Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
Hybrid Computational Intelligence Algorithm for Autonomous Handling of COVID-19 Pandemic Emergency in Smart Cities

Mohamed Abdel-Basset\textsuperscript{a}, Khalid A. Eldrandaly\textsuperscript{a}, Laila A. Shawky\textsuperscript{a,}\textsuperscript{*}, Mohamed Elhoseny\textsuperscript{b}, Nabil M. AbdelAziz\textsuperscript{a}

\textsuperscript{a} Faculty of Computers and Informatics, Zagazig University, Sharqiya, Egypt
\textsuperscript{b} Faculty of Computers and Information, Mansoura University, Mansoura, Egypt

ARTICLE INFO

Keywords: computational intelligence algorithms COVID-19 medical robot pandemic emergency services New Galala city optimization

ABSTRACT

New cities exploit the smartness of the IoT-based architecture to run their vital and organizational processes. The smart response of pandemic emergency response services needs optimizing methodologies of caring and limit infection without direct connection with patients. In this paper, a hybrid Computational Intelligence (CI) algorithm called Moth-Flame Optimization and Marine Predators Algorithms (MOMPA) is proposed for planning the COVID-19 pandemic medical robot’s path without collisions. MOMPA is validated on several benchmarks and compared with many CI algorithms. The results of the Friedman Ranked Mean test indicate the proposed algorithm can find the shortest collision-free path in almost all test cases. In addition, the proposed algorithm reaches an almost %100 success ratio for solving all test cases without constraint violation of the regarded problem. After the validation experiment, the proposed algorithm is applied to smart medical emergency handling in Egypt’s New Galala mountainous city. Both experimental and statistical results ensure the prosperity of the proposed algorithm. Also, it ensures that MOMPA can efficiently find the shortest path to the emergency location without any collisions.

1. Introduction

With the upgrowth of many lethal diseases such as COVID-19 and the Delta variant, public health systems all over the world need to deal with health emergencies and provide appropriate smart solutions to reduce the infection rate. In other words, traditional public health systems lacked how to deal with health emergencies and provided inappropriate solutions. Their proposed solutions had limited effectiveness because they lacked the pre-requisite infrastructure. In particular, traditional public safety officials relied on dispatchers and emergency call centers to alert them to active emergencies. The dispatchers act as a mediator between on-site witnesses and emergency responders. However, the information they transmit may not always be complete, and first responders may lack knowledge about the task before arrival (Soyata et al., 2019).

Therefore, cities need a way to give public safety officials information gathering tools on their communities to better assess pandemic emergencies. Besides, they must put into consideration the limit infection, reduction in response time, and deploy appropriate contrivance to respond (Qureshi et al., 2020). In order to tackle the traditional emergency service disadvantages, emerging smart cities provision their pandemic medical services with the Internet of Things (IoT) connected appliances that are based on Artificial Intelligence (AI) (Rouhanizadeh and Kermanshahi, 2020, Westraadt and Calitz, 2020). As shown in Fig. 1, smart cities around the world capture critical data for emergency first responders by installing sensors and cameras on traffic lights, street furniture, and utility poles to construct an extensive surveillance network. Once the emergency surveillance network is notified of a situation, they connect with the official agencies for sending dispatch (or mobile robots) teams to the specific emergency site (Francini et al., 2020, Dong et al., 2020, Kumar et al., 2020). Such a smart pandemic emergency response system can offer faster responses with better quality and accuracy, and with less cost.

Many efforts have been made to strengthen medical services through pandemic emergency situations, especially after the accretion of COVID-19 and the Delta variant. For instance, Egypt invented a robot called

* Corresponding author
E-mail addresses: mohamedbasset@zu.edu.eg (M. Abdel-Basset), khaled.eldrandaly@zu.edu.eg (K.A. Eldrandaly), englaila2013@gmail.com (L.A. Shawky), melhoseny@ieee.org (M. Elhoseny), N_Moustafa@zu.edu.eg (N.M. AbdelAziz).

https://doi.org/10.1016/j.scs.2021.103430
Received 22 January 2021; Received in revised form 1 October 2021; Accepted 1 October 2021
Available online 7 October 2021
2210-6767/© 2021 Elsevier Ltd. All rights reserved.
“Shams” which specialized in caring for coronavirus patients and analyzing the samples taken from them, as shown in Fig. 2(a). This robot aims to protect the medical team from infection, as it allows effective communication between pandemic patients and the medical team in a safe manner since the robot can transport medicines from the pharmacy to the places of isolation of patients, and similarly delivers samples to
the laboratory. Besides, this robot can take advantage of AI, by self-learning, such as learning to walk and analyzing the information it receives, such as discovering new symptoms, which benefits the scientific research process in the medical field. Also, a team of doctors and engineers at the University of Southern Denmark developed a fully automated swab robot, as shown in Fig. 2 (b), it depends on both computer vision and machine learning to identify the perfect target spot inside the patient’s throat, then a robotic arm reaches in to collect the sample in a swiftness that humans can’t match.

Medical Robots are also being used in a surveillance role to ensure that social distancing or lockdown guidelines are being followed. For example, a company for global leading AI, enterprise service robots, and humanoid robotics invented a robot called ATRIS enforcing smart navigation of abnormalities, threat detection, and intelligent tracking of patients (see Fig. 2(c)). Furthermore, researchers from the University of California showed that medical robots can be used for clinical care, logistics such as the delivery of medicines and handling of contaminated wastes, and monitoring compliance with voluntary quarantines (Klemes et al., 2020, Khan et al., 2020). They added that new generations of robots either large, small, micro, or swarm could be developed for continuously sanitizing, cleaning, and disinfecting areas (see Fig. 2(d)). In Fig. 2(e), a Spot nimble four-legged robot is presented which was developed by Boston Dynamics in Singapore. Spot robot looks for COVID-19 social-distancing violations. It can map its environment, sense, and avoid obstacles. Although Spot won’t bark at violators, it will play a recorded message reminding them to keep their distance.

From the above discussion, Intra-functions in smart medical robots rely heavily on AI to improve the efficiency of these functions. In other words, AI is considered the regulatory brain for them in the processes of moving, diagnosing various diseases, and recognizing patients. One of AI sub-categories is Computational Intelligence (CI) algorithms which are good alternatives to traditional mathematical or optimal-exact optimization methods. In particular, CI algorithms are nature-inspired algorithms that are able to find a good enough solution for the regarded problem with no need for modifications of algorithm structural steps (Abdel-Basset et al., 2018, Ramos-Figueroa et al., 2020). As the optimal-exact solution methods are exhaustive cost consumption methods, most of the proposed literature moves to the exploitation of near-optimal inspired solution methods or so-called metaheuristics. For instance, CI algorithms have been used in smart surveillance (Eldrany et al., 2019), smart home energy management (Kong et al., 2020), the evaluation of hybrid microgrid configurations (Haidar et al., 2020, Bagheri et al., 2020), energy-efficient fog computing (Reddy et al., 2020), smart healthcare (Ding et al., 2020), and smart agriculture (Abdel-Basset et al., 2020). As a consequence, CI algorithms are very suitable for optimizing the robot functions of movement, recognizing patients, and analyzing samples.

1.1. Contribution of This Paper

This paper introduces a hybrid CI algorithm for pandemic emergency services in smart cities. The contribution of this paper can be briefed as follows:

Ø A hybridization between Moth-Flame Optimization and Marine Predators Algorithms (MOMPA) is proposed
Ø The proposed MOMPA is employed for solving the robot path planning problem with obstacle avoidance constraints.
Ø After the validation experiment, MOMPA is applied for a case study on New Galala city in Egypt.

1.2. The structure of This Paper

The structure of the remaining sections is as follows: Section 2 gives a literature review of the problem of robot path planning. Section 3 discusses the problem definition. Section 4 presents the proposed solution method. In section 5, the validation experiment is conducted. The case study application is introduced in section 6. The conclusion is presented in section 7. Finally, the future works are given in section 8.
The main objective is to find the shortest path between a start and an endpoint without taking into account the collision with obstacles. The results of the hybrid algorithm were compared with the obtained results of ACO in terms of best-founded path and minimum elapsed time.

Garip et al. (Garip et al., 2018) hybrid GA with A* algorithm for solving robot path planning with more productivity. The authors compared GA performance with different parameter values and different path distance measurements. This comparison is conducted on only one environment and is based on the elapsed time. Another hybridization with the A* algorithm was proposed by Song et al. (Song et al., 2017). In particular, the authors hybrid the A* algorithm with a modified PSO algorithm. In this modified version of PSO, the updating of the best-founded path is according to the position points of both local and global best path solutions. Besides several experimental environments, the authors conducted a physical real application with an MT-R robot platform which indicated the efficiency of the proposed algorithm.

Dewang et al. (Dewang et al., 2018) introduced an Adaptive version of the Particle Swarm Optimization (APSO) algorithm for solving the robot path planning problem with obstacle avoidance. APSO was only compared with the original version of the PSO algorithm in five different environments. This comparison was only based on the best duration and founded path length.

Zhang et al. (Zhang et al., 2018) proposed a multi-objective hybrid version of PSO for solving the problem of robot path planning. In particular, the bare-bones version of PSO was hybrid with the crossover GA operand. Regarding the formulation of the problem, it was formulated as a multi-objective optimization problem where the first objective was the path length and the second objective was the safety of the selected path. The experimental comparison was conducted for the same proposed algorithm with different crossover operand types while searching over one environment. Also in (Mac et al., 2017), Mac et al. presented a multi-objective hybrid version of PSO for the same objectives. The hybridization of the proposed technique was applied through three different phases. In the first phase, the environment was sub-divided by triangular shapes. While in the second phase, Dijkstra’s

2. Literature Review

There are several efforts that have been made to solve the problem of path planning with various constraints and characteristics. The most exploited metaheuristic algorithms for solving this problem are Genetic Algorithm (GA) (Lim et al., 2020), Particle Swarm Optimization (PSO) (Keykhah et al., 2020), Ant Colony Optimization (ACO) (Uthayakumar et al., 2020), and Artificial Bee Colony (ABC) (Deng et al., 2020).

Li et al. (Li et al., 2017) presented a mixture between GA and a rearrangement technique of decision variables. In addition, the authors added a smoothing technique of solution path in order to avoid collision with obstacles. The proposed algorithm was compared with the basic version of GA based on the best-founded path length and the sum of the turning angles while solving three different environments.

For instance, Lamini et al. (Lamini et al., 2018) proposed an improved GA with a modified crossover operand by improving the same adjacency. The proposed algorithm was validated in four different environments. The conducted comparison between the improved GA and three compared algorithms was only based on the number of turns founded in the best robot path.

Another improved version of GA is introduced by Tan et al. (Tan et al., 2018) called Improved GA with Ordered Feasible Subpaths (IGAOFSP). In particular, the proposed GA was equipped with mutation and simplification operands to increase the feasibility and efficiency of the founded solutions. Also, the authors suggest the division of the search environment into sub-grids in the main diagonal direction. The performance of the proposed algorithm and the initialization technique was tested on only two environments with resolution 10 × 10 and 20 × 20 grids. In addition, the proposed algorithm was only compared with the GA version.

Regarding three-dimensional path planning, Wang et al. (Wang et al., 2018) proposed a hybridization between GA and another meta-heuristic technique which is Simulated Annealing (SA). The proposed algorithm was tested on only one terrain environment with different start and endpoints. The main objective is to find the shortest path

### Table 2

The descriptive statistics of MOMPA and the comparators.

| Scene No. | Metaheuristic | Minimum | Maximum | Mean | Std. | Success % |
|-----------|---------------|---------|---------|------|------|------------|
| Scene 1   | MOMPA         | 839.00  | 915.00  | 896.300 | 14.64323 | %100       |
| GA (Lamini et al., 2018) | 887.00 | 976.00 | 913.633 | 16.50597 | %100 |
| APSO (Dewang et al., 2018) | 910.00 | 1658.00 | 975.633 | 178.00629 | %100 |
| ACO (Rashid et al., 2016) | 901.00 | 1244.00 | 951.533 | 62.45756 | %100 |
| Scene 2   | MOMPA         | 946.00  | 1508.00 | 1005.5517 | 102.63061 | %96.7 |
| GA (Lamini et al., 2018) | 948.00 | 1399.00 | 1026.7308 | 96.10934 | %86.7 |
| APSO (Dewang et al., 2018) | 1009.00 | 1301.00 | 1154.9130 | 82.63662 | %76.7 |
| ACO (Rashid et al., 2016) | 990.00 | 1301.00 | 1091.8149 | 74.89741 | %90 |
| Scene 3   | MOMPA         | 838.00  | 923.00  | 893.033 | 15.87990 | %100 |
| GA (Lamini et al., 2018) | 890.00 | 1012.00 | 917.6667 | 22.95022 | %100 |
| APSO (Dewang et al., 2018) | 904.00 | 1543.00 | 993.4138 | 142.68910 | %100 |
| ACO (Rashid et al., 2016) | 910.00 | 1011.00 | 937.1667 | 26.62910 | %100 |
| Scene 4   | MOMPA         | 885.00  | 1829.00 | 1000.2143 | 208.77426 | %93.3 |
| GA (Lamini et al., 2018) | 881.00 | 1550.00 | 1002.2500 | 178.87915 | %93.3 |
| APSO (Dewang et al., 2018) | 928.00 | 1566.00 | 1100.0000 | 175.19382 | %95.7 |
| ACO (Rashid et al., 2016) | 912.00 | 1364.00 | 993.6333 | 100.76757 | %100 |
| Scene 5   | MOMPA         | 533.00  | 908.00  | 821.5000 | 80.08590 | %100 |
| GA (Lamini et al., 2018) | 742.00 | 1048.00 | 886.8000 | 63.01582 | %100 |
| APSO (Dewang et al., 2018) | 831.00 | 2071.00 | 1234.8000 | 415.64558 | %83.3 |
| ACO (Rashid et al., 2016) | 855.00 | 1438.00 | 973.3333 | 113.91235 | %100 |
| Scene 6   | MOMPA         | 662.00  | 954.00  | 718.9000 | 77.93027 | %100 |
| GA (Lamini et al., 2018) | 662.00 | 1340.00 | 802.2333 | 172.33842 | %100 |
| APSO (Dewang et al., 2018) | 894.00 | 1775.00 | 1279.1818 | 370.02292 | %26.7 |
| ACO (Rashid et al., 2016) | 682.00 | 1465.00 | 832.1333 | 197.93707 | %100 |
| Scene 7   | MOMPA         | 673.00  | 1046.00 | 890.7667 | 82.20536 | %100 |
| GA (Lamini et al., 2018) | 677.00 | 2091.00 | 960.4444 | 361.75122 | %90 |
| APSO (Dewang et al., 2018) | 708.00 | 1237.00 | 884.7500 | 249.37438 | %13.3 |
| ACO (Rashid et al., 2016) | 714.00 | 1367.00 | 940.2083 | 191.7446 | %80 |
| Scene 8   | MOMPA         | 680.00  | 1214.00 | 835.6000 | 138.56197 | %100 |
| GA (Lamini et al., 2018) | 680.00 | 1485.00 | 866.6207 | 211.19463 | %96.7 |
| APSO (Dewang et al., 2018) | 935.00 | 1496.00 | 1228.7778 | 182.28054 | %100 |
| ACO (Rashid et al., 2016) | 717.00 | 1178.00 | 902.8846 | 137.57189 | %86.7 |
algorithm was used for finding paths with no collisions. In the final phase, the proposed multi-objective PSO was applied to the mentioned objectives. The proposed technique was tested in three different environments and compared with Dijkstra’s algorithm.

In (Mandava et al., 2019), the authors enhanced the performance of PSO by adding a prediction method of obstacles called the Potential Field Method (PFM). The results of PFM were added to the fitness value as a summation of repulsive and attractive potentials. Besides, another heuristic was used for preserving the smoothness of the robot path. For experiments, the authors tested the proposed algorithm on five environments and compared it with the original PSO without PFM or smoothing mechanisms. Also, Wang et al. (Wang et al., 2018) hybrid PFM with ACO for finding the best robot path. The proposed ACO was tested on four $25 \times 25$ grid-based environments and compared with the original version of ACO.

Rashid et al. (Rashid et al., 2016) introduced ACO for solving the free-collision path planning problem. The authors depended on several environments for proving the efficiency of the proposed algorithm which reached the environment with ten obstacles.

Zeng et al. (Zeng et al., 2016) proposed an improved version of ACO with a free step length searching mechanism. In particular, the regarded environment was divided into small grids. Then, the improved algorithm was searching with the new rules of the proposed searching technique which didn’t limit the next cell movement to the neighbor cell. In addition, the length of the next step was expanded from one to the maximum dimension of the searching environment. The proposed algorithm was tested on one $30 \times 30$ resolution environment and compared with the original version of ACO.

Wang et al. (Wang et al., 2019) handling a three-dimensional terrain environment by a modified ACO. In the modified algorithm, the algorithm was enhanced by considering the altitude and distance of each cell. The proposed algorithm was tested on several environments and compared with the original version of ACO.
updating mechanism of the founded path was made by both the local and global best solution path sub-points. Furthermore, an additional mechanism of computing safety factors was applied to find a more safe path with no hills. In a grid-based environment, Li et al. (Li et al., 2018) presented a modified version of ABC. The proposed algorithm can handle such a grid-based environment by adding a new solution updating mechanism. The results representation of the proposed algorithm depended on the best solution simulation.

Nayyar et al. (Nayyar et al., 2020) proposed a modified version of ABC. In this modified version the basic searching structure of ABC was coped with the Arrhenius equation. The proposed algorithm was tested on four different environments and compared with three other metaheuristics.

Regarding other metaheuristics, the authors in (Yang and Li, 2017) presented an improved Biogeography-Based Optimization (BBO). To increase the convergence of the proposed algorithms, the authors employed several enhancement operations, including migration, mutation, and elite retention mechanisms. The proposed algorithms were compared with the basic version of BBO on only one test environment.

For dynamic environments, Patle et al. (Patle et al., 2018) introduced Firefly Algorithm (FA). The proposed algorithm was tested on different scenarios and compared with three other algorithms. In addition to simulation, the authors conducted a real application with a Khepera-II robot.

Saraswathi et al. (Saraswathi et al., 2018) proposed a hybridization between Cuckoo Search (CS) and Bat Algorithm (BA). In the proposed hybrid algorithm, the local best path solution was generated with CS then BA was used to find the globally best path. The validation experiment was conducted on two environments without any statistical comparisons.

3. The Problem of Mobile Robot Path Planning

3.1. Description

The main objective of the mobile robot path planning problem is to find the minimum path length between a target point and a destination point without stumbling into hindrances (Jafri and Kala, 2019). As shown in Fig. 3, the path of a robot is constructed by a set of points of turn which are predefined and considered as decision variables. These points are connected consequently until the destination point is reached. In the case of the straight line that connected two-point of turn lying on a hindrance, a very high penalty value will be added to the value of the objective function (Kala et al., 2009). In other words, the Death Penalty method will be used for handling the case of constraint violations.

3.2. Mathematical Formulation

Let $S$ be an environment Scene where $S = \mathbb{R}^2$ or a two-dimensional searching area. The scene $S$ contains several fixed Hindrances $H$ where $H \subseteq S$ and refers to the set of all points inside the hindrances. Assume a mobile Robot $R$ that is ordered to maneuver the scene $S$. Given a start $A$ and target $T$ places, the main objective of $R$ is to reach $T$ from $A$ with the shortest path without hitting any $H$.

In particular, a robot $R$ can reach $A$ by moving through a set of subpoints $P = p_1, p_2, \ldots, p_n$ in the path from $A$ to $T$, where $n$ is the number of predefined turning points (See Fig. 4). In other words, the structure of a robot path can be written as $(A, p_1, p_2, \ldots, p_n, T)$. Hence, the founded path length of $R$ can be calculated as a Euclidian distance between every two pairs of points, as
\[ D(p_{n-1}, p_n) = \sqrt{(y_{p_{n-1}} - y_{p_n})^2 + (x_{p_{n-1}} - x_{p_n})^2} \]

where \( x \) and \( y \) are the x-coordinate and y-coordinate of the regarded point.

Besides, \( R \) is able to turn left or right in the case of \( H \) facing. The change in the direction of \( R \) can be calculated as:

\[ x_{t+1} = x_t + v \cos(\Delta \theta) \]

\[ y_{t+1} = y_t + v \sin(\Delta \theta) \]

where \( x_t \) and \( y_t \) are the x-coordinate and y-coordinate at time \( t \), respectively. \( v \) is the velocity of \( R \). \( \Delta t \) is the changing in time instance. \( \theta \) is the angle of rotation.

| Satellite Real Image | Bitmap Image Test Case |
|----------------------|------------------------|
| ![Satellite Image 1](image1.png) | ![Bitmap Image 1](image2.png) |
| ![Satellite Image 2](image3.png) | ![Bitmap Image 2](image4.png) |
| ![Satellite Image 3](image5.png) | ![Bitmap Image 3](image6.png) |

Table 3

Representation of Galala Application Parcels.
Table 4

Characteristics of Galala Application Parcels.

| Parcel No. | Dimension | Turn Points (P) | Number of Hindrances (H) | Cases | Robot start location | Emergency location |
|------------|-----------|-----------------|--------------------------|-------|----------------------|-------------------|
|            | X Y       |                 |                          |       | X Y                  |                   |
| Parcel 1   | 500 × 500 | 6               | 6                        | Case 1| 5 5                  | 200 350           |
|            |           |                 |                          | Case 2| 5 5                  | 350 360           |
| Parcel 2   | 928 × 500 | 7               | 11                       | Case 1| 5 500                | 900 10            |
|            |           |                 |                          | Case 2| 850 500              | 150 300           |
| Parcel 3   | 960 × 601 | 7               | 15                       | Case 1| 5 600                | 470 166           |
|            |           |                 |                          | Case 2| 20 175               | 910 270           |
| Parcel 4   | 958 × 595 | 8               | 8                        | Case 1| 153 200              | 791 70            |
|            |           |                 |                          | Case 2| 35 500               | 892 370           |

Table 5

The descriptive statistics of MOMPA and the comparators for Galala Parcels.

| Parcel No. | Case No. | Heuristic        | Minimum | Maximum | Mean  | Std.  | Success % |
|------------|----------|------------------|---------|---------|-------|-------|-----------|
| Parcel 1   | Case 1   | MOMPA            | 511.00  | 1231.00 | 595.80| 132.51| 100       |
|            |          | GA (Lamini et al., 2018) | 524.00   | 1634.00 | 901.27| 353.88| 73.3      |
|            |          | APSO (Dewang et al., 2018) | 521.00   | 1335.00 | 862.60| 215.96| 66.7      |
|            |          | ACO (Rashid et al., 2016) | 568.00   | 1458.00 | 879.80| 185.02| 83.3      |
| Parcel 2   | Case 1   | MOMPA            | 720.00  | 1366.00 | 958.11| 200.82| 60.0      |
|            |          | GA (Lamini et al., 2018) | 740.00   | 1917.00 | 1062.69| 229.18| 43.3      |
|            |          | APSO (Dewang et al., 2018) | NA      | NA      | NA    | 0.0   | 0.0       |
|            |          | ACO (Rashid et al., 2016) | 805.00   | 1391.00 | 1109.90| 140.83| 66.7      |
| Parcel 3   | Case 1   | MOMPA            | 1350.00 | 1855.00 | 1493.90| 129.42| 93.3      |
|            |          | GA (Lamini et al., 2018) | 1343.00  | 2355.00 | 1751.47| 348.97| 63.3      |
|            |          | APSO (Dewang et al., 2018) | NA      | NA      | NA    | 0.0   | 0.0       |
|            |          | ACO (Rashid et al., 2016) | 1539.00  | 2465.00 | 1903.54| 278.35| 43.3      |
| Parcel 4   | Case 1   | MOMPA            | 985.00  | 2327.00 | 1526.57| 447.91| 23.3      |
|            |          | GA (Lamini et al., 2018) | 1312.00  | 3678.00 | 2241.00| 698.01| 36.7      |
|            |          | APSO (Dewang et al., 2018) | 1899.00  | 1899.00 | 1899.00| NA    | 3.3       |
|            |          | ACO (Rashid et al., 2016) | 1230.00  | 1764.00 | 1569.33| 294.93| 10.0      |

Algorithm 1

Moth-Flame Optimization Algorithm (MFO)

1: Set MFO parameters: $T_{max}, N, \text{ and } \beta$
2: Initialize the initial population of moths positions $i = 1, 2… N$
3: while ($t \leq T_{max}$) do
4: Evaluate each moth position
5: if ($t = 1$) then
6: Sort the fitness of the first population of moths
7: Sort the first population of moths according to sorted fitness
8: Update flames
9: else
10: Sort the concatenated fitness of the previous and best flame fitness
11: Sort the population of moths according to sorted fitness
12: Update flames
13: endif
14: Update the positions of moths with respect to the number of flames
15: Update the number of flames
16: endwhile
17: return the best solution

Algorithm 2

Marine Predators Algorithm (MPA)

1: Set MPA parameters: $i_{max}, n, FADS, \text{ and } \gamma$
2: Initialize the initial population of preys $i = 1, 2… N$
3: while ($it \leq i_{max}$) do
4: Evaluate each prey
5: Construct the matrix of elite
6: if ($it < i_{max}/3$) then
7: Apply the first phase of searching
8: elseif ($i_{max}/3 < it < 2i_{max}/3$) then
9: Split the population of preys
10: Apply the second phase of searching
11: else
12: Apply the third phase of searching
13: endif
14: Applying FADS Effect
15: Update the matrix of elite
16: endwhile
17: return the best solution
particular, they use a flying mechanism called transverse orientation. This logarithmic spiral movement is based on the following equation:

\[ x_{ij}^{new} = |F_{ij} - x_{ij}| \alpha^{b} \cos(2\pi b) + F_{ij} \]  \hspace{1cm} (4)

where \( x_{ij}^{new} \) is the new generated solution, \( F_{ij} \) is the corresponding flame solution, \( x_{ij} \) is the current solution, \( b \) is a predefined constant (equal to one), and \( \alpha \) is a random number between \([-1, 1]\).

Another point of power is that MFO uses an adaptive mechanism for updating solutions. In other words, the number of flames is adaptively reduced according to:

\[ m = \text{round} \left( N - t \times \frac{N - t}{T_{\text{max}}} \right) \]  \hspace{1cm} (5)

Where \( m \) is the adaptive number of flames, \( N \) is the maximum number of flames, \( t \) is the current iteration, and \( T_{\text{max}} \) is the maximum number of iterations. To applying MFO in the proposed problem, each solution represents the coordinates of a robot path. Algorithm 1 shows the MFO pseudo-code.

### 4.2. Marine Predators Algorithm

The main inspiration of the Marine Predators Algorithm (MPA) (Faramarzi et al., 2020) is the biological interaction between predator and prey in the marine environment. This interaction behavior mainly depends on Lévy and Brownian distribution behavior. The main governing assumptions of MPA are:

- Predators use the Lévy movement for the areas with a low intensity of prey and Brownian movement for the environment with abundant prey.
- The percentages of the Lévy and Brownian movements are the same.
- The behavior of a predator changes according to two factors natural eddy formation or human-caused (FADs).
- The best movement for a predator is the Lévy strategy in a low-Velocity ratio (\( V=0.1 \)) while the movement of the prey is either Brownian or Lévy.
- The best movement for a predator is the Brownian strategy if a prey moves in Lévy in the case of the unit Velocity ratio (\( V=1 \)). The other cases depend on system size.
- The best movement for a predator is not moving at all while the prey is moving either Brownian or Lévy in the high-Velocity ratio (\( V\geq 10 \)).
- In the high-velocity ratio, the prey utilizes good memory in reminding of their associates and the location of good foraging.

As MPA is a population-based metaheuristic, it begins with the random initialization of the starting population. After the initialization step, the obtained solutions are ordered according to their fitness in order to construct the matrix of the elite. This matrix contains the best solution or top predator which is replicated \( n \) times where \( n \) is the
population size. At the end of each search iteration, the matrix of the elite is updated with a better predator. Besides, another matrix of prey is created with the same size as the matrix of the elite. This matrix is responsible for updating the movement of the elite predator. In particular, the searching process of MPA consists of three main phases according to the level of the velocity ratio. In the first phase, the predator is not moving at all while the prey is moving either Brownian or Lévy at a premature phase of searching (in the first third of total searching iterations) or with a high-velocity ratio ($v > 10$). This is done by the following:

$$\text{step}_i = R_B \odot \left( \text{elite}_i - R_B \odot \text{prey}_i \right) \quad i = 1, 2, \ldots, N$$

(6)

$$\text{prey}_i(t+1) = \text{prey}_i(t) + \gamma \cdot R \odot \text{step}_i$$

(7)

where $\text{step}_i$ is the prey step size, $R_B$ is a vector of Brownian random numbers, the notation $\odot$ means entry-wise multiplications, $\text{elite}_i$ is the best predator from the matrix of the elite, $\text{prey}_i$ is a solution from the matrix of prey, $\gamma$ is a constant number, and $R$ is a vector of uniform random numbers between $[0, 1]$.

The second phase of searching happens in the second third of the total searching iterations. The best movement for a predator is the Brownian strategy if a prey moves in Lévy in the case of $V=1$. Particularly, the population of search agents is divided into two subpopulations. The first half is updated as follows:

$$\text{step}_i = R_L \odot \left( \text{elite}_i - R_L \odot \text{prey}_i \right) \quad i = 1, 2, \ldots, N/2$$

(8)

$$\text{prey}_i(t+1) = \text{prey}_i(t) + \gamma \cdot R \odot \text{step}_i$$

(9)

where $R_L$ is a vector of Lévy random numbers.

The second half of the population search agents is updated as follows:

$$\text{step}_i = R_B \odot \left( R_B \odot \text{elite}_i - \text{prey}_i \right) \quad i = N/2 + 1, \ldots, N$$

(10)
Marine Predators Algorithm (MOMPA) is introduced. In particular, the first population of flames is replaced with a duplication of the best-founded solution by MPA. Algorithm 3 discusses the phases of MOMPA. The proposed algorithm is used for finding the shortest collision-free path of the COVID-19 pandemic medical robot. Before applying to the regarded problem, MOMPA is tested on several minimization problems obtained from (Mirjalili, 2015). As shown in Fig. 5, the convergence speed of the proposed algorithm is very high compared to the original MFO and MPA. This indicates that the proposed algorithm can efficiently capture the best solution to an optimization problem.

5. Validation Experiment

5.1. Characteristics

In this section, the proposed algorithm is tested on several benchmarks of the path planning problem. The characteristics of all scenes with hindrances. The first five scenes are downloaded from (Rahul Kala 2020). While the rest scenes are generated by the authors with various hindrances’ shapes. The resolution of all scenes is fixed to $500 \times 500$. In addition, MOMPA is compared with several metaheuristics that previously solved the problem of path planning efficiently, including GA (Lamini et al., 2018), APSO (Dewang et al., 2018), and ACO (Rashid et al., 2016).

All algorithms are coded in MATLAB© 2020. For all algorithms, the total number of iterations is set to 100, and the number of search agents is set to 30. In particular, for MOMPA, the total number of iterations is divided into 70 for MPA and 30 for MFO. Whereas the number of search agents is divided as 10 for MPA and 30 for MFO. The other parameters of algorithms are kept as mentioned by their authors. The simulations of the scenes are coded in MATLAB© 2020 and carried on a 64-bit operating system with a 2.60 GHz CPU and 6 GB RAM.

5.2. Statistical Results

Table 1

In order to obtain statistical results fairly, each algorithm runs for 30 independent runs. Table 2 shows the descriptive statistics of the validation experiment. The discussed descriptive statistics include minimum founded path length (the best), maximum founded path length, mean of the path lengths during 30 independent runs, and the ratio of successful trials during 30 independent runs. As observed, the overall performance of the proposed algorithm is acceptable for solving the problem of robot path planning. MOMPA is able to find the shortest path length for scene 1, scene 2, scene 3, scene 5, and scene 7. While for scene 6 and scene 8, the best-founded path length of MOMPA is the same as that founded by GA. For scene 4, the proposed algorithm is unable to reach the best path length as GA. Fig. 6 shows a simulation of the best-founded paths by MOMPA for all test scenes. As observed, the proposed algorithm can find the shortest path between the beginning point to the target. In addition, the resultant paths are characterized by the absence of excessive deviation from the required.

Regarding the mean values, the best value is founded by MOMPA for scenes 1, 2, 3, 5, and 7. While for scenes 4, 6, and 8, the best-founded path length of MOMPA is the same as that founded by GA. For scene 4, the proposed algorithm is unable to reach the best path length as GA. Fig. 6 shows a simulation of the best-founded paths by MOMPA for all test scenes. As observed, the proposed algorithm can find the shortest path between the beginning point to the target. In addition, the resultant paths are characterized by the absence of excessive deviation from the required.

According to the mean values, the best value is founded by MOMPA for scene 1, scene 2, scene 3, scene 5, scene 6, and scene 8 with high successful ratios. While for scene 4, the best mean value is achieved by ACO with the highest success ratio. For scene 7, the best mean value is founded by APSO however its success rate is the worst.
In addition, the Friedman Ranked Mean test (Navarres and Aznarte, 2020) is performed for evaluating the performance of MOMPA and its comparators. As presented in Fig. 7, the proposed algorithm has the minimum ranked main value in general except for scene 4 and scene 7. This indicates the good performance and prosperity of the proposed algorithm.

6. Application

In this section, MOMPA is applied for smart autonomous handling of medical emergencies.

6.1. Description of Application Area

Currently, Egypt is entering an era of creating new cities that are smart and structured based on Artificial Intelligence (AI) and IoT. For instance, several new smart cities have been recently established, such as the New Administrative Capital (New Cairo City) (Eldrandaly et al., 2019), New Alamein (Attia, 2019), and New Galala City (Abdelazeem et al., 2019). In particular, Egypt is adopting the methodology of AI and automatic control in its newly built cities following Egypt’s Vision 2030 (Mondal et al., 2019).

New Galala City represents one of the most significant urban development projects in Egypt constructed under the supervision of the Authority of Engineering of the Armed Forces. The New Galala is projected in northwestern Egypt and is located on the northern Galala plateau on the western side of the Gulf of Suez. It includes luxury and middle-income housing, tourist resorts, a water park, a phosphate fertiliser plant, and a university (Galala University). The city will also include the first Olympic village in Egypt. In addition, it is linked to Cairo and other sites by a newly built highway.

6.2. Problem Statement

The hilly nature of the city makes it best to offer autonomous medical emergency services (See Fig. 8). Therefore, we select this city as an area of application. To do this, several parcels of New Galala are picked and captured by Google Earth©. After that, the captured satellite images are converted to bitmap format, as shown in Table 3. In Table 4, the characteristics of medical emergencies test cases are given.

6.3. Experimental Results

This section describes the results of MOMPA for solving the regarded problem. Table 5 shows the descriptive statistics of MOMPA and its previous comparators, including GA, APSO, and ACO. As observed, the proposed algorithm can find the best path length to the emergency locations in all cases except for case 1 of parcel 2. In this case, the best path is founded by GA; however, the mean value of MOMPA is still the highest. Fig. 9-12 show the best-founded way by MOMPA for all parcels.

Regarding the mean values, the proposed algorithm achieves the best mean values in all cases except for case 2 of parcel 4. In this case, the best mean value is founded by ACO, whereas its success ratio is significantly lower than MOMPA. The success ratio of the proposed algorithm is considerably higher than other compared algorithms in most cases.

7. Conclusion

Pandemic Emergency medical services are very critical issues in any smart city infrastructure. In this paper, a hybrid metaheuristic MOMPA is proposed for the smart handling of COVID-19 pandemic emergencies. As shown in the validation experiment, the efficiency of the proposed algorithm is acceptable in handling the problem of robot path planning. Despite the intense competition between the proposed algorithm and GA, the proposed algorithm can reach the best path length in several cases. In addition, the success ratio of MOMPA proves the excellent exploration and exploitation capabilities of the search space without hindrance collisions. After validation, MOMPA is applied to the COVID-19 pandemic emergency medical service in New Galala city. The experimental results show the significant-high-quality solution of MOMPA in almost all cases.

8. Future works

For future works, we suggest using the proposed hybrid algorithm for other related COVID-19 pandemic problems, such as smart detection of COVID-19 patients and the management of COVID-19 sanitary isolation. Also, MOMPA can solve different optimization problems such as wireless sensor coverage problems and power flow optimization. Also, the problem of collision-free robot path planning can be solved in a higher dimension with more complicated constraints.

Declaration of Competing Interest

This is hereby certify that the paper is original, neither the paper nor a part of it is under consideration for publication anywhere else. Also, we have no conflicts of interest to disclose.

References

Soyata, T., Habibzadeh, H., Ekenna, C., Nussbaum, B., & Lozano, J. (2019). Smart city in crisis: Technology and policy concerns. Sustainable Cities and Society, 50, Article 101566.
Qureshi, K. N., Tayyab, M. Q., Rehman, S. U., & Jeon, G. (2020). An interference aware energy efficient data transmission approach for smart cities healthcare systems. Sustainable Cities and Society, 62, Article 102392.
Rehanizadeh, S., & Kermanshachi, S. (2020). Post-disaster reconstruction of transportation infrastructures: Lessons learned. Sustainable Cities and Society, 63, Article 102505.
Westraadt, L., & Callitz, A. (2020). A modelling framework for integrated smart city planning and management. Sustainable Cities and Society, 63, Article 102444.
Francini, M., Gaudio, S., Palermo, A., & Viapiana, M. F. (2020). A performance-based approach for innovative emergency planning. Sustainable cities and society, 53, Article 101906.
Dong, S., Yu, T., Farahmand, H., & Mostafavi, A. (2020). Probabilistic modeling of cascading failure risk in interdependent channel and road networks in urban flooding. Sustainable Cities and Society, 62, Article 102398.
Kumar, S., Raut, R. D., & Narkhede, B. E. (2020). A proposed collaborative framework by using artificial intelligence-internet of things (AI-IoT) in COVID-19 pandemic situation for healthcare workers. https://doi.org/10.1080/20479700.2020.1810453, 13(4), 337–345. https://doi.org/10.1080/20479700.2020.1810453.
Klement, J. J., Fan, Y., Van, Tan, R. R., & Jiang, P (2020). Minimising the present and future plastic waste, energy and environmental footprints related to COVID-19. Renewable and Sustainable Energy Reviews, 127, Article 109883. https://doi.org/10.1016/J.RSER.2020.109883.
Khan, Z. H., Siddique, A., & Lee, C. W. (2020). Robotics Utilization for Healthcare Digitization in Global COVID-19 Management. International Journal of Environmental Research and Public Health, 17(11), 3819. https://doi.org/10.3390/IJEHPH17113819. 2020Vol. 17, Page 3819.
Abdel-Basset, M., Abdel-Fattah, L., & Sangaiah, A. K. (2018). Metaheuristic Algorithms: A Comprehensive Review. Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications, 185–231. https://doi.org/10.1109/9780-1-213284-9.00010-4.
Ramos-Figueroa, O., Quiroz-Castellanos, M., Mezura-Montes, E., & Schütze, O. (2020). Metaheuristics to solve grouping problems: A review and a case study. Swarm and Evolutionary Computation, 53, Article 100643.
Eldrandaly, K. A., Abdel-Basset, M., & Abdel-Fattah, I. (2019). P3T-Surveillance coverage based on artificial intelligence for smart cities. International Journal of Information Management, 49, 520–532.
Kong, X., Sun, B., Kong, D., & Li, B. (2020). Home energy management optimization method considering potential risk cost. Sustainable Cities and Society, 62, Article 102378.
Haidar, A. M., Fakhra, A., & Helwig, A. (2020). Sustainable energy planning for cost minimization of autonomous hybrid microgrid using combined multi-objective optimization algorithm. Sustainable Cities and Society, 62, Article 102391.
Bagheri, A., Jabbari, A., & Mobayen, S. (2020). An Intelligent ABC-Based Terminal Sliding Mode Controller for Load-Frequency Control of Islanded Micro-Grids. Sustainable Cities and Society, 64, Article 102544.
Reddy, K. H. K., Lubach, A. K., Pradhan, B., Dash, J. K., & Roy, D. S. (2020). A genetic algorithm for energy efficient fog layer resource management in context-aware smart cities. Sustainable Cities and Society, 63, Article 102428.
Ding, W., Abdel-Basset, M., Eldrandaly, K. A., Abdel-Fattah, L., & de Albuquerque, V. H. C. (2020). Smart Supervision of Cardiomyopathy Based on Fuzzy
