

Optimización multiobjetivo mediante enjambre de partículas para la reducción del esfuerzo mecánico en manipuladores industriales

Mario Peñacoba Yagüe^{a,*}, Jesús Enrique Sierra-García^a, Matilde Santos Peñas^b

^a Departamento de Digitalización, Universidad de Burgos, Campus Río Vena, Avda. Cantabria, s/n, 09006 Burgos, España.

^b Instituto de Tecnología del Conocimiento, Universidad Complutense de Madrid, Madrid, España.

To cite this article: Peñacoba Yagüe, M., Sierra García, J. E., Santos Peñas, M., 2026. Multi-objective trajectory optimization using particle swarm for the reduction of mechanical effort in industrial manipulators. *Revista Iberoamericana de Automática e Informática Industrial* 23, 125-136. <https://doi.org/10.4995/riai.2026.24977>

Resumen

La planificación de trayectorias en manipuladores industriales sigue siendo un desafío cuando se busca equilibrar simultáneamente productividad, suavidad de movimiento y durabilidad de los actuadores. En particular, los métodos tradicionales tienden a centrarse en un número reducido de métricas, lo que limita la capacidad de evaluar el rendimiento del robot desde una perspectiva integral. Este trabajo propone una metodología inteligente de optimización multiobjetivo para la generación de trayectorias, aplicada al robot colaborativo ABB GoFa5. La aproximación se basa en un modelo tridimensional realista del robot, que reproduce su comportamiento dinámico e incorpora las propiedades físicas de los eslabones, junto con la detección de colisiones tanto con el entorno como con el propio manipulador. La optimización se realiza mediante un algoritmo de enjambre de partículas multiobjetivo, considerando tres configuraciones experimentales: (i) minimización del tiempo de trayectoria y de los pares articulares medios, (ii) optimización del tiempo de trayectoria junto con las velocidades máximas de cada articulación, y (iii) una formulación ampliada que integra tiempo de trayectoria, pares medios y velocidades máximas, alcanzando un total de trece objetivos. Los resultados demuestran que la metodología propuesta facilita la identificación de trayectorias óptimas en función del criterio de interés, ya sea minimizar el tiempo de trayectoria, reducir las velocidades articulares máximas, o prolongar la vida útil del manipulador, garantizando en todos los casos soluciones suaves, libres de colisiones y físicamente eficientes.

Palabras clave: Optimización Multiobjetivo, Planificación y seguimiento de trayectorias, Robótica industrial, Manipulación robótica, Técnicas de control inteligente.

Multi-objective trajectory optimization using particle swarm for the reduction of mechanical effort in industrial manipulators.

Abstract

Trajectory planning in industrial manipulators remains a challenge when attempting to simultaneously balance productivity, motion smoothness, and actuator durability. Traditional approaches often focus on a limited set of metrics, which restricts the ability to comprehensively evaluate robot performance. This work proposes an intelligent multi-objective optimization methodology for trajectory generation, applied to the collaborative robot ABB GoFa5. The approach combines MATLAB-based simulation with a detailed dynamic model of the robot, incorporating the physical properties of its links and enabling collision detection with both the environment and the manipulator itself. Optimization is carried out using a multi-objective particle swarm algorithm, considering three experimental configurations: (i) minimization of execution time and average joint torques, (ii) optimization of execution time together with the maximum velocities reached by each joint, and (iii) an extended formulation that integrates execution time, average joint torques, and maximum velocities, for a total of thirteen objectives. The results demonstrate that the proposed methodology enables the identification of optimal trajectories depending on the prioritized criterion—whether minimizing execution time, reducing maximum joint velocities, or extending manipulator lifespan—while ensuring in all cases smooth, collision-free, and physically efficient solutions.

*Autor para correspondencia: mpenacoba@ubu.es

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

Keywords: Multi-objective optimization, Trajectory planning and tracking, Industrial robotics, Robotic manipulation, Intelligent control techniques.

1. Introducción

La planificación de trayectorias constituye uno de los pilares fundamentales en la programación de manipuladores industriales, ya que condiciona de manera directa la eficiencia de los procesos productivos, la calidad del movimiento y la vida útil de los actuadores. En los entornos de fabricación actuales, caracterizados por elevados niveles de exigencia y una creciente demanda de flexibilidad, no basta con garantizar la viabilidad geométrica de las trayectorias: resulta igualmente necesario optimizar parámetros dinámicos y energéticos que influyen de forma crítica en el rendimiento global del sistema.

Los métodos tradicionales de planificación, basados principalmente en formulaciones cinemáticas y en modelos simplificados del robot, han demostrado eficacia en aplicaciones convencionales (Martín-H. et al., 2009; Olayo, 2022). Sin embargo, presentan limitaciones notables cuando se requiere evaluar simultáneamente métricas como el tiempo de trayectoria, los pares articulares, o las velocidades máximas alcanzadas en cada articulación (Liu & Liu, 2022; Peñacoba et al., 2025). Esta restricción impide disponer de una visión integral del desempeño del robot y limita la posibilidad de adaptar las trayectorias a diferentes prioridades operativas, como maximizar la productividad, reducir el consumo energético o prolongar la durabilidad de los actuadores (Prado et al., 2025).

En este contexto, los algoritmos de optimización bioinspirados han emergido como una alternativa robusta para resolver problemas de planificación de elevada complejidad (Peñacoba et al., 2025). Entre ellos, los métodos basados en enjambres de partículas (Particle Swarm Optimization, PSO) han mostrado un notable potencial para abordar formulaciones multiobjetivo, gracias a su capacidad de explorar espacios de búsqueda de alta dimensionalidad y de generar frentes que equilibran criterios contrapuestos sin requerir ponderaciones previas. Esto resulta especialmente relevante en robótica industrial, donde es habitual encontrar objetivos en conflicto que deben gestionarse de manera simultánea (Alonso & Santos, 2017). Entre las distintas variantes, el algoritmo Multi-Objective Particle Swarm Optimization (MOPSO) se considera especialmente adecuado por su bajo número de parámetros, su eficacia en la exploración global y su eficiencia computacional, lo que lo convierte en una alternativa competitiva frente a técnicas como NSGA-II o MOEA/D (Verma et al., 2021; Cao et al., 2021).

El presente trabajo introduce una metodología de optimización multiobjetivo aplicada al robot colaborativo ABB CRB15000-95. La propuesta combina un modelo dinámico detallado del manipulador con un entorno de simulación que permite considerar tanto las propiedades físicas de los eslabones como la detección de colisiones con el entorno y entre partes del propio robot. La optimización se aborda mediante un algoritmo de enjambre de partículas multiobjetivo configurado para resolver tres escenarios experimentales: (i) minimización del tiempo de trayectoria junto con los pares articulares medios, (ii) tiempo de trayectoria y velocidades máximas alcanzadas en cada articulación, y (iii) una

formulación ampliada que integra de manera simultánea trece objetivos.

Este trabajo se enmarca en el proyecto europeo MANiBOT, orientado al desarrollo de plataformas robóticas bimanuales colaborativas capaces de operar en entornos industriales y logísticos con elevados requisitos de seguridad y flexibilidad (Macías et al., 2024; Bayona et al., 2025). La metodología planteada en este artículo se ha validado en uno de los escenarios definidos dentro del proyecto: un caso de manipulación en el entorno aeroportuario, donde el robot debe planificar trayectorias seguras y eficientes para interactuar con objetos voluminosos en espacios restringidos. Este caso de uso resulta especialmente representativo, al combinar restricciones geométricas complejas con exigencias de productividad y robustez propias del sector logístico aeroportuario (MANiBOT, 2023).

El artículo se estructura de la siguiente manera: la Sección 2 describe el funcionamiento del algoritmo de enjambre de partículas multiobjetivo y su configuración; en la Sección 3 se plantea el problema de optimización; en la Sección 4 se presenta el caso de uso; la Sección 5 describe el modelado del sistema; la Sección 6 recoge los resultados experimentales y, finalmente, en la Sección 7 y 8 se expone la discusión y las conclusiones principales respectivamente.

2. Algoritmo de optimización multiobjetivo mediante enjambre de partículas

La optimización de trayectorias en este trabajo se realiza mediante el algoritmo Multi-Objective Particle Swarm Optimization (MOPSO), una extensión del PSO clásico diseñada para abordar problemas con múltiples (típicamente dos o tres) objetivos en conflicto (MOO) (Ghaffour, 2024; Yang et al., 2025).

En optimización multiobjetivo, a diferencia de la optimización monoobjetivo, no existe un único óptimo global, sino un conjunto de soluciones no dominadas que representan diferentes compromisos entre los objetivos. Por ello, el MOPSO incorpora extensiones clave respecto al PSO clásico que permiten gestionar y preservar dicho conjunto de soluciones (Peñacoba et al., 2023; Bayona et al., 2024):

- I. **Repositorio de soluciones no dominadas:** se mantiene un conjunto A que almacena todas las soluciones no dominadas encontradas hasta el momento. En cada iteración, este repositorio se actualiza combinando las soluciones previamente almacenadas A^t con las nuevas candidatas generadas por el enjambre X^{t+1} , filtrando aquellas que son dominadas por otras en el espacio de objetivos. Este proceso de actualización de la matriz A^t se expresa matemáticamente según la ecuación 1.

$$A^{t+1} = \{x \in A^t \cup X^{t+1} | \nexists y \in A^t \cup X^{t+1}: y < x\} \quad (1)$$

Donde $y < x$ indica que la solución y domina a x en todos los objetivos. Es decir, y no es peor que x en todos los objetivos y es estrictamente mejor en al menos uno. De este modo, el repositorio A^{t+1} contiene únicamente las soluciones no dominadas del conjunto

combinado entre el repositorio anterior y las soluciones recién generadas, manteniendo una representación actualizada y aproximada del frente de Pareto a lo largo de las iteraciones (Duan et al., 2024).

- II. **Selección de líderes (líder global):** en el PSO clásico, el líder global g_i^t es único, mientras que en MOPSO se elige aleatoriamente un líder $g_i^{(t)}$ del conjunto A^t , siguiendo un criterio de diversidad. Por ejemplo, mediante un sistema de distancia de aglomeración para mantener una distribución homogénea de soluciones en el frente de Pareto.
- III. **Actualización de sub-frentes locales:** cada partícula i mantiene un subconjunto S_i de soluciones no dominadas individuales, que se actualiza en cada iteración según la ecuación 2.

$$S_i^{t+1} = \{x \in S_i^t \cup x_i^{t+1} | \nexists y \in S_i^t \cup x_i^{t+1}: y < x\} \quad (2)$$

- IV. **Mantenimiento de diversidad:** para evitar la convergencia prematura hacia una región concreta del espacio de Pareto, se pueden aplicar operadores de mutación controlada o técnicas de reubicación aleatoria sobre partículas dominadas.

En otras palabras, el algoritmo MOPSO, a diferencia del PSO clásico, gestiona simultáneamente un repositorio global de soluciones no dominadas (A) y los subconjuntos locales S_i^{t+1} , actualizados iterativamente para aproximar de forma progresiva el frente de Pareto. De este modo, el MOPSO permite obtener trayectorias de compromiso bajo múltiples criterios; en este trabajo, por ejemplo, se emplea para minimizar simultáneamente el tiempo de trayectoria y el par articular medio, manteniendo el equilibrio entre exploración y explotación característico del enfoque (Yang et al., 2022; Lalwani et al., 2013).

3. Metodología de optimización

La metodología desarrollada se basa en una arquitectura modular que integra un algoritmo de optimización multiobjetivo (MOPSO) con un entorno de simulación en el que se integra un manipulador robótico, permitiendo la evaluación iterativa de trayectorias y la búsqueda de soluciones óptimas según múltiples criterios. Como se muestra en la Figura 1, el proceso se inicia con la definición del problema, donde se establecen las variables de decisión, los límites de las mismas y la trayectoria inicial. A continuación, mediante el cálculo del modelo cinemático inverso del manipulador, se calcula la trayectoria en coordenadas articulares, que se ejecuta en el bloque de simulación externa, encargado de calcular los indicadores de desempeño de cada trayectoria, como por ejemplo el tiempo de trayectoria, el par articular o las posibles colisiones con el entorno. Los resultados obtenidos se vuelcan en forma de funciones de coste individuales al optimizador multiobjetivo, que actualiza las soluciones conforme a su estrategia interna (exploración y explotación del espacio de búsqueda). Este ciclo se repite hasta alcanzar los criterios de convergencia definidos, almacenando las soluciones más prometedoras y seleccionando finalmente

aquella que presenta el mejor compromiso entre los objetivos evaluados.

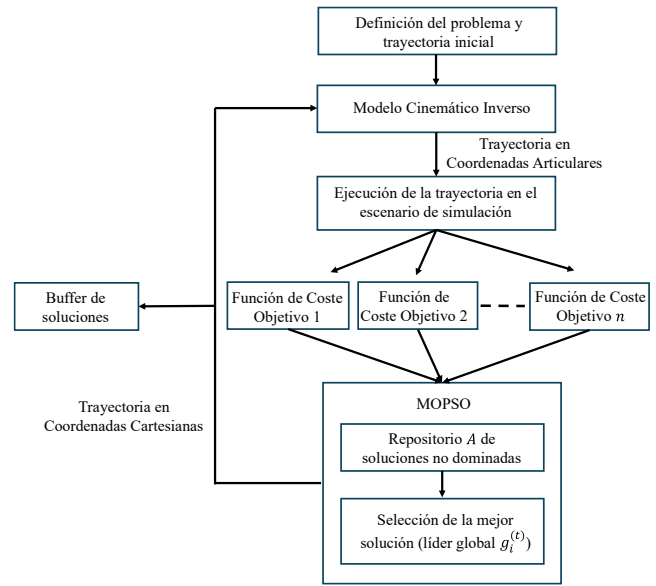


Figura 1: Metodología de la herramienta de optimización multiobjetivo.

La estructura propuesta permite una integración flexible con cualquier entorno de simulación o robot, adaptándose fácilmente a distintos escenarios y casos de uso.

3.1. Variables de decisión y control del manipulador

El problema de generación de trayectorias robóticas puede definirse en dos espacios distintos: el espacio articular y el espacio cartesiano. Trabajar en el espacio articular permite evitar posibles problemas de singularidades y discontinuidades en las configuraciones del robot, aunque las restricciones y los objetivos suelen resultar menos intuitivos de formular. Por el contrario, una definición en el espacio cartesiano ofrece restricciones más interpretables, lo que facilita la integración con el entorno de simulación y la definición de objetivos geométricos o cinemáticos relacionados con la tarea (Peñacoba et al., 2025).

En este trabajo se adopta este segundo enfoque, definiendo las variables de decisión en el espacio cartesiano. Por tanto, la trayectoria del efector final se describe a partir de un conjunto de puntos de paso definidos mediante sus coordenadas y orientaciones $[x_i, y_i, z_i, \phi_i, \theta_i, \psi_i]$ cuyas restricciones se definen en el punto 3.2.

Para validar cada trayectoria candidata, los puntos definidos en el espacio cartesiano deben convertirse en configuraciones articulares mediante cinemática inversa, según la Ecuación (3), donde q_i representa el vector articular asociado al efector final (Bugarín & Aguilar-Bustos, 2025; Rascón et al., 2024).

$$q_i = g_i(x, y, z, \phi, \theta, \psi) \quad (3)$$

Como no existe una solución analítica sencilla a este problema, el proceso de conversión de puntos cartesianos en configuraciones articulares se resuelve punto a punto a lo largo de la trayectoria interpolada, empleando un solucionador iterativo de cinemática inversa basado en técnicas de optimización numérica. Este método ajusta los valores

articulares para minimizar el error de posición y orientación entre la pose deseada y la pose obtenida mediante cinemática directa, asegurando la coherencia entre configuraciones consecutivas.

$$q_i \cong g_{it}(q^{semilla}, x, y, z, \phi, \theta, \psi) \quad (4)$$

Como se muestra en la Ecuación (4), el método iterativo necesita un punto inicial de búsqueda $q^{semilla}$. Con el fin de evitar saltos bruscos o la aparición de soluciones próximas a singularidades, se aplica un esquema multi-semilla. En cada iteración se generan varias configuraciones iniciales (semillas) en torno a la solución anterior q_{i-1} mediante una perturbación gaussiana, como se muestra en la Ecuación (5).

$$q_k^{semilla} = q_{i-1} + \varepsilon_k, \varepsilon_k \sim \mathcal{N}(0, \sigma^2 I) \quad (5)$$

Donde $k = 1, \dots, N_s$ representa el número de semillas consideradas (en este estudio, $N_s = 5$). Cada semilla se emplea como punto inicial, $q_k^{semilla}$, para el solucionador de cinemática inversa de la Ecuación (5), generando un conjunto de configuraciones candidatas $\{q_k^{cand}\}$.

De entre todas las soluciones obtenidas, se selecciona aquella que minimiza la distancia articular respecto a la configuración previa, según la Ecuación (6).

$$q_i = \underset{q_k^{cand}}{\operatorname{argmin}} \|q_k^{cand} - q_{i-1}\|_2 \quad (6)$$

Este criterio de selección garantiza una evolución suave de los valores articulares a lo largo de la trayectoria, reduciendo discontinuidades y evitando configuraciones no físicas o inestables.

En este estudio se emplea la norma euclídea estándar para medir la distancia entre configuraciones, aunque podrían incorporarse métricas ponderadas que consideren la sensibilidad cinemática o los rangos articulares de cada eje, mejorando así la equidad en el cálculo de proximidad.

3.2. Restricciones geométricas y físicas

La optimización de la trayectoria se realiza bajo un conjunto de restricciones geométricas y físicas que delimitan el espacio de búsqueda del algoritmo. Dentro de este conjunto, ciertos puntos permanecen fijos para satisfacer las condiciones impuestas por la tarea. Por ejemplo, en una tarea de pick and place, el punto inicial (“pick”) y final (“place”) deben permanecer invariantes, mientras que los puntos intermedios pueden ser optimizados y, por ende, constituyen las variables de decisión del problema.

Debido a la naturaleza tridimensional del problema, si se consideran n puntos intermedios optimizables, se genera un total de $6 * n$ variables continuas, correspondientes a las tres coordenadas espaciales y tres ángulos de orientación que describen los seis grados de libertad (DoF) del manipulador. En este contexto, el algoritmo MOPSO modifica las coordenadas de los puntos no fijos dentro de unos límites superior (ub) e inferior (lb) previamente definidos, de modo que cada partícula de la población se encuentre dentro del volumen operativo alcanzable por el robot.

La correcta definición de estos límites resulta fundamental, ya que determina directamente la eficiencia y el rendimiento del proceso de optimización. Límites más amplios incrementan el espacio de búsqueda y permiten explorar un conjunto mayor de soluciones potenciales, aumentando las probabilidades de hallar un resultado óptimo global, pero también incrementan el coste computacional y el tiempo necesario para la convergencia. Por el contrario, restricciones más estrechas reducen el tiempo de cómputo, aunque pueden limitar la diversidad de trayectorias evaluadas y, en consecuencia, la calidad de la solución final.

3.3. Formulación de las funciones de coste

El proceso de optimización multiobjetivo propuesto en este trabajo se fundamenta en un conjunto de métricas físicas independientes, cada una asociada a un objetivo específico de rendimiento del manipulador. Una característica distintiva de la metodología desarrollada es que cada función de coste individual incorpora dos fases complementarias (Ecuación 7):

- I. Una fase de validación geométrica, en la que el algoritmo detecta y penaliza automáticamente las trayectorias que presentan colisiones, tanto con el entorno como con el propio robot (primera fila de la Ecuación 7).
- II. Una fase de optimización, centrada en la mejora de las métricas de desempeño asociadas al objetivo considerado (segunda fila en Ecuación 7).

Esta formulación bifásica permite al sistema descartar trayectorias no viables antes de proceder a la optimización, garantizando que el proceso de búsqueda se realice exclusivamente sobre soluciones geoméricamente factibles y físicamente eficientes.

$$f_c = \begin{cases} 1 + t_{colisión} & \text{si } t_{colisión} > 0 \\ f_{física} & \text{si } t_{colisión} = 0 \end{cases} \quad (7)$$

En la primera fase, el algoritmo evalúa la validez geométrica de la trayectoria. Cuando se detectan colisiones, la función incrementa su valor en proporción al tiempo total de colisión $t_{colisión}$, penalizando las trayectorias no viables de acuerdo con su grado de interferencia con el entorno. En la segunda fase, una vez garantizada la ausencia de colisiones $t_{colisión} = 0$, el objetivo se centra en minimizar una métrica de desempeño, $f_{física}$.

Minimización del tiempo de ejecución de la trayectoria

El primer objetivo es reducir el tiempo total de ejecución de la trayectoria, $t_{trayectoria}$, garantizando también que el movimiento del manipulador sea geoméricamente viable. Esta variable constituye una medida directa de la eficiencia temporal y la productividad del sistema robótico.

La función de coste asociada a este objetivo se muestra en la Ecuación 8.

$$f_{c_{tt}} = \begin{cases} 1 + t_{colisión} & \text{si } t_{colisión} > 0 \\ \frac{t_{trayectoria}}{t_{trayectoria}^{max}} & \text{si } t_{colisión} = 0 \end{cases} \quad (8)$$

Para ello, el tiempo se normaliza en el rango $[0,1]$ dividiéndolo entre el valor máximo $t_{trayectoria}^{max}$, que actúa como referencia de escala. De este modo, al tender el algoritmo a minimizar el valor global de la función de coste, aprende de forma autónoma a generar trayectorias factibles y cada vez más eficientes temporalmente, combinando viabilidad geométrica y optimización del rendimiento.

Minimización del par máximo articular

La segunda métrica se centra en minimizar los pares máximos articulares generados durante la ejecución de la trayectoria. Esta métrica puede conformar desde un solo objetivo (evaluación de una única articulación) hasta un grupo de tantos objetivos como articulaciones tenga el manipulador. De ahora en adelante se asumirá un grupo de seis objetivos, pues coincide con el número de articulaciones del brazo robótico con el que se han realizado las pruebas.

Esta métrica refleja el esfuerzo mecánico requerido por el manipulador para completar el movimiento, y se relaciona directamente con la durabilidad del sistema, el consumo energético y la suavidad dinámica de la operación.

La reducción de los picos de par permite prolongar la vida útil del robot, disminuir la fatiga de los actuadores y mejorar la estabilidad general del movimiento. La función de coste fc_{Ti} para cada articulación i correspondiente a este objetivo se define en la Ecuación 9, donde N_a es el número de articulaciones totales.

$$fc_{Ti} = \begin{cases} 1 + \frac{t_{colisión}}{T_i^{max}} & \text{si } t_{colisión} > 0 \\ \frac{T_i^{max}}{T_i^{lim}} & \text{si } t_{colisión} = 0 \end{cases}, i = 1..N_a \quad (9)$$

En la segunda fase, una vez garantizada la viabilidad geométrica $t_{colisión} = 0$, la función evalúa el nivel de esfuerzo articular mediante el cociente entre el par máximo observado en la articulación (T_i^{max}) y los límites admisibles del robot para esa articulación (T_i^{lim}).

Minimización de la velocidad máxima articular

El tercer objetivo se orienta a minimizar las velocidades máximas articulares alcanzadas durante la trayectoria. Esta métrica está directamente relacionada con la suavidad del movimiento, la estabilidad dinámica del manipulador y la reducción de vibraciones o esfuerzos transitorios en los actuadores.

Al limitar la velocidad máxima de cada articulación, se favorece un desplazamiento más controlado, preciso y seguro, evitando picos cinemáticos que podrían comprometer la integridad mecánica del sistema. La función de coste asociada fc_{Vi} para cada articulación i se formula en dos fases complementarias (Ecuación 10), donde N_a es el número de articulaciones totales.

$$fc_{Vi} = \begin{cases} 1 + \frac{t_{colisión}}{V_i^{max}} & \text{si } t_{colisión} > 0 \\ \frac{V_i^{max}}{V_i^{lim}} & \text{si } t_{colisión} = 0 \end{cases}, i = 1..N_a \quad (10)$$

En la segunda fase, una vez garantizada la ausencia de colisiones, la función de coste se centra en reducir la velocidad máxima articular (V_i^{max}) con respecto al valor máximo

operativo (V_i^{lim}). El cociente $\frac{V_i^{max}}{V_i^{lim}}$ permite normalizar el valor de velocidad máximo alcanzado en cada articulación (V_i^{max}) respecto a su límite operativo (V_i^{lim}), favoreciendo trayectorias que mantengan las velocidades dentro de márgenes seguros.

3.4. Escenarios de optimización

Como se trata de una técnica de optimización multiobjetivo, se pueden seleccionar diferentes agrupaciones de las funciones de coste definidas anteriormente. Si solo se desea reducir el tiempo de trayectoria, se emplea exclusivamente fc_{tt} . Si se desea reducir el tiempo de trayectoria, y la velocidad en las articulaciones se emplean $[fc_{tt}, fc_{V1}, \dots, fc_{VN_a}]$, con un total de $N_a + 1$ objetivos. Si se desea reducir el tiempo de trayectoria, y el par articular se emplean $[fc_{tt}, fc_{T1}, \dots, fc_{TN_a}]$, con un total de $N_a + 1$ objetivos. Si se busca reducir simultáneamente tiempo, velocidad y par, se emplean $[fc_{tt}, fc_{T1}, \dots, fc_{TN_a}, fc_{V1}, \dots, fc_{VN_a}]$, con un total de $2N_a + 1$.

De esta manera, el algoritmo buscaría minimizar el valor del par máximo y velocidad articular, dirigiendo la búsqueda hacia trayectorias que reduzcan el esfuerzo mecánico global sin comprometer la factibilidad geométrica ni la eficiencia temporal. El resultado es un movimiento más estable, sostenible y energéticamente eficiente, que optimiza el uso de los actuadores y mejora el comportamiento dinámico del manipulador.

Finalmente se podrían plantear otras agrupaciones eliminando objetivos asociados a articulaciones que no resulten necesarios, o se encuentren muy lejos de sus restricciones físicas y no resulten relevantes.

4. Caso de uso

El presente trabajo se enmarca en los escenarios experimentales del proyecto europeo MANiBOT, orientado al desarrollo y validación de una plataforma robótica móvil bimanual para tareas de manipulación colaborativa en entornos reales. En este contexto, se consideran distintos casos de uso industriales (supermercados y aeropuertos) para evaluar la robustez, adaptabilidad y seguridad del sistema.

En particular, este estudio se centra en el escenario aeroportuario de manipulación de equipajes (Peñacoba et al., 2025). El ciclo de trabajo se compone de dos operaciones complementarias: carga de bultos desde una cinta transportadora a un carro (Figura 2a) y descarga desde el carro hacia la cinta (Figura 2b).



Representación de la carga de equipaje desde una cinta transportadora a un carro porta equipajes.



Representación de la carga de equipaje desde un carro porta equipajes a una cinta transportadora.

Figura 2: Representación del caso de uso

En este artículo, el caso se utiliza como banco de pruebas para evaluar la optimización de trayectorias en términos de tiempo de ejecución y métricas articulares (pares y velocidades) asociadas a la sollicitación mecánica y la suavidad del movimiento.

5. Modelado del sistema e implementación de la solución

El entorno de simulación empleado para el desarrollo y evaluación de la metodología se implementó íntegramente en MATLAB R2024b, aprovechando las capacidades de la *Robotics System Toolbox* para la representación cinemática y dinámica de manipuladores industriales.

El sistema permite generar, ejecutar y evaluar trayectorias candidatas de forma automatizada, integrando en un mismo entorno la resolución de la cinemática inversa, la detección de colisiones y el cálculo del esfuerzo articular.

El modelo robótico utilizado corresponde al ABB GoFa 5, importado mediante un archivo URDF que incluye la definición estructural del manipulador y las propiedades dinámicas asociadas a cada eslabón (masa, volumen e inercia).

A partir de este modelo se construye un entorno tridimensional de trabajo que reproduce la estación del caso de uso aeroportuario, incluyendo los obstáculos fijos relevantes para la tarea. La Figura 3 muestra la escena completa utilizada en los experimentos, formada por el manipulador ABB GoFa 5 y los principales elementos del entorno (mesa/sopORTE de carga, bulto y contenedor/carro), que se emplean como geometrías de referencia para la verificación de colisiones durante la evaluación de las trayectorias.

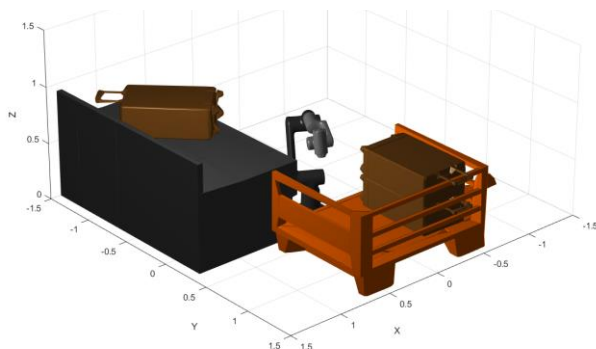


Figura 3: Entorno de simulación del caso de uso aeroportuario: ABB GoFa 5 y escena 3D (mesa/plataforma, bulto y carro/contenedor) utilizada para la evaluación de trayectorias y detección de colisiones.

5.1. Generación e interpolación de trayectorias

En este trabajo se consideran cuatro puntos intermedios optimizables, lo que genera un total de $4 * 6 = 24$ variables continuas. Este número de grados de libertad permite equilibrar adecuadamente la flexibilidad de la trayectoria con el coste computacional del proceso de optimización en el caso de uso aplicado.

Con el objetivo de simplificar el problema y acelerar la convergencia, las orientaciones del efector final $[\phi_i, \theta_i, \psi_i]$ se mantienen constantes en $[0, \pi, 0]$, posición que coincide con la configuración de agarre y depósito del objeto. En consecuencia, el algoritmo MOPSO solo modifica las coordenadas espaciales (x, y, z) de los puntos intermedios.

Cada partícula de la población representa una posible trayectoria candidata, definida por una secuencia específica de puntos de paso en el espacio cartesiano. Esta reducción del número de variables libres contribuye a mejorar la eficiencia computacional sin comprometer la capacidad de adaptación de la trayectoria.

Cada trayectoria generada se obtiene a partir del conjunto de puntos intermedios propuesto por el algoritmo (Palmieri et al., 2024). Para garantizar un movimiento continuo y físicamente realizable, las trayectorias se interpolan mediante polinomios de quinto grado, que garantizan la continuidad de posición, velocidad y aceleración (Ecuación 11).

$$p(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (11)$$

Los coeficientes $a_0, \dots, a_5$ se calculan a partir de las condiciones de contorno de cada segmento —posición, velocidad, y aceleración inicial y final—, lo que permite mantener una transición suave entre los puntos de paso. Este enfoque asegura que el robot pueda seguir las consignas sin generar vibraciones, discontinuidades ni picos bruscos de aceleración, garantizando así un comportamiento dinámicamente estable.

5.2. Verificación de colisiones

La detección de colisiones se realiza de manera continua durante el seguimiento de cada trayectoria. Para ello, se emplea la función `checkCollision()` de la *Robotics System Toolbox*, la cual evalúa en cada instante la distancia mínima entre los cuerpos del robot y los objetos presentes en el entorno. Esta comprobación incluye tanto colisiones externas, es decir, entre el robot y los obstáculos del escenario, como colisiones internas, que podrían producirse entre los distintos eslabones del manipulador.

El procedimiento se ejecuta punto a punto a lo largo de la trayectoria interpolada, de modo que se detecta cualquier interferencia con precisión milimétrica.

Los obstáculos fijos, como estructuras de soporte o zonas de almacenamiento del entorno aeroportuario, se definen previamente según la geometría del caso de uso, mientras que la morfología del robot se gestiona automáticamente a partir de los datos del modelo URDF.

5.3. Evaluación del esfuerzo articular y de las velocidades

Para cuantificar la demanda dinámica impuesta por cada trayectoria, se estima el perfil de par articular necesario para su ejecución. Este cálculo se lleva a cabo mediante el modelo dinámico del robot, utilizando la función `inverseDynamics()` incluida en la *Robotics System Toolbox*.

A partir de las configuraciones articulares obtenidas por cinemática inversa, se determinan las velocidades y aceleraciones de cada eje mediante un esquema de diferencias finitas. Los valores resultantes permiten calcular el par requerido en cada instante, registrando para cada trayectoria dos indicadores clave:

1. Par medio absoluto, que representa el esfuerzo mecánico global sostenido a lo largo del movimiento.

2. Par máximo por articulación, que refleja la máxima exigencia dinámica experimentada por el manipulador.

De manera análoga, se calculan las velocidades máximas articulares, obtenidas directamente a partir del vector de velocidades $\dot{q} = [\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5, \dot{q}_6]$ en cada instante de la trayectoria (Ecuación 12).

$$V^{max} = MAX([\dot{q}]) \quad (12)$$

Estos indicadores: tiempo total, par medio, par y velocidad máximos, se emplean posteriormente en las funciones de coste descritas en la Sección 3, permitiendo comparar de forma objetiva el comportamiento de las trayectorias optimizadas bajo diferentes criterios.

6. Resultados

Con el objetivo de evaluar el desempeño del algoritmo propuesto en distintos contextos de optimización se han diseñado tres configuraciones experimentales dentro del caso de uso aeroportuario, cada una centrada en un conjunto diferente de métricas físicas del manipulador. Estas configuraciones permiten analizar la capacidad del método para equilibrar productividad, suavidad de movimiento y esfuerzo mecánico, observando cómo varían las trayectorias obtenidas en función de los objetivos optimizados. Los resultados presentados corresponden a un caso de estudio representativo en el que se optimiza una trayectoria asociada a la manipulación de un único bulto (una maleta). No obstante, el método es general y puede aplicarse a cualquier tarea ejecutada por un brazo robótico (manipulación de diferentes objetos, inspección, soldadura o pintura, etc.) que sea enunciable como un problema de optimización de trayectoria bajo restricciones, ajustando las condiciones iniciales/finales y las restricciones del entorno.

La Tabla 1 resume los objetivos considerados en cada escenario experimental, indicando explícitamente qué métricas se han optimizado en cada uno de ellos.

Tabla 1: Objetivos en cada escenario de optimización

Objetivo	Escenario 1	Escenario 2	Escenario 3
$t_{trayectoria}$	Si	Si	Si
T_1^{max}	Si	No	Si
T_2^{max}	Si	No	Si
T_3^{max}	Si	No	Si
T_4^{max}	Si	No	Si
T_5^{max}	Si	No	Si
T_6^{max}	Si	No	Si
V_1^{max}	No	Si	Si
V_2^{max}	No	Si	Si
V_3^{max}	No	Si	Si
V_4^{max}	No	Si	Si
V_5^{max}	No	Si	Si
V_6^{max}	No	Si	Si

En la primera configuración, se optimizaron simultáneamente el tiempo total de trayectoria ($t_{trayectoria}$) y los pares máximos articulares ($\vec{T}^{max}$), priorizando la reducción del esfuerzo mecánico sin comprometer la eficiencia temporal.

La segunda configuración combina el tiempo de trayectoria ($t_{trayectoria}$) con las velocidades máximas articulares ($\vec{V}^{max}$), orientándose a obtener trayectorias más suaves y estables. Finalmente, la tercera configuración integra las tres métricas, tiempo de trayectoria ($t_{trayectoria}$), pares máximos ($\vec{T}^{max}$) y velocidades máximas articulares ($\vec{V}^{max}$), buscando un equilibrio global entre rapidez, sostenibilidad mecánica y suavidad cinemática.

Las simulaciones y ejecuciones del algoritmo de optimización se realizaron en un equipo con procesador AMD Ryzen 5 5600H (3.29 GHz), 8 GB de RAM, y sistema operativo de 64 bits. En todos los escenarios se estableció un límite temporal de 3 horas como criterio de parada para la ejecución del MOPSO, con el fin de garantizar una exploración suficiente del frente de Pareto y permitir una comparación homogénea entre distintas configuraciones experimentales.

6.1. Escenario 1: Optimización de tiempo de trayectoria y par articular

En la Figura 4 se muestra la comparativa de trayectorias obtenidas en el primer escenario, donde se consideraron como objetivos el tiempo total de ejecución y los pares máximos por articulación. Se representan tres soluciones relevantes:

1. La primera trayectoria libre de colisiones encontrada por el algoritmo (PLC).
2. La trayectoria con menor tiempo de trayectoria (MinT).
3. La trayectoria con menor media de pares máximos por articulación (MinPM).

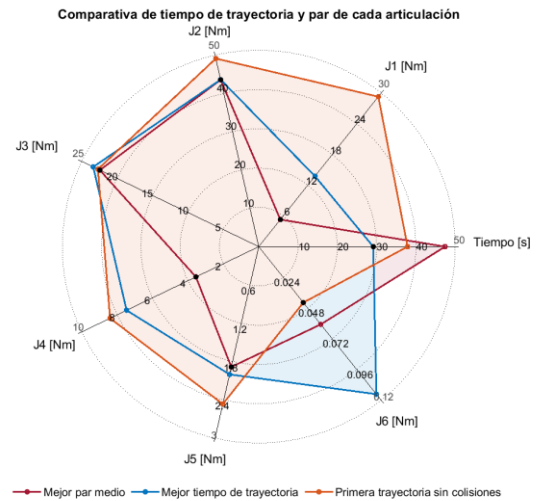


Figura 4. Comparativa de tiempo de trayectoria y par máximo de cada articulación.

En términos generales, la Figura 5 evidencia el compromiso esperado entre productividad y sollicitación mecánica: las soluciones que reducen el tiempo tienden a incrementar la exigencia dinámica en algunas articulaciones, mientras que las soluciones que minimizan el esfuerzo articular aceptan recorridos menos directos y, por tanto, mayores tiempos de ejecución. En particular, la trayectoria MinT prioriza un desplazamiento más directo, lo que se traduce en una reducción clara del tiempo, pero concentra mayores demandas de par en articulaciones clave para orientar el brazo durante el

movimiento. Por el contrario, MinPM redistribuye el movimiento entre articulaciones y evita cambios bruscos de configuración, disminuyendo el par máximo medio y suavizando la actuación del manipulador, a costa de un incremento del tiempo total.

Además, el radar permite identificar qué articulaciones actúan como “cuello de botella” del esfuerzo: se observa que los valores máximos no se reducen de forma uniforme en todas las juntas, sino que existen articulaciones que permanecen con niveles similares entre soluciones, mientras que otras concentran la mayor parte de la mejora. Esto es coherente con el hecho de que ciertas articulaciones asumen la mayor contribución al posicionamiento/orientación del brazo en el espacio de trabajo.

La Tabla 2 recoge los valores obtenidos para cada una de estas trayectorias en comparación con la inicial, expresados en magnitudes reales.

El análisis de estos resultados permite destacar dos aspectos:

1. La trayectoria con mejor tiempo consigue reducir la duración del movimiento de 37.91 s a 29.16 s, lo que supone una mejora del 23% respecto a la inicial.
2. La trayectoria con menor par medio logra reducirlo de 18.68 Nm (inicial) a 12.80 Nm, con una mejora del 31.5%, favoreciendo trayectorias más suaves y sostenibles para el manipulador.

Tabla 2. Valores comparativos de tiempo de trayectoria y par máximo articular (Escenario 1).

Magnitud	PLC	MinT	MinPM
$t_{trayectoria}$ (s)	37,91	29,16	47,45
T_1^{max} (Nm)	29,30	13,77	5,29
T_2^{max} (Nm)	49,13	43,62	43,57
T_3^{max} (Nm)	22,73	23,39	22,46
T_4^{max} (Nm)	8,41	7,47	3,55
T_5^{max} (Nm)	2,47	2,00	1,88
T_6^{max} (Nm)	0,04	0,12	0,06

La Figura 5 ilustra estas soluciones en el entorno tridimensional de simulación, donde puede observarse la diferencia en la suavidad y el recorrido espacial de cada trayectoria. En la imagen se representa la trayectoria inicial (línea negra discontinua), la primera trayectoria libre de colisiones (roja), la trayectoria con menor tiempo de ejecución (verde) y la trayectoria con menor media de pares máximos (morado). Se aprecia que la trayectoria optimizada en par presenta movimientos más regulares y transiciones suaves entre puntos, mientras que la optimizada en tiempo describe un recorrido más directo, priorizando la velocidad de ejecución.

Trayectoria inicial vs trayectorias optimizadas

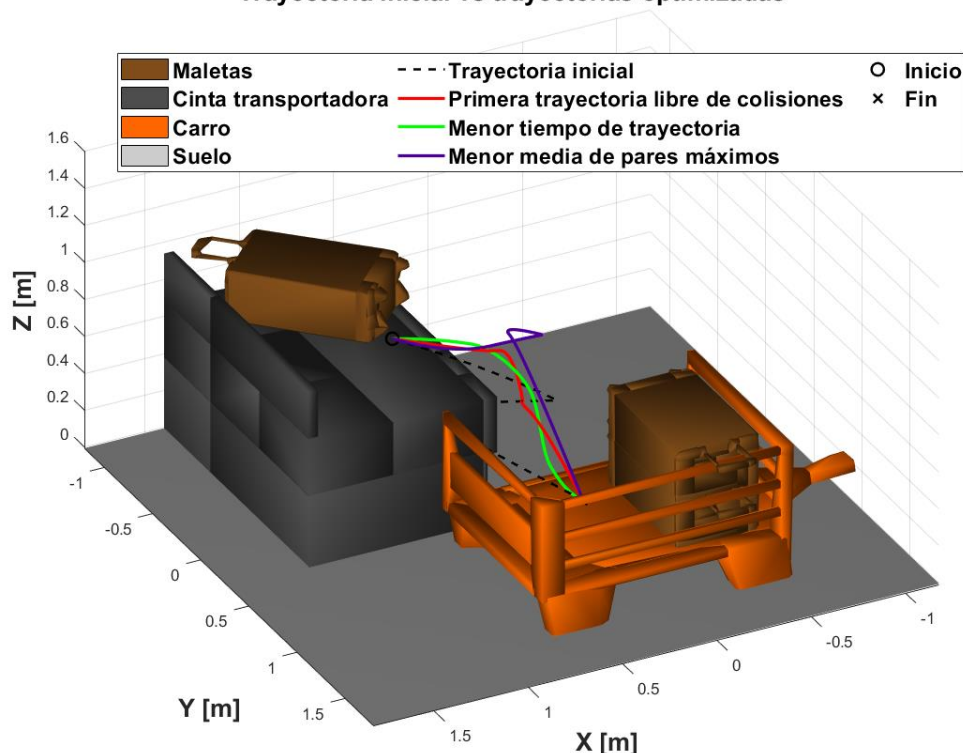


Figura 5. Comparativa entre la trayectoria inicial y las trayectorias optimizadas para la optimización de siete objetivos enfocada en minimizar el tiempo de trayectoria y los pares máximos de cada articulación.

Estos resultados evidencian que, tras aproximadamente 3 horas de ejecución del algoritmo MOPSO, es posible obtener trayectorias que mejoran significativamente tanto el tiempo de

trayectoria como la exigencia dinámica del manipulador, aportando diferentes soluciones en función del criterio prioritario de optimización.

6.2. Escenario 2: Optimización de tiempo de trayectoria y velocidades máximas articulares

La Figura 6 muestra la comparación de trayectorias en el segundo escenario de optimización, en el que se consideran como objetivos el tiempo total de ejecución y las velocidades máximas alcanzadas por cada articulación.

De nuevo se representan tres soluciones significativas:

1. La primera trayectoria libre de colisiones (PLC).
2. La trayectoria con menor tiempo de trayectoria (MinT).
3. La trayectoria con la menor media de velocidades máximas articulares (MinVM).

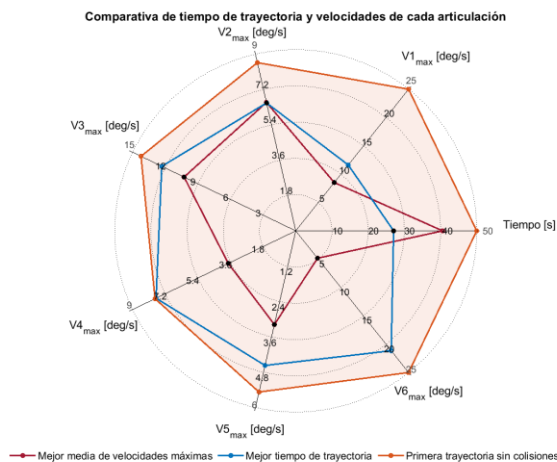


Figura 6. Comparativa de tiempo de trayectoria y velocidades máximas por articulación.

Esta Figura evidencia el compromiso entre productividad y suavidad cinemática. La solución MinT reduce el tiempo total a costa de incrementar los picos de velocidad en varias articulaciones, mientras que MinVM disminuye de forma global las velocidades máximas, suavizando el movimiento y reduciendo cambios bruscos, aunque con un aumento del tiempo de ejecución. En consecuencia, el frente de soluciones

permite seleccionar trayectorias más rápidas (MinT) o más suaves y estables (MinVM) según las prioridades operativas.

La Tabla 3 recoge los resultados obtenidos en unidades reales, comparando estas tres trayectorias con la inicial.

Tabla 3. Valores comparativos de tiempo de trayectoria y velocidad máxima articular (Escenario 2).

Magnitud	PLC	MinT	MinVM
$t_{trayectoria}$ (s)	62,01	27,05	40,58
V_1^{max} (m/s)	25,02	11,64	853
V_2^{max} (m/s)	8,57	6,52	6,55
V_3^{max} (m/s)	14,23	12,32	10,27
V_4^{max} (m/s)	7,77	7,71	3,72
V_5^{max} (m/s)	5,47	4,57	3,18
V_6^{max} (m/s)	25,83	21,15	4,83

El análisis de los resultados pone de relieve varios aspectos:

- I. La trayectoria con menor tiempo lo reduce en un 56% respecto a la trayectoria inicial, de 62.01s a 27.05s.
- II. La trayectoria optimizada en velocidades consigue reducir la media de velocidades máximas de las articulaciones, pasando de 14.90 deg/s (inicial) a 6.84 deg/s, lo que supone una mejora del 54% en términos de suavidad del movimiento.
- III. En ambas soluciones los valores se mantienen claramente por debajo de los límites cinemáticos del GoFa5, garantizando la viabilidad de su ejecución.

En la Figura 7 se muestran las trayectorias tridimensionales correspondientes en el entorno de simulación: la trayectoria inicial (línea negra discontinua), la primera trayectoria libre de colisiones (roja), la trayectoria con menor tiempo de ejecución (verde) y la trayectoria con menor media de pares máximos (morado). Se observa que las trayectorias optimizadas en velocidad tienden a generar curvas más amplias y regulares, evitando picos de aceleración, mientras que las optimizadas en tiempo priorizan trayectorias más directas con cambios de dirección más pronunciados.

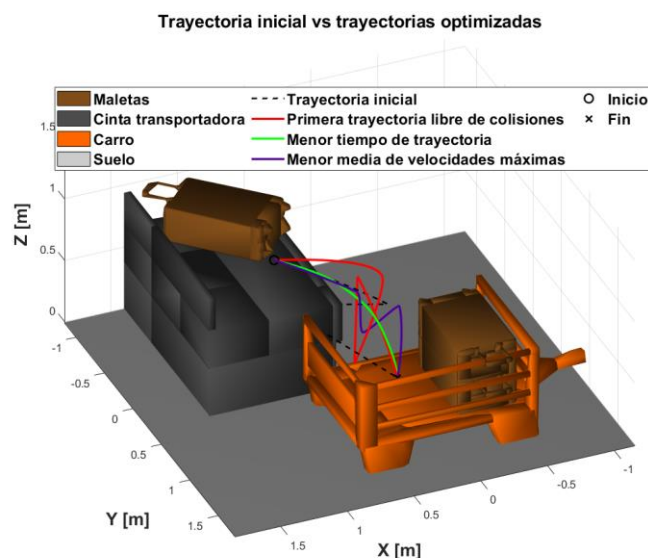


Figura 7. Comparativa entre la trayectoria inicial y las trayectorias optimizadas para la optimización de siete objetivos enfocada en minimizar el tiempo de trayectoria y las velocidades máximas de cada articulación.

En este escenario se observa que el algoritmo MOPSO es capaz de ofrecer soluciones adaptadas a distintos criterios: trayectorias más rápidas, que maximizan la productividad, o trayectorias más suaves, que reducen la exigencia cinemática sobre las articulaciones.

6.3. Escenario 3: Optimización de tiempo de trayectoria, pares y velocidades máximas articulares

El tercer escenario amplía la formulación del problema considerando de manera conjunta el tiempo de trayectoria, los pares y las velocidades máximas alcanzadas en cada

articulación. El objetivo de esta configuración es identificar trayectorias que representen un equilibrio global entre productividad, suavidad y durabilidad, incorporando en un mismo frente de Pareto todos los criterios de interés.

La Figura 8 muestra la comparativa de las distintas trayectorias obtenidas en este escenario:

1. La primera trayectoria libre de colisiones (PLC).
2. La trayectoria con menor tiempo de trayectoria (MinT).
3. La trayectoria con la menor media de pares máximos (MinPM).
4. La trayectoria con la menor media de velocidades máximas articulares (MinVM).

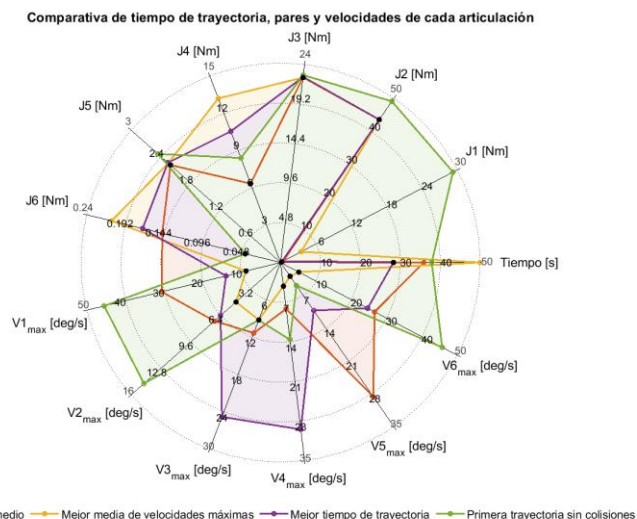


Figura 8. Comparativa de tiempo de trayectoria, pares y velocidades máximas articulares.

La Figura 8 refleja el compromiso multiobjetivo al considerar simultáneamente tiempo, pares y velocidades: no existe una trayectoria dominante en todos los criterios. MinT prioriza la productividad, mientras que MinPM y MinVM reducen, respectivamente, la sollicitación mecánica y la agresividad cinemática, normalmente a costa de mayores tiempos o de

penalizaciones en otras articulaciones. Por tanto, el conjunto de soluciones permite seleccionar el compromiso más adecuado según las prioridades operativas.

En la Figura 9 se muestran las trayectorias tridimensionales correspondientes en el entorno de simulación.

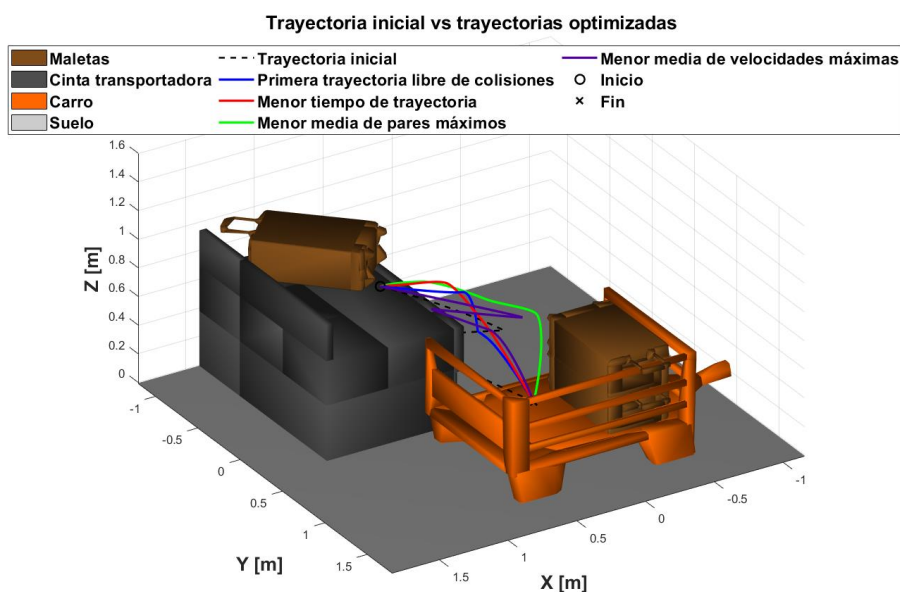


Figura 9. Comparativa entre la trayectoria inicial y las trayectorias optimizadas para la optimización de trece objetivos enfocada en minimizar el tiempo de trayectoria, los pares y velocidades máximas de cada articulación.

La Tabla 4 recoge los resultados comparativos de tiempo de trayectoria, par máximo articular y velocidad máxima articular (Escenario 3).

Magnitud	PLC	MinT	MinPM	MinVM
$t_{trayectoria}$ (s)	37,91	28,30	35,92	54,08
T_1^{max} (Nm)	29,30	0,18	0,46	3,50
T_2^{max} (Nm)	49,13	43,57	43,55	43,53
T_3^{max} (Nm)	22,73	22,51	22,44	22,43
T_4^{max} (Nm)	8,41	10,56	6,35	13,21
T_5^{max} (Nm)	2,47	2,26	2,21	2,26
T_6^{max} (Nm)	0,04	0,17	0,15	0,21
V_1^{max} (m/s)	45,54	14,07	30,67	8,91
V_2^{max} (m/s)	14,65	6,43	7,08	4,77
V_3^{max} (m/s)	9,28	24,84	11,35	9,23
V_4^{max} (m/s)	13,63	29,52	8,19	4,19
V_5^{max} (m/s)	4,92	10,24	28,65	2,95
V_6^{max} (m/s)	45,78	24,66	26,64	5,18

El análisis de los indicadores medios permite destacar que:

La optimización del par articular reduce el valor medio de 18,68 Nm a 12,52 Nm, lo que supone una mejora del 32,94%, incrementando la suavidad del movimiento.

La optimización de velocidades es la más destacada en este escenario, reduciendo la media de 22,30 deg/s a 5,87 deg/s, lo que equivale a una mejora del 73,7% respecto a la trayectoria inicial.

Dado que este escenario integra trece objetivos simultáneos y constituye el caso más exigente del estudio, se evaluó en él la repetibilidad de los resultados repitiendo la optimización en cinco ejecuciones independientes. Se obtuvo una variabilidad reducida en las métricas clave: el tiempo mínimo fue 28.53 ± 0.35 s (CV = 1.21%), el par medio mínimo 12.29 ± 0.34 Nm (CV = 2.76%) y la velocidad media mínima 6.62 ± 0.30 deg/s (CV = 4.573%). Esta estabilidad también se aprecia en la estructura del frente de Pareto, con 12.6 ± 0.55 soluciones no dominadas (CV = 4.35%) y un hipervolumen relativo de 0.127 ± 0.0030 (CV = 2.35%). En conjunto, todas las métricas presentan coeficientes de variación inferiores al 5% (máximo CV = 4.575%). De forma complementaria, se analizó la similitud de las soluciones comparando las trayectorias seleccionadas entre ejecuciones mediante la distancia euclídea punto a punto. En los tres criterios analizados (mejor tiempo, mejor par y mejor velocidad), las trayectorias no se diferenciaron en más de 10 mm en posición y 5° en orientación, lo que confirma una convergencia consistente bajo el presupuesto de cómputo fijado.

7. Discusión

Los tres escenarios evaluados reflejan el comportamiento propio de la optimización multiobjetivo: no existe una única solución óptima, sino un conjunto de soluciones no dominadas que materializan distintos compromisos entre tiempo de ejecución, suavidad (velocidades máximas) y sollicitación mecánica (pares máximos). En consecuencia, la aportación práctica del enfoque no se limita a una trayectoria concreta, sino a proporcionar un frente de soluciones factibles y libres de colisión que permite seleccionar la alternativa más adecuada según el criterio operativo de la aplicación.

En relación con metodologías alternativas, la planificación de trayectorias en manipuladores industriales suele abordarse

mediante parametrizaciones deterministas (como polinomios/splines) y ajustes manuales, planificadores con verificación de colisiones, o técnicas de optimización numérica/metaheurística. Frente a enfoques clásicos de “programación por puntos”, la propuesta integra una optimización sistemática de múltiples métricas físicas bajo restricciones explícitas, evitando depender de ajustes empíricos y ofreciendo múltiples compromisos. Además, frente a formulaciones multiobjetivo por agregación (suma ponderada), MOPSO permite aproximar directamente un conjunto de soluciones de compromiso sin requerir ponderaciones fijas a priori. Esta comparación se plantea de forma cualitativa, ya que el objetivo del trabajo es validar la formulación y analizar el efecto de diferentes conjuntos de objetivos en un caso de estudio común; una comparación cuantitativa con métodos alternativos constituye una extensión natural del estudio.

Por último, conviene explicitar tres limitaciones principales: (i) los resultados se obtienen en simulación y dependen de la fidelidad del modelo dinámico, por lo que es necesaria validación experimental mediante instrumentación del robot; (ii) el tiempo de cómputo reportado responde a un presupuesto experimental y la metodología está concebida para aplicarse de forma offline en la fase de programación/puesta en marcha; y (iii) la trayectoria inicial es representativa pero no proviene de operación real, por lo que una comparación directa con trayectorias usadas en entornos reales reforzaría la contextualización industrial.

Para el despliegue en una solución real en un aeropuerto se realizarían diferentes optimizaciones offline para diferentes maletas en diferentes posiciones y orientaciones. Después un sistema en tiempo real detectaría la maleta y su posición/orientación y seleccionaría la mejor trayectoria optimizada conocida en base a los resultados de las optimizaciones offline, corrigiendo las desviaciones con ayuda de visión artificial.

Asimismo, conviene señalar que el Escenario 3 constituye un caso de optimización “many-objective” (MaOP) al considerar trece objetivos simultáneos. En este régimen, es habitual que aumente el porcentaje de soluciones no dominadas, reduciéndose la presión selectiva del criterio de dominancia y dificultando la convergencia hacia un frente bien estructurado. Como posible mejora, se plantea como trabajo futuro estudiar técnicas de reducción/priorización de objetivos (análisis de correlación entre métricas, selección de un subconjunto de objetivos no redundantes, o formulaciones jerárquicas), así como el empleo de variantes de algoritmos diseñadas específicamente para problemas de optimización “many-objective” (Thomas and Moghadam, 2025).

8. Conclusiones

El presente trabajo ha propuesto una metodología de optimización multiobjetivo para la generación de trayectorias en el robot colaborativo ABB GoFa 5, basada en el algoritmo MOPSO y aplicada a un caso de uso en entorno aeroportuario. A partir de la ejecución del algoritmo en tres configuraciones experimentales, se han podido extraer las siguientes conclusiones principales:

Una de las observaciones más relevantes es que el robot, al enfrentarse al problema de minimización del par articular,

aprende a girar sobre su propio eje sin generar excentricidad significativa. Esta estrategia permite reducir el esfuerzo en la articulación 1, evitando la aparición de picos de par y favoreciendo la sostenibilidad mecánica del movimiento.

En el escenario de optimización de tiempo y par, el algoritmo generó trayectorias que redujeron en torno a un 31% el par medio y un 23% el tiempo de trayectoria, evidenciando que es posible simultáneamente aumentar la eficiencia mecánica y mejorar la productividad.

En el escenario de optimización de tiempo y velocidades máximas se obtuvieron trayectorias con una reducción superior al 50% en la media de velocidades máximas, lo que se traduce en movimientos más suaves y estables, con menor riesgo de generar aceleraciones bruscas o vibraciones en los actuadores.

En el escenario global con 13 objetivos, el algoritmo mostró su capacidad para equilibrar distintos criterios simultáneamente. Se observaron mejoras significativas tanto en el par medio (−33%) como en la velocidad media (−74%), lo que confirma que la metodología permite identificar soluciones integrales que combinan productividad, durabilidad y suavidad en la operación del robot.

La comparación de los tres escenarios pone de relieve que la selección de la función de coste es determinante en el resultado final. Dependiendo de si el objetivo es reducir tiempo, minimizar picos de par o suavizar las velocidades, el algoritmo ofrece trayectorias distintas, todas ellas libres de colisiones y físicamente factibles.

En conjunto, los resultados demuestran que el enfoque propuesto constituye una herramienta prometedora para la planificación avanzada de trayectorias en entornos industriales colaborativos, ya que ofrece soluciones adaptables a distintos criterios operativos y contribuye a prolongar la vida útil de los manipuladores sin comprometer la productividad.

Como líneas futuras, se plantea la validación experimental de las trayectorias optimizadas en un entorno real mediante la instrumentación del robot, integrando el modelo dinámico con sensores de par y datos de corriente articular para contrastar las predicciones simuladas. Asimismo, se prevé extender la metodología a escenarios con manipulación dual y movimientos coordinados entre varios robots, así como explorar la incorporación de métricas de energía y de seguridad colaborativa, y el uso de algoritmos diseñados específicamente para problemas de optimización “many-objective”.

Agradecimientos

Este trabajo ha sido realizado parcialmente gracias al apoyo de la Comisión Europea en el marco del proyecto europeo MANiBOT, número de referencia 101120823.

Referencias

Alonso Zotes, F., & Santos Peñas, M. (2017). Una revisión de técnicas de optimización heurística para el diseño de trayectorias interplanetarias en misiones espaciales. *Revista Iberoamericana de Automática e Informática industrial*, 14(1), 1-15.

Bayona, E., Sierra-García, J. E., & Santos, M. (2024). Comparative Analysis of Metaheuristic Optimization Methods for Trajectory Generation of Automated Guided Vehicles. *Electronics*, 13(4), 728.

Bayona, E., Sierra-García, J. E., & Santos Peñas, M. (2025). Improving Safety and Efficiency of Industrial Vehicles by Bio-Inspired Algorithms. *Expert Systems*, 42(3), e13836.

Bugarin, E., & Aguilar-Bustos, A. Y. (2025). Control de posición utilizando un servomotor inteligente en modo voltaje y considerando fricción: Tutorial. *Revista Iberoamericana de Automática e Informática industrial*, 22(1), 1-12.

Cao, J., Zhang, J., Zhao, F., & Chen, Z. (2021). A two-stage evolutionary strategy based MOEA/D to multi-objective problems. *Expert Systems with Applications*, 185, 115654.

Duan, Z., Zhang, Y., Wang, R., Xu, Z., & Xiang, Z. (2025). Robot path planning method in rough terrain based on multi-objective crossover-mutation particle swarm optimization. *Evolutionary Intelligence*, 18(3), 64.

García, J. M., & Duarte, F. (2025). Posicionamiento del brazo de un robot móvil como apoyo para superar obstáculos utilizando lógica difusa. *Revista iberoamericana de automática e informática industrial (RIAI)*, 22(4), 298-309.

Ghaffour, K. (2024). Multi-objective continuous review inventory policy using MOPSO and TOPSIS methods. *Computers & Operations Research*, 163, 106512.

Lalwani, S., Singhal, S., Kumar, R., & Gupta, N. (2013). A comprehensive survey: Applications of multi-objective particle swarm optimization (MOPSO) algorithm. *Transactions on combinatorics*, 2(1), 39-101.

Liu, S., & Liu, P. (2022). Benchmarking and optimization of robot motion planning with motion planning pipeline. *The International Journal of Advanced Manufacturing Technology*, 118(3), 949-961.

MANiBOT. (2023). MANiBOT project. <https://manibot-project.eu/>

Macías, F. F. T., de Hoyos Vázquez, F. F., Flores, R. A. V., de la Garza, P. G. V., Guerra, M. G. F., & Garza, O. A. (2024). Digitalización de procesos en la industria 4.0. *RICEA Revista Iberoamericana de Contaduría, Economía y Administración*, 13(25), 62-86.

Martín-H. J. A., de Lope J., Santos M. (2009). A method to learn the inverse kinematics of multi-link robots by evolving neuro-controllers, *Neurocomputing*, Issues 13–15, 2806-2814.

Palmieri, J., Di Lillo, P., Lippi, M., Chiaverini, S., & Marino, A. (2024). A control architecture for safe trajectory generation in human-robot collaborative settings. *IEEE Transactions on Automation Science and Engineering*, 22, 365-380.

Peñacoba, M., Bayona, E., Sierra-García, J. E., & Santos, M. (2024). Route Optimization for UVC Disinfection Robot Using Bio-Inspired Metaheuristic Techniques. *Biomimetics*, 9(12), 744.

Peñacoba, M., Sierra-García, J. E., Santos, M., & Mariolis, I. (2023). Path optimization using metaheuristic techniques for a surveillance robot. *Applied Sciences*, 13(20), 11182.

Peñacoba, M., Sierra-García, J. E., Santos, M. (2026). Human-intelligent trajectory optimization for robotic manipulators with hybrid PSO-PS algorithm. *Advanced Engineering Informatics*, 69, 103941.

Olayo Andrés, D. (2022). Diseño de un planificador RRT* kinodinámico mediante reconexión por splines para robots no-holonómicos con alta inercia (Doctoral dissertation, Industriales).

Prado, R., Peralta, M., Vidal, P., Galvez, C., Herrera, J., & Juárez, L. (2025). Desarrollo de un robot de bajo costo para aplicaciones agrícolas. *Revista Iberoamericana de Automática e Informática industrial*, 22(3), 196-207.

Rascón, R., Flores-Mendoza, A., Moreno-Valenzuela, J., & Aguilar-Avelar, C. (2024). Control para seguimiento de trayectorias cartesianas en robots manipuladores. *Revista Iberoamericana de Automática e Informática Industrial*, 21(3), 252-261.

Thomas, H., & Moghaddam, M. J. (2025). A review of the evolution of multi-objective evolutionary algorithms. *Computers, Materials, & Continua*, 85(3), 4203.

Verma, S., Pant, M., & Snel, V. (2021). A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems. *IEEE access*, 9, 57757-57791.

Yang, N. C., Su, C. H., & Yang, H. (2025). Optimal parameter tuning for voltage control thresholds in distribution networks using MOPSO and Pareto fronts. *Electric Power Systems Research*, 244, 111552.

Yang, Y., Han, S., Sang, H., & Liu, F. (2022). Preoperative planning method based on a MOPSO algorithm for robot-assisted cholecystectomy. *International Journal of Computer Assisted Radiology and Surgery*, 17(4), 731-744.