The Aeronautical Journal (2025), **129**, pp. 2622–2647
doi:10.1017/aer.2025.10025

ROYAL AERONAUTICAL SOCIETY

RESEARCH ARTICLE

A greedy initialiser based meta-heuristic approach for planning unmanned aerial vehicles

S. Aslan and T. Erkin

Aeronautical Engineering, Erciyes University, Kayseri, Turkey
Corresponding author: T. Erkin; Email: tevfikerkin@erciyes.edu.tr

Received: 2 January 2025; **Revised:** 24 April 2025; **Accepted:** 29 April 2025

Keywords: greedy initialiser; IP algorithm; meta-heuristic; path planning; unmanned aerial vehicles

Abstract

The unpredictable benefits and low operational costs of unmanned aerial vehicles (UAVs) compared to their conventional counterparts caused a tremendous change in commercial and military concepts, and different solutions to problems regarding modern aerial vehicles were tried to further improve flight and job performances. One of the most challenging problems about the UAVs is known as the path planning problem, and a solution should satisfy some objectives related to the enemy anti-air weapons, fuel or battery consumption, and manoeuvre capability of the UAV being operated optimally. Immune plasma algorithm (IP algorithm or IPA) is a recent meta-heuristic optimiser, and its competitive performance has been validated over a set of engineering problems. In this study, a greedy initialiser that is responsible for generating a population of IPA was first introduced. Also, the treatment schema of the IPA was completely redesigned for more robust and detailed search characteristics without requiring either IPA-specific control parameters or their subtle configurations. The new IPA-based path planner, called greedy initialiser IPA (gintIPA), was tested by using three battlefields and 12 test cases belonging to them, and the obtained results were compared with the results of the well-known meta-heuristic techniques. Comparative studies showed that gintIPA is capable of planning more robust and safe UAV paths than other techniques for 91.6% of all test cases. The proposed greedy initialiser gives a chance to start optimisation with qualified individuals and the newly designed treatment schema improving both exploitation and exploration routines significantly contributes to the gintIPA when outperforming other path planners.

Nomenclature

UAV	unmanned aerial vehicle
UCAV	unmanned combat aerial vehicle
IPA	immune plasma algorithm
gintIPA	greedy initialiser immune plasma algorithm

1.0 Introduction

Unmanned aerial vehicles (UAVs; also called the unmanned combat aerial vehicles or UCAVs) and similar equipment capable of carrying sophisticated weapon systems as well as monitoring and tracking equipment, revolutionised the world [1–3]. Well-known military doctrines and complex numerical strategies are designed by considering the advantageous sides of these unmanned vehicles in the production cost and maintenance, and increasing amounts of budgets are allocated for developing new forms or solving critical problems related to them. Path planning is among the most critical problems for the UAV or UCAV systems, and an optimal or near optimal solution according to some minimisation objectives of the enemy threats, fuel or battery consumption,and manoeuvres has a direct influence on the flight safety and task success [4–6].

© The Author(s), 2025. Published by Cambridge University Press on behalf of Royal Aeronautical Society.

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

In recent years, different graph-based path planning techniques such as Dijkstra, A∗, and D∗ lite, sample-based methods including rapidly-exploring random trees (RRTs), and probabilistic roadmaps (PRMs), popular field-based approaches like virtual force (VF) and artificial potential field (APF) have been explored successfully. Although all these path planners come with a set of unique solving characteristics, they generally have difficulties when addressing the bottlenecks stemming from the relatively high computational burdens, requirements of detailed battlefield maps and extra storage spaces for them. Meta-heuristic algorithms that are in the subgroup of artificial intelligence were also used for solving the UAV or UCAV path planning problem, and promising results compared to the results of the classical planners were obtained.

The number of meta-heuristic algorithms that are employed with their standard workflows or after a substantial number of changes on various stages demonstrates the importance of the UAV or UCAV path planning problem and the urgent demands to more versatile and stable path planners designed by bringing strong search characteristics of a modern meta-heuristic and genuine procedures together. Immune Plasma algorithm (IP algorithm or IPA) is a recent meta-heuristic that is inspired by the fundamental steps of a medical method that gained popularity again with the COVID-19 pandemic and called immune or convalescent plasma treatment [7]. The standard IPA or its modernisations have been used efficiently as a noise minimiser [8], neural network trainer [9], time series predictor [10], radio channel assigner [11], wireless sensor distributor [12] and UAV or UCAV path planner [13]. In this study:

- A greedy heuristic approach that guarantees the maximum safety while keeping the length of line segment and turning angle at minimum if possible for generating a UAV or UCAV path by evaluating the geometric description of the battlefield is utilised.
- The quality of the path discovered by the greedy approach gives inspiration to initialise the population of a meta-heuristic algorithm and IPA is chosen as the reference meta-heuristic.
- Additionally, the plasma collection and transfer procedures of the IP algorithm are completely remodeled to gain advantages from the qualified individuals of the initialised population more and an efficient search is maintained, eliminating the need for control parameters that set the number of donors and receivers.

The performance of the IPA-based path planner or greedy initialiser IPA (gintIPA) was investigated in detail over three different battlefield scenarios and their 12 test cases. A set of comparative studies between the gintIPA and other 14 path planners including IPA, Genetic algorithm (GA), a modified version of GA named GAPSO, Moth-Flame Optimisation (MFO), Salp Swarm algorithm (SSA), Pathfinder algorithm (PFA), Stain Bowerbird Optimisation (SBO), Sine Cosine algorithm (SCA), a variant of the Teaching Learning Based Optimisation (TLBO) called ECTLBO, Grey Wolf Optimiser (GWO), a hybridisation of GWO and Symbiotic Organism Search (SOS) or HSGWO-MSOS, an implementation of the Particle Swarm Optimisation (PSO)- a substantially enhanced Particle Swarm Optimisation (PSO) implementation for short CIPSO-, Artificial Ecosystem Optimiser (AEO), and neighbourhood-based search enhanced AEO (NSEAEO) was carried out. Comparative studies between gintIPA and other mentioned meta-heuristic-based path planners informed that gintIPA outperforms them for almost all test cases. The population generated by the proposed greedy initialiser contains candidates whose qualities are generally better than the qualities of the candidates generated by the common randomise initialisation method and allows gintIPA to start ptimisation in a search space narrowed subtly. Also, the newly designed treatment schema that differentiates the plasma collection and its transfer carefully enhances the exploitation performance of the gintIPA and assists it in finding safer and more efficient paths than other tested algorithms. The remainder parts of the manuscript are organised as follows: In Section 2, the mathematical properties of the path planning problem are given. While the fundamental steps of the IPA are summarised in Section 3, details of the greedy initialiser and treatment schema and how they integrate into the workflow of IPA are presented in Section 4. Section 5 is devoted to the experiments and comparative studies. Finally, in Section 6, some concluding remarks and proposals for future work are provided.

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

1.1 Related works

Zhang et al. proposed the Fitness Scaling Adaptive Chaotic PSO (FAC-PSO), which enhances the traditional PSO by adapting the parameters managing particle velocity and position using chaotic random numbers to improve the performance [14]. Duan and Qiao developed Pigeon-Inspired Optimisation (PIO) by leveraging the natural navigation abilities of pigeons as a point of inspiration [15]. The path planning problem for UAVs was addressed with the Artificial Bee Colony (ABC) algorithm combined with balance evolution (BE-ABC) [16]. In a comparative study on three different scenarios, BE-ABC outperformed standard ABC and two variants of ABC, those being I-ABC and IF-ABC [16]. In another study, Tang and Zhou hybridised Glowworm Swarm Optimisation (GSO) with PSO and called as PGSO [17]. Zhang et al. further developed their study by considering path planning for UAVs as a three-dimensional optimisation problem [18]. They also evaluated the new constraints that the variables of climbing and turning angles, in addition to fuel consumption and enemy threats, for the determination of the optimal path [18]. This comprehensive study was completed with a new variant of DE (differential evolution) designated as the DE algorithm with Level Comparison (DELC) [18]. An innovative study integrating the Wolf Colony algorithm (WCA) with the complex method to improve navigation in challenging optimisation environments was carried out by Zhou et al. with the discovery of the Complex Method-based WCA (CWCA) [19].

Zhang et al. considered the path planning problem under dynamic conditions in a three-dimensional environment and introduced a new method by reinforcing the PIO with predator-prey (PPPIO) [20]. In another study, Chen et al. introduced Modified CFO (MCFO) by updating Central Force Optimisation (CFO) with PSO and GA to plan paths for vertical take-off and landing (VTOL) vehicles [21]. A novel algorithm called Quantum Wind Driven Optimisation (QWDO) was designed for the problem by Zhou et al. [22]. Experimental results demonstrated that QWDO outperforms several existing algorithms, including Quantum BA (QBA) and PSO, in terms of efficiency and effectiveness in finding optimal flight paths while avoiding threats [22]. Wang et al. introduced the Improved Bat algorithm (IBA) by incorporating the mutation operator from the DE to enhance the diversity of solutions in the Bat algorithm (BA) [23]. The performance of the GWO in solving the problem was investigated by Zhang et al. [24]. In a separate study using the PSO algorithm as a reference, the adaptive decision operator was utilised, which eliminated the problem of trapping into local minima and improved the quality of the results [25]. In a further study, Liu et al. contributed to the existing literature by investigating the four-dimensional (4D) problem for multiple UAVs, in which time is also considered [26]. A novel approach, Quantum Encoding BA (QBA), which is built on quantum, NOT gate and quantum rotation methods, has been demonstrated by Luo et al. to obtain suitable paths for UAVs [27]. To overcome the premature convergence problem of the Collective Decision Optimisation algorithm (CDOA), Zhang et al. devised the Extended CDOA (ECDOA), which draws on an external archive and employs an operator that increases diversity in the generation of new agents [28].

Alihodzic et al. adapted Elephant Herding Optimisation (EHO) to solve the path planning problem and demonstrated it as Adjusted EHO (AEHO) [29]. Miao et al. carried out a series of experiments including GWO, ABC and BA, to investigate the potential of Modified SOS Algorithm Based on the Simplex (SMSOS) in path planning for UAVs, and the results proved the superiority of SMSOS [30]. In further study, the utilisation of Directional Drive-Rotation Invariant Quadratic Interpolation White Shark Optimisation algorithm (DD-RQIWSO) in planning of paths for UAVs has been investigated by Miao et al. [31]. Dolicanin et al. developed the Brain Storming Optimisation (BSO) algorithm to produce more appropriate solutions [32]. Lin et al. employed the APF method to increase the convergence speed of the BA integrating a success rate strategy and chaos theory to avoid local optimums and ensure solution diversity [33]. Pan et al. contributed to the field by creating CIJADE, a hybrid algorithm that integrates the strengths of CIPDE and JADE, two DE variants [34]. CIJADE has been put to 28 different tests with 11 different well-known DE variants before the path planning problem and proved its superiority [34]. Reinforcement Learning-based GWO (RLGWO) was introduced by Qu et al. to improve the balance between exploitation and exploration, enhancing overall solution quality, while also smoothing the generated paths with B-spline curves to ensure better flight compatibility [35]. In a similar study,

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

Qu et al. considered the path planning problem with the HSGWO-MSOS, which was created by bringing together the Simplified GWO (SGWO) and Modified SOS (MSOS) [36]. Yi et al. presented the Quantum Monarch Butterfly Optimisation (QMBO), utilising a quantum approach to refine poor solutions and enabling the algorithm to achieve global optimum results more effectively [37].

Neighbourhood Global Learning Based Flower Pollination algorithm (NGFPA) was used successfully in obtaining optimal routes for UAVs [38]. Zhu et al. conducted a survey of algorithms designed for the path planning problem, complemented by the contribution of the Spider Monkey Optimisation (SMO) algorithm to the literature [39]. Following this pioneering work, SMO was enhanced with the help of Cooperative Co-Evolution (CE) and demonstrated the superior performance of the newly developed CE-SMO algorithm in various scenarios compared to SMO, GWO, PSO and ABC [40]. The BA was updated by incorporating an advanced search step, leading to the development of the Improved BA (IBA) [41]. The novel method demonstrated a solution speed twice as fast as that of the standard BA, along with a 14% improvement in solution quality [41]. Mixed strategy-based GSA (MSGSA) was proposed by reinforcing the GSA algorithm with α coefficient and Cauchy approximation [42]. Jiang et al. conducted a study on navigating UAVs through complex landscapes including radar, missile and defence systems, employing the Partially Observable Markov Decision Process (POMDP) and Monte Carlo Tree Search (MCTS) methods along with the GWO [43]. Jarray et al. developed a new approach by parallelising the GWO algorithm to provide better solutions in a shorter time and announced it as Parallel Cooperative Coevolutionary GWO (PCCGWO) [44]. The Chimp Optimisation algorithm (ChOA) was enhanced with a dynamic technique for updating the foraging phase, leading to the Improved ChOA (IChOA), which significantly enhances solution quality and avoids local optima [45]. Wang et al. improved the MA by employing an adaptive Cauchy mutation for velocity adjustments and an exponentially decreasing inertia weight to balance exploration and exploitation [46]. Niu et al. modified the AEO algorithm to increase search capability and avoid local optimum solutions [47]. They also focused on possible ways to advance this work and conducted a new experiment in a three-dimensional environment with the newly proposed Modified AEO (MAEO) [48].

Jia, Qu and Li improved the consistency of the results by unequally dividing the line between the UAV start and target points using PSO and rotation-based approaches augmented with Double Layer Coding (DLCRPSO) [49]. Moreover, Jia et al. published an innovative study in which fitness distance correlation (FDC), adaptive K-neighbourhood-based learning strategy and differential hunger-hunting strategy methods were used besides the DLC to further improve the performance of GWO with DLC (DLC-K-FDGWO) [50]. Zhang et al. adapted the Search and Rescue (SAR) algorithm to the problem by combining it with Heuristic Crossing (HC-SAR) for the first time and performed real-time path planning for the UAV [51]. Aquila Optimisation (AO) was chaoticised using the chaotic map method (CAO) in combination with Simulated Annealing (SA), leading to the invention of CAOSA [52]. Also, Saadi et al. designed a new approach to the African Vulture Optimisation algorithm (AVOA) with the Chaotic Cauchy Opposition method (CCO-AVOA) [53]. GWO was hybridised with DE and the Hybrid GWODE (HGWODE) algorithm was employed in the CEC2014 benchmark functions as well as the path planning problem and the results indicated that HGWODE achieved noticeably more robust solutions [54]. A different approach from other studies, Chen attempted to plan the path by considering the UAV dynamics with the Truncated Mean Stable Strategy-Based Bat Optimisation (TMS-SBA) algorithm in the spherical coordinate system instead of the Cartesian coordinate system [55]. Huang et al. used Adaptive Cylindrical Vector (ACV) to constrain the search space based on the UAV dynamics and also enhanced the PSO algorithm by using the DE operator to avoid local optima and introduced ACVDEPSO to the literature [56]. Experimental results using Real Digital Elevation Model (DEM) maps with different PSO variations validated ACVDEPSO supremacy, demonstrating the performance of the method [56].

The idea of evolving the Firework algorithm (FWA) by utilising the advantageous sides of DE was suggested by Zhang et al. and the Hybrid FA with DE (HDEFWA) approach was applied to path planning for multiple UAVs [57]. Another study on hybridising algorithms has been carried out by Yu et al., and they introduced the Reinforcement Learning-based Multi-Strategy CS (RLMSCS) [58]. They also

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

considered the turning angle and collision avoidance when planning a path for UAV in three-dimensional environment [58]. Hu et al. tried to tackle the path planning problem for UAVs efficiently with a new algorithm called SaCHBA_PDN, which was created by leveraging the Honey Badger algorithm (HBA) with Bernoulli shift map, piecewise optimal decreasing neighbourhood, and horizontal crossing with strategy adaptation [59]. In addition to this research, Hu et al. introduced the Enhanced Multi-Strategy Bottlenose Dolphin Optimiser (MBDO), integrating a fish aggregating device strategy, random mixed mutation and chaotic opposition-based learning [60]. This approach effectively enhances the balance between exploration and exploitation phases, particularly in solving constrained optimisation problems [60]. The concept of hybridising PSO and Enhanced SSA algorithms (PESSA), along with the idea of operating them in parallel to achieve better solution was put forward by Wang et al. [61]. A comparison was drawn with 12 different meta-heuristics on ten different tests to verify its success, and PESSA was always the best on average [61]. In another novelty, the mutation and crossover steps of GA are improved with sigmoid function, the selection strategy is revised and combined with PSO and introduced as a path planner in three-dimensional environment by Deng et al. [62].

Aslan developed the IP algorithm or IPA followed by the introduction of the hospital IPA (hospIPA) and Q-learning IPA (Q-LIPA) algorithms on the path planning problem for UAVs [63, 64]. Fu et al. implemented Red-billed Blue Magpie Optimiser (RBMO) approach for five different engineering problems, including the UAV path planning problem [65]. Yang et al. enhanced the Sparrow Search algorithm (SSA) by incorporating an adaptive weight jumping mechanism alongside suicide mutation perturbation techniques, resulting in the Multi-Mechanism Balanced Advanced Sparrow Search (BALSSA) method, which demonstrated improved performance [66]. Haris et al. developed the Cognitive and Social Information-Based Hyperbolic Tangent PSO (HT-PSO) to improve convergence speed, demonstrating remarkable performance in tests conducted across three distinct environments [67]. Qu et al. addressed the path planning problem as two-stage that horizontal and vertical and they also presented a new method called Improved Elephant Clan Optimisation (IECO) for an efficient solution [68]. Another contribution to the literature was made by Chen et al. with Improved Dung Beetle Optimiser (IBDO) [69]. They utilised novel methods including Cauchy mutation method to augment the performance of IBDO [69].

2.0 Formulation of the Geometrical Path Planning Problem

Critical details of an enemy threat about how it detects or senses a UAV or UCAV, how the detection or sensing range and danger level are appointed, and how a path from the start point to the target point is represented are all related to the mathematical model of the path planning problem. A common mathematical model that can also be called the geometrical model for the UAV or UCAV path planning problem represents an enemy threat by a circle for which the centre coordinate and radius showing the sensing range are known in advance. Moreover, for representing a UAV or UCAV path, a set of points whose members are chosen subtly after considering the minimisation requirements to do with the enemy threats, fuel or battery consumption, flight length and turning manoeuvres is employed.

The path planning operations for the geometrical description of the problem are started by drawing a reference line from the start point or $p_s = \{x_s, y_s\}$ to the target point or $p_t = \{x_t, y_t\}$ and dividing it into $D + 1$ line segments with the help of D segmentation points. When the segmentation of the reference line is concluded, a line named as the L_1 passing through the first segmentation point and intersecting the reference line vertically, a line named as the L_2 passing through the second segmentation point and intersecting the reference line vertically and so on are drawn and then a line set $L = \{L_1, L_2, \ldots, L_{D-1}, L_D\}$ is obtained. The set L becomes a beacon for the subsequent operations of the path planning problem. If each line in the set L is used in a manner that the point p_1 from the line L_1, the point p_2 from the line L_2 and so on are chosen and then combined with the p_s and p_t appropriately, a point set $P = \{p_s, p_1, p_2, \ldots, p_{D-1}, p_D, p_t\}$ can be organised. The set P is capable of intrinsically describing a UAV or UCAV path. When the neighbour point pairs of the set P are connected with a line segment, a path from the point p_s to the point p_t is generated [47].

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

Some calculations required to find the members of the set L and set P should be supported with a strong geometrical background. Each line in the set L must vertically intersect the reference line by only passing through the matched segmentation point. Furthermore, the points of the set P ranging from the p_1 to p_D must satisfy the equation of the corresponding line in the set L. For reducing the calculation difficulties about the members of set L and set P, a new coordinate system that accepts the reference line between the p_s and p_t as the horizontal axis can be utilised. To transform the point p_k in the original coordinate system into the new coordinate system, Equation (1) can be referenced. While the x_k and y_k show the horizontal or x-axis value and vertical or y-axis value of the point p_k, the $\dot{x}_k$ and $\dot{x}_y$ represent the $\dot{x}$-axis and $\dot{y}$-axis values of the transformed point p_k or $\dot{p}_k$ in Equation (1) [47]. Also, the angle between the x-axis and reference line is symbolised with the θ term and $arctan((y_t - y_s)/(x_t - x_s))$ trigonometric operation yields its value.

$$\begin{bmatrix} \dot{x}_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} \times \begin{bmatrix} x_k - x_s \\ y_k - y_s \end{bmatrix} \tag{1}$$

The major effect of the coordinate system exchange is seen on the calculation of point $\dot{p}_i$ that is the transformed counterpart of the point p_i found on the line L_i of the set L. Because of the lines in the set L are vertical to the reference line, and they pass through the segmentation points equally dividing the reference line, the $\dot{x}$-axis value of the point $\dot{p}_i$ becomes also the same as the $\dot{x}$-axis value of the transformed counterpart of the *ith* segmentation point, and it is simply found as the result of $i|P_sP_t|/(D+1)$ operation. If only $\dot{y}$-axis values of the points are predicted and stored in a set such as $\{\dot{y}_s, \dot{y}_1, \dot{y}_2, \ldots, \dot{y}_{D-1}, \dot{y}_D, \dot{y}_t\}$, a path can be defined by using the previously calculated $\dot{x}$-axis values and the guessed $\dot{y}$-axis values of the points. A visual inspection of how the coordinate system transformation changes the obtainment procedure of the set P can be carried out with Fig. 1.

The points being selected from the lines of the set L are in the continuous search space, and ignorable changes on the decisions can trigger tremendous modifications on the corresponding path. For making a discrimination between the paths and deciding which one is more appropriate, a cost or score requiring minimisation should be assigned to each path. When calculating the cost or quality score to a path, Equation (2) can be guided. In Equation (2), the cost stemming from the fuel consumption is represented with the C_f, the C_t and C_s are used on behalf of the costs stemming from being sensed by the enemy threats and violations of the turning manoeuvres. While the weight of the C_f on the overall path cost or C is adjusted with the λ_f coefficient, the weight of the C_t and the weight of the C_s are calibrated with the λ_f and λ_s coefficients. Finally, the sum of all three coefficients must be equal to 1.

$$C = \lambda_f C_f + \lambda_t C_t + \lambda_s C_s = \lambda_f \int_0^{\ell} w_f d\ell + \lambda_t \int_0^{\ell} w_t d\ell + \lambda_s \sum_{j=1}^{D} \emptyset_j \tag{2}$$

The integration of the w_f term from 0 to the ℓ showing the length of the path can be purified while keeping the calculation accuracy at the desired level. If the w_f term is replaced with a positive integer constant such as 1, the value of the C_f becomes equal to the ℓ and increases proportionally to the length of the path. Another purification can also be made for the integration of the w_t term. Given that p_i and p_j are two consecutive points in the set P and L_{ij} is the length of line segment between them. Furthermore, the line segment of length L_{ij} is divided equally into ten partitions and the first, third, fifth, seventh and ninth partition points are marked for a subsequent usage. If the line segment between the p_i and p_j is in the effect or sensing range of the *kth* enemy threat, the cost of being sensed by the *kth* enemy threat or $C_{t,(ij),k}$ is determined with Equation (3) [47]. In Equation (3), t_k shows the threat grade or danger level. The Euclidean distances between the marked partition points and the centre of the *kth* enemy threat are represented as the $d^4_{0.1,i,k}$, $d^4_{0.3,i,k}$, $d^4_{0.5,i,k}$, $d^4_{0.7,i,k}$ and $d^4_{0.9,i,k}$ terms. After checking each line segment for all the enemy threats and summing their partial costs, the value of the C_t is estimated.

$$C_{t,(ij),k} = \frac{L_{ij} t_k}{5} \left(\frac{1}{d^4_{0.1,i,k}} + \frac{1}{d^4_{0.3,i,k}} + \frac{1}{d^4_{0.5,i,k}} + \frac{1}{d^4_{0.7,i,k}} + \frac{1}{d^4_{0.9,i,k}} \right) \tag{3}$$

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

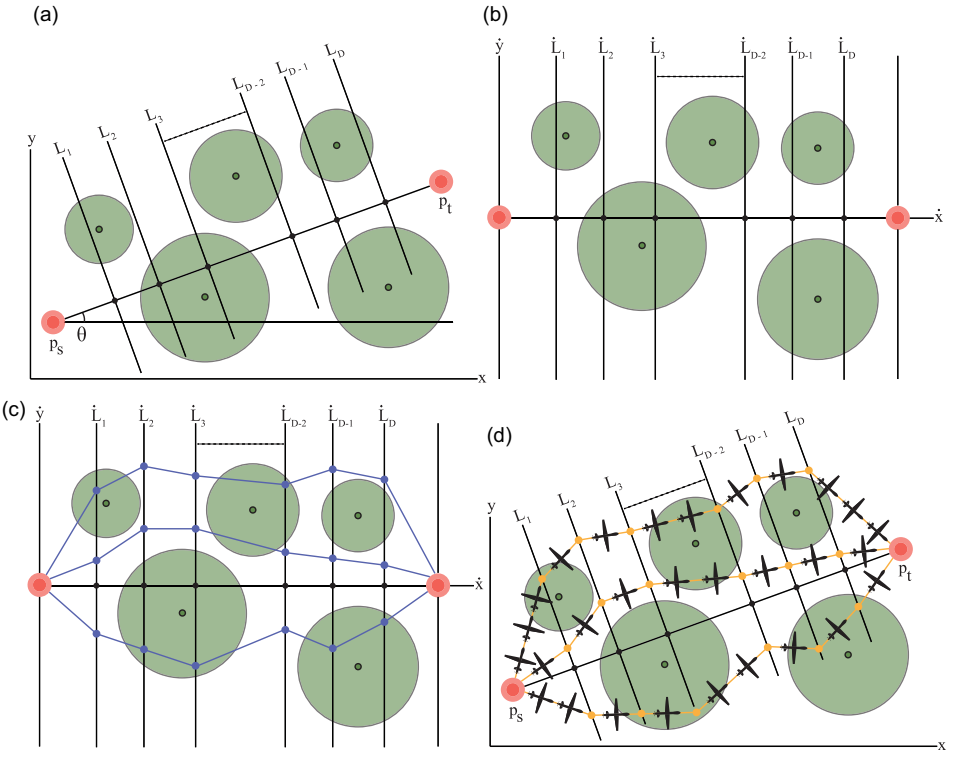


Figure 1. *Original coordinate system for a battlefield with four enemy threats (a), its conversion to the new coordinate system (b), different paths in the new coordinate system (c) and their equivalences in the original coordinate system (d).*

A path being tracked by a UAV or UCAV system necessitates a set of turning manoeuvres successfully [70, 71]. However, a UAV or UCAV has obvious kinematic limitations and some of the turning manoeuvres to track the path thoroughly can not be handled [72]. Assume that p_j, p_{j+1} and p_{j+2} are three successive points in the set P and $\overleftarrow{p_j p_{j+1}}$ and $\overleftarrow{p_{j+1} p_{j+2}}$ are two vectors drawn by referencing these points. If the first three points are considered, p_j, p_{j+1} and p_{j+2} correspond to the p_s, p_1 and p_2 elements. Similarly, if the last three points are considered, p_j, p_{j+1} and p_{j+2} correspond to the p_{D-1}, p_D and p_t, respectively. When the absolute value of the turning angle A_j or $|A_j|$ found as in Equation (4) is less than the maximum turning angle or $\emptyset_{max}$, it is accepted that the turning manoeuvre with the A_j degrees is within the limitations of the UAV system. Otherwise, the cost stemming from the turning with the A_j degrees is added to the existing value of the C_s [47].

$$A_j = arctan\left(\frac{\| \overleftarrow{p_j p_{j+1}} \times \overleftarrow{p_{j+1} p_{j+2}} \|}{\overleftarrow{p_j p_{j+1}} \cdot \overleftarrow{p_{j+1} p_{j+2}}}\right) and\ \emptyset_j = |A_j|, |A_j| > \emptyset_{max} 0, otherwise \tag{4}$$

3.0 Immune Plasma Algorithm

The COVID-19 pandemic taught the whole world that a medical method called immune or convalescent plasma treatment is still influential for helping critical patients. Even though the first practical usage of the immune or convalescent plasma treatment dates back more than a century ago and is associated with the 1918 Great Influenza, patients infected with MERS, Ebola, SARS and H1N1 have also tried to be cured by applying the plasma treatment. The idea biologically strong and ahead of its time lying behind the immune plasma treatment is transferring antibodies from a person recovered shortly before to

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

a patient. Antibodies that are actually protein structures specifically synthesised by the immune system cells can float in the blood for binding to the invaders triggering the infection and healing a critical patient can be supported with the transfer of the donated blood or antibody-rich part of the blood, also known as plasma.

The fundamental steps of the plasma treatment perfectly conform to two main stages namely exploration and exploitation of a meta-heuristic. Aslan and Demirci analysed the mentioned conformation and IP algorithm or IPA that brought intelligent optimisation techniques to the literature [7, 8]. IP algorithm sets an analogy between a solution of the problem and an individual or member of its population. The objective function of the problem is used to determine the immune response or antibody level of an individual. When the IPA decides to treat an individual labelled as a receiver because of poor quality or weak immune response, it first selects one of the top-qualified individuals from the population as a donor and then the plasma transfer from the donor to the receiver individual is conducted by guiding a mathematical model. Detailed explanations of IPA about the distribution of infection between the population members, selection of receivers and donors, implementation of plasma treatment and modification of donor individuals are given in the following subsections.

3.1 Generating individuals of population

An individual of population in IP algorithm is represented with a D-dimensional vector. Each element of the D-dimensional vector corresponds to a decision parameter of the optimisation problem with D different decision parameters. When the first value of the *jth* parameter for the *ith* individual in the population of size PS or x_{ij} is determined, a simple calculation process described with Equation (5) is driven [7]. In Equation (5), x_j^{min} and x_j^{max} are the lower and upper bounds of the *jth* parameter and *rand* (0, 1) is replaced with a random fractional number generated between 0 and 1.

$$x_{kj} = x_j^{min} + rand(0,1)\left(x_j^{max} - x_j^{min}\right) \tag{5}$$

3.2 Infecting individuals of population

A pathogen causing an infection can spread from one individual to another. For modelling the spreading or distribution of infection, IP algorithm selects two individuals. While one of them is infected, the other is accepted as susceptible [7]. Given that x_k is the *kth* individual of the population and x_m is another individual chosen randomly from the same population. For infecting the x_k individual due to the sudden encounter with the x_m, IPA offers the usage of Equation (6) [7]. While x_{kj}^{inf} represents the randomly determined *jth* parameter of x_k^{inf} and x_k^{inf} is matched with the infectious counterpart of x_k, x_{kj} and x_{mj} represent the *jth* parameters of x_k and x_m individuals, respectively. Finally, $rand(-1, 1)$ is a random fractional number between -1 and 1. As easily understood from the mentioned equation, IPA models the distribution of infection by changing only one parameter of the x_k individual and x_k^{inf} actually same as x_k except the newly calculated *jth* parameter.

$$x_{kj}^{inf} = x_{kj} + rand(-1,1)(x_{kj} - x_{mj}) \tag{6}$$

When the x_k individual meets with the infection, its immune system gives a response by synthesising antibodies. For measuring how strong the immune response of the x_k individual is, IPA utilises from the objective function f of the optimisation problem and the immune response of the x_k individual to the infection or synthesised antibody amount is measured as $f\left(x_k^{inf}\right)$ [7]. If the immune response of the x_k individual against to the infection is better than the immune response of the same individual before the infection calculated as $f\ (x_k)$, the immune memory of x_k individual is powered by changing x_{kj} with x_{kj}^{inf}. Otherwise, x_k individual protects the value of *jth* parameter as easily seen from Equation (7) for a minimisation problem [7].

$$x_{kj} = x_{kj}^{inf}, if\ f(x_k^{inf}) < f(x_k) x_{kj}, otherwise \tag{7}$$

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

3.3 Treating critical individuals

IP algorithm continues to distribute infection until all of *PS* individuals are affected. While some of the individuals are capable of handling the infection because of their strong immune systems, some of them require intense care or even the help of other individuals who recovered shortly before. For deciding who are capable of handling the infection and who requires a treatment, IPA introduces two control parameters called *NoD* and *NoR* [7]. The *NoD* parameter gets its name from the word group *number of donors*, and the best *NoD* individuals are assumed to have already recovered and are ready to become donors [7]. On the other hand, the *NoR* parameter gets its name from the word group *number of receivers* and the worst *NoR* individuals are labelled as receivers and tried to be treated with the plasma donations [7].

Given that x_k^{rcv} is the *kth* receiver from the set of receivers containing *NoR* individuals and x_m^{dnr} is the randomly selected *mth* donor from the set of donors containing *NoD* individuals. In order to represent the transfer of the first plasma dose from the x_m^{dnr} to the x_k^{rcv}, IPA executes Equation (8) where j is chosen sequentially from the set $\{1, 2, \ldots, D\}$ [7]. In Equation (8), x_{kj}^{rcv-p} shows the *jth* parameter of x_k^{rcv-p} and x_k^{rcv-p} is used on behalf of the plasma transferred counterpart of the x_k^{rcv} individual. Also, x_{kj}^{rcv} and x_{mj}^{dnr} are matched with the *jth* parameters of the x_k^{rcv} and x_m^{dnr} individuals. If the immune response of the x_k^{rcv} immediately after the first dose of plasma also measured as $f\left(x_k^{rcv-p}\right)$ is better than the immune response of donor individual measured as $f\left(x_m^{dnr}\right)$, x_k^{rcv} is updated with the parameters of x_k^{rcv-p} appropriately. Moreover, the preparation for the transfer of second dose begins. Otherwise, x_k^{rcv} is updated with the parameters of x_m^{dnr} and treatment is simply concluded [7].

$$x_{kj}^{rcv-p} = x_{kj}^{rcv} + rand(-1, 1)(x_{kj}^{rcv} - x_{mj}^{dnr}) \tag{8}$$

The second dose of plasma from the x_m^{dnr} is transferred to the x_k^{rcv} as described in Equation (8). However, the decision about whether the treatment will be continued with the third dose of plasma or not is made in a different manner when compared to the previous case. If the immune response of the x_k^{rcv} after the second dose of plasma is better than the immune response of the current x_k^{rcv}, a procedure modifying the parameters of x_k^{rcv} with the parameters of x_k^{rcv-p} is executed and then the treatment is maintained with the third dose of plasma [7]. Otherwise, the treatment is completed for the x_k^{rcv}, and next receiver if it exists is taken to start similar plasma transfer operations.

3.4 Modification of donor individuals

If an individual who recovered previously encounters the same infection frequently as time goes by, their immune memory can be specialised more to recognise the pathogen causing infection. As an expected result of this type of specialisation, the required amount of antibodies is synthesised immediately. In contrast, recognising the pathogen and synthesising the required amount of antibodies by the immune system of the considered individual takes longer. In the IP algorithm, only the individuals chosen as donors are assumed as recovered and their immune memories are specialised fully or partially according to a probabilistic mechanism. The probabilistic mechanism responsible for deciding which type of specialisation procedure is applied compares the ratio of t_{cr} and t_{max} with a random number between 0 and 1 [7]. While the t_{cr} shows the current evaluation number, the t_{max} represents the maximum evaluation number. If the ratio of t_{cr} and t_{max} is less than the random number, the donor individual is regenerated by using Equation (5). Otherwise, each parameter of the donor individual is changed somewhat as formulated in Equation (9) [7].

$$x_{mj}^{dnr} = x_{mj}^{dnr} + rand(-1, 1)x_{mj}^{dnr} \tag{9}$$

When the details about the model for specialising the immune memories of the donors are investigated, it is seen that the probability of updating immune memories partially with Equation (9) increases, while the run of the IPA reaches termination [7]. Because the ratio of t_{cr} and t_{max} gets close and then becomes equal to 1 at the end of a run, the probability of generating a random number less than t_{cr}/t_{max}

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

and applying a partial update with Equation (9) also increases and immune memories of the donors are preserved more.

4.0 Greedy Initialiser Immune Plasma Algorithm

The geometric description of the path planning problem requires intense calculations to do with the lines of the set L and their counterparts for the transformed coordinate system. However, all of these lines provide critical information about the battlefield and can be further referenced for generating an initial path or paths being used by a population-based meta-heuristic algorithm. Given that $\dot{L}_i$ is the *ith* line from the set $\dot{L}$ and it is divided equally into $D+1$ line segments with D different points similar to the segmentation of the reference line [73]. The points dividing the line $\dot{L}_i$ are also combined into a set such as $\dot{P}^i = \left\{\dot{p}_1^i, \dot{p}_2^i, \ldots, \dot{p}_{D-1}^i, \dot{p}_D^i\right\}$. If the first line is taken into account and a point from the point set $\dot{P}^1$ will be selected for a UAV or UCAV path, a greedy heuristic in which a point that is the nearest one to the $\dot{p}_s$ and outside the effect range of all enemy threats will be selected is employed [73]. When the line $\dot{L}_i$ where i is greater than 1 is considered and a point from the point set $\dot{P}^i$ will be selected for a path, the mentioned greedy approach in which a point that is nearest to the taken point in the set $\dot{P}^{i-1}$ of line $\dot{L}_{i-1}$ and outside the effect range of all enemy threats is chosen again [73]. The formulations for the selection of the first and other points by the heuristic method are given in Equations (10) and (11), respectively.

$$\dot{p}_1 = \dot{p}_j^1, d(t, \dot{p}_j^1) > r_t \wedge d\left(\dot{p}_s, \dot{p}_j^1\right) < \underset{\forall \dot{p}_m^1 \neq \dot{p}_j^1 \in \dot{P}^1}{d(\dot{p}_s, \dot{p}_m^1)}\, \dot{p}_1^1, otherwise \tag{10}$$

$$\dot{p}_i = \dot{p}_j^i, \underset{\forall t \in T}{d(t, \dot{p}_j^i)} > r_t \dot{p}_k^i, \underset{\forall t \in T}{d\left(t, \dot{p}_k^i\right)} > r_t \wedge d\left(\dot{p}_{i-1}, \dot{p}_k^i\right) < \underset{\forall \dot{p}_m^i \neq \dot{p}_k^i \in \dot{P}^i - \left\{\dot{p}_j^i\right\}}{d\left(\dot{p}_{i-1}, \dot{p}_m^i\right)}\, first\ of \dot{P}^i - \left\{\dot{p}_j^i\right\}, otherwise \tag{11}$$

The fundamental idea lying behind the introduced greedy approach is keeping the distance between the subsequent points and manoeuvre costs related to the turnings at minimum. Also, while the path safety is guaranteed partially when the line segment between two selected points intersects with the circles representing the enemy threats even though the selected points are outside the circles or fully when the line segment between two selected points does not intersect with any circles [73]. Because of this reason, the first point being controlled from the set $\dot{P}^i$ should be $\dot{p}_j^i$, if the point $\dot{p}_j^{i-1}$ in the set $\dot{P}^{i-1}$ where i is greater than 1 is chosen. If the point $\dot{p}_j^i$ is outside the effect range of all enemy threats, it is taken quickly without controlling other segmentation points in the set $\dot{P}^i$ of line $\dot{L}_i$. In Fig. 2, a graphical example of how the points are selected by the heuristic method is given. When the selection of the point from the line $\dot{L}_i$ concludes with the point $\dot{p}_j^i$, it should be noted that the length of line segment between $\dot{p}_j^{i-1}$ and $\dot{p}_j^i$ is minimised and safety of UAV or UCAV for this line segment is ensured. In addition to this, a turning manoeuvre is not carried out by considering that the point $\dot{p}_j^{i-2}$ from the line $\dot{L}_{i-2}$ is chosen. Otherwise, the remaining points in the set $\dot{P}^i$ are investigated for finding a point that is closest to the $\dot{p}_j^{i-1}$ and outside the effect range of all enemy threats.

A path generated by the greedy heuristic can be referenced to populate individuals of the IP algorithm. For maintaining the information provided by the greedy approach and satisfying the required population diversity, the individuals of IPA is created in a manner that a randomly determined one is equalised to the path of greedy heuristic and each remaining member of the population is created by adding randomly produced number between -1 and 1 to each parameter of the path inherited from the greedy heuristic. By slightly modifying the inherited path when generating the $PS-1$ members of the population, the new path candidates without executing compute intensive operations are discovered. Subsequently, an initial population that contains the promising solutions is passed to the IP algorithm for the following phases or stages.

The population generated as detailed above and passed to the IP algorithm should be exploited more carefully for converging to the optimum solution or solutions. When the properties of the IP algorithm

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

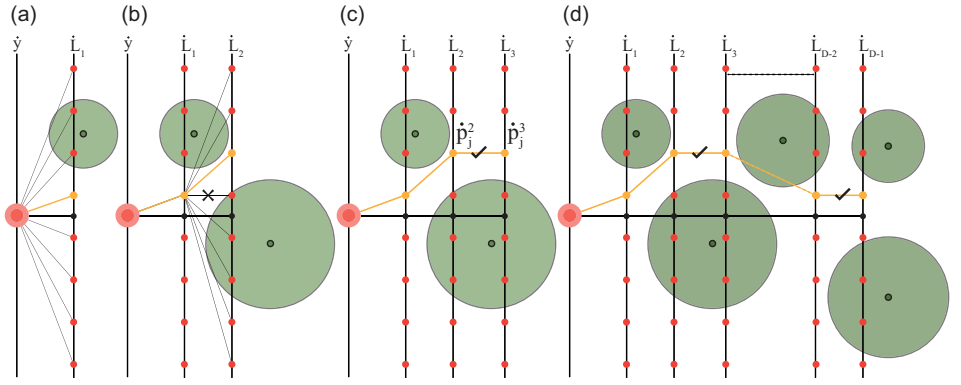


Figure 2. *Selecting the points from the lines $\dot{L}_1$ (a), $\dot{L}_2$ (b), $\dot{L}_3$ (c) and $\dot{L}_{D-1}$ (d) according to the greedy heuristic.*

are investigated, it is seen that the fundamental phase related to the plasma treatment maintains the exploitation dominant operations. However, IPA uses a generalised model for providing an algorithmic simplicity by skipping some subtle parts about the acquisition of plasma. If the IP algorithm is supported with an extra mechanism that models the acquisition of the plasma and its transfer to a receiver, the vicinity of the promising solutions in the population is searched particularly. Given that x^b is the best solution found by the algorithm until the current evaluation and x_1^{dnr} and x_2^{dnr} are two donors chosen from the existing population. For modeling the acquisition or collection of the plasma that is represented as x^p and initialised with the x^b, Equation (12) given below can be served. In Equation (12), x_j^{p-t} is the newly calculated *jth* parameter of x^p with the help of *jth* parameters of x_1^{dnr} and x_2^{dnr}. When the newly calculated *jth* parameter of x^p improves the solution quality, a replacement between x_j^p and x_j^{p-t} is approved. After controlling all the D parameters and updating them if required, x^p becomes ready for the transfer.

$$x_j^{p-t} = x_1^{dnr} + rand(-1, 1)\left(x_1^{dnr} - x_2^{dnr}\right) \tag{12}$$

The quality of the solution represented with the x^p is equal to or most likely better than the quality of the solution represented with the x^b and its transfer to a receiver should be carried out by considering the conservation of x^p or a better candidate for utilising it in future cycles. Assume that x^{rcv} is the most critical individual of the population. The transfer of plasma or x^p to the x^{rcv} individual can be described as in Equation (13). In Equation (13), x_j^{rcv-p} is used on behalf of the *jth* parameter of x^{rcv-p} and x^{rcv-p} shows the plasma transferred counterpart of the x^{rcv} individual. While x_j^{rcv} and x_j^p are the *jth* parameters of x^{rcv} and x^p, j_r is a random integer between 1 and D. Since only one parameter index will be equal to the j_r, x^{rcv-p} directly configures its $D-1$ different parameters with the corresponding parameter values of x^p. If the objective function value of x^{rcv} immediately after the treatment or $f\left(x^{rcv-p}\right)$ is better than the objective function value of the same individual before the treatment or $f\left(x^{rcv}\right)$, x^{rcv} is updated with the x^{rcv-p} and treatment is ended. Otherwise, x^{rcv} is updated with the x^p and then treatment is concluded.

$$x_j^{rcv-p} = x_j^p, if\ j \neq j_r\ x_j^{rcv} + rand(-1, 1)\left(x_j^{rcv} - x_j^p\right), otherwise \tag{13}$$

The details of the proposed treatment schema that models how the plasma is collected and then transferred also give some clues about the changed requirement trend of the *NoR* and *NoD* control parameters of the IPA. As stated previously, the proposed treatment schema selects two donors from the population and transfers the collected plasma to the most critical individual. By considering this preset for the donors and receivers, there is no further need to the *NoR* and *NoD* control parameters and their sensitive configurations according to the population size, number of parameters and specific properties of the problem being solved optimally. The operational steps about how the plasma is collected first and transferred in the renovated treatment schema can be examined by guiding Algorithm 1. The IP algorithm, where the greedy heuristic is used to initially generate the population and the proposed

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

Algorithm 1. Details of the treatment schema in gintIPA

$x^b \leftarrow$ get the best solution found so far
$x^p \leftarrow$ assign a copy of x^b for required initialisation
$x_1^{dnr}, x_2^{dnr} \leftarrow$ select the first and second best individuals as donors
/ Plasma acquisition with the help of selected donors */*
for $j \leftarrow 1 \ldots D$ **do**
    if $t_{cr} < t_{max}$ **then**
        $t_{cr} \leftarrow t_{cr} + 1$
        $x_j^{p-t} \leftarrow$ calculate new value for the *jth* parameter of x^p with Equation (12)
        if $f\left(x^p\right) < f\left(x^b\right)$ **then**
            Replace x^b with x^p
        else
            Convert the *jth* parameter of x^p into its previous value
        end if
    end if
end for
/ Transfer of plasma to the most critical individual */*
if $t_{cr} < t_{max}$ **then**
    $t_{cr} \leftarrow t_{cr} + 1$
    $x^{rcv} \leftarrow$ get the most critical individual
    $j_r \leftarrow$ a random integer between 1 and D
    for $j \leftarrow 1 \ldots D$ **do**
        if $j \neq j_r$ **then**
            $x_j^{rcv-p} \leftarrow$ assign a copy of x_j^p by guiding Equation (13)
        else
            $x_j^{rcv-p} \leftarrow$ find a new value by guiding Equation (13)
        end if
    end for
    if $f\left(x^{rcv-p}\right) < f\left(x^p\right)$ **then**
        Update x^{rcv} with x^{rcv-p} and conclude the treatment
    else
        Update x_k^{rcv} with x^p and conclude the treatment
    end if
    if $f\left(x^{rcv}\right) < f\left(x^b\right)$ **then**
        Replace x^b with x^{rcv}
    end if
end for

treatment scheme is responsible for the collection and transfer of plasma, is called greedy initialisation IPA for short gintIPA[1].

5.0 Experimental Studies

The performance of the gintIPA can vary with the properties of the battlefield in terms of number of enemy threats, their centre locations, sensing or detecting ranges determined by radius, danger levels or grades. For evaluating how the performance of gintIPA varies, three battlefield scenarios detailed in

[1]Source code is available upon request.

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

Table 1. *Information about the battlefields used in the experiments*

Sc.	Threat Centres	Threat Radius	Threat Grade	Start-Target Point
1	(12, 48), (24, 33), (27, 58), (30, 70), (55, 80), (59, 52), (70, 34), (70, 65)	12, 9, 9, 10, 9, 10, 12, 7	1, 12, 3, 2, 7, 9, 13, 5	(10, 15) (80, 75)
2	(20, 70), (25, 19), (25, 39), (45, 20), (47, 41), (50, 61), (70, 53), (75, 74), (78, 20)	20, 9, 9, 9, 9, 9, 9, 9, 20	7, 5, 5, 5, 5, 5, 5, 5, 7	(5, 5) (95, 95)
3	(10, 50), (20, 20), (30, 42), (30, 80), (50, 55), (60, 10), (60, 80), (65, 38), (75, 65), (90, 80)	10, 9, 8, 10, 10, 10, 10, 11, 8, 10	8, 6, 5, 4, 7, 6, 7, 6, 8, 10	(10, 0) (80, 100)

Table 2. *Results of gintIPA with varying PS values for Scenario-*1

		PS				
D		30	40	50	75	100
10	Best	38.366	38.406	38.377	38.363	38.409
	Worst	38.459	38.504	38.458	38.460	38.443
	Mean	38.403	38.436	38.418	38.419	38.433
	Std.	0.040	0.031	0.034	0.038	0.009
15	Best	38.279	38.267	38.279	38.300	38.375
	Worst	38.300	38.313	38.373	38.420	38.656
	Mean	38.290	38.287	38.321	38.355	38.437
	Std.	0.008	0.019	0.038	0.050	0.079
20	Best	38.267	38.267	38.268	38.393	38.433
	Worst	38.360	38.320	38.539	38.548	38.731
	Mean	38.298	38.293	38.434	38.464	38.549
	Std.	0.033	0.023	0.078	0.055	0.100
25	Best	38.308	38.385	38.390	38.612	38.694
	Worst	38.633	38.754	38.809	39.730	45.168
	Mean	38.458	38.518	38.595	39.013	40.216
	Std.	0.088	0.115	0.166	0.429	2.526

Table 1 were used [47]. Each battlefield contains four test cases generated by configuring the number of segmentation points or D as 10, 15, 20 and 25. The gintIPA was experimented with 30 times for each test case with the different PS values including 30, 40, 50, 75 and 100. When completing a run or test after consuming 6000 evaluations, the best solution was found and its objective function value was recorded. The best, worst, mean best objective function values and standard deviations calculated over 30 records were presented in Table 2 for the cases of Scenario-1, Table 3 for the cases of Scenario-2 and finally Table 4 for the cases of Scenario-3.

The results given in Tables 2, 3 and 4 provide critical information about the relatively consistent solving capabilities of the gintIPA. The population or colony size of a meta-heuristic algorithm can impact the quality of the final solutions. If the population or colony size is chosen high, the solution space is sampled more and the algorithm gets a chance of starting the subsequent operations with a set of qualified candidates. However, it should be noticed that the number of evaluations per cycle or iteration increases and a run of the algorithm terminates without executing specially designed operations enough. On the other hand, if the population or colony size is chosen small, the probability of starting a run of a meta-heuristic with a set of qualified initial candidates decreases. Even though the probability

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

Table 3. *Results of gintIPA with varying PS values for Scenario-2*

		PS				
D		30	40	50	75	100
10	Best	57.752	60.448	58.145	54.783	60.759
	Worst	60.777	63.807	61.370	63.377	63.956
	Mean	59.762	61.821	60.042	59.161	61.950
	Std.	1.344	1.547	1.380	3.262	1.444
15	Best	54.093	54.110	54.100	54.147	54.205
	Worst	56.274	56.378	54.422	58.826	56.424
	Mean	54.296	54.742	54.203	54.892	55.770
	Std.	0.539	1.005	0.088	1.306	0.954
20	Best	53.750	53.704	53.732	53.706	53.729
	Worst	57.270	53.867	53.868	53.891	53.997
	Mean	54.581	53.789	53.778	53.781	53.814
	Std.	1.509	0.060	0.048	0.055	0.105
25	Best	53.581	53.632	53.671	53.762	53.711
	Worst	53.785	53.761	53.740	53.878	54.770
	Mean	53.705	53.670	53.712	53.809	53.990
	Std.	0.071	0.048	0.025	0.039	0.362

Table 4. *Results of gintIPA with varying PS values for Scenario-3*

		PS				
D		30	40	50	75	100
10	Best	49.783	49.776	49.777	49.777	49.774
	Worst	49.814	49.865	49.832	49.795	49.857
	Mean	49.796	49.800	49.796	49.785	49.807
	Std.	0.013	0.025	0.026	0.008	0.039
15	Best	49.755	49.759	49.755	49.759	49.757
	Worst	49.772	49.773	49.767	49.778	49.785
	Mean	49.765	49.765	49.761	49.772	49.770
	Std.	0.007	0.006	0.004	0.006	0.010
20	Best	49.816	49.809	49.791	49.816	49.821
	Worst	49.875	49.849	49.826	49.824	49.869
	Mean	49.848	49.825	49.809	49.819	49.853
	Std.	0.024	0.016	0.015	0.003	0.016
25	Best	49.848	49.808	49.807	49.876	49.903
	Worst	49.951	49.915	49.913	49.955	49.976
	Mean	49.888	49.860	49.866	49.906	49.940
	Std.	0.041	0.038	0.031	0.028	0.025

of initialising the population with qualified candidates decreases, the considered algorithm iterates more compared to the previously mentioned configuration about population or colony size. From the mean best objective function values in Tables 2, 3 and 4, it is seen that gintIPA protects the qualities of final solutions while the different values are assigned to the *PS* parameter. One of the main reasons behind the stable performance of gintIPA is related to the employed greedy initialisation method that works by guiding only number of segmentation points. When the greedy initialiser determines its path, one of the individuals in the gintIPA is configured with the determined path, and the remaining members of the

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

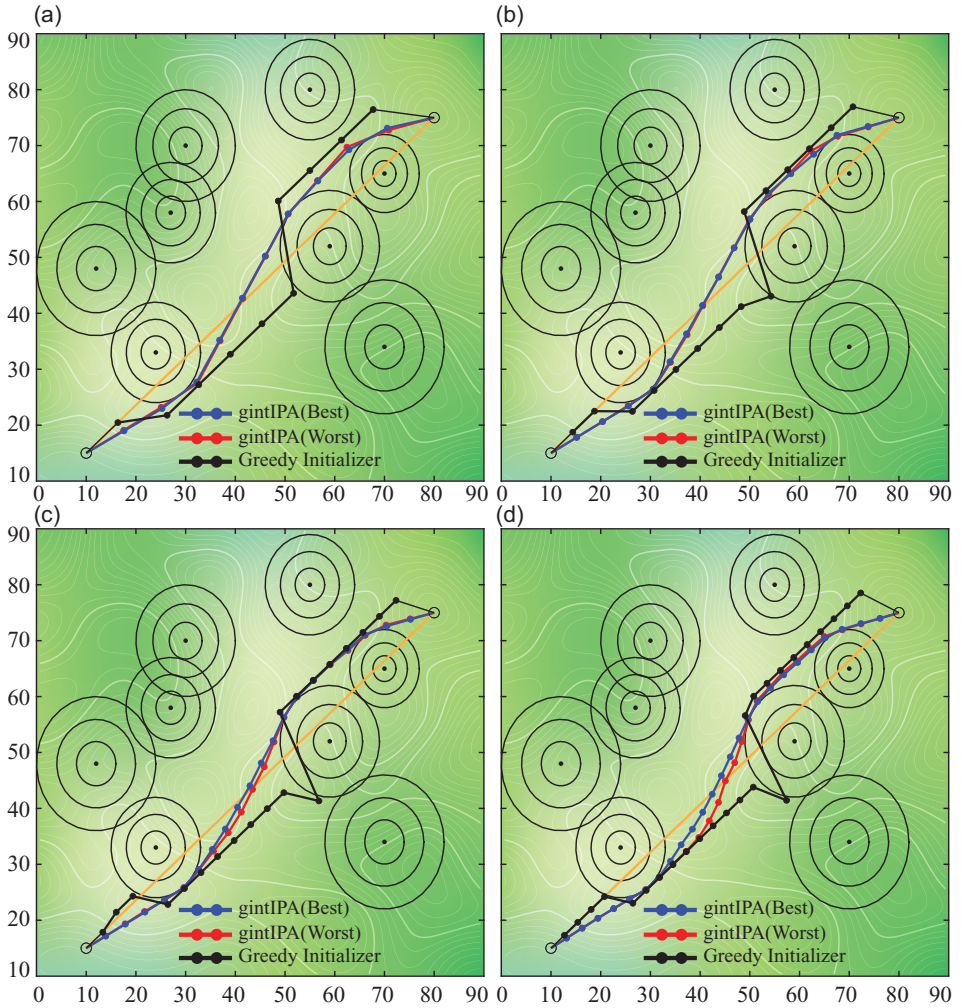


Figure 3. *The best and worst paths found by gintIPA for* 10 *(a),* 15 *(b),* 20 *(c) and* 25*-dimensional (d) cases of Scenario*-1.

population are also produced randomly in a narrow neighbourhood of the path of the greedy initialiser. By generating the population of gintIPA with the help of a greedy initialiser, algorithm starts its phases about the distribution of infection, plasma treatment and modification of the donor or donors with the promising population members. Moreover, it is important to declare that if the *PS* parameter of gintIPA is set to a higher value, an intrinsic advantage about the searching the neighbourhood of the qualified path determined by greedy initialiser is gained and gintIPA can protect the qualities of its solutions for the running configurations causing the huge amount of evaluation consumption per cycle.

The stable performance of gintIPA for different values assigned to the *PS* parameter is also related to the modernised treatment schema that combines both newly introduced plasma collection mechanism and transfer approach and removes the necessity of the inherited control parameters from IPA called *NoR* and *NoD*. When the gintIPA decides to treat the most critical individual of the population, it first collects the plasma. The plasma is actually a new solution generated by guaranteeing its objective function value is at least equal to or less than the objective function value of the best solution found so far. Also, the transfer approach used for the plasma performs an extra search and tries to find a solution better than the solution represented by the plasma. As an expected result of the detailed exploitation of the eligible

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

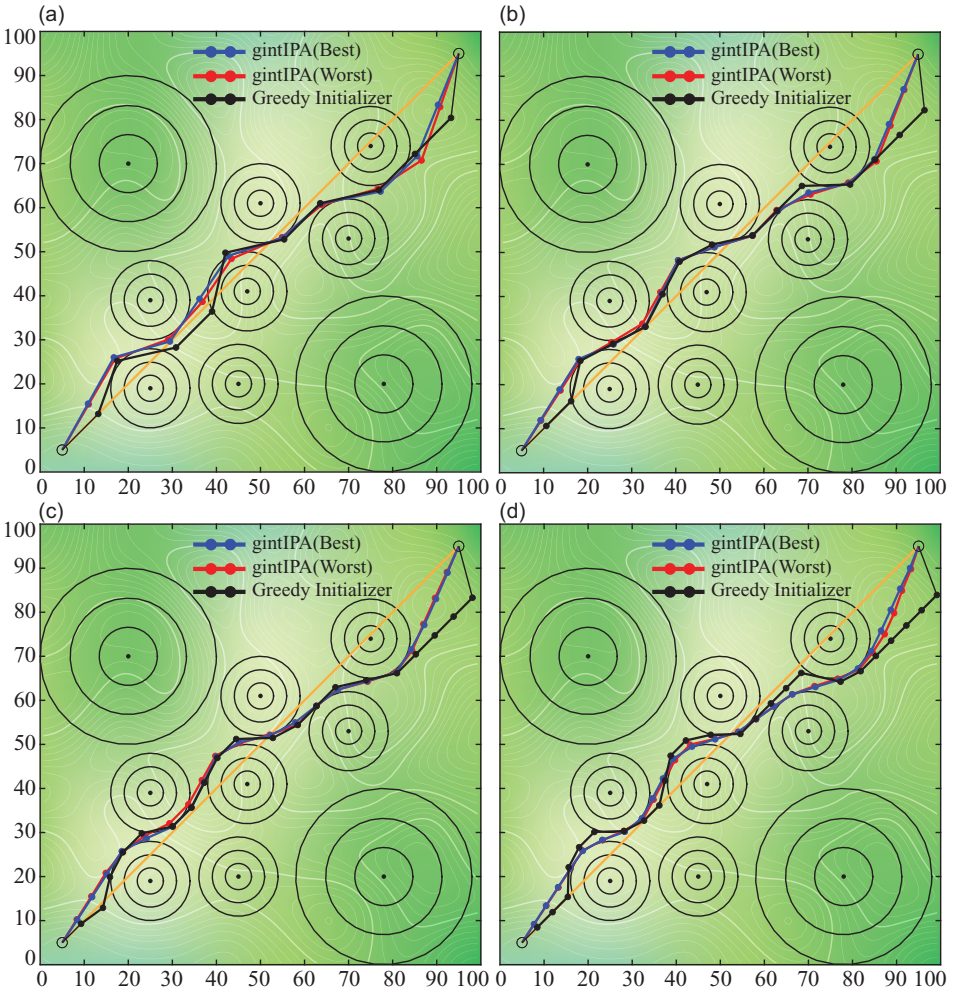


Figure 4. *The best and worst paths found by gintIPA for* 10 *(a),* 15 *(b),* 20 *(c) and* 25*-dimensional (d) cases of Scenario-*2*.*

solutions in the population for the collection of plasma and subtle search when transferring it, gintIPA handles some problems originating from the quick termination without executing treatment operations adequately. The small differences between the best and worst paths found by gintIPA with 30 individuals can be visually investigated over Fig. 3 for Scenario-1, Fig. 4 for Scenario-2 and Fig. 5 for Scenario-3.

In order to understand how successful the gintIPA is as a path planner and how eligible the calculated paths by gintIPA are, a set of comparative studies should be carried out. For this purpose, the paths found by the gintIPA were compared with the paths of fourteen different meta-heuristic techniques including IPA, GA, GAPSO, MFO, SSA, PFA, SBO, SCA, ECTLBO, HSGWO-MSOS, CIPSO, GWO, AEO and NSEAEO whose control parameters are set as in Table 5. Each test case belonging to the Scenario-1, Scenario-2 and Scenario-3 was solved 30 times after equalising the running conditions by setting the population or colony size to 30 and maximum evaluation number to 6000 for the gintIPA and other mentioned algorithms [47, 63]. When the results given in Tables 6, 7 and 8 are investigated, the competitive performance of gintIPA against other path planners can be seen clearly. While the gintIPA outperforms all of other techniques for 11 of 12 test cases and gets ranked as the best path planner, it lags behind IPA and NSEAEO only for one test case belonging to the Scenario-2 and gets ranked as the third best path planner and allows to state that gintIPA is still capable of obtaining better solutions than remaining

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

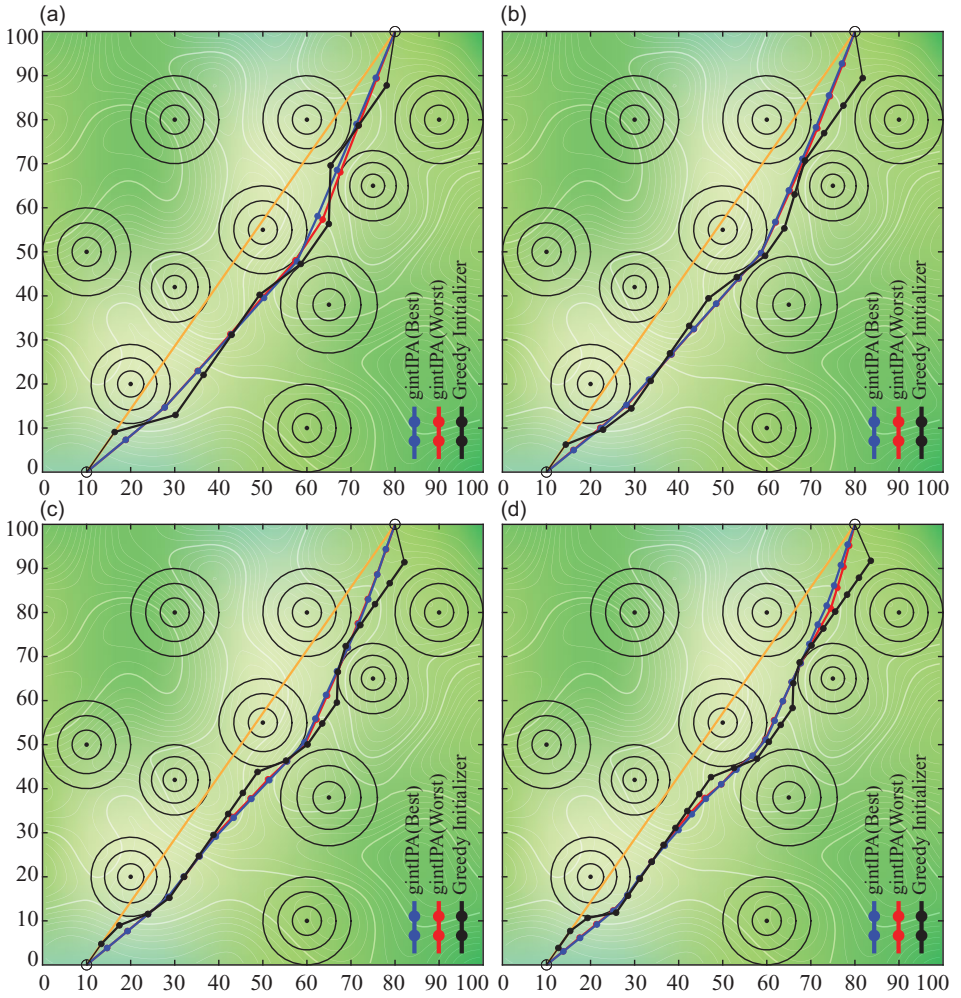


Figure 5. *The best and worst paths found by gintIPA for* 10 *(a),* 15 *(b),* 20 *(c) and* 25*-dimensional (d) cases of Scenario-*3*.*

12 algorithms. The visually apparent contribution of the greedy initialiser as validated from Figs 3, 4 and 5 and improved exploitation performance with the newly designed treatment schema allow gintIPA to perform better than the considered algorithms in 91.6% of all test cases. The comparative studies between gintIPA and other path planners also provide important information about the advantageous sides of gintIPA becoming more evident with the increasing number of segmentation points. When the number of segmentation points and difficulty of the path planning problem are increased, the greedy initialiser of gintIPA can determine more points on the path reducing the cost of enemy threats, fuel or battery consumption or turning manoeuvres. In addition to this, when the number of segmentation points is increased, the chance of collecting more qualified plasma is also increased intrinsically due to the sequential control of all parameters with the help of the selected donors. As an expected result of the self-configurable searching characteristics according to the number of segmentation points, gintIPA handles the difficulties of the path planning problem more robustly and calculates better paths than its competitors.

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

Table 5. *Compared algorithms with gintIPA and their parameter adjustments*

Algorithms	Parameter Configurations
IPA [13]	$NoR = 1, NoD = 1$
GA [47, 63]	$P_c, P_m = \{0.7, 0.3\}$
GAPSO [47, 63]	$P_c, P_m = \{0.7, 0.3\}, c_1, c_2 = \{1.5, 2\}, w = 1$
MFO [47, 63]	$r \in [-1, -2]$
SSA [47, 63]	$c_1 = c_2 = c_3 = rand$
PFA [47, 63]	$\alpha, \beta \in [0, 2]$
SBO [47, 63]	$\alpha = 0.94, z = 0.02, P_m = 0.05$
SCA [47, 63]	$\alpha_0 = 2$
ECTLBO [47, 63]	$r \in [-1, -2]$
HSGWO-MSOS [47, 63]	$c_1 = c_2 = c_3 = and$
CIPSO [47, 63]	$c_{min},\ c_{max} = \{3.5, 0.5\}, V_1, V_2 = \{0.5, 0.1\}, w_{min}, w_{max} = \{0.9, 0.4\}, \mu = 4$
GWO [47, 63]	α *decreasing from* 2 *to* 0
AEO [47, 63]	$r = r_1 = r_2 = rand$
NSEAEO [47, 63]	$w = 1$

The greedy initialiser used with the gintIPA brings an extra computational burden. Also, the modernised treatment schema changes the number of evaluations spent per infection cycle and gintIPA can iterate less than the standard implementation of the IPA even though the termination criteria and other running configurations are the same. In order to analyse that how the modifications change the execution time, gintIPA and IPA were experimented with 30 times by setting the population size to 30 and maximum evaluation number to 6000 for each test case of Scenario-1, Scenario-2 and Scenario-3 and the time passed until the end of a run was recorded. Both gintIPA and IPA were coded in C programming language and a computer equipped with an Intel i5-10500 processor and managed by Fedora-34 operating system was used. The best, worst, mean elapsed times in terms of seconds and standard deviations were determined by using the records and then summarised in Table 9. When the mean execution times in Table 9 are investigated, it is understood that gintIPA requires more time compared to the IPA. While the differences between the mean execution times of the gintIPA and IPA are 0.002 and 0.006 seconds at most for 10 and 15-dimensional cases, the differences between the mean execution times of the gintIPA and IPA are 0.008 and 0.010 seconds at most for 20 and 25-dimensional cases and indicate the direct relationship with the computational complexity of the greedy initialiser and number of segmentation points.

The proposed greedy initialiser and treatment schema can also affect the convergence performance of the algorithm. For evaluating the convergence performance of a meta-heuristic algorithm, the success rate abbreviated as *Sr* and mean evaluations abbreviated as *Me* are referenced frequently [63]. If a run of the considered meta-heuristic concludes with a solution whose objective function value is better than the previously determined threshold, this run is accepted as successful and the ratio between the number of successful runs and the total number of runs corresponds to the *Sr* metric. If a run is determined as successful, the minimum number of evaluations required to find a solution whose objective function value is less than the threshold is stored. The average of the stored values represents the value of the *Me* metric. The *Sr* and *Me* metrics belonging to the gintIPA and IPA by setting the threshold to 50 for the Scenario-1 and Scenario-3 and 60 for the Scenario-2 were calculated and presented in Table 10. The *Sr* and *Me* metric values given in Table 10 prove the superior convergence performance of gintIPA against the IP algorithm. While the *Sr* metric of gintIPA is better than the *Sr* metric of IPA for 9 of 12 test cases, the *Me* metric of gintIPA is better than the metric of IPA for all test cases. However, it should be noted that even though the *Me* metric value of gintIPA is better than the *Me* metric value of IPA for

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

Table 6. *Comparison between gintIPA and other path planners for Scenario-*1

D		gintIPA	IPA	GA	GAPSO	MFO	SSA	PFA	SBO	SCA	ECTLBO	HSGWO-MSOS	CIPSO	GWO	AEO	NSEAEO
10	Best	38.366	38.543	39.647	39.754	38.703	39.241	38.587	38.582	51.367	39.080	38.565	40.103	38.565	38.642	38.506
	Worst	38.459	39.800	149.678	98.436	89.657	122.790	318.996	241.127	109.966	57.912	132.095	245.441	124.542	52.378	46.526
	Mean	38.403	39.315	65.919	61.725	63.210	57.037	88.227	69.774	78.536	45.379	57.994	96.501	63.559	46.562	38.929
	Std.	0.040	0.419	34.300	24.078	15.100	22.882	68.817	52.248	15.700	5.016	20.323	52.917	25.320	2.902	1.526
	Rank	1	3	11	8	9	6	14	12	13	4	7	15	10	5	2
15	Best	38.279	42.791	41.148	42.253	42.814	39.022	38.466	38.402	63.348	51.425	39.068	39.912	38.560	38.302	38.291
	Worst	38.300	49.029	232.634	80.672	266.532	105.756	210.693	203.260	209.946	112.515	115.202	223.591	97.052	58.778	58.042
	Mean	38.290	46.278	81.644	57.736	90.566	56.959	69.610	66.273	117.484	82.711	61.265	92.743	52.706	45.410	39.074
	Std.	0.008	2.439	50.303	13.416	44.985	19.697	45.982	38.427	30.720	17.811	13.564	49.986	15.555	9.140	3.584
	Rank	1	4	11	7	13	6	10	9	15	12	8	14	5	3	2
20	Best	38.267	62.250	49.553	51.378	60.290	44.282	39.543	39.351	66.430	49.978	43.637	42.977	39.083	39.508	38.369
	Worst	38.360	70.632	301.088	118.726	396.167	219.03	145.096	174.852	373.008	211.499	141.557	503.643	97.515	52.182	39.606
	Mean	38.298	67.819	135.674	80.166	158.211	81.580	59.488	65.309	181.787	131.767	67.328	137.171	51.652	43.195	38.769
	Std.	0.033	2.709	59.459	14.844	84.234	40.491	25.292	31.510	71.888	44.952	22.644	93.903	11.872	3.994	0.293
	Rank	1	8	12	9	14	10	5	6	15	11	7	13	4	3	2
25	Best	38.308	92.030	102.967	84.872	84.829	60.426	41.970	47.433	57.273	81.008	43.554	65.411	41.514	41.017	39.299
	Worst	38.633	117.280	439.336	199.769	2246.004	276.890	166.076	137.64	580.501	454.877	167.021	647.234	90.889	51.240	40.108
	Mean	38.458	98.989	173.069	122.905	282.449	122.920	77.936	68.971	252.098	187.488	81.527	205.902	50.307	42.697	39.615
	Std.	0.088	7.803	68.499	25.532	388.776	50.105	30.544	21.100	150.970	100.894	33.750	141.196	11.722	1.995	0.182
	Rank	1	8	11	9	15	10	6	5	14	12	7	13	4	3	2
Average Rank		1.000	5.750	11.250	8.250	12.750	8.000	8.750	8.000	14.250	9.750	7.250	13.750	5.750	3.500	2.000
Overall Rank		1	4	12	9	13	8	10	7	15	11	6	14	4	3	2

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

Table 7. *Comparison between gintIPA and other path planners for Scenario-2*

D		gintIPA	IPA	GA	GAPSO	MFO	SSA	PFA	SBO	SCA	ECTLBO	HSGWO-MSOS	CIPSO	GWO	AEO	NSEAEO
10	Best	57.752	54.888	61.668	61.721	59.158	59.726	57.896	58.135	88.336	60.541	57.869	59.643	61.239	58.590	56.204
	Worst	60.777	58.378	312.343	130.981	270.787	221.180	598.749	525.925	221.805	99.426	109.680	1677.440	120.087	72.425	69.813
	Mean	59.762	56.518	99.630	94.912	90.261	103.382	166.529	107.482	123.145	75.787	67.683	158.779	79.191	65.734	57.474
	Std.	1.344	1.427	47.060	20.382	40.877	48.416	159.441	104.792	26.882	10.033	13.555	292.587	14.446	6.742	2.659
	Rank	3	1	10	9	8	11	15	12	13	6	5	14	7	4	2
15	Best	54.093	57.657	66.193	67.163	60.710	57.056	58.235	57.025	105.006	69.484	57.225	70.080	55.091	54.796	55.949
	Worst	56.274	60.106	551.464	241.5386	670.571	398.873	268.370	230.585	334.849	235.419	201.640	537.313	127.408	75.777	66.180
	Mean	54.296	58.855	162.766	111.416	171.648	131.444	122.142	82.483	184.436	152.439	80.750	230.860	78.095	57.660	60.250
	Std.	0.539	0.958	106.642	47.446	154.733	79.183	68.468	45.101	56.189	36.838	32.343	134.219	17.432	4.269	2.654
	Rank	1	3	12	8	13	10	9	7	14	11	6	15	5	2	4
20	Best	53.750	63.442	79.686	97.730	68.284	61.780	54.954	57.605	104.952	91.201	64.063	93.525	57.437	54.659	54.606
	Worst	57.270	83.848	706.929	227.695	3271.735	369.486	353.765	218.618	842.337	505.444	216.044	2238.837	182.698	84.463	58.735
	Mean	54.581	73.302	207.937	130.496	314.948	184.588	133.978	113.440	341.186	249.007	102.336	397.446	77.195	57.304	55.451
	Std.	1.509	7.423	148.025	27.769	576.712	92.907	82.512	51.390	193.248	87.854	42.546	471.670	23.585	5.34452	0.861
	Rank	1	4	11	8	13	10	9	7	14	12	6	15	5	3	2
25	Best	53.581	74.979	126.733	111.400	83.876	74.451	61.648	65.162	90.491	153.223	60.073	78.704	57.743	55.240	55.061
	Worst	53.785	112.172	1609.607	272.376	2182.624	366.400	489.869	577.207	1503.886	905.050	432.937	1980.693	110.751	63.672	56.795
	Mean	53.705	93.682	368.502	160.852	352.786	168.946	175.648	162.158	455.569	495.212	136.757	460.333	71.813	56.626	55.922
	Std.	0.071	12.728	285.403	33.317	370.204	67.626	107.691	107.937	283.937	211.767	85.818	401.653	14.139	1.532	0.433
	Rank	1	5	12	7	11	9	10	8	13	15	6	14	4	3	2
Average Rank		1.500	3.250	11.250	8.000	11.250	10.000	10.750	8.500	13.500	11.000	5.750	14.500	5.250	3.000	2.500
Overall Rank		1	4	12	7	12	9	10	8	14	11	6	15	5	3	2

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

Table 8. *Comparison between gintIPA and other path planners for Scenario-3*

D		gintIPA	IPA	GA	GAPSO	MFO	SSA	PFA	SBO	SCA	ECTLBO	HSGWO-MSOS	CIPSO	GWO	AEO	NSEAEO
10	Best	49.783	49.927	56.638	55.832	57.290	53.727	53.576	53.350	70.344	56.020	53.967	53.589	53.639	52.638	51.186
	Worst	49.814	53.470	424.807	466.463	186.974	140.715	499.851	330.582	270.192	100.391	159.386	454.467	290.474	62.066	78.808
	Mean	49.797	51.208	124.320	105.328	97.579	79.831	144.949	120.515	132.559	76.156	77.725	137.143	82.311	59.555	55.885
	Std.	0.013	1.341	100.904	89.177	28.811	28.453	121.329	82.407	55.161	12.884	21.098	103.686	43.614	4.187	5.391
	Rank	1	2	12	10	9	7	15	11	13	5	6	14	8	4	3
15	Best	49.755	57.908	57.131	58.074	58.535	51.142	52.823	52.724	103.034	80.003	55.461	59.655	53.811	50.429	50.155
	Worst	49.772	61.259	239.798	146.225	333.608	206.804	423.168	230.734	472.427	193.046	236.558	381.843	160.144	220.555	54.568
	Mean	49.766	59.875	98.429	74.533	103.670	77.403	125.829	84.734	213.126	130.755	93.354	151.807	70.679	61.643	52.163
	Std.	0.007	0.970	42.726	17.151	58.943	42.210	90.463	50.770	90.269	21.543	44.605	83.722	25.917	31.234	1.434
	Rank	1	3	10	6	11	7	12	8	15	13	9	14	5	4	2
20	Best	49.816	66.924	86.357	64.500	76.072	52.148	54.123	53.267	104.708	95.829	57.597	60.565	53.736	50.843	50.257
	Worst	49.875	89.067	397.440	261.488	1048.996	303.329	435.366	306.776	573.087	380.192	203.090	779.707	365.019	64.468	54.184
	Mean	49.848	81.597	180.662	96.442	236.951	91.752	150.474	113.194	258.060	215.535	108.225	224.292	82.149	54.906	52.492
	Std.	0.024	6.418	83.360	35.996	217.852	53.819	96.867	58.481	119.930	59.956	38.526	154.675	57.171	2.108	1.113
	Rank	1	4	11	7	14	6	10	9	15	12	8	13	5	3	2
25	Best	49.848	64.484	118.945	95.917	96.659	57.896	55.473	51.546	83.408	113.601	59.155	100.647	54.723	53.190	51.090
	Worst	49.951	115.114	789.659	230.781	1150.201	262.789	461.069	296.169	926.916	688.007	239.553	888.148	166.014	57.591	54.327
	Mean	49.888	92.706	227.699	138.983	395.133	105.547	129.069	112.060	356.789	347.054	111.295	367.574	74.745	55.191	53.648
	Std.	0.041	16.350	133.016	28.790	283.375	48.219	93.393	53.088	206.941	159.762	44.611	196.039	25.892	0.916	0.649
	Rank	1	5	11	10	15	6	9	8	13	12	7	14	4	3	2
Average Rank		1.000	3.500	11.000	8.250	12.250	6.500	11.500	9.000	14.000	10.500	7.500	13.750	5.500	3.500	2.250
Overall Rank		1	3	11	8	13	6	12	9	15	10	7	14	5	3	2

Table 9. *Comparison between the execution times of gintIPA and IPA*

		Scenario-1		Scenario-2		Scenario-3	
D		gintIPA	IPA	gintIPA	IPA	gintIPA	IPA
10	Best	0.077	0.097	0.083	0.102	0.091	0.105
	Worst	0.115	0.130	0.132	0.130	0.132	0.126
	Mean	0.102	0.101	0.107	0.105	0.109	0.108
	Std.	0.010	0.006	0.013	0.008	0.012	0.009
15	Best	0.088	0.090	0.095	0.093	0.108	0.099
	Worst	0.116	0.117	0.131	0.126	0.176	0.147
	Mean	0.102	0.097	0.106	0.101	0.112	0.108
	Std.	0.010	0.007	0.010	0.009	0.016	0.012
20	Best	0.115	0.121	0.123	0.117	0.125	0.127
	Worst	0.174	0.145	0.159	0.156	0.162	0.171
	Mean	0.140	0.132	0.138	0.131	0.139	0.132
	Std.	0.018	0.007	0.010	0.011	0.011	0.010
25	Best	0.152	0.144	0.146	0.148	0.163	0.157
	Worst	0.214	0.198	0.182	0.191	0.208	0.192
	Mean	0.166	0.158	0.161	0.154	0.177	0.167
	Std.	0.015	0.012	0.011	0.010	0.014	0.009

Table 10. *Sr and Me metrics of gintIPA and IPA*

		Scenario-1		Scenario-2		Scenario-3	
D		gintIPA	IPA	gintIPA	IPA	gintIPA	IPA
10	*Sr*	100.000	100.000	30.000	100.000	100.000	16.667
	Me	145.933	2664.200	332.000	4142.733	417.667	4567.000
15	*Sr*	100.000	100.000	100.000	73.333	100.000	0.000
	Me	514.667	5009.367	831.133	4418.591	746.933	–
20	*Sr*	100.000	0.000	100.000	0.000	100.000	0.000
	Me	966.767	–	387.133	–	1617.933	–
25	*Sr*	100.000	0.000	100.000	0.000	100.000	0.000
	Me	1537.400	–	437.967	–	3681.700	–

10-dimensional test case of Scenario-2, gintIPA obtains final solutions whose objective function values are less than the threshold for only 10 of 30 runs.

The comparative studies between the gintIPA and IPA indicate that gintIPA is capable of planning more qualified paths than IPA by bringing negligible amount of computational burden. However, demonstrating the effectiveness of the gintIPA against IPA with an appropriate statistical test is important to prove the critical assistance of the modifications. Wilcoxon signed rank test is a well-known non-parametric statistical test and has a widespread usage for the comparison of meta-heuristics [63]. If the significance level (ρ) of Wilcoxon signed rank test is found less than a constant usually fixed to 0.05, the difference between the compared algorithms is enough to state that one of the competitors is statistically better [63]. The test results of the comparison between the gintIPA and IPA were given in Table 11 where Z represents the test statistics and $W+$ and $W-$ show the sum of ranks for which gintIPA lags behind IPA and the sum of ranks for which gintIPA outperforms IPA. As easily seen from the ρ values in Table 11, gintIPA is statistically better than IPA for all cases of Scenario-1 and Scenario-3. Moreover, the Wilcoxon signed rank test validates that the difference between the qualities of the paths found by gintIPA and IPA is enough to generate a statistical significance in favour of the gintIPA for 15, 20 and

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

Table 11. *Results of the Wilcoxon signed rank test for gintIPA and IPA*

		Scenario-1	Scenario-2	Scenario-3
D		gintIPA vs IPA	gintIPA vs IPA	gintIPA vs IPA
10	*Z*-Val.	−4.7911	4.7984	−4.7947
	ρ-Val.	1.658e-6	1.599e-6	1.629e-6
	$W+$	0	465	0
	$W-$	465	0	465
	Sign.	gintIPA	IPA	gintIPA
15	*Z*-Val.	−4.7849	−4.7984	−4.7880
	ρ-Val.	1.711e-6	1.599e-6	1.685e-6
	$W+$	0	0	0
	$W-$	465	465	465
	Sign.	gintIPA	gintIPA	gintIPA
20	*Z*-Val.	−4.7855	−4.7936	−4.7849
	ρ-Val.	1.705e-6	1.638e-6	1.711e-6
	$W+$	0	0	0
	$W-$	465	465	465
	Sign.	gintIPA	gintIPA	gintIPA
25	*Z*-Val.	−4.7777	−4.7825	−4.7830
	ρ-Val.	1.773e-6	1.731e-6	1.727e-6
	$W+$	0	0	0
	$W-$	465	465	465
	Sign.	gintIPA	gintIPA	gintIPA

25-dimensional cases of Scenario-2. Utilising from the properties of the geometrical model as in the introduced greedy initialiser and redesigning the exploitation dominant operations by considering the gains related to the initialisation boosts the planning capabilities of the gintIPA and assist it to obtain solutions whose superiorities are statistically prominent compared to the solutions found by the standard IPA.

6.0 Conclusion

The technological advancements changed the design approaches and usage purposes of UAVs and various countries started to reorganise their air forces and defence systems according to the existence of UAVs or weaponised variants of them called UCAVs. In this study, the IPA was supported with a newly designed greedy initialiser and fine-tuned treatment schema and then a recent UAV or UCAV path planner named as gintIPA was proposed. The performance of the gintIPA was investigated by planning paths for three different battlefields and 12 test cases generated by adjusting the number of segmentation points and obtained results were compared with the results of other meta-heuristic-based techniques including IPA, GA, GAPSO, MFO, SSA, PFA, SBO, SCA, ECTLBO, HSGWO-MSOS, CIPSO, GWO, AEO and NSEAEO. Comparative studies allowed us to state that gintIPA handles challenging objectives about the enemy threats, fuel or battery consumption and turning manoeuvres more robustly and calculates better paths than all of 14 techniques for the 91.6% of the test cases.

The greedy initialiser generates a population containing individuals corresponding to the promising path candidates by adding acceptable amount of computational burden and helps gintIPA to start its operations with the solutions in the effectively narrowed search space. Moreover, the modernised treatment schema in which two separate procedures, on managing the collection of the plasma and, of the transferring the collected plasma to the receiver, significantly improves the exploitation characteristics

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

of gintIPA and removes the further requirement to the *NoD* and *NoR* control parameters. In the future, the greedy initialiser can be redesigned by using a repair module for determining whether a line segment between two selected points intersects with a circle while keeping the computational complexity still polynomial and integrated into the workflow of the IPA or other meta-heuristic algorithms. Also, parallel implementations of the IPA-based path planners for which each core or node of a cluster uses a unique initialiser and treatment schema can be developed and tested for two and three-dimensional battlefield environments containing dynamic enemy threats and non-flight zones.

Acknowledgment. We would like to thank the Proofreading & Editing Office of the Dean for Research at Erciyes University for the copyediting and proofreading service for this manuscript.

Funding. This study was not funded by any organisation.

Competing interests. The author declares that he has no conflict of interest.

Ethical approval. This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Bushnell, D.M. The 'air side' of future warfare: Military aeronautics, *Aeronaut. J.*, 2003, **107**, (1072), pp 301–305.
[2] Uzun, M. and Oktay, T. Simultaneous UAV having actively sweep angle morphing wing and flight control system design, *Aircr. Eng. Aerosp. Technol.*, 2023, **95**, (7), pp 1062–1068.
[3] Yesilbas, E., Özgür, B., Ozen, E. and Oktay, T. Simultaneous and stochastic design of piston-prop TUAV vertical tail and its autonomous system, *Aircr. Eng. Aerosp. Technol.*, 2025, **97**, (4), pp 395–404.
[4] Gutierrez-Martinez, M.A., Rojo-Rodriguez, E.G., Cabriales-Ramirez, L.E., Estabridis, K. and Garcia-Salazar, O. Genetic algorithm-based path planning of quadrotor UAVs on a 3D environment, *Aeronaut. J.*, 2024, **129**, (1334), pp 1–37.
[5] Kose, O. and Oktay, T. Simultaneous design of morphing hexarotor and autopilot system by using deep neural network and SPSA, *Aircr. Eng. Aerosp. Technol.*, 2023, **95**, (6), pp 939–949.
[6] Rostami, M., Bardin, J., Neufeld, D. and Chung, J. Small aircraft flight trajectory optimisation using a multidisciplinary approach, *Aeronaut. J.*, 2024, **129**, (1333), pp 1–19.
[7] Aslan, S. and Demirci, S. Immune plasma algorithm: A novel meta-heuristic for optimization problems, *IEEE Access*, 2020, **8**, pp 220227–220245.
[8] Aslan, S. and Demirci, S. An improved immune plasma algorithm with a regional pandemic restriction, *Sig. Image Video Process.*, 2022, **16**, (8), pp 2093–2101.
[9] Ajay, V. and Nesasudha, M. Detection of attackers in cognitive radio network using optimized neural networks, *Intell. Automat. Soft Comput.*, 2022, **34**, (1), pp 193–204.
[10] Arslan, S. Solving the problem of time series prediction using immune plasma programming, *Avrupa Bilim ve Teknoloji Dergisi*, 2022, **29**, pp 219–224.
[11] Kisa, M., Demirci, S., Arslan, S. and Aslan, S. Solving channel assignment problem in cognitive radio networks with immune plasma algorithm, In *2021 6th International Conference on Computer Science and Engineering (UBMK)*, IEEE, 2021, pp 818–822.
[12] Tasdemir, A., Demirci, S. and Aslan, S. Performance investigation of immune plasma algorithm on solving wireless sensor deployment problem, In *2022 9th International Conference on Electrical and Electronics Engineering (ICEEE)*, IEEE, 2022, pp 296–300.
[13] Aslan, S. and Erkin, T. An immune plasma algorithm based approach for UCAV path planning, *J. King Saud Univ.-Comput. Inform. Sci.*, 2023, **35**, (1), pp 56–69.
[14] Zhang, Y., Wu, L. and Wang, S. UCAV path planning by fitness-scaling adaptive chaotic particle swarm optimization, *Math. Probl. Eng.*, 2013, (2013), pp 1–9.
[15] Duan, H. and Qiao, P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning, *Int. J. Intell. Comput. Cybernet.*, 2014, **7**, (1), pp 24–37.
[16] Li, B., Gong, L. and Yang, W. An improved artificial bee colony algorithm based on balance-evolution strategy for unmanned combat aerial vehicle path planning, *ScientificWorldJournal*, 2014, **2014**, (4), pp 1–10.
[17] Tang, Z. and Zhou, Y. A glowworm swarm optimization algorithm for uninhabited combat air vehicle path planning, *J. Intell. Syst.*, 2015, **24**, (1), pp 69–83.
[18] Zhang, X. and Duan, H. An improved constrained differential evolution algorithm for unmanned aerial vehicle global route planning, *Appl. Soft Comput.*, 2015, **26**, pp 270–284.
[19] Zhou, Q., Zhou, Y. and Chen, X. A wolf colony search algorithm based on the complex method for uninhabited combat air vehicle path planning, *Int. J. Hybrid Inform. Technol.*, 2014, **7**, (1), pp 183–200.
[20] Zhang, B. and Duan, H. Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, 2015, **14**, (1), pp 97–107.
[21] Chen, Y., Yu, J., Mei, Y., Wang, Y. and Su, X. Modified central force optimization (MCFO) algorithm for 3D UAV path planning, *Neurocomputing*, 2016, **171**, pp 878–888.

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

[22] Zhou, Y., Bao, Z., Wang, R., Qiao, S. and Zhou, Y. Quantum wind driven optimization for unmanned combat air vehicle path planning, *Appl. Sci.*, 2015, **5**, (4), pp 1457–1483.
[23] Wang, G.G., Chu, H.E. and Mirjalili, S. Three-dimensional path planning for UCAV using an improved bat algorithm, *Aerosp. Sci. Technol.*, 2016, **49**, pp 231–238.
[24] Zhang, S., Zhou, Y., Li, Z. and Pan, W. Grey wolf optimizer for unmanned combat aerial vehicle path planning, *Adv. Eng. Softw.*, 2016, **99**, pp 121–136.
[25] Liu, Y., Zhang, X., Guan, X. and Delahaye, D. Adaptive sensitivity decision based path planning algorithm for unmanned aerial vehicle with improved particle swarm optimization. *Aerosp. Sci. Technol.*, 2016, **58**, pp 92–102.
[26] Liu, Y., Zhang, X., Zhang, Y. and Guan, X. Collision free 4d path planning for multiple UAVs based on spatial refined voting mechanism and PSO approach, *Chin. J. Aeronaut.*, 2019, **32**, (6), pp 1504–1519.
[27] Luo, Q., Li, L. and Zhou, Y. A quantum encoding bat algorithm for uninhabited combat aerial vehicle path planning, *Int. J. Innovative Comput. Appl.*, 2017, **8**, (3), pp 182–193.
[28] Zhang, Q., Wang, R., Yang, J., Ding, K., Li, Y. and Hu, J. Modified collective decision optimization algorithm with application in trajectory planning of UAV, *Appl. Intell.*, 2018, **48**, (8), pp 2328–2354.
[29] Alihodzic, A., Tuba, E., Capor-Hrosik, R., Dolicanin, E. and Tuba, M. Unmanned aerial vehicle path planning problem by adjusted elephant herding optimization. In *2017 25th Telecommunication Forum (Telfor)*, IEEE, 2017, pp 1–4.
[30] Miao, F., Zhou, Y. and Luo, Q. A modified symbiotic organisms search algorithm for unmanned combat aerial vehicle route planning problem, *J. Oper. Res. Soc.*, 2019, **70**, (1), pp 21–52.
[31] Miao, F., Li, H., Yan, G., Mei, X., Wu, Z., Zhao, W., Liu, T. and Zhang, H. Optimizing UAV path planning in maritime emergency transportation: A novel multi-strategy white shark optimizer, *J. Mar. Sci. Eng.*, 2024, **12**, (7), p 1207.
[32] Dolicanin, E., Fetahovic, I., Tuba, E., Capor-Hrosik, R. and Tuba, M. Unmanned combat aerial vehicle path planning by brain storm optimization algorithm, *Stud. Inf. Control*, 2018, **27**, (1), pp 15–24.
[33] Lin, N., Tang, J., Li, X. and Zhao, L. A novel improved bat algorithm in UAV path planning, *J. Comput. Mater. Cont.*, 2019, **61**, pp 323–344.
[34] Pan, J., Liu, N. and Chu, S. A hybrid differential evolution algorithm and its application in unmanned combat aerial vehicle path planning, *IEEE Access*, 2020, **8**, pp 17691–17712.
[35] Qu, C., Gai, W., Zhong, M. and Zhang, J. A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning, *Appl. Soft Comput.*, 2020, **89**, p 106099.
[36] Qu, C., Gai, W., Zhang, J. and Zhong, M. A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning, *Knowl.-Based Syst.*, 2020, **194**, p 105530.
[37] Yi, J., Lu, M. and Zhao, X. Quantum inspired monarch butterfly optimisation for UCAV path planning navigation problem, *Int. J. Bio-Inspir. Computat.*, 2020, **15**, (2), pp 75–89.
[38] Chen, Y., Pi, D. and Xu, Y. Neighborhood global learning-based flower pollination algorithm and its application to unmanned aerial vehicle path planning, *Exp. Syst. Appl.*, 2021, **170**, p 114505.
[39] Zhu, H., Wang, Y., Ma, Z. and Li, X. A comparative study of swarm intelligence algorithms for UCAV path-planning problems, *Mathematics*, 2021, **9**, (2), p 171.
[40] Zhu, H., Wang, Y. and Li, X. UCAV path planning for avoiding obstacles using cooperative co-evolution spider monkey optimization, *Knowl.-Based Syst.*, 2022, **246**, p 108713.
[41] Zhou, X., Gao, F., Fang, X. and Lan, Z. Improved bat algorithm for UAV path planning in three-dimensional space, *IEEE Access*, 2021, **9**, pp 20100–20116.
[42] Xu, H., Jiang, S. and Zhang, A. Path planning for unmanned aerial vehicle using a mix-strategy-based gravitational search algorithm, *IEEE Access*, 2021, **9**, pp 57033–57045.
[43] Jiang, W., Lyu, Y., Li, Y., Guo, Y. and Zhang, W. UAV path planning and collision avoidance in 3d environments based on Pompd and improved grey wolf optimizer, *Aerosp. Sci. Technol.*, 2022, **121**, p 107314.
[44] Jarray, R., Al-Dhaifallah, M., Rezk, H. and Bouallègue, S. Parallel cooperative coevolutionary grey wolf optimizer for path planning problem of unmanned aerial vehicles, *Sensors*, 2022, **22**, (5), p 1826.
[45] Du, N., Zhou, Y., Deng, W. and Luo, Q. Improved chimp optimization algorithm for three-dimensional path planning problem, *Multimedia Tools Appl.*, 2022, **81**, (19), pp 27397–27422.
[46] Wang, X., Pan, J., Yang, Q., Kong, L., Snášel, V. and Chu, S. Modified mayfly algorithm for UAV path planning, *Drones*, 2022, **6**, (5), p 134.
[47] Niu, Y., Yan, X., Wang, Y. and Niu, Y. An adaptive neighborhood-based search enhanced artificial ecosystem optimizer for UCAV path planning, *Expert Syst. Appl.*, 2022, **208**, p 118047.
[48] Niu, Y., Yan, X., Wang, Y. and Niu, Y. Three-dimensional UCAV path planning using a novel modified artificial ecosystem optimizer, *Expert Syst. Appl.*, 2023, **217**, p 119499.
[49] Jia, Y., Qu, L. and Li, X. A double-layer coding model with a rotation-based particle swarm algorithm for unmanned combat aerial vehicle path planning, *Eng. Appl. Artif. Intell.*, 2022, **116**, p 105410.
[50] Jia, Y., Qu, L. and Li, X. Automatic path planning of unmanned combat aerial vehicle based on double-layer coding method with enhanced grey wolf optimizer, *Artif. Intell. Rev.*, 2023, **56**, (10), pp 12257–12314.
[51] Zhang, C., Zhou, W., Qin, W. and Tang, W. A novel UAV path planning approach: Heuristic crossing search and rescue optimization algorithm, *Expert Syst. Appl.*, 2023, **215**, p 119243.
[52] Ait-Saadi, A., Meraihi, Y., Soukane, A., Ramdane-Cherif, A. and Gabis, A.B. A novel hybrid chaotic Aquila optimization algorithm with simulated annealing for unmanned aerial vehicles path planning, *Comput. Electr. Eng.*, 2022, **104**, p 108461.

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press

[53] Ait-Saadi, A., Meraihi, Y., Soukane, A., Yahia, S., Ramdane-Cherif, A. and Gabis, A.B. An enhanced African vulture optimization algorithm for solving the unmanned aerial vehicles path planning problem, *Comput. Electr. Eng.*, 2023, **110**, p 108802.
[54] Xiaobing Yu, Jiang, N., Wang, X. and Li, M. A hybrid algorithm based on grey wolf optimizer and differential evolution for UAV path planning, *Expert Syst. Appl.*, 2023, **215**, p 119327.
[55] Chen, B., Jin, Y., Zhang, H. and Yang, M. An improved spherical vector and truncated mean stabilization based bat algorithm for UAV path planning, *IEEE Access*, 2023, **11**, pp 2396–2409.
[56] Huang, C., Zhou, X., Ran, X., Wang, J., Chen, H. and Deng, W. Adaptive cylinder vector particle swarm optimization with differential evolution for UAV path planning, *Eng. Appl. Artif. Intell.*, 2023, **121**, p 105942.
[57] Zhang, X., Zhang, X. and Miao, Y. Cooperative global path planning for multiple unmanned aerial vehicles based on improved fireworks algorithm using differential evolution operation, *Int. J. Aeronaut. Space Sci.*, 2023, **24**, (5), pp 1346–1362.
[58] Yu, X. and Luo, W. Reinforcement learning-based multi-strategy cuckoo search algorithm for 3d UAV path planning, *Expert Syst. Appl.*, 2023, **223**, p 119910.
[59] Hu, G., Zhong, J. and Wei, G. SaCHBA_PDN: Modified honey badger algorithm with multi-strategy for UAV path planning, *Exp. Syst. Appl.*, 2023, **223**, p 119941.
[60] Hu, G., Huang, F., Seyyedabbasi, A. and Wei, G. Enhanced multi-strategy bottlenose dolphin optimizer for UAVs path planning, *Appl. Math. Modell.*, 2024, **130**, pp 243–271.
[61] Wang, Z., Sun, G., Zhou, K. and Zhu, L. A parallel particle swarm optimization and enhanced sparrow search algorithm for unmanned aerial vehicle path planning, *Heliyon*, 2023, **9**, (4), p e14784.
[62] Deng, L., Chen, H., Zhang, X. and Liu, H. Three-dimensional path planning of UAV based on improved particle swarm optimization, *Mathematics*, 2023, **11**, (9), p 1987.
[63] Aslan, S. A hospitalization mechanism based immune plasma algorithm for path planning of unmanned aerial vehicles, *Int. J. Mach. Learn. Cybern.*, 2024, **15**, pp 3169–3199.
[64] Aslan, S. and Demirci, S. An immune plasma algorithm with q-learning based pandemic management for path planning of unmanned aerial vehicles, *Egypt. Inform. J.*, 2024, **26**, p 100468.
[65] Fu, S., Li, K., Huang, H., Ma, C., Fan, Q. and Zhu, Y. Red-billed blue magpie optimizer: A novel metaheuristic algorithm for 2D/3D UAV path planning and engineering design problems, *Artif. Intell. Rev.*, 2024, **57**, (6), pp 1–89.
[66] Yang, C., Yang, H., Zhu, D., Hu, Y., Zhang, Y., Ma, H. and Zhang, D. A multi-mechanism balanced advanced learning sparrow search algorithm for UAV path planning, *Cluster Comput.*, 2024, **27**, (5), pp 6623–6666.
[67] Haris, M., Bhatti, D.M.S. and Nam, H. A fast-convergent hyperbolic tangent PSO algorithm for UAVs path planning, *IEEE Open J. Vehicul. Technol.*, 2024, **5**, pp 681–694.
[68] Qu, L., Jia, Y., Li, X. and Fan, J. Two-stage control model based on enhanced elephant clan optimization for path planning of unmanned combat aerial vehicle, *J. Supercomput.*, 2024, **80**, pp 24938–24974.
[69] Chen, Q., Wang, Y. and Sun, Y. An improved dung beetle optimizer for UAV 3D path planning, *J. Supercomput.*, 2024, **80**, pp 26537–26567.
[70] Oktay, T., Arik, S., Turkmen, I., Uzun, M. and Celik, H. Neural network based redesign of morphing UAV for simultaneous improvement of roll stability and maximum lift/drag ratio, *Aircr. Eng. Aerosp. Technol.*, 2018, **90**, (8), pp 1203–1212.
[71] Oktay, T. and Coban, S. Simultaneous longitudinal and lateral flight control systems design for both passive and active morphing TUAVs, *Elektronika ir elektrotechnika*, 2017, **23**, (5), pp 15–20.
[72] Arik, S., Turkmen, I. and Oktay, T. Redesign of morphing UAV for simultaneous improvement of directional stability and maximum lift/drag ratio, *Adv. Electr. Comput. Eng.*, 2018, **18**, (4), pp 57–62.
[73] Aslan, S. Back-and-Forth (BAF): A new greedy algorithm for geometric path planning of unmanned aerial vehicles, *Computing*, 2024, **106**, pp 2811–2849.

Cite this article: Aslan S. and Erkin T. (2025). A greedy initialiser based meta-heuristic approach for planning unmanned aerial vehicles. *The Aeronautical Journal*, **129**, 2622–2647. https://doi.org/10.1017/aer.2025.10025

https://doi.org/10.1017/aer.2025.10025 Published online by Cambridge University Press