U.P.B. Sci. Bull., Series D, Vol. 86, Iss. 1, 2024 ISSN 1454-2358

PERFORMANCE EVALUATION OF POPULATION-BASED METAHEURISTIC ALGORITHMS FOR SOLVING THE INVERSE KINEMATIC OF A CLASS OF CONTINUUM ROBOTS

Amel DJEDILI [1,*], Ammar AMOURI [1], Yazid LAIB DIT LEKSIR[2] and Ayman BELKHIRI[1]

The Inverse Kinematic (IK) plays a crucial role in effectively controlling robots during real-time applications. However, solving the IK problem for continuum robots has proven to be an extremely complex and challenging task so far. Since analytical and numerical methods may not provide satisfactory solutions, Population-based Metaheuristic Algorithms (PbMAs) can offer an alternative approach. By leveraging Forward Kinematic (FK) relationships, various algorithms can be utilized. This research paper aims to evaluate the performance of five PbMAs, namely: Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Bat Algorithm (BA), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO). The evaluation will focus on both execution time and position error when these algorithms are employed to solve the IK problem for a specific continuum robot known as Dual-Cross-Module Sections Cable-Driven Continuum Robot (DCM-S-CDCR). Firstly, an overview of each algorithm is provided, along with the FK relationships of the robot being examined and the mathematical formulation of the IK problem for optimization. Then, a series of numerical experiments are conducted. The findings reveal that out of all tested algorithms, PSO proves to be the most suitable as it delivers satisfactory tracking accuracy within a reasonable timeframe. Specifically, GA followed by PSO demonstrated superior accuracy when it came to position error; meanwhile, PSO and ABC ranked as the two swiftest algorithms respectively. In contrast, BA showcased inferior outcomes across both indicators whereas ACO's performance stood second last in terms of effectiveness.

Keywords: Continuum robot, cable-driven continuum robot, inverse kinematic, population-based metaheuristic algorithms, optimization.

1. Introduction

Recently, the class of hyper-redundant robots known as continuum robots has gained a lot of attention in the robotic research community. These robots have been proven to be superior to rigid-link robot counterparts when it comes to their flexibility, dexterity, and ability to safely interact with humans [1]. Because of these exceptional characteristics, continuum robots are well-suited for various

[1] Department of Mechanical Engineering, Laboratory of Mechanics, Frères Mentouri Constantine 1, University, Algeria
[2] Department of Mechanical Engineering, University of L'Arbi Ben M'hidi, Oum el Bouaghi, Algeria.
[*] E-mail: ameldjedili85@gmail.com (corresponding author)

applications including exploration and inspection in confined environments [2], medical fields [3], and rescue operations [4].

Due to their highly nonlinear mathematical expressions, continuum robots are more difficult to model compared to robots with rigid-link. However, in an effort to find practical solutions, researchers have relied on simplifications and assumptions to facilitate the modeling tasks. The commonly used kinematic approach is the Constant Curvature Kinematic Approach (CCKA) [5–6]. Consequently, researchers have developed numerous Forward Kinematic (FK) models for different continuum robots using various methods and theories such as [5–9]. Despite this simplification, solving Inverse Kinematic (IK) models of continuum robots still presents a challenge.

The fundamental problem of IK modeling in continuum robots is the infinite number of Degrees-of-Freedom (DoFs) provided by FK models. This results in multiple solutions or even an infinite number of solutions for any given IK problem. In the case of a regular robot with 6 DoFs, analytical methods such as that described in [10] can be utilized. However, for robots with more than 6 DoFs, numerical methods and optimization algorithms like those explained in [11–12] may be employed as they work independently from robot redundancy. Both approaches have their advantages and disadvantages. Closed-form solutions offer the advantage that all IK solutions are expressed as a function depending on End-Effector (EE) pose variables and are computationally efficient, providing all possible IK solutions for a given EE pose along with natural ones. On the other hand, deriving closed-form expressions becomes challenging when dealing with robots exceeding 6 DoFs.

Regarding numerical methods, they possess the advantage of being applicable regardless of the number of DoFs possessed by the robot. Nonetheless, there are various drawbacks associated with these methods such as high computational costs and lengthy execution times, along with common occurrences of singularities. Consequently, optimization algorithms persist as an alternative solution to mitigate these issues. However, it is worth mentioning that while optimization algorithms have been successfully employed in various research areas to resolve different optimization problems, their implementation within continuum robot modeling remains limited [13–16]. Therefore, it would be beneficial to investigate and evaluate the performance capabilities of certain optimization algorithms in solving the IK problem for continuum robots. Notably among these algorithms is the class known as Population-based Meta-heuristic Algorithms (PbMAs).

PbMAs are a class of approaches that use a set of candidate solutions and population characteristics to guide the search iteratively towards near-optimal or optimal solutions in a reasonable amount of time, thanks to their ability for global exploration and local exploitation. This class includes, but is not restricted to the following algorithms: Artificial Bee Colony (ABC) [17], Ant Colony Optimization (ACO) [18], Bat Algorithm (BA) [19], Genetic Algorithm (GA) [20], and Particle Swarm Optimization (PSO) [21]. Among the various algorithms used to solve the IK problem in continuum robots, only GA and PSO have been previously applied [12]. A comprehensive comparative study was conducted to evaluate their performance. Additionally, another study [16] performed a comparative analysis of three algorithms: GA, PSO, and ABC. Unfortunately, this study was considered superficial and did not sufficiently evaluate the efficacy of these algorithms. The aim of this paper is to provide an in-depth evaluation of five different algorithms: GA and PSO which were previously studied along with ABC as well as BA and ACO when dealing with IK solutions for a specific type of continuum robot called Dual-Cross-Module Sections Cable-Driven Continuum Robot (DCM-S-CDCR). The specific focus will be on evaluating execution time as well as position error.

To do so, the subsequent sections of the paper are organized as follows: The next section briefly introduces the selected PbMAs. Section 3 summarizes the FK of the studied DCM-S-CDCR and then describes the formulation of IK problem. Section 4 presents numerical experiments and summarizes a comparison of algorithm performances in terms of execution times and tracking accuracy. Concluding remarks and future scope for this work are discussed in Section 5.

2. Population-based metaheuristic algorithms

In this section, we briefly introduce the five population-based metaheuristic algorithms studied in this work.

2.1. Artificial Bee Colony algorithm

The Artificial Bee Colony (ABC) algorithm was proposed by D. Karaboga in 2005 [17]. The algorithm imitates the foraging behavior of honey bees that fly around their surroundings in search of high-quality food sources. It consists of three key components: employed bees, onlooker bees, and scouts. ABC begins with the random initialization of scout and employed bees, both primarily focusing on mutation. Selection is related to honey or objective; crossover is not explicitly present in this algorithm. In this method, employed bees examine their food sources using fitness values assigned to each bee's associated food source, sharing this information to recruit onlooker bees. Based on this shared information from the employed bees, the onlooker-bees make a decision when choosing a particular food source among options available. The quality evaluation made by an individual-employed bee for a given food source is defined as its “fitness value” associated with its position within it. Mathematically speaking, the expression for a specific $prob_i$ food source can be represented as follows:

$$prob_i = \frac{fit_i}{\sum_{q=1}^{N} fit_i} \tag{1}$$

where fit_i is the fitness value of the solution i evaluated by its employed bee, which is proportional to the nectar amount of the food source in position i, and N represents the number of solutions in population.

Accordingly, the employed bees share information with onlookers, and a potential new food source can be mathematically represented by replacing the old one using the following expression:

$$\mathrm{x}_i^{new} = \mathrm{x}_i^{old} + \mathrm{r}_i\left(\mathrm{x}_i^{old} - x_\mathrm{r}\right) \tag{2}$$

where x_i^{new} is the new food position, x_i^{old} is the previously assigned food source, r_i is a random number with uniform distribution between 0 and 1 used for adjustment, and x_r is the randomly selected food source $i \in \{1, 2, ..., N\}$.

2.2. Ant Colony Optimization algorithm

The Ant Colony Optimization (ACO) algorithm was proposed by M. Dorigo in 1992 [18]. This algorithm mimics the foraging behavior of ants, who use pheromones to find shorter paths between their colony and food sources. The effectiveness of the algorithm is closely linked to pheromone concentration, as it determines the probability for ants to select a route from one node to another using the following Equation:

$$prob_{i,j} = \frac{\mathrm{C}_{i,j}^{\alpha}\mathrm{d}_{i,j}^{\beta}}{\sum_{i,j=1}^{N} \mathrm{C}_{i,j}^{\alpha}\mathrm{d}_{i,j}^{\beta}} \tag{3}$$

where $\mathrm{C}_{i,j}$ is the pheromone concentration on the route between nodes i and j, $\mathrm{d}_{i,j}$ is the desirability of that same route, and α and β are positive parameters of influence.

2.3. Bat Algorithm

The Bat Algorithm (BA), proposed by X. S. Yang in 2010 [19], draws its inspiration from the echolocation abilities exhibited by small bat species. These bats emit exceptionally loud sonic pulses and then listen attentively for the echoes that bounce off their environment, employing this technique to either detect prey or navigate through challenging conditions during twilight hours.

BA starts with the random initialization of the population in the search space. Each bat b is associated with a velocity v_b^t, position vector x_b^t, variable frequency f_b, pulse emission factor pe_b and loudness factor l_b. The current best solution among all bats is denoted as x_{best}. At each iteration t, each bat updates its velocity and position vector according to the following Equations:

$$v_b^{t+1} = v_b^t + f_b\left(x_b^t - x_{best}\right) \tag{4}$$

$$x_b^{t+1} = x_b^t + v_b^{t+1} \tag{5}$$

such that:

$$f_b = F_{max}\,\gamma\left(F_{max} - F_{min}\right) \tag{6}$$

where f_{min} and f_{max} represent the minimum and maximum frequencies, and γ is a random vector uniformly distributed between 0 and 1.

In addition, the loudness and pulse emission rates are regulated by the following Equations:

$$l_b^{t+1} = \rho l_b^t \tag{7}$$

$$pe_b^{t+1} = pe_b^0\left(1 - e^{-\beta t}\right) \tag{8}$$

where $0 < \rho < 1$ and $\beta > 0$ are constant factors, pe_b^0 represents the initial value of emission rate.

2.4. Genetic Algorithm

The Genetic Algorithm (GA), proposed by Holland in 1975 [20], is a search optimization algorithm that is based on the mechanics of the natural selection process. More specifically, it utilizes biological factors like reproduction (selection), mutation, and crossover. In GA, convergence improvement often involves implementing the elitism procedure.

2.5. Particle Swarm Optimization algorithm

The Particle Swarm Optimization (PSO) algorithm was proposed by Kennedy and Eberhart in 1995 [21]. It simulates the movement of flocking birds and their interactions with neighbors within a swarm. The initial step in PSO involves randomly initializing a group of particles throughout the search space. Each particle, denoted as p, moves within this search space with its own velocity vector v_p^t and position vector x_p^t. To update its velocity, each particle considers both the best previous positions P_{best} it has attained so far as well as the global best position g_{best} achieved by its neighboring particles using the following equation:

$$v_p^{t+1} = \omega v_p^t + c_1 r_1\left(P_{best} - x_p^t\right) + c_2 r_2\left(g_{best} - x_p^t\right) \tag{9}$$

where ω is the inertia weight, c_1 and c_2 are positive constants, and r_1 and r_2 are random numbers with a uniform distribution between 0 and 1.

In addition, the position vector of each particle p is updated using the following Equation:

$$x_p^{t+1} = x_p^t + v_p^{t+1} \tag{10}$$

3. IK problem formulation for optimization process

3.1. Material and method

To evaluate the effectiveness of the five PbMAs, a case study was undertaken using a continuum robot known as Dual-Cross-Module Sections Cable-Driven Continuum Robot (DCM-S-CDCR) [22]. The DCM-S-CDCR comprises four identical modules connected in series, with each module having one degree-of-freedom (1-DOF). Figure 1(a) presents an illustration of the scheme design for DCM-S-CDCR, while Table 1 provides information regarding its modules' basic geometric parameters and their corresponding kinematic nomenclature. The deformation of module k, with $k = 1,\ldots,4$, upon actuation can be defined by three geometrical parameters $(\theta_k, \varphi_k, \kappa_k)$, as shown in Figure 1(b) where θ_k and φ_k represent the bending and orientation angles respectively. $\kappa_k = \theta_k / l_k$ denotes the curvature, and l_k signifies the length of module k.

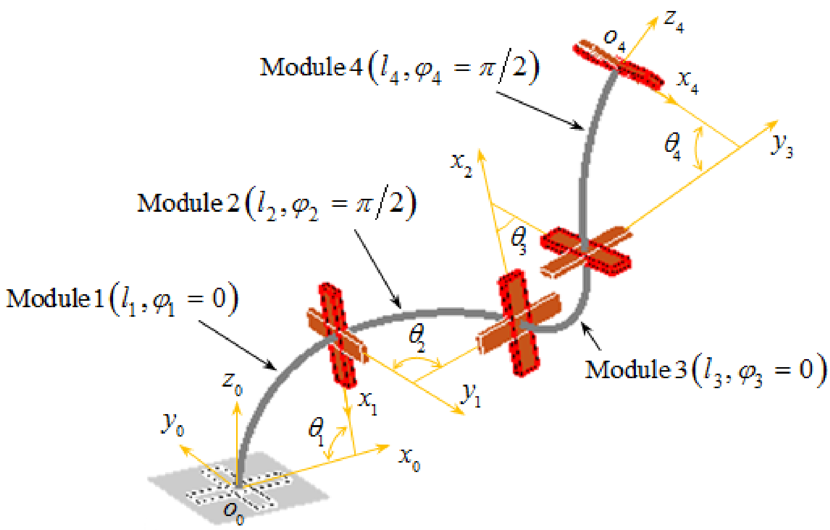


Fig. 1. Schematic of DCM-S-CDCR and its kinematic nomenclature. Note that the intermediate components in the four modules are not shown, while the sheet-like backbones have been represented as circular arcs for better visual.

Table 1

Geometric parameters of the DCM-S-CDCR.

	Module 1	Module 2	Module 3	Module 4
Bending angle (rad)	θ_1	θ_2	θ_3	θ_4
Orientation angle (rad)	$\varphi_1 = 0$	$\varphi_2 = \pi/2$	$\varphi_3 = 0$	$\varphi_4 = \pi/2$
Module's length (mm)	$l_1 = 90$	$l_2 = 90$	$l_3 = 90$	$l_4 = 90$
Bending axis (u_{k-1})	y_0	x_1	y_2	x_3

In this study, the forward kinematics of the DCM-S-CDCR was modeled by considering the constant curvature kinematic assumption [5]. Consequently, the FK can be formally expressed as follows:

$$\mathrm{x} = f_{\text{independent}}(\mathrm{k}), \text{with } \mathrm{k} = f_{\text{specific}}(\mathrm{q}) \tag{11}$$

where x, k and q represent the task space variables, configuration space, and actuator space respectively. $f_{\text{independent}}$ is a robot-independent mapping that transforms the configuration space to the task space, whereas f_{specific} is a robot-specific mapping that transforms the actuator space to the configuration space.

This part focuses on robot-independent mapping. Consequently, the homogeneous transformation matrix defining the position and orientation of the robot's end-tip with respect to the reference frame $(o_0 x_0 y_0 z_0)$ can be calculated as follows:

$$\mathrm{H} = \prod_{k=1}^{n=4} \mathrm{H}_k(\mathrm{k}_k) \tag{12}$$

where H_k is the (4×4) homogeneous transformation matrix that contains both rotational R_k and translational x_k, defining the reference frame $(o_k x_k y_k z_k)$ in frame $(o_{k-1} x_{k-1} y_{k-1} z_{k-1})$. As stated in [22], the position vector and rotation matrix are given as follows:

$$x_k = \frac{1}{\kappa_k}\left\{(1-\mathrm{c}(\theta_k))\mathrm{c}(\varphi_k),\ (1-\mathrm{c}(\theta_k))\mathrm{s}(\varphi_k),\ \mathrm{s}(\theta_k)\right\}^T \tag{13}$$

$$\mathrm{R}_k = \mathrm{rot}(u_{k-1}, \theta_k) \tag{14}$$

where the abbreviations $\mathrm{c}(.)$ and $s(.)$ respectively represent $\cos(.)$ and $\sin(.)$, and u_{k-1} denotes the module's bending axis defined in Table 1.

3.2. IK problem statement and the objective function

By definition, Inverse Kinematic (IK) involves the task of finding a potential and achievable solution that enables a robotic system to attain a predetermined posture in terms of position and orientation. In mathematical terms, the IK of a continuum robot according to CCKA can be formally expressed as follows [5]:

$$\mathrm{k} = f_{\text{independent}}^{-1}\left(\mathrm{x}\right) \tag{15}$$

However, the main issue with Eq. 15 is the challenge of obtaining analytical expressions for the IK function. Additionally, applying numerical methods becomes difficult, particularly for multi-section continuum robots. To tackle this problem, metaheuristic algorithms can be employed. This study utilizes each of the aforementioned algorithms to guide a search using a set of particles in order to find optimal or nearly optimal solutions to the IK problem for a continuum robot. In pursuit of this objective, solution quality is assessed using a quadratic objective function that naturally measures the distance error between the robot's end-tip $x_{\text{generated}}$ and its required target x_{desired}. Thus, the objective function can be written as follows:

$$f_{obj} = \frac{1}{2}\left(x_{\text{desired}} - x_{\text{generated}}\right)^2 \tag{16}$$

where $x_{\text{generated}}$ represents the first three components of the fourth column of the $\left(4\times 4\right)$ homogeneous transformation matrix that defines the robot-independent transformation given by Eq. (12).

3.3. Boundary conditions

Naturally, for a stationary continuum robot, two boundary conditions must be satisfied regarding the mapping of the curve of the central axis of the robot to both its base and end-tip. Formally, these terms can be expressed as follows:

$$\mathrm{x}_0 = \mathrm{x}_{\text{base}}, \mathrm{x}_n = \mathrm{x}_{\text{desired}} \tag{17}$$

3.4. Physical limitations

To ensure effective operation of the robot and prevent any overlapping in their bending sections (refer to [5]), it is important to take into account additional constraints on actuator variables. In order to meet these requirements, these limitations can be defined in terms of bending angles as follows:

$$\theta_k^{\min} \le \theta_k \le \theta_k^{\max}, \text{ with } k = 1,\ldots,4. \tag{18}$$

3.5. Problem Setting

To determine the bending angles θ_k, with $k = 1,\ldots,4$, that allow the robot to track a specified reference trajectory accurately, the aforementioned population-based metaheuristic algorithms are employed. These algorithms aim to minimize the objective function described in Eq. (16). At each sampling instant, the applied optimization algorithm generates sequences of bending angles as solutions to the IK problem, but only the last one is considered to be the optimal solution for application on the robot.

In the search space, the dimension of the problem is 2n. The particle or individual's coordinates in each optimization algorithm represent the variables of the optimization problem, specifically referring to bending angles in this case. The population within all algorithms can be regarded as a set consisting of potential solutions. Generally speaking, Eq. (16) often possesses an infinite number of solutions. To effectively choose a suitable solution from these multiple ones, additional constraints can be incorporated into the objective function.

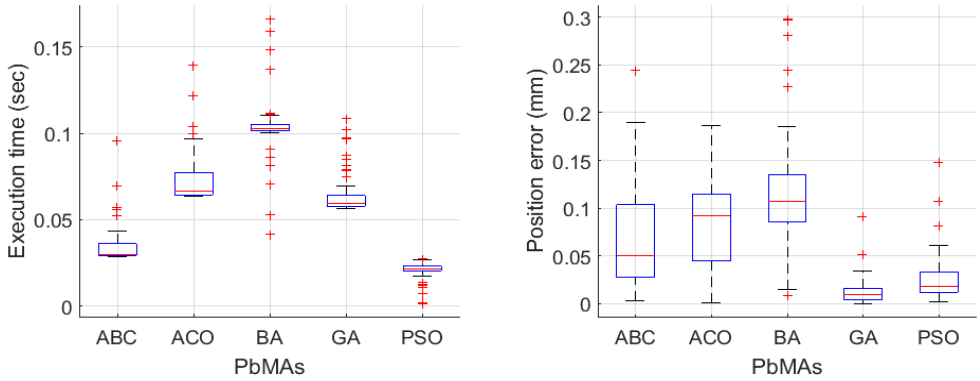
4. Numerical experiments

To evaluate the performance of the aforementioned PbMAs in solving the IK problem for a specific continuum robot known as Dual-Cross-Module Sections Cable-Driven Continuum Robot (DCM-S-CDCR), three numerical experiments were conducted. The first experiment focused on a target point, while the second one involved tracking a trajectory. Both experiments were conducted in a free environment. After comparing and analyzing the results obtained from all five PbMAs, a third experiment was conducted to evaluate the performance of the best algorithm in a confined environment for a specific target point. All numerical experiments were performed using MATLAB R2016a software on a computer with the following specifications: Intel(R) Xeon(R) CPU E5-1620 0 @ 3.60 GHz, 16 GB of RAM, and a 64-bit Windows 7 operating system. Additionally, each optimization was conducted considering a search swarm/agents/population size of 20 and a maximum limit of 80 iterations.

4.1. Obtaining the IKs for a target point

This numerical experiment aims to randomly search for solutions to the IK problem in a free environment, meaning there are no additional constraints on the cost function. The target point chosen to position the robot's end-tip is at coordinates $[60, 60, 270]$ (mm), and its initial position is at coordinates $[0, 0, 360]$ (mm) relative to reference frame $(o_0 x_0 y_0 z_0)$. To achieve this purpose, each algorithm is repeated 50 times and their results are recorded and visually presented using box plots in Figure 2. Noteworthy results are highlighted in Table 2 where information such as minimum, maximum, mean values of Execution time

(Ex. time) and Position error (P. error), as well as the number of outlier results are provided.



(a) (b)

Fig. 2. Results of target point tracking in a free environment: (a) Execution time box plot, (b) Position error box plot.

Table 2

The notable results of target point tracking in a free environment.

	Minimum value		Mean value		Maximum value		Number of outliers	
	Ex. time (ms)	P. error (mm)	Ex. time (ms)	P. error (mm)	Ex. time (ms)	P. error (mm)	Ex. time	P. error
ABC	28.47	0.00294	30.01	0.04976	95.41	0.24402	6	1
ACO	63.32	0.00095	66.59	0.09237	139.76	0.18610	4	0
BA	41.58	0.00838	102.85	0.10670	166.01	0.29773	13	7
GA	56.30	0.00020	59.19	0.00923	108.86	0.09073	11	2
PSO	01.45	0.00214	21.33	0.01774	.27.37	0.14830	11	3

Based on the results obtained, it is evident that PSO followed by ABC perform better in terms of execution time compared to other algorithms. Their mean execution time values are less than 22 ms and 31 ms respectively, whereas BA has the highest mean value at around 103 msec. While all algorithms exhibit outlier results, PSO shows a smaller data distribution. However, when considering position error as the metric for evaluation, GA followed by PSO achieve superior results with mean values less than 0.01 mm and 0.018 mm respectively; where BA ranks last with a mean position error value of approximately 0.107 mm. Additionally, GA demonstrates a smaller data distribution while ACO does not produce any outlier results.

4.2. Obtaining the IKs for tracking a trajectory in a free environment

In this numerical experiment, the point-to-point technique was used to track a line-shaped trajectory defined as: $\left[10t, 0, 360-10t\right]$. Randomly generated

IK solutions were used. The obtained results are presented in Figures 3 and 4. From Figure 3 depicting execution time analysis, it is evident that PSO followed by ABC algorithms outperformed other algorithms while BA algorithm showed the worst performance. In terms of position error analysis shown in Figure 4, GA followed by PSO demonstrated superior performance compared to other algorithms whereas BA exhibited poor performance.

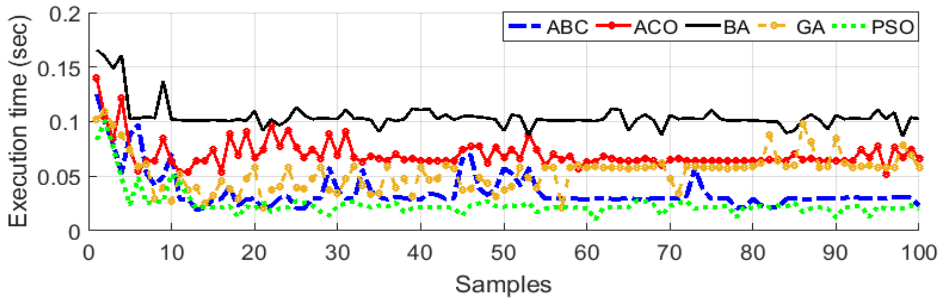


Fig. 3. Results of line-shaped trajectory tracking in a free environment: Execution time graphs.

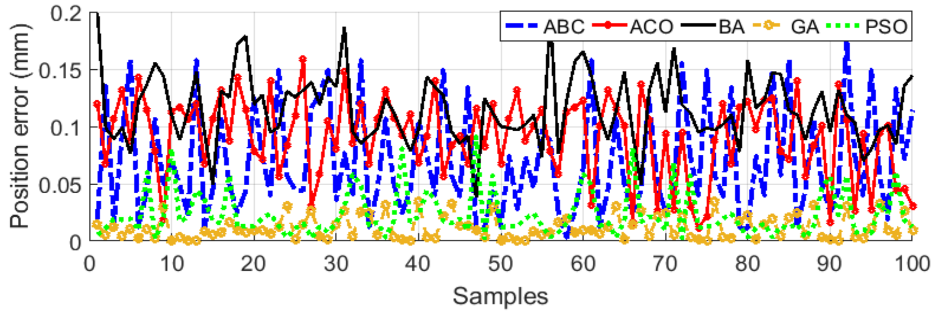


Fig. 4. Results of line-shaped trajectory tracking in a free environment: Position error graphs.

To sum up, Table 3 presents notable results, with the best ones highlighted in bold. According to the Table, PSO achieved the minimum execution time value. However, in terms of position error, GA performed better and was followed closely by PSO.

Table 3

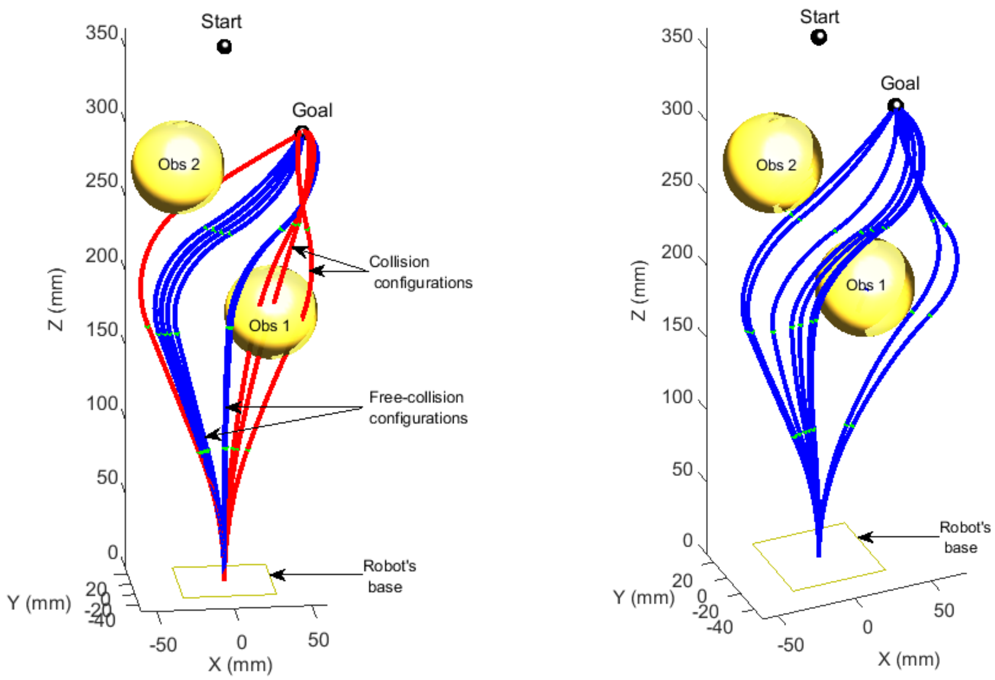
The Notable results of line-shaped trajectory tracking in a free environment.

	ABC	ACO	BA	GA	PSO
Mean execution time (ms)	36.3	70.1	104.7	53.6	**24.7**
Mean position error (mm)	0.0689	0.0908	0.1153	**0.0118**	0.0234

4.3. Obtaining the IKs for a target point in a confined environment

Based on the results from the first two numerical experiments, which showed that the PSO algorithm has a shorter execution time and lower position error compared to other algorithms, a third experiment was conducted using the PSO algorithm. This experiment aimed to evaluate the ability of the algorithm to track target points within a confined environment, specifically in the presence of obstacles. The robot began at its initial position at $\left[0,0,360\right]$ (mm) and had to navigate towards a goal located at $\left[40,100,180\right]$ (mm). Two static obstacles were positioned during this experiment; one obstacle was placed at $\left[30,0,180\right]$ (mm) while another was positioned at $\left[30,0,280\right]$ (mm).

First, a random search is conducted 50 times to find solutions for the IK problem. Figure 5(a) displays different robot configurations that reach the goal, with some being feasible while others collide with static obstacles. In this Figure, collision and non-collision configurations are depicted by red and blue colors respectively. Secondly, in order to avoid collisions with obstacles, a penalty is added to the objective function when the distance between an obstacle and specific points on the robot falls below a certain value. The obtained results in terms of robot's configurations are shown in Figure 5(b). However, in order to select the appropriate solution to the IK problem among the multiple IK solutions, additional constraints must be added to the objective function.



(a) (b)

Fig. 5. Results of target point tracking in the presence of obstacles: (a) Randomly generated IK solutions, (b) No-collision IK solutions.

5. Conclusion

This study compared the performances of five population-based algorithms, namely: Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Bat Algorithm (BA), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). These algorithms were evaluated in terms of their tracking accuracy and execution time for solving the Inverse Kinematics problem of a Dual-Cross-Module Sections Cable-Driven Continuum Robot. The results indicated that PSO outperformed GA in tracking accuracy and ABC in execution time, making it the best overall algorithm. However, this study has some limitations that should be addressed in future research. Firstly, although PSO showed good performance in terms of execution time, its runtimes could be improved further by considering its variants. Secondly, this study only compared basic versions of the algorithms while there are many improved versions available that may yield better results for solving IK problems. Lastly, more algorithms should be included to ensure a comprehensive comparison. Overall findings from this research not only provide valuable insights for solving IK problems but also guide researchers working on similar complex issues.

Acknowledgments

The authors express their gratitude to the anonymous referees for thoroughly reviewing our manuscript. We are grateful for their diligent efforts in providing numerous valuable comments and suggestions, which greatly contributed to enhancing and refining this paper.

R E F E R E N C E S

[1]. *S.M. Youssef, M. Soliman, M.A. Saleh, et al.*, "Underwater soft robotics: a review of bioinspiration in design, actuation, modeling, and control," Micromachines, **vol. 13**, no 1, pp.110, 2022.

[2]. *S. Liu, Z. Yang, Z. Zhu, et al.* "Development of a dexterous continuum manipulator for exploration and inspection in confined spaces," Industrial Robot., **vol. 43**, no 3, pp. 284–295, 2016.

[3]. *Y. Zhang and M. Lu*, "A review of recent advancements in soft and flexible robots for medical applications," The International Journal of Medical Robotics and Computer Assisted Surgery, **vol. 16**, no. 3, p.e2096, 2020.

[4]. *H. Tsukagoshi, A. Kitagawa, M. Segawa*, "Active hose: an artificial Elephant's nose with maneuverability for rescue operation," In Proceedings of IEEE International Conference on Robotics and Automation, Seoul, Korea, 2001, pp. 2454–2459.

[5]. *R. J. Webster and B. A. Jones*, "Design and kinematic modeling of constant curvature continuum robots: A review," International Journal of Robotics Research, **vol. 29**, no. 13, pp. 1661–1683, 2010.

[6]. *A. Amouri,* "Investigation of the constant curvature kinematic assumption of a 2-Dofs cable-driven continuum robot," UPB Scientific Bulletin, Series D: Mechanical Engineering, **vol. 81**, pp. 27–38, 2019.

[7]. *B. A. Jones, I. D. Walker*, "Kinematics for multi-section continuum robots," IEEE Transactions on Robotics, **vol. 22**, no. 1, pp. 43–55, 2006.

[8]. *M. W. Hannan and I. D. Walker*, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," Journal of Robotic Systems, **vol. 20,** pp. 45–63, 2003.

[9]. *S. Djeffal, A. Amouri, and C. Mahfoudi*, "Kinematics modeling and simulation analysis of variable curvature kinematics continuum robots," UPB Scientific Bulletin, Series D: Mechanical Engineering., **vol. 83**, pp. 28–42, 2021.

[10]. *S. Neppalli, M. A. Csencsits, B. A. Jones, et al.,* "Closed-form inverse kinematics for continuum manipulators," Advanced Robotics, **vol. 23**, pp.2077–2091, 2012.

[11]. *R.V. Ram, P. Pathak and S. J. Junco*, "Inverse kinematics of mobile manipulator using bidirectional particle swarm optimization by manipulator decoupling," Mechanism and Machine Theory **vol. 131**, pp. 385–405, 2019.

[12]. *A. Amouri, C. Mahfoudi, A. Zaatri, et al.*, "A metaheuristic approach to solve inverse kinematics of continuum manipulators, " Journal of Systems and Control Engineering, **vol. 231**, no. 5, 2017, pp. 380–394.

[13]. *S. Iqbal, S. Mohammed and Y. Amirat*, "A guaranteed approach for kinematic analysis of continuum robot based catheter, " In Proceeding of the IEEE international conference on robotics and biomimetics, 2009, pp. 1573–1578.

[14]. *A. Amouri, H. Merabti, A. cherfia, at al.,* "Nonlinear model predictive control for trajectory tracking of a class of continuum robots," UPB Scientific Bulletin, Series D: Mechanical Engineering, **vol. 84**, no. 3, pp. 19–32, 2022.

[15]. *A. Merrad, A. Amouri, A. Cherfia, et al*., "A reliable algorithm for obtaining all-inclusive inverse kinematics' solutions and redundancy resolution of continuum robots," Arabian Journal for Science and Engineering; vol. 48, pp. 3351–3366, 2023.

[16]. *S. Djeffal, C. Mahfoudi and A. Amouri*, "Comparison of three meta-heuristic algorithms for solving inverse kinematics problems of variable curvature continuum robots," In Proceeding of the European Conference on Mobile Robots (ECMR), Bonn, Germany, 2021, pp. 1–6.

[17]. *D. Karaboga*, "An idea based on honey bee swarm for numerical optimization," Technical Report - Erciyes University, 2005.

[18]. *M. Dorigo, V. Maniezzo, and A. Colorni*, "Ant system: optimization by a colony of cooperating agents," IEEE Transactions on Systems, Man, and Cybernetics-Part B, **vol. 26**, no. 1, pp. 29–41, 1996.

[19]. *X. S. Yang,* "A New Metaheuristic Bat-Inspired Algorithm," in Nature Inspired Cooperative Strategies for Optimization (NISCO 2010). Studies in Computational Intelligence, **vol. 284**, pp. 65–74, 2010.

[20]. *J. H. Holland*, "Adaptive in natural and artificial systems," Ann Arbor, MI: University of Michigan Press, 1975.

[21]. *R. Eberhart and J. Kennedy*, "Particle swarm optimization," In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 1995, **vol. 4**, pp. 1942–1948.

[22]. *A. Amouri, A. Cherfia, A. Belkhiri, et al*., "Bio-inspired a novel dual-cross-module sections cable-driven continuum robot: design, kinematics modeling and workspace analysis," Journal of the Brazilian Society of Mechanical Sciences and Engineering, **vol. 45**, article: 265, 2023.