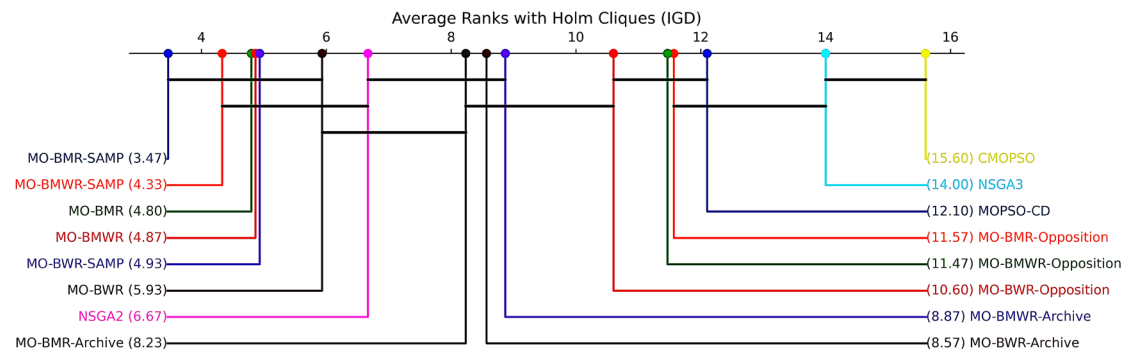


Figure 23. IGD average rank diagram with Holm cliques for WBR trajectory optimization.

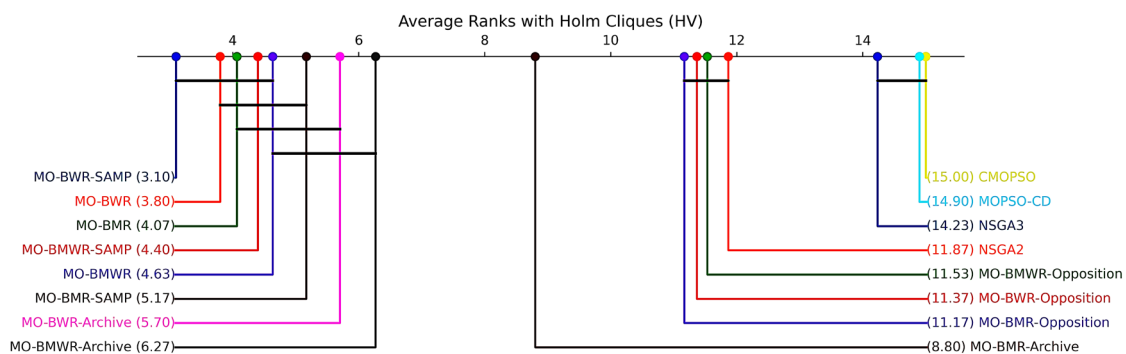


Figure 24. HV average rank diagram with Holm cliques for WBR trajectory optimization.

Table 17 shows the pairwise A_{12} effect size comparison for MO-BMR-SAMP in wall-building robot trajectory optimization.

Table 17. Pairwise A_{12} effect size comparison for MO-BMR-SAMP in wall-building robot trajectory optimization.

Comparison	A_{12} (IGD)	A_{12} (HV)
MO-BMR-SAMP vs. NSGA2	0.657	0.999
MO-BMR-SAMP vs. NSGA3	1.000	1.000
MO-BMR-SAMP vs. CMOPSO	1.000	1.000
MO-BMR-SAMP vs. MOPSO-CD	0.986	1.000
MO-BMR-SAMP vs. MO-BMR	0.604	0.397

This five-objective problem produces a markedly different ranking pattern from the lower-dimensional cases. SAMP and base variants now lead: MO-BMR-SAMP ranks first on IGD (3.5, mean 0.2001) and MO-BWR-SAMP first on HV (3.1, mean 0.8910), while the archive variants drop to mid-table (ranks 8–9 on IGD). The A_{12} values for MO-BMR-SAMP against established algorithms are decisive on HV (0.999–1.000) but more modest on IGD (0.657 against NSGA-2, 0.986–1.000 against others). Notably, the comparison with base MO-BMR is not significant, meaning the SAMP mechanism provides only a marginal IGD improvement over the base variant, though both massively outperform all established algorithms on HV. All four established algorithms occupy the bottom ranks (12–16) on both metrics, except for NSGA2 in IGD (rank 6.7), indicating that their convergence mechanisms are inadequate for this constrained five-objective trajectory problem. The parallel coordinate plots in Figures 25–28 show that MO-BMR-SAMP is able to find region of solutions that are missed by its base version, which explains the small improvement in IGD. Compared to MO-BMR-SAMP, the MO-BWR-SAMP and the base variants find a few extreme solutions in one or more objective, contributing to larger HV values. The parallel plot of MO-BMR

shows a high concentration of solutions in some regions, which would explain its lower mean GD value, but due to poor coverage, scores higher on IGD.

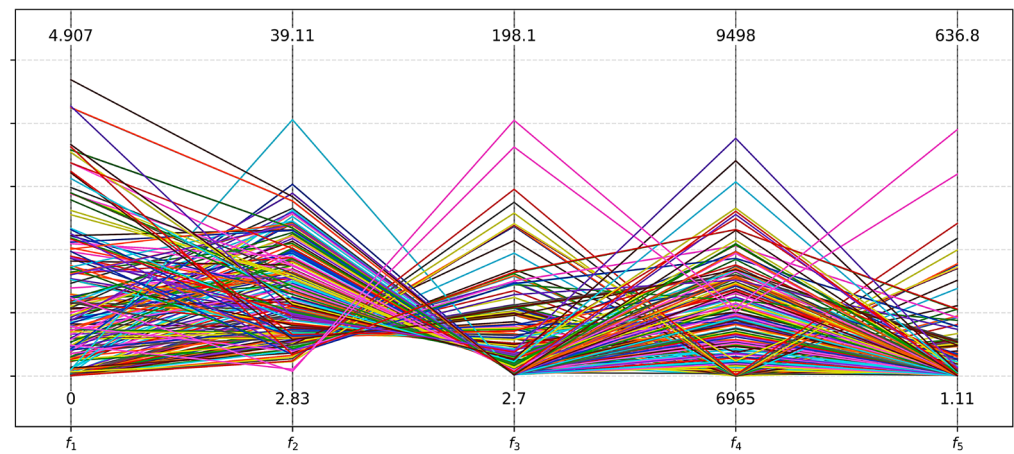


Figure 25. WBR Pareto front: MO-BMR-SAMP.

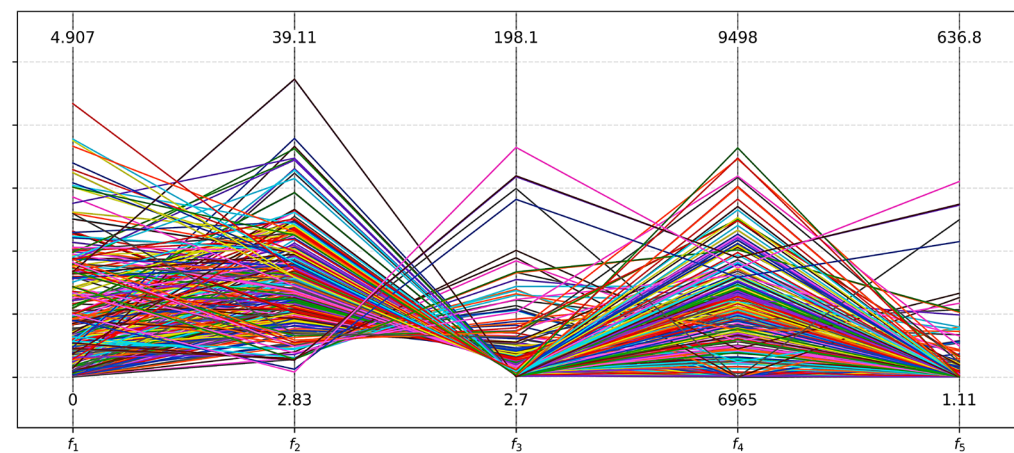


Figure 26. WBR Pareto front: MO-BMR.

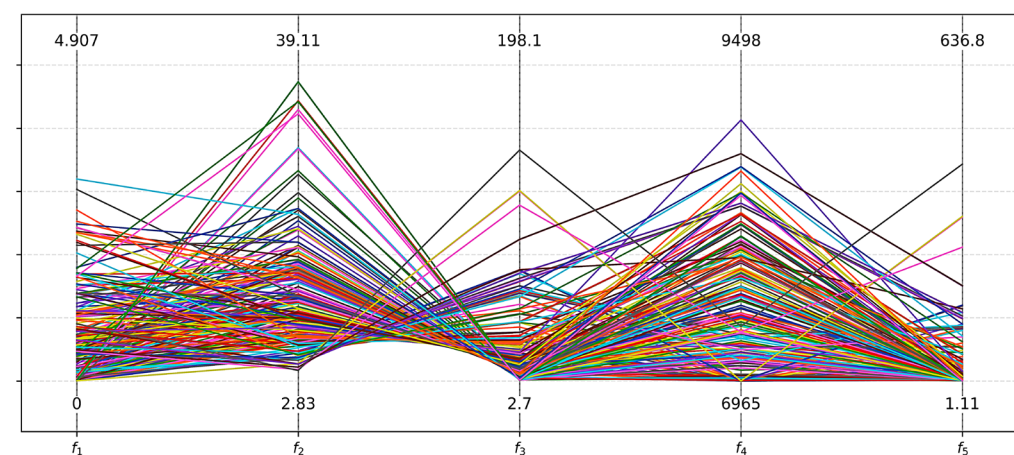


Figure 27. WBR Pareto front: MO-BWR-SAMP.

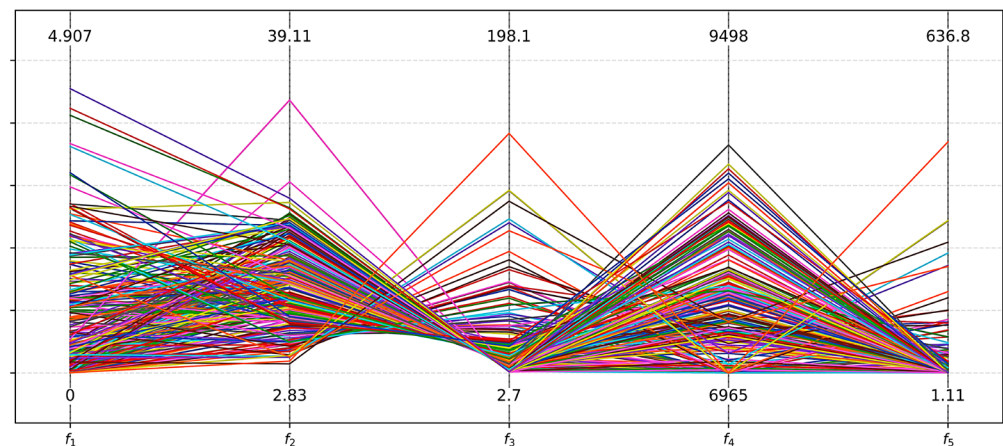


Figure 28. WBR Pareto front: MO-BWR.

Table 18 compares representative solutions from the MO-BMR-SAMP Pareto front against the TOPSIS solution identified by Shi et al. [7], with all objective values computed using the same trajectory evaluation model to ensure a consistent comparison. The compromise solution dominates the TOPSIS solution on all five objectives, achieving a 98% reduction in masonry error (f_1 : 0.002 vs. 0.127), a 5% reduction in contact force (f_4 : 6965 vs. 7317), and improvements in completion time, energy, and stability. Four solutions on the front dominate the TOPSIS solution simultaneously.

Table 18. Representative Pareto-optimal solutions of MO-BMR-SAMP and comparison with the TOPSIS solution by Shi et al. [7]. All objectives are minimized.

Solution	t_1 (s)	t_2 (s)	t_3 (s)	x (mm)	l (mm)	h (mm)	v (mm/s)	f_1 Error	f_2 Time	f_3 SQ	f_4 F_{max}	f_5 SJ
Shi et al. [7]—TOPSIS solution												
TOPSIS (E)	2.94	5.08	7.67	279.7	2.43	13.6	43.6	0.127	8.742	27.58	7317.0	33.77
MO-BMR-SAMP (this study)												
Compromise [†]	2.03	4.01	7.24	252.0	0.10	10.0	10.0	0.002	8.555	25.74	6965.0	30.20
Best f_1	0.87	2.67	5.11	252.0	0.11	10.0	10.0	0.002	6.426	86.47	6965.0	193.59
Best f_2	0.73	1.70	3.06	252.0	0.10	16.5	63.0	1.212	3.369	160.80	7511.4	496.85
Best f_3/f_5	5.72	11.19	20.00	292.2	5.71	16.3	13.5	0.414	24.83	3.21	7577.7	1.35
Best f_4	3.34	6.08	9.40	252.0	0.10	10.0	10.0	0.002	10.717	17.52	6965.0	17.07

[†] Closest to the ideal point in normalized objective space; dominates the TOPSIS solution on all five objectives. SQ = Sum of squared joint torques and SJ = Sum of squared joint jerks.

The extreme solutions reveal the full scope of trade-offs available to the designer. Minimizing completion time ($f_2 = 3.37$ s) requires aggressive joint motion that increases energy and jerk by an order of magnitude, while minimizing energy and jerk ($f_3 = 3.21$, $f_5 = 1.35$) demands a slow trajectory ($f_2 = 24.8$ s) with interior values of the viscoelastic variables. A notable pattern emerges in the design variables: masonry height (x) converges to its lower bound of 252 mm in 80% of the solutions, and position compensation (l) reaches its lower bound in over half the front. This indicates that for the present wall-building configuration, the optimizer consistently favors minimal masonry height and position compensation, with the primary trade-offs governed by the time variables (t_1, t_2, t_3) and masonry speed (v).

These results paint a nuanced picture. The base BxR algorithms are already very strong on this problem, and while SAMP variants rank marginally higher on mean IGD, the improvement is not statistically significant—MO-BMR-SAMP versus MO-BMR yields an A_{12} of just 0.604 on IGD and 0.397 on HV, the latter actually favoring the base variant. The real finding here is not that SAMP provides a breakthrough, but that the algorithm-specific

parameter-free BxR algorithms—with or without enhancements—substantially outperform all four established algorithms on HV (A_{12} of 0.999–1.000), demonstrating their strength on heavily constrained many-objective problems. The SAMP mechanism shows promise in this setting but requires further development to produce statistically meaningful gains over the already-competitive base algorithms.

3.5. Optimization of Reconfigurable Parallel Mechanism for Cutting and Grinding Operations

Parallel mechanisms are mechanical systems in which multiple kinematic chains connect a fixed base to a moving platform, providing high stiffness, accuracy, and load-carrying capacity compared to serial manipulators. These mechanisms are widely used in applications requiring precision and stability, such as robotics and machining. However, traditional parallel mechanisms have fixed configurations and limited adaptability. This limitation can be mitigated using reconfigurable versions, which allow the mechanism to change its structure or degrees of freedom to perform different tasks, thereby enhancing flexibility and broadening their range of applications.

Complex components often need to be split into simpler subcomponents, manufacturing each part individually using specialized machines and then assembling them. This can increase equipment cost and limit the quality of such complex parts. In situ machining is a potential solution for this challenge. It involves the use of robots that can move around, machine and inspect the part in its original location, without the need for disassembly and reassembly. These robots must have the capability to carry out multiple machining processes as required. To this end, reconfigurable parallel mechanisms are the optimal choice as they can switch between multiple modes. And since the structural parameters are fixed in each mode, it can be optimized such that each mode can support a particular machining operation. Wang et al. [19] presented a reconfigurable parallel mechanism with two modes: cutting and grinding. Figures 29 and 30 show the cutting and grinding mode configurations. The main difference between the two modes is the degree of freedom (DOF) associated with them. Cutting mode has one translational and two rotational DOFs (1T2R), while grinding mode has two translational and one rotational DOF (2T1R). In cutting mode, the mechanism needs to be agile in order to process complex workpieces; having two rotational DOFs helps in this regard. Meanwhile, grinding is a finishing process where the mechanism will be exposed to significant impact forces and the 2T1R configuration can provide higher load capacity and better stability.

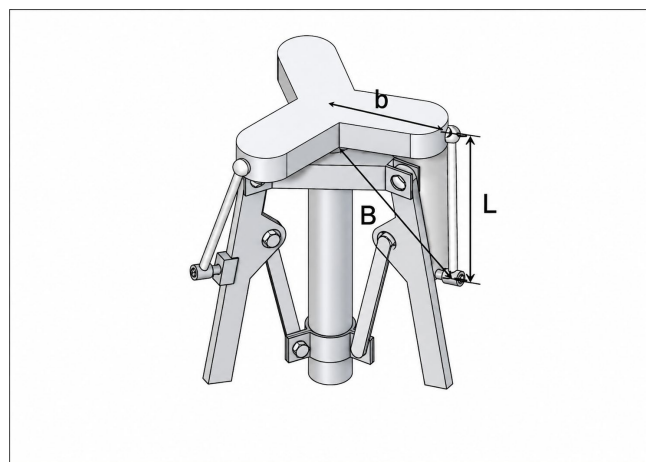


Figure 29. Parallel mechanism in cutting mode.

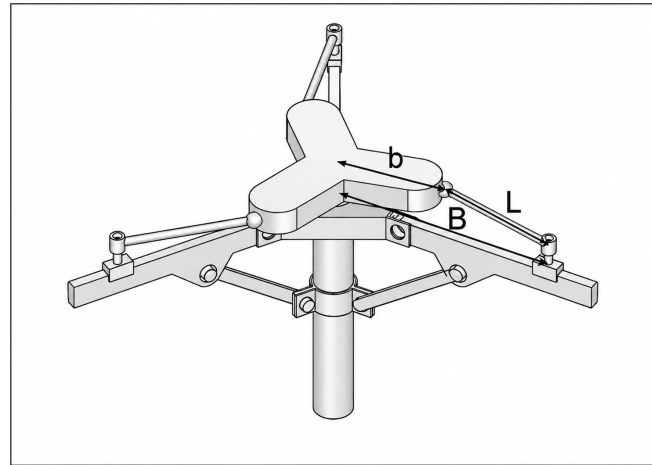


Figure 30. Parallel mechanism in grinding mode.

Wang et al. [19] formulated a general optimization problem for the reconfigurable mechanism, considering cutting mode, grinding mode, and transition modes. Cutting mode requires agility, and strong motion/force transmission capability. Therefore, virtual power is selected as a performance indicator, and consequently two objectives were introduced: mean virtual power $\bar{\eta}_W$ and its variance $\hat{\eta}_W$. Higher $\bar{\eta}_W$ indicates good motion/force transmission and $\hat{\eta}_W$ relates to the overall stability. Grinding mode needs to withstand impact forces, hence the maximum load capacity $\bar{F}_{max}$ was considered as the next indicator. Both cutting and grinding modes must ensure as large a workspace as possible; in the case of cutting it is the workspace volume V_{1T2R} and for grinding it is the area S . The stability of the transition mechanism, measured using the fluctuation index ($\hat{v}$), between the two modes must also be optimized.

The primary design variables are moving platform radius b , fixed platform radius B , and link length L , represented in Figures 29 and 30. These design variables are not present directly in the RSM models; instead, the following modified variables are used: b , $\lambda_B = B/b$, and $\lambda_L = L/b$. This modification is done in order to address the issues related to differences in units and scales of the variables. Table 19 lists the range of each modified variable.

Table 19. Modified design variables and their ranges.

Modified Design Variables	b	λ_B	λ_L
Range	[80, 120]	[0.75, 1.05]	[1.10, 1.30]

The optimization problem can be formulated as follows:

$$\begin{aligned} & \text{Maximize} \{f_1 = \bar{\eta}_W, f_3 = V_{1T2R}, f_4 = S, f_5 = \bar{F}_{max}\} \\ & \text{Minimize} \{f_2 = \hat{\eta}_W, f_6 = \hat{v}\} \end{aligned}$$

The calculation of these objectives using analytical equations is very complex and time consuming, which is detrimental to optimization algorithms. To avoid this, Wang et al. [19] employed Response Surface Methodology to develop high-accuracy surrogate models for each objective.

The RSM models developed by Wang et al. [19] are given in Equations (20)–(25).

$$\begin{aligned} \text{Mean Virtual Power } (\bar{\eta}_W) = & -3432.52053485 - 6.21861267b + 13908.56659919\lambda_B \\ & + 1739.85437824\lambda_L + 0.09077242b^2 - 23574.95864703\lambda_B^2 \\ & - 2176.21861868\lambda_L^2 + 0.00702898b\lambda_B - 0.00239827b\lambda_L \\ & - 2.57075884\lambda_B\lambda_L - 0.00058842b^3 + 17733.02708304\lambda_B^3 \\ & + 1209.69560550\lambda_L^3 + 1.429 \times 10^{-6}b^4 - 4993.83500365\lambda_B^4 \\ & - 251.74415634\lambda_L^4 \end{aligned} \quad (20)$$

$$\text{Variance of Virtual Power } (\hat{\eta}_W) = -0.63418796 + 0.0000213b + 1.02576783\lambda_B - 0.10525385\lambda_L \quad (21)$$

$$\begin{aligned} \text{Cutting Workspace Volume } (V_{1T2R}) = & 49582134.03146700 - 141997.43299023b - 209863908.10123200\lambda_B \\ & + 1134167.69712162\lambda_L + 2197.88225988b^2 + 355404454.09198600\lambda_B^2 \\ & - 2092057.70252440\lambda_L^2 - 996.75469469b\lambda_B - 460.91139663b\lambda_L \\ & + 119185.17454121\lambda_B\lambda_L - 14.94064098b^3 - 266637521.17325500\lambda_B^3 \\ & + 4.86183301\lambda_L^3 + 0.03805668b^4 + 74751229.37562560\lambda_B^4 \\ & - 374138.36239284\lambda_L^4 \end{aligned} \quad (22)$$

$$\begin{aligned} \text{Grinding Workspace Area } (S) = & 68898.97051581 - 445.58791653b + 1358.44806545\lambda_B - 100196.89445736\lambda_L \\ & + 1.44761511b^2 - 1545.32830287\lambda_B^2 + 39740.10716568\lambda_L^2 \\ & + 0.94212994b\lambda_B + 322.89733917b\lambda_L - 83.71115823\lambda_B\lambda_L \\ & - 0.00191225b^3 + 590.84556373\lambda_B^3 - 2554.59470115\lambda_L^3 \end{aligned} \quad (23)$$

$$\begin{aligned} \text{Maximum Load Capacity } (\bar{F}_{max}) = & -2896.84517833 + 2.26246212b - 287.80447259\lambda_B + 9365.17110769\lambda_L \\ & - 0.00088772b^2 + 288.68258499\lambda_B^2 - 11107.70189219\lambda_L^2 \\ & - 0.31908913b\lambda_B + 0.05912618b\lambda_L + 62.02176312\lambda_B\lambda_L \\ & + 3.855 \times 10^{-6}b^3 - 179.95070708\lambda_B^3 + 5881.11713850\lambda_L^3 \\ & - 4.355 \times 10^{-9}b^4 + 44.99995029\lambda_B^4 - 1182.03132555\lambda_L^4 \end{aligned} \quad (24)$$

$$\begin{aligned} \text{Fluctuation Index } (\hat{\vartheta}) = & 0.03207663 + 9.921 \times 10^{-7}b - 0.13237850\lambda_B - 1.489 \times 10^{-8}b^2 \\ & + 0.22002280\lambda_B^2 - 3.857 \times 10^{-9}b\lambda_B + 9.9219 \times 10^{-11}b^3 \\ & - 0.16016749\lambda_B^3 - 2.471 \times 10^{-13}b^4 + 0.04540062\lambda_B^4 \end{aligned} \quad (25)$$

$$s.t. \left\{ \begin{array}{l} 0.3958 \leq \bar{\eta}_W \leq 0.8254 \\ 0.0526 \leq \hat{\eta}_W \leq 0.3803 \\ 43.8 \leq V_{1T2R} \leq 19105 \\ 4807.6 \leq S \leq 15045 \\ 1.574 \leq \bar{F}_{max} \leq 2.4296 \\ 0.0034 \leq \hat{\vartheta} \leq 0.0055 \end{array} \right.$$

Wang et al. [19] used Particle Swarm Optimization (PSO) with a population size of 100 and 100 generations. To make a fair comparison, the same number of maximum function evaluations is used in this study.

The resulting average rank diagrams for IGD and HV, after 30 independent runs of each algorithm, are shown in Figures 31 and 32. On IGD, the three archive variants occupy the top three ranks (MO-BWR-archive: 2.3, MO-BMWR-archive: 3.0, MO-BMR-archive: 3.3), significantly outperforming all four established algorithms. CMOPSO, NSGA-3, and MOPSO-CD rank at the bottom (ranks 14, 15.0, and 16 respectively), confirming that their solutions cover the reference front poorly despite other strengths. On HV, the ranking reverses: CMOPSO (1.6) and NSGA-3 (1.7) lead, with archive variants in the middle tier (MO-BMR-archive: 7.5).

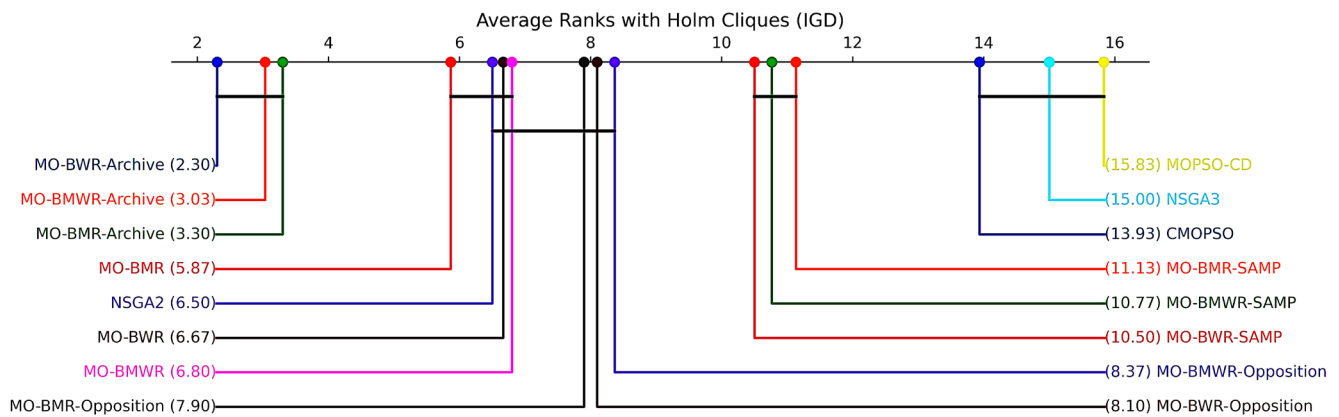


Figure 31. IGD average ranks with Holm cliques for optimization of parallel mechanism.

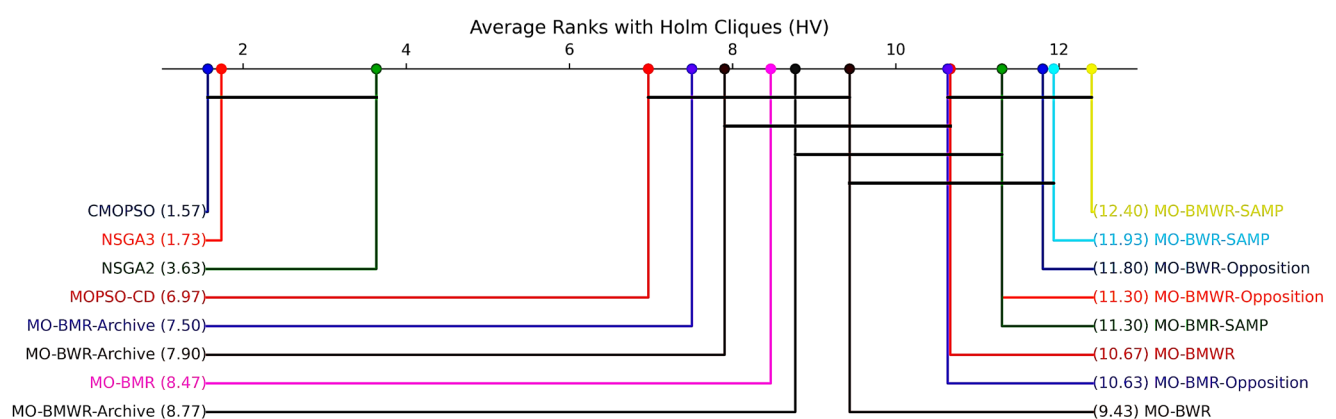


Figure 32. HV average ranks with Holm cliques for optimization of parallel mechanism.

This divergence is explained by the complementary metrics. MOPSO-CD achieves the best GD (rank 1.0), indicating its individual solutions lie closest to the reference front, yet its Pareto front (Figure 33) reveals extreme clustering in a narrow region—consistent with its poor IGD despite strong convergence. NSGA-3's reference-direction approach produces solutions aligned to predefined directions, which inflates HV through extreme boundary coverage, but results in the worst SPC (mean of 0.3721, rank 16.0) due to non-uniform spacing. In contrast, the archive variants maintain spacing quality statistically equivalent to CMOPSO and NSGA-2, while their IGD superiority confirms that they achieve the best overall combination of proximity and coverage across the entire reference front. The divergence between IGD and HV in this six-objective problem is consistent with the known sensitivity of HV to extreme boundary solutions in many-objective settings [26]. Figures 33–36 show how the archive variant improves the base MO-BWR algorithms by maintaining a more Pareto-optimal solution across the objective space.

It is notable that the archive variants reclaim the top IGD ranks on this six-objective problem despite having dropped to mid-table on the five-objective trajectory problem. This suggests that the number of objectives alone does not determine whether the archive mechanism is beneficial—other problem characteristics such as the number of design variables (only three in this case versus seven for the trajectory problem) and the constraint structure likely play an important role. With only three design variables, the archive's capacity is not stressed even when the Pareto surface spans six dimensions, allowing it to maintain good coverage.

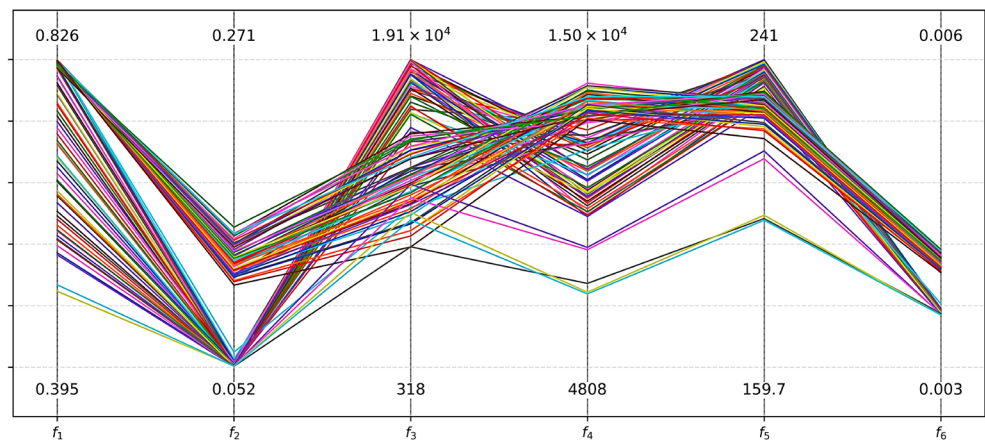


Figure 33. RPM Pareto front: CMOPSO.

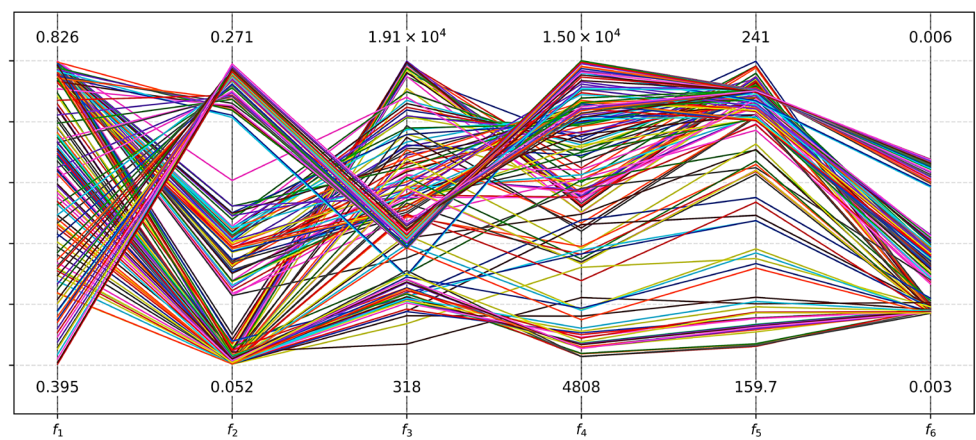


Figure 34. RPM Pareto front: MO-BWR-archive.

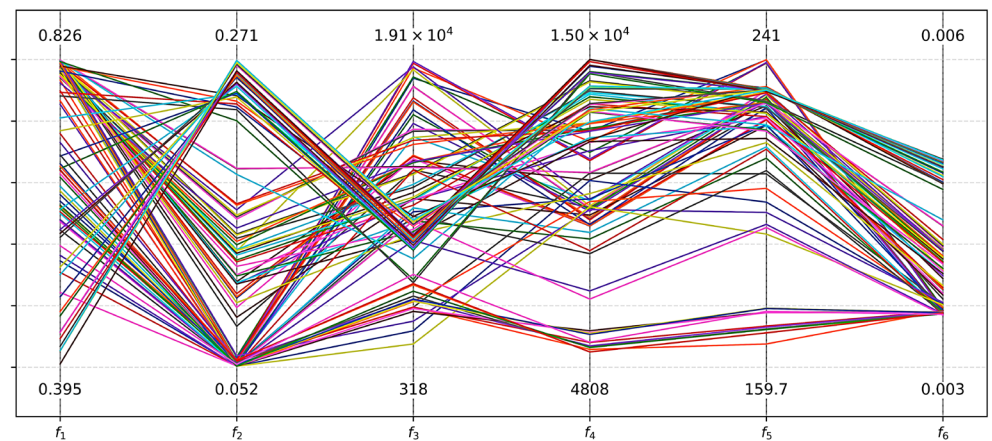


Figure 35. RPM Pareto front: MO-BWR.

The compact metrics table confirms the IGD-HV divergence in the CD diagrams. MO-BWR-archive achieves the best IGD (0.0736), but its HV (0.6866) is below CMOPSO (0.7834) and NSGA-3 (0.7817). The A_{12} values in Table 20 reflect this split, where MO-BWR-archive has $A_{12} = 0.999$ – 1.000 on IGD against CMOPSO, NSGA-3, and MOPSO-CD, while HV A_{12} is only 0.020 vs. CMOPSO and 0.019 vs. NSGA-3, indicating HV's superiority over those established methods. Against NSGA-2, the comparison is more balanced ($A_{12} = 0.782$ IGD, 0.152 HV). The archive-to-base comparison is significant ($A_{12} = 0.886$ on IGD), confirming

that the archive mechanism still improves coverage in this many-objective setting, even without HV dominance.

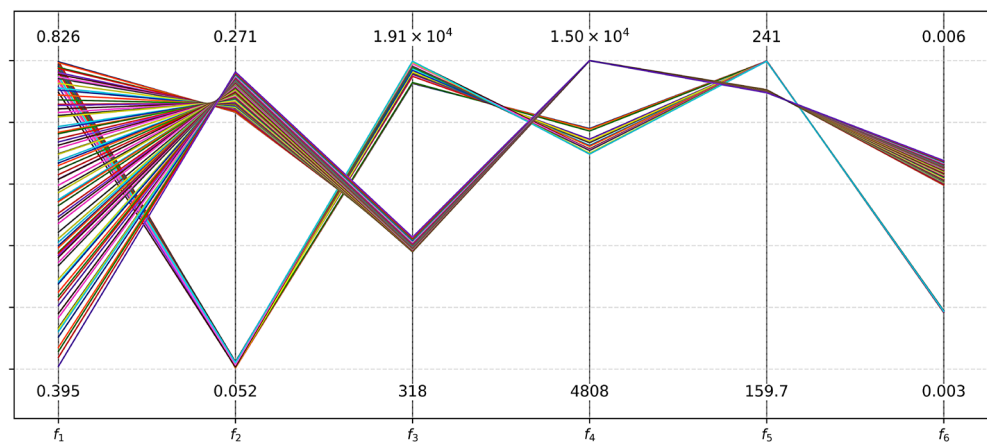


Figure 36. RPM Pareto front: MOPSO-CD.

Table 20. Pairwise A_{12} effect size comparison for MO-BWR-archive in reconfigurable parallel mechanism optimization.

Comparison	A_{12} (IGD)	A_{12} (HV)
MO-BWR-Archive vs. NSGA-2	0.782	0.152
MO-BWR-Archive vs. NSGA-3	1.000	0.019
MO-BWR-Archive vs. CMOPSO	0.999	0.020
MO-BWR-Archive vs. MOPSO-CD	1.000	0.372
MO-BWR-Archive vs. MO-BWR	0.886	0.587

Table 21 shows the performance of IGD and HV for reconfigurable parallel mechanism optimization.

Table 21. IGD and HV performance for reconfigurable parallel mechanism optimization.

Algorithm	IGD (Mean)	HV (Mean)	GD (Mean)	SPC (Mean)
MO-BWR-Archive	0.0736 (0.0089)	0.6866 (0.0628)	0.0538 (0.0110)	0.0707 (0.0096)
MO-BMWR-Archive	0.0755 (0.0066)	0.6766 (0.0619)	0.0520 (0.0111)	0.0717 (0.0081)
MO-BMR-Archive	0.0759 (0.0072)	0.6967 (0.0497)	0.0500 (0.0100)	0.0718 (0.0088)
MO-BMR	0.0852 (0.0076)	0.6829 (0.0576)	0.0539 (0.0111)	0.0901 (0.0134)
MO-BMWR	0.0906 (0.0117)	0.6466 (0.0777)	0.0598 (0.0154)	0.0892 (0.0114)
MO-BWR	0.0906 (0.0143)	0.6584 (0.0888)	0.0609 (0.0151)	0.0915 (0.0141)
MO-BMR-Opposition	0.0966 (0.0090)	0.6400 (0.0647)	0.0639 (0.0120)	0.0965 (0.0260)
MO-BMWR-Opposition	0.0974 (0.0117)	0.6287 (0.0735)	0.0666 (0.0135)	0.0967 (0.0152)
MO-BWR-Opposition	0.0994 (0.0196)	0.6217 (0.0779)	0.0684 (0.0143)	0.0949 (0.0180)
MO-BWR-SAMP	0.1097 (0.0130)	0.6105 (0.0861)	0.0561 (0.0181)	0.1287 (0.0189)
MO-BMWR-SAMP	0.1100 (0.0123)	0.6054 (0.0715)	0.0536 (0.0140)	0.1334 (0.0279)
MO-BMR-SAMP	0.1105 (0.0082)	0.6238 (0.0824)	0.0554 (0.0144)	0.1334 (0.0233)
NSGA2	0.1300 (0.0822)	0.7578 (0.0138)	0.0392 (0.0063)	0.0774 (0.0123)
CMOPSO	0.4163 (0.0930)	0.7834 (0.0123)	0.0271 (0.0062)	0.0700 (0.0246)
NSGA3	0.5787 (0.1102)	0.7817 (0.0099)	0.0179 (0.0059)	0.3721 (0.1692)
MOPSO-CD	0.6691 (0.0683)	0.7210 (0.0155)	0.0061 (0.0004)	0.0204 (0.0419)

Table 22 presents representative solutions from the Pareto front alongside the single optimum reported by Wang et al. [19]. One solution from MO-BWR-archive dominates Wang et al.'s result on all six objectives, achieving improvements of 3.3% in mean virtual

power, 10.1% in virtual power variance, and 13.2% in cutting workspace volume. Beyond this direct comparison, the table illustrates the inherent advantage of a multi-objective approach over weighted-sum optimization: the Pareto front spans workspace volumes from 2873 to 19,087 (a range Wang et al.'s single-point method cannot reveal), and shows that maximizing grinding workspace area (S) requires a fundamentally different design configuration ($\lambda_B > 1$) than maximizing load capacity or cutting workspace. These trade-offs provide the designer with actionable choices that a single weighted-sum solution cannot offer.

Table 22. Representative Pareto-optimal solutions of MO-BWR-archive and comparison with Wang et al. [19].

Solution	b	λ_B	λ_L	$\bar{\eta}_W \uparrow$	$\hat{\eta}_W \downarrow$	$V_{172R} \uparrow$	$S \uparrow$	$\bar{F}_{max} \uparrow$	$\hat{\phi} \downarrow$
Wang et al. [19]—PSO									
Single optimum	116.04	0.8921	1.2648	0.7974	0.1508	12,197.7	13,034.2	228.73	0.0041
MO-BWR-Archive (this study)									
Dominating [†]	117.25	0.8768	1.2562	0.8240	0.1355	13,810.1	13,216.5	231.68	0.0041
Best $\bar{\eta}_W$	118.27	0.9103	1.1132	0.8254	0.1849	16,858.3	10,398.8	230.77	0.0043
Best $\hat{\eta}_W$	93.83	0.7942	1.2337	0.7218	0.0526	2872.8	8096.6	188.13	0.0036
Best V_{172R}	118.35	0.7875	1.1190	0.7043	0.0584	19,087.0	10,506.2	236.67	0.0035
Best S	120.00	1.0056	1.3000	0.3966	0.2630	8353.6	15,042.5	232.48	0.0050
Best $\bar{F}_{max}$	120.00	0.7919	1.1965	0.8139	0.0548	18,328.0	12,372.1	240.83	0.0036
Best $\hat{\phi}$	92.95	0.7807	1.1000	0.4972	0.0528	7554.1	6332.5	185.58	0.0035

[†] This solution dominates Wang et al.'s optimum on all six objectives simultaneously. $\uparrow$ = maximize, $\downarrow$ = minimize.

4. Discussion and Conclusions

Five robotics optimization problems, spanning two to six objectives, were solved using 16 algorithms: nine proposed MO-BxR variants (three base algorithms $\times$ three variant strategies), four established algorithms (NSGA-2, NSGA-3, CMOPSO, MOPSO-CD), and the three unmodified base MO-BxR algorithms. Table 23 summarizes the top-ranked algorithm on each primary metric for every case study.

Table 23. Cross-case summary of top-ranked algorithms on IGD and HV. Ranks are mean Friedman ranks over 30 independent runs. Best values in bold.

Case	Problem	#Obj	IGD Best Algorithm	IGD Rank	HV Best Algorithm	HV Rank
3.1	AUV Shape	2	MO-BWR-Archive	1.3	MO-BWR-Archive	1.8
3.2	Inspection Robot	3	MO-BMR-Archive	1.3	MO-BMR-Archive	1.5
3.3	Inverse Kinematics	3	MO-BMWR-Archive	2.6	MO-BMWR-Archive	3.1
3.4	Trajectory Planning	5	MO-BMR-SAMP	3.5	MO-BWR-SAMP	3.1
3.5	Parallel Mechanism	6	MO-BWR-Archive	2.3	CMOPSO	1.6

The present work contributes to the field of robotics optimization by systematically developing and evaluating three enhancement strategies—archive, opposition, and self-adaptive multi-population (SAMP)—within the recently proposed MO-BxR framework. Although these mechanisms individually originate from established concepts in multi-objective optimization, their integration and comparative investigation within reduced-parameter MO-BxR algorithms for robotics optimization problems have not previously been explored. Another important contribution of the study is the detailed analysis of how these enhancement strategies influence convergence, diversity preservation, robustness, and stability across robotics problems involving different objective dimensionalities, constraint structures, and surrogate modeling characteristics. Furthermore, the work pro-

vides a rigorous comparative assessment involving 16 algorithms, multiple complementary performance metrics, and statistical analyses using Friedman ranking, Conover post hoc testing with Holm correction, and Vargha–Delaney effect sizes.

Three patterns emerge from the results:

Archive variants as the most consistent improvement mechanism. In four of the five case studies (AUV, inspection robot, inverse kinematics, and parallel mechanism), archive variants occupy the top IGD rank, with A_{12} effect sizes exceeding 0.95 against all established algorithms in most pairwise comparisons. On three of these (AUV, inspection robot, inverse kinematics), archive variants also lead on HV. The external archive doubles the effective storage of non-dominated solutions and applies a second round of non-dominated sorting each iteration, which translates into denser and more uniformly distributed Pareto approximations. This benefit holds across a range of objective dimensionalities—from the two-objective AUV problem to the six-objective parallel mechanism—suggesting that the archive mechanism’s advantage is not simply a function of the number of objectives.

Base BxR algorithms as strong competitors. A notable finding is the consistently strong performance of the unmodified base BxR algorithms. On the five-objective trajectory planning problem—the only case where archive variants drop to mid-table (IGD ranks 8–9)—the base variants rank among the top algorithms alongside the SAMP variants. MO-BMR-SAMP achieves the best mean IGD on this problem, but its advantage over MO-BMR is not statistically significant ($A_{12} = 0.604$ on IGD, 0.397 on HV), indicating that the SAMP mechanism provides at best a marginal gain beyond what the base algorithm already delivers. Across all five problems, every base BxR variant outperforms at least two of the four established algorithms on IGD, confirming that the algorithm-specific parameter-free update rules are inherently competitive even without the proposed enhancements.

The IGD–HV divergence in many-objective settings. On the six-objective parallel mechanism problem, BxR archive variants rank first on IGD but CMOPSO and NSGA-3 lead on HV. The complementary metrics resolve this apparent contradiction: MOPSO-CD achieves the best GD (rank 1.0) yet the worst coverage of the objective space, indicating that its solutions converge closely to the reference front but cluster in a narrow region. NSGA-3’s reference-direction mechanism inflates HV through extreme boundary coverage while producing the worst SPC (rank 16.0). This behavior is consistent with the known sensitivity of HV to boundary solutions in many-objective settings (Ishibuchi et al. [27]). When both convergence and coverage matter, IGD provides a more balanced assessment, and on this metric the archive variants remain superior. That said, the strong HV performance of CMOPSO and NSGA-3 reflects a genuine strength in convergence that should not be dismissed—practitioners whose applications prioritize individual solution quality over full-front coverage may find these algorithms more suitable.

Opposition variant assessment. The opposition-based variants do not rank first on any primary metric in any case study, and they generally rank below their corresponding base algorithms on IGD. The magnitude of this degradation varies across problems—on the parallel mechanism problem the differences are modest and often not statistically significant, while on the trajectory planning and inverse kinematics problems the opposition variants rank substantially lower than the base algorithms. The simple opposition mechanism generates solutions by reflecting the current population about the midpoint of the search space, which may not produce useful candidate solutions when the feasible or Pareto-optimal regions occupy a small fraction of the design space. Despite the inconsistent results, the opposition variants do not cause catastrophic performance collapse in any case, and the underlying idea of leveraging opposite solutions to improve diversity remains sound. Future work could explore quasi-opposition Rao and Pawar [2] or generalized opposition strategies that may provide a more targeted and reliable benefit.

Across all five case studies, no established algorithm achieves IGD rank 1 in any case. On HV, established algorithms lead only in the six-objective parallel mechanism problem. The strongest established performer varies by problem: CMOPSO is strongest on the inspection robot and parallel mechanism, NSGA2 on the trajectory planning problem, and NSGA3 on the parallel mechanism HV. Despite their widespread use in the robotics optimization literature, all four established algorithms are consistently outranked on IGD by the proposed variants across problems of varying dimensionality, constraint structure, and surrogate model type. Importantly, the proposed algorithms achieve this without requiring any problem-specific parameter tuning, which is a practical advantage for engineering applications.

The principal findings are:

- The archive mechanism provides the most consistent improvement, achieving IGD rank 1 in four of five cases and HV rank 1 in three of five cases. Its advantage spans both lower-dimensional (2–3 objectives) and higher-dimensional (six objectives) problems.
- The base BxR algorithms are inherently strong multi-objective optimizers. Even without the proposed enhancements, they consistently outperform established algorithms on IGD, confirming the practical value of the algorithm-specific parameter-free update rules.
- The SAMP mechanism shows marginally improved performance on the five-objective trajectory planning problem, but its advantage over the base variants is not statistically significant and it does not improve performance on other problems. The mechanism needs deeper investigation to identify under what problem characteristics it provides meaningful benefit.
- The opposition mechanism does not provide reliable improvement across any case study and generally degrades performance relative to the base algorithms, though the degradation is inconsistent in magnitude. Refined opposition strategies warrant further study.
- The divergence between IGD and HV rankings on the six-objective problem underscores the importance of using multiple complementary metrics and understanding their theoretical properties when evaluating multi-objective algorithms.

Future work may include a detailed sensitivity analysis of important algorithmic design choices, particularly the archive size and SAMP sub-population adaptation strategy, in order to further evaluate the robustness, scalability, and computational trade-offs of the proposed framework. In addition, alternative adaptation mechanisms and different archive management policies may be investigated and compared across a wider range of many-objective robotics problems to better understand their influence on convergence and diversity characteristics of the obtained Pareto-optimal solution sets. In addition, the proposed variants may be further evaluated on established benchmark suites such as DTLZ, WFG, and MaF to assess their broader generalization capability under standardized optimization environments.

Author Contributions: Conceptualization, R.V.R.; methodology, R.V.R.; validation, H.M.V.; formal analysis and writing—original draft preparation, H.M.V.; writing—review and editing, J.P.D. and R.V.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding to meet the APC.

Data Availability Statement: All original contributions from this study are contained within the article; further inquiries may be addressed to the corresponding author. The proposed algorithms and the statistical tests used in the study are compiled into a Python library available at <https://github.com/hari729/loares>, built on pymoo 0.6.1.6 (accessed on 25 March 2026). The library provides all proposed MO-BxR variants and can be used to replicate the experimental framework.

Acknowledgments: Support from the Anusandhan National Research Foundation of the Government of India (MTR/2023/000071) is gratefully acknowledged by the first author. However, there is no provision to pay the APC in the project. All authors have consented to the acknowledgement. Certain figures and interpretations were prepared with the assistance of ChatGPT-5.3. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

This section contains the average rank diagrams with Holm cliques for supporting metrics in each case study.

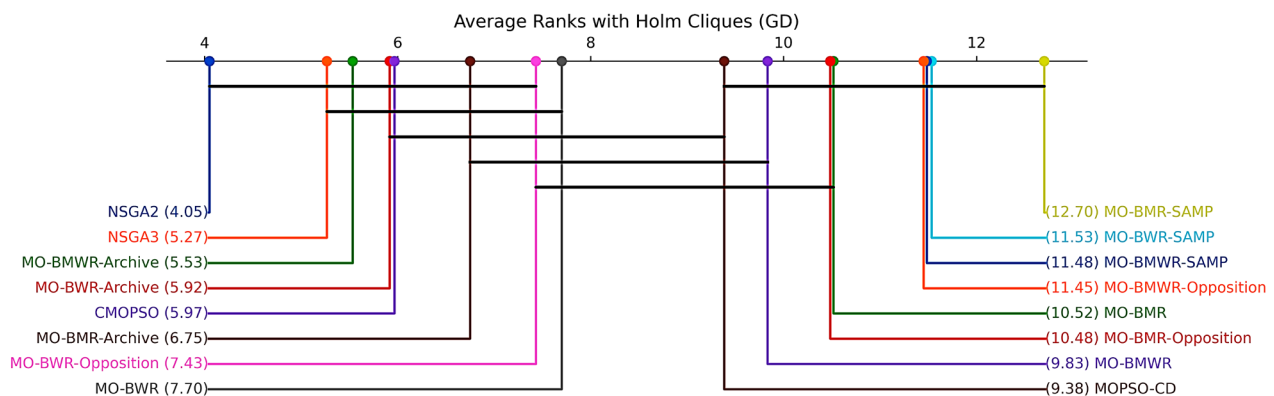


Figure A1. Shape optimization of AUV: GD rank cliques.

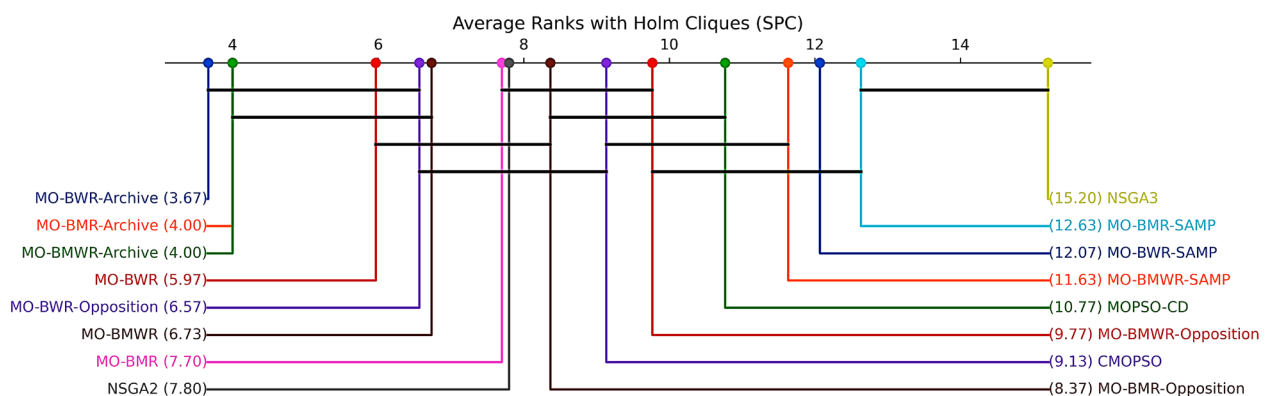


Figure A2. Shape optimization of AUV: SPC rank cliques.

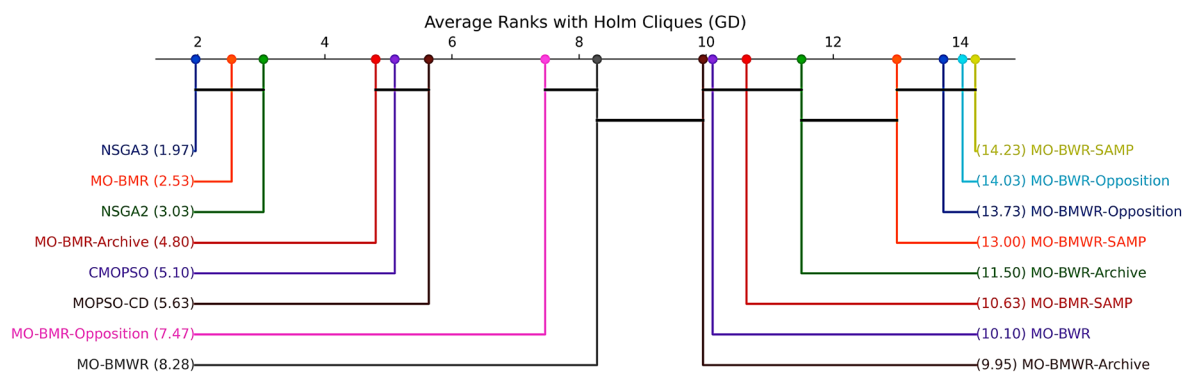


Figure A3. Structural optimization of inspection robot: GD rank cliques.

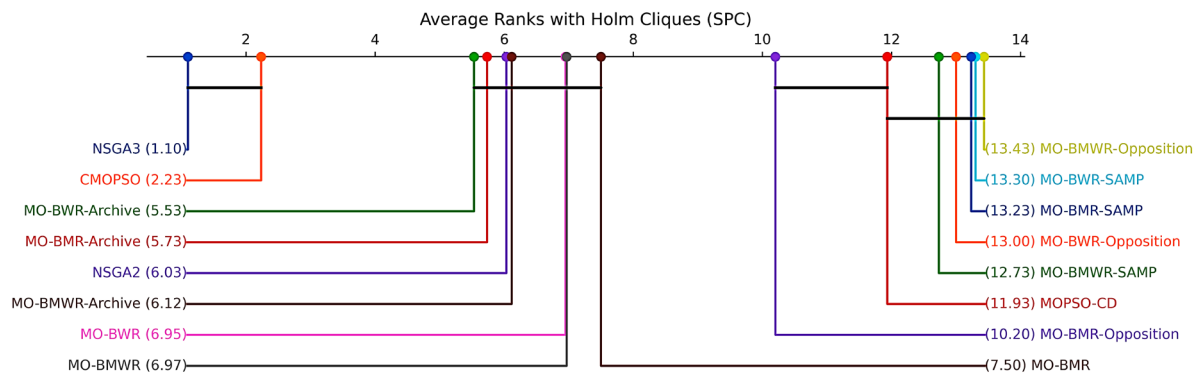


Figure A4. Structural optimization of inspection robot: SPC rank cliques.

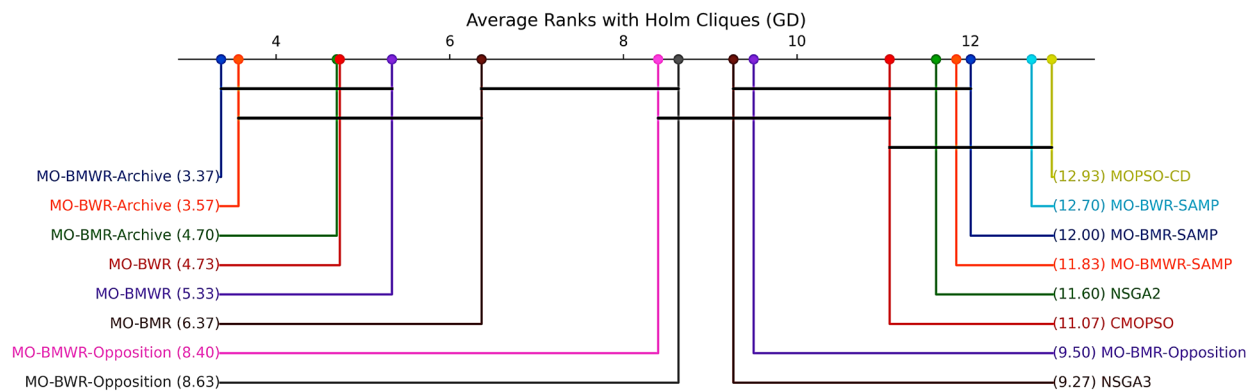


Figure A5. Inverse kinematics solution: GD rank cliques.

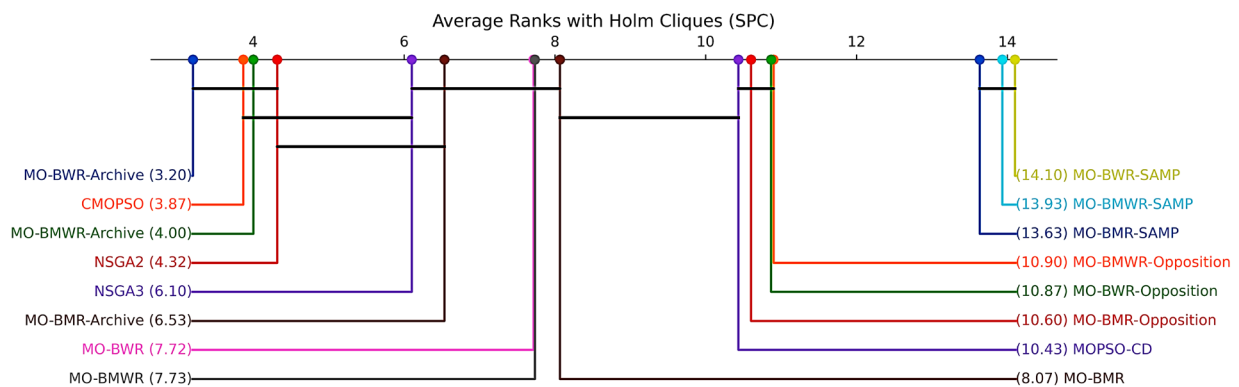


Figure A6. Inverse kinematics solution: SPC rank cliques.

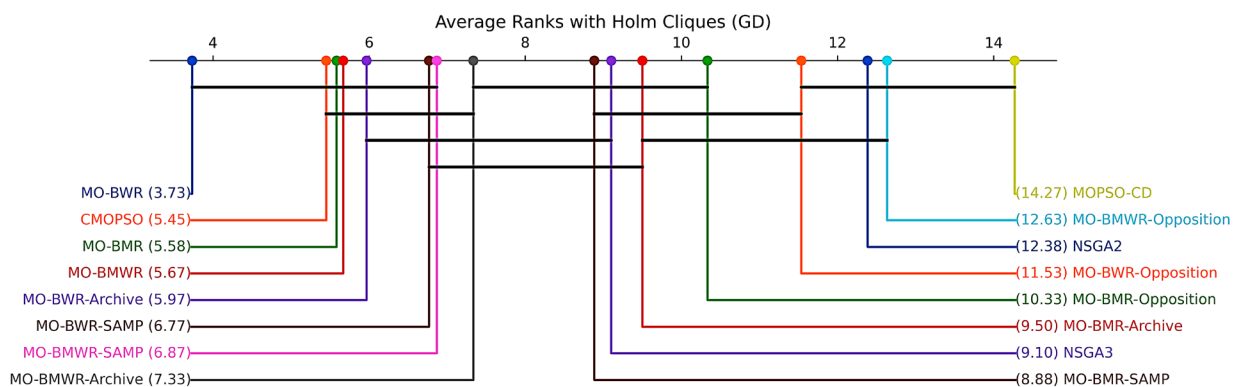


Figure A7. Wall-building robot trajectory optimization: GD rank cliques.

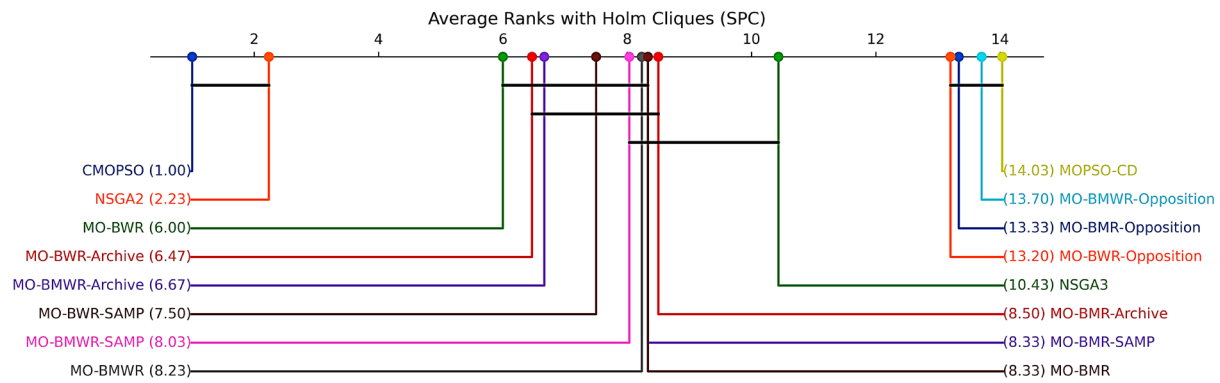


Figure A8. Wall-building robot trajectory optimization: SPC rank cliques.

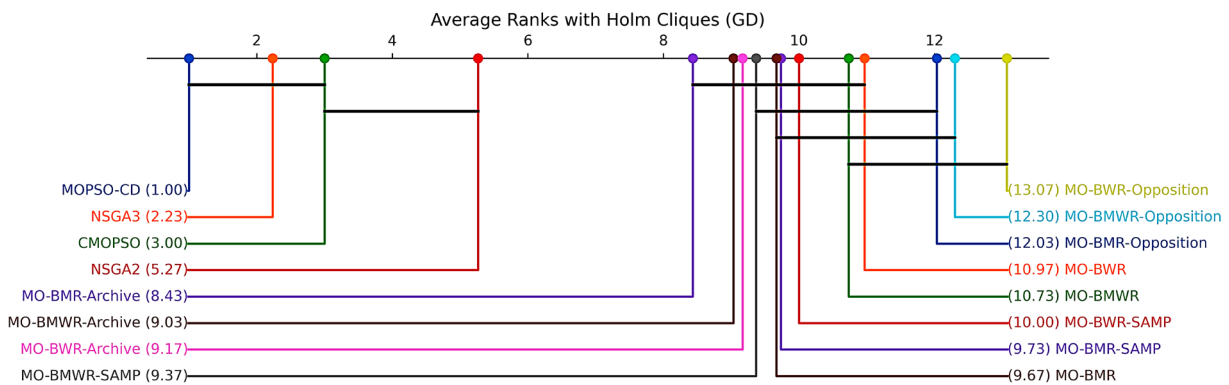


Figure A9. Reconfigurable parallel mechanism optimization: GD rank cliques.

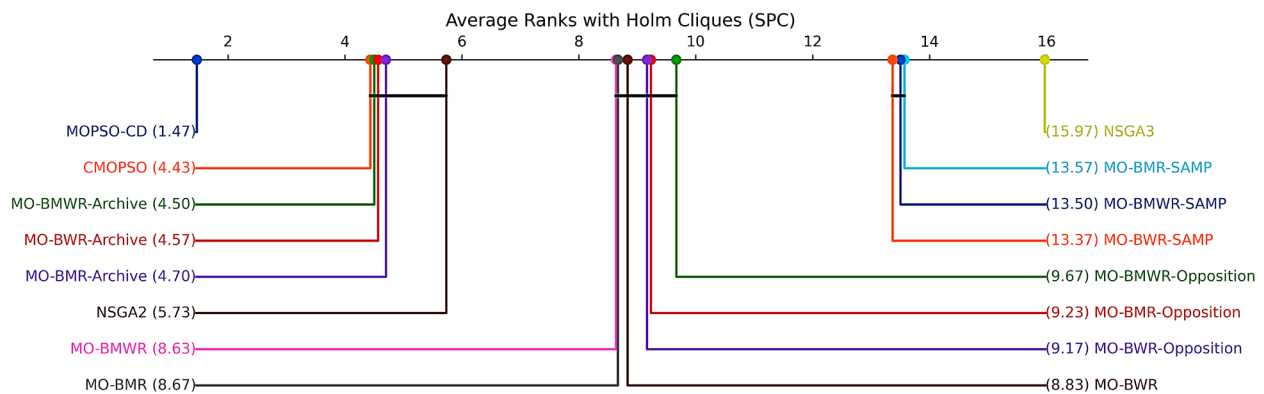


Figure A10. Reconfigurable parallel mechanism optimization: SPC rank cliques.

References

1. Rao, R.V.; Davim, J.P. Optimization of Different Metal Casting Processes Using Three Simple and Efficient Advanced Algorithms. *Metals* **2025**, *15*, 1057. [\[CrossRef\]](#)
2. Rao, R.V.; Pawar, R.B. Design Optimization of Cam-Follower Mechanisms Using Rao Algorithms and Their Variants. *Evol. Intell.* **2024**, *17*, 745–770. [\[CrossRef\]](#)
3. Tang, Q.; Li, Y.; Deng, Z.; Chen, D.; Guo, R.; Huang, H. Optimal Shape Design of an Autonomous Underwater Vehicle Based on Multi-Objective Particle Swarm Optimization. *Nat. Comput.* **2020**, *19*, 733–742. [\[CrossRef\]](#)
4. Nabavi, S.N.; Shariatee, M.; Enferadi, J.; Akbarzadeh, A. Parametric Design and Multi-Objective Optimization of a General 6-PUS Parallel Manipulator. *Mech. Mach. Theory* **2020**, *152*, 103913. [\[CrossRef\]](#)
5. Rokbani, N.; Neji, B.; Slim, M.; Mirjalili, S.; Ghandour, R. A Multi-Objective Modified PSO for Inverse Kinematics of a 5-DOF Robotic Arm. *Appl. Sci.* **2022**, *12*, 7091. [\[CrossRef\]](#)
6. Zhao, Z.; Tao, Y.; Wang, J.; Hu, J. The Multi-Objective Optimization Design for the Magnetic Adsorption Unit of Wall-Climbing Robot. *J. Mech. Sci. Technol.* **2022**, *36*, 305–316. [\[CrossRef\]](#)

7. Shi, Q.; Wang, Z.; Ke, X.; Zheng, Z.; Zhou, Z.; Wang, Z.; Fan, Y.; Lei, B.; Wu, P. Trajectory Optimization of Wall-Building Robots Using Response Surface and Non-Dominated Sorting Genetic Algorithm III. *Autom. Constr.* **2023**, *155*, 105035. [\[CrossRef\]](#)
8. Liu, J.; Huang, H.; Fan, Q.; Ma, C. A Multi-Strategy Enhanced Moth-Flame Optimization Algorithm for Complex Inverse Kinematics Problems in Series Robots. *Clust. Comput.* **2025**, *28*, 290. [\[CrossRef\]](#)
9. Yoon, J.; Lim, K.; Park, S.; Doh, J. Multi-Objective Optimization of Aerodynamic Blade Shapes for Quadcopter System to Enhance Hovering Thrust and Power Consumption Efficiency. *Int. J. Aeronaut. Space Sci.* **2023**, *24*, 689–700. [\[CrossRef\]](#)
10. Jin, P.; Qin, X.; Lei, J.; Wang, Y.; Jia, W.; Zhang, S. Design and Stability Performance Optimization of a Novel Hybrid Inspection Robot Walking Device for Smart Grid Applications. *Alex. Eng. J.* **2024**, *109*, 970–988. [\[CrossRef\]](#)
11. Duan, P.; Yu, Z.; Gao, K.; Meng, L.; Han, Y.; Ye, F. Solving the Multi-Objective Path Planning Problem for Mobile Robot Using an Improved NSGA-II Algorithm. *Swarm Evol. Comput.* **2024**, *87*, 101576. [\[CrossRef\]](#)
12. Yue, P.; Xu, B.; Zhang, M. An Improve Nonlinear Robust Control Approach for Robotic Manipulators with PSO-Based Global Optimization Strategy. *Sci. Rep.* **2024**, *14*, 21447. [\[CrossRef\]](#)
13. Yang, P.; Sun, L.; Zhang, M.; Chen, H. A Lightweight Optimal Design Method for Magnetic Adhesion Module of Wall-Climbing Robot Based on Surrogate Model and DBO Algorithm. *J. Mech. Sci. Technol.* **2024**, *38*, 2041–2053. [\[CrossRef\]](#)
14. Zeid, A.; Al-Yahmedi, A.; Zaier, R.; Bahadur, I. Multi-Objective Design Optimization of an in-Pipe Inspection Robot. *Frankl. Open* **2024**, *6*, 100071. [\[CrossRef\]](#)
15. Song, Q.; Lu, C.; Song, Q.; Jiang, H.; Liu, B. Research on Extreme Obstacle-Crossing Performance and Multi-Objective Optimization of Tracked Mobile Robot. *Robot. Auton. Syst.* **2024**, *180*, 104759. [\[CrossRef\]](#)
16. Cui, J.; Wu, L.; Huang, X.; Xu, D.; Liu, C.; Xiao, W. Multi-Strategy Adaptable Ant Colony Optimization Algorithm and Its Application in Robot Path Planning. *Knowl.-Based Syst.* **2024**, *288*, 111459. [\[CrossRef\]](#)
17. Duan, G.; Yao, L.; Zhan, Z.; Kang, T.; Guo, C. Multi-Objective Optimal Control Method for the 6-DOF Robotic Crusher. *Appl. Sci.* **2024**, *14*, 9397. [\[CrossRef\]](#)
18. Wang, Y.; Qin, X.; Jia, W.; Lei, J.; Wang, D.; Feng, T.; Zeng, Y.; Song, J. Multiobjective Energy Consumption Optimization of a Flying-Walking Power Transmission Line Inspection Robot During Flight Missions Using Improved NSGA-II. *Appl. Sci.* **2024**, *14*, 1637. [\[CrossRef\]](#)
19. Wang, H.; Qi, Y.; Tao, X.; Yu, S. Multi-Objective Optimization Design of Reconfigurable Parallel Mechanism for Integrated Machining of Heavy-Duty Components. *J. Braz. Soc. Mech. Sci. Eng.* **2025**, *47*, 577. [\[CrossRef\]](#)
20. Gupta, P.; Pratihar, D.K.; Deb, K. Dynamic Performance Evaluation of Evolutionary Multi-Objective Optimization Algorithms for Gait Cycle Optimization of a 25-DOFs NAO Humanoid Robot. *Swarm Evol. Comput.* **2025**, *99*, 102144. [\[CrossRef\]](#)
21. Wang, L.; Zhang, F.; Yan, M.; Liu, Y. Dynamic Modeling and Optimization Method for Large Amplitude Ultrasonic Osteotome for Surgical Robots. *Appl. Acoust.* **2025**, *235*, 110694. [\[CrossRef\]](#)
22. Zhang, Z.; Zhao, W.; Bian, X.; Zhao, H. A Collaborative Swarm-Differential Evolution Algorithm for Multi-Objective Multi-Robot Task Assignment. *Appl. Sci.* **2025**, *15*, 10627. [\[CrossRef\]](#)
23. You, D.; Yu, J.; Jia, Z.; Zhang, Y.; Yang, Z. Mobile Robot Path Planning Based on Fused Multi-Strategy White Shark Optimisation Algorithm. *Appl. Sci.* **2025**, *15*, 8453. [\[CrossRef\]](#)
24. Xing, H.; Liu, Y.; Chen, J.; Li, W.; Ding, L.; Tavakoli, M. Variable Admittance Control for Door Opening with a Wheeled Mobile Manipulator Considering Ground Obstacles. *Intell. Serv. Robot.* **2026**, *19*, 41. [\[CrossRef\]](#)
25. Zitzler, E.; Thiele, L.; Laumanns, M.; Fonseca, C.M.; Fonseca, V.G. da Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Trans. Evol. Comput.* **2003**, *7*, 117–132. [\[CrossRef\]](#)
26. Li, M.; Yao, X. Quality Evaluation of Solution Sets in Multiobjective Optimisation: A Survey. *ACM Comput. Surv.* **2019**, *52*, 39. [\[CrossRef\]](#)
27. Ishibuchi, H.; Imada, R.; Masuyama, N.; Nojima, Y. Comparison of HV, IGD and IGD+ from the Viewpoint of Optimal Distributions of Solutions. In *Evolutionary Multi-Criterion Optimization*; Deb, K., Goodman, E., Coello Coello, C.A., Klamroth, K., Miettinen, K., Mostaghim, S., Reed, P., Eds.; Springer International Publishing: Cham, Switzerland, 2019; Volume 11411, pp. 332–345, ISBN 978-3-030-12597-4, 978-3-030-12598-1.
28. Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
29. García, S.; Fernández, A.; Luengo, J.; Herrera, F. Advanced Nonparametric Tests for Multiple Comparisons in the Design of Experiments in Computational Intelligence and Data Mining: Experimental Analysis of Power. *Inf. Sci.* **2010**, *180*, 2044–2064. [\[CrossRef\]](#)
30. Vargha, A.; Delaney, H.D. A Critique and Improvement of the CL Common Language Effect Size Statistics of McGraw and Wong. *J. Educ. Behav. Stat.* **2000**, *25*, 101–132. [\[CrossRef\]](#)
31. Deb, K.; Jain, H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEE Trans. Evol. Comput.* **2014**, *18*, 577–601. [\[CrossRef\]](#)
32. Xue, Y.; Li, M.; Liu, X. An Effective and Efficient Evolutionary Algorithm for Many-Objective Optimization. *Complex Intell. Syst.* **2023**, *9*, 487–509. [\[CrossRef\]](#)

33. Salazar, J.Z.; Hadka, D.; Reed, P.; Seida, H.; Deb, K. Diagnostic Benchmarking of Many-Objective Evolutionary Algorithms for Real-World Problems. *Eng. Optim.* **2025**, *57*, 287–308. [[CrossRef](#)]
34. Schütze, O.; Hernández Castellanos, C.I. *Archiving Strategies for Evolutionary Multi-Objective Optimization Algorithms*; Springer: Cham, Switzerland, 2021.
35. Lakbichi, R.; Naimi, M.; Raghay, S.; Tairi, H. Opposition-Based Learning Techniques in Metaheuristics: Classification, Comparison, and Convergence Analysis. *Swarm Evol. Comput.* **2023**, *73*, 101112. [[CrossRef](#)] [[PubMed](#)]
36. Lakbichi, R.; Naimi, M.; Raghay, S.; Tairi, H. Opposition-Based Learning: A Literature Review. *Expert Syst. Appl.* **2020**, *163*, 113765.
37. Ishibuchi, H.; Imada, R.; Setoguchi, Y.; Nojima, Y. Difficulties in Fair Performance Comparison of Multi-Objective Evolutionary Algorithms. *IEEE Comput. Intell. Mag.* **2022**, *17*, 86–101. [[CrossRef](#)]
38. Blank, J.; Deb, K. Pymoo: Multi-Objective Optimization in Python. *IEEE Access* **2020**, *8*, 89497–89509. [[CrossRef](#)]
39. Rao, R.V.; Taler, J.; Taler, D.; Lakshmi, J. A Unified Optimization Approach for Heat Transfer Systems Using the BxR and MO-BxR Algorithms. *Energies* **2025**, *19*, 34. [[CrossRef](#)]
40. Carpentier, J.; Saurel, G.; Buondonno, G.; Mirabel, J.; Lamiriaux, F.; Stasse, O.; Mansard, N. The Pinocchio C++ Library: A Fast and Flexible Implementation of Rigid Body Dynamics Algorithms and Their Analytical Derivatives. In Proceedings of the 2019 IEEE/SICE International Symposium on System Integration (SII), Paris, France, 14–16 January 2019; pp. 614–619.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.