

Article

Multi-Objective Optimization of Interaction Forces in Cooperative Dual-Arm Robotic Manipulation

Mario Peñacoba-Yagüe ^{1,*}, Jesús-Enrique Sierra-García ^{1,*} and Matilde Santos-Peñas ²

¹ Department of Digitalization, University of Burgos, 09001 Burgos, Spain

² Institute of Knowledge Technology, University Complutense of Madrid, 28040 Madrid, Spain; msantos@ucm.es

* Correspondence: mpenacoba@ubu.es (M.P.-Y.); jesierra@ubu.es (J.-E.S.-G.)

Abstract

This paper addresses the multi-objective optimization of cooperative dual-arm robotic manipulation, focusing on the reduction and balancing of interaction forces during the coordinated transport of a shared payload. The manipulation task is formulated from an object-centric perspective, where candidate trajectories are defined through intermediate object poses that are simultaneously mapped to both robotic manipulators under rigid grasping assumptions. Within this framework, the optimization problem is posed as a constrained multi-objective search in which the force demands associated with each robot are minimized while preserving kinematic feasibility and collision-free cooperative motion. Two representative population-based multi-objective algorithms, Multi-Objective Particle Swarm Optimization (MOPSO) and Non-dominated Sorting Genetic Algorithm II (NSGA-II), are evaluated under equivalent trajectory bounds and objective definitions. The results provide a set of non-dominated cooperative trajectories that support the selection of force-efficient motions with lower peak demands and improved load-sharing behavior. The comparative analysis demonstrates the potential of multi-objective metaheuristic optimization for force-aware dual-arm manipulation and highlights the different convergence and solution-distribution behaviors of MOPSO and NSGA-II in a constrained robotic manipulation scenario.

Keywords: multi-objective optimization; cooperative dual arm manipulation; energy-efficient path planning; MOPSO; NSGA-II; robotic trajectory optimization; multiple robots

Received: 24 June 2026

Revised: 16 July 2026

Accepted: 23 July 2026

Published: 24 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Recent advances in robotic manipulation have renewed interest in cooperative dual-arm systems, especially for tasks in which objects must be handled, transported, or reoriented in ways that are difficult to achieve with a single robotic arm [1]. By coordinating two manipulators around a shared payload, dual-arm systems can improve stability, increase the admissible payload range, distribute mechanical effort, and enable more dexterous object motions in constrained workspaces [2]. These advantages are especially relevant in industrial scenarios where bulky, heavy, or irregular objects must be moved safely while maintaining a stable grasp and avoiding collisions with surrounding equipment [3].

However, cooperative manipulation also introduces additional planning and control challenges. Unlike single-arm manipulation, the motion of both robots is strongly coupled through the transported object [4]. A candidate trajectory must therefore be simultaneously feasible for two kinematic chains, must preserve the relative grasp configuration, and must avoid collisions between the robots, the payload, and the environment [5,6]. In addition to these geometric constraints, cooperative execution must also consider the mechanical interaction between the manipulators and the object. Poorly planned motions may lead to unbalanced force distributions, excessive effort in one of the arms, or large interaction forces that can compromise safety, reduce execution robustness, and increase mechanical stress on the robotic system [7,8].

Most trajectory-planning approaches for robotic manipulation focus primarily on geometric feasibility, collision avoidance, or execution time. Although these aspects are essential, they do not fully capture the physical quality of a cooperative manipulation task [9]. In dual-arm systems, a trajectory that is collision-free and kinematically feasible may still be undesirable if it generates high force demands or if the load is unevenly distributed between both manipulators [10]. This is particularly important when the payload is transported under rigid grasping assumptions, since the motion of the object directly determines the force requirements at the end effectors [11]. Therefore, incorporating interaction-force-related criteria into the trajectory generation process is a necessary step toward safer and more mechanically efficient cooperative manipulation.

From an optimization perspective, force-aware cooperative manipulation is naturally a multi-objective problem [12]. Reducing the force required by one manipulator may increase the effort demanded from the other, and improving the balance between both arms may require a different shape from the one obtained by optimizing a single scalar objective. Consequently, aggregating all criteria into a weighted cost function can hide relevant trade-offs and make the result highly dependent on manually selected weights [13]. Multi-objective optimization offers a more informative alternative by generating a set of non-dominated solutions that represent different compromises between the objectives.

In this work, cooperative dual-arm trajectory generation is formulated as a constrained multi-objective optimization problem aimed at minimizing and balancing the interaction forces required during the transport of a shared payload. Two representative population-based multi-objective algorithms are considered: Multi-Objective Particle Swarm Optimization (MOPSO) and Non-dominated Sorting Genetic Algorithm II (NSGA-II) [14,15]. Within the proposed framework, both algorithms are applied under equivalent bounds, constraints, and objective definitions to compare their ability to generate diverse Pareto-optimal cooperative trajectories and identify meaningful trade-offs in the force demands of the two manipulators.

The main contributions of this paper can be summarized as follows:

1. An object-centric multi-objective optimization framework for cooperative dual-arm robotic manipulation, where the transported payload trajectory defines synchronized references for both manipulators.
2. A force-aware trajectory evaluation strategy that considers the interaction-force requirements of both robots, enabling the analysis of load sharing and force distribution during cooperative execution.
3. A comparative assessment of MOPSO and NSGA-II for constrained cooperative manipulation, focusing on the quality and distribution of the obtained Pareto fronts and their suitability for selecting force-efficient dual-arm trajectories.

The remainder of the paper is organized as follows. Section 2 reviews the state of the art on cooperative dual-arm manipulation and multi-objective optimization in robotics. Section 3 introduces the multi-objective metaheuristic algorithms considered in this work, namely MOPSO and NSGA-II. Section 4 presents the proposed force-aware multi-objective

tive optimization framework, including the object-centric trajectory representation, feasibility constraints, end-effector mapping, and force-based objective formulation. Section 5 describes the rack pick-and-place use case. Section 6 reports and discusses the results. Finally, Section 7 summarizes the main conclusions and outlines future research directions.

2. State of the Art

2.1. Cooperative Bimanual Robotic Manipulation

Cooperative bimanual manipulation has become an increasingly relevant research field in robotics due to its potential to extend the capabilities of single-arm manipulators [16]. By coordinating two robotic arms around a common object, a dual-arm system can perform tasks that require higher payload capacity, improved object stability, larger grasping support, and more complex reorientation motions [17,18]. These characteristics are especially important in industrial environments where objects may be bulky, heavy, fragile, or difficult to manipulate safely with a single end effector.

Smith et al. distinguish between coordinated manipulation tasks in which the arms perform different parts of the same task and bimanual manipulation, where both arms physically interact with the same object. In goal-coordinated manipulation, the arms contribute to a common objective without directly sharing physical interaction, whereas “bimanual manipulation is defined as physically interacting with the same object” [19]. Under this definition, cooperative dual-arm manipulation can be understood as a strongly coupled bimanual task, since the manipulated object acts as the physical link between both robotic arms. This coupling introduces a closed-chain-like behavior, in which the relative pose between the object and each gripper must remain constant throughout the task. Consequently, trajectory planning for cooperative dual-arm systems must simultaneously address reachability, joint limits, collision avoidance, inter-arm coordination, and object stability.

Early and classical approaches to cooperative manipulation focus mainly on the kinematic and dynamic modeling of multiple manipulators grasping a shared rigid object [20]. In these formulations, the grasp matrix plays a central role in relating the object wrench to the forces and torques applied by each manipulator [21]. The main strength of these model-based formulations is that they provide a rigorous mathematical description of the relationship between object motion, contact wrenches, and load distribution among the manipulators. This makes them particularly useful for analyzing force closure, internal forces, and cooperative equilibrium conditions. However, these methods usually require accurate dynamic and contact models and are often formulated at the control or force-distribution level rather than as global trajectory planners in cluttered environments. As a result, they do not always address the simultaneous optimization of object motion, collision avoidance, reachability, and force-related performance along a complete manipulation trajectory.

A second group of approaches addresses cooperative manipulation through coordinated motion planning and whole-body or closed-chain control strategies [22]. These methods are effective for enforcing the geometric consistency between the arms and the object, since they explicitly consider the coupled kinematic structure generated by the shared payload. Their main advantage is the ability to generate coordinated motions that preserve the relative grasp configuration and avoid infeasible inter-arm motions. Nevertheless, many of these approaches primarily focus on feasibility, tracking, or coordination accuracy, while the physical quality of the resulting trajectory, in terms of contact-force reduction or balanced load sharing, is usually considered only indirectly or at a later control stage.

More recent works have expanded cooperative manipulation toward more complex and realistic scenarios, including mobile dual-arm platforms, human–robot collaboration, unstructured environments, deformable objects, and learning-based bimanual manipulation [23,24]. These contributions have significantly increased the applicability of dual-arm

systems by addressing perception uncertainty, environmental variability, deformable-object behavior, and generalization across tasks. For example, learning-based and benchmark-oriented approaches are useful for improving adaptability in unstructured environments, while deformable-object manipulation methods extend the scope of bimanual manipulation beyond rigid payloads. However, these methods often emphasize task execution, perception-action mapping, or motion generation, and they do not necessarily provide an explicit multi-objective analysis of the trade-off between the force demands imposed on each manipulator.

In industrial cooperative transport tasks, the manipulated object is frequently held by both arms under rigid or semi-rigid grasping assumptions. In these cases, the object trajectory directly determines the end-effector references and strongly influences the contact forces required to maintain the grasp. Therefore, a trajectory that is collision-free and kinematically valid may still be mechanically undesirable if it produces high gripping forces, large variations in contact effort, or an uneven distribution of load between the two manipulators. This limitation is especially relevant when the task is performed in constrained workspaces, where small changes in the object path can significantly modify the posture of each arm and the resulting force requirement.

Despite these advances, many trajectory generation methods for dual-arm manipulation still prioritize geometric feasibility over physical performance. A trajectory may be collision-free and kinematically valid, but it can still generate high interaction forces, poor load distribution, or excessive effort in one of the manipulators. In contrast to approaches that mainly focus on kinematic coordination, dynamic modeling, or task execution, the present work formulates the cooperative transport problem directly as a force-aware multi-objective trajectory optimization problem. The proposed method uses an object-centric representation, so that the payload trajectory defines synchronized references for both manipulators, and evaluates each candidate motion according to the contact-force requirements of both robots. This allows the optimizer to generate a Pareto set of feasible cooperative trajectories that explicitly represents different compromises between the force demanded from each manipulator. Therefore, the contribution of this work lies not only in generating feasible cooperative motions, but also in providing a structured way to analyze force-sharing trade-offs during dual-arm manipulation.

2.2. Multi-Objective Optimization in Robotic Trajectory Planning

Robotic trajectory planning is inherently a multi-criteria problem. In practical applications, a trajectory is rarely evaluated according to a single performance indicator [25]. Instead, it must satisfy several competing requirements, such as reducing execution time, minimizing energy consumption, improving smoothness, avoiding obstacles, limiting joint torques, reducing mechanical stress, and maintaining safe interaction forces [26]. These criteria are often conflicting: for instance, a faster trajectory may require higher accelerations and larger forces, while a smoother trajectory may increase execution time or path length. For this reason, multi-objective optimization has become a valuable tool for robotic trajectory generation.

Traditional single-objective approaches usually combine multiple criteria into a scalar cost function through weighted sums or penalty terms [27]. Although this strategy is simple and computationally convenient, it presents an important limitation: the result strongly depends on the selected weights [28]. In robotic applications, this dependence can be problematic because the relative importance of time, smoothness, energy, collision clearance, torque, or force may change depending on the task, payload, environment, or safety requirements. Moreover, different weight combinations may lead to different solutions, and an inappropriate choice of weights can hide relevant trade-offs or bias the optimizer toward one performance criterion. This can be restrictive in cooperative manipu-

lation, where different operating conditions may require different compromises between performance, safety, and mechanical effort.

Multi-objective optimization addresses this limitation by searching for a set of non-dominated solutions instead of a single optimum [29]. In robotic trajectory planning, this is particularly useful because it allows us to analyze several feasible trajectories and select the most appropriate one according to the desired operational criterion. For example, one solution may reduce the force required by the left manipulator, another may reduce the force required by the right manipulator, and another may provide a more balanced compromise between both arms. This is particularly relevant in cooperative dual-arm manipulation, because minimizing the effort of one manipulator may not necessarily minimize the effort of the other. Therefore, preserving the objectives separately provides more information than aggregating them into a single scalar index.

Among multi-objective optimization methods, evolutionary and swarm-based algorithms are especially attractive for robotics because they do not require analytical gradients and can handle nonlinear, non-convex, and simulation-based objective functions [30]. This is important in trajectory planning, where each candidate solution may need to be evaluated through inverse kinematics, collision checking, dynamic or quasi-static calculations, and constraint verification. In such cases, the objective landscape may be discontinuous or difficult to differentiate, making population-based metaheuristics a practical alternative to classical gradient-based optimization. Their main drawback is the relatively high computational cost associated with evaluating many candidate trajectories, especially when collision checking and force estimation are performed at multiple sampled points. However, this cost is acceptable in offline planning scenarios, where the objective is to obtain high-quality feasible trajectories before execution.

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) is one of the most widely used evolutionary algorithms for multi-objective optimization [31,32]. Its popularity is mainly due to its elitist selection strategy, non-dominated sorting mechanism, and crowding-distance operator, which helps preserve diversity along the Pareto front. These characteristics make NSGA-II suitable for robotic trajectory planning problems where both convergence toward high-quality solutions and diversity among trade-off alternatives are required. In trajectory optimization, NSGA-II is particularly useful when the feasible region is irregular or highly constrained, because selection, crossover, and mutation allow the algorithm to explore different regions of the search space. Its diversity-preservation mechanism is also advantageous when the final decision maker needs a broad set of alternative trajectories. Nevertheless, NSGA-II may require a sufficiently large population and number of generations to obtain a dense front, especially when feasibility constraints strongly restrict the admissible search space.

Multi-Objective Particle Swarm Optimization (MOPSO) extends the principles of Particle Swarm Optimization to multi-objective problems by incorporating Pareto dominance and an external archive of non-dominated solutions [33]. In this type of algorithm, particles explore the decision space while being guided by leaders selected from the archive [34]. This mechanism is particularly useful in continuous optimization problems, such as trajectory planning, where candidate solutions are represented by real-valued variables defining waypoints, control points, or motion parameters [35]. MOPSO has therefore been applied to several robotic optimization problems involving trade-offs between time, energy, smoothness, tracking quality, or obstacle avoidance. The main advantage of MOPSO is its ability to exploit promising regions of a continuous search space efficiently by combining individual memory, swarm interaction, and archive-based leader selection. This can lead to fast convergence toward low-cost regions. However, its performance can be sensitive to archive management and leader-selection strategies, and the resulting Pareto front may become sparse or concentrated in specific regions if diversity is not sufficiently maintained.

Recent studies in robotic trajectory planning have increasingly adopted multi-objective formulations to optimize performance indicators simultaneously. However, the application of multi-objective optimization to cooperative dual-arm manipulation remains unexplored. In this work, MOPSO and NSGA-II are selected as representative population-based multi-objective algorithms to address force-aware cooperative trajectory optimization.

3. Multi-Objective Metaheuristic Algorithms

This section presents the two multi-objective metaheuristic algorithms considered in this work: Multi-Objective Particle Swarm Optimization (MOPSO) and Non-dominated Sorting Genetic Algorithm II (NSGA-II). Both methods are population-based optimization techniques designed to approximate the Pareto-optimal set of problems involving several conflicting objectives. Their main advantage in robotic trajectory optimization is that they do not require analytical gradients and can operate with nonlinear, non-convex, and simulation-based objective functions, which are common when candidate trajectories must be evaluated through kinematic feasibility, collision checking, and force-related performance indicators.

3.1. Multi-Objective Particle Swarm Optimization

Multi-Objective Particle Swarm Optimization extends the original Particle Swarm Optimization algorithm to multi-objective problems by incorporating Pareto dominance and an external archive of non-dominated solutions. In the original PSO, each candidate solution is represented by a particle that moves through the search space according to its own experience and the collective experience of the swarm. In MOPSO, this idea is preserved, but the single global best solution is replaced by a set of leaders selected from an archive that stores the best non-dominated solutions found during the search.

At iteration t , each particle i is characterized by its position x_i^t and velocity v_i^t . The particle velocity is updated by combining three terms: an inertial component, a cognitive component associated with the particle's personal best position, and a social component guided by a leader selected from the external archive. This velocity update is expressed in Equation (1).

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (p_i^t - x_i^t) + c_2 r_2 (g^t - x_i^t) \quad (1)$$

where ω is the inertia weight, c_1 and c_2 are the cognitive and social acceleration coefficients, r_1 and r_2 are random numbers uniformly distributed in $[0,1]$, p_i^t is the personal best position of particle i , and g^t is the leader selected from the archive. Once the velocity has been updated, the new position of the particle is obtained according to Equation (2).

$$x_i^{t+1} = x_i^t + v_i^t \quad (2)$$

The personal best of each particle is updated using Pareto dominance instead of a single scalar fitness value. If the new position dominates the previous personal best, it replaces it; if the previous personal best dominates the new position, it is preserved. When both solutions are mutually non-dominated, a selection criterion is applied, commonly based on diversity or random choice. This rule can be expressed as

$$p_i^{t+1} = \begin{cases} x_i^{t+1} & \text{if } F(x_i^{t+1}) < F(p_i^t) \\ p_i^t & \text{if } F(x_i^{t+1}) > F(p_i^t) \\ \text{select}(x_i^{t+1}, p_i^t) & \text{otherwise} \end{cases} \quad (3)$$

A key element of MOPSO is the external archive A , which stores the non-dominated solutions discovered by the swarm. After each iteration, the archive is updated by combining the previous archive with the current population and retaining only the non-dominated solutions. This mechanism is mathematically expressed in Equation (4).

$$A^{t+1} = ND(A^t \cup \{x_1^{t+1}, x_2^{t+1}, \dots, x_N^{t+1}\}) \quad (4)$$

where ND denotes the operator that extracts the non-dominated subset and N is the number of particles. Since the archive size is usually limited, diversity-preservation mechanisms are required when the number of non-dominated solutions exceeds the maximum archive capacity. A common approach consists of dividing the objective space into hypercubes and giving preference to solutions located in less populated regions. This strategy improves the spread of the Pareto front and prevents the algorithm from concentrating excessively around a narrow region of the objective space.

The leader g_i^t used in the velocity update is selected from the archive. In many MOPSO variants, leaders are chosen probabilistically, favoring less crowded regions of the objective space to promote diversity. This balance between convergence and diversity is one of the main strengths of MOPSO. The swarm progressively moves toward promising areas of the Pareto front, while the archive preserves alternative trade-off solutions found during the search. For continuous robotic trajectory optimization problems, this behavior is particularly useful because candidate trajectories are usually encoded by real-valued variables, such as waypoints, control points, timing parameters, or object poses.

3.2. Non-Dominated Sorting Genetic Algorithm II

The Non-dominated Sorting Genetic Algorithm II is one of the most widely used evolutionary algorithms for multi-objective optimization. It is based on three main mechanisms: elitism, non-dominated sorting, and diversity preservation through crowding distance. Unlike MOPSO, which updates the population using velocity-based motion, NSGA-II generates new candidate solutions through genetic operators such as selection, crossover, and mutation.

At generation (t), the algorithm starts with a parent population $P^t = \{x_1^t, x_2^t, \dots, x_N^t\}$. From this population, an offspring population Q^t is generated by applying selection, crossover, and mutation. The parent and offspring populations are then merged into a combined population, as defined in Equation (5).

$$R^t = P^t \cup Q^t \quad (5)$$

where N is the population size, P^t is the parent population at generation t , Q^t is the offspring population generated from P^t , and R^t is the combined population used for the non-dominated sorting process.

This combined population is size $2N$ and is sorted into different non-dominated fronts according to Pareto dominance. The sorting process can be represented in Equation (6).

$$R^t = F_1 \cup F_2 \cup F_k \quad (6)$$

where $F_1, F_2, \dots, F_k$ are the non-dominated fronts obtained after sorting the combined population R^t , and K is the total number of fronts generated in the sorting process. The first front, F_1 , contains the non-dominated solutions of R^t . The second front, F_2 , contains the solutions that are dominated only by the solutions in F_1 . In general, each subsequent front F_k , with $k = 2, \dots, K$, contains the solutions that become non-dominated after removing all the solutions assigned to the previous fronts ($F_1, \dots, F_{k-1}$). Therefore, lower values of k indicate better Pareto ranks. Each solution is then assigned a rank according to the front to which it belongs:

$$\text{rank}(x) = r, x \in F_r \quad (7)$$

The new parent population P^{t+1} is constructed by adding complete fronts in increasing order of rank until the population size (N) is reached. If the next front cannot be fully included, NSGA-II applies the crowding-distance criterion to select the most diverse sub-

set of solutions. The crowding distance estimates the density of solutions surrounding each individual in the objective space. For solution i , this distance is calculated according to Equation (8).

$$d_i = \sum_{m=1}^M \frac{f_m(i+1) - f_m(i-1)}{f_m^{\max} - f_m^{\min}} \quad (8)$$

where M is the number of objectives, $f_m(i+1)$ and $f_m(i-1)$ are the neighboring objective values after sorting the front according to objective m , and $f_m^{\max}$ and $f_m^{\min}$ are the maximum and minimum values of that objective within the same front. Boundary solutions are assigned to an infinite crowding distance to preserve the extreme regions of the Pareto front.

The selection process in NSGA-II is governed by the crowded-comparison operator. Given two candidate solutions x_a and x_b , solution x_a is preferred over x_b if it has a lower non-domination rank or, if both solutions belong to the same front, if it has a larger crowding distance. This criterion is expressed in Equation (9).

$$x_a \prec_n x_b \Leftrightarrow [\text{rank}(x_a) < \text{rank}(x_b)] \vee [\text{rank}(x_a) = \text{rank}(x_b) \wedge d_a > d_b] \quad (9)$$

This operator allows NSGA-II to simultaneously promote convergence toward the Pareto-optimal region and preserve a well-distributed set of solutions along the front.

The offspring population is produced using genetic operators. In continuous optimization problems, crossover generates new candidate solutions by combining information from two parent individuals. A simplified representation of this crossover operation is shown in Equation (10).

$$x_c = \alpha x_a + (1 - \alpha) x_b \quad (10)$$

where x_a and x_b are parent solutions, x_c is the generated offspring, and α is a recombination coefficient. Mutation introduces local perturbations into selected decision variables to maintain diversity and explore new regions of the search space. This operation is defined in Equation 11.

$$x'_j = x_j + \Delta_j \quad (11)$$

where x_j is the original value of the (j)-th decision variable and Δ_j is a bounded perturbation.

The elitist structure of NSGA-II ensures that the best non-dominated solutions are preserved from one generation to the next, while the crowding-distance mechanism avoids premature loss of diversity. This makes NSGA-II particularly suitable for problems where it is important not only to approach the Pareto-optimal front but also to obtain a broad and well-distributed set of trade-off solutions.

3.3. Comparison Between MOPSO and NSGA-II

Although MOPSO and NSGA-II share the same general objective of approximating the Pareto front, their search mechanisms are different. MOPSO relies on the collective motion of particles guided by personal best positions and leaders selected from an external archive. This gives the algorithm a strong memory component and makes it especially effective in continuous search spaces, where particles can progressively refine promising regions of the solution space.

NSGA-II, in contrast, evolves the population through selection, crossover, and mutation. Its main strengths are its elitist replacement strategy and its explicit diversity preservation through the crowding-distance mechanism. As a result, NSGA-II often provides a well-distributed Pareto front, particularly when the trade-off surface contains different regions that must be explored simultaneously.

In robotic trajectory optimization, these differences are relevant because the search space is usually continuous, constrained, and highly nonlinear. MOPSO can provide fast convergence by exploiting the archive of non-dominated solutions, while NSGA-II can offer robust population diversity through non-dominated sorting and genetic recombination. A common evaluation framework therefore enables a consistent comparison of both algorithms, considering not only the quality of the obtained solutions but also their convergence behavior and distribution across the Pareto front.

4. Force-Aware Multi-Objective Optimization Framework for Cooperative Dual-Arm Manipulation

4.1. Object-Centric Cooperative Manipulation Model

The cooperative manipulation task is formulated from an object-centric perspective. Instead of planning the motion of each manipulator independently, the transported payload is considered the central element of the trajectory generation process. The trajectory is therefore defined in the task space of the object, and the motion of both end effectors is derived from the object pose.

The payload is modeled as a rigid box grasped simultaneously by two robotic manipulators through fixed contact points located on opposite sides of the object. Under this assumption, any admissible motion of the box directly determines the required motion of the left and right end effectors. This formulation ensures that both robots remain coordinated during the transport task and that the relative pose between each gripper and the box is preserved throughout the trajectory.

The pose of the box with respect to the world frame is represented by a six-dimensional vector:

$$p^{box} = [x, y, z, roll, pitch, yaw]^T \in SE(3) \quad (12)$$

where x , y , and z define the Cartesian position of the box reference frame, while $roll$, $pitch$, and yaw define its orientation. Since the grasp is assumed to be rigid, the relative transformations between the box frame and both end-effector frames remain constant during the complete motion. Consequently, once a candidate box pose is defined, the corresponding left and right end-effector references are uniquely obtained from the fixed grasp geometry.

This formulation transforms the cooperative planning problem into a search for a feasible object trajectory that can be simultaneously followed by both manipulators. The generated trajectory must satisfy the reachability and joint-limit constraints of both robots, avoid collisions with the environment and between the manipulators, and reduce the contact-force requirements associated with the cooperative transport of the payload.

The following assumptions are considered in this study. First, the transported object is rigid, and its geometry does not change during the task. Second, the grasp points remain fixed with respect to the box, and no regrasping or slipping is considered during trajectory execution. Third, the environment is static and represented by collision bodies. Finally, the force analysis is based on a simplified frictional contact model that estimates the normal and tangential forces required at each end effector to satisfy the translational force equilibrium of the payload.

Under these assumptions, the no-slip condition at the grasp points is only considered valid when the tangential contact forces required to support and move the payload remain within the available friction limits generated by the normal contact forces. Therefore, for each sampled pose of the trajectory, the estimated contact forces must satisfy both the payload force equilibrium and the friction-cone condition. If these conditions are not fulfilled, the candidate trajectory is considered physically infeasible and its simulation is stopped.

Figure 1 shows the initial cooperative configuration used in the simulation, where two robotic manipulators grasp the same rigid box under fixed bimanual contact conditions. This configuration illustrates the object-centric formulation adopted in the study, since the box trajectory defines the synchronized motion references for both manipulators.

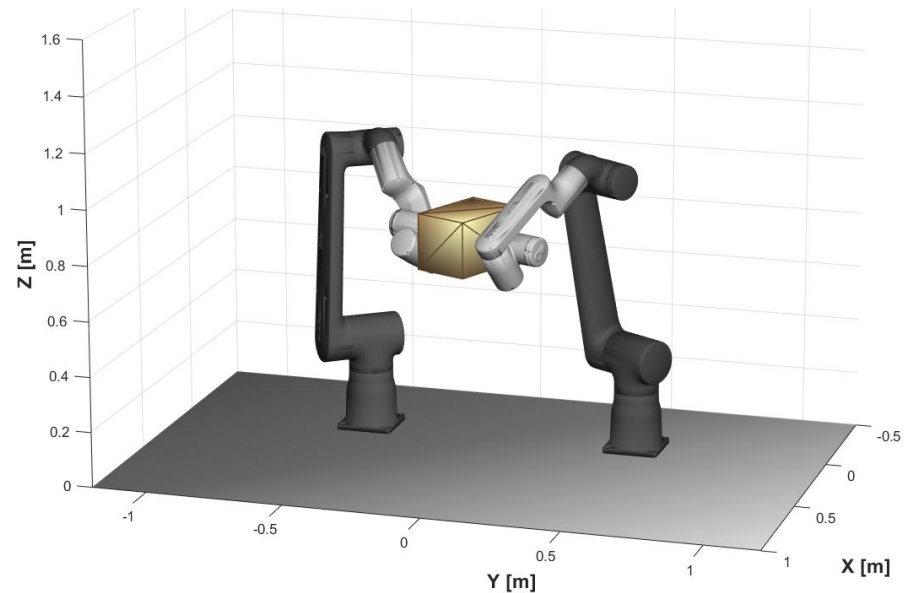


Figure 1. Representative configuration of the cooperative dual-arm manipulation task, where two robotic manipulators grasp and transport the same rigid box under fixed bimanual contact conditions.

4.2. Decision Variables and Trajectory Parameterization

The trajectory is parameterized by a set of intermediate object poses between the fixed initial and final configurations of the task. These intermediate poses are the decision variables optimized by the multi-objective algorithms. The pose representation follows the object-centric formulation introduced in Section 4.1, where each box pose is expressed as a six-dimensional vector containing the Cartesian position and roll–pitch–yaw orientation of the payload.

Let the complete object trajectory be defined by n_{total} box poses, including both fixed and optimized poses. If m poses are fixed by the task definition, the number of optimized object poses is given in Equation (13):

$$n = n_{\text{total}} - m \quad (13)$$

Since each optimized object pose contains six scalar variables, the total number of decision variables is:

$$n_{\text{vars}} = 6n_{\text{opt}} = 6(n_{\text{total}} - m) \quad (14)$$

Each optimized waypoint is defined as

$$p_k^{\text{box}} = \begin{bmatrix} x_k \\ y_k \\ z_k \\ \phi_k \\ \theta_k \\ \psi_k \end{bmatrix} \quad (15)$$

where x_k , y_k , and z_k are the Cartesian coordinates of the box reference frame, and ϕ_k , θ_k , and ψ_k denote roll, pitch, and yaw, respectively. Therefore, a candidate solution generated by MOPSO or NSGA-II is encoded as shown in Equation (16).

$$x = \left[(p_1^{box})^T \quad (p_2^{box})^T \quad \dots \quad (p_{n_{opt}}^{box})^T \right]^T \in \mathbb{R}^{6n_{opt}} \quad (16)$$

The fixed initial and final poses are imposed as boundary conditions and are not modified by the optimizer. This guarantees that every candidate trajectory starts from the same pick configuration and ends at the same place configuration.

Once the optimized intermediate poses are generated, they are assembled with the fixed poses to obtain the complete ordered set of object waypoints:

$$\mathcal{P}^{box} = \{p_0^{box}, p_1^{box}, \dots, p_{n_{total}-1}^{box}\} \quad (17)$$

where the fixed and optimized poses are arranged according to their order in the task trajectory.

The complete set of waypoints is converted into a continuous object trajectory using quintic polynomial interpolation. For each trajectory segment and for each pose component s , with

$$s \in \{x, y, z, \phi, \theta, \psi\} \quad (18)$$

The interpolation is performed component-wise using a fifth-order polynomial:

$$s_k(\lambda) = a_{0,k} + a_{1,k}\lambda + a_{2,k}\lambda^2 + a_{3,k}\lambda^3 + a_{4,k}\lambda^4 + a_{5,k}\lambda^5 \quad (19)$$

where $\lambda \in [0,1]$ is the normalized interpolation parameter of segment k , and $a_{0,k}, \dots, a_{5,k}$ are the polynomial coefficients. Quintic interpolation was selected because it provides smooth transitions between consecutive waypoints and allows continuous position, velocity, and acceleration profiles, which is particularly important because the contact-force estimation depends directly on the translational acceleration of the payload.

The resulting continuous trajectory is sampled into N discrete object poses:

$$p^{box}(t_i) = \begin{bmatrix} x(t_i) \\ y(t_i) \\ z(t_i) \\ \phi(t_i) \\ \theta(t_i) \\ \psi(t_i) \end{bmatrix}, i = 1, \dots, N \quad (20)$$

These sampled poses are then used for inverse kinematics, collision checking, contact-force estimation, and residual moment evaluation.

The temporal law is assigned from the Cartesian displacement of the box center. Let the Cartesian position of the box at sample i be:

$$r^{box}(t_i) = \begin{bmatrix} x(t_i) \\ y(t_i) \\ z(t_i) \end{bmatrix} \quad (21)$$

The Euclidean distance between two consecutive sampled box positions is:

$$\Delta \ell_i = \| \mathbf{r}^{box}(t_{i+1}) - \mathbf{r}^{box}(t_i) \|_2 \quad (22)$$

The timestamp associated with each sampled pose is computed as

$$t_1 = 0 \quad (23)$$

$$t_i = \sum_{j=1}^{i-1} \frac{\Delta \ell_j}{v_{\text{ref}}}, i = 2, \dots, N \quad (24)$$

where v_{ref} is the reference linear velocity of the box. This timing law assigns longer durations to longer Cartesian displacements and provides a common velocity reference for all candidate trajectories.

The translational velocity of the box center of mass is computed from the sampled Cartesian trajectory and the associated timestamps as shown in Equation (25).

$$\dot{r}^{\text{box}}(t_i) = \frac{dr^{\text{box}}(t_i)}{dt} = \begin{bmatrix} \dot{x}(t_i) \\ \dot{y}(t_i) \\ \dot{z}(t_i) \end{bmatrix} \quad (25)$$

The translational acceleration is then obtained as

$$a(t_i) = \dot{r}^{\text{box}}(t_i) = \frac{dr^{\text{box}}(t_i)}{dt} = \begin{bmatrix} a_x(t_i) \\ a_y(t_i) \\ a_z(t_i) \end{bmatrix} \quad (26)$$

In the numerical implementation, these derivatives are evaluated from the sampled trajectory using finite-difference gradient approximations with respect to the computed timestamps.

Regarding orientation, roll–pitch–yaw angles are interpolated component-wise together with the Cartesian coordinates. Before computing the box rotation matrices, the angular sequences are unwrapped to avoid artificial discontinuities caused by 2π jumps. The rotation matrix of the box at each sampled instant is then obtained using the yaw–pitch–roll convention:

$$R_{\text{box}}(t_i) = R_{ZYX}(\psi(t_i), \theta(t_i), \phi(t_i)) \quad (27)$$

or equivalently:

$$R_{\text{box}}(t_i) = R_z(\psi(t_i))R_y(\theta(t_i))R_x(\phi(t_i)) \quad (28)$$

Although roll–pitch–yaw interpolation does not define a geodesic interpolation on $SO(3)$, it was considered adequate for the present task because the angular variations are bounded and remain away from singular configurations. Therefore, no Euler-angle singularity was encountered in the analyzed trajectories. The interpolated orientation is used to compute the end-effector references, grasp-frame directions, collision geometry, and the angular quantities required for the residual moment evaluation described in Section 4.6. For tasks involving larger orientation changes, quaternion-based interpolation could be incorporated as a future extension.

4.3. Mapping from Object Motion to End-Effector References

For each sampled instant, the box pose is expressed as a homogeneous transformation matrix in Equation (29).

$$T_{\text{box}}(t_i) = \begin{bmatrix} R_{\text{box}}(t_i) & p_{\text{box}}(t_i) \\ 0^\top & 1 \end{bmatrix} \quad (29)$$

where $p_{\text{box}}(t_i)$ is the Cartesian position of the box and $R_{\text{box}}(t_i)$ is its rotation matrix. The two end-effectors are defined by constant offsets with respect to the box frame. These offsets represent the fixed grasping locations on both sides of the payload.

The left and right end-effector positions are obtained by transforming the grasp offsets from the box frame to the world frame:

$$p_L(t_i) = p_{\text{box}}(t_i) + R_{\text{box}}(t_i) r_L \quad (30)$$

$$p_R(t_i) = p_{\text{box}}(t_i) + R_{\text{box}}(t_i) r_R \quad (31)$$

where r_L and r_R are the constant grasp offsets expressed in the box reference frame. Similarly, the end-effector orientations are defined by applying fixed relative rotations with respect to the box orientation:

$$R_L(t_i) = R_{box}(t_i) R_{Lrel} \quad (32)$$

$$R_R(t_i) = R_{box}(t_i) R_{Rrel} \quad (33)$$

The geometric meaning of the grasp offsets used in Equations (16)–(19) is illustrated in Figure 2, where the box reference frame and the fixed offsets to the left and right grasping points are represented.

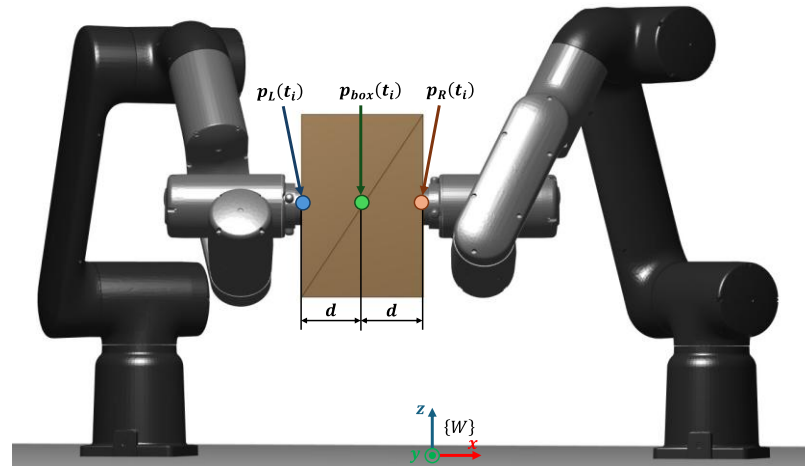


Figure 2. Geometric interpretation of the object-to-end-effector position mapping and contact representation [13].

This mapping produces two synchronized Cartesian references, one for each manipulator. These references are then converted into joint-space configurations through inverse kinematics. If a sampled pose cannot be reached by one of the manipulators or violates joint limits, the corresponding candidate trajectory is considered infeasible.

4.4. Contact-Force Estimation Under Frictional Grasp

The force model estimates the contact forces required at both end effectors to satisfy the translational force equilibrium of the payload. The acceleration of the box center of mass is obtained from the sampled object trajectory. The required resultant force acting on the payload is then computed as shown in Equation (34).

$$F_{req}(t) = m(a(t) - g) \quad (34)$$

where m is the mass of the box, $a(t)$ is the translational acceleration of the box center of mass, and g is the gravity vector:

$$\begin{bmatrix} F_{req,x}(t) \\ F_{req,y}(t) \\ F_{req,z}(t) \end{bmatrix} = m \left(\begin{bmatrix} a_x(t) \\ a_y(t) \\ a_z(t) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \right) \quad (35)$$

At each contact, the force applied by manipulator i is decomposed into a normal component and a tangential component:

$$F_i(t) = N_i(t)n_i(t) + t_i(t) \quad (36)$$

where $N_i(t)$ is the normal contact force, $n_i(t)$ is the contact normal direction, and $t_i(t)$ is the tangential force component:

$$\begin{bmatrix} F_{i,x}(t) \\ F_{i,y}(t) \\ F_{i,z}(t) \end{bmatrix} = N_i(t) \begin{bmatrix} n_{i,x}(t) \\ n_{i,y}(t) \\ n_{i,z}(t) \end{bmatrix} + \begin{bmatrix} t_{i,x}(t) \\ t_{i,y}(t) \\ t_{i,z}(t) \end{bmatrix} \quad (37)$$

The tangential component is limited by the friction cone [36]:

$$\|t_i(t)\| \leq \mu N_i(t) \quad (38)$$

where μ is the friction coefficient. This condition ensures that the tangential force required to support and move the object does not exceed the frictional capacity generated by the normal gripping force.

The normal contact force is selected as the main force-related optimization variable because it directly represents the gripping effort required at each end effector to maintain the payload under a no-slip condition. For a given friction coefficient, higher normal forces increase the available tangential friction capacity, but they also imply a higher mechanical demand for the manipulators. These contact forces are transmitted through the robot kinematic chain and are reflected as joint torque demands, increasing the load supported by the actuators, transmissions, and bearings. Therefore, although the proposed model does not compute a direct wear index or a complete joint-level fatigue model, minimizing the required normal contact force provides a physically meaningful proxy for reducing the mechanical demand imposed on the manipulators. In this sense, lower normal forces are expected to reduce the contact-induced loading transmitted to the joints and to promote smoother, less demanding cooperative trajectories.

The relationship between the optimized normal contact force and the mechanical loading of the robot can be interpreted through the contribution of the external contact wrench to joint torques. In general, the joint torque vector of manipulator i can be expressed as

$$\tau_i(t) = M_i(q)\ddot{q} + C_i(q, \dot{q})\dot{q} + G_i(q) + J_i^T(q)w_{c,i}(t) \quad (39)$$

where M_i , C_i , and G_i denote the inertial, Coriolis/centrifugal, and gravitational terms, respectively, and $w_{c,i}(t)$ is the contact wrench applied at the end effector. The contact wrench can be written as

$$w_{c,i}(t) = \begin{bmatrix} M_{c,i}(t) \\ F_i(t) \end{bmatrix} \quad (40)$$

where $M_{c,i}(t)$ is the contact moment and $F_i(t)$ is the contact force. In the main force-allocation model used in this work, each contact is represented by a resultant force applied at the nominal contact point. Under this point-contact assumption, no local contact moment is explicitly imposed, i.e., $M_{c,i}(t) = 0$. Therefore, joint torque vector, $\tau_i(t)$, can be computed from the contact-force contribution to the joint torques, $\tau_{c,i}(t)$, as

$$\tau_i(t) = M_i(q)\ddot{q} + C_i(q, \dot{q})\dot{q} + G_i(q) + \tau_{c,i}(t), \quad \tau_{c,i}(t) = J_{v,i}^T(t)F_i(t) \quad (41)$$

where $J_{v,i}(t)$ is the translational part of the manipulator Jacobian at the contact configuration.

Since the contact force is decomposed into a normal and a tangential component, and the tangential component is constrained by the friction cone, the contact-force magnitude can be bounded as

$$\|F_i(t)\| = \sqrt{N_i^2(t) + \|t_i(t)\|^2} \leq N_i(t)\sqrt{1 + \mu^2} \quad (42)$$

assuming that $n_i(t)$ is a unit vector and that $t_i(t)$ lies in the tangent plane of the contact.

Therefore, using the contact force bound in Equation (42), the contact-force-induced joint torque contribution satisfies:

$$\| \tau_{F,i}(t) \| \leq \| J_{v,i}^T(t) \|_2 \| F_i(t) \| \quad (43)$$

and, under the friction-cone assumption:

$$\| \tau_{F,i}(t) \| \leq \| J_{v,i}^T(t) \|_2 N_i(t) \sqrt{1 + \mu^2} \quad (44)$$

This relation shows that minimizing the maximum normal contact force reduces an upper bound of the contact-force magnitude and, consequently, of the contact-induced joint torque contribution for a given manipulator posture and friction coefficient. Therefore, although joint torque is not directly included as an optimization objective in this work, the proposed normal-force objective is physically related to the mechanical loading transmitted to the robot joints.

Joint torque indicators were not included as direct optimization objectives in this study to keep the objective formulation focused on contact-level load sharing between the two manipulators and to avoid introducing additional robot-model-dependent dynamic terms into the Pareto search. Instead, the torque-utilization analysis presented in Section 6.3 is used as an independent post hoc validation of whether the force-optimized trajectories also reduce the mechanical demand at the joint level. The incorporation of explicit torque-based objectives is considered a relevant extension for future work.

The two end-effectors must jointly provide the force required by the payload:

$$F_L(t) + F_R(t) = F_{req}(t) \quad (45)$$

At each sampled instant, the force distribution between both contacts is obtained by solving a minimum-effort problem subject to the translational equilibrium and friction constraints. This provides the normal forces $N_L(t)$ and $N_R(t)$, as well as the tangential components $t_L(t)$ and $t_R(t)$. Normal components are used as the main optimization objectives because they represent the gripping effort required at each manipulator.

Although the main optimization objectives are based on the normal contact forces obtained from the translational force equilibrium, an additional rotational consistency analysis was included to evaluate whether the moment terms neglected by the point-contact translational model could significantly affect the results. For this purpose, the rotational moment required by the payload motion was compared with the moment generated by the contact-force couple at the payload center of mass.

First, the inertia tensor of the rectangular payload with respect to its center of mass is defined in the payload reference frame as

$$I_b = \begin{bmatrix} \frac{1}{12} m(L_y^2 + L_z^2) & 0 & 0 \\ 0 & \frac{1}{12} m(L_x^2 + L_z^2) & 0 \\ 0 & 0 & \frac{1}{12} m(L_x^2 + L_y^2) \end{bmatrix} \quad (46)$$

where L_x , L_y , and L_z are the payload dimensions. At each sampled instant, this inertia tensor is expressed in the world frame as

$$I_w(t) = R_{box}(t) I_b R_{box}^T(t) \quad (47)$$

where $R_{box}(t)$ is the rotation matrix of the payload. The rotational moment required by the payload motion is then estimated as

$$\tau_{req}(t) = I_w(t) \alpha(t) + \omega(t) \times (I_w(t) \omega(t)) \quad (48)$$

where $\omega(t)$ and $\alpha(t)$ are the angular velocity and angular acceleration of the payload, respectively, obtained from the sampled orientation trajectory.

In a very simplified point-contact model, it is assumed that the contact point is the center of the end-effector of the robot. Thus, the moment generated by the contact-force couple about the payload center of mass is computed as

$$\tau_F^p(t) = r_L(t) \times F_L(t) + r_R(t) \times F_R(t) \quad (49)$$

where $r_L(t)$ and $r_R(t)$ are the vectors from the payload center of mass to the left and right center of the end-effector. The residual rotational moment is therefore defined as

$$\tau_{res}^p(t) = \tau_{req}(t) - \tau_F^p(t) \quad (50)$$

This residual represents the rotational contribution that is not explicitly captured when each distributed contact interaction is reduced to a resultant force applied at a fixed nominal point.

However, in a real grasp, the interaction between the end effector and the payload is distributed over a finite contact surface rather than concentrated at a mathematical point. For modeling purposes, this distributed pressure and friction interaction can be approximated by an equivalent resultant force applied at an effective center of pressure, r_{eff} . This effective point should not be interpreted as the only physical contact point, but as the equivalent application point of the resultant contact force that reproduces the moment generated by the distributed contact. Therefore, allowing the effective center of pressure to move within the contact patch provides a more realistic approximation than forcing the resultant force to act at the geometric center of the end effector, while still keeping the model computationally tractable.

Figure 3 illustrates this approximation. The nominal contact point corresponds to the center of the assumed contact patch, r_i , whereas the effective center of pressure may shift within this region due to a non-uniform pressure distribution, r_{eff} .

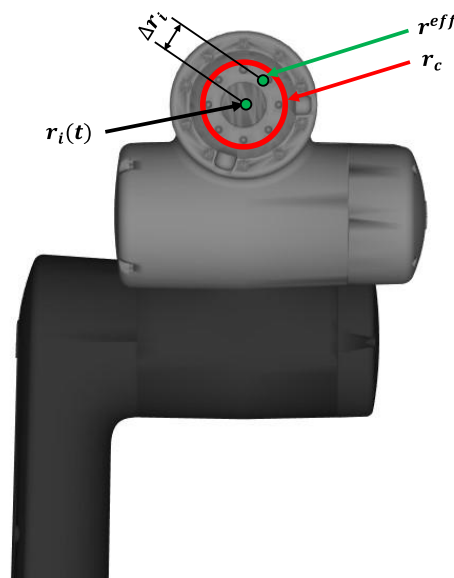


Figure 3. Detail of the GoFa 12 end-effector contact surface.

To account for this effect, the effective contact locations are expressed as

$$r_L^{eff}(t) = r_L(t) + \Delta r_L(t) \quad (51)$$

$$r_R^{eff}(t) = r_R(t) + \Delta r_R(t) \quad (52)$$

where $\Delta r_L(t)$ and $\Delta r_R(t)$ are the displacements of the effective centers of pressure within the left and right contact patches. These displacements are constrained to lie inside the local tangent plane of each contact surface:

$$\Delta r_i(t) \cdot n_i(t) = 0, i \in \{L, R\} \quad (53)$$

and their magnitude is bound by the radius of the assumed circular contact patch:

$$\|\Delta r_i(t)\| \leq r_c = \frac{D_c}{2}, i \in \{L, R\} \quad (54)$$

where D_c is the equivalent contact patch diameter and r_c is the corresponding radius.

Considering this finite contact patch, the moment generated by the contact forces becomes:

$$\tau_F^c(t) = (r_L(t) + \Delta r_L(t)) \times F_L(t) + (r_R(t) + \Delta r_R(t)) \times F_R(t) \quad (55)$$

The required center-of-pressure displacements are obtained by minimizing the remaining residual moment:

$$[\Delta r_L^*(t), \Delta r_R^*(t)] = \arg \min_{\Delta r_L, \Delta r_R} \|\tau_{req}(t) - \tau_F^c(t)\|^2 \quad (56)$$

The remaining residual moment after considering the finite contact patch is:

$$\tau_{res}^c(t) = \tau_{req}(t) - \tau_F^c(t) \quad (57)$$

A candidate trajectory is considered rotationally consistent with the finite contact-patch assumption if the center-of-pressure displacement remains within the available contact surface for both robots:

$$\max(\|\Delta r_L^*(t)\|, \|\Delta r_R^*(t)\|) \leq r_c \quad (58)$$

This condition verifies whether the residual moment obtained with the simplified point-contact model can be compensated by a physically feasible shift in the effective center of pressure inside the contact patch. Therefore, the rotational analysis does not introduce a new optimization objective but provides a validation criterion to ensure that the neglected rotational contribution can be absorbed by the distributed contact interaction for the considered payload geometry and motion range.

It should be noted that the proposed optimization model still estimates the contact forces required to transport the payload under the assumed frictional grasp and uses the normal contact forces as the main force-aware trajectory objectives. The finite contact-patch analysis is included as an additional validation step. It does not modify the main Coulomb friction constraint applied to the resultant contact forces but evaluates whether a realistic contact surface could compensate for the residual moment associated with the point-contact approximation.

4.5. Force-Aware Multi-Objective Evaluation

The main objective of this work is to optimize the cooperative trajectory by minimizing the normal contact forces required by both manipulators. Unlike single-objective formulations, where different criteria are combined into a weighted cost function, the proposed approach keeps the force objectives separated. This allows the optimizer to generate a Pareto front of non-dominated trajectories representing different compromises between the force demands of Robot 1 and Robot 2.

Collision constraints are incorporated directly into the objective evaluation through a piecewise penalty function. If any collision is detected during the trajectory simulation, the evaluation is stopped and the same penalized value is assigned to both objectives. Otherwise, the objectives are computed from the maximum normal contact forces required along the collision-free trajectory.

This way for each candidate trajectory, the evaluation function returns a two-dimensional objective vector:

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} \quad (59)$$

where $f_1(x)$ and $f_2(x)$ are given by (60):

$$f_1(x) = \begin{cases} 1 + T_{coll}(x), & \text{if } T_{coll}(x) > 0 \\ \frac{\max_{t \in [0, T]} N_1(t)}{N_{ref,1}}, & \text{if } T_{coll}(x) = 0 \end{cases}, \quad f_2(x) = \begin{cases} 1 + T_{coll}(x), & \text{if } T_{coll}(x) > 0 \\ \frac{\max_{t \in [0, T]} N_2(t)}{N_{ref,2}}, & \text{if } T_{coll}(x) = 0 \end{cases} \quad (60)$$

where $T_{coll}(x)$ is the accumulated collision time detected during the trajectory evaluation, $N_1(t)$ and $N_2(t)$ are the normal contact forces required at both end effectors, and $N_{ref,1}$ and $N_{ref,2}$ are the corresponding reference force values used for normalization.

With this definition, colliding trajectories are penalized with objective values greater than one, while collision-free trajectories are evaluated according to their normalized force requirements. Therefore, the optimization process first promotes feasible collision-free trajectories and then compares them according to the force demand imposed on each manipulator.

The use of separated objectives allows the optimization process to identify trajectories that, while avoiding collisions, reduce the force requirement of one manipulator, the other manipulator, or different compromises between both. Consequently, the final output is not a single optimized trajectory, but a set of non-dominated solutions from which a final trajectory can be selected according to the desired force-sharing criterion.

4.6. Feasibility Constraints

The optimization variables are defined at the object level; therefore, the feasibility constraints must also be expressed with respect to the motion of the transported box. However, the physical limitations of the system are imposed by the two robotic manipulators and by the surrounding environment. Consequently, each candidate object trajectory is accepted only if all its sampled poses generate valid end-effector references for both robots and remain compatible with the admissible manipulation workspace.

The first constraint is related to the reachable workspace of both manipulators. For a given sampled object pose, the corresponding left and right end-effector positions are obtained from the fixed grasp geometry. Therefore, a candidate object pose is only valid if the induced end-effector positions lie inside the admissible workspace of their corresponding robots, as shown in Equation (61).

$$p_L(t_i) \in \mathcal{W}_L, p_R(t_i) \in \mathcal{W}_R, i = 1, \dots, N \quad (61)$$

where $p_L(t_i)$ and $p_R(t_i)$ are the left and right end-effector positions generated by the object pose at sample t_i , and $\mathcal{W}_L$ and $\mathcal{W}_R$ represent the reachable workspaces of the left and right manipulators, respectively. Since the object is grasped simultaneously by both robots, the admissible region for the box is not the complete workspace of either manipulator, but the common region in which both arms can satisfy the grasp constraint at the same time.

This idea is represented in Figure 4. The individual reachable regions of the left and right robots are shown together with the restricted workspace used for the object trajectory. The restricted region defines the admissible volume in which the box waypoints can be placed during the optimization. In this way, the search space is reduced before evaluating the candidate trajectories, preventing the algorithms from generating object poses that would be unreachable for one of the manipulators.

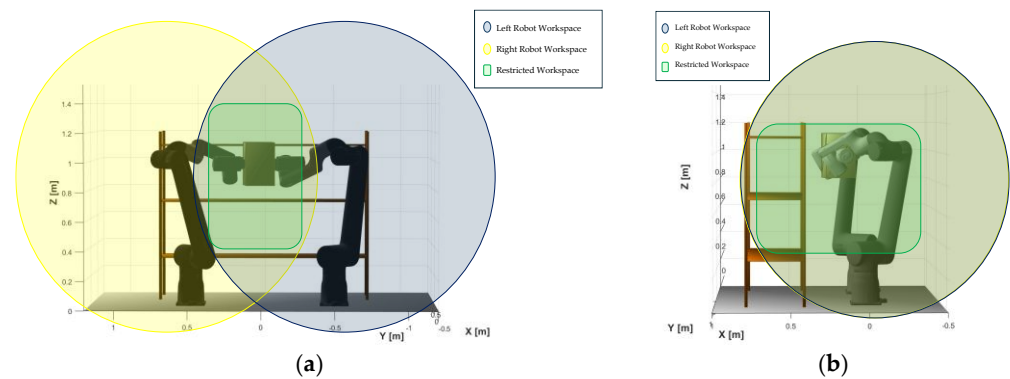


Figure 4. Reachability constraint mapping for the cooperative dual-arm manipulation task: (a) Front view showing the individual reachable workspaces of both manipulators and their common admissible region; (b) side view showing the restricted workspace used to bound the object waypoints during optimization.

In addition to this workspace restriction, every candidate trajectory must satisfy joint feasibility. After mapping the object trajectory to the two end-effector references, inverse kinematics is solved for both manipulators. A trajectory is rejected if any sampled configuration violates the joint limits of either robot:

$$q_{i,min} \leq q_i(t_k) \leq q_{i,max}, i \in \{L, R\}, k = 1, \dots, N \quad (62)$$

where $q_i(t_k)$ is the joint configuration of manipulator i at sample k , and $q_{i,min}$ and $q_{i,max}$ are the corresponding lower and upper joint limits.

Collision avoidance is also enforced along the complete sampled motion. Three types of collisions are considered: collisions between each robot and the rack environment, collisions between the transported box and the environment, and inter-arm collisions between the two manipulators. Therefore, a trajectory is considered feasible only if no collision is detected at any sample of the motion:

$$C_{robot-env}(t_k) = 0, C_{box-env}(t_k) = 0, C_{inter-arm}(t_k) = 0, k = 1, \dots, N \quad (63)$$

where each collision indicator is equal to 1 when a collision is detected and 0 otherwise. The surrounding environment is represented by static collision bodies during the evaluation.

Finally, the rigid grasp condition must be preserved throughout the task. This is ensured by deriving both end-effector references from the same object pose and from constant grasp offsets. As a result, the relative pose between the box and each end effector remains fixed during the complete trajectory, and the two robots execute a coordinated motion consistent with a shared payload.

If the required effective center-of-pressure shift lies outside the available end-effector contact patch, the trajectory evaluation is immediately stopped, and the candidate solution is marked as infeasible.

Overall, a candidate trajectory is considered feasible only if it satisfies all the previous conditions: both end-effectors must remain inside their reachable workspaces, all inverse-kinematics solutions must respect the joint limits, the motion must remain collision-free, and the rigid grasp constraint must be maintained. These constraints define the feasible search region explored by MOPSO and NSGA-II.

5. Use Case: Force-Aware Rack Pick-And-Place Optimization in Cooperative Dual-Arm Manipulation

The proposed framework was evaluated in a representative pick-and-place scenario involving the cooperative transport of a shared object by two robotic manipulators. The task reproduces a constrained industrial manipulation situation in which a payload must

be moved from an initial pick configuration to a final place configuration while maintaining a stable bimanual grasp and avoiding collisions with the shelf, stored items, the ground plane, and the opposite manipulator.

5.1. Work-Cell Layout and Rack Environment

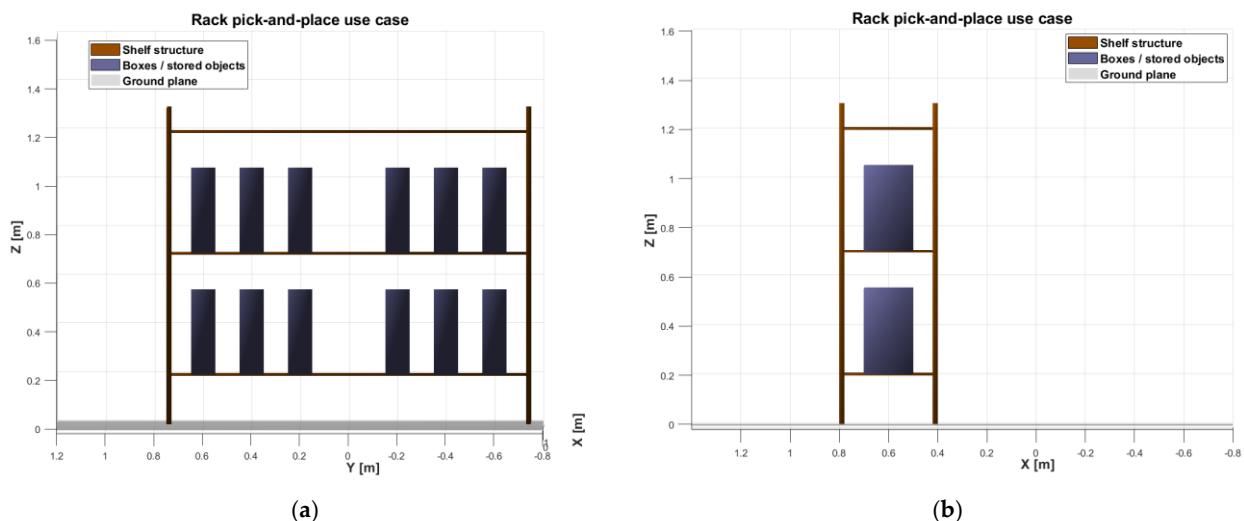
The work cell consists of a shelf-like structure containing several stored objects that restrict the available free space for the cooperative transport task. The two manipulators are positioned on opposite sides of the rack, generating a shared manipulation region in which the payload must be moved while avoiding the surrounding structure and stored items.

The rack, stored boxes, and ground plane are modeled as static collision bodies. The rack has a footprint of 1.00 m in length and 0.40 m in depth, with three shelf levels located at heights of 0.20 m, 0.60 m, and 1.00 m from the ground plane. These dimensions define the main spatial limitations of the benchmark, since the shelf levels and the stored objects reduce the available clearance for the transported payload and constrain the admissible object orientations during cooperative motion.

These elements impose the main geometric constraints of the task and require the optimizer to generate trajectories that avoid collisions while maintaining a valid cooperative grasp. The initial and final object poses are fixed and imposed as boundary conditions for every candidate trajectory evaluated during the optimization process.

The scenario is particularly relevant for cooperative manipulation because feasibility depends on several coupled factors: the reachability of both manipulators, the relative pose between the object and each end effector, the avoidance of collisions with the environment, and the mechanical interaction generated during the motion. Therefore, the same trajectory must be acceptable from both a geometric and a force-aware perspective.

Figure 5 illustrates the rack pick-and-place environment used in the optimization, including the shelf structure, stored objects, and ground plane from different viewpoints.



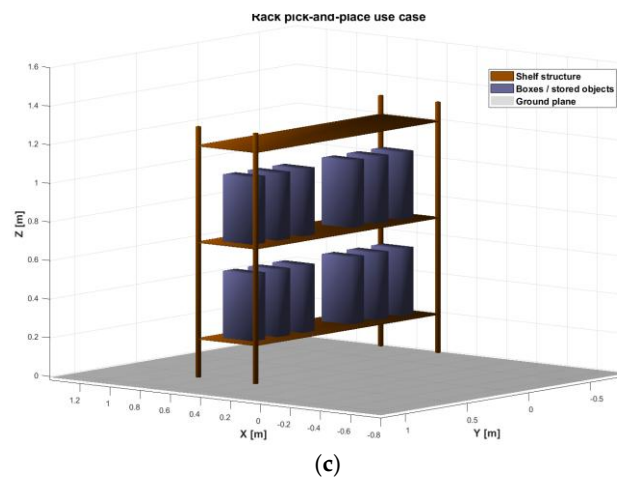


Figure 5. Rack pick-and-place environment considered in the cooperative manipulation task: (a) front view of the shelf structure and stored objects; (b) side view showing the rack depth and shelf levels; (c) isometric view of the complete constrained workspace.

5.2. Dual-Arm Robotic Setup

The validation was conducted using a cooperative dual-arm robotic setup composed of two ABB GoFa 12 (ABB, Zurich, Switzerland) collaborative manipulators placed on opposite sides of the rack-like workspace. Each robot is a 6-axis collaborative manipulator from the ABB CRB 15000 family (ABB, Zurich, Switzerland), with a nominal payload capacity of 12 kg and a reach of 1.27 m. The GoFa family also provides a maximum TCP speed of up to 2.2 m/s and a repeatability of 0.02 mm, which makes it suitable for precise manipulation tasks in constrained environments.

In the proposed setup, both manipulators are required to transport a shared rigid object under a fixed bimanual grasp. The object is treated as the central element of the task, and its trajectory is planned in task space. Each object pose is then mapped to the corresponding end-effector references of both manipulators through fixed grasp transformations. This object-centric representation ensures that the cooperative constraint is preserved throughout the complete motion, since both robots follow synchronized references derived from the same object trajectory.

The simulated robotic model includes the kinematic chains of both GoFa 12 manipulators, the transported object, and the surrounding collision bodies that define the rack environment. Since the focus of this study is the optimization of interaction forces, the system is evaluated in terms of kinematic feasibility, collision-free execution, and force-related objective values associated with each manipulator.

Figure 6 shows the ABB GoFa 12 collaborative manipulator used as the basis for the dual-arm setup and its corresponding workspace representation. The physical robot is shown in Figure 6a, while Figure 6b illustrates the reach envelope considered for defining the manipulation workspace.

5.3. Multi-Objective Evaluation Metrics

To compare MOPSO and NSGA-II under the same experimental conditions, a set of metrics was defined to evaluate feasibility, Pareto-front quality, and force-related performance. Since the proposed problem is multi-objective, the comparison is not limited to a single best solution but also considers the number of feasible and non-dominated trajectories generated by each algorithm and the force behavior of representative solutions selected from the Pareto front.

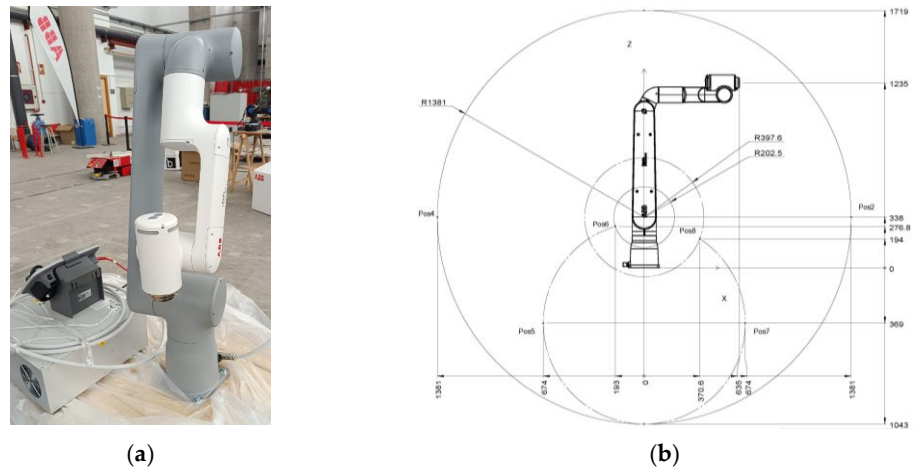


Figure 6. ABB GoFa 12 collaborative manipulator and workspace representation: (a) physical robot used as reference for the simulation model; (b) robot reach diagram used to define the available manipulation workspace.

The metrics considered in this study are the following:

(a) Feasible Solutions

This metric represents the number of candidate trajectories that satisfy the feasibility conditions imposed in the optimization process. A trajectory is considered feasible when it satisfies the kinematic constraints of both manipulators and remains collision-free throughout the complete motion. This value provides a direct indication of the ability of each algorithm to explore valid regions of the search space.

(b) Percentage of Feasible Solutions

The percentage of feasible solutions is computed as the ratio between the number of feasible trajectories and the total number of evaluated candidates. This metric normalizes the feasibility performance of each algorithm and allows a direct comparison between MOPSO and NSGA-II under equivalent optimization settings.

The percentage of feasible solutions is computed as the ratio between the number of feasible trajectories and the total number of evaluated candidates:

$$P_{\text{feas}} = \frac{N_{\text{feas}}}{N_{\text{eval}}} \cdot 100 \quad (64)$$

where N_{feas} is the number of feasible trajectories and N_{eval} is the total number of evaluated candidate solutions. This metric normalizes the feasibility performance of each algorithm and allows a direct comparison between MOPSO and NSGA-II under equivalent optimization settings.

(c) Non-Dominated Solutions

This metric indicates the number of Pareto-optimal solutions obtained by each algorithm. These solutions represent different trade-offs between the force objective of Robot 1 and the force objective of Robot 2. A larger number of non-dominated solutions generally provides a broader set of alternatives for selecting a final cooperative trajectory.

(d) Hypervolume

The hypervolume indicator was used to evaluate the quality of the obtained Pareto front in terms of both convergence and spread. For a minimization problem, the hypervolume measures the area of the objective space dominated by the non-dominated front with respect to a predefined reference point. Since the optimization objectives correspond to the normalized force-related objectives of Robot 1 and Robot 2, the hypervolume was computed in the normalized objective space using the reference point: $r_{HV} =$

[1.1, 1.1]. This point was selected to be slightly worse than the upper bound of the normalized objective range, so that all feasible non-dominated solutions were included in the hypervolume calculation. A larger hypervolume value indicates a better approximation of the Pareto front, since it reflects both a wider coverage of the objective space and a closer approach to low-force regions.

(e) Inverted Generational Distance

The inverted generational distance was used to quantify the distance between the Pareto front obtained by each algorithm and an approximate reference Pareto front. Since the true Pareto front of the problem is unknown, the reference front was constructed as the non-dominated union of all final Pareto fronts obtained by both algorithms across all replicas. Therefore, the reference set P^{ref} contains the best non-dominated solutions found in the complete experimental campaign, considering both MOPSO and NSGA-II.

All IGD calculations were performed in the normalized objective space. For a Pareto front P obtained by a given algorithm and the approximate reference front P^{ref} , IGD is computed as

$$IGD(P, P^{ref}) = \frac{1}{|P^{ref}|} \sum_{z \in P^{ref}} \min_{p \in P} \|z - p\| \quad (66)$$

where z is a point of the reference front and p is a point of the Pareto front obtained by the algorithm. Thus, unlike the hypervolume indicator, IGD does not use a single predefined reference point, but a reference set composed of the non-dominated solutions obtained from all algorithms and replicas. Lower IGD values indicate that the obtained front is closer to the approximate reference Pareto front.

(f) Spacing

The spacing metric was used to evaluate the uniformity of the distribution of non-dominated solutions along the Pareto front. It is based on the distance between each non-dominated solution and its nearest neighbor in the objective space. Lower spacing values indicate a more uniform distribution of solutions, which is desirable because it provides a more regular set of trade-off alternatives.

For a Pareto front with N_p non-dominated solutions, the spacing metric can be expressed as

$$S = \sqrt{\frac{1}{N_p - 1} \sum_{i=1}^{N_p} (d_i - \bar{d})^2} \quad (67)$$

where d_i is the distance from solution i to its nearest neighboring solution in the Pareto front, and $\bar{d}$ is the mean of these nearest-neighbor distances. A lower value of S indicates a more evenly distributed Pareto front.

(g) Maximum Normal Contact Force of the Selected Solution

To evaluate the mechanical behavior of a representative trajectory, a balanced solution is selected from the Pareto front of each algorithm. For this selected trajectory, the maximum normal contact force is computed for each manipulator as

$$F_{N,\max,i} = \max_{t \in [0,T]} F_{N,i}(t) \quad (68)$$

where $F_{N,i}(t)$ is the normal contact force of manipulator i at time t . This metric provides a direct measure of the peak gripping effort required by each robot during the cooperative execution.

(h) Pareto-Front Distribution

The distribution of the Pareto front is analyzed to evaluate how well the non-dominated solutions cover the objective space. In this work, the objective space is defined by the force-related objective of Robot 1 and the force-related objective of Robot 2. A well-distributed Pareto front is desirable because it provides different force-sharing alternatives, ranging from solutions that favor one manipulator to more balanced trajectories.

6. Results

This section presents the results obtained from the multi-objective optimization of the cooperative dual-arm manipulation task using MOPSO and NSGA-II. The analysis focuses on three complementary aspects: the distribution of feasible and non-dominated solutions, the evolution of the force objectives during the optimization process, and the force profiles associated with representative balanced trajectories selected from the resulting Pareto fronts.

6.1. Multi-Objective Optimization Comparison

Figure 7 shows a representative Pareto-front comparison obtained from the best-performing run of MOPSO and NSGA-II in the objective space defined by the force objective of Robot 1 and the force objective of Robot 2. Both algorithms were able to generate feasible cooperative trajectories and approximate a set of non-dominated solutions. However, clear differences can be observed in the density and distribution of the solutions.

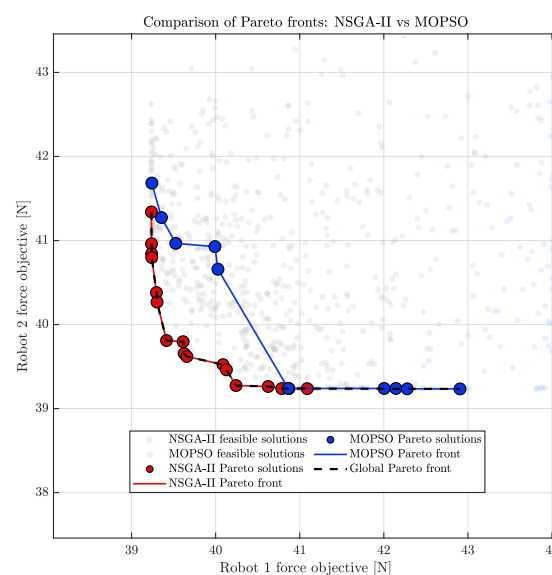


Figure 7. Comparison of Pareto fronts obtained with MOPSO and NSGA-II. Feasible solutions are represented as scattered points, while the non-dominated solutions and the global Pareto front are highlighted in the objective space defined by the force objectives of Robot 1 and Robot 2.

The Pareto-front comparison shown in Figure 6 provides a representative visualization of the behavior of both algorithms in the objective space. NSGA-II produces a denser and more continuous distribution of non-dominated solutions, especially in the region where both force objectives are simultaneously reduced. MOPSO also identifies valid Pareto-optimal solutions, but its front is sparser and shows a more pronounced trade-off between the force objective of Robot 1 and that of Robot 2. Therefore, Figure 6 is used as a representative visual comparison, while the quantitative assessment of the algorithms is based on the statistical results obtained from ten independent runs.

To provide a more robust comparison, MOPSO and NSGA-II were executed over ten independent replicas using the same computational budget of 1000 evaluated candidate

solutions per run. For each replica, the number of feasible solutions, the number of non-dominated solutions, and three standard multi-objective performance indicators were computed: hypervolume (HV), inverted generational distance (IGD), and spacing. Hypervolume evaluates the extension and convergence of the obtained Pareto front, with higher values indicating better performance. IGD measures the distance with respect to the approximate reference Pareto front, with lower values indicating better convergence. Spacing evaluates the uniformity of the non-dominated solutions, with lower values indicating a more regular distribution. Table 1 summarizes the statistical results obtained over the ten replicas.

Table 1. Quantitative results of the multi-objective optimization process in rack scenario.

Metric	MOPSO	NSGA-II
Feasible Solutions Mean	296	784
% of feasible solutions	29.6%	78.4%
Non-dominated solutions mean	9.8	17.7
Non-dominated solutions std	1.5492	1.4944
HV mean	1.172	1.2019
HV std	0.00060694	0.00011677
IGD mean	0.052489	0.040696
IGD std	0.0014644	0.0090187
Spacing mean	0.116	0.018459
Spacing std	0.084916	0.0086637

Overall, these results confirm that NSGA-II provides a more robust and better distributed approximation of the Pareto front for the considered cooperative manipulation scenario. MOPSO is also able to find feasible and competitive solutions, but it generates fewer feasible and non-dominated solutions and shows lower Pareto-front quality according to the HV, IGD, and spacing indicators.

It should be noted that the statistical comparison in Table 1 is based on the ten independent replicas, whereas the subsequent figures showing convergence, force profiles, torque utilization, residual moment, payload sensitivity, and trajectory shapes are presented for selected representative solutions. These representative solutions are used to illustrate the mechanical behavior of the optimized trajectories and should not be interpreted as average curves over all replicas.

6.2. Evolution of Force Objectives

Figure 8 reports the best-so-far evolution of the force objectives for the same representative runs shown in Figure 6. Therefore, this plot should be interpreted as an illustrative convergence example, while the robustness of the comparison between algorithms is assessed using the statistical indicators reported in Table 1.

The results show a rapid initial reduction in the force objectives for both algorithms. In the case of MOPSO, the first objective exhibits a sharp decrease during the early iterations, moving from a high initial value to the low-force region in a short number of iterations. NSGA-II also shows a progressive reduction in both objectives, with a more gradual convergence trend. After the initial improvement phase, all curves converge toward a narrow force range close to the final Pareto region.

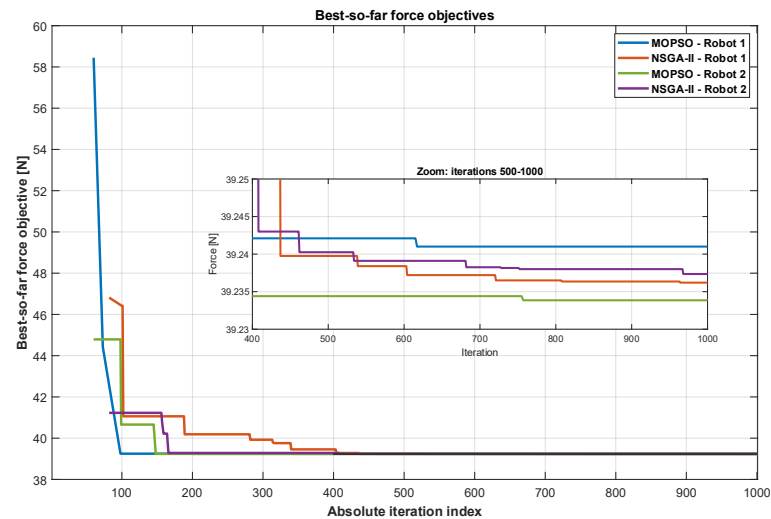


Figure 8. Best-so-far evolution of the force objectives for MOPSO and NSGA-II. The main plot shows the complete optimization process, while the inset highlights the late-stage refinement between iterations 500 and 1000.

The zoomed view between iterations 500 and 1000 highlights the late-stage refinement behavior. In this region, the improvements are small but still relevant, since the algorithms are already operating close to the minimum force levels found in the search. NSGA-II maintains a slightly more progressive refinement of the objectives, while MOPSO reaches a stable low-force region earlier. This behavior is consistent with the mechanisms of both algorithms: MOPSO tends to exploit the archive of non-dominated solutions efficiently, whereas NSGA-II preserves diversity through non-dominated sorting and crowding distance.

The convergence curves confirm that both algorithms are able to reduce the force objectives significantly. However, the previous Pareto-front analysis shows that NSGA-II provides a larger and more diverse set of feasible trade-off solutions, which is particularly relevant when the final trajectory must be selected according to a balanced force-sharing criterion.

6.3. Force Profiles of the Selected Balanced Trajectories

To further analyze the mechanical behavior of the optimized solutions, one representative trajectory was selected from the Pareto front of each algorithm. This trajectory-level analysis is complementary to the statistical comparison reported in Table 1 and is intended to illustrate the behavior of selected balanced solutions rather than the average response over all replicas. The selection criterion was the minimum mean normal contact force between both manipulators, computed from the force values of Robot 1 and Robot 2 for each non-dominated solution. Therefore, the force profiles shown in Figure 8 correspond to the same selected representative solutions reported quantitatively in Table 2.

Figure 9a,b compares the left and right normal contact forces obtained during the execution of the selected trajectories. In both cases, the force values remain within a narrow range, indicating that the optimized trajectories provide feasible and mechanically reasonable cooperative motions. However, differences can be observed in the temporal distribution and balance of the forces.

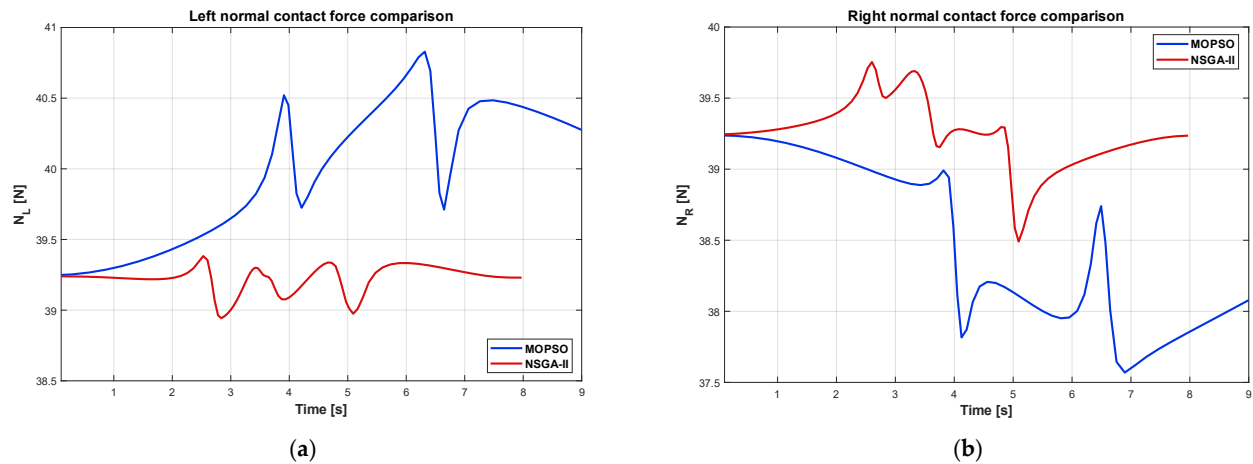


Figure 9. Normal contact-force comparison for the selected representative minimum-mean-force trajectories obtained with MOPSO and NSGA-II: (a) left normal contact force; (b) right normal contact force.

The quantitative minimum force values associated with the selected representative solutions are summarized in Table 2.

Table 2. Minimum normal contact-force values of the selected representative trajectories in the rack scenario.

Multi-Objective Algorithm	R_L Min Force (N)	R_R Min Force (N)	Mean Contact Force (N)
MOPSO	37.569	38.491	40.052
NSGA-II	39.657	38.943	39.638

For the left normal contact force, NSGA-II produces a smoother profile with lower variability around the mean force level. The MOPSO trajectory, in contrast, shows a more pronounced variation, with a decrease during the middle part of the motion followed by an increase near the final part of the trajectory. This suggests that the selected MOPSO trajectory introduces a less uniform distribution of the contact effort during execution.

A similar behavior is observed in the right normal contact force. The NSGA-II profile remains comparatively smoother and more stable, whereas MOPSO produces a higher force variation during the central part of the motion. This indicates a less balanced transfer of mechanical effort between the manipulators during part of the trajectory. These differences are consistent with the trajectory shapes discussed in Section 6.4.

To complement the force-profile analysis, the maximum joint torque contribution induced by the estimated contact wrench was also evaluated for each manipulator. In this case, the contact interaction was mapped into the joint space using the complete geometric Jacobian, considering both the force and moment components of the wrench, i.e., ($\tau_F = J_v^T F$). The resulting values were normalized with respect to the maximum admissible torque of each GoFa12 joint, so that the radar plots in Figure 9 represent the percentage of joint torque capacity used by each trajectory.

Figure 10a,b compares the initial trajectory with the optimized trajectories obtained with MOPSO and NSGA-II for the right and left manipulators, respectively. In both robots, the optimized trajectories reduce the joint torque utilization with respect to the initial motion, confirming that the proposed multi-objective optimization not only improves the contact-force behavior but also decreases the mechanical loading transmitted to the robot joints. This reduction is not uniform for all axes, since each joint is affected differently depending on the manipulator posture, the contact wrench direction, and the trajectory geometry. However, the overall trend clearly shows lower torque demand after optimization.

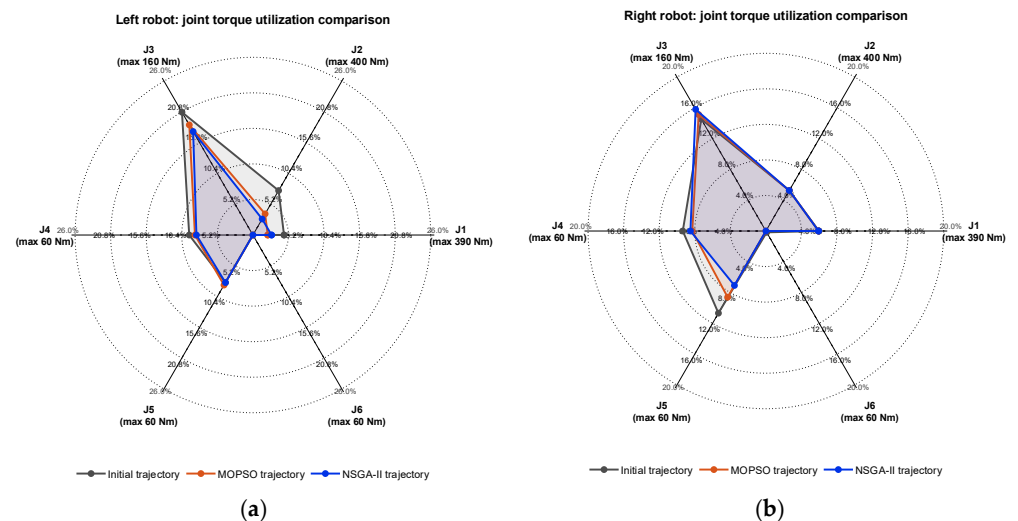


Figure 10. Maximum joint torque utilization with respect to the GoFa12 joint limits for the initial trajectory and the selected optimized trajectories obtained with MOPSO and NSGA-II: (a) right manipulator; (b) left manipulator.

Among the two multi-objective algorithms, NSGA-II generally provides the lowest joint torque utilization values across the evaluated joints, especially when compared with the initial trajectory and, in several axes, also with the MOPSO solution. This indicates that NSGA-II achieves a more favorable redistribution of the mechanical effort along the kinematic chain, leading to trajectories that are not only feasible in terms of contact-forces but also less demanding from the joint-loading perspective. Therefore, the torque-utilization analysis reinforces the conclusions drawn from the normal contact force profiles and supports the suitability of NSGA-II for generating balanced cooperative manipulation trajectories in the considered rack scenario.

To verify whether the rotational terms could significantly affect the force-based evaluation, an additional residual moment analysis was carried out for the selected trajectories. In the simplified point-contact model, the residual rotational moment represents the difference between the rotational moment required by the payload motion and the moment generated by the contact-force couple at the box center of mass. However, in a real grasp, the contact between the end effector and the payload is not applied at a mathematical point, but over a finite contact area. Therefore, a non-uniform pressure distribution over the contact patch can shift the effective center of pressure and generate an additional compensating moment.

Figure 11 shows the required displacement of the effective center of pressure with respect to the nominal contact point for the selected MOPSO and NSGA-II trajectories. The dashed horizontal line represents the maximum admissible displacement, defined by the radius of the assumed circular contact patch. In this analysis, a contact patch diameter of 50 mm was considered, corresponding to the GoFa 12 end-effector diameter, which results in a maximum admissible center-of-pressure shift of 25 mm.

The results show that the required center-of-pressure displacement remains below the contact patch radius for both optimized trajectories. The MOPSO trajectory requires a larger displacement, reaching approximately 15 mm, whereas the NSGA-II trajectory remains below approximately 7 mm. In both cases, the required displacement is physically contained within the assumed contact surface. This indicates that the residual rotational contribution obtained with the simplified point-contact model can be compensated by a feasible shift in the effective center of pressure within the finite contact area.

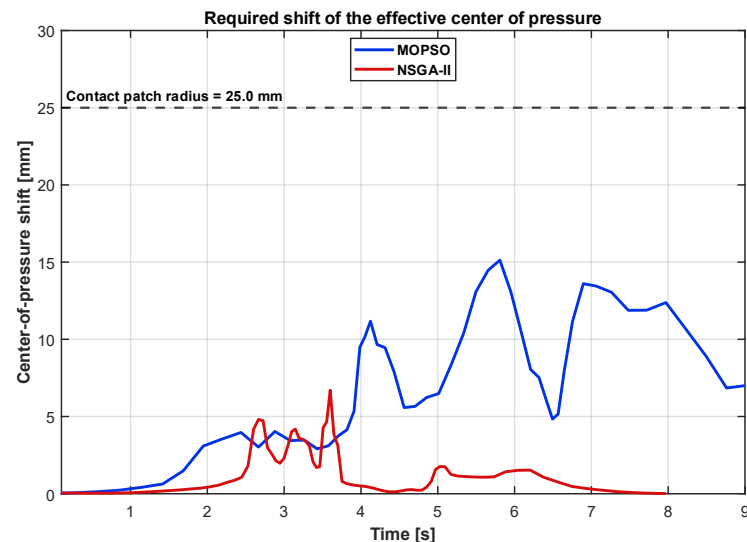


Figure 11. Required displacement of the effective center of pressure with respect to the nominal contact point for the selected MOPSO and NSGA-II trajectories. The dashed line represents the maximum admissible displacement defined by the radius of the assumed circular contact patch.

Therefore, for the considered payload geometry and motion range, the neglected rotational terms do not invalidate the force-based comparison performed in this study. The analysis suggests that the residual moment is not large enough to require an effective contact point outside the available contact patch. Consequently, the simplified normal-force optimization remains valid, while the point-contact residual can be interpreted as a conservative estimate of the rotational contribution that may be absorbed by the distributed contact interaction.

Finally, a payload sensitivity analysis was carried out to estimate the maximum total box mass that could be transported by the cooperative system before either of the two manipulators reached the imposed force limit. For this purpose, the selected MOPSO and NSGA-II trajectories were evaluated by progressively increasing the total mass of the box, while computing the maximum resultant contact force exerted by each robot. In this analysis, the force values are expressed in kilogram-force (kgf) because this unit provides an intuitive interpretation of the payload-related force level for the reader. However, the corresponding SI values are also reported in parentheses using the conversion ($1 \text{ kgf} = 9.80665 \text{ N}$). Therefore, the imposed force limit of 12 kgf corresponds to 117.68 N .

Figure 12 shows the maximum force exerted by each robot as a function of the total box mass. As expected, the force increases almost linearly with the transported mass, since the same trajectory is evaluated under increasing payload conditions. The MOPSO trajectory reaches the 12 kgf (117.68 N) limit at a total box mass of approximately 16.06 kg , whereas the NSGA-II trajectory reaches the same limit at approximately 16.71 kg . Therefore, both optimized trajectories allow the cooperative system to transport a box mass higher than the individual 12 kg reference, but NSGA-II provides a slightly higher payload margin.

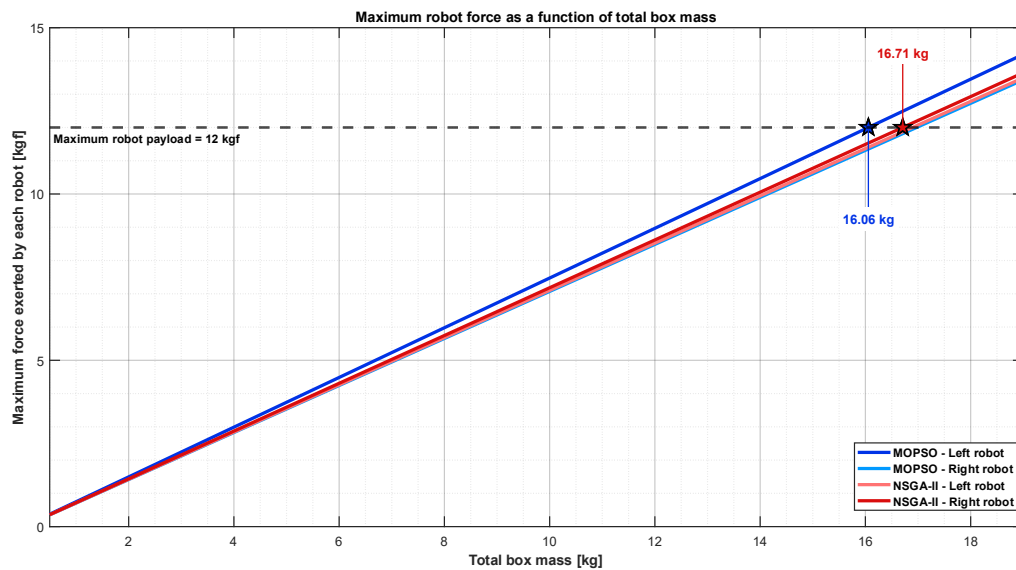


Figure 12. Maximum force exerted by the left and right robots as a function of the total box mass for the selected MOPSO and NSGA-II trajectories.

6.4. Trajectory Analysis

Figure 13a–c compares the initial trajectory with the optimized trajectories obtained using MOPSO and NSGA-II.

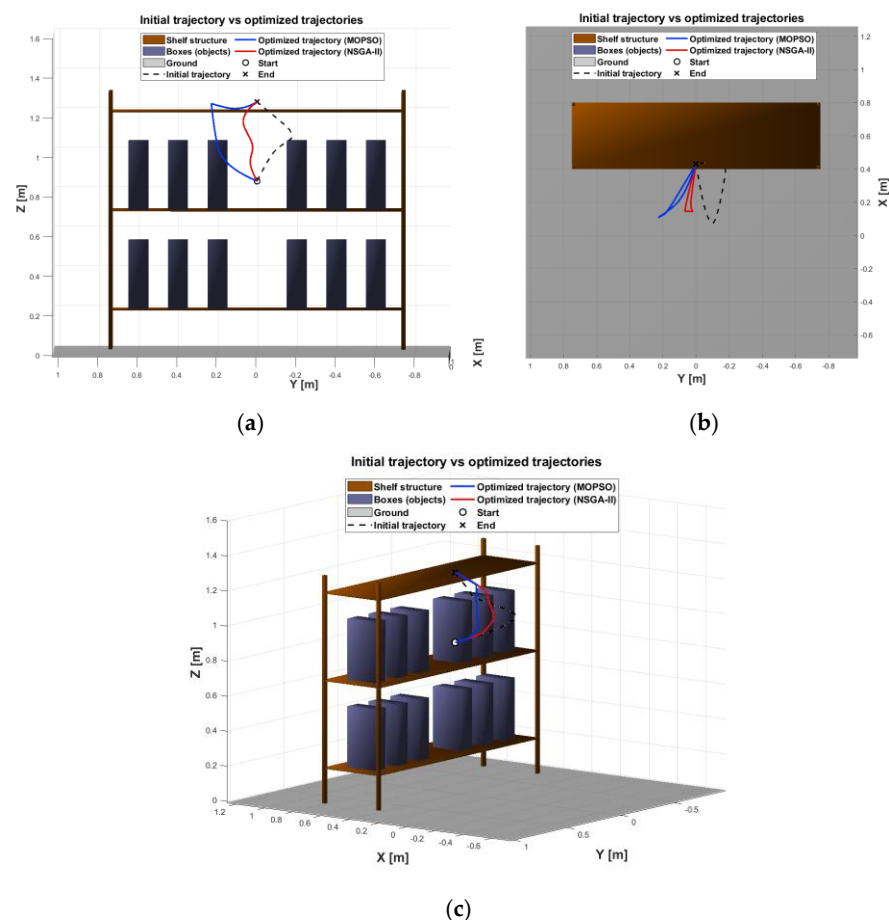


Figure 13. Comparison between the initial trajectory and the optimized trajectories obtained with MOPSO and NSGA-II in the rack-like workspace: (a) orthographic projection onto the Y–Z plane; (b) orthographic projection onto the X–Y plane; (c) three-dimensional perspective view.

Both optimized trajectories modify the initial object path to reduce the force requirements while preserving feasibility within the rack-like environment. The optimized paths remain collision-free and respect the cooperative manipulation constraint imposed by the shared payload.

The trajectory obtained with MOPSO shows a more pronounced lateral deviation. This behavior explains the larger difference observed between the left and right normal contact forces, since the object motion induces a less symmetric contribution of the two manipulators. Although this trajectory remains feasible and force-efficient in terms of the objective values, its execution produces a more uneven temporal distribution of the contact forces. The step-by-step MOPSO-optimized trajectory visualization is depicted in Figure 14a–d.

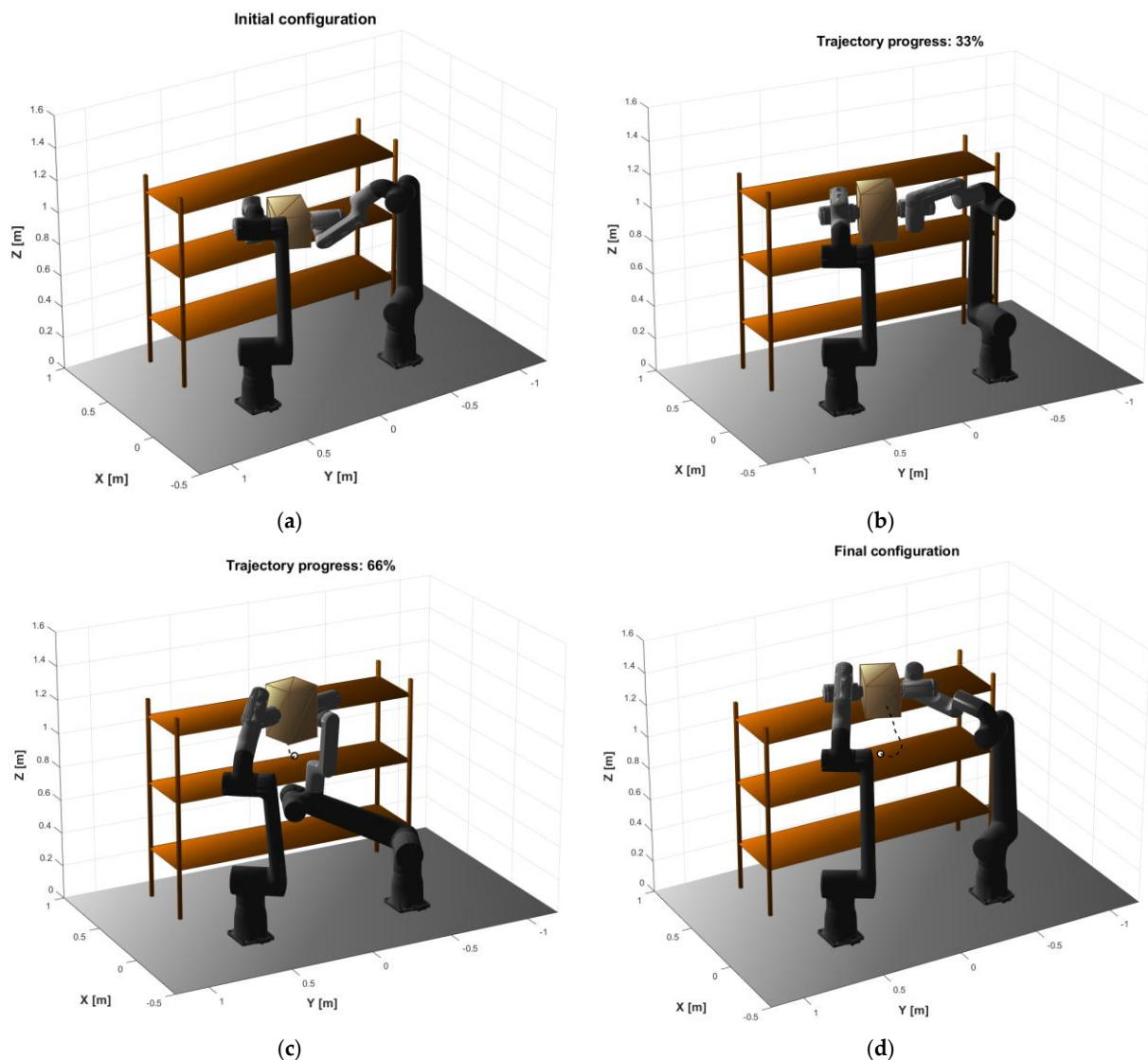


Figure 14. Motion sequence of the optimized trajectory obtained with MOPSO: (a) initial configuration; (b) intermediate configuration at 33% of the trajectory; (c) intermediate configuration at 66% of the trajectory; (d) final configuration.

The trajectory obtained with NSGA-II appears more balanced from the perspective of force sharing. Its path reduces excessive lateral displacement and results in smoother normal-force profiles for both manipulators. This suggests that, among the selected balanced solutions, NSGA-II provides a more mechanically homogeneous cooperative motion. The NSGA-II sequence is presented in Figure 15a–d.

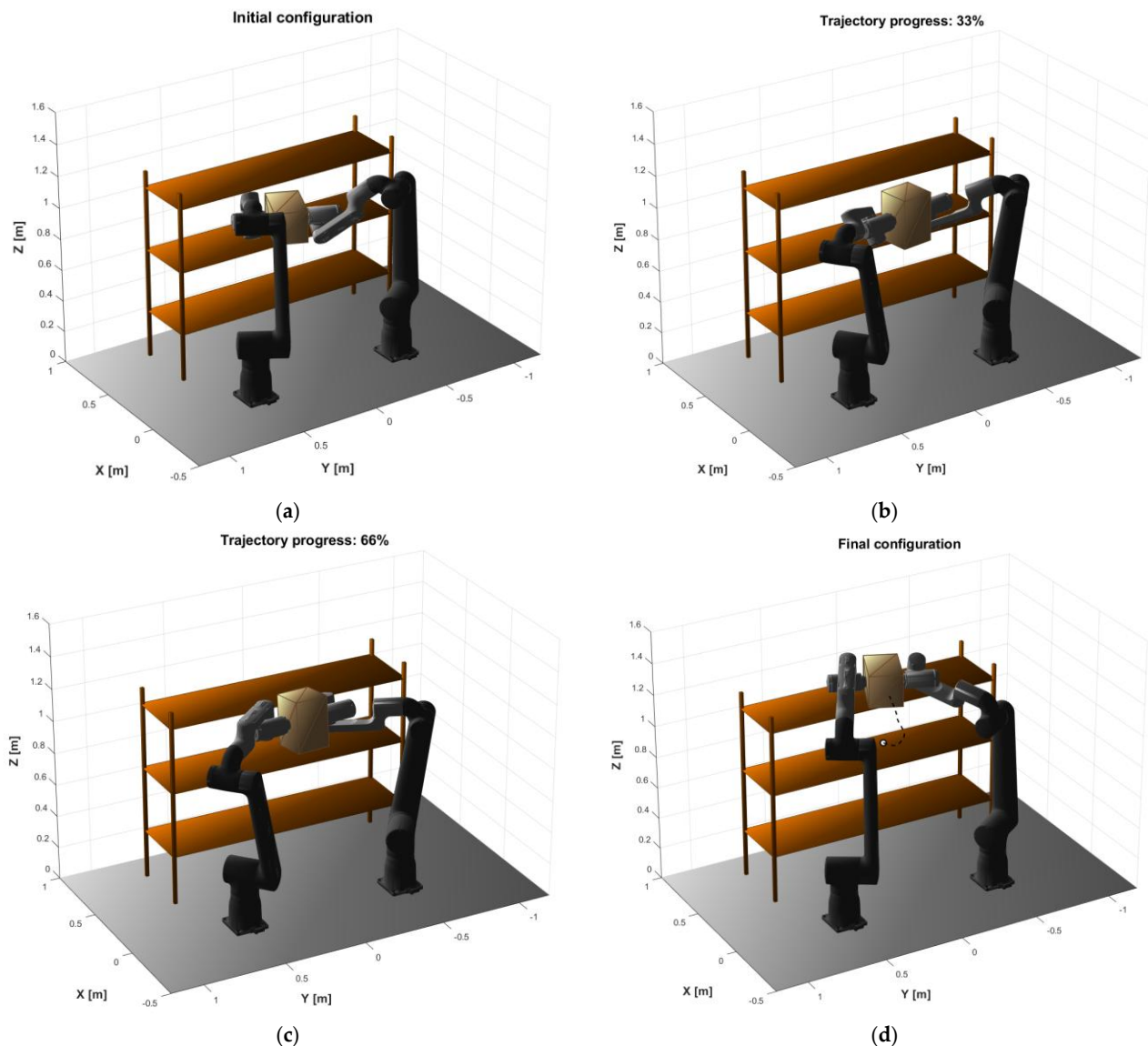


Figure 15. Motion sequence of the optimized trajectory obtained with NSGA-II: (a) initial configuration; (b) intermediate configuration at 33% of the trajectory; (c) intermediate configuration at 66% of the trajectory; (d) final configuration.

7. Conclusions

This work presented a force-aware multi-objective optimization framework for cooperative dual-arm robotic manipulation. The proposed approach formulates the generation of cooperative trajectories as a constrained multi-objective problem, where the interaction-force requirements of both manipulators are minimized while preserving kinematic feasibility and collision-free motion. The task was defined from an object-centric perspective, allowing the motion of the shared payload to be mapped consistently to both robotic arms under rigid grasping assumptions.

Two representative population-based multi-objective algorithms, MOPSO and NSGA-II, were evaluated under equivalent bounds, constraints, and objective definitions. Both algorithms were able to generate feasible cooperative trajectories and identify non-dominated solutions in the force-objective space. However, their behavior differed significantly in terms of feasibility rate, Pareto-front distribution, and force-balance characteristics.

NSGA-II obtained a higher number of feasible solutions, with 694 feasible trajectories compared with 296 for MOPSO, corresponding to feasibility rates of 69.4% and 29.6%,

respectively. It also produced a larger number of non-dominated solutions, resulting in a denser and more continuous Pareto front. This indicates that NSGA-II provided a broader set of trade-off alternatives for selecting cooperative trajectories with balanced force requirements. MOPSO also reached competitive low-force solutions, but its Pareto front was sparser and showed a more limited distribution of feasible alternatives.

The analysis of the selected balanced trajectories further confirmed these differences. The NSGA-II solution produced smoother and more homogeneous normal contact-force profiles for both manipulators, suggesting a more balanced mechanical contribution during cooperative execution. In contrast, the MOPSO trajectory showed a more pronounced lateral motion, which led to larger differences between the left and right contact forces. This behavior highlights the importance of considering not only the final objective values, but also the trajectory shape and the temporal distribution of the interaction forces.

Overall, the results demonstrate the suitability of multi-objective metaheuristic optimization for force-aware cooperative dual-arm manipulation. In the evaluated scenario, NSGA-II provided the most robust compromise between force reduction, load-sharing behavior, and Pareto-front diversity. Future work will extend the proposed formulation by incorporating additional objectives such as execution time, energy consumption, torque-based indicators, and robustness margins. Further validation on real dual-arm robotic platforms will also be considered to assess tracking accuracy, repeatability, and force behavior under experimental conditions.

Future work will extend the proposed formulation by incorporating additional objectives such as execution time, energy consumption, explicit joint torque indicators, and robustness margins. In particular, including torque-based objectives directly in the optimization problem would allow the Pareto front to capture both contact-level force sharing and joint-level mechanical loading.

Author Contributions: M.P.-Y.: conceptualization, methodology, software, validation, writing—original draft preparation, writing—review and editing. J.-E.S.-G.: conceptualization, methodology, formal analysis, supervision, software, writing—original draft preparation, writing—review and editing. M.S.-P.: validation, writing—review and editing, supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the European Commission, under the European Project MANiBOT, grant number 101120823.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors acknowledge the use of AI-based language tools to assist with English language polishing and stylistic improvements. The authors acknowledge the use of ChatGPT (OpenAI, GPT-5.6) solely for English-language polishing and stylistic improvements. The authors take full responsibility for the content of the manuscript.

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

References

- Jo, Y.; Park, Y.; Seol, J.; Pak, J.; Kim, B.; Kim, C.; Son, H.I. A Review on Dual-Arm Manipulation in Agriculture. *IEEE Access* **2025**, *13*, 150379–150399.
- Singh, G.; Kalwar, S.; Karim, M.F.; Sen, B.; Govindan, N.; Sridhar, S.; Krishna, K.M. Constrained 6-dof grasp generation on complex shapes for improved dual-arm manipulation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Abu Dhabi, United Arab Emirates, 14–18 October 2024; IEEE: New York, NY, USA, 2024; pp. 7344–7350.
- Qin, Y.; Escande, A.; Kanehiro, F.; Yoshida, E. Dual-arm mobile manipulation planning of a long deformable object in industrial installation. *IEEE Robot. Autom. Lett.* **2023**, *8*, 3039–3046.
- Huang, H.; Zeng, C.; Cheng, L.; Yang, C. Toward generalizable robotic dual-arm flipping manipulation. *IEEE Trans. Ind. Electron.* **2023**, *71*, 4954–4962.
- Sun, F.; Chen, Y.; Wu, Y.; Li, L.; Ren, X. Motion planning and cooperative manipulation for mobile robots with dual arms. *IEEE Trans. Emerg. Top. Comput. Intell.* **2022**, *6*, 1345–1356.
- Peñacoba, M.; Sierra-García, J.E.; Santos, M.; Mariolis, I. Path Optimization Using Metaheuristic Techniques for a Surveillance Robot. *Appl. Sci.* **2023**, *13*, 11182. <https://doi.org/10.3390/app132011182>.
- Lv, N.; Liu, J.; Jia, Y. Dynamic modeling and control of deformable linear objects for single-arm and dual-arm robot manipulations. *IEEE Trans. Robot.* **2022**, *38*, 2341–2353.
- Abbas, M.; Narayan, J.; Dwivedy, S.K. A systematic review on cooperative dual-arm manipulators: Modeling, planning, control, and vision strategies: M. Abbas et al. *Int. J. Intell. Robot. Appl.* **2023**, *7*, 683–707.
- Li, K.; Zhang, H.; Chen, B.; Jiang, Y.; Yang, C.; Wang, Y. Modified dynamic movement primitives with control: A novel trajectory planning framework for dual-arm asynchronous tasks. *IEEE Trans. Ind. Electron.* **2025**, *72*, 10401–10410.
- Qian, L.; Hao, L.; Cui, S.; Gao, X.; Zhao, X.; Li, Y. Research on motion trajectory planning and impedance control for dual-arm collaborative robot grinding tasks. *Appl. Sci.* **2025**, *15*, 819.
- Yang, W.T.; Chen, B.H.; Lin, P.C. A dual-arm manipulation strategy using position/force errors and Kalman filter. *Trans. Inst. Meas. Control.* **2022**, *44*, 820–834.
- Zhang, Y.; Jia, Y. Motion planning of redundant dual-arm robots with multicriterion optimization. *IEEE Syst. J.* **2023**, *17*, 4189–4199.
- Peñacoba-Yagüe, M.; Sierra-García, J.-E.; Santos-Peñas, M. Bio-Inspired Metaheuristics for Time-Optimal Trajectory Planning in Cooperative Dual-Arm Bimanipulation. *Biomimetics* **2026**, *11*, 173.
- Fan, Y.; Peng, Y.; Liu, J. Advanced multi-objective trajectory planning for robotic arms using a multi-strategy enhanced NSGA-II algorithm. *PLoS ONE* **2025**, *20*, e0324567.
- Yagüe, M.P.; García, J.E.S.; Peñas, M.S. Optimización multiobjetivo mediante enjambre de partículas para la reducción del desgaste articular en manipuladores industriales. *Rev. Iberoam. Automática Informática Ind.* **2026**, *23*, 125–136.
- Rizk, Y.; Awad, M.; Tunstel, E.W. Cooperative heterogeneous multi-robot systems: A survey. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–31.
- Sepúlveda, D.; Fernández, R.; Navas, E.; Armada, M.; González-De-Santos, P. Robotic aubergine harvesting using dual-arm manipulation. *IEEE Access* **2020**, *8*, 121889–121904.
- Zhou, C.; Jiang, R.; Luan, F.; Meng, S.; Wang, Z.; Dong, Y.; Zhou, Y.; He, B. Dual-arm robotic fabric manipulation with quasi-static and dynamic primitives for rapid garment flattening. *IEEE/ASME Trans. Mechatron.* **2025**, *30*, 5845–5855.
- Smith, C.; Karayiannidis, Y.; Nalpantidis, L.; Gratal, X.; Qi, P.; Dimarogonas, D.V.; Kragic, D. Dual arm manipulation—A survey. *Robot. Auton. Syst.* **2012**, *60*, 1340–1353.
- Wang, Y.; Smith, C.; Karayiannidis, Y.; Ögren, P. Whole body control of a dual-arm mobile robot using a virtual kinematic chain. *Int. J. Humanoid Robot.* **2016**, *13*, 1550047.
- Park, C.; Park, K. Design and kinematics analysis of dual arm robot manipulator for precision assembly. In *2008 6th IEEE International Conference on Industrial Informatics, Daejeon, Republic of Korea, 13–16 July 2008*; IEEE: New York, NY, USA, 2008; pp. 430–435.
- Mu, Y.; Chen, T.; Peng, S.; Chen, Z.; Gao, Z.; Zou, Y.; Lin, L.; Xie, Z.; Luo, P. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). In *European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024*; Springer Nature: Cham, Switzerland, 2024; pp. 264–273.
- Wu, Y.; Jia, X.; Li, T.; Liu, J. A real-time collision avoidance method for redundant dual-arm robots in an open operational environment. *Robot. Comput.-Integr. Manuf.* **2025**, *92*, 102894.
- Yu, M.; Lv, K.; Wang, C.; Jiang, Y.; Tomizuka, M.; Li, X. Generalizable whole-body global manipulation of deformable linear objects by dual-arm robot in 3-d constrained environments. *Int. J. Robot. Res.* **2025**, *44*, 607–639.

25. Liu, R.; Pan, F. A multi-objective trajectory planning method of the dual-arm robot for cabin docking based on the modified cuckoo search algorithm. *Machines* **2024**, *12*, 64.
26. Yagüe, M.P.; García, J.E.S.; Penas, M.S. Human-intelligent trajectory optimization for robotic manipulators with hybrid PSO-PS algorithm. *Adv. Eng. Inform.* **2026**, *69*, 103941.
27. Lopez-Franco, C.; Diaz, D.; Hernandez-Barragan, J.; Arana-Daniel, N.; Lopez-Franco, M. A metaheuristic optimization approach for trajectory tracking of robot manipulators. *Mathematics* **2022**, *10*, 1051.
28. Rodríguez-Molina, A.; Herroz-Herrera, A.; Aldape-Pérez, M.; Flores-Caballero, G.; Antón-Vargas, J.A. Dynamic path planning for the differential drive mobile robot based on online metaheuristic optimization. *Mathematics* **2022**, *10*, 3990.
29. Sharma, S.; Kumar, V. A Comprehensive Review on Multi-objective Optimization Techniques: Past, Present and Future: S. Sharma, V. Kumar. *Arch. Comput. Methods Eng.* **2022**, *29*, 5605–5633.
30. Li, W.; Zhang, T.; Wang, R.; Huang, S.; Liang, J. Multimodal multi-objective optimization: Comparative study of the state-of-the-art. *Swarm Evol. Comput.* **2023**, *77*, 101253.
31. Lu, Z.; Zhao, Z.; Long, L.; Ma, Y.; LeiLi, L.; Liu, Z.; Dai, F.; Zhang, Y.; Li, J. Multi-robot task allocation in agriculture scenarios based on the improved NSGA-II algorithm. In *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, Hong Kong, China, 10–13 October 2023; IEEE: New York, NY, USA, 2023; pp. 1–6.
32. Liu, S.; Tian, Q.; Tang, C. Mobile robot path planning algorithm based on NSGA-II. *Appl. Sci.* **2024**, *14*, 4305.
33. Lalwani, S.; Singhal, S.; Kumar, R.; Gupta, N. A comprehensive survey: Applications of multi-objective particle swarm optimization (MOPSO) algorithm. *Trans. Comb.* **2013**, *2*, 39–101.
34. Coello, C.C.; Lechuga, M.S. MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, Honolulu, HI, USA, 12–17 May 2002; IEEE: New York, NY, USA, 2002; Volume 2, pp. 1051–1056.
35. Zou, W.; Xu, H.; Chen, C.; Wu, C. An Enhanced MOPSO Algorithm for Multi-Objective UAV Path Planning in Mountainous Environments. *Symmetry* **2025**, *17*, 1890.
36. Zhang, Y.; Mu, X. An inexact multiblock alternating direction method for grasping-force optimization of multifingered robotic hands. *J. Inequalities Appl.* **2023**, *2023*, 30.

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