Annals of Emerging Technologies in Computing (AETiC)
Vol. 8, No. 4, 2024

Research Article

Whale-Based Trajectory Optimization Algorithm for 6 DOF Robotic Arm

Mahmoud A. A. Mousa[1,2,*], Abdelrahman T. Elgohr[1,3] and Hatem A. Khater[3]

[1]Zagazig University, Egypt.
[2]Heriot Watt University, Dubai, UAE
mamosa@zu.edu.eg; m.mousa@hw.ac.uk
[3]Horus University – Egypt, Egypt
atarek@horus.edu.eg; hatem.a.khater@gmail.com
*Correspondence: mamosa@zu.edu.eg

Received: 23th April 2024; Accepted: 29th September 2024; Published: 1st October 2024

Abstract: Trajectory optimal control for a robotic arm with a high degree of freedom (DOF) is challenging. The design space for that problem is complex and the search for an optimal solution is demanding. The design of a robotic arm's trajectory is based on solving the inverse kinematics problem, considering additional refinements influenced by factors like total rotating angle, reachability time, minimum execution time, obstacle avoidance, and energy consumption minimization. Due to the complexity of the design space, in this paper, genetic algorithm (GA) optimization and whale optimization algorithm (WOA) have been used to achieve robotic arm trajectory control while maintaining a minimum reachability time. To validate the suggested techniques, a case study was conducted on a 6 DOF KUKA KR 4 R600 robot arm to control subject to its constraints. Sets of consecutive points forming four different paths were inputted to the algorithms. The goal was to reach all these points, in order, with a minimum total reachability time. As a result of this paper, we shown that the whale optimization algorithm provides better performance than the genetic algorithm with a factor of more than 2.5 while satisfying the reachability constraints.

Keywords: ***Genetic Algorithm; Kinematics Analysis; Trajectory Optimization; Whale Optimization Algorithm; 6 DOF Robotic Arm***

---

1. Introduction

Robotic arms are commonly used specially in cutting-edge robotics. The use of robot fingers is increasing in various industries as they can be used to automate jobs which are risky, repetitive, or require excessive precision. Robotic hands can be used to carry out a wide range of tasks, along with welding, painting, and assembly, increasing performance and lowering labor costs. Robotic arms are used in clinical surgeries to increase precision and accuracy and reduce human errors [1]. They are used to assist people with disabilities, permitting them to perform tasks that had been formerly not possible. The robotic arm may be attached to a wheelchair to help the operator in gaining access to objects on high shelves or performing tasks that require arm mobility. Robotic arms can also be used in agriculture to automate the process of planting and harvesting crops, thereby lowering the need for manual labor. Robotic arms are deployed for tasks along with repairing satellites or building extraterrestrial habitats [2].

Trajectory optimization is the cornerstone for the overall effectiveness and efficiency of robotic arm systems [2]. It contributes to the overall effectiveness and efficiency of robotic arm systems by ensuring that a robotic arm will move along predetermined paths quickly and efficiently. By carefully designing the form of a robotic arm [3], engineers can optimize various parameters such as arrival time, energy consumption, and adherence to the physical limitations of the robot's joints [4]. Through trajectory optimization, robotic arms can perform tasks with accuracy, agility, and efficiency. This makes trajectory optimization an important

Mahmoud A. A. Mousa, Abdelrahman T. Elgohr and Hatem A. Khater, "Whale-Based Trajectory Optimization Algorithm for 6 DOF Robotic Arm", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 99-114, Vol. 8, No. 4, 1st October 2024, Published by International Association for Educators and Researchers (IAER), DOI: 10.33166/AETiC.2024.04.005, Available: http://aetic.theiaer.org/archive/v8/v8n4/p5.html.

aspect of robotic management and automation systems. In this regard, this research explores the problem area presented by trajectory optimization for robotic gloves, and proposes strategies for improving robotic technology, discussing the challenges and implications for advancing the field [5], [6].

In this study, the proposed Whale-based trajectory Optimization Algorithm (WOA) is compared to the Genetic-based trajectory optimization Algorithm (GA). WOA demonstrates strong potential for addressing complex optimization challenges in nonlinear and multimodal problems. GA has a wide recognition and frequent application in optimization tasks, particularly in robotic trajectory planning. As a reliable, traditional baseline, GA's performance in trajectory optimization is extremely high which makes it a perfect baseline method in comparison with the proposed algorithm [6].

1.1. Related Work

Offline robot paths save work cycle time in automated automotive production lines. This allows for enhanced online robot controller set enhanced points [7]. Heim and Von Stryk (2000) provide methods for directly transcribing restricted trajectory optimization using complete dynamic robot models. Optimization methodologies are effectively compatible with CAR tools and current robot controllers. Simulation and experimental results illustrate this using an ABB IRB 6400 industrial robot [7].

Trajectory optimization for robotic grasping with occlusions seems promising. Kahn et al. (2015) use trajectory optimization to actively investigate the environment and find the robot's optimal grabbing places, even with obstacles. By considering the robot's kinematics, dynamics, and sensory system uncertainties, trajectory optimization algorithms can create optimal pathways for obstacle adaptation and object manipulation. This method works when things are partially blocked or the robot has little prior knowledge. Exploring the environment helps the robot learn about items and gripping places, improving grabbing success. Optimizing trajectory algorithms in active exploration can improve robotic manipulation systems' autonomy and efficiency while grabbing items with occlusions [8].

Mei *et al.* [9] proposed a novel 6-degrees-of-freedom high-speed parallel robot to address the limitations of existing parallel robots in meeting the operational demands of non-planar industrial production lines. Kinematic and dynamic analyses are conducted to evaluate the performance of the proposed robot. A trajectory optimization method is introduced to improve the smoothness of robot end-effector motion, targeting the average cumulative effect of joint jerk. To address deformation issues in the horizontal motion stage of the trajectory, a mapping model is established, along with a trajectory deformation evaluation index constructed to optimize trajectory smoothness and minimize deformation. Comparative analysis demonstrates significant reductions in maximum robot joint jerk and torque, highlighting the efficacy of the proposed trajectory optimization approach in improving robot performance [9].

In 2020, Benotsmane and colleagues [10] developed a "whip-lashing" technique for robotic arm movement. This method optimizes the robot arm's trajectory to increase part velocity, decrease motion cycle times, and maximize joint torque. Researchers used SolidWorks and MATLAB to examine trajectory planning for a five-degree-of-freedom RV-2AJ manipulator arm and validated the use of whip-lashing for trajectory optimization. Two trajectories were built for the robot: the original path and one to test motion produced by whip-lashing. The method decreased the cycle time of the RV-2AJ robot arm by 33%. The study's main achievement was the creation and implementation of the "whip-lashing" technique, which decreased torque usage and the time taken for manipulator arm movements, thereby improving productivity [10].

Choubey *et al*. [11] developed an optimal trajectory generation (OTG) technique for smooth, accurate, and rapidly converging continuous path motion. Their OTG technique utilized the Grey Wolf Optimization (GWO) method to find the trajectory path that minimizes tracking error while considering combined speed, joint jerk (avoiding abrupt changes in speed), and smooth motion along the path without errors [11].

Wang *et al.* [12] proposed a time-domain model for the energy consumption of a fluidic soft robotic arm. Using forward kinematic analysis, the researchers optimized the trajectory of a soft robotic arm to save energy while considering motion limitations using an interior point technique. Experiments were conducted to assess the proposed model and refined trajectory. The time-based model described the dynamic energy characteristics of fluidic soft actuators during different movements. Trajectory optimization minimized energy consumption in soft robotic arms. This study provided evidence for the optimization of energy-based soft robots [12].

Mousa *et al*. [13] conducted a study on trajectory optimization for a robotic arm, emphasizing the reduction of reachability time. Conventional techniques have demonstrated restricted progress, leading to the investigation of novel ways. This research presents two techniques: rule-based optimization and a genetic algorithm, utilizing the robot's kinematics to save operating time. The study, employing the KUKA KR 4 R600 robot, showed that the genetic algorithm is better at obtaining the shortest trip time. Genetic algorithm solutions are around three times quicker than rule-based solutions for identical pathways. This demonstrates how advanced optimization approaches improve the control of robotic arm trajectories [13]. Table 1. Conclude the literature review about robotic arm trajectory optimization.

Table 1. Literature summery

Author, year	Study focus	Optimization Technique	Robotic System
Heim and Von Stryk [7]	Trajectory optimization using complete dynamic robot models, tested on ABB IRB 6400.	Dynamic models compatible with CAR tools and robot controllers.	ABB IRB 6400 industrial robot.
Kahn *et al.* [8]	Trajectory optimization for robotic grasping with occlusions and environment exploration.	Considering kinematics, dynamics, and sensor uncertainties.	Robotic systems with grasping and occlusions.
Mei *et al.* [9]	Optimization for a 6-DOF high-speed parallel robot focusing on joint jerk and deformation.	Mapping model and deformation evaluation to improve trajectory smoothness.	6-DOF high-speed parallel robot for non-planar industrial lines.
Benotsmane *et al.* [10]	Whip-lashing technique for reducing cycle times and improving robot arm productivity.	Whip-lashing method to optimize joint torque and cycle times.	RV-2AJ manipulator arm (5-DOF).
Choubey *et al.* [11]	Optimal Trajectory Generation (OTG) using Grey Wolf Optimization for smooth continuous motion.	Grey Wolf Optimization minimizing tracking errors and joint jerk.	Continuous path motion for general robotic arms.
Wang *et al.* [12]	Time-domain model for optimizing soft robotic arm energy consumption.	Interior point technique to minimize energy consumption.	Fluidic soft robotic arm.
Mousa *et al.* [13]	Genetic Algorithm for reducing robotic arm reachability time, tested on KUKA KR 4 R600.	Genetic Algorithm compared to rule-based optimization for shortest trip time.	KUKA KR 4 R600 robotic arm.

This research paper investigates the use of Whale Optimization Algorithm for optimizing robotic arm trajectories, prioritizing minimal total reachability time. As a reference, a genetic-based trajectory optimization approach has been deployed as well to compare its findings to the results from the proposed whale-based trajectory optimization algorithm. The main aim is to leverage inverse kinematics to minimize operating time during the entire cycle. The effectiveness of each method is evaluated by simulating the computational complexity of the robotic arm's movements. The techniques mentioned before were validated and compared using a six-degree-of-freedom (DOF) KUKA KR 4 R600 robot for trajectory optimization. The findings of the application show that optimizing the robot's operating time based on Whale-based algorithm is faster than the Genetic-based approach.

2. Kinematic Analysis and Trajectory Optimization

2.1. Getting Equations for Reachability Time Problem

Kinematic analysis involves two main parts: forward kinematics and inverse kinematics. The detailed analysis for a 6 DOF robotic arm is shown in our previous research paper in Mousa et al. [13]. This involves deducing the Denavit-Hartenberg (DH) parameters and substituting them into the homogeneous matrix to describe the end-effector's kinematics and also the decoupling approach steps for the inverse kinematics. These steps allow us to describe the position of each joint as a function of the desired final position for arm body joints, and orientation for spherical wrist joints.

To commence, an equation was deduced to calculate the Reachability time ($R.T_i$) of each joint to rotate one unit with maximum velocity (ω_n) as shown in Equation 1. The total reachability time from point to another ($R.T_{P2P}$) as shown in Equation 2 by finding the sum of the angles of rotation (q_i) of each joint to get from one point (a) to another point (b) multiplied by single rotating unit operating time for each joint $R.T_i$. The total operating time over a complete path ($R.T_{Path}$) is shown in Equation 3, all of which is dependent on the number of arm joints (n) and the total number of points on a path (k) [14].

$$R.T_i = \frac{1\,rad}{\omega_n} \tag{1}$$

$$R.T_{P2P} = \sum_{i=1}^{n}(|(q_{bi} - q_{ai})|) * R.T_i \tag{2}$$

$$R.T_{Path} = \sum_{j=1}^{k}\sum_{i=1}^{n}(|(q_{bij} - q_{aij})|) * R.T_i \tag{3}$$

2.2. Trajectory Optimization

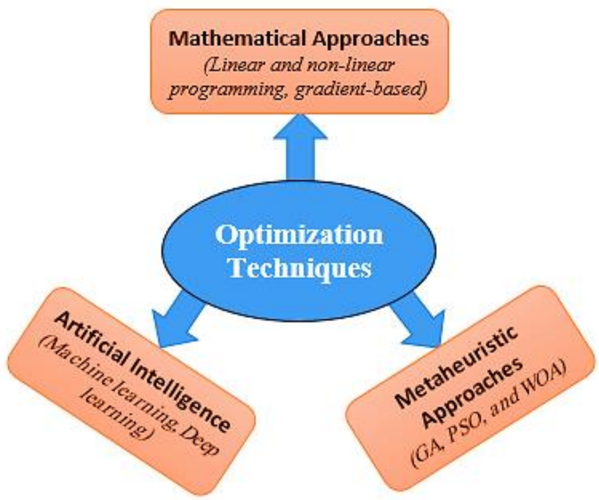


Figure 1. Optimization techniques [15].

Optimization techniques, as mentioned in Figure 1., are indispensable for enhancing the performance and efficiency of robotic systems. By categorizing these techniques into mathematical, artificial intelligence (AI), and metaheuristic approaches, we gain access to a diverse toolkit for addressing a wide range of challenges in robotics. From gradient-based optimization methods to reinforcement learning and nature-inspired algorithms like genetic algorithms (GA), particle swarm optimization (PSO), and whale optimization algorithm (WOA) each category offers unique capabilities for optimizing objectives, solving complex problems, and enabling autonomous decision-making in robotic systems. By harnessing the power of these optimization techniques, researchers and engineers can design more efficient, adaptive, and intelligent robotic systems capable of navigating diverse environments, performing complex tasks, and ultimately advancing the field of robotics towards new frontiers of exploration and innovation.[15].

2.3. Genetic Algorithm

Genetic algorithms (GAs) represent a powerful class of optimization algorithms inspired by the process of natural selection and genetics as shown in Figure 2. Developed to tackle complex optimization problems, genetic algorithms have found widespread applicability across various domains, including engineering, finance, and artificial intelligence. This academic discourse endeavors to elucidate the main conceptual framework, general procedural steps, common applications, and specific utilization within the realm of robotics, with a focus on optimizing the trajectory of robot arms [13], [16].

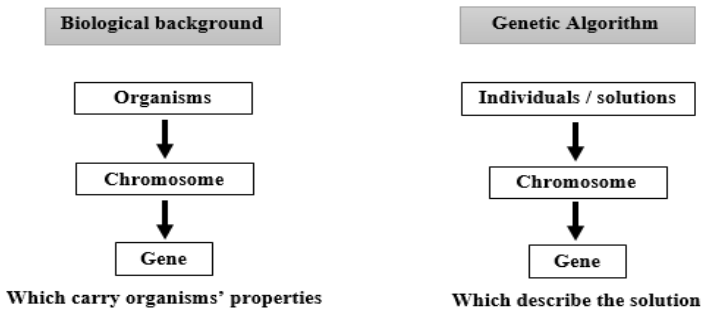


Figure 2. Genetic algorithm vs biological system [13].

The main concept underlying genetic algorithms is rooted in the principles of evolution. Drawing inspiration from the Darwinian theory of natural selection, GAs simulate the process of evolution through iterative improvement. Solutions to optimization problems are treated as individual "chromosomes," represented as strings of parameters. These chromosomes undergo evolution through a series of genetic operators, such as crossover, mutation, and selection, emulating the biological processes of reproduction, genetic recombination, and survival of the fittest [17].

In its procedural essence, a genetic algorithm unfolds through a sequence of steps that collectively strive to iteratively enhance potential solutions. The algorithm initiates with the creation of an initial population of candidate solutions. Through successive generations, individuals with higher fitness—quantifying their suitability to the optimization objective—are more likely to contribute to the subsequent population. The genetic operators then intervene to produce offspring, introducing diversity and facilitating exploration of the solution space. This cyclic process of selection, recombination, and mutation continues until a satisfactory solution is achieved or a predetermined convergence criterion is met [3], [18].

The general applicability of genetic algorithms spans a multitude of optimization challenges, owing to their ability to navigate complex, nonlinear solution spaces. Common usage scenarios encompass function optimization, parameter tuning, scheduling, and pattern recognition. In engineering disciplines, genetic algorithms have proven valuable for optimizing intricate systems where traditional methods may struggle [19].

Despite the strengths of genetic algorithms, there also are drawbacks to be considered of their application. One of those drawbacks is the computational complexity associated with GAs, mainly for massive problems. The iterative nature of the set of rules and the evaluation of a couple of candidate answers make contributions to the excessive computational price. Parameter sensitivity is another problem for GAs. The algorithm includes tuning numerous parameters which include population length, crossover, and mutation rate. Choosing suitable parameter values is a challenging task and can require tremendous experimentation. Premature convergence is a risk of GAs, where the population converges to a suboptimal answer before the solution area is thoroughly explored. Balancing research and its implementation are critical challenges in designing effective GAs [20].

The concept for solving the robot's trajectory optimization problem using GA is to specify an objective function constrained by some parameters' constraints. Set of rules, which are implemented in the GA approach are used to minimize the objective function. The technique begins with initial populations and their fitness are evaluated which determines how "good" a solution is for our problem while the higher fitness score is better. Then, a set of iterative processes are used with the help of the selection, crossover, and mutation concepts to new possible better solutions with lower total trip time for a certain path. The GA keeps attempting to generate possible new solutions until convergence, in which the optimization goal is stable, and it meets the predefined criteria [17].

Upon convergence, the trajectory with the minimum reachability time is extracted as the optimized solution. This trajectory efficiently navigates the 6 DOF robot arm through the designated five points, minimizing reachability time. The algorithm strikes a balance between speed and accuracy, making it well-suited for time-sensitive applications where efficient trajectory planning is critical [13], [16].

The flexibility of GAs allows them to adapt and evolve solutions without explicit rule formulations, making them particularly well suited for problems with intricate and changing conditions. This inherent adaptability positions Genetic Algorithms as a superior choice when faced with optimization challenges that demand flexibility, scalability, and the ability to navigate dynamic environments [13], [21]. On the other hand, genetic algorithm, as we show in the following sections, may provide local optimum solutions for the optimization problems, which in our case study provides longer reachability time than the minimum.

2.4. Whale Optimization Algorithm

Whale Optimization Algorithm (WOA) represents a nature-inspired optimization algorithm rooted in the social behavior of humpback whales as display in Figure 3. Developed as a population-based metaheuristic algorithm, WOA draws inspiration from the collaborative hunting behavior of whales to iteratively refine potential solutions to optimization problems. The main conceptual framework of WOA centers on the emulation of the social hierarchy and coordinated movement exhibited by whale pods during hunting [3].

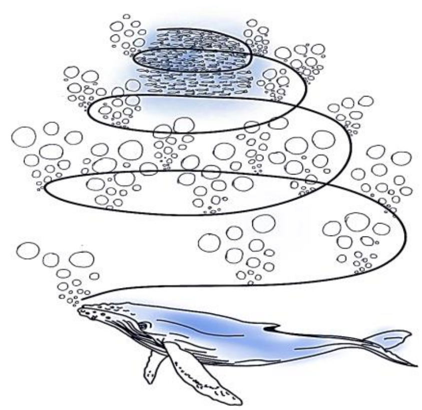


Figure 3. The behavior of humpback whales [3].

In its operational essence, drawing from the social hierarchy and coordinated motion of whale pods, WOA iteratively refines capacity answers to optimization issues [22], [23]. The algorithm commences with the initialization of a population of potential solutions, symbolizing the positions of individual whales in the search space. As the algorithm progresses, individuals within the population engage in exploration and exploitation phases, mirroring the collective movement patterns of whales in nature. These phases are orchestrated by a set of mathematical equations that guide the movement of individuals towards promising regions of the solution space [24].

The general steps of WOA seamlessly blend exploration and exploitation strategies. Whales, representing potential solutions, adjust their positions iteratively based on the best-known solutions within the population. The algorithm incorporates a dynamic balance between exploration, achieved through the repositioning of individuals, and exploitation, facilitated by the convergence toward promising regions identified during the search process [25], [26].

3. Case Study

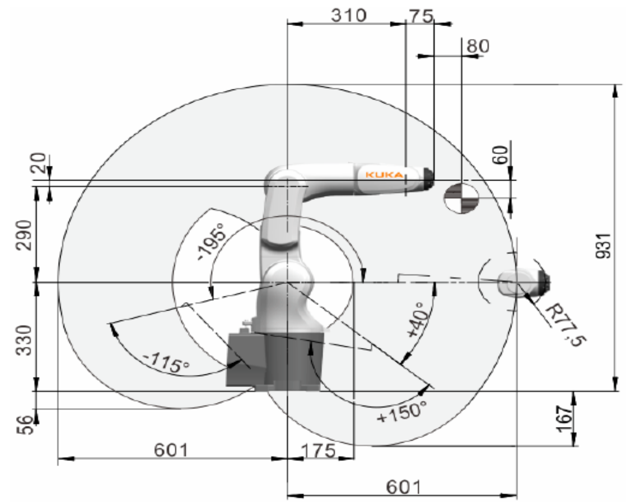


Figure 4. Robotic arm configuration [26].

KUKA model KR 4 R600 robotic arm, with dimensions as shown in Figure 4, was used in our case study. Forward Kinematics were applied together with the inferred DH parameters to configure the robot's joint in the XYZ axes. The KR 4 R600 is a 6-axis (6 Degree of Freedom) industrial robot from KUKA Robotics with a payload of 3 kg. It was built primarily for laboratory businesses that utilize flexible robot-based automation. The robot has an open structure that lends itself to various applications, and it can communicate extensively with other systems [13]. Table 2. conclude the specifications of studied robotic arm.

Table 2. KUKA KR 4 R600 robotic arm specifications

Joints Motors specification	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Rotation range (rad)	2.96 to - 2.96	0.69 to - 3.4	2.62 to - 2	3.23 to - 3.23	2.09 to - 2.09	6.1 to - 6.1
Rotation Speed (rad/sec)	4.364	4.364	4.364	5.586	5.586	7.331

Table 3 displays four different paths to study the quality of the algorithms under study, where each path consists of five points representing a different task required to be executed by the robotic arm. The tracks are designed to test every detail of the robotic arm's workspace. These paths are used to test the genetic algorithm and the whale optimization algorithm in parallel to arrive at a conclusion about the advantage of achieving a minimal reachability time between the two algorithms.

Table 3. Path cases points

Point	Coordinate (x, y, z) (m)			
	Case 1	Case 2	Case 3	Case 4
P0	(0.31, 0, -0.055)	(0.20, 0.40, 0.10)	(0.10, 0.30, 0.20)	(0.23, -0.14, 0.10)
P1	(0.48, 0.19, 0.21)	(0.32, -0.15, 0.05)	(0.17, 0.21, -0.18)	(0.16, 0.19, -0.26)
P2	(0.17, 0.50, 0.20)	(-0.13, 0.24, 0.13)	(-0.20, -0.13, 0.30)	(-0.36, 0.21, 0.14)
P3	(0.10, 0.34, 0)	(-0.09, 0.30, -0.21)	(0.31, 0.22, 0.15)	(-0.15, 0.18, -0.20)
P4	(-0.33, 0.21, 0)	(0.17, -0.19, 0.20)	(0.40, -0.25, 0.10)	(0.27, -0.29, 0.11)

MATLAB software was used as a platform for modeling the robotic arm and implementing the proposed and the baseline algorithms. RoboDK software was used for verifying the reachability outcomes from the two mentioned algorithms by providing simulations for KUKA robotic arm using the algorithms' outcomes along the specified paths.

3.1. Optimization Approaches, Simulation and Results

The optimization algorithms proposed in the following sub-sections solve the reachability problem for a defined path while maintaining the minimum operating time. MATLAB Ver. 2020a, was deployed on a laptop with an Intel Core i7 CPU, Intel(R) HD Graphics 400 GPU, and 16 gigabytes of RAM, to test the suggested optimization approaches. Here is the general approach we used for solving the specified optimization problem using genetic algorithm and whale optimization algorithm.

Algorithm 1. GA/WOA Optimizer

Input: Robot's Parameters for six joints (a_k, α_k, d_k, and θ_k) where $k \in \{1, \dots, 6\}$
and desired locations $\{(P_{6x,1}, P_{6y,1}, P_{6z,1}), \dots, (P_{6x,w}, P_{6y,w}, P_{6z,w})\}$
where $\mathbf{w}$ is number of points in a specific path.

Output: The optimal trajectory RT^*.

1. **Calculate** the forward kinematic homogenous matrix.

$$H_6^0 = \begin{bmatrix} R_6^0 & P_6^0 \\ 0 & 1 \end{bmatrix} = \Pi_{k\in\{1,\dots,6\}} A_i^{i-1}$$

Where

$$A_i^{i-1} = \begin{bmatrix} \mathrm{Cos}\,[\theta i] & -\mathrm{Sin}\,[\theta i] * \mathrm{Cos}\,[\alpha i] & \mathrm{Sin}(\theta i) * \mathrm{Sin}\,[\alpha i] & ai * \mathrm{Cos}\,[\theta i] \\ \mathrm{Sin}\,[\theta i] & \mathrm{Cos}\,[\theta i] * \mathrm{Cos}\,[\alpha i] & -\mathrm{Cos}\,[\theta i] * \mathrm{Sin}\,[\alpha i] & ai * \mathrm{Sin}[\theta i] \\ 0 & \mathrm{Sin}\,[\alpha i] & \mathrm{Cos}\,[\alpha i] & di \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. **Calculate** $\Pi_{k\in\{1,\dots,6\}} A_i^{i-1} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} P_6^0 \\ 1 \end{bmatrix}$ where P_6^0 has a dimension of 3×1
3. **Find** $P_6^0 = \begin{bmatrix} P_{6,x}^0 & P_{6,y}^0 & P_{6,z}^0 \end{bmatrix}^T$ where $P_{6,x}^0, P_{6,y}^0, P_{6,z}^0 = \mathcal{F}(\theta_k)$ and $k \in \{1, \dots, 6\}$
4. **Run** GA/WOA algorithm for the following and find $\mathbf{ot}^*$ given that:

$$\textit{Fitness Value } (FV) = \sum_{j=1}^{w} \sqrt{\left(P_{6,x}^0 - P_{6x,j}\right)^2 + \left(P_{6,y}^0 - P_{6y,j}\right)^2 + \left(P_{6,z}^0 - P_{6z,j}\right)^2}$$

$$\textit{Objevtive Function } = FV + RT^*$$

Where

$$RT^* = \sum\nolimits_{z=1}^{w} RT_z$$

And

$$RT_z = \min_{|P_{j,ValidAngles}|} \left(\sum_{i=1}^{k} \left(\left| (\theta_{a_z,i} - \theta_{a_{z-1},i}) \right| \right) * \frac{1\,rad}{\omega_n};\ z \in \{1, \dots w\} \right.$$

Constrained by

$$-2.96 \le \theta_1 \le 2.96$$
$$-3.4 \le \theta_2 \le 0.69$$
$$-2 \le \theta_3 \le 2.62$$
$$-3.23 \le \theta_4 \le 3.23$$
$$-2.09 \le \theta_5 \le 2.09$$
$$-6.1 \le \theta_6 \le 6.1$$

3.1.1. Genetic Algorithm Technique

For solving the trajectory optimization problem for a 6 DOF robotic arm using genetic algorithms, we encoded the trajectory as a sequence of joint angles. The objective function for our problem was introduced in Algorithm 1 to optimize the total trip time subject to the constraints set mentioned in Table 1. Algorithm 1 illustrated how GA is used to perform the required trajectory optimization. We started with generating a random initial population that contains a number of 30 genes as described in Equation 4. Steps 1 is used to calculate the forward kinematics for the 6 DOF KUKA robotic arm which is applied in steps 2 and 3 to calculate the position equation as a function of the different joint angles which is used in the objective function. Step 4 shows the GA objective function and its constraints.

$$Q = \{q_{kn}\} \text{ where } k \in \mathbb{Z}, n \in \mathbb{Z}, 1 \le k \le 6 \text{ and } 0 \le n \le 4 \tag{4}$$

As outcome to running Algorithm 1 on the four different paths shown in Table 3, the following results were obtained and recorded in Table 4, which describes the optimized values for the different joints to reach the points along every path. The graphical representation for Table 6 is shown in Figure 5(a). Table 5 presents the minimal time to reach every point in each path and its graphical representation is shown in Figure 5(b).

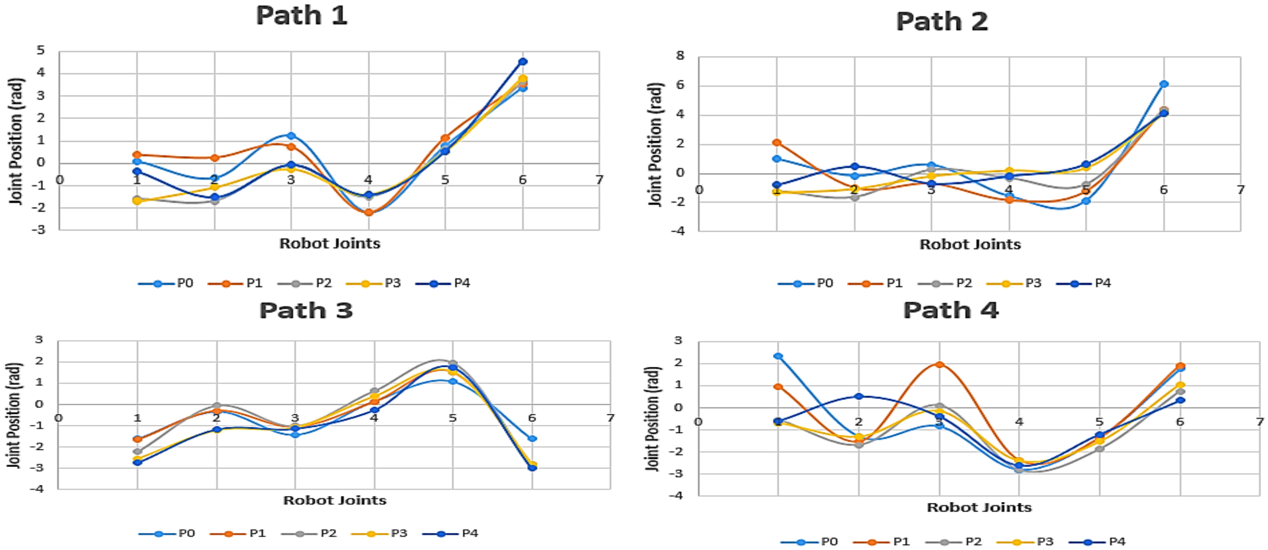
Table 4. GA angles according to grantee the minimum operating time

Path 1 steps	**x (rad)**	**y (rad)**	**z (rad)**	**a (rad)**	**b (rad)**	**c (rad)**
P0	0.0876	-0.6624	1.2177	-2.1979	0.7800	3.3754
P1	0.3695	0.2504	0.7349	-2.1943	1.1482	3.5605
P2	-1.5826	-1.6747	-0.0961	-1.4998	0.6231	3.6259
P3	-1.7021	-1.0709	-0.2520	-1.3893	0.5292	3.8212
P4	-0.3696	-1.5147	-0.0671	-1.4202	0.5338	4.5462
Path 2 steps	**x (rad)**	**y (rad)**	**z (rad)**	**a (rad)**	**b (rad)**	**c (rad)**
P0	0.9923	-0.1615	0.5833	-1.5507	-1.8552	6.0953
P1	2.1036	-0.9921	-0.7044	-1.8431	-1.2178	4.3274
P2	-1.2228	-1.6369	0.2430	-0.3263	-0.7474	4.3688
P3	-1.3096	-1.0447	-0.1621	0.2196	0.4014	4.1153
P4	-0.8102	0.4562	-0.7252	-0.2068	0.6279	4.0942
Path 3 steps	**x (rad)**	**y (rad)**	**z (rad)**	**a (rad)**	**b (rad)**	**c (rad)**
P0	-1.6075	-0.3321	-1.4191	0.1469	1.0653	-1.6157
P1	-1.6514	-0.2967	-1.0595	0.1219	1.5184	-2.7975
P2	-2.1963	-0.0537	-1.0088	0.6320	1.9462	-2.8601
P3	-2.5553	-1.2233	-1.1020	0.3755	1.5216	-2.8127
P4	-2.7266	-1.1687	-1.1433	-0.2727	1.7210	-2.9969
Path 4 steps	**x (rad)**	**y (rad)**	**z (rad)**	**a (rad)**	**b (rad)**	**c (rad)**
P0	2.3384	-1.2875	-0.8275	-2.7927	-1.3378	1.7597
P1	0.9479	-1.5316	1.9395	-2.3807	-1.3630	1.9059
P2	-0.5444	-1.6745	0.1058	-2.8072	-1.8635	0.7377
P3	-0.6834	-1.3032	-0.1542	-2.3836	-1.5265	1.0409
P4	-0.5854	0.5228	-0.3777	-2.6106	-1.1946	0.3353

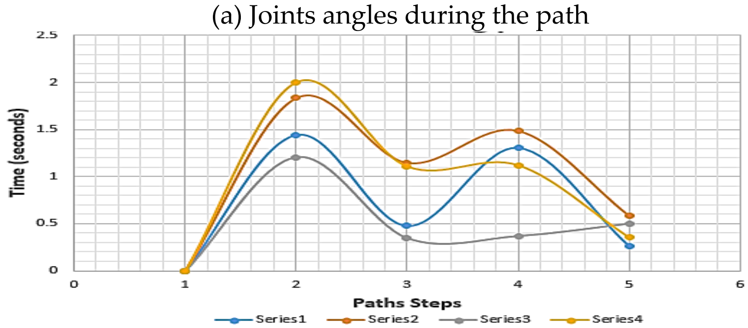
Table 5. Operating time for each path step based on GA

Operating time	**initial to P0**	**P0 to P1**	**P1 to P2**	**P2 to P3**	**P3 to P4**	**Total**
Path 1 (seconds)	1.44202	0.475873	1.305381	0.264485	0.554069	**4.04251**
Path 2 (seconds)	1.836413	1.146447	1.487699	0.586086	0.706757	**5.763402**
Path 3 (seconds)	1.205861	0.346813	0.368437	0.499756	0.23796	**2.658827**
Path 4 (seconds)	1.998507	1.106108	1.119186	0.353781	0.687782	**5.265366**

To verify the reachability solution obtained from the proposed GA approach in Algorithm 1, a simulation was done using RoboDK software. RoboDK software is a powerful which provides a comprehensive solution for robot programming, simulation, and cell design, and is compatible with a wide range of robots and other equipment. We used it to simulate the movement of the KUKA robotic arm based on the findings of the suggested technique. As a depiction of a function performed by the arm in its working environment, the movement was represented as a closed route beginning at the starting position and finishing at the same location. Figure 6 shows a snapshot from the RoboDK software during trajectory simulation. From the simulation, it is possible to know the shape of the movement path for the end effector when observing the connecting lines between the main points of the path. During the arm movement, the program records the six joints position at each moment and saves them in the definition of each joint. It is also possible, through the simulation program, to determine the movement restrictions on each joint, where the speed is determined, and through it, the time of executing the path is known. This ensures that the results generated from the suggested algorithm satisfy the reachability constraints for the different paths.

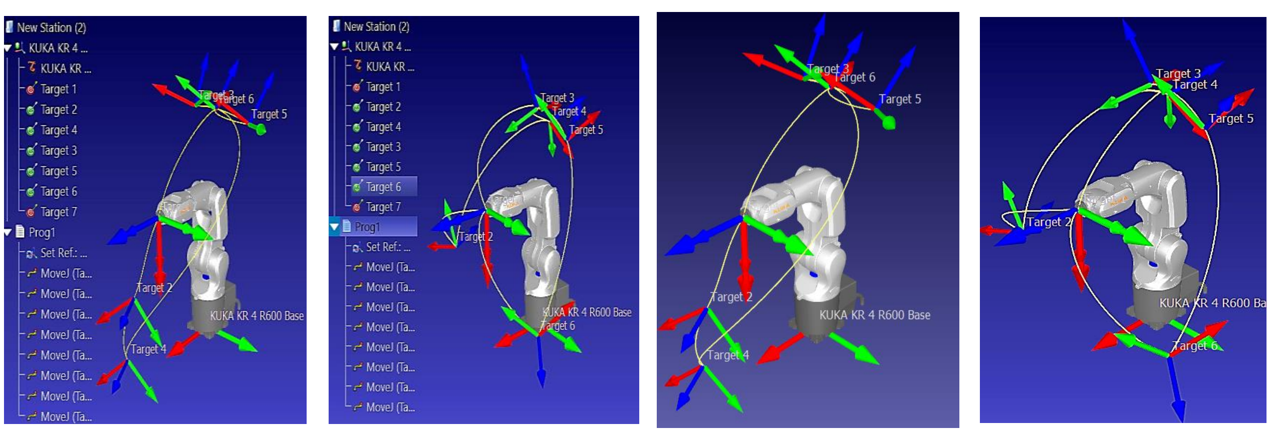


(a) Joints angles during the path



(b) operating time scenarios

Figure 5. GA characteristics during the paths for minimum operating time



Path 1 Path 2 Path 3 Path 4

Figure 6. RoboDK GA paths results simulation

3.1.2. Whale Optimization Technique

The algorithm integrates a unique whale encircling mechanism, where each whale, excluding the leaders (whom embodying trajectories with most efficient traits, guide the populace's movement [25]), adjusts its position towards them, emulating nature's coordinated movement. This mechanism balances exploration and exploitation, allowing the population to converge towards trajectories that minimize reachability time while maintaining diversity.

The trajectory optimization process using WOA for a 6 Degree of Freedom (DOF) robot arm navigating through five specified points involves a series of coherent steps as shown in Algorithm 1. The algorithm begins by initializing a population of potential trajectories, akin to the positions of individual whales in a search space. it was started by initializing a number of 100 whales where each one contains 6 different suggested joints' values to reach 5 different positions as in Equation 4. Each trajectory, which acts as a favored solution, affects the movement of the robotic arm around unique locations.

Fitness is examined, the usage of a task designed to reduce get right of entry to time. It was combined the forward kinematics with the Whale algorithm to solve the objective function [26], passing through updating the algorithm's factors, and ending with a deduction of the angles that guarantee the best path that preserves the shortest reachability time [27]. The principal issues are the smoothness of the trails, the compliance with motion restrictions, and the potential to transport quick to precise regions, leading whales are identified on the idea of their power, that is the pathways with the best characteristics of the movement of the population as a whole [28].

Following the whale encircling mechanism, trajectory positions are updated, and fitness is re-evaluated to account for the adjustments. Convergence is monitored to assess stabilization of optimization objectives. Upon convergence, the trajectory with the minimum reachability time is extracted from the population, representing the optimized solution achieved through the collaborative and adaptive exploration facilitated by WOA [24].

The same set of measurements were performed to show the results of the Whale algorithm-based optimizer shown in Algorithm 1. Table 6 and Figure 7(a) show the angles of rotation that should be used for each joint at each point to guarantee the minimal operating time while Table 7 and Figure 7(b) show the overall amount of time required for the entire voyage.

Figure 8 shows a snapshot from the RoboDK software during trajectory simulation for the results generated from the WOA algorithm while the same simulation settings, as described before in section 4.1.1, were set. The paths drawn in this figure by the simulation software supports the algorithm's findings to solve the reachability problem.

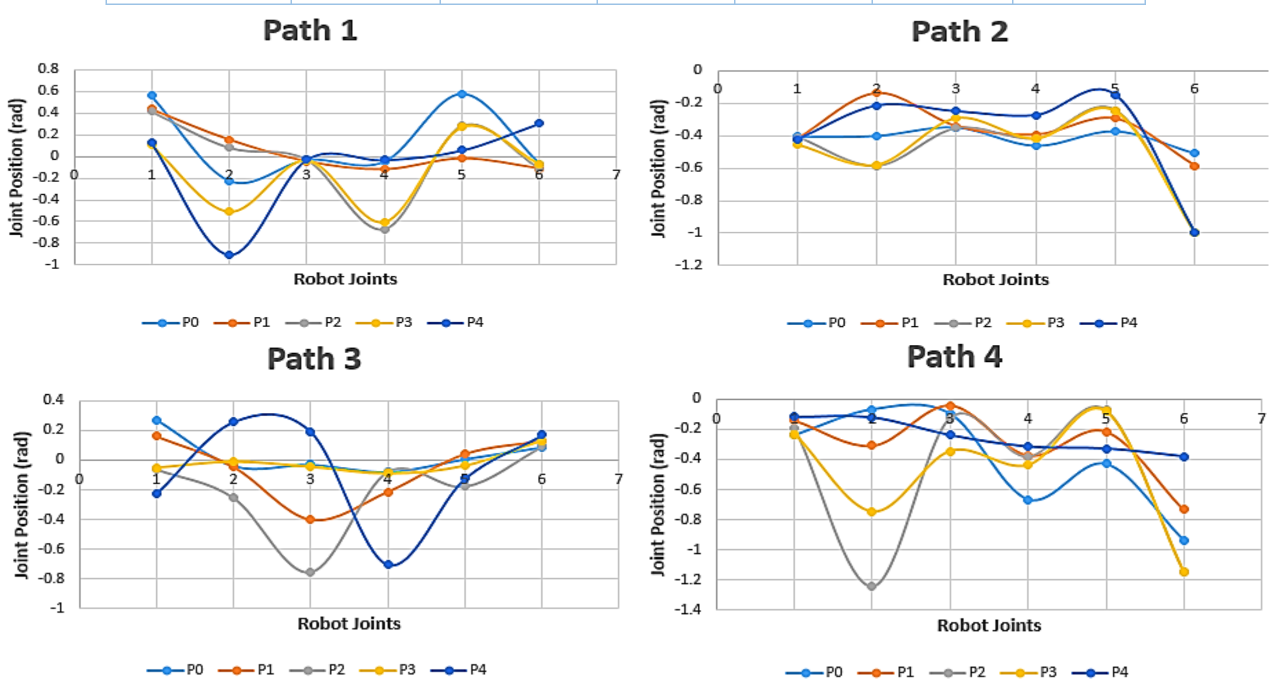
Table 6. WOA angles according to grantee the minimum operating time

Path 1 steps	x (rad)	y (rad)	z (rad)	a (rad)	b (rad)	c (rad)
P0	0.5578	-0.2227	-0.0316	-0.0468	0.5775	-0.0661
P1	0.4377	0.1567	-0.0443	-0.1190	-0.0176	-0.1108
P2	0.4147	0.0803	-0.0305	-0.6724	0.2825	-0.1117
P3	0.1013	-0.5084	-0.0358	-0.6038	0.2694	-0.0727
P4	0.1285	-0.9045	-0.0284	-0.0322	0.0575	0.3063
Path 2 steps	**x (rad)**	**y (rad)**	**z (rad)**	**a (rad)**	**b (rad)**	**c (rad)**
P0	-0.4065	-0.3994	-0.3474	-0.4601	-0.3711	-0.5076
P1	-0.4208	-0.1382	-0.3406	-0.3936	-0.2947	-0.5854
P2	-0.4066	-0.5862	-0.3538	-0.4158	-0.2435	-0.9933
P3	-0.4526	-0.5776	-0.2909	-0.4157	-0.2497	-0.9986
P4	-0.4241	-0.2157	-0.2506	-0.2742	-0.1505	-0.9991
Path 3 steps	**x (rad)**	**y (rad)**	**z (rad)**	**a (rad)**	**b (rad)**	**c (rad)**
P0	0.2710	-0.0374	-0.0293	-0.0776	0.0082	0.0865
P1	0.1604	-0.0454	-0.4014	-0.2132	0.0417	0.1296
P2	-0.0599	-0.2516	-0.7517	-0.0781	-0.1770	0.0921
P3	-0.0491	-0.0092	-0.0432	-0.0876	-0.0353	0.1320

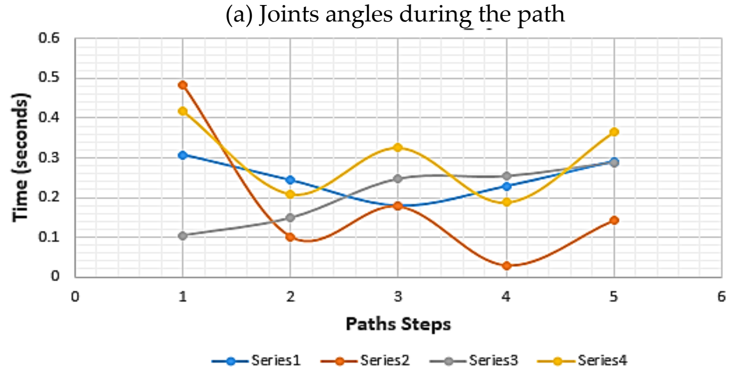
P4	-0.2298	0.2566	0.1930	-0.7025	-0.1227	0.1688
Path 4 steps	**x (rad)**	**y (rad)**	**z (rad)**	**a (rad)**	**b (rad)**	**c (rad)**
P0	-0.2395	-0.0697	-0.0996	-0.6657	-0.4308	-0.9335
P1	-0.1431	-0.3093	-0.0419	-0.3735	-0.2150	-0.7325
P2	-0.1927	-1.2408	-0.1210	-0.3796	-0.0691	-1.1410
P3	-0.2343	-0.7459	-0.3455	-0.4386	-0.0835	-1.1410
P4	-0.1160	-0.1201	-0.2387	-0.3154	-0.3283	-0.3811

Table 7. Operating time for each path step based on WOA

Operating time	**initial to P0**	**P0 to P1**	**P1 to P2**	**P2 to P3**	**P3 to P4**	**Total**
Path 1 (seconds)	0.30671	0.24282	0.178822	0.227723	0.290421	**1.246495**
Path 2 (seconds)	0.481924	0.100807	0.17748	0.028756	0.141784	**0.93075**
Path 3 (seconds)	0.104456	0.148501	0.246317	0.252721	0.287055	**1.039049**
Path 4 (seconds)	0.416845	0.208425	0.32555	0.187408	0.364075	**1.502302**

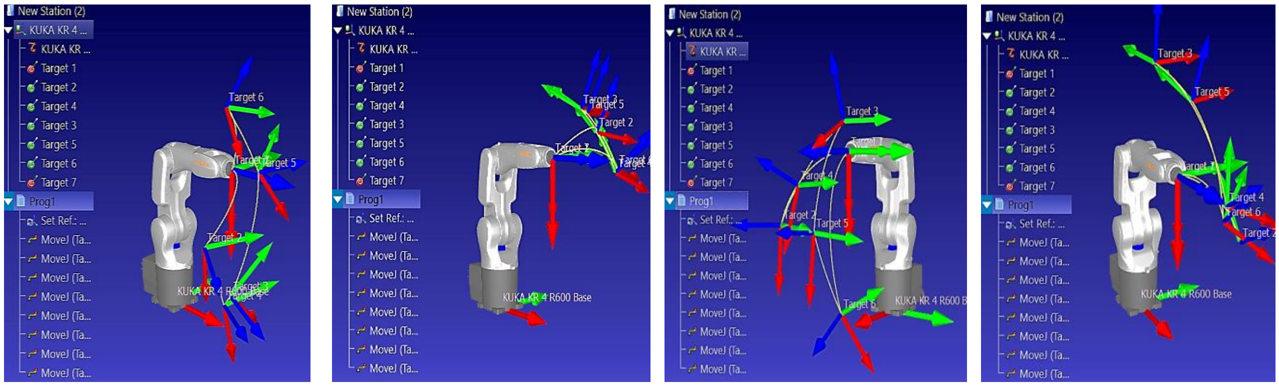


(a) Joints angles during the path



(b) operating time scenarios

Figure 7. WOA characteristics during the paths for minimum operating time



Path 1 Path 2 Path 3 Path 4

Figure 8. RoboDK WOA paths results simulation

4. Discussion

Table 8 presents the outcomes obtained from the two methods introduced in this study, which are GA and WOA algorithms. Each row in the table reflects the total time taken for executing paths using the respective approach outlined in the first column during the investigated scenarios.

An in-depth analysis of the results from the whale optimization algorithm reveals a substantial leap in improvement. This remarkable enhancement can be attributed to the search methodology employed in the whale optimization algorithm. Unlike limiting itself to the exploration of proposed solutions, this algorithm processes solutions during the search, deriving more suitable alternatives and thereby achieving a superior improvement value. This iterative process of dynamic refinement and collaborative exploration within the solution space showcases the efficacy of WOA in minimizing reachability time for a 6 DOF robot arm navigating through a predetermined set of points.

In contrast to Genetic Algorithms (GAs), the Whale Optimization Algorithm (WOA) emerges as a sturdy and versatile optimization tool. Unlike GAs, that could face challenges in untimely convergence, WOA's dynamic balance between exploration and exploitation lets it to navigate answer spaces with better efficacy. The social conduct-inspired mechanism of WOA permits it to conform dynamically to evolving hassle landscapes, a feature that complements its global optimization capabilities as compared to GAs. WOA's ability to deal with a various variety of optimization challenges with fewer algorithmic parameters and its adaptability to complex, non-linear problem role it as a favorable preference, especially when confronted with complicated and dynamic optimization situations. This evaluation considers various factors influencing the algorithms, encompassing the number of proposed solutions and search attempts.

In general, we noticed that the whale optimization algorithm solves the specified reachability problem with a total trip time lower than the solutions generated by genetic optimization algorithm with a factor of at least 2.5 as shown in Table 8. The WOA algorithm's adaptability and nature-inspired collaborative behavior position it as a promising tool for achieving efficient trajectory optimization in complex robotic scenarios.

Table 8. Summary of the optimization techniques results.

NO.	Techniques	Average Path 1 time (seconds)	Average Path 2 time (seconds)	Average Path 3 time (seconds)	Average Path 4 time (seconds)
1	GA	4.04251	5.763402	2.658827	5.265366
2	WOA	1.246495	0.93075	1.039049	1.502302

5. Conclusion and Future Works

This research focuses on trajectory control for a high-degree-of-freedom robotic arm, specifically applied to the KUKA KR 4 R600 robot. The primary objective was to investigate the reachability time problem across sets of points while minimizing the overall trip time through four different paths. Two distinct approaches (GA and WOA) have been employed to achieve the desired behavior.

The genetic optimization algorithm is designed to search for optimum solutions. This involves utilizing a population of candidate solutions and iteratively improving them through selection, crossover, and mutation operations. Furthermore, the whale optimization algorithm, provides more accurate and smooth exploration of global solutions. This approach facilitates the identification of the best path, ensuring the shortest time to execute a complete trajectory for the robotic arm.

The case study, encompassing four different paths, demonstrates that the total trip time for the robotic arm to reach all points using the whale optimization algorithm is less than that calculated by the genetic algorithm by a factor of at least 2.5. These findings highlight the efficacy of the whale optimization algorithm in producing outstanding outcomes, fostering a significant improvement in reachability time and reducing wear and strain on the equipment's motors.

Moving forward, our future work will extend beyond the reachability problem, incorporating considerations of power consumption and obstacle avoidance during the robotic arm’s trajectory control. Also, we intend to investigate more optimization algorithms applied to various robotic arm control problems.

Acknowledgements

We thank Horus University (Egypt) for its explicit support throughout this research and for providing access to KUKA robotic arm used in the case study.

Author Contributions

All authors contributed to the research, development, and testing of the proposed techniques. Mahmoud A. A. Mousa, Abdelrahman T. Elgohr and Hatem A. Khater wrote the manuscript. Abdelrahman T. Elgohr and Hatem A. Khater led the mathematical modeling for the robotic arm. Mahmoud A. A. Mousa was responsible for developing the proposed optimization techniques and its implementation was done by Abdelrahman T. Elgohr. The testing was performed by Abdelrahman T. Elgohr and analyzed by Mahmoud A. A. Mousa and Hatem A. Khater.

Data Availability

All data generated and analyzed during this study are included in this article.

Declarations

Ethics approval and consent to participate: All applicable institutional and national guidelines were followed.

Consent for Publication: Informed consent was obtained from all the co-authors of this publication.

Competing interests: The authors declare that they have no conflict of interest.

References

[1] Kazim Raza, Tauseef Aized Khan and Naseem Abbas, “Kinematic analysis and geometrical improvement of an industrial robotic arm”, *Journal of King Saud University - Engineering Sciences*, Print ISSN: 1018-3639, Online ISSN: 2213-1558, Vol. 30, No. 3, pp. 218-223, July 2018, Published by Elsevier B.V., DOI: 10.1016/j.jksues.2018.03.005, Available: https://www.sciencedirect.com/science/article/pii/S1018363917304464.

[2] Alessandro Gasparetto, Paolo Boscariol, Albano Lanzutti and Renato Vidoni, “Trajectory Planning in Robotics”, *Mathematics in Computer Science*, Print ISSN: 1661-8270, Online ISSN: 1661-8289, Vol. 6, No. 1, pp. 269-279, 30th August 2012, Published by Springer, DOI: 10.1007/s11786-012-0123-8, Available: https://link.springer.com/article/10.1007/s11786-012-0123-8.

[3] Mahmoud A. A. Mousa, Abdelrahman Tarek Elgohr and Hatem Khater, “Path Planning for a 6 DoF Robotic Arm Based on Whale Optimization Algorithm and Genetic Algorithm”, *Journal of Engineering Research*, Print ISSN: 2356-

9441, Online ISSN: 2735-4873, Vol. 7, No. 5, pp. 160-168, November 2023, Published by Arab Journals Platform, DOI: 10.21608/erjeng.2023.237586.1256, Available: https://erjeng.journals.ekb.eg/article_325086.html.

[4] Mariana Ratiu and Mariana Adriana Prichici, "Industrial robot trajectory optimization- a review", in *Proceedings of MATEC Web of Conferences, Annual Session of Scientific Papers IMT ORADEA 2017*, Sanmartin, Romania, 27-29 May 2017, E-ISSN: 2261-236X, pp. 1-6, Published by MATEC, DOI: 10.1051/matecconf/201712602005, Available: https://www.matec-conferences.org/articles/matecconf/pdf/2017/40/matecconf_imtoradea2017_02005.pdf.

[5] Hind Hadi Abdulridha and Tahseen Fadhel Abaas, "Differential Motion Analysis of Lab-Volt R5150 Robot System", *Global Journal of Engineering Science and Research Management*, ISSN: 2349-4506, Vol. 5, No. 10, pp. 152-160, October 2017, Published by GJESRM, DOI: 10.5281/zenodo.1034505, Available: https://www.gjesrm.com/Issues%20PDF/Archive-2017/October-2017/14.pdf.

[6] Giuseppe Carbone and Fernando Gomez-Bravo, Motion and Operation Planning of Robotic Systems: Background and Practical Approaches, 1st ed. New York, USA: Springer, 2015, ISBN: 978-3-319-14704-8, Available: https://link.springer.com/book/10.1007/978-3-319-14705-5.

[7] Alexander Heim and Oskar Von Stryk, "Trajectory optimization of industrial robots with application to computer-aided robotics and robot controllers", *Optimization*, ISSN: 0233-1934, Vol. 47, No. 3, pp. 407–420, January 2000, Published by Taylor & Francis, DOI: 10.1080/02331930008844489, Available: https://www.tandfonline.com/doi/abs/10.1080/02331930008844489.

[8] Gregory Kahn, Peter Sujan, Sachin Patil, Shaunak Bopardikar, Julian Ryde *et al.*, "Active exploration using trajectory optimization for robotic grasping in the presence of occlusions", in *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, ISSN: 1050-4729, 26-30 May 2015, pp. 4783–4790, Published by IEEE Xplore, DOI: 10.1109/ICRA.2015.7139864, Available: https://ieeexplore.ieee.org/document/7139864.

[9] Jiangping Mei, Fan Zhang and Han Yan, "Trajectory optimization of the 6-degrees-of-freedom high-speed parallel robot based on B-spline curve", *Science Progress*, ISSN: 0036-8504, Vol. 103, No. 1, pp. 1-26, January 2020, Published by Sage, DOI: 10.1177/0036850419880115, Available: https://journals.sagepub.com/doi/10.1177/0036850419880115.

[10] Rabab Benotsmane, László Dudás and György Kovács, "Trajectory Optimization of Industrial Robot Arms Using a Newly Elaborated 'Whip-Lashing' Method", *Applied Sciences*, ISSN: 2076-3417, Vol. 10, No. 23, p. 8666, December 2020, Published by MDPI, DOI: 10.3390/app10238666, Available: https://www.mdpi.com/2076-3417/10/23/8666.

[11] Chandan Choubey and Jyoti Ohri, "Optimal Trajectory Generation for a 6-DOF Parallel Manipulator Using Grey Wolf Optimization Algorithm", *Robotica*, ISSN: 0263-5747, Vol. 39, No. 3, pp. 411–427, May 2020, Published by Cambridge University Press, DOI: 10.1017/S0263574720000442, Available: https://dx.doi.org/10.1017/S0263574720000442.

[12] Tao Wang, Shengda Yao and Shiqiang Zhu, "Energy-saving trajectory optimization of a fluidic soft robotic arm", Smart Materials and Structures, ISSN: 1361-665X, Vol. 31, No. 11, p. 115011, 13 October 2022, Published by IOP science, DOI: 10.1088/1361-665X/ac9768, Available: https://iopscience.iop.org/article/10.1088/1361-665X/ac9768.

[13] Mahmoud A. A. Mousa, Abdelrahman T. Elgohr and Hatem A. Khater, "Trajectory Optimization for a 6 DOF Robotic Arm Based on Reachability Time", *Annals of Emerging Technologies in Computing (AETiC)*, ISSN: 2516-0281, Vol. 8, No. 1, pp. 22–35, January 2024, Published by International Association for Educators and Researchers (IAER), DOI: 10.33166/AETiC.2024.01.003, Available: https://aetic.theiaer.org/archive/v8/v8n1/p3.html.

[14] Stanislav Števo, Ivan Sekaj and Martin Dekan, "Optimization of Robotic Arm Trajectory Using Genetic Algorithm", in *Proceedings of the 19th World Congress, The International Federation of Automatic Control 2014*, Cape Town, South Africa, 24-29 August 2014, ISSN: 1474-6670, pp. 1748-1753, Published by Elsevier IFAC Proceedings, DOI: 10.3182/20140824-6-ZA-1003.01073, Available: https://www.sciencedirect.com/science/article/pii/S1474667016418653.

[15] Abdelrahman O. Ali, Mohamed R. Elmarghany, Mohamed M. Abdelsalam, Mohamed Nabil Sabry and Ahmed M. Hamed, "Closed-loop home energy management system with renewable energy sources in a smart grid: A comprehensive review", *Journal of Energy Storage*, Print ISSN: 2352-152X, Online ISSN: 2352-1538, Vol. 50, pp. 1-24, June 2022, Published by Elsevier, DOI: 10.1016/j.est.2022.104609, Available: https://www.sciencedirect.com/science/article/pii/S2352152X22006259.

[16] Manoj Kumar, Mohammad Husain, Naveen Upreti and Deepti Gupta, "Genetic Algorithm: Review and Application", *International Journal of Information Technology and Knowledge Management*, Print ISSN: 1461-4111, Online ISSN: 1741-5179, Vol. 2, No. 2, pp. 451-454, 1st December 2010, DOI: 10.2139/ssrn.3529843, Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3529843.

[17] Siew Mooi Lim, Abu Bakar Md. Sultan, Md. Nasir Sulaiman, Aida Mustapha and K. Y. Leong, "Crossover and mutation operators of genetic algorithms", *International Journal of Machine Learning and Computing*, ISSN: 2010-3700, Vol. 7, No. 1, pp. 9–12, February 2017, DOI: 10.18178/ijmlc.2017.7.1.611, Available: https://www.ijmlc.org/index.php?m=content&c=index&a=show&catid=69&id=704.

[18] Sourabh Katoch, Sumit Singh Chauhan and Vijay Kumar, "A review on genetic algorithm: past, present, and future", *Multimedia Tools and Applications*, Print ISSN: 1380-7501, Online ISSN: 1573-7721, Vol. 80, No. 5, pp. 8091-8126, February 2021, Published by Springer, DOI: 10.1007/s11042-020-10139-6, Available: https://link.springer.com/article/10.1007/s11042-020-10139-6.

[19] Jiayan Zhang, Qingxi Meng, Xugang Feng and Hao Shen, "A 6-DOF robot-time optimal trajectory planning based on an improved genetic algorithm", *Robotics and Biomimetics*, ISSN: 2197-3768, Vol. 5, No. 1, Article No. 3, December 2018, Published by Springer Nature, DOI: 10.1186/s40638-018-0085-7, Available: https://jrobio.springeropen.com/articles/10.1186/s40638-018-0085-7.

[20] Raffaele Di Gregorio, "Kinematics of a new spherical parallel manipulator with three equal legs: The 3-URC wrist", *Journal of Robotic Systems*, Print ISSN: 1556-4959, Online ISSN: 1556-4967, Vol. 18, No. 5, pp. 213-219, 10th April 2001, Published by Wiley Periodicals, DOI: 10.1002/rob.1017, Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.1017.

[21] Mahmoud A. A. Mousa, Abdelrahman T. Elgohr and Hatem A. Khater, "A Novel Hybrid Deep Neural Network Classifier for EEG Emotional Brain Signals", *International Journal of Advanced Computer Science and Applications (IJACSA)*, ISSN: 2156-5570, Vol. 15, No. 6, pp. 1045-1055, 2024, DOI: 10.14569/IJACSA.2024.01506107, Available: https://thesai.org/Publications/ViewPaper?Volume=15&Issue=6&Code=ijacsa&SerialNo=107.

[22] H. Azaria and A. Dvir, "Algorithm optimization using a rule-based system. A case study: The Direct Kinematic Solution in robotics", *Journal of Intelligent & Robotic Systems*, Print ISSN: 0921-0296, Online ISSN: 1573-0409, Vol. 8, pp. 309-324, December 1993, Published by Springer, DOI: https://doi.org/10.1007/BF01257947, Available: https://link.springer.com/article/10.1007/BF01257947.

[23] Devendra Rawat, Mukul Kumar Gupta and Abhinav Sharma, "Trajectory Control of Robotic Manipulator using Metaheuristic Algorithms", *International Journal of Mathematical, Engineering and Management Sciences*, ISSN: 2455-7749, Vol. 8, No. 2, pp. 264–281, April 2023, DOI: 10.33889/IJMEMS.2023.8.2.016, Available: https://www.ijmems.in/article_detail.php?vid=8&issue_id=37&article_id=487.

[24] Juan Du, Jie Hou, Heyang Wang and Zhi Chen, "Application of an improved whale optimization algorithm in time-optimal trajectory planning for manipulators", *Mathematical Biosciences and Engineering*, ISSN: 1551-0018, Vol. 20, No. 9, pp. 16304–16329, August 2023, Published by AIMS Press, DOI: 10.3934/mbe.2023728, Available: https://www.aimspress.com/article/doi/10.3934/mbe.2023728.

[25] Jolly Atit Shah, S. S. Rattan and B. C. Nakra, "End-Effector Position Analysis Using Forward Kinematics For 5 Dof Pravak Robot Arm", *International Journal of Robotics and Automation (IJRA)*, Vol. 3, No. 3, pp. 112-116, September 2013, Print ISSN: 2089-4856, Online ISSN: 2722-2586, Published by Institute of Advanced Engineering and Science (IAES), DOI: 10.11591/ijra.v2i3.pp112-116, Available: https://ijra.iaescore.com/index.php/IJRA/article/view/921.

[26] Ting Wang, Zhijie Xin, Hongbin Miao, Huang Zhang, Zhenya Chen *et al.*, "Optimal Trajectory Planning of Grinding Robot Based on Improved Whale Optimization Algorithm", *Mathematical Problems in Engineering*, ISSN: 1563-5147, Vol. 2020, pp. 1-8, August 2020, Published by Hindawi, DOI: 10.1155/2020/3424313, Available: https://onlinelibrary.wiley.com/doi/10.1155/2020/3424313.

[27] Abdelrahman T. Elgohr, Mohamed S. Elhadidy, M.A. Elazab, Raneem Ahmed Hegazii and Moataz M. El Sherbin., "Multi-Classification Model for Brain Tumor Early Prediction Based on Deep Learning Techniques", *Journal of Engineering Research*, ISSN: 2356-9441, Vol. 8, No. 3, pp. 1-9, September 2024, Published by Arab Journals Platform, Available: https://digitalcommons.aaru.edu.jo/cgi/viewcontent.cgi?article=1646&context=erjeng.

[28] Mohamed S. Elhadidy, Waleed S. Abdalla, Alaa A. Abdelrahman, Soliman Elnaggar and Mostafa Elhosseini, "Assessing the accuracy and efficiency of kinematic analysis tools for six-DOF industrial manipulators: The KUKA robot case study", *AIMS Mathematics*, ISSN: 2473-6988, Vol. 9, No. 6, pp. 13944–13979, 2024, Published by AIMS Press, DOI: 10.3934/math.2024678, Available: https://www.aimspress.com/article/doi/10.3934/math.2024678.