Theory and Practice of Science and Technology
2025, VOL. 6, NO. 8, 52-57
DOI: 10.47297/taposatWSP2633-456910.20250608

Hybrid Computational Framework for Multi-Objective Robotic Motion Planning: Integrating Nonlinear Programming, Swarm Intelligence, and Genetic Optimization

Chen Zhang*, Jiacheng Li, Haoyue Guan, Enhao Liu, Jianshu Song, Jiaji Liu, Hongtao Zhu

Shandong Jianzhu University, Jinan, Shandong, 250000, China

ABSTRACT

This paper addresses a multi-objective optimization problem for robotic manipulator motion planning, focusing on minimizing end-effector positioning error and energy consumption while optimizing both joint angle trajectories and mobile base paths. We propose a computational framework integrating forward kinematics modeling with a hybrid optimization strategy that combines nonlinear programming, particle swarm optimization, deep/breadth-first search, and genetic algorithms. The method first optimizes joint trajectories under kinematic constraints, then determines optimal base positioning and shortest feasible routes while avoiding obstacles and finally extends to multi-target task planning through genetic algorithm – based sequence optimization. Experimental results demonstrate that the proposed approach achieves high positioning accuracy with significantly reduced energy consumption, while maintaining robust adaptability to diverse task configurations. This work provides a generalizable solution for intelligent robotic path planning and multi-objective decision optimization.

KEYWORDS

Particle swarm optimization; Path planning; Genetic algorithm

1 Introduction

Robotic motion planning tasks often require simultaneous optimization of joint trajectories and mobile-platform paths in cluttered environments. Key computational challenges include minimizing end-effector positioning error, reducing energy consumption, and ensuring obstacle avoidance under kinematic and dynamic constraints.

Existing solutions for manipulator trajectory optimization often rely on meta-heuristic algorithms—such as particle swarm optimization (PSO) or genetic algorithms—and graph search methods. However, these are typically applied in isolation, limiting simultaneous control over base motion, joint kinematics, energy efficiency, and multi-target route optimization. Techniques like PSO-based trajectory planning [1-2] and graph search for path planning [3] provide partial solutions, but integrated frameworks covering all objectives remain under-explored.

This paper proposes a hybrid computational framework that unifies forward kinematics modeling, nonlinear programming, PSO, depth/breadth-first search, and genetic algorithms into a single pipeline. First, joint trajectories are optimized under kinematic constraints via nonlinear programming. Next, base pre-positioning is determined by PSO, followed by BFS graph-based motion planning for obstacle avoidance. Finally, multi-target visiting sequences are optimized via genetic algorithms. This integrated pipeline achieves high precision in end-effector positioning and significant reductions in energy consumption while generating collision-free paths for both single- and multi-target operations. We validate this framework experimentally, comparing efficiency gains against benchmarks in recent literature [4-5].

2 Kinematic modeling and joint trajectory optimization

2.1 Establishment of the initial manipulator configuration

The zero-position configuration of the manipulator is defined as the predetermined initial setup when the control system is activated or reset. This configuration provides a standardized reference for calibration and motion planning. According to the given joint parameters ($\theta_1=0^\circ,\theta_2=-90^\circ,\theta_3=0^\circ,\theta_4=180^\circ,\theta_5=-90^\circ,\theta_6=0^\circ$) and initial structural parameters, a schematic diagram of the manipulator was constructed. The provided Denavit – Hartenberg (D-H) parameters were identified as modified D-H parameters rather than the standard form; thus, the modeling process was performed entirely under the modified convention.

*** Corresponding Author:** Chen Zhang, 2540901518@qq.com

2.2 Forward kinematics modeling

The manipulator's motion is characterized by coordinated variations of multiple joints, mathematically represented by continuous transformations between local D-H coordinate frames. This transformation integrates two fundamental spatial coordinate operations: translation and rotation. Figure 1 illustrates a general coordinate transformation in 3D space, which serves as the foundation for constructing the forward kinematics model.

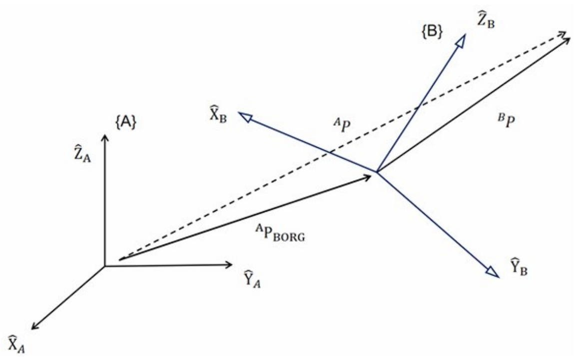

Figure 1 General coordinate transformation between spatial frames

For two coordinate frames A and B, the vector transformation can be expressed as:

$$^{A}P = {}^{A}_{B}R\,{}^{B}P + {}^{A}P_{BORG} \tag{1}$$

Introducing the transformation operator T:

$$^{A}P = {}^{A}_{B}T\,{}^{B}P \tag{2}$$

The homogeneous transformation matrix form is:

$$\begin{bmatrix} ^{A}P \\ 1 \end{bmatrix} = \begin{bmatrix} ^{A}_{B}R & ^{A}P_{\mathrm{BORG}} \\ 000 & 1 \end{bmatrix} \begin{bmatrix} ^{B}P \\ 1 \end{bmatrix} \tag{3}$$

Translation and rotation operators are defined respectively as:

$$D_Q(q) = \begin{bmatrix} 1 & 0 & 0 & q_x \\ 0 & 1 & 0 & q_y \\ 0 & 0 & 1 & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$R_Z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

The transformation from frame $i-1$ to frame i is given by:

$$^{i-1}_{i}T = R_X(\alpha_{i-1})\,D_X(a_{i-1})\,R_Z(\theta_i)\,D_Z(d_i) \tag{6}$$

The complete forward kinematics model is then obtained by successive matrix multiplications:

$$^{0}_{n}T = {}^{0}_{1}T\,{}^{1}_{2}T \cdots\cdots {}^{n-1}_{n}T \tag{7}$$

2.3 Joint trajectory optimization via nonlinear programming

The objective is to minimize the Euclidean distance between the actual end-effector position and the target position (1500,mm,1200,mm,200,mm). This is formulated as a nonlinear programming problem, where both the objective function and constraints may be nonlinear. The optimization process follows a convex optimization paradigm to ensure convergence to the global minimum.

The optimization variables are the six joint angles θ_1 to θ_6, with search bounds given by the specified joint limits. The initial guess vector corresponds to the zero-position configuration. The optimization was implemented using the minimize function from the SciPy library, which supports multiple algorithms such as gradient descent, conjugate gradient, and trust-region methods. Bound constraints were enforced to ensure numerical stability and prevent infeasible solutions.

As shown in Table.1, the optimization yields an end-effector position error of approximately 1.66×10^{-7} mm, confirming extremely high precision in the computed trajectory.

Table 1 Optimized joint angle trajectories for Problem 1

θ_i	Initial Joint Angle Path	Final Joint Angle Path
θ_1	0°	38.65°
θ_2	-90°	-33.82°
θ_3	0°	17.56°
θ_4	180°	179.76°
θ_5	-90°	-93.21°
θ_6	0°	0°

3 Joint trajectory optimization considering energy consumption using particle swarm optimization

3.1 Energy consumption modeling

To account for both positioning accuracy and operational efficiency, the optimization objective includes minimizing total energy consumption while constraining the allowable end-effector position error to ±200 mm. The energy consumption model considers three primary components: inertial work, gravitational work, and frictional work.

The inertial work is expressed as:

$$W_i = J \cdot \omega^2 \tag{8}$$

The gravitational work is calculated as:

$$W_g = mgh \tag{9}$$

where $g = 9.80665$ m/s^2 is the gravitational acceleration, and h is determined from the forward kinematics model.

The frictional work is defined as:

$$W_f = F_\mu \cdot l \tag{10}$$

For rotational motion:

$$l = \Delta\theta \cdot r \tag{11}$$

where $\Delta\theta$ is the joint rotation angle and r is the joint radius. Considering its negligible effect on optimization results, r is set to 0.1 m and F_μ to 10 N.

Thus, the total work is given by:

$$W = \left| W_i + W_g + W_f \right| \tag{12}$$

3.2 Particle swarm optimization formulation

The multi-objective optimization problem is addressed using Particle Swarm Optimization (PSO), which efficiently searches for optimal joint trajectories under the given constraints. The velocity and position update rules are:

$$v_i^d = \omega \times v_i^d + c_1 \times \mathrm{rand}_i^d \times \left(pBest_i^d - X_i^d \right) + c_2 \times \mathrm{rand}_2^d \times \left(pBest^d - X_i^d \right) \tag{13}$$

$$X_i^d = x_i^d + v_i^d \tag{14}$$

Here, ω is the inertia weight, c_1 and c_2 are acceleration coefficients, rand_1^d and $\mathrm{rand2}^d$ are uniformly distributed random numbers in[0,1], and*pBest*, *gBest* denote the personal and global best solutions, respectively. Velocity is clamped by a predefined $V\max$ for stability.

The objective function simultaneously minimizes end-effector position error and total energy consumption, with a penalty term enforcing the error bound of ±200 mm. The algorithm includes individual generation, boundary correction, and mutation/crossover operations to enhance diversity and convergence reliability.

3.3 Experimental configuration and results

Extensive parameter tuning indicates that the optimal settings are: population size 200, crossover probability 70%, mutation probability 30%, and 300 generations. The Pareto front solutions are filtered using a practical criterion: if the position error is within 100 mm—smaller than the effective gripper range—the solution is considered acceptable. Among these, the trajectory with the lowest total energy consumption is selected.

The final optimal solution yields an energy consumption of 69.43 J and a position error of 95.21 mm. As shown in Table.2, the optimized joint angle trajectories demonstrate substantial improvement in both accuracy and efficiency.

4 Integrated optimization of base motion and joint trajectory

4.1 Determination of optimal base distance

The mobile base of the manipulator can be positioned arbitrarily; thus, the optimal distance D_{best} between the base

Table 2 Optimized joint trajectories considering energy consumption

θ_i	Initial Joint Angle Path	Final Joint Angle Path
θ_1	0°	39.62°
θ_2	-90°	-39.89°
θ_3	0°	23.87°
θ_4	180°	-179.15°
θ_5	-90°	112.30°
θ_6	0°	-179.95°

origin and the target object projection along the X-axis must be determined. The projection coordinates of the grasping point in the base coordinate system are defined as:

$$^{n}P=[D_{\text{best}} \quad 0 \quad 200]^T \tag{15}$$

The optimization is formulated as a constrained minimization problem, where D_{best} is introduced as an additional decision variable. This study employs the same Particle Swarm Optimization (PSO) framework described in Section 5.2 to determine the optimal D_{best} value.

4.2 Pre-positioning of the mobile base

To improve task execution efficiency, a search strategy is designed to determine feasible pre-positioning locations for the mobile base. A depth-first search (DFS) method is applied to evaluate spatial relationships within the discretized workspace, identifying collision-free base positions. The resulting feasible pre-position positions are visualized on a grid map, with color coding to indicate optimal configurations for subsequent motion planning.

4.3 Pre-positioning of the mobile base

The problem of determining the shortest feasible base movement path is addressed using the Breadth-First Search (BFS) algorithm. BFS is well suited for finding shortest paths in discrete, uniformly weighted graphs. The algorithm explores nodes level-by-level from the starting position, updating path costs and storing predecessors until the goal is reached. The recorded path is then reconstructed from the goal node back to the start, yielding the globally shortest feasible base movement route.

The BFS-based planning approach ensures that the base reaches the target position while avoiding obstacles, minimizing both travel distance and redundant movements. This method is integrated with the previously optimized joint trajectory to generate the final coordinated motion plan. The resulting grid-based base movement path is shown in Figure 2.

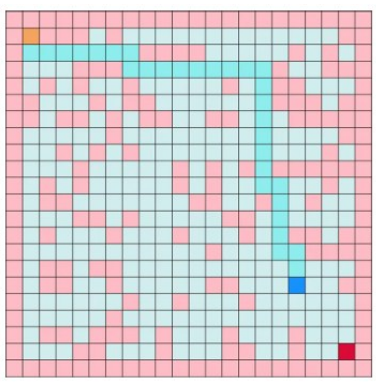

Figure 2 Optimized mobile base movement path on discretized grid map

4.4 Experimental results

The final solution minimizes end-effector positioning error and total energy consumption while ensuring obstacle avoidance and task feasibility. As shown in Table.3, the optimized joint trajectories demonstrate significant improvements in accuracy compared with the initial configuration.

5 Multi-target path planning using genetic algorithms

5.1 Problem formulation

The objective of this stage is to determine an optimal visiting sequence for multiple target points while minimizing total travel distance and ensuring feasible manipulator configurations at each location. The optimization problem is modeled as a combinatorial search problem, where each target corresponds to a node in a complete weighted graph, and edge weights represent the cost of moving between targets.

The total cost function is expressed as:

$$L = \sum_{i=1}^{n-1} d\left(P_i, P_{i+1}\right) \tag{16}$$

where P_i denotes the i-th target point in the visiting sequence and $d(\cdot)$ represents the distance metric. The optimization seeks a permutation of targets that minimizes L under manipulator reachability constraints.

Table 3 Optimized joint trajectories for integrated base and joint motion planning

θ_i	Initial Joint Angle Path	Final Joint Angle Path
θ_1	0°	-0.59°
θ_2	-90°	-52.84°
θ_3	0°	78.16°
θ_4	180°	-179.57°
θ_5	-90°	97.61°
θ_6	0°	-179.57°

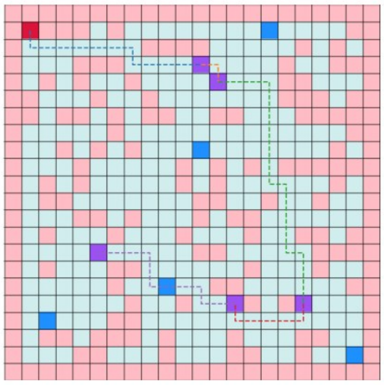

Figure 3 Optimized multi-target visiting path generated by the Genetic Algorithm

5.2 Genetic algorithm framework

A Genetic Algorithm (GA) is adopted to search for the optimal sequence. Each chromosome encodes a complete visiting order of all targets. Standard GA operations are applied:

Selection: Individuals are chosen based on a fitness-proportional rule, where fitness is inversely related to the total path length.

Crossover: An order-based crossover operator is used to preserve relative target positions while exchanging subsequences between parents.

Mutation: A swap mutation is applied to maintain diversity and avoid premature convergence.

The evolution process is repeated until convergence criteria are met, such as a maximum generation limit or stagnation in the best fitness value.

5.3 Experimental results

Extensive tests are conducted to evaluate the performance of the GA-based multi-target path planning method. The convergence curves show that the algorithm quickly reduces total path length in early generations and stabilizes toward the global optimum in later stages.

As shown in Figure 3, the optimal path generated by the proposed GA framework connects all target points in a sequence that minimizes travel distance while satisfying manipulator operational constraints.

6 Conclusions

This paper proposes a hybrid computational framework for robotic manipulator motion planning that integrates forward kinematics modeling, nonlinear programming, particle swarm optimization, graph search algorithms, and genetic algorithms to address multi-objective optimization of joint trajectories and base motion. The method achieves high end-effector positioning accuracy while significantly reducing total energy consumption, and it successfully generates collision-free paths for both single-target and multi-target scenarios. Experimental results demonstrate that the proposed approach not only meets stringent accuracy constraints but also improves planning efficiency and adaptability to varying task configurations. However, the current framework assumes fixed manipulator geometry and simplified energy models, which may limit accuracy under conditions involving complex joint dynamics or varying payloads. Additionally, collision detection is based on discretized maps, which may reduce precision in highly cluttered environments. Future work will focus on incorporating full dynamic modeling, real-time collision avoidance in continuous space, and adaptive optimization strategies for changing task requirements. The proposed framework provides a scalable and generalizable solution for intelligent robotic path planning, with potential applicability to a wide range of automated manufacturing and autonomous manipulation tasks.

References

[1] Akbari A, Mozaffari S, Singh R, et al. Movement Optimization of Robotic Arms for Energy and Time Reduction using Evolutionary Algorithms [C]//2023 International Symposium on Signals, Circuits and Systems (ISSCS). IEEE, 2023: 1-4.

[2] Wang J, Zhang Y, Zhu S, et al. A Novel Multi-Objective Trajectory Planning Method for Robots Based on the Multi-Objective Particle Swarm Optimization Algorithm[J]. Sensors (Basel, Switzerland), 2024, 24(23): 7663.

[3] Marcucci T, Petersen M, von Wrangel D, et al. Motion planning around obstacles with convex optimization[J]. Science robotics, 2023, 8(84): eadf7843.

[4] Xin J, Li Z, Zhang Y, et al. Efficient real-time path planning with self-evolving particle swarm optimization in dynamic scenarios[J]. Unmanned Systems, 2024, 12(02): 215-226.

[5] Liu J, Yap H J, Khairuddin A S M. Review on motion planning of robotic manipulator in dynamic environments[J]. Journal of Sensors, 2024, 2024(1): 5969512.