Journal of Mathematical Modeling
Vol. -, No. -, 2026, pp. -. Research Article

JMM

Robotic optimization with high-dimensional Pareto front visualization

Artem Maminov[†]

[†]*Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences (FRC CSC RAS), Moscow, Russia*

Email(s): amaminov@frccsc.ru

Abstract. In this paper, we consider an approach for multi-criteria optimization of key design characteristics for robots. We use 5 criteria: Workspace Area, Space Utilization Index, Global Dexterity Index, Global Manipulability Index and Global Resistivity Index. The first two characterize workspace, while the latter three evaluate kinematic performance throughout the workspace. The Pareto-set visualization for such problem can be a challenging task, since the objective space is five-dimensional. We consider clustering approach for efficient reduction of the number of Pareto points. The calculation of the indexes is performed automatically using interval analysis techniques. The experimental validation was performed for three parallel manipulators: 2-RPR, DexTar, PRRRP. We compare the proposed approach with random sampling method and exact Pareto front, calculated with "brute force" algorithm.

Keywords: Robot workspace, multiobjective optimization, interval analysis, parallel robots, kinematic performance
AMS Subject Classification 2010: 90C29, 58E17

1 Introduction

Parallel robotic architectures [7, 10, 15] continue to attract significant research interest due to their inherent advantages in stiffness, precision, and dynamic performance. These characteristics make them indispensable for advanced manufacturing, medical applications, and space exploration. A fundamental challenge in robotic design lies in the simultaneous optimization of multiple performance criteria that often conflict with each other.

Our previous work [12] addressed the Pareto-optimal design of a 2-RPR parallel robot, simultaneously maximizing two key performance indices: Global Dexterity Index (GDI), evaluated rigorously using interval methods for solving the underlying non-linear systems, and Workspace Area (W), employing Genetic Algorithms (GAs) to effectively explore the trade-off frontier. While this approach provided

*Corresponding author
Received: 09 September 2025/ Revised: 05 December 2025/ Accepted: 10 April 2026
DOI: 10.22124/jmm.2026.31644.2849

valuable insights into the bi-objective design space, real-world robotic design necessitates considering a broader spectrum of often conflicting performance criteria to achieve truly optimal and robust solutions.

However, comprehensive robotic design requires consideration of additional critical characteristics: *Space Utilization Index* quantifying volumetric efficiency, *Global Manipulability Index* measuring motion transmission quality, and *Global Resistivity Index* evaluating disturbance rejection capabilities. The integration of these five criteria – Workspace Area (W), Space Utilization Index (SUI), Global Dexterity Index (GDI), Global Manipulability Index (GMI), and Global Resistivity Index (GRI) – provides a holistic assessment framework. The first two characterize workspace capabilities, while the latter three quantify kinematic performance throughout the workspace.

This multi-objective formulation generates a 5-dimensional Pareto frontier, presenting significant visualization challenges not encountered in lower-dimensional optimizations. Conventional 2D/3D visualization techniques become inadequate for interpreting trade-offs between five competing objectives. To address this complexity, we propose a clustering methodology for intelligently reducing Pareto solution density while preserving frontier topology characteristics.

Our approach leverages interval analysis techniques for automated computation of performance indices, enabling rigorous evaluation throughout the continuous workspace. Combined with evolutionary multi-objective optimization, this allows efficient exploration of the design space. Experimental validation covers three distinct parallel architectures: 2-RPR, DexTar, and PRRRP manipulators, demonstrating the universality of our framework.

2 Related works

Performance indices play a crucial role in quantitative evaluation of robotic manipulators. The GDI, introduced by Gosselin and Angeles [7], measures uniformity of velocity transmission across the workspace through the condition number of the Jacobian matrix. This index has become fundamental for kinematic optimization, as demonstrated in studies of planar and spatial manipulators [6, 8].

The SUI, particularly relevant for compact mechanisms, quantifies volumetric efficiency by comparing the robot's footprint to its workspace volume. The GRI evaluates force transmission capabilities, while the GMI assesses motion transmission quality through singular value decomposition of the Jacobian. These metrics have been jointly applied in 2-DOF mechanism optimization [16], though typically with analytical calculation methods requiring significant manual derivation.

Multi-objective optimization approaches for robotic systems have evolved substantially. Early works like Huang et al. [10] focused on single-objective formulations, while Gallant et al. [5] considered Pareto-optimal solutions for parallel mechanisms. Recent studies employ evolutionary algorithms [18] to handle competing objectives, particularly for 2-RPR [13, 16] and PRRRP [2, 11] architectures. However, these implementations often rely on pre-derived analytical expressions for performance indices, limiting their adaptability to new manipulator configurations.

The DexTar manipulator [16] exemplifies more complex architectures where analytical index computation becomes prohibitively difficult.

The main contribution of this work is automation calculation of 5 criteria through interval analysis with Pareto-set calculation with NSGA-II methods. We also propose a method for visualizing the 5D Pareto front through clustering in the parameter space and displaying clusters in the criteria space (parallel coordinates).

3 Problem statement

We consider a k-dimensional multi-criteria optimization problem where all objectives are maximized. The problem is formally defined as:

$$\max_{x \in X} \mathbf{f}(x), \tag{1}$$

where $\mathbf{f}(x) = [f_1(x), f_2(x), \ldots, f_k(x)]$ is a vector-valued objective function mapping $X \to \mathbb{R}^k$, n denotes the number of design variables, k represents the number of objectives, and $X \subseteq \mathbb{R}^n$ is the feasible region defined by constraints.

Since objectives typically conflict, no single solution optimizes all criteria simultaneously. We therefore seek the Pareto-optimal set [4] where solutions satisfy the non-dominance condition:

$$\mathbf{P} = \{x^* \in X \mid \nexists x \in X : \mathbf{f}(x) \succ \mathbf{f}(x^*)\}. \tag{2}$$

The dominance relation $\mathbf{f}(a) \succ \mathbf{f}(b)$ holds when:

$$\begin{cases} f_i(a) \geq f_i(b), & \forall i \in \{1, \ldots, k\}, \\ f_j(a) > f_j(b), & \exists j \in \{1, \ldots, k\}. \end{cases}$$

Sometimes additional methods can be used for finding best point across Pareto-set (VIKOR methods [17]) or ranging points (TOPSIS method [20]). To facilitate decision-making in high-dimensional spaces, we introduce the utopian point $\mathbf{up} = (up_1, \ldots, up_k)$ where $up_i = \sup_{x \in X} f_i(x)$. The compromise solution minimizes the Euclidean distance to this utopian reference:

$$x_{\text{ideal}} = \arg\min_{x \in \mathbf{P}} \|\mathbf{f}(x) - \mathbf{up}\|_2. \tag{3}$$

Given heterogeneous measurement scales across objectives, we apply min-max normalization:

$$f_i^{\text{norm}}(x) = \frac{f_i(x) - f_i^{\min}}{f_i^{\max} - f_i^{\min}}, \quad f_i^{\min} = \min_{x \in \mathbf{P}} f_i(x), \quad f_i^{\max} = \max_{x \in \mathbf{P}} f_i(x). \tag{4}$$

4 Criteria calculation

This section details the five objective functions used for robotic manipulator optimization: W, SUI, GDI, GMI, and GRI. The first two characterize workspace area capabilities, while the latter three evaluate kinematic performance throughout the workspace.

The kinematic performance indexes usually defined as follows:

$$Index = \frac{A}{B},$$

where $A = \iint_v \alpha dv$, and B is the workspace area in input parameters. The integral can be approximated with uniform grid:

$$Index = \frac{\sum \alpha}{m^n}, \tag{5}$$

where α is the value, which reflects some characteristic of the robot. m is the number of the points in computational grid ($m = 30$ was used for experimental part) and n is the dimension of the problem ($n = 2$ in our case).

Let us also introduce $\sigma_{\min}, \sigma_{\max}$, where $\sigma_{\min}$ and $\sigma_{\max}$ is the smallest and greatest singular values of Jacobi matrix J respectively. Jacobi matrix depends on input and output parameters of the kinematic system. For each node on uniform grid we need to find the corresponding output solutions (*direct kinematic problem*). We also use interval method for this problem. The calculation of the kinematic index with interval analysis technique is described in details in [12].

4.1 The W

The W represents the reachable region defined by the robot's kinematic structure without considering joint limits or singularities.

We implement an interval method for workspace approximation based on solution existence tests. This method is described in details in [1,14]. The approach discretizes the output parameter space $\mathbf{u}$ into uniform boxes L_u and applies the interval Hansen-Sengupta operator [9] to verify solution existence in the input space $\mathbf{v}$. We classify boxes as:

- L_{in}: fully contained in workspace,
- L_b: partially contained.

The W is then computed as:

$$W = Sq(|L_{in}| + 0.5 \cdot |L_b|), \tag{6}$$

where Sq denotes unit box area. The coefficient 0.5 for border boxes was empirically validated against analytical solutions.

4.2 The SUI

The SUI quantifies the effectiveness of workspace usage by comparing achievable operational area to theoretical capacity:

$$SUI = \frac{W}{A}, \tag{7}$$

where A is area of the smallest rectangular that encloses all possible output positions, ignoring the restrictions on traversing links/singularities, but within the limits of variation of the input parameters. $SUI = 1$ indicates perfect utilization, while $SUI < 1$ reflects workspace fragmentation or inaccessible zones.

4.3 Kinematic indexes

We use three kinematic indexes: GDI, GMI, and GRI. In Table 1 the formulas for its calculation are presented. The local indexes reflects the characteristic for one point in output parameters. Its calculation is based on singular values of Jacobi matrix. The global indexes are calculated by substituting local indexes into equation (5).

The GDI measures workspace-averaged motion isotropy, critical for precision tasks. It is also the conditioning number for Jacobi matrix. GDI = 1 is the ideal isotropy robot. The GMI evaluates motion efficiency through velocity ellipsoid volume. his reflects the robot's ability to generate end-effector

velocities across orientations. Higher GMI indicates superior dynamic performance and kinetic energy transmission. The GRI quantifies robustness against external disturbances through minimum force transmission capability. Higher GRI is preferable.

Table 1: Description of kinematic performance indexes

Characteristic	Local index	Global index
Dexterity	$\eta = \frac{\sigma_{\min}}{\sigma_{\max}}$	GDI $= \frac{\sum \eta}{m^n}$
Manipulability	$\mu = \sigma_{\min} \cdot \sigma_{\max}$	GMI $= \frac{\sum \mu}{m^n}$
Resistivity	$\rho = \sigma_{\min}$	GRI $= \frac{\sum \rho}{m^n}$

5 Experimental results

We consider three parallel robots: 2-RPR, DexTar and PRRRP. All robots are planar and parallel. All of them have 2 input parameters and 2 output parameters. The implementation was done on `Python` programming language.

5.1 The 2-RPR robot

The 2-RPR robot (Fig. 1(a)) is controlled by two linear actuators capable of changing the lengths of the rods v_1, v_2, thereby controlling the position of the robot's working tool fixed at point P. Let d be the distance between the actuators, and let points A, B be fixed at the same distance ($\frac{d}{2}$) from 0 along the Ox axis.

Let the lower l and upper L limits of length variation be the same for both rods: $l \le v_1 \le L$, $l \le v_2 \le L$. And the position of the working tool P — point $u = (u_1, u_2)$.

The kinematic system of the 2-RPR robot is defined by the following system of equations:

$$F_{2RPR}(u,v) = \begin{cases} v_1^2 - (u_1 + 0.5d)^2 - u_2^2 = 0, \\ v_2^2 - (u_1 - 0.5d)^2 - u_2^2 = 0. \end{cases} \tag{8}$$

And its Jacobi matrix is defined as follows:

$$J_{2RPR}(u,v) = \begin{bmatrix} \frac{d+2u_1}{2v_1} & \frac{u_2}{v_1} \\ \frac{d+2u_1}{2v_2} & \frac{u_2}{v_2} \end{bmatrix}. \tag{9}$$

The workspace of the 2-RPR can be found geometrically. In fact, both equation in system (8) define the circles in output parameters with centers at points A and B. The radii of both circles is defined by its lower limit (l) and upper limit (L) on the rod length. Thus, each rods possible position is in rings, restrained by rods' limits . And total workspace is defined as intersection of these rings.

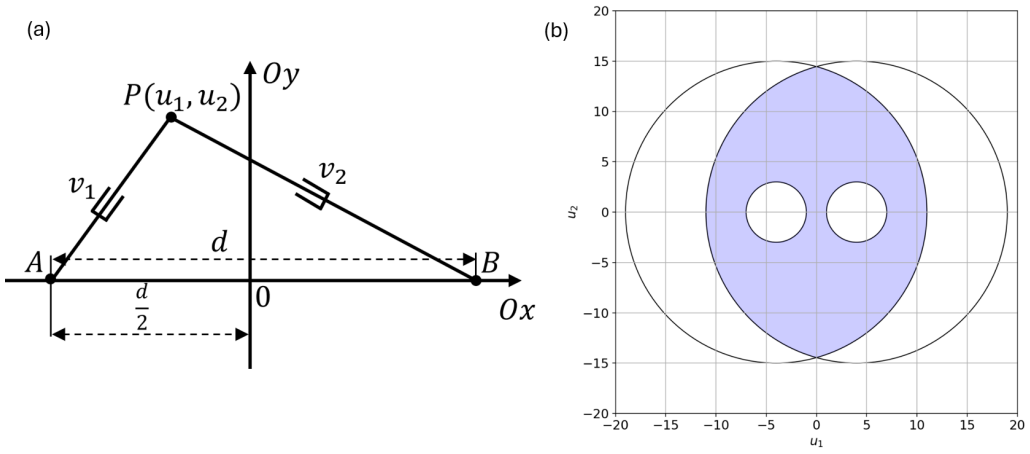

Figure 1: 2-RPR robot scheme (a) and its workspace visualization (b)

5.2 DexTar robot

Let us consider the DexTar robot. Fig. 2 (a) shows the robot scheme. The robot consists of 4 rods of constant length, which are pairwise equal: $AB = DC = a, CP = BP = b$. The distance between the rod attachment points is d. The robot is controlled by two rotary motors fixed at points A and D, which are located at an equal distance from the origin. Rotation of rods AB and CD (angles v_1 and v_2) changes the rotation angle of rods BP and CP, thereby controlling the position of the working tool fixed at point P with coordinates (u_1, u_2). Similar to the 2-RPR robot, the working area of the DexTar robot can be found geometrically as the intersection of rings formed by circles at points A and D with radii $a+b$ and $a-b$ (Fig. 2 (b)), and its kinematic system is as follows:

$$F_{DEX}(u,v) = \begin{cases} (u_1 + 0.5d - a\cos(v_1))^2 + (u_2 - a\sin(v_1))^2 - b^2 = 0, \\ (u_1 - 0.5d - a\cos(v_2))^2 + (u_2 - a\sin(v_2))^2 - b^2 = 0. \end{cases} \tag{10}$$

And its Jacobian matrix is defined as follows:

$$J_{DEX}(u,v) = \begin{bmatrix} \frac{2a\cos(v_1)+d-2u_1}{-ad\sin(v_1)+2au_1\sin(v_1)-2au_2\cos(v_1)} & \frac{2a\sin(v_1)-2u_2}{-ad\sin(v_1)+2au_1\sin(v_1)-2au_2\cos(v_1)} \\ \frac{-2a\cos(v_2)-d-2u_1}{ad\sin(v_2)+2au_1\sin(v_2)-2au_2\cos(v_2)} & \frac{2a\sin(v_2)-2u_2}{ad\sin(v_2)+2au_1\sin(v_2)-2au_2\cos(v_2)} \end{bmatrix}. \tag{11}$$

5.3 PRRRP robot

The parallel PRRRP mechanism typically consists of two parts, each of which is a PRR chain (R-pivot joint and P-prismatic joint). The two arms are connected to the end effector point by a common R-joint. The mechanism can be positioned at a point on the plane when the P-joint in each of the two legs is driven by a linear actuator. The schematic of the PRRRP robot is shown in Fig. 3 (a). The distance between two parts is $2a$, the rods' length is b. The input parameter is v_1 and v_2, which can move in range

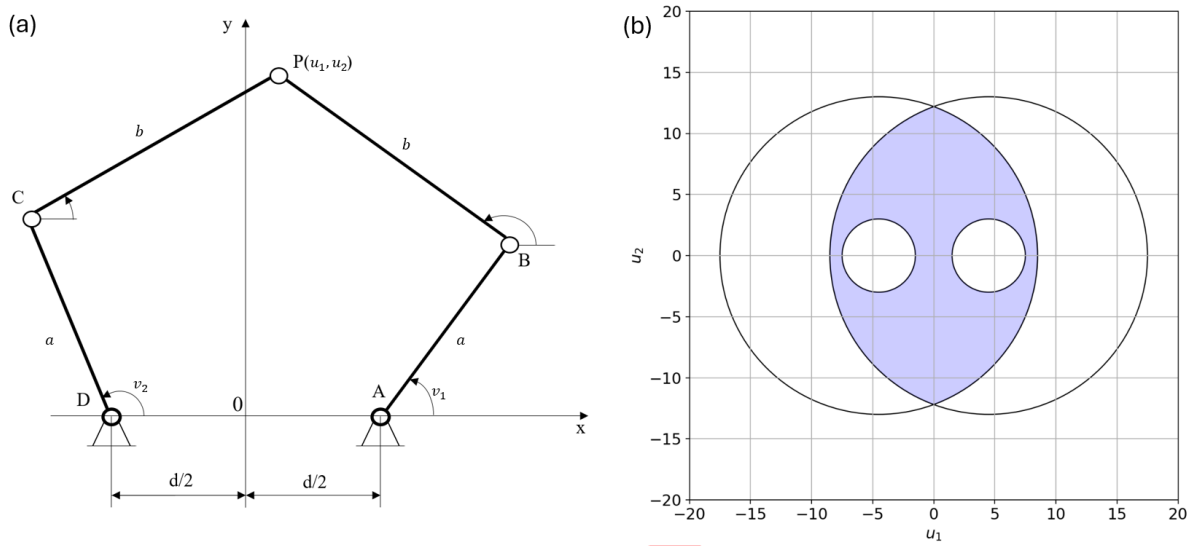

Figure 2: DexTar robot scheme (a) and its workspace visualization (b)

$[c,d]$. The output parameter is position of P point with coordinates (u_1,u_2). The geometrically found workspace is shown in Fig. 3 (b). The kinematic system of PRRRP robot is defined as follows:

$$F_{PRRRP}(u,v) = \begin{cases} (u_1-v_1)^2+(u_2-a)^2-b^2=0, \\ (u_1-v_2)^2+(u_2+a)^2-b^2=0. \end{cases} \tag{12}$$

And its Jacobi matrix is defined as follows:

$$J_{PRRRP}(u,v) = \begin{bmatrix} 1 & \frac{-2a+2u_2}{2u_1-2v_1} \\ 1 & \frac{2a+2u_2}{2u_1-2v_2} \end{bmatrix}. \tag{13}$$

5.4 Pareto-set approximation and visualization

On each generation we use NSGA-II [3] algorithm for Pareto-set approximation (according to equation (2)). In this work, we use Genetic Algorithm (GA) for Pareto set calculation. The parameters, that are used for GA is listed below:

1. Algorithm: Non-dominated Sorting Genetic Algorithm (NSGA-II);
2. Number of generations: 40;
3. Population size: 40;
4. Number of offspring: 10;
5. Selection: Tournament Selection;

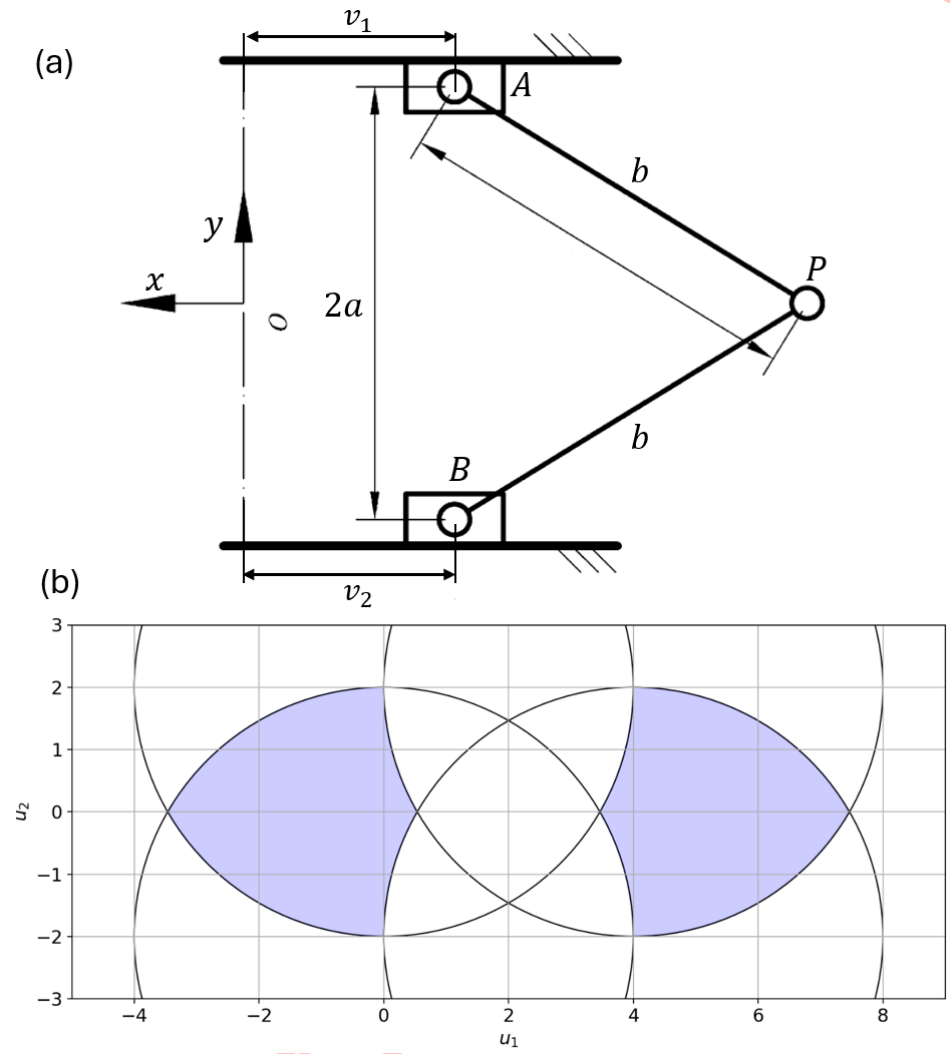

Figure 3: PRRRP robot scheme (a) and its workspace visualization (b)

6. Sampling: Integer Random Sampling;
7. Crossover: Simulated Binary Crossover, probability = 0.9;
8. Mutation: Polynomial Mutation, probability = 0.1.

For better visualization we use two types of plots: Pareto-set in parameters space visualization (3D plot) and parallel coordinates for Pareto-set in criteria space. The criteria values for points in cluster are averaged for visualization. We use the following constraints and parameters for each robot in centimeters:

1. 2RPR: $l \in [1,9], L \in [9,15], d \in [1,20]$;
2. DexTar: $a \in [1,20], b \in [1,20], d \in [1,20], v_1, v_2 \in [-\pi, \pi]$;

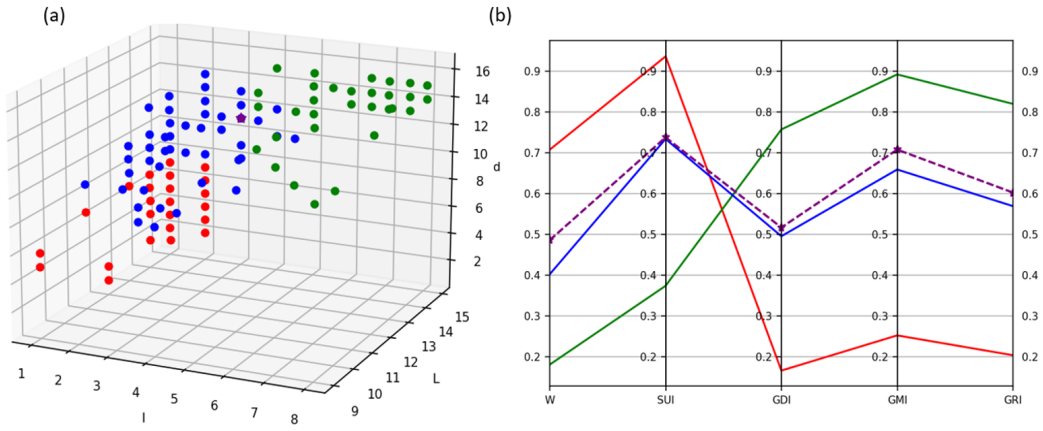

Figure 4: Clusterized Pareto-set for parameters (a) and criteria (b) for 2-RPR robot

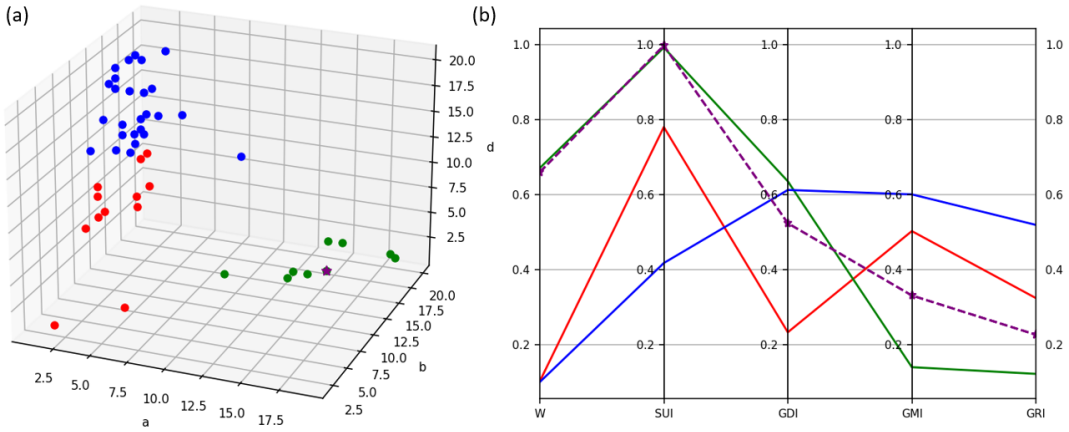

Figure 5: Clusterized Pareto-set for parameters (a) and criteria (b) for DexTar robot

3. PRRRP: $a \in [1,20], b \in [1,20], d \in [1,20], c = 0$

Since there are a lot of points in Pareto set we classify them into 3 clusters with `KMeans` algorithm [19]. Number of clusters was chosen empirically. The resulting plots are presented in Figs. 4, 5, 6 for 2-RPR, DexTar and PRRRP robot respectively. The ideal point is outlined with purple star on parameters plots and with dashed purple line in criteria plots.

We calculate ideal point according to equation (3). Each criteria value for each point in Pareto set was normalized according to formula (4).

For 2-RPR robot we can see, that there is quite strong correlation between performance indexes and workspace criteria. Green cluster points has highest l and d values. It leads to good GDI, GMI and GRI indexes, but workspace area and SUI are quite low. Vice versa, the red cluster has small l and d values, which produces big workspace area. The blue cluster is the compromise between these two

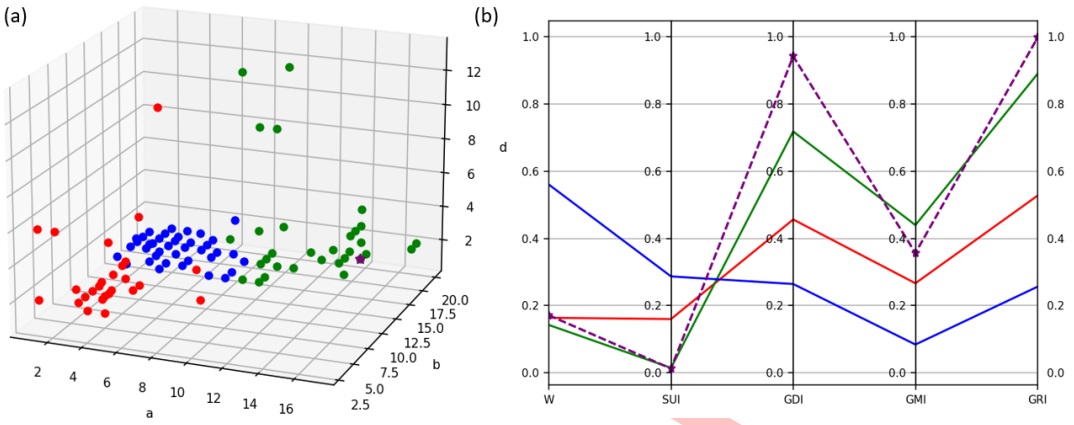

Figure 6: Clusterized Pareto-set for parameters (a) and criteria (b) for PRRRP robot

characteristics. The ideal point ($d = 11, l = 3, L = 15$) belongs to blue cluster.

For DexTar robot situation is different. The green cluster has the best workspace characteristics (due to high a and b values and small d values), but it is also has good GDI. Overall, kinematic criteria quite poor for DexTar robot and difference between clusters not so notable. That's why the ideal point ($a = 15, b = 16, d = 1$) belongs to green cluster.

Finally, for PRRRP the mean workspace characteristics for each cluster is quite bad. Green cluster has the best kinematic performance due to higher a and b values. In this case kinematic characteristics have more impact and ideal point ($a = 14, b = 19, d = 1$) belongs to green cluster.

5.5 Comparison with exact pareto-front and randomsampling method

We compare the proposed approach with two baseline methods: Random Sampling and exact Brute Force algorithm. The methodology for each approach is as follows:

Exact Brute Force algorithm:

1. Create a discrete uniform grid in the parameter space considering only integer values;
2. Calculate objective values for all grid points;
3. Extract Pareto-optimal points using equation (2).

Random Sampling method:

1. Sample N points uniformly from the parameter space;
2. Calculate objective values for these points;
3. Extract Pareto-optimal points using equation (2).

Since the design parameters in our robotic optimization problem are constrained to integer values, we can calculate the exact Pareto set using the Brute Force algorithm, which serves as a ground truth for comparison.

We evaluate the methods using two key metrics: execution time and hypervolume value. The hypervolume metric quantifies the volume of the objective space dominated by the Pareto front relative to a reference point. It provides a comprehensive measure of both convergence (proximity to the true Pareto front) and diversity (uniform distribution along the front). The hypervolume is calculated as follows according to equation (4):

$$HV(P,r) = \lambda\left(\bigcup_{p\in P}\{x \in \mathbb{R}^k \mid p \prec x \prec r\}\right),$$

where P is the Pareto set, r is a reference point (we use the nadir point), λ is the Lebesgue measure, and $k = 5$ is the number of objectives. Before hypervolume calculation, all criteria values are normalized to $[0,1]$ range using min-max scaling. Higher hypervolume values indicate superior Pareto front quality.

All experiments were conducted on a local infrastructure server equipped with an Intel Xeon Gold 6248R processor @ 3.0 GHz. For the RandomSampling algorithm, we use $N = 480$ points, which equals the total number of function evaluations performed by our NSGA-II algorithm.

Table 2: Comparison of the proposed approach with exact Pareto-front calculation

Robot	**2-RPR**			**DexTar**			**PRRRP**		
Method	Our	*E*	*RS*	Our	*E*	*RS*	Our	*E*	*RS*
Execution time, s	631	1093	595	689	9874	623	643	8279	584
Hypervolume	0.181	0.184	0.112	0.173	0.198	0.121	0.236	0.247	0.142

The comparison results for the three parallel manipulators are presented in Table 2, where *E* is the result for exact solution and *RS* is the results for Random Sampling method. The exact method consistently provided the highest hypervolume values but required the longest computation time: 14.3 times longer than the proposed method for DexTar and 12.9 times longer for PRRRP. Random sampling was the fastest but produced the lowest hypervolume values across all three robotic platforms. The proposed method's execution time was 6-16% higher than random sampling, while its hypervolume was 61-66% higher. Compared to the exact solution, it delivered 87.4-98.4% of the hypervolume value using 42-93% less computation time.

6 Conclusion

In this work we extend the method for multi-criteria optimization of robots' key characteristics. In this work we consider 5 criteria problem, where 2 criteria reflects workspace characteristics (workspace area and SUI) and kinematic indexes (GDI, GMI and GRI). The optimization was done for 3 parallel manipulators (2-RPR, DexTar, PRRRP) with 3 design parameters. Interval analysis technique with Genetic Algorithms provides automated approach for Pareto-set calculation.

Big number of Pareto points and five-dimensional criteria space requires additional techniques for visualization. We use clusterization algorithm for classifying Pareto points. In this work, we have only 3 parameters, so they can be visualized on 3D plots. The clusters in criteria space were visualized with parallel axis. Such visualization can provide additional information for engineers for decision-making on a design task.

The experimental validation included comparison with two baseline methods: exact brute-force enumeration and uniform random sampling. The exact method consistently yielded the highest hypervolume but required 12.9–14.3 times longer computation for the more complex architectures. Random sampling was the fastest but produced the lowest hypervolume in all cases. The proposed method delivered 87.4–98.4% of the exact hypervolume value with 42–93% less computation time and outperformed random sampling by 61–66% in hypervolume with only 6–16% additional runtime.

The optimization of spatial (3D) robots remains a challenging task. We plan to test this approach for 3-RPR, Tripod and Delta robots. The additional constraints (singularities, joint constrains) can also be useful for robot optimization.

Acknowledgments

This work was supported by the Ministry of Science and Higher Education of the Russian Federation, project no. 075-15-2024-544.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

[1] S. Caro, D. Chablat, A. Goldsztejn, D. Ishii, C. Jermann, *A branch and prune algorithm for the computation of generalized aspects of parallel robots*, Artif. Intell. **211** (2014) 34–50.

[2] Y. Chen, X. Han, F. Gao, Z. Wei, Y. Zhang, *Workspace analysis of a 2-dof planar parallel mechanism*, Atlantis Press (2015) 192–195.

[3] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Trans. Evol. Comput. **6(2)** (2002) 182–197.

[4] M. Ehrgott, *Multicriteria Optimization*, Springer Science & Business Media **491** (2005).

[5] M. Gallant, R. Boudreau, *The synthesis of planar parallel manipulators with prismatic joints for an optimal, singularity-free workspace*, J. Robot. Syst. **19(1)** (2002) 13–24.

[6] C. Gosselin, J. Angeles, *A global performance index for the kinematic optimization of robotic manipulators*, J. Mech. Des. **113** (1991) 220–226.

[7] C. Gosselin, J. Angeles, *Singularity analysis of closed-loop kinematic chains*, IEEE Trans. Robot. Autom. **6(3)** (1990) 281–290.

[8] C. Gosselin, *Dexterity indices for planar and spatial robotic manipulators*, Proc. IEEE Int. Conf. Robot. Autom. (1990) 650–655.

[9] E. Hansen, S. Sengupta, *Bounding solutions of systems of equations using interval analysis*, BIT **21(2)** (1981) 203–211.

[10] T. Huang, M. Li, Z. Li, D. Chetwynd, D. Whitehouse, *Optimal kinematic design of 2-DOF parallel manipulators with well-shaped workspace bounded by a specified conditioning index*, IEEE Trans. Robot. Autom. **20(3)** (2004) 538–543.

[11] D. Lera, M. Posypkin, Y. Sergeyev, *Space-filling curves for numerical approximation and visualization of solutions to systems of nonlinear inequalities with applications in robotics*, Appl. Math. Comput. **390** (2021).

[12] A. Maminov, *Automated Multi-criteria Optimization of Parallel Robots*, Lecture Notes in Comput. Sci. **15218** (2024) 125–138.

[13] A. Maminov, M. Posypkin, *Constrained multi-objective robot's design optimization*, IEEE Conf. Russ. Young Res. Electr. Electron. Eng. (2020) 1992–1995.

[14] A. Maminov, M. Posypkin, S. Shary, *Reliable bounding of the implicitly defined sets with applications to robotics*, Procedia Comput. Sci. **186** (2021) 227–234.

[15] J.-P. Merlet, *Parallel robots*, Springer Science & Business Media **128** (2006).

[16] Y.-J. Nam, M.-K. Park, *Workspace optimization and kinematic performance evaluation of 2-DOF parallel mechanisms*, J. Mech. Sci. Technol. **20(10)** (2006) 1614–1625.

[17] S. Opricovic, G.-H. Tzeng, *Extended VIKOR method in comparison with outranking methods*, Eur. J. Oper. Res. **178(2)** (2007) 514–529.

[18] Z. Qu, P. Zhang, Y. Hu, H. Yang, T. Guo, K. Zhang, J. Zhang, *Optimal design of agricultural mobile robot suspension system based on NSGA-III and TOPSIS*, Agriculture **13(1)** (2023).

[19] K. Sinaga, M.-S. Yang, *Unsupervised K-means clustering algorithm*, IEEE Access **8** (2020) 80716–80727.

[20] V. Pandey, H. Dincer, *A review on TOPSIS method and its extensions for different applications with recent development*, Soft Comput. **27(23)** (2023) 18011–18039