Implementation of Crow Search Algorithm for Achieving Optimal Control of a Single-Link Flexible Manipulator

Venkata Satya Durga Manohar Sahu
School of Electrical Engineering, Kalinga Institute of Industrial Technology, Deemed to be University, India
manohar17.me@gmail.com
(corresponding author)

Padarbinda Samal
School of Electrical Engineering, Kalinga Institute of Industrial Technology, Deemed to be University, India
padarbindasamal87@gmail.com

Chinmoy Kumar Panigrahi
School of Electrical Engineering, Kalinga Institute of Industrial Technology, Deemed to be University, India
panigrahichinmoy@gmail.com

Received: 30 September 2022 | Revised: 29 October 2022 | Accepted: 16 November 2022

ABSTRACT

A metaheuristic optimizer based on the cognitive behavior of crows, called the Crow Search Algorithm (CSA), is suggested in this paper for the optimization of the performance of the single link flexible manipulator as a constrained optimal control problem, which comprises objective functions, constraints, and decision variables. The CSA outperformed other algorithms, such as Differential Evolution, Grey Wolf Optimization, and Particle Swarm Optimization. Parameter setting is another highlight of the current paper. In conclusion, CSA yields more promising results than other approaches.

Keywords-optimal control system; crow search algorithm; single link flexible manipulator

I. INTRODUCTION

Optimization has played a critical part in a wide range of problems during the last few decades. Engineers have relied on conventional search methods for decades to solve design problems. Despite the fact that these approaches give promising outcomes, they may fail in more complicated design challenges. Optimization problems are often complicated due to their complex objective functions, a substantial amount of decision variables, and many constraints. So, traditional optimization strategies often have limited effectiveness. The decision parameters may be too many and their influence on the objective function can be complex in real-world design challenges [1]. This means that the objective function might have multiple local optimal solutions while the design is only interested in the optimal global solution. These challenges can't be solved by conventional methods that only discover local optima in a specific location. We need efficient optimization methods in these situations. Consequently, authors in [2] developed a new field of research called Swarm Intelligence (SI) in the late 1980s, which is still being explored today. Today's trend is to use natural metaheuristic algorithms [3, 4] to solve challenging issues, and metaheuristics have been proved to be surprisingly efficient. Metaheuristic algorithms have shown promising performance in handling the most severely nonlinear and multimodal real-world optimization problems. Particular randomization and local search are used by all metaheuristic algorithms [5]. These methods can find good answers to challenging optimization problems, but optimum solutions cannot be found. However, it is hoped that these methods will work in most cases. It is possible to use metaheuristic algorithms for global optimization.

Single-link flexible manipulators are often formulated as optimization problems [6-9] in order to optimize the performance index while meeting various constraints. As it has the potential advantage of lower cost, larger work volume, higher operational speed, greater payload-to-manipulator-

weight ratio, smaller actuators, lower energy consumption, better transportability, and safer operation due to reduced inertia over traditional heavy and bulky robots. The main drawback of these manipulators is the vibration problem caused by their poor rigidity [2]. The Lagrangian-assumed modes technique [10] is used to create a dynamic model of a single-link flexible manipulator. Controlling was conducted by various law [11] and intelligent techniques [12, 13]. An overall goal is to maximize system characteristics [14] for accessibility, which is represented by a vector. Particle Swarm Optimization (PSO) algorithm was used to model a flexible manipulator system in [15]. An optimization technique begins by creating an objective function capable of generating the problem objectives without removing any constraints in order to identify the best feasible solution. According to the Glover Convention, all current procedures influenced by nature are considered metaheuristics [16]. Differential Evolution (DE) algorithm [9, 17], based on natural selection and PSO which is based on bird flock and fish schooling social behavior [18], are some of the best-known metaheuristic algorithms. The Crow Search Algorithm (CSA) [19] was presented in 2016. The social intelligence and food-collecting mechanism of a crow group are included in the design. Crows are a bird species that has expanded far across the globe and are considered very intelligent. They can memorize faces, utilize tools, communicate, and conceal food in various complicated ways.

The main objectives of the work at hand are:

- Parameter setting is done with the manipulator model using the proposed approach.
- Performance comparison of the proposed CSA with the existing DE, PSO, and Grey Wolf Optimizer (GWO).
- The CSA's ability to identify an appropriate alternative strategy for addressing complex engineering optimization problems is investigated.

II. PROBLEM FORMULATION

An optimization problem [20] is the objective of identifying the optimal solution among all possible options. Optimization problems can be classified as either unconstrained or constrained based on whether the variables are discrete or continuous. This section uses the SLFM to formulate the optimal control problem [9] described in detail.

A. Modeling of the Dynamic System

Regarding the modeling of the dynamic system, assume the following set of state equations:

$$\dot{z} = g(z(t), v(t), t) \quad (1)$$

where z is a state vector, v is a manipulator control torque input vector, t is continuous-time, and g is a vector function.

B. Performance Index

An objective function or its inverse (referred to as a profit function, fitness function, utility function, or a reward function in some disciplines) is an objective function that should be maximized. It is referred to as:

$$J(v) = \varphi(z(t_f), t_f) + \int_{t_0}^{t_f} L(z(t), v(t), t)dt \quad (2)$$

L is a real-valued state function and the input vector that may or may not be explicitly time-dependent depending on the situation. The cost function is a term that is often used to describe this function. The function is a real-valued function of the vector of the final state and its time vector, with the final state vector serving as the input. When it comes to quality, a performance index should be constructed, so that it is constrained from below, and the higher the index, the poorer the controller's performance. For example, ensuring that the first term in (2) and the integrand in the second term are positive for all values of $z(t)$, $v(t)$ and t, may be achieved by requiring. The first term in (2) denotes a constraint on the state vector's final or terminal value. The value of the performance index decreases as the end state vector approaches the desired value. After specifying the cost function and constraint, the objective is to find the optimal controller, that is, the value of the control signal $v(t)$ for the time period $t_0 \le t \le t_f$ that yields the lowest values J under the assumption that the state vector obeys the state equation (1).

C. Constraints

When solving an optimization problem, a constraint is a requirement that the solution must meet. There are numerous different types of constraints, the most common of which are equality constraints, inequality constraints, and integer constraints.

- Inequality constraints exist when a function has a border or limitations, e.g. less than equal to, less than, greater than, greater than equal to, and is expressed as:

$$G(z(t)) \le 0 \quad (3)$$

where $G(.)$ denotes the limitations imposed by inequality on the vector mapping of the state variables.

- Equality restrictions exist when a function is just the equal to the requirements, i.e. it perfectly matches the value of the resource, and they are expressed as:

$$\Psi(z(t)) = 0 \quad (4)$$

where $\Psi(.)$ represents equality constraints of the state variables. It's possible that this variable is a state or a control variable. It may alter based on the problem's design and demand. This variable may have limits on rare occasions. It's referred to as:

$$v_{\min} \le v(t) \le v_{\max} \quad (5)$$

where $v(.)$ is a control variable with lower and upper limits. Correspondingly, state variables may be specified based on the design and demand, but it is more complicated to identify boundary values, although a set of minimum and maximum values are used for estimations. Our ultimate objective is, therefore, to achieve the best value possible.

D. Optimal Control Problem (OCP) Formulation

Consider the following description of an OCP [21] to be designed:

$$\min_{v(k)} J = \varphi(z(t_f), t_f) + \int_{t_0}^{t_f} L(z(t), v(t), t)dt$$

subjected to (6)

$$G(z(t)) \le 0$$
$$v_{\min} \le v(t) \le v_{\max}$$

where J is the performance index, with state constraints $G(.)$ and control constraints $v(t)$ .The dynamic model of the SLFM assessed the performance index as well as inequality and equality constraints. For example, in [7, 8, 21], we determine the best solution to the given issue using the CSA and the constrained performance index.

E. Example

1) Single-Link Flexible Manipulator Model

Assume a flexible manipulator model [6, 22, 23], as shown in Figure 1.

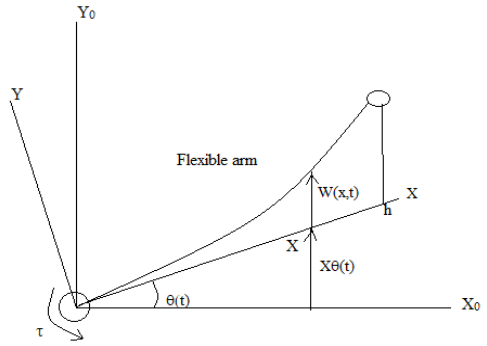

Fig. 1. Flexible link manipulator.

$$\left(J_h + J_l\right)\ddot{\theta} + J_l\ddot{\alpha} - mgh\sin(\theta + \alpha) = \tau$$
$$J_1\ddot{\theta} + J_l\ddot{\alpha} + K_s\alpha - mgh\sin(\theta + \alpha) = 0 \quad (7)$$

Let us assume the above equation by considering:

$$z_1 = \theta,\ z_2 = \alpha,\ z_3 = \dot{\theta},\ z_4 = \dot{\alpha} \quad (8)$$

and the above equation can be rewritten as follows:

$$\dot{z}_1 = z_3$$
$$\dot{z}_2 = z_4$$
$$\dot{z}_3 = \frac{k_s}{J_h}z_2 - \frac{k_m^2k_g^2}{R_mJ_h}z_3 + \frac{k_mk_g}{R_mJ_h}v \quad (9)$$
$$\dot{z}_4 = -\frac{k_s}{J_h}z_2 + \frac{k_m^2k_g^2}{R_mJ_h}z_3 - \frac{k_mk_g}{R_mJ_h}v \cdots$$
$$\cdots + \frac{mgh}{J_1}\sin(z_1 + z_2) - \frac{k_s}{J_1}z_2$$

where m is the arm end mass, J_1 is the inertia of the arm, g is the gravity acceleration, B_m is the rotational friction, J_h is the moment of inertia of the motor, r is the radius, z_1 is the arm end position (i.e. the angular position of the motor θ), z_2 is the arm axis velocity (i.e. the angular displacement of the flexible joint α), z_3 is the arm position, z_4 is the arm end velocity. The parameter values are taken from [6].

2) Control Requirements

Initially, consider the manipulator was at $z=[0.09 \quad 0.09 \quad 0.09 \quad 0.09]^T$ and we wish to move it into the desired position, e.g. $z=[0 \quad 0.4 \quad 0 \quad 0]^T$ at t =0.78s with the following constraints on the velocity and control signal:

$$z_2(t) \le 0.4, \quad t \in [0, 0.78]$$
$$-10 \le v(t) \le 10, \quad t \in [0, 0.78] \quad (10)$$

3) Formulation as a Constrained Optimal Control Problem

The OCP is defined and formulated as:

$$J = \int_{t_0}^{t_f} z_1^2 + z_2^2 + v^2 dt \quad (11)$$

Subject to:

$$\begin{cases} \dot{z}_1 = z_3 \\ \dot{z}_2 = z_4 \\ \dot{z}_3 = \frac{k_s}{J_h}z_2 - \frac{k_m^2k_g^2}{R_mJ_h}z_3 + \frac{k_mk_g}{R_mJ_h}v \\ \dot{z}_4 = -\frac{k_s}{J_h}z_2 + \frac{k_m^2k_g^2}{R_mJ_h}z_3 - \frac{k_mk_g}{R_mJ_h}v \\ \quad + \frac{mgh}{J_1}\sin(z_1 + z_2) - \frac{k_s}{J_1}z_2 \\ z_2(t) - 0.4 \le 0 \\ -10 \le v(t) \le 10 \end{cases} \quad (12)$$

III. SOLUTION METHODOLOGY

CSA [19, 22], is based on the foraging behaviors of crows. Crows have various distinct characteristics that they display while hunting for food. When compared to other bird species, they have a large memory for remembering the locations of potential food stocks. Crows steal their owner's food supply after following other birds to food hiding locations. The crow moves the food source from its current site to one that is more barren as it scavenges the available food in order to protect it from other crows. Each point in the search space indicates a solution in the context of CROW. Crows have been observed to watch other birds, noting where they keep their food and then stealing it once the owner has left. If it has previously been a victim of theft, a crow will take extra precautions, such as changing hiding places, to avoid being a victim again. The previous behavior of an individual may be used to predict the future behavior of a thief and find the safest strategy to safeguard their stockpiles from theft. Based on the above-

mentioned intelligent behaviors, the population-based metaheuristic algorithm, CSA, is implemented in this study. To attain the objectives, we must follow specific principles:

- Crows live in groups of individuals (flocks).
- Crows memorize their hiding places.
- Crows congregate in order to steal.
- Crows use measures to protect their caches against stealing.

Each crow is required to make a move based on 1 of the 2 fundamental CSA rules: either protect its own hiding place or find the locations of other members. Imagine, for instance, that a crow decides on iteration *iter* to steal food from a hiding spot. Depending on the situation, either (1) the individual crow is not aware that it is being watched, or (2) the individual crow recognizes the presence of the plunderer and acts dishonestly. According to the first scenario, the crow may discover and pillage the hiding location. The crow can be relocated as follows:

$$z^{i,iter+1} = z^{i,iter} + r_i \times fl^{i,iter} \times (m^{i,iter} - z^{i,iter}) \quad (13)$$

where r_i is a random number with an even distribution between 0 and 1 and $fl^{(i,iter)}$ signifies the flight length of the crow i at iteration *iter*.

It's worth noting that $fl^{(i,iter)}$ is a parameter of the algorithm and can affect the method's search capabilities. Assume that smaller fl values result in a local search near $z^{(i,iter)}$, whereas more significant values of fl expand the search space. In terms of optimization, smaller fl values contribute to the intensification of the results, whereas more significant fl values contribute to the diversification of the results. Both high-intensification and high-diversification are desirable properties of an efficient optimization technique.

Alternatively, the j^{th} crow might think a fellow flock member is chasing it (say the i^{th} crow). The j^{th} crow flew erroneously over a non-hideout area to guard its food source. If the i^{th} crow is arbitrarily placed in the d-dimensional option space, the CSA can repeat the operation. Therefore, for the 2 conditions mentioned above, the crows' tailing motion may be explained as follows:

$$z^{(i,iter+1)} = \begin{cases} z^{i,iter+1} = z^{i,iter} + \cdots & \\ \cdots r_i \times fl^{i,iter} \times (m^{i,iter} - z^{i,iter}) & \text{rand}() \geq AP \\ \text{random crow position} & \text{otherwise} \end{cases} \quad (14)$$

In metaheuristic algorithms, diversity and intensity should be well-balanced. When it comes to CSA, the system is essentially deepened and diversified by the Awareness Probability (AP) component. When a suitable solution already exists in place, CSA is more likely to look locally by lowering the AP. Low AP levels can be used to boost intensity. However, when knowledge grows, there may be a chance to find a workable solution, and CSA often expands the search area globally (randomization). The use of high AP values enables the achievement of greater variation

A. CSA Implementation for Optimization

The step-wise procedure for the implementation of CSA is given in this section.

Step 1: Initialize parameters AP, flight length fl, randomly initialize the control variable.

Step 2: Initialize the crows' position and memory.

Step 3: Estimate fitness (objective) function.

Step 4: Generate new position using (14).

Step 5: Inspect the possibility of new positions.

Step 6: Update fitness of new positions

Step 7: Update the memory

Step 8: Verify that the end criteria has been met

Repeat steps 4–7 as many times as necessary until the maximum number of iterations is reached. After that, an optimal memory position in terms of the objective function's value is provided to solve an optimization problem.

B. Crow search Algorithm Pseudocode

```
Initialize the parameters
Randomly initialize the position of N
crows flows in the search space
Assess position, performance, and fitness
State the crow memory
while Termination conditions not satisfied
for N of the flock's crows
Pick a crow at random from the group
State awareness probability
if rand()>AP ,
Update the position using (13)
else
Update the position randomly
end if
end for
Assess the new performance and fitness
with the new crow's position
Update the crow's memory
end while
```

IV. SIMULATION RESULTS AND DISCUSSION

Optimal control problems can't be solved by a single search algorithm. Another way to put it is that one algorithm may answer some problems better and others worse than other algorithms. For a fair evaluation of the proposed CSA, problems are discussed and solved. CSA has been run successfully on a laptop with an i5 processor and 16GB of RAM in MATLAB environment. The flowchart of CSA implementation can be seen in [24].

A. CSA Paramater Setting for Implementation Optimization

The CSA's parameter settings have been optimized in a number of ways for 50 runs and 450 iterations. According to

Tables I-III, the best performance index obtained by CSA is 0.001985 and 0.001982 for the single link flexible manipulator. Fair AP values produce better results, nevertheless, bigger values of AP are suggested to avoid becoming trapped in local optima. If AP remains constant (AP = 0.1) and the value of *fl* is increased from 0.3 to 4, it is projected that the overall performance of CSA will be improved on average.

In Table I, the impact of the AP is shown to assess the CSA's performance. AP and *fl* have been set to 0.1 and 2, respectively, in CSA. CSA is also analyzed to see how different parameter settings affect its performance. AP and *fl* values influence the results for the SLFM, as shown in Tables I and II. Table I shows that AP = 0.25 results in CSA poor performance. Figure 2 shows its impact. In Table II, it can be seen that when the value of AP is adjusted to 0.1, and the value of *fl* is increased from 0.3 to 4, the overall performance of the CSA improves. As a result, fine tuning CSA, like other optimization methods, is a process best accomplished by trial and error. The impact of the flight length can be observed in Figure 3.

TABLE I. IMPACT OF VARYING AP VALUES FOR fl=2

Performance index (*J*)	**AP = 0.1**	**AP = 0.25**	**AP = 0.5**	**AP = 0.75**
Best value	0.00200	0.002645	0.001985	0.001998
Worst value	5.1851	18.591	17.1683	8.839
Mean	0.0967	0.3692	0.1079	0.038
Standard deviation	0.4213	1.4912	0.8261	0.5583

TABLE II. THE EFFECT OF *fl* ON THE PERFORMANCE OF CSA

Index	***fl* = 0.3 AP = 0.1**	***fl* = 0.5 AP = 0.1**	***fl* = 1 AP = 0.1**	***fl* = 1.5 AP = 0.1**	***fl* = 1.75 AP = 0.1**	***fl* = 2 AP = 0.1**	***fl* = 3 AP = 0.1**	***fl* = 4 AP = 0.1**
Best value	0.0073622	0.007324	0.016931	0.00496991	0.00883244	0.00200153	0.00256202	0.00250375
Worst value	10.1884	11.9041	5.5145	15.0302	8.1574	5.1851	21.1613	10.3002
Mean	0.9474	0.2573	0.3259	0.0733	0.1488	0.0967	0.129	0.1551
Standard deviation	1.6151	1.1226	0.7193	0.6806	0.7486	0.4213	1.3639	0.7575

TABLE III. THE EFFECT OF ITERATION NUMBER ON THE PERFORMANCE OF CSA

Index	**500**	**1000**	**1500**	**2000**	**2500**	**3000**	**3500**	**4000**	**5000**
Best Value	0.002002	0.001982	0.002217	0.001985	0.00198	0.001989	0.002019	0.001994	0.001988
Worst Value	5.1851	12.3677	16.9836	22.2053	9.3461	16.9836	8.2768	8.2161	16.9836
Mean	0.0967	0.1526	0.0845	0.08	0.0196	0.0433	0.0181	0.0316	0.0268
Std	0.4213	1.0856	0.8877	0.777	0.2444	0.636	0.3138	0.2933	0.493

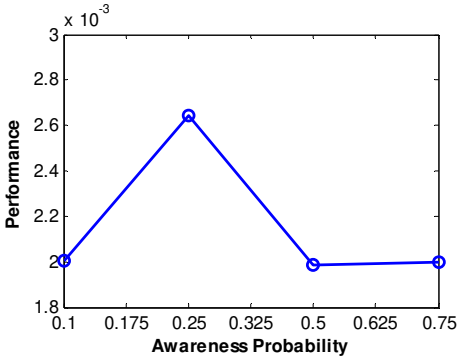

Fig. 2. The impact of varying AP values on CSA performance.

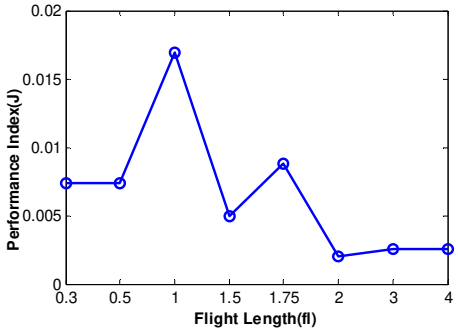

Fig. 3. Effect of *fl* on the performance of the CSA.

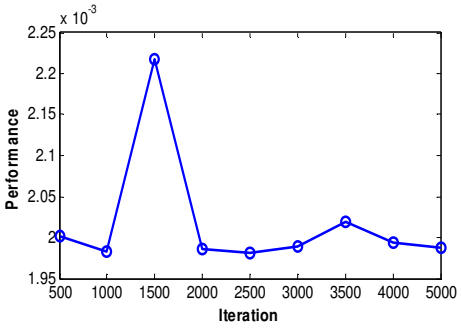

Fig. 4. Effect of iteration number on the performance of CSA.

From Table III we can note that by increasing the iteration number more than 2000 and by keeping the AP as 0.1 and *fl* as 2, the performance remains almost the same, showing poor performance which can be clearly seen in Figure 4.

B. Comparison Analysis of DE, PSO, GWO, and CSA

Optimal control problem can be solved by using the CSA parameters as shown in Table IV. The statistical findings produced using CSA are compared in Table V. In comparison to the other methods on the SLFM, the results suggest that CSA delivers promising results. When it comes to the best index, CSA surpasses DE, PSO, and GWO. In this example, CSA's results are roughly 45% more accurate than the other methods used in terms of performance.

TABLE IV. UTLIZIED CSA PARAMETERS

Parameter	Value
Flight Length	2
Awareness probability	0.1
Number of population	50
Number of iteration	450

TABLE V. SLFM RESULT COMPARISON ACHIEVED BY CSA AND OTHER ALGORITHMS

Algorithm	Worst	Mean	Best	Std.	Convergence iteration
DE	1.3561	0.039259	0.0041536	0.15383	435
PSO	1.8147	0.040232	0.003053	0.19363	363
CSA	5.1851	0.0967	0.002109	0.4213	261
GWO	0.0709	0.0515	0.042707	0.0073	448

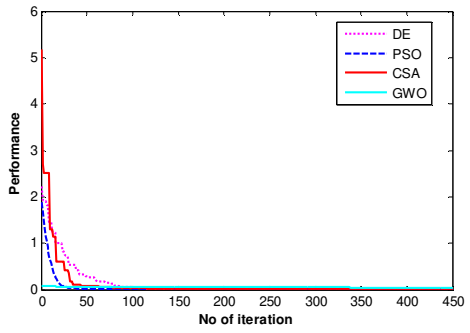

Fig. 5. Algorithm comparison graph.

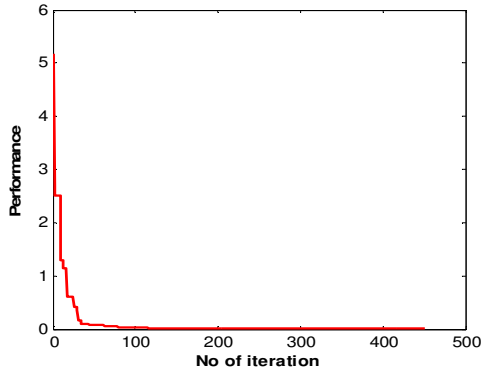

Fig. 6. Convergence graph of CSA of SLFM.

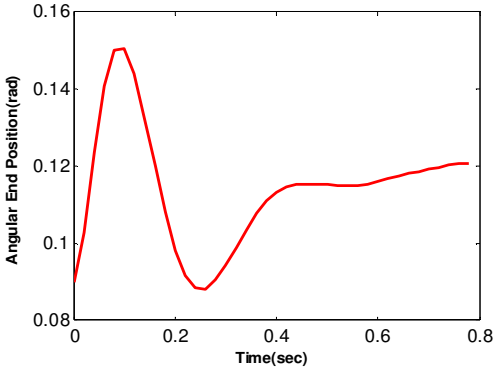

Fig. 7. Arm axis position (z_1) versus time (s) of CSA.

CSA shows encouraging performance in terms of worst and mean indices. The std indicates the CSA's substantial flexibility. The CSA's performance index value is minimum. The convergence rate of the CSA for identifying the optimum SLFM solution is shown in Figures 5 and 6. The CSA algorithm exhibits an excellent convergence rate.

Figure 7 explains the arm end position of the SLFM which became stable 0.12 at nearly 0.7s and minimum position 0.088 at nearly 0.26s. Figure 8 shows the comparison graph between the CSA, DE, PSO, and BWO and it is clearly observed that there is more disturbance in GWO, which is unstable.

TABLE VI. COMPARISON OF DE, PSO, GWO, AND CSA FOR ARM END POSITION

Value	CSA	DE	GWO	PSO
Maximum	0.1506	0.1238	1.4439	0.1239
Minimum	0.0882	5.5226e-4	-0.2679	4.9009e-4
Mean	0.1143	0.0293	0.5935	0.0290

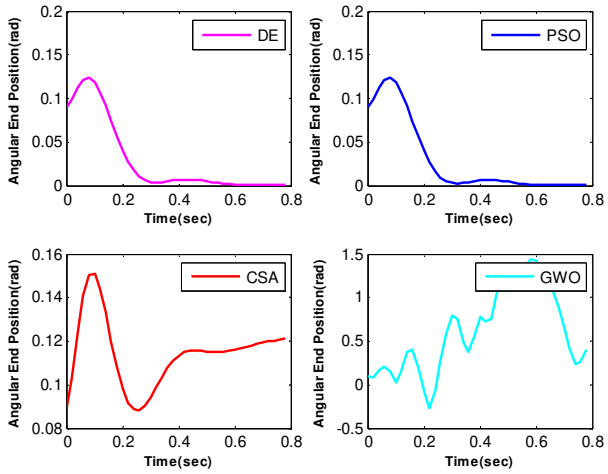

Fig. 8. Comparison graph of arm axis position for DE, PSO, GWO, and CSA.

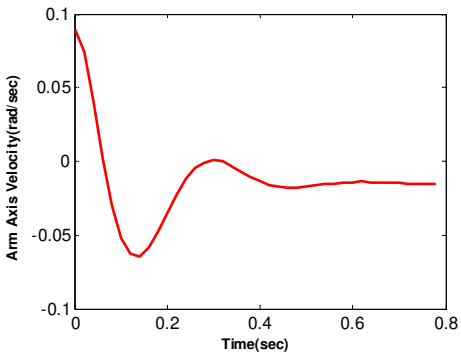

Fig. 9. Arm axis position (z_2) versus time (s) of CSA.

Table VI shows the minimum, maximum, and mean positions attained by the algorithm comparison. DE, PSO, and CSA gave almost the same results of attained angular position, which can be observed clearly in Figure 8.

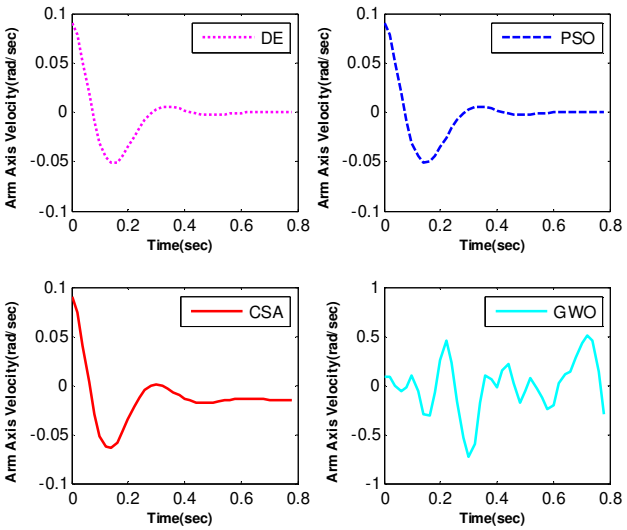

Fig. 10. Comparison graph of arm axis velocity for DE, PSO, GWO, and CSA.

Figure 9 explains the arm axis velocity of the SLFM. It can be seen that less velocity was required to move the SLFM. Figure 10 shows the comparison graph between the CSA, DE, PSO, and GWO. Table VIII shows the minimum, maximum, and mean velocity attained by the algorithm comparison.

TABLE VII. COMPARISON OF DE, PSO, GWO, AND CSA FOR ARM AXIS VELOCITY

Value	CSA	DE	GWO	PSO
Maximum	0.0900	0.0900	0.5039	0.0900
Minimum	-0.0644	-0.0515	-0.7292	-0.0510
Mean	-0.0127	-0.0018	-0.0070	-0.0017

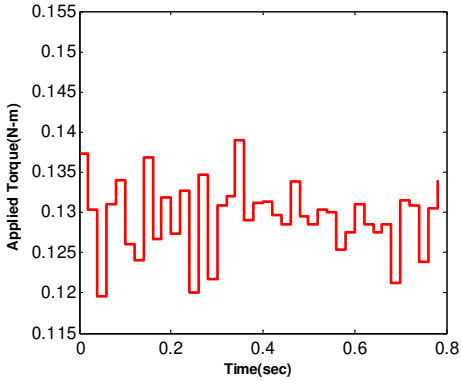

Fig. 11. Applied torque versus time curve.

TABLE VIII. COMPARSION OF DE, PSO, GWO, AND CSA FOR THE APPLIED TORQUE

Value	CSA	DE	GWO	PSO
Maximum	0.1389	0.0018	10	0.0528
Minimum	0.1196	-0.1874	-9.9246	-0.1862
Mean	0.1295	-0.0398	0.0805	-0.0385

Table VIII compared the minimum, maximum, and mean torque applied to SLFM by the considered algorithms. Figure 11 gives an idea of how much torque is applied to SLFM and we can see that for a small torque driven by SLFM, it attains its position. Figure 12 shows the comparison graph between the CSA, DE, PSO, and GWO, in the presence of negative torque which leads the manipulator to brake on in order to move to the desired position. At the same time, CSA has a positive torque which leads the manipulator to move to the desired position during the simulated period of time.

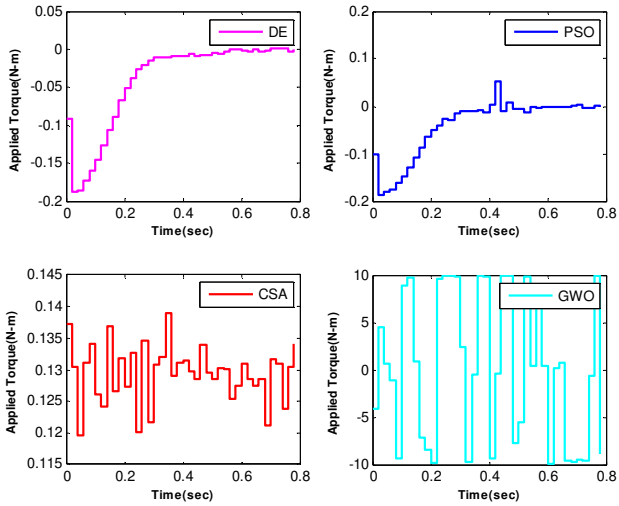

Fig. 12. Comparison of DE, PSO, GWO, and CSA for the applied torque and time.

V. CONCLUSION

In this paper, Crow Search Algorithm (CSA), is proposed to solve OCP, i.e. SLFM. The two regulating factors in CSA are flight duration and awareness probability. CSA is simpler to utilize and has fewer factors to adjust than DE, PSO, and GWO. The simulation results show that the CSA performance is promising since it provided results that were competitive with those of the other methods. Despite the reputation of the PSO as a quick approach among population-based algorithms, CSA's performance surpasses PSO along with the other algorithms.

REFERENCES

[1] M. I. Ullah, S. A. Ajwad, M. Irfan, and J. Iqbal, "Non-linear Control Law for Articulated Serial Manipulators: Simulation Augmented with Hardware Implementation," *Elektronika ir Elektrotechnika*, vol. 22, no. 1, pp. 3–7, Feb. 2016, https://doi.org/10.5755/j01.eee.22.1.14094.

[2] G. Beni and J. Wang, "Swarm Intelligence in Cellular Robotic Systems," in *Robots and Biological Systems: Towards a New Bionics?*, Berlin, Heidelberg, Germany, 1993, pp. 703–712, https://doi.org/10.1007/978-3-642-58069-7_38.

[3] X.-S. Yang, "Metaheuristic Optimization," *Scholarpedia*, vol. 6, no. 8, Aug. 2011, Art. no. 11472, https://doi.org/10.4249/scholarpedia.11472.

[4] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: a comprehensive survey," *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2191–2233, Dec. 2019, https://doi.org/10.1007/s10462-017-9605-z.

[5] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, Jun. 2003, https://doi.org/10.1145/937503.937505.

[6] N. Razmjooy, M. Ramezani, and A. Namadchian, "A New LQR Optimal Control for a Single-Link Flexible Joint Robot Manipulator Based on Grey Wolf Optimizer," *Majlesi Journal of Electrical Engineering*, vol. 10, no. 3, Oct. 2016, Art. no. 53.

[7] B. Farzanegan, S. Dehghan Banadaki, and M. B. Menhaj, "Direct artificial neural network control of single link flexible joint," in *2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA)*, Qazvin, Iran, Jan. 2016, pp. 131–135, https://doi.org/10.1109/ICCIAutom.2016.7483149.

[8] K. Groves and A. Serrani, "Modeling and Nonlinear Control of a Single-link Flexible Joint Manipulator," 2004.

[9] V. S. Durga Manohar Sahu, P. Samal, and C. K. Panigrahi, "Application of Differential Evolution Algorithm to Optimal Control Problem with State Variable Constraints," in *2020 IEEE 17th India Council International Conference (INDICON)*, New Delhi, India, Sep. 2020, https://doi.org/10.1109/INDICON49873.2020.9342273.

[10] N. Mishra, S. P. Singh, and B. C. Nakra, "Dynamic analysis of a single link flexible manipulator using Lagrangian-assumed modes approach," in *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, Pune, India, Feb. 2015, pp. 1144–1149, https://doi.org/10.1109/IIC.2015.7150920.

[11] J. Iqbal, "Modern Control Laws for an Articulated Robotic Arm: Modeling and Simulation," *Engineering, Technology & Applied Science Research*, vol. 9, no. 2, pp. 4057–4061, Apr. 2019, https://doi.org/10.48084/etasr.2598.

[12] H. Medjoubi, A. Yassine, and H. Abdelouahab, "Design and Study of an Adaptive Fuzzy Logic-Based Controller for Wheeled Mobile Robots Implemented in the Leader-Follower Formation Approach," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 6935–6942, Apr. 2021, https://doi.org/10.48084/etasr.3950.

[13] M. Fouzia, N. Khenfer, and N. E. Boukezzoula, "Robust Adaptive Tracking Control of Manipulator Arms with Fuzzy Neural Networks," *Engineering, Technology & Applied Science Research*, vol. 10, no. 4, pp. 6131–6141, Aug. 2020, https://doi.org/10.48084/etasr.3648.

[14] K. L. Teo, "A unified computational approach to optimal control problems," in *A unified computational approach to optimal control problems*, De Gruyter, 2011, pp. 2763–2774, https://doi.org/10.1515/9783110883237.2763.

[15] M. S. Alam and M. O. Tokhi, "Dynamic Modelling of a Single-Link Flexible Manipulator System: A Particle Swarm Optimisation Approach," *Journal of Low Frequency Noise, Vibration and Active Control*, vol. 26, no. 1, pp. 57–72, Mar. 2007, https://doi.org/10.1260/026309207781487466.

[16] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & Operations Research*, vol. 13, no. 5, pp. 533–549, Jan. 1986, https://doi.org/10.1016/0305-0548(86)90048-1.

[17] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec. 1997, https://doi.org/10.1023/A:1008202821328.

[18] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, Aug. 1995, vol. 4, pp. 1942–1948, https://doi.org/10.1109/ICNN.1995.488968.

[19] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Computers & Structures*, vol. 169, pp. 1–12, Jun. 2016, https://doi.org/10.1016/j.compstruc.2016.03.001.

[20] G. Lindfield and J. Penny, *Introduction to Nature-Inspired Optimization*. Elsevier Academic Press, 2017, https://doi.org/10.1016/C2015-0-00099-5.

[21] B. Subudhi and A. S. Morris, "Soft computing methods applied to the control of a flexible robot manipulator," *Applied Soft Computing*, vol. 9, no. 1, pp. 149–158, Jan. 2009, https://doi.org/10.1016/j.asoc.2008.02.004.

[22] V. S. D. M. Sahu, P. Samal, and C. K. Panigrahi, "Application of Crow Search Algorithm to Solve Discrete Optimal Control Problem with Constraints for a Flexible manipulator," in *2022 3rd International Conference for Emerging Technology (INCET)*, Belgaum, India, Feb. 2022, https://doi.org/10.1109/INCET54531.2022.9825392.

[23] V. Satya Durga Manohar Sahu, P. Samal, and C. Kumar Panigrahi, "Modelling, and control techniques of robotic manipulators: A review," *Materials Today: Proceedings*, vol. 56, pp. 2758–2766, Jan. 2022, https://doi.org/10.1016/j.matpr.2021.10.009.

[24] B. Zolghadr-Asli, O. Bozorg-Haddad, and X. Chu, "Crow Search Algorithm (CSA)," in *Advanced Optimization by Nature-Inspired Algorithms*, O. Bozorg-Haddad, Ed. Singapore: Springer, 2018, pp. 143–149, https://doi.org/10.1007/978-981-10-5221-7_14.