



# Designing INS/GNSS integrated navigation systems by using IPO algorithms

Ali Mohammadi<sup>1,2</sup> · Farid Sheikholeslam<sup>1</sup> · Mehdi Emami<sup>3</sup> · Seyedali Mirjalili<sup>4,5</sup>

Received: 3 February 2022 / Accepted: 21 March 2023

© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2023

## Abstract

The application of soft computing techniques can be largely found in engineering sciences. These include the design and optimization of navigation systems for use in land, sea, and air transportation systems. In this paper, an attempt is made to leverage on novel metaheuristic optimization approaches for designing integrated navigation systems. For this purpose, a simplified version of the inclined planes system optimization (called SIPO) algorithm alongside its two standard and modified versions are used in comparison with the two conventional methods of genetic algorithm and particle swarm optimization. Considerations are made on an INS/GNSS problem with IMU MEMS modules. Outputs are presented in terms of statistical and performance indicators, such as runtime, fitness, convergence, navigation accuracy (velocity, latitude, longitude, altitude, roll, pitch, yaw), and routing along with the ranking of algorithms. Competitive performance and relative superiority of the standard IPO over other methods in evaluating results have been confirmed. So that compared to other state-of-the-art algorithms (GA, PSO, IPO, and MIPO), the best runtime rank with a value of 6/4 by SIPO and the best performance rank of fitness, navigation accuracy for the two assumed IMU modules, and the total rank with values of 4/4, 149/60, 165/60, and 332/128 obtained by IPO, respectively.

**Keywords** Inclined planes system optimization · IMU MEMS · INS/GNSS integrated navigation · Literature review · Optimal design · Soft computing

✉ Ali Mohammadi  
a.mohammadi98@pd.iut.ac.ir;  
a.mohammadi98@yahoo.com

Farid Sheikholeslam  
sheikh@iut.ac.ir

Mehdi Emami  
m.emami@stu.yazd.ac.ir

Seyedali Mirjalili  
ali.mirjalili@torrens.edu.au; ali.mirjalili@gmail.com

<sup>1</sup> Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran

<sup>2</sup> Department of Electrical and Computer Engineering, Esfarayen University of Technology, Esfarayen 96619-98195, Iran

<sup>3</sup> Department of Electrical and Computer Engineering, Yazd University, Yazd 89158-18411, Iran

<sup>4</sup> Centre for Artificial Intelligence Research and Optimization, Torrens University Australia, Brisbane, Australia

<sup>5</sup> Yonsei Frontier Lab, Yonsei University, Seoul, South Korea

## 1 Introduction

In autonomous vehicles, navigation has one of two meanings: (a) estimating the position, velocity, and attitude of the vehicle relative to a specific reference derived from sensor observations; (b) design and carry out vehicle movements to reach the desired location. Position, velocity, and attitude are called navigation states. The term vehicle is also used for a moving body whose position and attitude must be determined. Positioning methods are divided into three groups: inertial, satellite, and integrated navigation. In inertial navigation, gyroscopes and accelerometers measure rotation and specific force, respectively, and then acceleration is obtained from the specific force.

An inertial navigation system (INS) suffers from a sharp increase in error over time, which is due to both the nature of the sensors and the type of mechanization. This can be solved by considering the required conditions, such as achieving high accuracy sensors along with the use of appropriate and effective navigation algorithms. To

overcome the problems in INSs, however, aided navigation systems such as Global Positioning System (GPS), also known as Global Navigation Satellite System (GNSS) navigation systems, can be used alongside them. The main problem in integrated navigation is using a properly integrated filter that provides a navigation response in the shortest possible time and with the least error.

In integrated navigation, two models are used: one to model the state transition process INS and the other to model GPS observation. In such models, the uncertainties in INS and GPS sensors are modeled with two noise sources, called process noise and measurement noise, respectively (which can generally be non-Gaussian and non-white). Based on these two models, and using the correction and updating equations, the states estimated by the INS are corrected based on GPS observations.

There are several significant challenges in designing and implementing navigation systems to achieve this goal. One of them is accurate modeling of the INS transition process and GPS observation. Therefore, if linearized models of the system and Gaussian white noise are used, since the INS is essentially a nonlinear system with non-Gaussian and non-white noise, the accuracy of the navigation system will be reduced. Navigation algorithms based on the linear Kalman filter (LKF) [1, 2] and the extended Kalman filter (EKF) [2, 3] use linear error state models. Despite the advantages of Micro-Electro-Mechanical System (MEMS) inertial sensors (such as low-cost, small size, and low power consumption), as one of the most widely used sensors in the field of inertial navigation, KF-based methods when using MEMS inertial measurement units (IMUs), from divergence during GPS outages, which result from linearization process approximations and undesirable system modeling, are also not safe. Kalman navigation filter as the core of the navigation system, especially integrated navigation, is an optimal estimation tool that provides a sequential recursive algorithm for estimating system states [4].

In various studies [5–15], nonlinear methods have been proposed to compensate KF defects. The use of other estimating filters, such as unscented Kalman filter (UKF) [16] and particle filter (PF) [17] and other theoretical methods to solve navigation problems have also been pursued in them. Consequently, they have accepted the high volume of calculations and theoretical considerations, and operational assumptions to achieve the desired answers. In contrast, some studies have achieved intelligent and optimal navigation systems by focusing on integrated navigation and exploiting the unique potential of artificial intelligence (AI)-based soft computing approaches such as artificial neural networks, fuzzy logic, deep and reinforcement learning, and evolutionary or metaheuristic optimization algorithms [18–32]. Also, in [33–42], these

approaches have been used in other navigation applications.

In the present study, in line with many similar studies, emphasis has been placed on using new approaches, including KF and AI hybrid systems, to complement each other. Using a KF as the main integration filter and adjusting its parameters can improve the performance of an INS [43]. Such methods are known as adaptive Kalman filters (AKF) [44–48]. The main methodology in such research is a method and algorithm that can intelligently and completely optimally estimate the control values of the algorithm and provide the appropriate answer and solution with the least possible computational and time volume.

Therefore, contributions of the current research are the application of a simplified version of the inclined planes system optimization algorithm, for the first time, alongside two standard and modified versions (IPO[49], MIPO [50], and SIPO[51]) in this regard to intelligently estimate the covariance noise matrices of an EKF from an INS/GNSS system and achieve an optimal navigation algorithm and desired results. In such a way that, over time and based on the measured values reached to the filter, the correlation matrices of process noise and the measurement noise, denoted by  $Q$  and  $R$ , respectively, are adapted to obtain the least estimation error.

In simulations, based on technical and theoretical considerations governing an assumed INS/GNSS navigation problem with two sets of IMU-based INSs with unique features [52], the IPO algorithms for estimating matrices  $Q$  and  $R$  are used. The results of the algorithms are compared with two well-known algorithms, genetic algorithm (GA) [53] and particle swarm optimization (PSO) [54]. The rest of this paper is organized as follows: a comprehensive literature review of recent related research is provided in Sect. 2. The navigation problem is presented in Sect. 3. Section 4 describes the concept of intelligent optimization and the proposed algorithms. Section 5 presents the proposed approach with its considerations. The results and analysis are reported in Sect. 6 and, finally, the paper concluded in Sect. 7.

## 2 Literature review

This section reviews some recent related research. Here, all references in relation to the use of different types of soft computing techniques in the design of inertial and integrated navigation problems on different applications are reviewed. For this purpose, the studied problem, the proposed approach(s) along with the main outputs are reported. The references include all researches from 2019 to 2021, respectively, which give a complete perspective on

the application of such soft computing methods and how they work in this regard.

In 2019, Sathiya & Chinnadurai used a Heterogeneous multi-objective differential evolution (HMODE) and elitist non-dominated sorting genetic algorithm (NSGA-II) for trajectory planning of mobile robot. They obtained desirable outputs based on the optimal trajectory planning of differential-driven wheeled mobile robot (WMR). Jalali et al. [19] designed an autonomous robot navigation problem by training multilayer perceptron (MLP) networks using six evolutionary algorithms (EAs), multi-verse optimizer (MVO), moth-flame optimization (MFO), PSO, cuckoo search (CS), grey wolf optimizer (GWO), and bat algorithm. In [19], the overall evaluation indicated the robust performance of the MVO in terms of two evaluation metrics (accuracy and area under curve (AUC)), the convergence profile, and the t-test results.

In 2020, several similar works have been done. For example, in [20], using the deep reinforcement learning approach in 3D robot navigation, a robot navigation environment is launched in various dense pedestrian environments. In [21], the authors were able to design a MEMS-based GNSS/INS land vehicle navigation system optimally by enhancing the signal-to-noise ratio (SNR) of MEMS-INS raw measurements, utilizing a hybrid denoising algorithm with wavelet transform and support vector machine (SVM), as well as improving the positioning accuracy by a SVM-based data fusion approach. They have finally effectively eliminated stochastic errors of MEMS-IMUs and also improved the overall positioning accuracy [21]. Also in [22], quantum monarch butterfly optimization (QMBO) has been used in path planning navigation for uninhabited combat air vehicles (UCAV). The results showed that the proposed method searches and finds a much shorter path than the standard version MBO. In [23], Bellemare et al. also proposed another method based on reinforcement learning for use in autonomous navigation of stratospheric balloons. The proposed controller in [23] outperformed Loon's algorithm and was robust to the natural diversity in stratospheric winds. They acknowledged that reinforcement learning is an effective solution to real-world autonomous control problems in which neither conventional methods nor human intervention suffice.

An integrated MEMS-based INS/GNSS problem based on UKF and nonlinear autoregressive neural networks with external inputs (NARX) (namely NARX aided UKF) has been investigated in [24]. The proposed approach in [24] has resulted the improvement in position accuracy about 85–90%, in velocity about 65–75%, and in attitude about 40–65%, which indicates its success. Using multiple fading factor square root cubature kalman filter (MSCKF) and generalized dynamic fuzzy NN model based on MSCKF (MSCKF-GDFNN), Wang et al. were able to design a

strapdown inertial navigation system (SINS) and GNSS, which reduced the position errors in latitude and longitude by 85.00%, 89.71%, and the velocity errors in east and north by 94.57%, 83.11% [25]. The results in [25] indicate the success and good performance of new hybrid approaches between kalman filters and intelligent soft computing techniques.

Gul et al. recently proposed a new hybrid methodology called PSO–GWO algorithm with evolutionary programming as a multi-objective path planning algorithm [26] and used it to solve path planning for autonomous guided robot. In [26], they have shown that their proposed method has achieved a more feasible path with a short distance and overcoming the shortcomings of other conventional technique. The reported results indicate the success and usefulness of the multi-objective optimization approach in such important problems in the field of navigation. In another study in 2021, the authors solved a vision-based 3D robotic navigation challenge by using deep reinforcement learning techniques [27]. In this research [27], due to the proposed approach, the ability to avoid collisions has been met and the target object can be quickly traced and detected. Also, in the continuation of recent researches, in [28], a gravity compensation method based on multilayer feedforward neural network has been used to overcome the challenges of optimal design of INS problem. In [28], researchers have improved the radial position error performance more than 31.43%, and ensured the real-time performance of the gravity compensation.

Proposing a novel meta reinforcement learning framework based on transfer learning and providing a dynamic proximal policy optimization with covariance matrix adaptation evolutionary strategies (dynamic-PPO-CMA) to extend original proximal policy optimization (PPO) algorithm is a new approach that has recently been applied in [29] to the problem of multi-robot path-planning for autonomous navigation. In [29], using the above method, a faster convergence rate is achieved and the vehicle arrived quickly to the destination. The authors in [30] have also designed and implemented another integrated INS/GNSS system with a combination of UKF and NARX. They have shown that the proposed intelligent hybrid approach can dramatically improve the accuracy of the designed system during GNSS outages. In another recent study, Yang et al. designed a SINS/GPS integrated navigation system [31]. In this regard, they have used a deep neural network (DNN)-based real-time sequence analyzer (RTSA) and covariance matching algorithm hybrid adaptive nonlinear filter to estimate the dynamic state, and time-varying noise covariances  $Q$  and  $R$  of SINS and GPS [31]. The outputs of the proposed approach indicated high generalization capability and accurately label multi-frequency compound

vibrations, as well as better accuracy, robustness, and flexibility.

Finally, [32] provides a survey on population-based meta-heuristic algorithms for solving aircraft motion planning (AMP) problems. There have been some effective suggestions on how to select appropriate meta-heuristic methods in such problems, including the outputs of the present work. The overall assessment indicates the need and initial motivations of this paper for evaluating new AI-based approaches in the optimal design of integrated navigation systems.

### 3 Problem statement

Navigation involves moving and finding the way from one place to another. To achieve this goal, a variety of equipment is available. Navigation covers a wide range of applications, from industrial and commercial applications to military applications [55].

The four types of sample coordinate devices used in navigation systems are: inertia device, ground device, navigation device, and body device. These coordinate devices are used because of INS mechanized outputs including attitude, velocity, and position need to be converted into user-understandable navigation information [56].

#### 3.1 Inertial navigation system

The heart of an INS is its navigation processor, which uses IMU measurements using mechanization. Mechanization refers to generating navigation responses from a set of raw measurements obtained from sensors. This process begins with the initialization and alignment of the system, followed by differential equations to provide navigation responses [56].

#### 3.2 Integrated navigation

Generally speaking, the final goal of integrated navigation is to estimate a state vector  $x_k$  of a moving device in the current time step  $k$  by having a set of measurements (observations) collected in time steps of  $0, 1, \dots, k$  ( $Z_k = \{z_0, \dots, z_k\}$ ). The state vector of a moving device is in the form of Eq. (1):

$$x_k = [\phi_k, \theta_k, \psi_k, v_k^N, v_k^E, v_k^D, \gamma_k, \lambda_k, h_k]^T \tag{1}$$

where  $\phi_k, \theta_k, \psi_k, v_k^N, v_k^E, v_k^D, \gamma_k, \lambda_k$ , and  $h_k$  are roll, pitch, and yaw angles of the moving device, along with velocity vectors, as well as the latitude, longitude, and altitude, respectively.

The state transition model (motion model) of the system is described as Eq. (2):

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \tag{2}$$

where  $u_k$  is the control input which is the read values of IMU,  $w_k$  is the process noise which is independent of the past and present states of the system and is considered due to the uncertainty in the movement of the moving device and the read values of IMU.

The state measurement model is also:

$$z_k = h(x_k, v_k) \tag{3}$$

where  $v_k$  is a measurement noise that is independent of the past and current states of the system as well as independent of process noise and is considered due to the uncertainty in the read values of GPS.

The functions  $f$  and  $h$  are inherently nonlinear in the state transition model and the measurement model, and the process and measurement noises are essentially non-Gaussian and non-white. Therefore, the main problem of integrated navigation is the modeling of these functions and noises, which is also considered by researchers in the field of intelligent optimization based on artificial intelligence.

#### 3.3 Assumed INS/GNSS integrated navigation problem based on NaveGo simulation framework

The basic problem and considerations of the present article are entirely based on [52]; therefore, some important issues are briefly stated here and the rest are referred to [52]. In the integrated structure designed in [52], a version of the extended Kalman filter called the complementary filter has been implemented. By disturbing the navigation equations, the SINS error model is acquired (defined by a series of first-order differential equations) [52]. The assumed structure is described in [52].

In the time domain, the continuous state-space of the system is equal to [52]:

$$\delta \dot{\hat{x}}_{(t)} = F_{(t)} \delta \hat{x}_{(t)} + G_{(t)} u_{(t)} + w_{(t)} \tag{4}$$

$$\delta \dot{\hat{z}}_{(t)} = H \delta \hat{x}_{(t)} + v_{(t)} \tag{5}$$

The state space model, discrete time, is [52]:

$$\delta \hat{x}_{(+)} = \Phi \delta \hat{x} + Gu + w \tag{6}$$

$$\delta \hat{z} = H \delta \hat{x} + v \tag{7}$$

where  $w \sim N(0, Q)$  and  $v \sim N(0, R)$  are process and measurement noise vectors, respectively; and the vectors  $u \in \mathbb{R}^{21}$ ,  $\delta \hat{x} \in \mathbb{R}^{21}$ , and  $\delta \hat{z} \in \mathbb{R}^6$  are also calculated from Eqs. (8) to (13) [52]:

$$u = [\tilde{\omega}^{bT}, \tilde{f}^{bT}, \eta_{g^{db}}^T, \eta_{f^{db}}^T]^T \tag{8}$$

$$\delta\hat{x} = [\delta\hat{e}^T, \delta\hat{v}^{nT}, \delta\hat{p}^{nT}, \hat{b}_g^T, \hat{b}_f^T, \delta\hat{b}_g^T, \delta\hat{b}_f^T]^T \tag{9}$$

$$\delta\hat{z} = [\delta\hat{z}_v^T, \delta\hat{z}_p^T]^T \tag{10}$$

$$\delta\hat{z}_v = [\hat{v}^n - \hat{v}_G^n] \tag{11}$$

$$\delta\hat{z}_p = \hat{T}_p^r(\hat{p}^n - \hat{p}_G^n) + \hat{C}_p^{l_b} \tag{12}$$

$$\hat{T}_p^r = \text{diag}([\hat{R}_M + \hat{h}], [\hat{R}_N + \hat{h}] \cos(\hat{\gamma}), -1] \tag{13}$$

where  $l_{ba}^l$  is the lever arm from the GPS antenna to the IMU module and the parameter  $\hat{T}_p^r$  is also the curvilinear-to-Cartesian transformation matrix [52].

Assuming that  $I_{\{3 \times 3\}}$  is an identical matrix and  $0_{\{3 \times 3\}}$  is an empty matrix, the state space matrices will be according to the relations with (14) to (16) [52]:

$$F_{(t)\{21 \times 21\}} = \begin{bmatrix} F_{ee} & F_{ev} & F_{ep} & 0 & -\hat{C}_b^n & 0 & -\hat{C}_b^n & 0 \\ F_{ve} & F_{vv} & F_{vp} & 0 & \hat{C}_b^n & 0 & 0 & -\hat{C}_b^n \\ 0 & F_{pv} & F_{pp} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_g} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\tau_g} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_f} \end{bmatrix} \tag{14}$$

$$G_{\{21 \times 12\}} = \begin{bmatrix} -\hat{C}_b^n & 0 & 0 & 0 \\ 0 & \hat{C}_b^n & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \tag{15}$$

$$H_{\{6 \times 21\}} = \begin{bmatrix} 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \hat{T}_p^r & 0 & 0 & 0 & 0 \end{bmatrix} \tag{16}$$

The  $Q$  and  $R$  covariance matrices are also described as Eqs. (17) and (18), respectively [52]:

$$Q_{\{12 \times 12\}} = \text{diag}([n_g^{2T}, n_f^{2T}, n_{g^{db}}^{2T}, n_{f^{db}}^{2T}]) \tag{17}$$

$$R_{\{6 \times 6\}} = \text{diag}([\sigma_v^{2T}, \sigma_{\gamma m}^2, \sigma_{\lambda m}^2, \sigma_h^2]) \tag{18}$$

The values of the main diameter  $R$  are provided by the GPS error profile, and:

$$n_g^2 = \sigma_g^2 \delta t \tag{19}$$

$$n_f^2 = \sigma_f^2 \delta t \tag{20}$$

$$n_{g^{db}}^2 = \sigma_{g^{db}}^2 \circ \tau_g \tag{21}$$

$$n_{f^{db}}^2 = \sigma_{f^{db}}^2 \circ \tau_f \tag{22}$$

where  $n_{f^{db}}^{2T}$ ,  $n_{g^{db}}^{2T}$ ,  $n_f^2$ , and  $n_g^2$  are the power spectral densities of gyroscopes and accelerometers random noises, as well as gyroscopes and accelerometers dynamic biases [52].

The covariance matrix  $P_{\{21 \times 21\}}$  as a diagonal matrix must be formed at the beginning of the mechanization process [52]. The  $P_{\{21 \times 21\}}$  entries are selected according to the expected initial variances  $\delta\hat{x}$  [52]:

$$P_{(1)} = \text{diag}([\sigma_\theta^2, \sigma_\theta^2, \sigma_\psi^2, \sigma_v^{2T}, \sigma_\gamma^2, \sigma_\lambda^2, \sigma_h^2, b_g^{2T}, b_f^{2T}, \sigma_{g^{db}}^{2T}, \sigma_{f^{db}}^{2T}]) \tag{23}$$

In [52], a simulation framework for low-cost INS, abbreviated NaveGo, is embedded as a MATLAB toolbox. In this framework, data is generated and received by a trajectory generator (Eq. (24)). This trajectory generator is required to provide the correct input data to the simulator.

$$T = [e, p, v^n, a^n, t_s] \tag{24}$$

where the vectors  $e = \{e_i\}_{i=1}^{k_s}$ ,  $p = \{p_i\}_{i=1}^{k_s}$ ,  $v^n = \{v_i^n\}_{i=1}^{k_s}$ ,  $a^n = \{a_i^n\}_{i=1}^{k_s}$ , and  $t_s = \{t_i\}_{i=1}^{k_s}$  are discrete-time sequences with elements equal to  $k_s$ ,  $e = [\varnothing, \theta, \psi]$  is the attitude vector, including the angles of roll, pitch and yaw, respectively. The  $p$  parameter also contains the true position vector mechanized in the local North-East-Down (NED) coordinate system. The vectors  $v^n$  and  $a^n$  are the true velocity and acceleration vectors, respectively, and the value of  $t_s$  is also a vector of simulation time [52].

In NaveGo [52], the state vector  $\tilde{x}$  is considered in the form of the following Eq. (25):

$$\tilde{x} = [\hat{e}^T, \hat{v}^{nT}, \hat{p}^{nT}, \hat{b}_g^T, \hat{b}_f^T, \delta\hat{b}_g^T, \delta\hat{b}_f^T]^T \tag{25}$$

In Appendix, descriptions about mathematical notations are provided.

### 4 Optimization and metaheuristic algorithms

Optimization is the process by which optimal output/solution/result (maximum or minimum) is made by setting the inputs of a problem or the specifications of a component. Optimization in mathematics means the desire to produce the desired result, which can be the minimum or maximum value of an index in the form of one or more objective functions. In practice, the application of definitive solution approaches and methods in optimization is not simply expressed; however, this has been met by using a variety of stochastic optimization methods, also called heuristics and metaheuristics.

A set of intelligent methods that complement each other to create robust and inexpensive systems are categorized into soft computing techniques. Soft computing includes methods such as neural networks, fuzzy logic, evolutionary computations (including genetic algorithms), swarm intelligence, and heuristic and metaheuristic approaches with random and probability-based reasoning, which can deal with uncertainty, ambiguity, incomplete or partial truth, machine learning, and optimization problems. These features allow the creation of inexpensive intelligent systems with a high degree of machine intelligence.

Many optimization problems in engineering are naturally more complex and challenging than can be solved by conventional optimization methods such as mathematical programming methods and the like. On the other hand, classical mathematics methods have two basic forms: (a) they consider the local optimal point as the global optimal point, and (b) each of these methods is used only for a specific problem. Therefore, the main purpose of intelligent methods is to find the optimal answer to engineering problems.

The use of metaheuristic algorithms for intelligent optimization leads to significant improvements in reducing time and computational volume required to solve the desired problems. These algorithms are heuristic search methods mainly based on counting methods, with the difference that they use additional information to guide the search. These methods are quite general in terms of application and can solve very complex problems. They mimic and exploit biological and physical processes. Some of the optimization algorithms include genetic algorithm (GA) [53, 57], ant colony optimization (ACO) [58], particle swarm optimization (PSO) [54], simulated annealing (SA) [59], gravitational search algorithm (GSA) [60], differential evolution (DE) [61–63], and gray wolf optimization (GWO) [64]. These metaheuristic models have been successfully used in various applications [65–68].

In the following, a detailed review of candidate algorithms for use in this research is given. Therefore, the inclined planes system optimization (IPO) algorithm [49], its improved version MIPO [50], and its simplified version SIPO [51] will be described, respectively.

### 4.1 Inclined planes system optimization

The basis of this algorithm has inspired by the dynamics of sliding motion on a frictionless inclined plane [49]. In IPO, a collection of agents (tiny balls) cooperate and move toward better positions in the search space according to Newton’s second law and equations of motion [49].

Consider a system with  $NP$  balls (see Fig. 1), the position of the ball  $i$  is defined by Eq. (26):

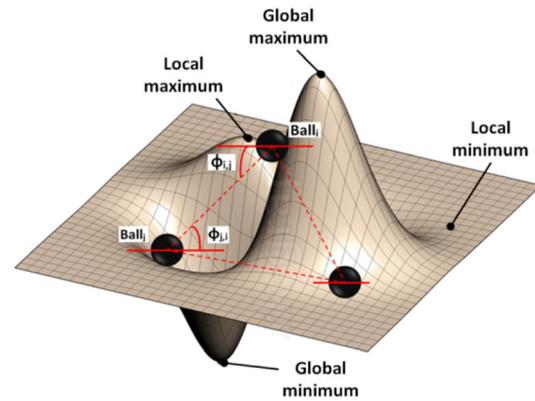


Fig. 1 A sample search space with three balls in IPO [49]

$$\bar{p}_i = (\bar{p}_i^1, \dots, \bar{p}_i^d, \dots, \bar{p}_i^D), \text{ for } i = 1, 2, \dots, NP \quad (26)$$

That

$$\bar{p}_j^{\min} \leq \bar{p}_j \leq \bar{p}_j^{\max}, 1 \leq j \leq n \quad (27)$$

So that  $\bar{p}_i^d$  is the position of the ball  $i$  in the dimension  $d$ , in the  $D$ -dimensional space. At a specified time/step, such as  $G$ , the angle between the ball  $i$  and  $j$  in dimension  $d$ ,  $\phi_{i,j}^d$ , is calculated as Eq. (28):

$$\phi_{(i,j),G}^d = \arctan\left(\frac{f(\bar{p}_{j,G}) - f(\bar{p}_{i,G})}{\bar{p}_{i,G}^d - \bar{p}_{j,G}^d}\right) \quad (28)$$

$$\forall i, j = 1, \dots, NP, i \neq j, d = 1, \dots, D$$

In Eq. (28),  $f(\bar{p}_{i,G})$  is the value of the objective function (height) for the  $i$ -th ball at step  $G$ . Since a certain ball must move to the lowest height on an inclined surface, only the balls at lower heights are used to calculate its acceleration. The amount and direction of acceleration for the ball  $i$  in step (iteration)  $G$ , in dimension  $d$ , is determined by Eq. (29):

$$\ddot{\bar{p}}_{i,G}^d = \sum_{j=1}^{NP} U(f(\bar{p}_{j,G}) - f(\bar{p}_{i,G})) \cdot \sin(\phi_{(i,j),G}^d) \quad (29)$$

where  $U(\cdot)$  is the unit step function:

$$U(\omega) = \begin{cases} 1 & \text{if } \omega > 0 \\ 0 & \text{otherwise} \end{cases} \quad (30)$$

Also,  $\dot{\bar{p}}_{i,G}^d$  is the velocity of the ball  $i$  in dimension  $d$ , at step  $G$ , which is calculated according to Eq. (31):

$$\dot{\bar{p}}_{i,G}^d = \frac{\bar{p}_{\text{best},G}^d - \bar{p}_{i,G}^d}{\Delta G} \quad (31)$$

where  $\bar{p}_{\text{best},G}^d$  is the ball with the lowest height (fitness) in the total iteration until the current iteration  $G$ .

Finally, Eq. (32) is used to update the position of balls:

$$\bar{p}_{i,G+1}^d = k_1 \text{rand}_1 \ddot{\bar{p}}_{i,G}^d \Delta G^2 + k_2 \text{rand}_2 \dot{\bar{p}}_{i,G}^d \Delta G + \bar{p}_{i,G}^d \quad (32)$$

where  $\text{rand}_1$  and  $\text{rand}_2$  are two random weights distributed uniformly over the interval  $[0, 1]$  [49].

Algorithm I presents the general IPO operation, which starts with a random initial population  $\mathcal{P}_0 \in \mathbb{R}^{NP \times D}$  of  $NP$  balls distributed inside the search space. This collection is moving along the search space during  $G_{max}$  displacements. Its movements are driven by the acceleration  $\ddot{\bar{p}}_i$  and velocity  $\dot{\bar{p}}_i$  of  $i$ -th ball in the inclined plane by crossing straight lines from the center of the particular ball to the center of other balls. The new ball position is computed at each movement  $G$  as seen in Eq. (32), where  $rand_1$  and  $rand_2$  are two random weights in the interval  $[0, 1]$ , and  $k_1$  and  $k_2$  are exploration and exploitation control parameters of the algorithm. Those last parameters are defined in Eqs. (33) and (34), where  $c_1$ ,  $c_2$ ,  $scale_1$ ,  $scale_2$ ,  $shift_1$ , and  $shift_2$  are tunable parameters that are empirically determined for each problem [49].

$$k_{1,G} = \frac{c_1}{1 + e^{(G - shift_1)scale_1}} \tag{33}$$

$$k_{2,G} = \frac{c_2}{1 + e^{-(G - shift_2)scale_2}} \tag{34}$$

---

**Algorithm I** Pseudo-code of the standard IPO algorithm

---

- 1:  $G \leftarrow 0$
- 2: Generate the initial population of balls  $\mathcal{P}_{G=0} \leftarrow rand_i[\bar{p}_{min}, \bar{p}_{max}] \forall i = 1, 2, \dots, NP$ .
- 3: Evaluate the heights (fitness)  $f(\bar{p}_{i,0}) \forall i = 1, 2, \dots, NP$
- 4: **While**  $G < G_{max}$  **do**
- 5:     **for**  $i = 1$  to  $NP$  **do**
- 6:         Calculate the angle  $\phi_{i,j}^d$  according to Eq. (28)
- 7:         Calculate the acceleration  $\ddot{\bar{p}}_i$  according to Eq. (29)
- 8:         Calculate the velocity  $\dot{\bar{p}}_i$  according to Eq. (31)
- 9:         Apply the random boundary handling method for all  $\bar{p}_{i,G}$
- 10:        Update the ball position  $\bar{p}_{i,G}$  according to Eq. (32)
- 11:        Evaluate the fitness  $f(\bar{p}_{i,G})$
- 12:     **end for**
- 13:     Update the best ball  $\bar{p}_{best}$
- 14:      $G \leftarrow G + 1$
- 15: **end while**

## 4.2 Modified inclined planes system optimization

To reduce the complexity of the standard IPO, a modified version called MIPO was proposed by Mohammadi et al. in

2017 [50]. Control parameters of MIPO are two parameters of  $k_1$  and  $k_2$ , which change under the proposed damping/friction coefficients  $k_{1damp}$  and  $k_{2damp}$  with iterations, as follows:

$$k_{1,G} = k_{1damp} \left( \frac{G_{max} - G}{G_{max}} \right) \tag{35}$$

$$k_{2,G} = k_{2damp} \left( \frac{G}{G_{max}} \right) \tag{36}$$

where  $G_{max}$  is the total number of iterations and  $G$  is the current iteration.

So that high values of  $k_1$  and lower values of  $k_2$ , cause greater accelerations. This leads to the greater motility of agents/balls. It means that, global searching or exploitation occurs with larger values of  $k_1$  and lower measures of  $k_2$ . On the contrary, if the values of  $k_1$  and  $k_2$  become smaller and larger, respectively, exploration is intensified [50].

## 4.3 Simplified inclined planes system optimization

This algorithm was presented by Mohammadi-Esfahrood et al. in 2019 [51]. The SIPO has improved the performance and simplification of the standard IPO from two perspectives of improving the execution mechanism and establishing a favorable compromise between the exploration and extraction parameters.

- **Mechanism simplification**

In SIPO, the following equation is proposed to update acceleration:

$$\ddot{\bar{p}}_{i,G}^d = \frac{\dot{\bar{p}}_{i,G}^d - \dot{\bar{p}}_{mean,G}^d}{\Delta G} \tag{37}$$

where  $\dot{\bar{p}}_{mean,G}^d$  is defined by Eq. (38):

$$\dot{\bar{p}}_{mean,G}^d = \frac{\bar{p}_{best,G}^d - \bar{p}_{mean,G}^d}{\Delta G} \tag{38}$$

By placing  $\dot{\bar{p}}_{i,G}^d$  and  $\dot{\bar{p}}_{mean,G}^d$  in Eq. (37), Eq. (39) will be obtained (for simplifying, the power of 2 parameter  $\Delta G$  is ignored):

$$\ddot{\bar{p}}_{i,G}^d = \frac{\bar{p}_{mean,G}^d - \bar{p}_{i,G}^d}{\Delta G} \tag{39}$$

where  $\bar{p}_{mean,G}^d$  is the average position of the search agents with a better fitness than the ball  $i$ . It is obtained from Eq. (40):

$$\bar{p}_{mean,G}^d = \frac{\bar{p}_{Sbetter_i,G}^d}{S_{i,G}} \tag{40}$$

$$S_{i,G} = s_{1,G} + s_{2,G} + \dots + s_{i,G} \tag{41}$$

where  $\bar{p}_{\text{better},i,G}^d$  is the sum of the position of the balls with better fitness than the ball  $i$  in step  $G$ . Also,  $s_1$ ,  $s_2$ , and  $s_i$  represent the number of balls with better fitness than the first ball (the first member of a population), the second ball and the ball  $i$  in step  $G$ , respectively.

Finally, the  $P_{\text{mean}}$  factor to control the mean position is included in Eq. (39):

$$\ddot{p}_{i,G}^d = \frac{P_{\text{mean}}\bar{p}_{\text{mean},G}^d - \bar{p}_{i,G}^d}{\Delta G} \tag{42}$$

That  $P_{\text{mean}}$  is defined as Eq. (43):

$$P_{\text{mean}} = F \times \frac{(G_{\text{max}})}{G} \tag{43}$$

where  $G$ ,  $G_{\text{max}}$ , and  $F$  represent the current iteration, the total number of iterations, and a control volume of  $P_{\text{mean}}$ , respectively.

• **Exploration & exploitation improvement**

The parameters  $k_1$  and  $k_2$  in SIPO are defined according to Eqs. (44) and (45):

$$k_{1,G} = \left(\frac{1}{G}\right)^\beta \tag{44}$$

$$k_{2,G} = \frac{c}{1 + e^{-(G-T)}} \tag{45}$$

That

$$T = m_{\text{ratio}} \times G_{\text{max}} \tag{46}$$

where  $\beta$ ,  $c$ , and  $m_{\text{ratio}}$  are the convergence controller, a constant, and indicating the ratio of  $T$  from  $G_{\text{max}}$ , respectively [51].

### 5 The proposed approach

The INS/GNSS integration problem is a nonlinear problem. We look for an optimal nonlinear filter in the form of intelligently estimating the values of the process and measurement noise covariance matrices ( $Q$  &  $R$ ) for accurately determining the navigation response. Therefore, the proposed solution is to improve the efficiency of the Kalman integration filter by the SIPO algorithm to design an optimal navigation system.

The optimization is done in two ways: one is the optimal estimation of Kalman parameters, and the other is the definition of an intelligent mechanism in the objective function of the problem. The proposed approach improves the performance of the INS/GNSS system only by adjusting the noise covariance matrices in the integrated navigation algorithm without changing the structure of the

assumed integration problem simulated in [52]. Therefore, the basis of comparison is based on the results extracted from the reference navigation system in [52].

To validate the performance and outputs of the proposed approach, the outputs are compared with the standard and modified versions of IPO and MIPO, as well as two well-known competitor algorithms, GA and PSO, that the results of them have been previously reported in [69]. The implementation considerations of the assumed INS/GNSS problem are entirely consistent with [52]. All data and route specifications, inertia sensors, GPS, navigation and integration mechanism, calculations, problem-solving relations, etc., are also described in detail in [52] and included in its MATLAB source.

In the proposed approach, instead of replacing the values obtained from the GPS error profile in the main diameter of the measurement matrix  $R$  and instead of inserting the noise and bias values of each sensor in the process covariance matrix  $Q$ , the values estimated by the proposed metaheuristic methods are used. Given that in [52], the root-mean-square error (RMSE) for both actual and simulated INS systems per MEMS IMU module are compared and reported, the sum of these two errors as a weighted sum function (with the same weight and simple algebraic sum as Eq. (47)) is minimized as a single-objective function of the problem.

$$\text{objective} = \text{sum}([\text{RMSE}(\text{IMU1}), \text{RMSE}(\text{IMU2})]) \tag{47}$$

where RMSE is the difference between the simulated INS/GNSS system and the actual registered reference values as well as the independent GNSS system with the reference for the two sets of IMU based on MEMS technology in [52]. RMSE values include the difference of all values of performance parameters, including latitude and longitude, altitude, roll, pitch, yaw, and velocity in three directions.

The flowchart of Fig. 2 shows the general framework of the proposed approach. The purpose of optimizing matrices ( $Q$ ) and ( $R$ ) is to estimate their elements so that the fitness error of Eq. (47) is minimized, so the search agents' vector is defined as follows:

$$X^T = [x_1^R, x_2^R, \dots, x_6^R, x_7^Q, \dots, x_N^Q] \tag{48}$$

The solution is an  $N$ -dimensional vector.  $N$  is the number of search agents and is equal to the sum of the main diameters, two measurement and process noise covariance matrices, and is considered in the proposed optimization algorithms.

The pseudo-code of the proposed INS/GNSS integration algorithm (based on SIPO algorithm) is given as follows (**Algorithm II**):

**Algorithm II** Pseudo-code of the proposed SIPO-based INS/GNSS integration algorithm

**1:** Initialize algorithm control parameters such as number of *Iterations* (*num.of.runs*), design variables (equivalent to length/*Dimensions* of vector  $X$  equal to  $N$  (*num.of.dims*) = 18), control variables of exploration and exploitation ( $c_1$ ,  $c_2$ ,  $scale_1$ ,  $scale_2$ ,  $shift_1$ ,  $shift_2$ ) and finally determine the selectable range of variables ( $Min\_var$  &  $Max\_var$ ).

**2:** Generate a random initial population with dimensions  $N$  and the number of *numofballs* containing the noise covariance values of  $R$  and  $Q$  proposed by the algorithm (Eq. (48)).

**3:** Calling the objective function (Eq. (47)); Starting the NaveGo algorithm ("Algorithm 2" in [52]).

**4:** Executing "Algorithm 2 in [52]" up to step 10 and then inserting the proposed values of  $R$  &  $Q$  and continuing its steps from 11 to 30 (performing optimal INS/GNSS mechanization)

**5:** Extracting the fitness values of the objective function

**6:** Updating fitness indicators (mean, best, and worst)

**7:** Check stop criteria (obtaining minimum error (zero) or maximum iteration):

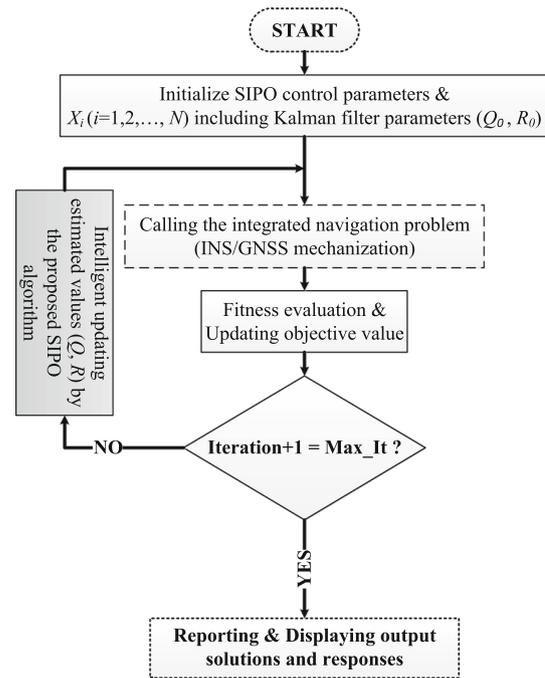
**8:** If the stop criteria are not met: Add the iteration,

**9:** Intelligently updating the values of the design variables and reset from **Step 3**.

If the stop criteria are met: Extract the runtime values, print the estimated values of the process and measurement covariance noise matrices  $Q$  &  $R$ , and plot trajectory and accuracy/error diagrams of the performance indicators of the problem.

## 5.1 Pros and cons of the proposed approach

The main strengths and distinguishing features of the present work compared to other related researches are as follows: (1) novel metaheuristic optimization approaches for designing integrated navigation systems, (2) overcoming the high volume of calculations and theoretical considerations, and operational assumptions of the standard INS/GNSS problem, (3) intelligent estimation of covariance noise matrix of an EKF and achievement of an optimal navigation filter; (4) implement the structure offline and achieve a powerful and robust integration algorithm for online applications (by increasing the executive iterations



**Fig. 2** Optimal design of the INS/GNSS integrated navigation problem using the proposed SIPO algorithm

of the proposed algorithms), and (5) utilizing statistical and performance indicators.

Also, among the weaknesses of the present work, the following can be mentioned: (1) full reliance on the calculations and considerations of the assumed basic problem in [52], (2) the need for powerful hardware to complete the offline intelligent design process, (3) requirement of understanding and knowledge of the assumed algorithms and proper adjustment of their control parameters, and (4) finally, not using other types of Kalman filters, such as CKF, UKF, etc. in the proposed structure.

## 5.2 Computational complexity of the proposed approach

For calculating the computational complexity of the proposed approach based on IPO algorithm, four main steps should be taken into account. In the beginning of the algorithm, all initial population  $NP$  must be randomly generated and evaluated. So, the computational complexity of the initialization step is  $O(NP)$ . In the second step, according to all  $NP$  number of population, operations are performed to calculate the angle between the balls, acceleration, velocity, update the control variables  $k_{1,G}$  and  $k_{2,G}$ , and then update the positions and evaluate their fitness, respectively, which corresponds to  $O(NP)$  complexity.

The algorithm may need to be evaluated between 0 or 1 times (if the best current value is better than before) to update the best fitness values (positions). Therefore, the

computational complexity of the third process is between  $O(0)$  and  $O(1)$ , regarding the quality of solutions. It should be noted that the first step is performed only once and the second and third steps are repeated as many times as the number of iterations ( $G$ ). Putting these together, the computational complexity of the proposed algorithm is between  $O(NP(G + 1))$  and at most  $O(NP + G(NP + 1))$ .

## 6 Results and analyses

In this section, outputs are reported according to the following indicators and criteria: trajectory (in terms of latitude, longitude, and altitude), attitude (roll, pitch, and yaw), errors of roll, pitch and yaw, changes in velocity (received from GPS including components north velocity  $V_N$ , east velocity  $V_E$ , and vertical velocity  $V_D$ ), velocity errors, variations in latitude, longitude, and altitude, and finally errors of latitude, longitude, and altitude. All of these are extracted by simulating the movement of a vehicle on a specific trajectory [52], and in three formats (REF or True (correct vehicle route), single GPS output, and using two modules IMU1 and IMU2) are displayed distinctly and symmetrically.

Here, the INS/GNSS navigation is formulated as a single-objective optimization problem and is simulated for all the proposed SIPO in comparison with IPO and MIPO along with GA and PSO algorithms. Due to the very high volume of data collected in the reference problem [52], implementations have led to a lot of time spent. Therefore, several computer systems have been used. Simulations have been carried out in MATLAB (versions R2015b and R2019b) on a laptop and three other computer systems with the following specifications: Intel (R) Core (TM) i3-2348 M CPU@2.30 GHz, 6 GB RAM under Windows 7 Ultimate; Intel (R) Core (TM) i5-6500 CPU@3.20 GHz, 8 GB RAM under Windows 10 Pro; Intel (R) Core (TM) i7-4790 CPU@3.60 GHz, 16 GB RAM under Windows 10 Pro; Intel (R) Core (TM) i5-3570 CPU@3.40 GHz, 20 GB RAM under Windows 10 Pro.

Each algorithm is executed for five independent trials/runs with a different number of iterations. Graphic results are presented for the best trial of each algorithm and numerical and statistical analyzes for all of them. Iterations are selected to estimate the operational success of each method for low to high iterations. Control values have been adopted based on numerous experimental results and similar studies. In PSO, the inertia coefficient  $w$  decreases linearly with a friction factor of  $w_{damp}$  corresponding to the iteration step of the algorithm.

The range of variations of the design variables in the form of vectors of minimum and maximum selectable intervals (equal to 18 variables including the main diameters

of the matrices  $R$  and  $Q$ , respectively) in Eqs. (49) and (50) is considered in a specific and limited way:

$$\text{Min\_var} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1e^{-10} & 1e^{-10} & 1e^{-10} & 1e^{-10} & 1e^{-10} \\ 1e^{-10} & 1e^{-10} \end{bmatrix}; \quad (49)$$

$$\text{Max\_var} = [1 \ 1 \ 1 \ 100 \ 100 \ 100 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]; \quad (50)$$

In the analysis of the results, the best numerical values are displayed in bold. The Outputs are reported in the form of estimated values of design variables ( $X^T$  of Eq. (48)), convergence curves, statistical analysis of fitness values of the objective function, execution times, numerical values of performance criteria of the problem, all based on 5 trials of all algorithms, and graphical results for the best trial of each algorithm, respectively.

Due to the long-running time, it is reported by the hour. The values reported for the performance criteria of the problem represent the output accuracy of the INS/GNSS integrated system optimized by the proposed intelligent algorithms. So that the value in each row corresponding to the column of an algorithm, indicates the system error in terms of the parameters of attitude (roll, pitch, and yaw), velocities, and position (latitude, longitude, and altitude).

Figure 3 shows the estimated design variables per 5 runs for all algorithms. The convergence curves of the algorithms for 5 trials are also shown in Fig. 4.

Figure 5 shows the best routing (trajectory) of each algorithm during the 5 independent runs.

Finally, in Table 1, to evaluate the overall performance of the algorithms, a final ranking for each algorithm is presented. Table 1 shows the numerical superiority of the statistical indicators of the algorithms (based on each of the performance criteria and dividing by the total number of them). This analysis represents the overall performance of each algorithm in terms of a combination of statistical indicators and performance metrics that confirm the success and superiority of algorithms, respectively.

In Table 1, the results of the total rating calculations of each algorithm based on the use of each of the inertial navigation systems IMU1 and IMU2 along with the ratings obtained from the statistical results of runtimes and fitness values of the objective function and finally an overall and final ranking are reported in the last row. As can be seen in Table 1, the final ranking indicates that the success and operational superiority of the algorithms in the given integrated navigation problem has been achieved through the use of IPO, PSO, SIPO, GA, and finally, MIPO methods, respectively.

As a general assessment of the results, it is clear that all motivations and policies of the current research are well

met, some of which are listed in detail below: (a) achieving optimal results and proper performance outputs from the assumed integrated navigation problem than the volume of considerations and calculations of techniques and heavy theories to improve the performance of these problems indicates the success of the proposed method for using in this regard and also overcoming the challenge of optimal design of such problems for the first time, (b) the tangible superiority of the standard IPO over its modified and simplified versions and other competing methods due to the embedding of a precise exploration and exploitation mechanism through control with 6 parameters, and performing a continuous and powerful navigation process during its execution, (c) presenting and reviewing various performance aspects of algorithms and integrated navigation problem along providing comprehensive statistical analysis on the operation of metaheuristic optimization algorithms for integrated navigation systems, (d) finding the operational gaps in improved versions of IPO algorithm (arising from inaccuracies in improving the computational complexity of the proposed versions and the lack of execution of a continuous and powerful process in exploration and exploitation mechanisms that have led to unfavorable results compared to the standard version and competing methods), (e) estimation of performance stability and reliability of the proposed approaches by performing a variety of independent trials, as well as providing multiple rankings based on fitness, runtime, for IMU1 & IMU2 modules, and overall ranking, and, finally, (f) covering various types of soft computing techniques based on evolutionary and metaheuristic processes and providing statistical and functional analysis to comprehensively assess how they function and their desirability or lack of

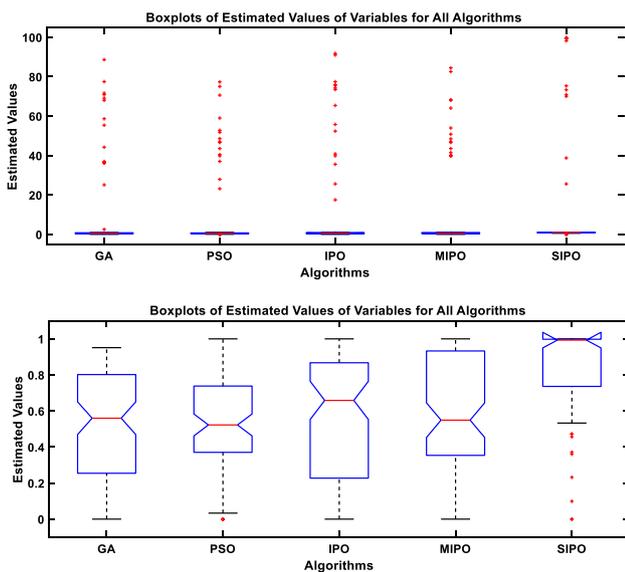


Fig. 3 Estimated design variables for all algorithms

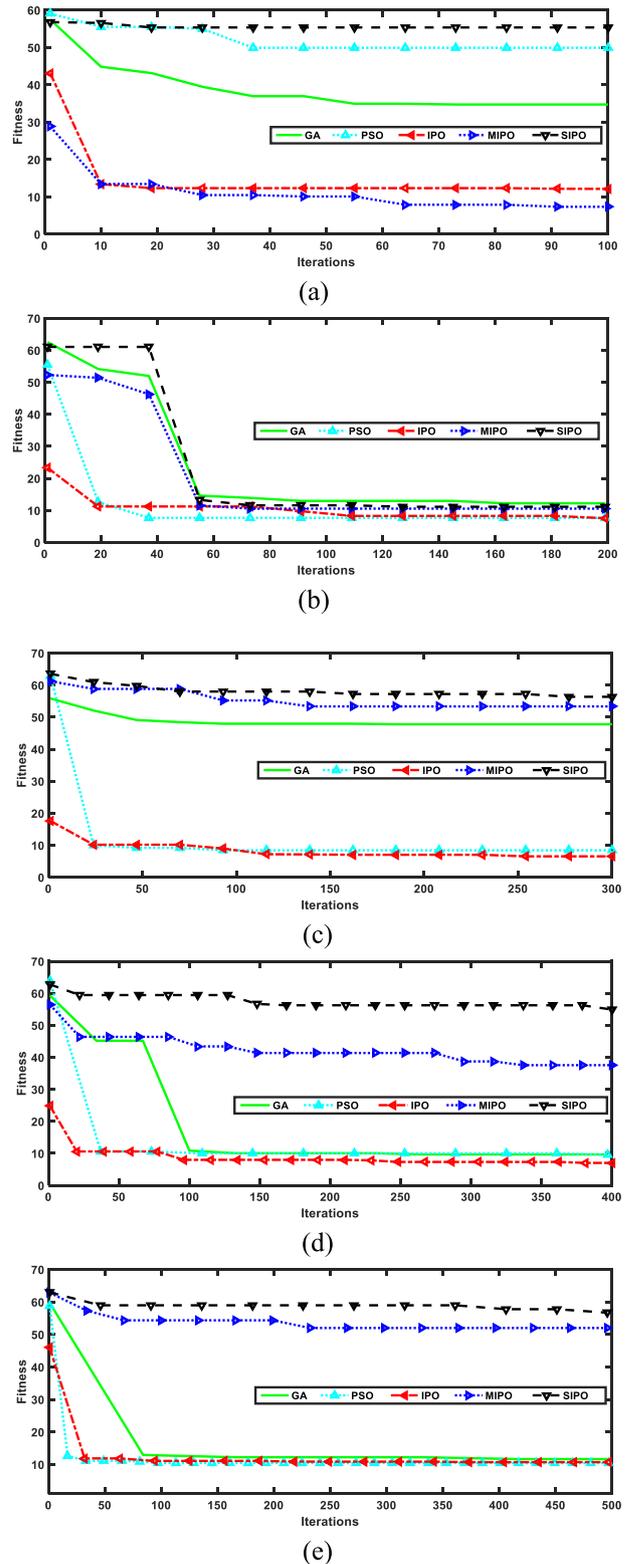
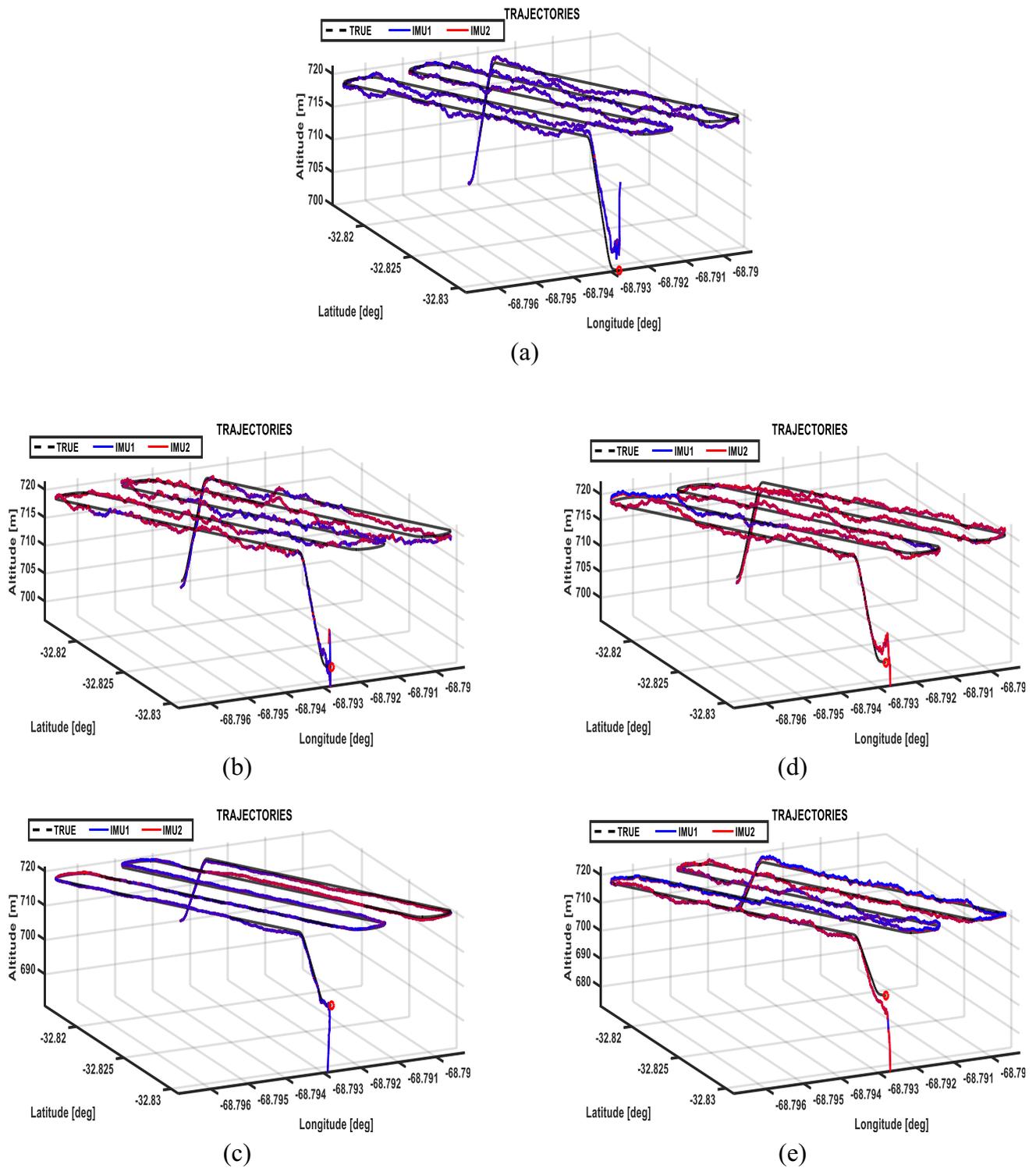


Fig. 4 Convergence curves of algorithms for iterations a 100, b 200, c 300, d 400, and e 500



**Fig. 5** Trajectories of optimized navigation systems by the best trial of **a** GA, **b** PSO, **c** IPO, **d** MIPO, **e** and the proposed SIPO algorithm

desirability in using them in designing and smartening integrated navigation systems.

## 7 Conclusions

In this research, the intelligent optimization of INS/GNSS integrated navigation systems was investigated and analyzed based on a reference problem using two IMU

**Table 1** Performance ranking of algorithms based on the total statistical results

Algorithms → Performance ranking ↓	GA [69]	PSO [69]	IPO [69]	MIPO [69]	SIPO
Fitness ranking	12/4	11/4	<b>4/4</b>	16/4	21/4
Runtime ranking	23/4	16/4	14/4	21/4	<b>6/4</b>
Final ranking for IMU1	171/60	202/60	<b>149/60</b>	223/60	198/60
Final ranking for IMU2	233/60	196/60	<b>165/60</b>	236/60	201/60
Total ranking	439/128	425/128	<b>332/128</b>	492/128	430/128

modules with MEMS sensor technology. The implementations and simulations were performed under MATLAB, and the resulting outputs were presented and analyzed in detail.

The basis of this work is the use of intelligent methods and algorithms of evolutionary and metaheuristic optimization based on artificial intelligence. In such a way that acceptable results can be extracted with minimal operational considerations, and these approaches and estimated results can be implemented operationally. Therefore, for the first time, the capabilities and robustness of the SIPO algorithm were used along with two standard and modified versions of IPO and MIPO. Also, the results were evaluated using two well-known and common evolutionary algorithms, GA and PSO. The performance of the proposed method was measured to intelligently estimate the numerical values of process and measurement noise covariance matrices  $R$  and  $Q$ . From the general assessment, it can be acknowledged that the use of metaheuristic IPO approaches to optimally design and smartize integrated navigation systems is successful and can be a good candidate compared to the computational volume of other mathematical and theoretical methods.

Some suggestions for future work include: (1) replacing navigation algorithms with techniques based on reinforcement or deep learning; (2) utilizing multi-objective optimization versions of evolutionary and metaheuristic algorithms; (3) using the potential and capabilities of other intelligent algorithms; (4) applying hybrid approaches such as fuzzy logic and artificial neural networks to improve performance criteria of navigation problems.

## Appendix

Note that variables with subscript (-) correspond to the previous sample (such as  $t_{(k-1)}$ ), and (+) to the next sample (such as  $t_{(k+1)}$ ); the tilde symbol (-) is for noisy measurements, and the estimated variables based on these measurements also have a hat (^); Finally, an entry-wise product is expressed by the symbol '○'.

All source codes are fully and publicly available at <https://github.com/ali-ece>

**Data availability statement** All data generated or analyzed during this study are included in this published article (and its supplementary information files).

## Declarations

**Conflict of interest** The authors declare that there are no conflicts of interest.

## References

1. Bar-Shalom Y, Li XR, Kirubarajan T (2004) Estimation with applications to tracking and navigation: theory algorithms and software, 1st edn. Wiley
2. Brown RG, Hwang PYC (2012) Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions, 4th edn. Wiley, Hoboken
3. Farrell J (2008) Aided navigation: GPS with high rate sensors, 1st edn. McGraw-Hill Inc, New York
4. Minkler G, Minkler J (1993) Theory and applications of Kalman filtering. Magellan Book Company
5. Khan N, Jabbar A, Bilal H, Gul U (2020) Compensated closed-loop Kalman filtering for nonlinear systems. Measurement 151:107129
6. Cui Y, Wang Z (2021) "Auto-calibration Kalman filters for nonlinear systems with direct feedthrough. IET Control Theory Appl 15(6):890–899
7. Urrea C, Agramonte R (2021) Kalman filter: historical overview and review of its use in robotics 60 years after its creation. J Sens 2021:9674015
8. Shakhtarin BI, Shen K, Neusylin KA (2016) Modification of the nonlinear kalman filter in a correction scheme of aircraft navigation systems. J Commun Technol Electron 61(11):1252–1258
9. Zheng B, Fu P, Li B, Yuan X (2018) A robust adaptive unscented kalman filter for nonlinear estimation with uncertain noise covariance. Sensors 18(3):1–15
10. Psiaki ML (2005) Backward-smoothing extended Kalman filter. J Guid Control Dyn 28(5):885–894
11. Nguyen NH, Doğançay K (2017) Improved pseudolinear Kalman filter algorithms for bearings-only target tracking. IEEE Trans Signal Process 65(23):6119–6134
12. Chang L, Hu B, Chang G, Li A (2013) Robust derivative-free Kalman filter based on Huber's M-estimation methodology. J Process Control 23(10):1555–1561
13. Rigatos G, Siano P, Zervos N, Cecati C (2015) Control and disturbances compensation for doubly fed induction generators using the derivative-free nonlinear Kalman filter. IEEE Trans Power Electron 30(10):5532–5547

14. Fang J, Gong X (2010) Predictive iterated Kalman filter for INS/GPS integration and Its application to SAR motion compensation. *IEEE Trans Instrum Meas* 59(4):909–915
15. Bai Y-T, Wang X-Y, Jin X-B, Zhao Z-Y, Zhang B-H (2020) A neuron-based Kalman filter with nonlinear autoregressive model. *Sensors (Basel)* 20(1):299
16. Julier SJ, Uhlmann JK (1997) New extension of the Kalman filter to nonlinear systems. *Signal Process Sens Fus Target Recognit VI* 3068:182–193
17. Ristic B, Arulampalam S, Gordon N (2004) Beyond the Kalman filter: Particle filters for tracking applications. Illustrate. Artech Print on Demand
18. Sathiya V, Chinnadurai M (2019) Evolutionary algorithms-based multi-objective optimal mobile robot trajectory planning. *Robotica* 37(8):1363–1382
19. Jalali SMJ, Khosravi A, Kebria PM, Hedjam R, Nahavandi S (2019) Autonomous robot navigation system using the evolutionary multi-verse optimizer algorithm. In: 2019 IEEE international conference on systems, man and cybernetics (SMC), 2019, pp. 1221–1226.
20. Liu Q, Li Y, Liu L (2020) A 3D simulation environment and navigation approach for robot navigation via deep reinforcement learning in dense pedestrian environment. In: 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), 2020, pp. 1514–1519
21. Cong L, Yue S, Qin H, Li B, Yao J (2020) Implementation of a MEMS-based GNSS/INS integrated scheme using supported vector machine for land vehicle navigation. *IEEE Sens J* 20(23):14423–14435
22. Yi J-H, Lu M, Zhao X-J (2020) Quantum inspired monarch butterfly optimisation for UCAV path planning navigation problem. *Int J Bio-Inspired Comput* 15(2):75–89
23. Bellemare MG et al (2020) Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature* 588(7836):77–82
24. Al Bitar N, Gavrilov AI (2020) Neural networks aided unscented Kalman filter for integrated INS/GNSS systems. In: 2020 27th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS), 2020, pp. 1–4.
25. Wang J, Ma Z, Chen X (2021) Generalized Dynamic Fuzzy NN Model Based on Multiple Fading Factors SCKF and its Application in Integrated Navigation. *IEEE Sens J* 21(3):3680–3693
26. Gul F, Rahiman W, Alhady SSN, Ali A, Mir I, Jalil A (2021) Meta-heuristic approach for solving multi-objective path planning for autonomous guided robot using PSO–GWO optimization algorithm with evolutionary programming. *J Ambient Intell Humaniz Comput* 12(7):7873–7890
27. Zieliński P, Markowska-Kaczmar U (2021) 3D robotic navigation using a vision-based deep reinforcement learning model. *Appl Soft Comput* 110:107602
28. Gao D, Lyu X, Qin F, Chang L, Hu B (2021) A real time gravity compensation method for high precision INS based on neural network. In: 2021 28th Saint Petersburg international conference on integrated navigation systems (ICINS), 2021, pp. 1–5
29. Wen S, Wen Z, Zhang D, Zhang H, Wang T (2021) A multi-robot path-planning algorithm for autonomous navigation using meta-reinforcement learning based on transfer learning. *Appl Soft Comput* 110:107605
30. Al Bitar N, Gavrilov A (2021) A novel approach for aiding unscented Kalman filter for bridging GNSS outages in integrated navigation systems. *Navigation* 68(3):521–539
31. Yan F, Li S, Zhang E, Guo J, Chen Q (2021) An adaptive nonlinear filter for integrated navigation systems using deep neural networks. *Neurocomputing* 446:130–144
32. Wu Y (2021) A survey on population-based meta-heuristic algorithms for motion planning of aircraft. *Swarm Evol Comput* 62:100844
33. Silva C, Ribeiro B (2003) Navigating mobile robots with a modular neural architecture. *Neural Comput Appl* 12(3):200–211
34. Braga APS, Araújo AFR (2003) A topological reinforcement learning agent for navigation. *Neural Comput Appl* 12(3):220–236
35. Do Q, Jain L (2010) Application of neural processing paradigm in visual landmark recognition and autonomous robot navigation. *Neural Comput Appl* 19(2):237–254
36. Kan EM, Lim MH, Ong YS, Tan AH, Yeo SP (2013) Extreme learning machine terrain-based navigation for unmanned aerial vehicles. *Neural Comput Appl* 22(3):469–477
37. El-Shafie A, Najah A, Karim OA (2014) Amplified wavelet-ANFIS-based model for GPS/INS integration to enhance vehicular navigation system. *Neural Comput Appl* 24(7):1905–1916
38. Guo C, Li F, Tian Z, Guo W, Tan S (2020) Intelligent active fault-tolerant system for multi-source integrated navigation system based on deep neural network. *Neural Comput Appl* 32(22):16857–16874
39. Hodge VJ, Hawkins R, Alexander R (2021) Deep reinforcement learning for drone navigation using sensor data. *Neural Comput Appl* 33(6):2015–2033
40. Fu C et al (2022) Memory-enhanced deep reinforcement learning for UAV navigation in 3D environment. *Neural Comput Appl* 34(17):14599–14607
41. Luo Q, Shao Y, Li J, Yan X, Liu C (2022) A multi-AUV cooperative navigation method based on the augmented adaptive embedded cubature Kalman filter algorithm. *Neural Comput Appl* 34(21):18975–18992
42. Millán-Arias C, Fernandes B, Cruz F (2022) Proxemic behavior in navigation tasks using reinforcement learning. *Neural Comput Appl*
43. Herrera EP, Kaufmann H (2010) Adaptive methods of Kalman filtering for personal positioning systems. In: Proceedings of the 23rd international technical meeting of the satellite division of the institute of navigation (ION GNSS 2010), 2010, pp. 584–589
44. Mohamed AH, Schwarz KP (1999) Adaptive Kalman filtering for INS/GPS. *J Geod* 73(4):193–203
45. Hide C, Moore T, Smith M (2003) Adaptive Kalman filtering for low-cost INS/GPS. *J Navig* 56(1):143–152
46. Hide C, Moore T, Smith M (2004) Adaptive Kalman filtering algorithms for integrating GPS and low cost INS. In: PLANS 2004. Position location and navigation symposium (IEEE Cat. No.04CH37556), 2004, pp. 227–233
47. Gao W, Yang Y, Cui X, Zhang S (2007) Application of adaptive Kalman filtering algorithm in IMU/GPS integrated navigation system. *Geo-spatial Inf Sci* 10(1):22–26
48. Zhang L, Wang S, Selezneva MS, Neusypin KA (2021) A new adaptive Kalman filter for navigation systems of carrier-based aircraft. *Chinese J Aeronaut (in Press)*
49. Mozaffari MH, Abdy H, Zahiri SH (2016) IPO: an inclined planes system optimization algorithm. *Comput Inform* 35(1):222–240
50. Mohammadi A, Zahiri SH (2017) IIR model identification using a modified inclined planes system optimization algorithm. *Artif Intell Rev* 48(2):237–259
51. Mohammadi-Esfahrood S, Mohammadi A, Zahiri SH (2019) A Simplified and efficient version of inclined planes system optimization algorithm. In: 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), 2019, pp. 504–509
52. Gonzalez R, Giribet JI, Patino HD (2015) NaveGo: a simulation framework for low-cost integrated navigation systems. *J Control Eng Appl Inform* 17(2):110–120

53. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Publishing Company
54. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, 1995, vol 4, pp. 1942–1948.
55. Georgy JFW (2010) Advanced nonlinear techniques for low cost land vehicle navigation. Queen's University
56. Abdul Rahim K (2012) Heading drift mitigation for low-cost inertial pedestrian navigation. University of Nottingham
57. Holland JH (1975) Adaptation in natural and artificial systems. The University of Michigan Press
58. Dorigo M, Maniezzo V, Colomi A (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern Part B*. 26(1):29–41
59. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
60. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. *Inf Sci (Ny)* 179(13):2232–2248
61. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11(4):341–359
62. Price K, Storn RM, Lampinen JA (2005) Differential evolution: a practical approach to global optimization. Springer-Verlag, Berlin Heidelberg
63. Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. *IEEE Trans Evol Comput* 15(1):4–31
64. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
65. Lin C-W, Zhang B, Yang K-T, Hong T-P (2014) Efficiently hiding sensitive itemsets with transaction deletion based on genetic algorithms. *Sci World J* 2014:398269
66. Lin JC-W, Liu Q, Fournier-Viger P, Hong T-P, Voznak M, Zhan J (2016) A sanitization approach for hiding sensitive itemsets based on particle swarm optimization. *Eng Appl Artif Intell* 53:1–18
67. Djenouri Y, Djenouri D, Belhadi A, Fournier-Viger P, Lin JC-W (2018) A new framework for metaheuristic-based frequent itemset mining. *Appl Intell* 48(12):4775–4791
68. Lin JCW, Lv Q, Yu D, Srivastava G, Chen CH (2022) Adaptive particle swarm optimization model for resource leveling. *Evol. Syst*
69. Mohammadi A, Sheikholeslam F, Emami M (2022) Metaheuristic algorithms for integrated navigation systems. In: *Computational Intelligence for Unmanned Aerial Vehicles Communication Networks*, First Edit., M. Ouaisa, I. U. Khan, M. Ouaisa, Z. Boulouard, and S. B. Hussain Shah, Eds. Cham: Springer International Publishing, 2022, pp. 45–72

**Publisher's Note** All data generated or analyzed during this study are included in this published article (and its supplementary information files).

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.