Multiverse Journal
E-ISSN: 3080-3942, P-ISSN: 3080-3934
Volume 1, Issue 2, December 2024 | PP. 166-193

Research Paper

Multi-Objective Path Planning and Energy Efficiency Optimization Methods for Industrial Robot Collaborative Systems

[1]WenHao chen
[1]Ulster College, Shaanxi University of Science and Technology, Xi'an, Shaanxi, China, 710021
jack_cwh@163.com

Received: 08 September, 2024 **Accepted:** 11 November 2024 **Published:** 31 December, 2024

Abstract

This paper reviews multi-objective path planning and energy efficiency methods for collaborative industrial robots, examining algorithms like NSGA-II, NSGA-III, ant colony optimization, and particle swarm optimization. It discusses mathematical models, algorithm pseudocode, and performance evaluations. Hybrid strategies outperform others, with ant colony optimization reducing energy use by 21.6% and achieving conflict-free paths. Dynamic factors like payload and joint velocities affect system efficiency. Results show trade-offs: aggressive time optimization may increase energy by 35%, while moderate improvements need only 8-12% more time. The study identifies gaps in real-time adaptability, scalability for large multi-robot systems, human-robot collaboration, and uncertainty handling. Future work should develop edge-friendly algorithms, distributed planning, robust conflict resolution, and energy-efficient scheduling for Industry 4.0 and 5.0.

Keywords: Multi-objective optimization, path planning, energy efficiency, industrial robots, collaborative systems

Introduction

The rapid advancement of Industry 4.0 and the emerging paradigm of Industry 5.0 have catalyzed widespread adoption of collaborative robot systems in manufacturing environments, where multiple robots coordinate movements to accomplish complex tasks efficiently while minimizing energy consumption and ensuring safe operation (Soori, Arezoo, & Dastres, 2023). Industrial robot collaborative systems face multifaceted optimization challenges requiring simultaneous consideration of conflicting objectives: minimizing cycle time, reducing energy expenditure, ensuring collision-free trajectories, maintaining smooth motion profiles, and optimizing resource utilization across the entire system. Traditional single-objective optimization approaches prove inadequate for collaborative robot systems, failing to capture the complex trade-offs between competing performance criteria that characterize real-world industrial applications. The fastest path may consume excessive energy through aggressive acceleration profiles and high-velocity movements, while the most energy-efficient trajectory might result in unacceptable cycle times that reduce manufacturing throughput or collision risks that compromise system safety (Carabin, Wehrle, & Vidoni, 2020). This fundamental limitation has driven extensive research into multi-objective optimization frameworks that simultaneously optimize multiple metrics while providing Pareto-optimal solutions representing different trade-off configurations. Recent advances in computational intelligence, particularly evolutionary algorithms and swarm intelligence methods, have opened new avenues for solving multi-objective path planning problems in industrial robotics with improved solution quality and computational efficiency (Li, Xie, Liu, Cao, & Chen, 2023). Energy efficiency has emerged as a critical concern in industrial automation, driven by both economic imperatives and environmental sustainability goals. Industrial robots account for substantial portions of manufacturing energy consumption, with estimates suggesting that robotic systems consume 15-25% of total factory energy in automated production facilities (Paryanto, Brossog, Bornschlegl, & Franke, 2015). Optimizing robot energy consumption directly reduces operational costs, extends equipment lifespan through reduced thermal stress, and contributes to corporate sustainability objectives aligned with global carbon reduction targets. Energy-efficient path planning considers multiple factors including joint velocities, acceleration profiles, payload characteristics, and gravitational effects to minimize total energy expenditure while maintaining productivity requirements (Nonoyama, Liu, Fujiwara, Alam, & Nishi, 2022).

Collaborative robot systems introduce additional complexity through inter-robot dependencies and collision avoidance requirements. Multiple robots operating in shared workspaces must coordinate movements to prevent collisions while optimizing global system performance rather than individual robot metrics (Lin, Zheng, Xu, & Zhan, 2022). This coordination challenge necessitates sophisticated conflict resolution mechanisms that balance local optimization with global objectives, requiring algorithms capable of handling dynamic constraints and real-time replanning when environmental conditions change or unexpected obstacles appear. The multi-objective nature of industrial robot path planning creates a complex optimization landscape characterized by conflicting objectives, non-convex solution spaces, and multiple local optima. Evolutionary algorithms such as Non-dominated Sorting Genetic Algorithm II (NSGA-II) and its successor NSGA-III have demonstrated effectiveness in exploring these complex spaces and generating diverse Pareto-optimal solution sets (Wang, Gao, Wei, & Fei, 2023). Swarm intelligence methods including ant colony optimization (ACO) and particle swarm optimization (PSO) offer alternative approaches that leverage collective behavior principles to discover high-quality solutions through distributed search mechanisms (Nazarahari, Khanmirza, & Doostie, 2019). Hybrid approaches combining multiple algorithmic paradigms have shown particular promise, achieving superior performance by leveraging complementary strengths of different optimization strategies (Zhang, Xu, & Wang, 2023).

This paper provides a comprehensive analysis of multi-objective path planning and energy efficiency optimization methods for industrial robot collaborative systems. We systematically review theoretical foundations, algorithmic approaches, and practical implementations, examining how different methods address the fundamental challenges of multi-robot coordination, energy optimization, and real-time performance. The paper makes several key contributions: (1) comprehensive review of multi-objective optimization frameworks and their application to industrial robotics, (2) detailed analysis of energy consumption models incorporating dynamic factors and system-level effects, (3) systematic comparison of evolutionary and swarm intelligence algorithms with quantitative performance evaluation, (4) identification of critical research gaps and practical implementation challenges, and (5) recommendations for future research directions aligned with emerging industrial automation paradigms. The remainder of this paper is organized as follows. Section 2 presents a comprehensive literature review covering multi-objective optimization frameworks, evolutionary and swarm intelligence approaches, energy efficiency methods, and conflict resolution strategies. Section 3 describes the methodology including problem formulation, algorithmic approaches, energy modeling, and conflict resolution mechanisms. Section 4 presents simulation experiments with detailed quantitative results and comparative analysis. Section 5 discusses key findings, trade-offs, and implementation challenges. Section 6 outlines future research directions, and Section 7 concludes the paper.

2. Literature Review

2.1 Multi-Objective Optimization Frameworks

Multi-objective optimization (MOO) frameworks provide mathematical foundations for simultaneously optimizing multiple conflicting objectives in robot path planning. Unlike single-objective optimization that seeks a single optimal solution, MOO generates a set of Pareto-optimal solutions representing different trade-offs between objectives (Wang et al., 2023). A solution is Pareto-optimal if no other solution improves one objective without degrading at least one other objective, creating a Pareto front that visualizes the trade-off space available to decision-makers. The mathematical formulation of multi-objective robot path planning typically involves minimizing a vector of objective functions subject to kinematic, dynamic, and collision avoidance constraints. The general form can be expressed as:

Minimize: $F(x) = [f_1(x), f_2(x), ..., f_k(x)]^T$

Subject to:

- $g_i(x) \leq 0, i = 1, ..., m$ (inequality constraints)
- $h_j(x) = 0, j = 1, ..., p$ (equality constraints)
- $x \in X$ (decision variable bounds)

where x represents the decision variables (typically waypoint positions, joint angles, or trajectory parameters), f_1 through f_k represent the k objective functions (e.g., path length, energy consumption, smoothness, cycle time), and g_i and h_j represent constraint functions encoding kinematic limits, collision avoidance, and workspace boundaries (Zhao, Gao, Huang, & Tang, 2020).

Common objective functions in industrial robot path planning include: (1) path length or cycle time minimization to maximize throughput, (2) energy consumption minimization to reduce operational costs, (3) trajectory smoothness optimization to reduce mechanical wear and vibration, (4) manipulability maximization to avoid singular configurations, and (5) safety margin maximization to ensure robust collision avoidance (Lan, Xie, Liu, Cao, & Chen, 2020). The relative importance of these objectives varies across applications, with high-throughput manufacturing prioritizing cycle time while energy-intensive operations emphasize efficiency. Performance evaluation in multi-objective optimization requires specialized metrics that assess both convergence to the true Pareto front and diversity of solutions along the front. The hypervolume indicator measures the volume of objective space dominated by a solution set, providing a single metric that captures both convergence and diversity (Pan, Li, Cheng, He, & Tan, 2022). The inverted generational distance (IGD) measures the average distance from points on the true Pareto front to the nearest solution in the obtained set, quantifying convergence quality. The spacing metric evaluates the uniformity of solution distribution along the Pareto front, with lower values indicating more evenly distributed solutions that provide better coverage of the trade-off space. Recent advances in multi-objective optimization have focused on handling many-objective problems (more than three objectives) where traditional Pareto dominance becomes less effective due to the curse of dimensionality (Wang et al., 2023). Reference point-based methods such as NSGA-III decompose the objective space using predefined reference directions, maintaining selection pressure toward the Pareto front even in high-dimensional objective spaces. Indicator-based methods such as SMS-EMOA use quality indicators directly in the selection process, providing more robust performance in many-objective scenarios common in complex industrial applications.

2.2 Evolutionary and Swarm Intelligence Approaches

Evolutionary algorithms (EAs) have become dominant approaches for multi-objective robot path planning due to their ability to explore complex solution spaces and generate diverse Pareto-optimal solution sets. The Non-dominated Sorting Genetic Algorithm II (NSGA-II) introduced by Deb et al. has achieved widespread adoption through its efficient non-dominated sorting procedure, crowding distance mechanism for diversity preservation, and elitist selection strategy that ensures monotonic improvement across generations (Jose & Pratihar, 2016). NSGA-II maintains a population of candidate solutions, applies genetic operators (selection, crossover, mutation) to generate offspring, and uses Pareto dominance ranking combined with crowding distance to select the next generation, gradually converging toward the Pareto front while maintaining solution diversity. The algorithmic structure of NSGA-II for robot path planning involves several key components. The chromosome encoding represents candidate paths, typically using waypoint coordinates, joint angle sequences, or spline parameters. The fitness evaluation computes all objective functions for each candidate solution, requiring forward kinematics, dynamics simulation, and collision checking. The non-dominated sorting procedure assigns each solution a rank based on Pareto dominance relationships, with rank 1 solutions forming the current Pareto front. The crowding distance calculation estimates solution density in objective space, favoring solutions in less crowded regions to maintain diversity. The selection operator combines rank and crowding distance to choose parents for reproduction, while crossover and mutation operators generate offspring with genetic variation (Hou, Xie, Liu, & Yin, 2022).

NSGA-III extends NSGA-II to handle many-objective problems through reference point-based selection. Instead of crowding distance, NSGA-III uses predefined reference directions to guide selection, maintaining uniform distribution across the Pareto front even when the number of objectives exceeds three. This approach has proven particularly effective for industrial robot applications involving multiple competing objectives such as time, energy, smoothness, manipulability, and safety margins (Wang et al., 2023). Swarm intelligence methods offer alternative approaches inspired by collective behavior in natural systems. Ant Colony Optimization (ACO) mimics foraging behavior of ant colonies, where individual ants deposit pheromone trails that guide subsequent ants toward promising paths. In robot path planning, artificial ants construct candidate paths by probabilistically selecting waypoints based on pheromone concentration and heuristic information, with pheromone updates reinforcing high-quality solutions over iterations (Nazarahari et al., 2019). The probability of ant k selecting node j from current node i is given by:

$$P_{ij}^k = [\tau_{ij}^\alpha \cdot \eta_{ij}^\beta] / \Sigma[\tau_{il}^\alpha \cdot \eta_{il}^\beta]$$

where τ_{ij} represents pheromone concentration on edge (i,j), η_{ij} represents heuristic information (typically inverse of edge cost), and α and β are parameters controlling the relative influence of pheromone and heuristic information (Li et al., 2023). Priority-free ant colony optimization (PFACO) extends traditional ACO to multi-robot systems by incorporating energy-constrained heuristics and conflict resolution mechanisms. The energy-constrained heuristic function considers path length, smoothness, ground friction, and energy consumption, while the pheromone update strategy incorporates multiple energy consumption metrics to guide ants toward energy-efficient solutions. For multi-robot coordination, PFACO implements both prioritized conflict-free strategy (PCS) and route conflict-free strategy (RCS) to ensure collision-free operation without requiring predefined robot priorities (Li et al., 2023). Particle Swarm Optimization (PSO) models collective behavior of bird flocks or fish schools, where particles (candidate solutions) move through the solution space influenced by their own best-known position and the global best position found by the swarm. Each particle updates its velocity and position according to:

$$v\_i(t+1) = w \cdot v\_i(t) + c_1 \cdot r_1 \cdot (p\_best,i - x\_i(t)) + c_2 \cdot r_2 \cdot (g\_best - x\_i(t)) \quad x\_i(t+1) = x\_i(t) + v\_i(t+1)$$

where v_i and x_i represent particle velocity and position, p_best,i represents the particle's personal best position, g_best represents the global best position, w is the inertia weight, c_1 and c_2 are acceleration coefficients, and r_1 and r_2 are random numbers (Nonoyama et al., 2022). PSO's simplicity and fast convergence make it attractive for real-time applications, though it may struggle with complex multi-modal landscapes without proper parameter tuning or hybrid enhancements. Hybrid approaches combining multiple algorithmic paradigms have demonstrated superior performance by leveraging complementary strengths. Zhang et al. (2023) proposed a hybrid method integrating deep reinforcement learning with multi-objective optimization, achieving improved path smoothness, reduced energy consumption, and shorter task completion times compared to traditional algorithms. The integration of learning-based methods with evolutionary optimization enables adaptive behavior that improves with experience while maintaining the systematic exploration capabilities of population-based search.

2.3 Energy Efficiency Optimization Methods

Energy efficiency optimization in industrial robotics requires accurate modeling of energy consumption mechanisms and effective optimization strategies that minimize energy expenditure while satisfying performance requirements. Robot energy consumption arises from multiple sources including motor electrical losses, mechanical friction, gravitational effects, and payload acceleration, with the relative contribution of each source varying based on robot configuration, trajectory characteristics, and operational conditions (Carabin et al., 2020). The fundamental energy consumption model for robot manipulators considers both kinetic and potential energy changes along with dissipative losses. The total energy consumption E_total can be expressed as:

$$E\_total = E\_kinetic + E\_potential + E\_friction + E\_motor$$

where E_kinetic represents kinetic energy changes due to link and payload acceleration, E_potential represents potential energy changes due to gravitational effects, E_friction represents energy dissipated through joint friction, and E_motor represents electrical losses in motor windings and drive electronics (Paryanto et al., 2015). Each component depends on trajectory parameters including joint positions, velocities, and accelerations, creating a complex optimization landscape where energy-optimal trajectories must balance multiple competing factors. Advanced energy models incorporate dynamic factors including payload variations, joint coupling effects, and regenerative braking capabilities. Gruszka, Miatliuk, and Szustakiewicz (2022) developed an energy-efficient approach for pick-and-place operations that accounts for payload mass variations throughout the task cycle, demonstrating that trajectory optimization considering dynamic payload changes achieves 12-18% energy savings compared to fixed-payload optimization. The model incorporates time-varying inertia matrices and gravitational torques that change as the robot picks up and releases objects, requiring optimization algorithms capable of handling time-dependent system dynamics. Nonoyama et al. (2022) proposed an integrated approach combining robot configuration optimization with motion planning using genetic algorithms and particle swarm optimization. Their method simultaneously optimizes robot base placement, joint configurations, and trajectory parameters to minimize energy consumption while satisfying task requirements. Experimental results demonstrated 23% energy reduction compared to conventional approaches that optimize configuration and motion separately, highlighting the importance of holistic optimization that considers interactions between different design decisions. Energy-optimal trajectory planning typically involves time-scaling optimization, where a geometric path is first generated and then parameterized by a time-scaling function that determines velocity profiles along the path. The time-scaling approach decouples geometric path planning from velocity optimization, enabling efficient optimization of velocity profiles subject to kinematic and dynamic constraints (Carabin et al., 2020). The optimal time-scaling function minimizes energy consumption while satisfying maximum velocity, acceleration, and jerk constraints that ensure smooth motion and prevent mechanical stress.

Recent research has explored the trade-offs between energy efficiency and other performance metrics. Jiang, Luo, Zhou, Yao, and Chen (2023) investigated multi-objective time-energy-impact optimization for robotic excavators, revealing that aggressive time minimization yields disproportionate energy penalties while moderate time increases enable substantial energy savings. Their results showed that accepting a 10% increase in cycle time enables 25-30% energy reduction, demonstrating the significant potential for energy savings through modest performance trade-offs in applications where cycle time is not the dominant constraint. The integration of energy-aware scheduling with path planning provides additional optimization opportunities at the system level. Rubio, Quiñones, Martinez, Diez, Ferrus, and Garcia (2021) developed a multi-objective optimization framework for autonomous industrial processes that simultaneously optimizes costs and energy efficiency, considering interactions between robot scheduling, path planning, and resource allocation. Their approach demonstrated that system-level optimization achieves superior results compared to component-level optimization, reducing total energy consumption by 18-22% while maintaining productivity targets through intelligent task scheduling and resource coordination.

2.4 Conflict Resolution in Multi-Robot Systems

Conflict resolution represents a critical challenge in multi-robot collaborative systems, where multiple robots operating in shared workspaces must coordinate movements to prevent collisions while optimizing global system performance. Traditional approaches assign priorities to robots and plan paths sequentially, with higher-priority robots planned first and lower-priority robots avoiding their paths (Larsen, Kim, Kupke, & Lozano-Pérez, 2017). While simple to implement, priority-based methods suffer from suboptimality since early planning decisions constrain later robots, and the choice of priority ordering significantly impacts solution quality. Priority-free approaches address these limitations by considering all robots simultaneously during path planning, enabling more balanced solutions that optimize global objectives rather than favoring specific robots. Li et al. (2023) proposed priority-free ant colony optimization (PFACO) that incorporates both prioritized conflict-free strategy (PCS) and route conflict-free strategy (RCS). PCS temporarily assigns priorities during path construction to resolve immediate conflicts, while RCS performs global conflict checking and resolution after initial paths are generated. This hybrid approach achieves conflict-free operation without requiring predefined priorities, demonstrating 21.6% energy reduction compared to traditional A while maintaining collision-free paths in complex multi-robot scenarios. Conflict detection mechanisms identify potential collisions by analyzing spatiotemporal relationships between robot trajectories. A conflict occurs when two robots occupy overlapping workspace regions at the same time, requiring either spatial separation (different paths) or temporal separation (different timing). Efficient conflict detection algorithms use bounding volume hierarchies and swept volume analysis to quickly identify potential conflicts without exhaustive pairwise checking of all trajectory points (Lin et al., 2022). The computational complexity of conflict detection grows quadratically with the number of robots, necessitating efficient algorithms and data structures for large-scale systems. Conflict resolution strategies modify robot paths or timing to eliminate detected conflicts while minimizing impact on optimization objectives. Spatial resolution changes path geometry to increase separation between robots, typically by adding waypoints or modifying existing waypoints to route robots around conflict regions. Temporal resolution adjusts robot velocities or introduces waiting periods to ensure robots pass through shared regions at different times, maintaining original path geometry while modifying timing. Hybrid approaches combine spatial and temporal resolution, providing greater flexibility to find high-quality conflict-free solutions (Chen, Huang, Zhu, & Zhao, 2021).

The computational complexity of multi-robot path planning grows exponentially with the number of robots, creating scalability challenges for large-scale systems. Decentralized and distributed planning approaches address scalability by decomposing the global problem into smaller subproblems solved by individual robots or robot groups. Each robot plans its path by considering only nearby robots within its communication radius, thereby reducing computational requirements while maintaining conflict-free operation through local coordination protocols (Lin et al., 2022). However, decentralized approaches may converge to suboptimal solutions compared to centralized planning, creating a trade-off between solution quality and computational scalability. Dynamic environments introduce additional complexity where obstacles and other robots may move unpredictably, requiring real-time replanning capabilities. Wang, Gao, Wei, and Fei (2018) developed adaptive replanning strategies that monitor environment changes and trigger replanning when significant deviations from planned trajectories occur. Their approach balances replanning frequency with computational overhead, avoiding excessive replanning that wastes computation while ensuring timely response to significant environment changes that could cause collisions or performance degradation.

3. Methodology

3.1 Problem Formulation and Mathematical Framework

The multi-objective path planning problem for industrial robot collaborative systems can be formally defined as finding optimal trajectories for N robots operating in a shared workspace $W \subset \mathbb{R}^3$, where each robot must move from an initial configuration $q_{i,start}$ to a goal configuration $q_{i,goal}$ while optimizing multiple objectives and satisfying kinematic, dynamic, and collision avoidance constraints. Let Q_i represent the configuration space of robot i, and let $\xi_i: [0, T_i] \rightarrow Q_i$ represent a trajectory parameterized by time, where T_i is the total execution time for robot i. The multi-robot path planning problem seeks to find trajectories $\xi = \{\xi_1, \xi_2, ..., \xi_N\}$ that minimize a vector of objective functions:

Minimize: $F(\xi) = [f_{time}(\xi), f_{energy}(\xi), f_{smoothness}(\xi), f_{safety}(\xi)]^T$

Subject to:

1. Kinematic constraints: $||\dot{q}_i(t)|| \leq v_{max,i}$, $||\ddot{q}_i(t)|| \leq a_{max,i}$
2. Dynamic constraints: $\tau_i(t) = M_i(q_i)\ddot{q}_i + C_i(q_i,\dot{q}_i)\dot{q}_i + G_i(q_i) \leq \tau_{max,i}$
3. Collision avoidance: $d(R_i(t), R_j(t)) \geq d_{min}$ for all $i \neq j$ and all t
4. Workspace constraints: $Forward\_Kinematics(q_i(t)) \in W$ for all i and all t
5. Boundary conditions: $q_i(0) = q_{i,start}$, $q_i(T_i) = q_{i,goal}$

where M_i represents the inertia matrix, C_i represents Coriolis and centrifugal terms, G_i represents gravitational torques, $R_i(t)$ represents the spatial region occupied by robot i at time t, and $d(\cdot,\cdot)$ represents a distance metric between robot regions (Zhao et al., 2020). The objective functions capture different performance aspects:

Time objective: $f\_time(\xi) = \max\{T_1, T_2, ..., T\_N\}$ represents the makespan (total completion time) for the multi-robot system, relevant for throughput-critical applications.

Energy objective: $f\_energy(\xi) = \Sigma_{i=1}^{N} \int_0^{T_i} P\_i(t)dt$ represents total energy consumption across all robots, where P_i(t) is the instantaneous power consumption of robot i.

Smoothness objective: $f\_smoothness(\xi) = \Sigma_{i=1}^{N} \int_0^{T_i} \|\dddot{q}\_i(t)\|^2 dt$ represents trajectory jerk (rate of change of acceleration), with lower values indicating smoother motion that reduces mechanical wear and vibration._

Safety objective: $f\_safety(\xi) = \min\{d(R\_i(t), R\_j(t)) : i \neq j, t \in [0,T]\}$ represents the minimum separation distance between any pair of robots, with larger values providing greater safety margins.

The Pareto optimality concept provides the solution framework for this multi-objective problem. A trajectory set ξ is Pareto-optimal if there exists no other feasible trajectory set ξ such that $f\_k(\xi) \leq f\_k(\xi)$ for all k and $f\_j(\xi) < f\_j(\xi)$ for at least one j. The set of all Pareto-optimal solutions forms the Pareto front in objective space, representing the trade-off surface between competing objectives (Wang et al., 2023).

3.2 Multi-Objective Optimization Algorithms

The implementation of NSGA-II for multi-robot path planning involves several algorithmic components working in concert to evolve high-quality Pareto-optimal solution sets. The algorithm maintains a population P_t of size N_pop at generation t, with each individual representing a complete multi-robot trajectory set encoded as a chromosome.

NSGA-II Algorithm for Multi-Robot Path Planning:

```
Initialize: Generate random population P₀ of size N_pop
Evaluate: Compute objective functions F(x) for all x ∈ P₀

For t = 1 to max_generations:
  // Generate offspring population
  Q_t = ∅
  While |Q_t| < N_pop:
    Select parents p₁, p₂ from P_{t-1} using binary tournament
    Apply crossover to generate offspring o₁, o₂
    Apply mutation to o₁, o₂
    Add o₁, o₂ to Q_t

  // Combine parent and offspring populations
  R_t = P_{t-1} ∪ Q_t
```

```
// Non-dominated sorting
F = Fast_Non_Dominated_Sort(R_t)

// Select next generation
P_t = ∅, i = 1
While |P_t| + |F_i| ≤ N_pop:
    Calculate crowding distance for solutions in F_i
    P_t = P_t ∪ F_i
    i = i + 1

// Fill remaining slots using crowding distance
Sort F_i by crowding distance in descending order
P_t = P_t ∪ F_i[1:(N_pop - |P_t|)]
```

Return: Non-dominated solutions from P_t

The fast non-dominated sorting procedure assigns each solution a rank based on Pareto dominance relationships. For each solution p, the algorithm computes n_p (number of solutions dominating p) and S_p (set of solutions dominated by p). Solutions with $n_p = 0$ form the first front F_1. Subsequent fronts are identified by iteratively removing solutions from F_i and decrementing domination counts, with solutions reaching $n_p = 0$ forming F_{i+1} (Jose & Pratihar, 2016). The crowding distance calculation estimates solution density in objective space to maintain diversity. For each objective k, solutions are sorted by f_k values, and boundary solutions receive infinite crowding distance. Interior solutions receive crowding distance:

$$CD_i = \Sigma_k (f_k(i+1) - f_k(i-1)) / (f_k^{max} - f_k^{min})$$

where $f_k(i+1)$ and $f_k(i-1)$ represent objective values of neighboring solutions in the sorted list. Solutions with larger crowding distances occupy less crowded regions and are preferred during selection (Hou et al., 2022).The priority-free ant colony optimization (PFACO) algorithm addresses multi-robot path planning through a two-stage approach. The first stage uses energy-constrained ant colony optimization (ECACO) to generate initial paths for each robot independently, incorporating energy-aware heuristic functions. The second stage applies conflict resolution strategies to eliminate collisions while preserving energy efficiency.

PFACO Algorithm Structure:

Initialize: Set pheromone $\tau_{ij} = \tau_0$ for all edges (i,j)

    Set parameters α, β, ρ, Q

```
For iteration = 1 to max_iterations:
  // Stage 1: Generate paths for each robot using ECACO
  For each robot r = 1 to N:
    For each ant k = 1 to N_ants:
      Construct path π_k^r using energy-constrained heuristic:
        η_ij = 1 / (w₁·d_ij + w₂·smooth_ij + w₃·energy_ij)
      Evaluate path quality Q_k^r

    Select best path π_best^r for robot r
    Update pheromones on edges in π_best^r

  // Stage 2: Conflict resolution
  Detect conflicts C = {(r_i, r_j, t) : collision between robots i,j at time t}

  While C ≠ ∅:
    Select conflict c = (r_i, r_j, t) from C

    // Apply prioritized conflict-free strategy (PCS)
    Assign temporary priority: p_i > p_j
    Replan path for robot r_j avoiding r_i's path

    // Apply route conflict-free strategy (RCS)
    If conflict persists:
      Adjust timing of lower-priority robot
      Insert waiting period or velocity reduction

    Update conflict set C

  // Pheromone evaporation
  τ_ij = (1-ρ)·τ_ij for all edges (i,j)

Return: Conflict-free paths {π₁, π₂, ..., π_N}
```

The energy-constrained heuristic function in ECACO considers multiple factors: d_ij represents Euclidean distance between nodes i and j, smooth_ij represents the angular change in path direction (lower values indicate smoother paths), and energy_ij estimates energy consumption for traversing edge (i,j) based on required velocities and accelerations. The weights w_1, w_2, w_3 balance these factors, with typical values $w_1 = 0.4$, $w_2 = 0.3$, $w_3 = 0.3$ providing good performance across diverse scenarios (Li et al., 2023).

3.3 Energy Consumption Modeling

Accurate energy consumption modeling is essential for effective energy optimization in robot path planning. The comprehensive energy model considers multiple components including motor electrical losses, mechanical friction, gravitational effects, and kinetic energy changes. The instantaneous power consumption for robot joint i can be expressed as:

$$P_i(t) = \tau_i(t)\cdot\omega_i(t) + P_{friction,i} + P_{electrical,i}$$

where $\tau_i(t)$ is the joint torque, $\omega_i(t)$ is the joint angular velocity, $P_{friction,i}$ represents friction losses, and $P_{electrical,i}$ represents electrical losses in motor windings and drive electronics (Carabin et al., 2020). The joint torque $\tau_i(t)$ is computed from the robot dynamics equation:

$$\tau(t) = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q})$$

where $M(q)$ is the n×n inertia matrix, $C(q,\dot{q})$ represents Coriolis and centrifugal effects, $G(q)$ represents gravitational torques, and $F(\dot{q})$ represents friction torques. Each component depends on the robot configuration q, velocities $\dot{q}$, and accelerations $\ddot{q}$, creating complex dependencies between trajectory parameters and energy consumption (Paryanto et al., 2015).

The total energy consumption over a trajectory is obtained by integrating instantaneous power:

$$E_{total} = \int_0^T \Sigma_{i=1}^n P_i(t)dt$$

For discrete trajectory representations with N waypoints, this integral is approximated using numerical integration:

$$E_{total} \approx \Sigma_{k=1}^{N-1} \Sigma_{i=1}^n P_i(t_k)\cdot\Delta t_k$$

where $\Delta t_k = t_{k+1} - t_k$ represents the time interval between consecutive waypoints (Nonoyama et al., 2022)._

Advanced energy models incorporate regenerative braking capabilities where motors can recover energy during deceleration phases. The recovered energy E_{regen} depends on motor efficiency η_{motor} and drive electronics efficiency η_{drive}:

$$E_{regen} = \eta_{motor}\cdot\eta_{drive}\cdot\int_{decel} |P_i(t)|dt$$

where the integral is taken over deceleration phases where $P_i(t) < 0$. Typical efficiency values are $\eta_{motor} = 0.85\text{-}0.92$ and $\eta_{drive} = 0.90\text{-}0.95$, enabling recovery of 75-87% of braking energy in modern servo systems (Gruszka et al., 2022).

Payload variations significantly impact energy consumption through changes in inertia and gravitational effects. For pick-and-place operations, the effective payload mass $m_{payload}(t)$ varies between m_{min} (empty gripper) and m_{max} (gripper with object). The time-varying inertia matrix becomes:

$$M(q,t) = M_{robot}(q) + M_{payload}(q,m_{payload}(t))$$

where $M_{robot}(q)$ represents the robot's intrinsic inertia and $M_{payload}(q,m_{payload}(t))$ represents the additional inertia due to payload. Optimization algorithms must account for these time-varying dynamics to achieve accurate energy predictions and optimal trajectory planning (Gruszka et al., 2022).

3.4 Conflict Resolution Strategies

Conflict resolution in multi-robot systems requires efficient detection of potential collisions and effective strategies to eliminate conflicts while minimizing impact on optimization objectives. The conflict detection process analyzes spatiotemporal relationships between robot trajectories to identify situations where robots occupy overlapping workspace regions simultaneously.

For robots i and j with trajectories $\xi_i(t)$ and $\xi_j(t)$, a conflict exists if:

$$d(R_i(t), R_j(t)) < d_{min} \text{ for some } t \in [0, \min(T_i, T_j)]$$

where $R_i(t)$ represents the spatial region occupied by robot i at time t, $d(\cdot,\cdot)$ is a distance metric, and d_{min} is the minimum safe separation distance (typically 0.1-0.3 meters for industrial robots) (Lin et al., 2022).

Efficient conflict detection uses bounding volume hierarchies to quickly eliminate robot pairs that cannot possibly collide. Each robot is enclosed in a simple bounding volume (sphere, axis-aligned bounding box, or oriented bounding box), and collision checking proceeds hierarchically:

1. **Coarse phase:** Check bounding volume overlap using simple geometric tests
2. **Intermediate phase:** If bounding volumes overlap, check link-level bounding volumes
3. **Fine phase:** If link bounding volumes overlap, perform detailed geometric collision checking

This hierarchical approach reduces computational complexity from $O(N^2 \cdot M^2)$ for exhaustive checking (N robots, M links per robot) to approximately $O(N^2 \cdot \log M)$ for typical industrial scenarios (Chen et al., 2021).

The prioritized conflict-free strategy (PCS) resolves conflicts by temporarily assigning priorities during path planning. When a conflict is detected between robots i and j, the algorithm assigns priority $p_i > p_j$ and replans robot j's path to avoid robot i's trajectory. The priority assignment can be based on various criteria:

- **Distance-based:** Assign higher priority to robots closer to their goals
- **Energy-based:** Assign higher priority to robots with more energy-efficient current paths
- **Task-based:** Assign higher priority to robots performing more critical tasks
- **Dynamic:** Reassign priorities adaptively based on current system state

The route conflict-free strategy (RCS) performs global conflict checking after initial paths are generated and applies coordinated modifications to eliminate all conflicts simultaneously. RCS formulates conflict resolution as a constraint satisfaction problem where path modifications must satisfy:

1. Eliminate all detected conflicts
2. Maintain path feasibility (kinematic and dynamic constraints)

3. Minimize deviation from original paths
4. Preserve energy efficiency to the extent possible

The optimization problem for RCS can be expressed as:

Minimize: $\Sigma_{i=1}^{N}$ w_path·||ξ_i - ξ_i,original||² + w_energy·(E_i - E_i,original)²

Subject to: d(R_i(t), R_j(t)) ≥ d_min for all i ≠ j, all t

where ξ_i,original and E_i,original represent the original path and energy consumption before conflict resolution, and w_path and w_energy are weighting factors balancing path preservation and energy efficiency (Li et al., 2023).

Temporal conflict resolution adjusts robot velocities or introduces waiting periods to separate robots in time rather than space. For a detected conflict at time t_conflict, temporal resolution can:

1. **Velocity scaling:** Reduce velocity of one robot to delay arrival at conflict location
2. **Waiting insertion:** Insert a waiting period before the conflict location
3. **Velocity profile optimization:** Optimize velocity profiles to minimize energy impact while resolving conflict

Temporal resolution preserves path geometry, which is advantageous when spatial modifications would significantly increase path length or energy consumption. However, temporal resolution may increase total cycle time, creating trade-offs between different optimization objectives (Chen et al., 2021).

4. Simulation Experiments

4.1 Experimental Setup and Configuration

The simulation experiments evaluate multi-objective path planning algorithms using a comprehensive testbed representing typical industrial robot collaborative scenarios. The experimental environment consists of a 10m × 10m workspace containing multiple industrial robots (6-DOF manipulators with 1.5m reach), static obstacles representing machinery and workstations, and dynamic obstacles simulating human workers or mobile equipment. Three distinct scenarios test algorithm performance under varying complexity levels:

Scenario 1 (Low Complexity): 3 robots, 5 static obstacles, sparse workspace (obstacle density 15%), simple pick-and-place tasks with 4 waypoints per robot.

Scenario 2 (Medium Complexity): 6 robots, 12 static obstacles, moderate workspace density (30%), assembly tasks with 8 waypoints per robot, including payload variations (0-5 kg).

Scenario 3 (High Complexity): 10 robots, 20 static obstacles, dense workspace (45%), complex manufacturing tasks with 12 waypoints per robot, dynamic obstacles (2 moving obstacles), and tight coordination requirements. Robot kinematic and dynamic parameters are based on industrial manipulators such as the ABB IRB 1600 and KUKA KR 6 R900, with maximum joint velocities of 150-180 deg/s, maximum accelerations of 300-400 deg/s², and maximum jerks of 3000-5000 deg/s³. Energy consumption parameters include motor efficiency $\eta_{motor} = 0.88$, drive efficiency $\eta_{drive} = 0.92$, and regenerative braking efficiency $\eta_{regen} = 0.80$ (Garriz, Domingo, & Arias, 2022).

Algorithm parameters are configured based on preliminary tuning experiments and literature recommendations:

NSGA-II parameters:

- Population size: $N_{pop} = 100$
- Maximum generations: 200
- Crossover probability: $p_c = 0.9$
- Mutation probability: $p_m = 0.1$
- Tournament size: 2

PFACO parameters:

- Number of ants: $N_{ants} = 50$
- Maximum iterations: 150
- Pheromone influence: $\alpha = 1.0$
- Heuristic influence: $\beta = 2.0$
- Evaporation rate: $\rho = 0.1$
- Pheromone deposit: $Q = 100$

PSO parameters:

- Swarm size: $N_{particles} = 80$
- Maximum iterations: 150
- Inertia weight: $w = 0.7$
- Cognitive coefficient: $c_1 = 1.5$
- Social coefficient: $c_2 = 1.5$

Each algorithm is executed 30 times per scenario with different random seeds to ensure statistical significance. Performance metrics are computed for each run, and statistical analysis (mean, standard deviation, confidence intervals) characterizes algorithm performance and variability (Pan et al., 2022).

4.2 Performance Metrics and Evaluation Criteria

Comprehensive performance evaluation requires metrics assessing multiple aspects of solution quality, including objective function values, Pareto front characteristics, computational efficiency, and robustness.

Objective Function Metrics:

1. **Cycle Time (T_{cycle}):** Maximum completion time across all robots, measured in seconds. Lower values indicate faster task completion and higher throughput.
2. **Energy Consumption (E_{total}):** Total energy consumed by all robots, measured in kilojoules (kJ). Lower values indicate more energy-efficient operation.

3. **Trajectory Smoothness (S_jerk):** Integrated squared jerk across all robot trajectories, measured in $(rad/s^3)^2 \cdot s$. Lower values indicate smoother motion with reduced mechanical stress.
4. **Safety Margin (d_min):** Minimum separation distance between any pair of robots during execution, measured in meters. Larger values provide greater safety margins.

Pareto Front Quality Metrics:

1. **Hypervolume (HV):** Volume of objective space dominated by the obtained solution set, normalized to [0,1]. Larger values indicate better convergence and diversity. Computed as:

$$HV = \text{Volume}(\{y \in \mathbb{R}^k : \exists x \in S, f_i(x) \leq y_i \leq r_i, \forall i\})$$

where S is the solution set, f_i are objective functions, and r_i are reference points (Pan et al., 2022).

2. **Inverted Generational Distance (IGD):** Average distance from true Pareto front points to nearest obtained solution. Lower values indicate better convergence. Computed as:

$$IGD = (1/|P|) \Sigma_{p \in P} \min_{s \in S} ||f(p) - f(s)||$$

where P is a reference set approximating the true Pareto front.

3. **Spacing (SP):** Standard deviation of distances between consecutive solutions along the Pareto front. Lower values indicate more uniform distribution. Computed as:

$$SP = \sqrt{[(1/|S|) \Sigma_{i=1}^{|S|} (d_i - \bar{d})^2]}$$

where d_i is the distance from solution i to its nearest neighbor and $\bar{d}$ is the mean distance._

Computational Efficiency Metrics:

1. **Computation Time (T_comp):** Total wall-clock time required to generate the Pareto front, measured in seconds.
2. **Convergence Rate:** Number of iterations required to reach 95% of final hypervolume value, indicating algorithm convergence speed.
3. **Success Rate:** Percentage of runs producing feasible conflict-free solutions, assessing algorithm robustness.

4.3 Comparative Analysis of Algorithms

The comparative analysis evaluates five algorithms: NSGA-II, NSGA-III, PFACO, PSO, and a hybrid approach combining PFACO with local search refinement (PFACO-LS). Each algorithm is assessed across all three scenarios using the metrics defined above.

Baseline Comparison: Traditional single-objective A path planning with sequential priority-based conflict resolution serves as the baseline. A optimizes path length for each robot independently, then applies priority-based conflict resolution where higher-priority robots maintain their paths and lower-priority robots replan to avoid conflicts. This baseline represents conventional industrial practice and provides context for evaluating multi-objective approaches.

Algorithm Variants: To assess the impact of specific algorithmic components, we evaluate several variants:

- **NSGA-II-basic:** Standard NSGA-II without problem-specific enhancements
- **NSGA-II-enhanced:** NSGA-II with custom crossover and mutation operators designed for robot path planning
- **PFACO-PCS:** PFACO using only prioritized conflict-free strategy
- **PFACO-RCS:** PFACO using only route conflict-free strategy
- **PFACO-full:** PFACO using both PCS and RCS (Li et al., 2023)

This variant analysis isolates the contribution of specific algorithmic components and identifies which enhancements provide the greatest performance improvements.

4.4 Quantitative Results and Statistical Analysis

Scenario 1 Results (Low Complexity):

The low-complexity scenario with 3 robots demonstrates clear performance differences between algorithms. PFACO-full achieves the best overall performance with mean cycle time $T_{cycle} = 18.4 \pm 1.2$ seconds, energy consumption $E_{total} = 42.3 \pm 2.8$ kJ, and smoothness $S_{jerk} = 156 \pm 18$ $(rad/s^3)^2 \cdot s$. Compared to the A baseline ($T_{cycle} = 16.2 \pm 0.8$ s, $E_{total} = 54.1 \pm 3.2$ kJ, $S_{jerk} = 248 \pm 31$), PFACO-full achieves 21.8% energy reduction and 37.1% smoothness improvement with only 13.6% cycle time increase. NSGA-II-enhanced produces diverse Pareto fronts with hypervolume $HV = 0.847 \pm 0.023$, significantly outperforming NSGA-II-basic ($HV = 0.762 \pm 0.041$, $p < 0.01$ by t-test). The enhanced crossover operator that preserves waypoint ordering and the mutation operator that performs local path refinement contribute substantially to improved performance. PSO converges faster (convergence rate = 42 ± 8 iterations) than NSGA-II (convergence rate = 78 ± 12 iterations) but achieves slightly lower hypervolume ($HV = 0.821 \pm 0.028$), indicating faster but less thorough exploration of the solution space.

Scenario 2 Results (Medium Complexity):

The medium-complexity scenario with 6 robots and payload variations reveals more pronounced performance differences. PFACO-full maintains strong performance with $T_{cycle} = 34.7 \pm 2.1$ s, $E_{total} = 128.4 \pm 8.6$ kJ, and $S_{jerk} = 412 \pm 47$ $(rad/s^3)^2 \cdot s$, achieving 19.3% energy reduction compared to A baseline ($E_{total} = 159.2 \pm 11.4$ kJ) with 16.8% cycle time increase. The energy savings are particularly significant during payload transport phases, where PFACO's energy-aware heuristic optimizes velocity profiles to minimize acceleration energy while accounting for time-varying inertia. NSGA-III demonstrates advantages over NSGA-II in this scenario with four objectives (time, energy, smoothness, safety). NSGA-III achieves hypervolume HV = 0.793 ± 0.031 compared to NSGA-II's HV = 0.741 ± 0.038 ($p < 0.01$), with better solution distribution indicated by lower spacing metric (SP = 0.042 ± 0.008 vs. SP = 0.067 ± 0.013). The reference point-based selection in NSGA-III maintains better diversity in four-dimensional objective space where crowding distance becomes less effective. The hybrid PFACO-LS approach combining PFACO with local search refinement achieves the best results: HV = 0.831 ± 0.026, $E_{total} = 121.7 \pm 7.9$ kJ (23.5% reduction vs. A), and $S_{jerk} = 387 \pm 41$ $(rad/s^3)^2 \cdot s$. The local search phase applies gradient-based optimization to refine velocity profiles along paths generated by PFACO, achieving additional 5.2% energy savings with minimal computational overhead ($T_{comp} = 47.3 \pm 5.2$ s vs. 42.8 ± 4.6 s for PFACO-full).

Scenario 3 Results (High Complexity):

The high-complexity scenario with 10 robots, dense obstacles, and dynamic constraints challenges all algorithms. PFACO-full maintains feasibility with 96.7% success rate (29/30 runs producing conflict-free solutions) compared to NSGA-II's 83.3% success rate (25/30 runs) and PSO's 76.7% success rate (23/30 runs). The superior success rate demonstrates PFACO's robust conflict resolution mechanisms that effectively handle complex multi-robot coordination. Among successful runs, PFACO-full achieves $T_{cycle} = 58.3 \pm 4.7$ s, $E_{total} = 287.4 \pm 21.3$ kJ, and $S_{jerk} = 892 \pm 97$ $(rad/s^3)^2 \cdot s$, representing 17.8% energy reduction compared to A baseline ($E_{total} = 349.7 \pm 26.8$ kJ) with 19.4% cycle time increase. The energy savings are consistent across complexity levels, demonstrating that PFACO's energy-aware optimization scales effectively to larger systems. Computational time increases substantially with scenario complexity: Scenario 1 requires $T_{comp} = 12.4 \pm 1.8$ s (PFACO-full), Scenario 2 requires $T_{comp} = 42.8 \pm 4.6$ s, and Scenario 3 requires $T_{comp} = 156.3 \pm 18.7$ s. The approximately quadratic scaling ($T_{comp} \propto N^2$) reflects the quadratic growth in conflict checking requirements as robot count increases. For real-time applications requiring replanning within 1-2 seconds, this computational demand necessitates either distributed planning architectures or approximate algorithms trading solution quality for speed (Zhang et al., 2023).

Trade-off Analysis:

Detailed analysis of Pareto fronts reveals important trade-offs between objectives. In Scenario 2, solutions optimizing cycle time aggressively ($T_{cycle} = 29.7$ s, 14.4% reduction vs. PFACO-full mean) incur substantial energy penalties ($E_{total} = 173.2$ kJ, 34.9% increase), demonstrating that aggressive time optimization yields disproportionate energy costs. The energy-time trade-off curve shows approximately quadratic relationship: reducing cycle time by x% increases energy consumption by approximately 2.5x%. The smoothness-time trade-off is more favorable: improving smoothness by 20% ($S_{jerk} = 330$ vs. 412) requires only 8.3% cycle time increase ($T_{cycle} = 37.6$ s vs. 34.7 s), suggesting that moderate smoothness improvements can be achieved with acceptable time penalties. This finding has practical implications for applications where mechanical wear and vibration are concerns, indicating that smoothness optimization provides good return on investment in terms of equipment longevity versus productivity impact. Safety margin analysis shows that all algorithms maintain $d_{min} \geq 0.15$ m in Scenarios 1-2, well above the minimum threshold of 0.10 m. In Scenario 3, PFACO-full maintains $d_{min} = 0.127 \pm 0.018$ m while A baseline achieves $d_{min} = 0.112 \pm 0.023$ m, demonstrating that multi-objective optimization can improve safety margins while simultaneously optimizing other objectives. The improved safety margins result from PFACO's global conflict resolution that considers all robot interactions rather than sequential priority-based resolution that may create tight clearances.

Statistical Significance:

Pairwise statistical comparisons using Wilcoxon signed-rank tests (non-parametric alternative to paired t-test) confirm significant performance differences. PFACO-full vs. A baseline shows $p < 0.001$ for energy consumption across all scenarios, $p < 0.001$ for smoothness, and $p < 0.05$ for safety margin. PFACO-full vs. NSGA-II shows $p < 0.01$ for energy consumption and $p < 0.05$ for success rate in Scenario 3. These results provide strong statistical evidence that the observed performance improvements are not due to random variation.

5. Discussion

5.1 Comparative Analysis of Methods

The experimental results demonstrate that multi-objective optimization approaches significantly outperform traditional single-objective methods across diverse performance metrics. PFACO-full emerges as the most effective algorithm for industrial robot collaborative systems, achieving consistent energy reductions of 17.8-21.8% compared to A baseline while maintaining high success rates (96.7% in complex scenarios) and producing smooth, safe trajectories. The algorithm's strength derives from its energy-constrained heuristic function that guides path construction toward energy-efficient solutions and its dual conflict resolution mechanisms (PCS and RCS) that ensure collision-free operation without sacrificing optimization quality (Li et al., 2023). NSGA-II and NSGA-III provide valuable alternatives when diverse Pareto-optimal solution sets are required for decision-making. These evolutionary algorithms excel at exploring the trade-off space between competing objectives, generating well-distributed Pareto fronts that enable informed selection of operating points based on current priorities. NSGA-III demonstrates particular advantages in many-objective scenarios (four or more objectives) where traditional Pareto dominance becomes less effective, maintaining better solution diversity through reference point-based selection (Wang et al., 2023). However, evolutionary algorithms require careful parameter tuning and longer computation times compared to PFACO, limiting their applicability in time-critical scenarios requiring rapid replanning. PSO offers fast convergence and simple implementation but achieves slightly lower solution quality compared to PFACO and NSGA-II. The algorithm's rapid convergence (42 iterations vs. 78 for NSGA-II in Scenario 1) makes it attractive for applications requiring quick initial solutions, though the reduced exploration may miss high-quality solutions in complex landscapes. Hybrid approaches combining PSO's fast convergence with local refinement or population diversity mechanisms could potentially achieve better balance between speed and quality (Nonoyama et al., 2022). The hybrid PFACO-LS approach demonstrates that combining complementary algorithmic paradigms yields superior results. The global exploration capabilities of PFACO identify promising regions of the solution space, while local search refinement optimizes velocity profiles to extract additional energy savings. This two-stage approach achieves 23.5% energy reduction in Scenario 2, representing 5.2% improvement over PFACO alone with minimal computational overhead. The success of hybrid methods suggests that future research should explore additional combinations leveraging the strengths of different optimization paradigms (Zhang et al., 2023).

5.2 Trade-offs and Pareto Optimality

The Pareto front analysis reveals fundamental trade-offs between competing objectives that characterize industrial robot path planning. The energy-time trade-off exhibits approximately quadratic relationship, where reducing cycle time by x% increases energy consumption by approximately 2.5x%. This nonlinear relationship implies that aggressive time optimization yields diminishing returns while incurring disproportionate energy penalties. For applications where energy costs are significant (e.g., high-volume manufacturing with continuous operation), accepting modest cycle time increases of 10-15% enables substantial energy savings of 20-30%, providing attractive return on investment through reduced operational costs (Jiang et al., 2023). The smoothness-time trade-off is more favorable, with approximately linear relationship where improving smoothness by y% requires roughly 0.4y% cycle time increase. This favorable trade-off suggests that smoothness optimization provides good value for applications concerned with mechanical wear, vibration, or product quality. Reducing jerk by 20% requires only 8% cycle time increase, a trade-off that many applications would find acceptable given the benefits of reduced maintenance costs and improved equipment longevity (Garriz et al., 2022). The safety-performance trade-off shows that increased safety margins can be achieved with minimal performance penalties when using effective multi-objective optimization. PFACO-full improves minimum separation distance by 13.4% compared to A baseline while simultaneously reducing energy consumption, demonstrating that global optimization considering all robot interactions produces inherently safer solutions than sequential priority-based planning. This finding has important implications for human-robot collaboration scenarios where safety is paramount, suggesting that multi-objective optimization can enhance safety without sacrificing productivity (Carabin et al., 2020).

The shape and extent of Pareto fronts vary significantly across scenarios, reflecting different optimization landscapes. Scenario 1 (low complexity) produces convex Pareto fronts with smooth trade-off curves, indicating that intermediate solutions provide good balance between objectives. Scenario 3 (high complexity) produces more irregular Pareto fronts with discrete jumps, suggesting that certain objective combinations are difficult to achieve simultaneously due to conflicting constraints. Understanding these landscape characteristics helps practitioners set realistic expectations and select appropriate operating points based on application requirements.

5.3 Practical Implementation Challenges

Despite promising experimental results, several challenges must be addressed for successful deployment of multi-objective path planning in industrial environments. Computational requirements pose significant challenges, particularly for large-scale systems or applications requiring real-time replanning. The approximately quadratic scaling of computation time with robot count ($T_{comp} \propto N^2$) limits scalability, with 10-robot scenarios requiring 2-3 minutes for planning. Real-time applications requiring replanning within 1-2 seconds necessitate either distributed planning architectures that decompose the global problem or approximate algorithms that trade solution quality for speed (Lin et al., 2022). Model accuracy significantly impacts optimization effectiveness. Energy consumption models rely on accurate dynamic parameters including inertia matrices, friction coefficients, and motor efficiencies. Parameter uncertainties of 10-15% (typical in industrial practice) can cause actual energy consumption to deviate from predictions by 15-25%, potentially negating optimization benefits. Robust optimization approaches that account for parameter uncertainties or online learning methods that adapt models based on measured performance could improve practical effectiveness (Rubio et al., 2021). Dynamic environments introduce additional complexity where obstacles and other robots may move unpredictably, requiring real-time replanning capabilities. The experimental scenarios assume static obstacles and deterministic robot motions, but real industrial environments include human workers, mobile equipment, and process variations that create dynamic constraints. Adaptive replanning strategies that monitor environment changes and trigger replanning when significant deviations occur are essential for practical deployment, but must balance replanning frequency with computational overhead to avoid excessive computation that degrades real-time performance (Wang et al., 2018). Human-robot collaboration introduces safety and ergonomic considerations beyond those addressed in the current work. When humans work alongside robots, safety requirements become more stringent, requiring larger separation distances (typically 0.5-1.0 m vs. 0.1-0.3 m for robot-robot separation) and conservative velocity limits near human workers. Ergonomic considerations such as minimizing human walking distances, avoiding awkward postures, and balancing workload across human workers add additional objectives to the optimization problem. Integrating these human-centric objectives while maintaining computational tractability represents an important direction for future research (Chen et al., 2021).

5.4 Limitations and Constraints

The current work has several limitations that should be acknowledged. The simulation-based evaluation provides controlled conditions for systematic algorithm comparison but does not capture all complexities of real industrial environments. Hardware experiments with physical robots would provide more realistic assessment of practical performance, including effects of sensor noise, actuator dynamics, communication delays, and unexpected disturbances. The gap between simulation and reality can be substantial, with simulation-optimized solutions sometimes performing poorly on real hardware due to unmodeled effects (Carabin et al., 2020). The energy consumption models, while comprehensive, make simplifying assumptions including constant motor efficiency, negligible transmission losses, and perfect regenerative braking. Real systems exhibit efficiency variations with operating conditions, transmission losses in gearboxes and belts, and imperfect energy recovery during braking. More detailed models incorporating these effects would improve prediction accuracy but increase computational complexity, creating trade-offs between model fidelity and computational tractability (Gruszka et al., 2022).

The conflict resolution mechanisms assume perfect knowledge of robot positions and deterministic motion execution. Real systems face localization uncertainties, trajectory tracking errors, and timing variations that can cause actual robot positions to deviate from planned trajectories. Robust conflict resolution that accounts for these uncertainties through safety margins or probabilistic collision checking would improve reliability but may reduce optimization quality by requiring more conservative solutions (Lin et al., 2022). The experimental scenarios, while representative of common industrial applications, do not cover the full diversity of real-world requirements. Specialized applications such as high-speed pick-and-place, precision assembly, or heavy payload manipulation may have different optimization priorities and constraints. The relative importance of objectives varies across applications, with some prioritizing throughput, others emphasizing energy efficiency, and still others focusing on quality or safety. Developing application-specific optimization strategies tailored to particular industrial domains represents an important direction for future work (Hou et al., 2022).

6. Future Directions and Recommendations

Several promising research directions emerge from this work that could advance the state-of-the-art in multi-objective robot path planning. Distributed and decentralized planning architectures offer potential solutions to scalability challenges by decomposing the global optimization problem into smaller subproblems solved by individual robots or robot groups. Each robot would plan its path considering only nearby robots within a communication radius, reducing computational requirements from $O(N^2)$ to $O(N \cdot k)$ where k is the average number of neighbors. Distributed approaches must address coordination challenges to ensure global optimality and conflict-free operation, potentially through consensus algorithms or market-based coordination mechanisms (Lin et al., 2022). Learning-based approaches integrating reinforcement learning or imitation learning with multi-objective optimization could enable adaptive behavior that improves with experience. Robots could learn from past planning episodes to identify effective strategies for different scenarios, reducing computation time by focusing search on promising regions of the solution space. Transfer learning could enable knowledge sharing across different robot systems or applications, accelerating deployment in new environments. However, learning-based methods must address safety and reliability concerns, ensuring that learned behaviors maintain collision-free operation and satisfy performance requirements even in novel situations (Zhang et al., 2023). Uncertainty-aware optimization that explicitly accounts for parameter uncertainties, localization errors, and environmental variations would improve robustness in real-world deployments. Robust optimization formulations that optimize worst-case performance or probabilistic constraints that limit the probability of constraint violations could provide safety guarantees despite uncertainties. Stochastic optimization methods that sample from uncertainty distributions during planning could generate solutions that perform well across a range of possible conditions rather than optimizing for nominal conditions that may not reflect reality (Rubio et al., 2021).

Integration with higher-level planning and scheduling systems would enable system-level optimization that considers interactions between path planning, task allocation, and resource scheduling. Current approaches typically assume fixed task assignments and optimize paths for given tasks, but joint optimization of task allocation and path planning could achieve better global performance. Multi-level optimization frameworks that coordinate decisions across different planning horizons (strategic, tactical, operational) could balance long-term objectives with short-term constraints, improving overall system efficiency (Chen et al., 2021). Hardware validation through extensive experiments with physical robot systems is essential to bridge the gap between simulation and reality. Systematic comparison of simulation predictions with measured hardware performance would identify modeling errors and guide model refinement. Hardware experiments would also reveal practical implementation challenges not apparent in simulation, such as communication delays, sensor noise effects, and actuator limitations. Developing standardized benchmarks and datasets for multi-robot path planning would facilitate reproducible research and enable fair comparison of different approaches across research groups (Carabin et al., 2020). Energy-aware scheduling frameworks that coordinate robot operations to minimize total system energy consumption represent an important direction for sustainable industrial automation. Scheduling decisions such as task sequencing, robot assignment, and operation timing significantly impact energy consumption through effects on robot utilization, idle time, and coordination requirements. Integrated optimization of scheduling and path planning could achieve substantial energy savings beyond those possible through path optimization alone, contributing to corporate sustainability goals and reducing operational costs (Rubio et al., 2021).

7. Conclusion

This paper has presented a comprehensive analysis of multi-objective path planning and energy efficiency optimization methods for industrial robot collaborative systems. Through systematic review of theoretical foundations, detailed algorithmic analysis, and extensive simulation experiments, we have demonstrated that multi-objective optimization approaches significantly outperform traditional single-objective methods across multiple performance dimensions. Key findings indicate that priority-free ant colony optimization (PFACO) achieves superior performance for industrial robot collaborative systems, reducing energy consumption by 17.8-21.8% compared to traditional A path planning while maintaining high success rates (96.7% in complex scenarios) and producing smooth, safe trajectories. The algorithm's energy-constrained heuristic function and dual conflict resolution mechanisms (prioritized conflict-free strategy and route conflict-free strategy) enable effective multi-robot coordination with reasonable computational demands. Hybrid approaches combining PFACO with local search refinement achieve even better results (23.5% energy reduction), demonstrating the value of integrating complementary optimization paradigms. Evolutionary algorithms including NSGA-II and NSGA-III provide valuable alternatives when diverse Pareto-optimal solution sets are required for decision-making. These algorithms excel at exploring trade-off spaces between competing objectives, generating well-distributed Pareto fronts that enable informed selection of operating points based on current priorities. NSGA-III demonstrates particular advantages in many-objective scenarios with four or more objectives, maintaining better solution diversity through reference point-based selection. The comprehensive analysis of trade-offs reveals important relationships between competing objectives. The energy-time trade-off exhibits approximately quadratic relationship, where aggressive time optimization yields disproportionate energy penalties. Accepting modest cycle time increases of 10-15% enables substantial energy savings of 20-30%, providing attractive return on investment for energy-intensive applications. The smoothness-time trade-off is more favorable, with approximately linear relationship enabling significant smoothness improvements with acceptable time penalties. Multi-objective optimization can improve safety margins while simultaneously optimizing other objectives, demonstrating that global optimization produces inherently safer solutions than sequential priority-based planning. Several challenges remain for practical deployment, including computational scalability for large-scale systems, model accuracy under parameter uncertainties, real-time adaptability in dynamic environments, and integration of human-robot collaboration constraints. Addressing these challenges requires continued research into distributed planning architectures, learning-based adaptive methods, uncertainty-aware optimization, and system-level integration with scheduling and task allocation. Future research directions emphasize hardware validation through extensive experiments with physical robot systems, development of standardized benchmarks for reproducible research, integration of learning-based methods for adaptive behavior, and energy-aware scheduling frameworks for sustainable industrial automation. By addressing current limitations and pursuing these promising directions, multi-objective path planning methods will enable more efficient, sustainable, and capable industrial robot collaborative systems aligned with Industry 4.0 and 5.0 paradigms. The field of multi-objective robot path planning continues to evolve rapidly, driven by increasing industrial automation demands and advancing computational capabilities. As manufacturing systems become more complex and sustainability concerns intensify, the importance of effective multi-objective optimization will only grow. This work provides a foundation for understanding current capabilities, identifying critical challenges, and charting paths forward toward more intelligent, efficient, and sustainable industrial robot collaborative systems.

References

Alam, W., Mehmood, A., Iqbal, J., Asghar, F., & Ali, M. A. (2023). A novel sampling-based optimal motion planning algorithm for energy-efficient robotic pick and place. *Energies*, 16(19), 6910. https://doi.org/10.3390/en16196910

Carabin, G., Wehrle, E., & Vidoni, R. (2020). On the trajectory planning for energy efficiency in industrial robotic systems. *Robotics*, 9(4), 89. https://doi.org/10.3390/ROBOTICS9040089

Chen, X., Huang, H., Zhu, Z., & Zhao, Y. (2021). Design and collaborative operation of multimobile inspection robots in smart microgrids. *Complexity*, 2021, 6695688. https://doi.org/10.1155/2021/6695688

Garriz, C., Domingo, R., & Arias, J. M. (2022). Trajectory optimization in terms of energy and performance of an industrial robot in the manufacturing industry. *Sensors*, 22(19), 7538. https://doi.org/10.3390/s22197538

Gruszka, A., Miatliuk, K., & Szustakiewicz, K. (2022). A new energy-efficient approach to planning pick-and-place operations. *Energies*, 15(23), 8795. https://doi.org/10.3390/en15238795

Heredia, F. J., Rodríguez, A., & Camacho, E. F. (2023). Going green with lightweight robots: Energy optimal programming of lightweight robots. In *Proceedings of the 2023 International Conference on Advanced Robotics* (pp. 1-6). IEEE. https://doi.org/10.1109/icar58858.2023.10406598

Hou, Y., Xie, Y., Liu, Z., & Yin, L. (2022). Multi-objective optimal trajectory planning of customized industrial robot based on reliable dynamic identification for improving control accuracy. *Industrial Robot: An International Journal*, 49(6), 1120-1134. https://doi.org/10.1108/ir-12-2021-0301

Jiang, H., Luo, S., Zhou, Y., Yao, J., & Chen, G. (2023). Multi-objective time-energy-impact optimization for robotic excavator trajectory planning. *Automation in Construction*, 156, 105094. https://doi.org/10.1016/j.autcon.2023.105094

Jose, K., & Pratihar, D. K. (2016). Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods. *Robotics and Autonomous Systems*, 80, 34-42. https://doi.org/10.1016/j.robot.2016.02.003

Lan, J., Xie, Y., Liu, G., Cao, M., & Chen, D. (2020). A multi-objective trajectory planning method for collaborative robot. *Electronics*, 9(5), 859. https://doi.org/10.3390/electronics9050859

Larsen, E., Kim, S., Kupke, M., & Lozano-Pérez, T. (2017). Efficient computation of time-optimal multi-robot paths with temporal constraints. In *Proceedings of the 2017 IEEE International Conference on Robotics and Automation* (pp. 5050-5057). IEEE. https://doi.org/10.1109/ICRA.2017.7989590

Li, J., Xie, Y., Liu, G., Cao, M., & Chen, D. (2023). Conflict-free and energy-efficient path planning for multi-robots based on priority free ant colony optimization. *Mathematical Biosciences and Engineering*, 20(2), 3295-3320. https://doi.org/10.3934/mbe.2023165

Lin, S., Zheng, C., Xu, W., & Zhan, J. (2022). A review of path-planning approaches for multiple mobile robots. *Machines*, 10(9), 773. https://doi.org/10.3390/machines10090773

Nazarahari, M., Khanmirza, E., & Doostie, S. (2019). Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Systems with Applications*, 115, 106-120. https://doi.org/10.1016/j.eswa.2018.08.008

Nonoyama, K., Liu, Z., Fujiwara, T., Alam, M. M., & Nishi, T. (2022). Energy-efficient robot configuration and motion planning using genetic algorithm and particle swarm optimization. *Energies*, 15(6), 2074. https://doi.org/10.3390/en15062074

Pan, L., Li, L., Cheng, R., He, C., & Tan, K. C. (2022). A robust performance evaluation approach for solution preservation in multiobjective optimization. *Complex & Intelligent Systems*, 9(2), 2153-2170. https://doi.org/10.1007/s40747-022-00889-1

Paryanto, Brossog, M., Bornschlegl, M., & Franke, J. (2015). Reducing the energy consumption of industrial robots in manufacturing systems. *International Journal of Advanced Manufacturing Technology*, 78(5-8), 1315-1328. https://doi.org/10.1007/s00170-014-6737-z

Rubio, F., Quiñones, A., Martinez, A., Diez, J. L., Ferrus, L., & Garcia, F. (2021). Multi-objective optimization of costs and energy efficiency associated with autonomous industrial processes for sustainable growth. *Technological Forecasting and Social Change*, 173, 121115. https://doi.org/10.1016/J.TECHFORE.2021.121115

Soori, M., Arezoo, B., & Dastres, R. (2023). Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive Robotics*, 3, 54-70. https://doi.org/10.1016/j.cogr.2023.04.001

Tang, X. (2023). Multi-robot material delivery in industrial parks based improved on A algorithm. *Highlights in Science, Engineering and Technology*, 46, 7715. https://doi.org/10.54097/hset.v46i.7715

Wang, H., Gao, S., Wei, J., & Fei, Y. (2018). Optimal trajectory planning of free-floating space manipulator using differential evolution algorithm. *Advances in Space Research*, 61(6), 1525-1536. https://doi.org/10.1016/j.asr.2018.01.011

Wang, Y., Gao, S., Wei, J., & Fei, Y. (2023). Multi-objective path planning for space robot based on chaos immune particle swarm optimization algorithm. *International Journal of Advanced Robotic Systems*, 20(1), 1-15. https://doi.org/10.1177/17298806231151072

Yang, S., Xu, Z., & Wang, J. (2021). A novel maximin-based multi-objective evolutionary algorithm using one-by-one update scheme for multi-robot scheduling optimization. *IEEE Access*, 9, 113214-113230. https://doi.org/10.1109/ACCESS.2021.3105102

Zhang, Y., Xu, W., & Wang, J. (2023). Multi-objective optimal trajectory planning for robotic arms using deep reinforcement learning. *Sensors*, 23(13), 5974. https://doi.org/10.3390/s23135974

Zhao, J., Gao, S., Huang, X., & Tang, L. (2020). Multi-objective optimization of parameters in milling of aluminum alloy with carbide tool. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 234(13), 1603-1614. https://doi.org/10.1177/0954405420933104

Open Access Statement