Journal of Advanced Marine Engineering and Technology, Vol. 49, No. 6, pp. 570~580, 2025
J. Advanced Marine Engineering and Technology (JAMET)
https://doi.org/10.5916/jamet.2025.49.6.570

ISSN 2234-7925 (Print)
ISSN 2765-4796 (Online)
Original Paper

An adaptive pressure genetic algorithm with a long immigration strategy based on directional selection for estimating low-cost inertial measurement unit errors

Minh Thao Pham[1] · Kyoung-Kuk Yoon†

(Received December 16, 2025 ; Revised December 21, 2025 ; Accepted December 21, 2025)

Abstract: Low-cost inertial measurement unit (LIMU) is key to optimizing cost and size for orientation estimation applications. However, the reliability of LIMU raw measurement is significantly reduced by the influence of various types of errors. Traditional calibration methods often use gradient-based algorithms to estimate LIMU errors with a small dataset. When higher LIMU accuracy is required, the number of datasets is increased. These algorithms suffer from reduced performance. Therefore, it is necessary to develop a highly precise algorithm to estimate errors of multiple datasets. This work proposed an adaptive pressure genetic algorithm (APGA) with a long immigration strategy (LIS) based on directional selection to estimate errors of 72 positions of a 3-axis accelerometer (ACCE) and continuous rotation of a 3-axis gyroscope (GYR). The proposed algorithm is compared to traditional AGA (TAGA), APGA, and elitist-based APGA (EAPGA) with a similar parameter set. The results showed that the proposed algorithm has better performance than other algorithms in all aspects: stability, exploitation, and exploration across 30 trials. ACCE and GYR measurements, after error removal, are fused by a Kalman filter and tested by an inverted pendulum set. The results of this test show that error estimation using the proposed algorithm allows LIMU to achieve high accuracy.

Keywords: Adaptive genetic algorithm, Directional selection, Long immigration strategy, Low-cost inertial measurement unit error

1. Introduction

Improving the accuracy of low-cost inertial measurement units (LIMU) have achieved many interests and attention because they are more and more friendly with customers by their wide range of applications in daily life, such as motion detection, self-balancing vehicles, measuring attitude, etc. Most internal elements affecting the precision of LIMU can be eliminated by calibration in initialization. They contain bias, non-orthogonality, misalignment, and scale factors. Generally, calibration includes sampling methods, equipment, and algorithms. Sampling methods are developed independently for each sensor in LIMU. A most basic method is to place accelerometers (ACCE) at rest on the flatted surface for sensing local earth gravity. This method is simple, time-saving, and effective for linear-based applications. However, if the plane that ACCE is placed on forms an arbitrary angle with the horizontal at the start of calibration, this angle will remain in ACCE's output until turning off the power. Therefore, multi-position methods are proposed to collect ACCE's sample in many orientations, such as 4 **[1]**, 6 **[2]**, 8 **[3]**, 12 **[4]**,18 **[5]**, 20 **[6]**, 24 **[7]**, and 29 positions **[8]**. The collected orientations increase caused a significant improvement in accuracy whereas increasing the complexity of the equipment. Multi-face equipment crafted by available resources or 3D-printed equipment is convenient and budget-saving for users. ACCE calibration method is feasible for gyroscope (GYR) calibration. However, only bias and scale factors are estimated, whereas misalignment, non-orthogonality have remained **[9]**. Applying rotation on each axis is a common and effective method for GYR calibration. Rotatable equipment is integrated with a high-precision angular velocity sensor as a reference because GYR at a low-cost level cannot sense Earth's rotation speed, which sinks under the noise and error of GYR. **[3][4][10]** proposed GYR calibration by a turning table, which includes high-precision sensors and a mechanical system. However, this is extremely high-cost equipment. Reference **[7]** proposed a manually driven rate table integrated measurement system based on a fiber optic gyroscope instead of an

† Corresponding Author (ORCID: http://orcid.org/0000-0001-8612-9574): Professor, Division of Maritime AI & Cyber Security, Korea Maritime & Ocean University, 727, Taejong-ro, Yeongdo-gu, Busan 49112, Korea, E-mail: kkyoon@kmou.ac.kr, Tel: +051-410-4265

1 Ph. D. Candidate, Maritime AI & Cyber Security, National Korea Maritime & Ocean University, E-mail: sqrtthao96@g.kmou.ac.kr

auto-turning table. Although the budget has been saved significantly, common users still find it difficult to own this equipment. Reference **[2]** takes advantage of available chair for rotation. However, a reference created by a simultaneous localization and mapping system based on a laser scanner is hard to implement. This work proposed 3D-printed equipment integrated with an encoder for serving 72-position data for ACCE calibration and rotation for GYR calibration.

Algorithms play an essential role in error estimation. In reference **[4][6]**, sensor error models (SEM) are estimated by the Levenberg-Marquardt algorithm (L-MA) with the least square fitness function (FF). Although L-MA performed well in stability, the extensibility is limited by FF dependence. Quasi-Newton nonlinear optimization algorithm (Q-NNOA) in **[7]** has improved the drawback of L-MA and increased the computational speed because the Hessian matrix is approximated instead of calculated exactly for each output. However, both L-MA and Q-NNOA face difficulty in high-variable and large-data-point problems due to significant computational cost. Furthermore, noisy data can cause the missing global optimum for them. SEM estimation is a multi-object problem. Therefore, it is necessary to use global optimization algorithms such as particle swarm optimization (PSO) **[11][12]**, differential evolution (DE) **[13]-[15]**, and genetic algorithm (GA) **[16]**. PSO is based on swarm intelligence. Hence, this algorithm is suitable for over 50-dimensional problems and straightforward to implement. Fast convergence is an advantage but also is a disadvantage of PSO due to premature convergence. Algorithms based on the evolution approach, such as DE or GA, can solve the premature convergence problem of PSO by using crossover and mutation operators to balance the diversity of the population. DE is powerful in exploration. Therefore, they can take a long time to reach a near-optimal area, leading to the limitation in addressing iteration-limited problems. GA has good performance in multi-dimensional (M-D), nonlinear, and noisy problems. Additionally, GA is feasible and compatible with many algorithms to improve efficiency, exploration, and exploitation, and avoid local optimum. Many publications proposed hybrid GA to take advantage of the global search of GA and the local search of tabu search **[17]-[19]**, statistical filter **[20]**, variance neighborhood search **[21][22]**, K-means **[23]**, k-nearest neighbors **[24]**, and moving least square **[25]**. However, this combination requires significant computational resources when local search algorithms (LSA) are repeated many times. Furthermore, the imbalance between global search and local search causes premature convergence due to excessive LSAs. **[26]** proposed adaptive LS schemes for HGA: the iterative hill climbing method based on average fitness value and the similarity coefficient method. These schemes have better performances in stability and speed convergence than traditional GAs and HGAs. However, their schemes had just been tested on benchmark functions. Adaptive GA (AGA) **[27]-[32]**, which are another approach, also gains much attention. The principle of AGA is based on an adaptation of crossover and mutation rate to save computational time and prevent premature convergence. However, stability can be reduced significantly if adaptive strategy is poor. Reference **[28]** has improved stability of AGA by creating feasible mutation point (FMP) space. After that, the FF of each FMP is calculated to bring out the best optimal FMP for the corresponding iteration. This strategy is expensive in computational cost for complex problems. Immigration strategy (IS) is a potential candidate for GA to remain stable. Random immigration (RI) **[33]**, elitism-based immigration (EI) **[34]**, and hybrid immigration (HI) **[35]-[38]** are the main approaches in IS. RI randomly replaces weak individuals with new individuals in each iteration to increase the diversity of the population. However, a slowly changing environment can cause a decrease in convergence speed. In EI, this limitation has been addressed by replacing weak individuals of the recent iteration with the good individuals of the nearest preceding iteration. However, the difference between individuals is reduced, leading to premature convergence. Therefore, HI takes advantage of both RI and EI and limits their disadvantages to make GA feasible in dynamic environments. This work proposes an adaptive pressure genetic algorithm (APGA) with a long LS (LIS) based on directional selection to address the aforementioned problems.

The effectiveness of the proposed algorithm was validated through estimating SEM from the dataset of 72 positions of a triaxial ACCE and continuous rotation of a triaxial GYR. The results show that the proposed algorithm has outperformed in convergence speed, exploration capability, and stability compared to other conventional GAs.

2. Low-cost Inertial Measurement Unit

2.1 Sensor Error Model Parameters

The main advantages of LIMU are reasonable cost, small size, high sampling rate, and high durability. However, their efficiency of orientation estimation is reduced significantly due to many kinds of errors in raw data. There are two typical types: varying

and unvarying errors. The causes of varying errors are random noise, temperature changes, environmental variations, and drift. They are often solved by compensation. Whereas unvarying errors can be eliminated in calibration by estimating SEM parameters, including bias, non-orthogonality, misalignment, and scale factors. Defining $\widehat{f^a} = \left(f_x^a, f_y^a, f_z^a\right)^T$ and $\widehat{Fg} = \left[g_{ij}\right]_{3x3}$,where $i,j \in \{x,y,z\}$, as the calibrated measurements of ACCE and GYR, respectively. According to unvarying errors, the SEM of LIMU can be represented in **Equation (1)**.

$$\widehat{f^a} = K^a\left(\overline{f^a} - b^a\right),$$
$$\widehat{Fg} = K^g\left(\overline{Fg} - B^g\right) \quad (1)$$

where K^a and K^g represent the unvarying error matrix, $\overline{f^a} = \left(f_{ux}^a, f_{uy}^a, f_{uz}^a\right)^T$ and $\overline{Fg} = \left[g_{uij}\right]_{3x3}$ are the raw measurements of ACCE and GYR, $b^a = \left(b_{ax}, b_{ay}, b_{az}\right)^T$ and $B^g = \left[b_{ij}\right]_{3x3}$ are the bias.

K^a contains non-orthogonal matrix N^a and scale factor matrixS^a, whereas K^g contains non-orthogonal matrix N^g, misalignment matrix M^g, and scale factor matrix S^g. K^a and K^g are defined as **Equation (2).**

$$K^a = N^a S^a,$$
$$K^g = M^g N^g S^g \quad (2)$$

Where $N^a = \begin{bmatrix} 1 & 0 & 0 \\ \alpha_{yx} & 1 & 0 \\ \alpha_{zx} & \alpha_{zy} & 1 \end{bmatrix}$, $S^a = diag\left(s_{ax}, s_{ay}, s_{az}\right)$,

$M^g = \left[\gamma_{ij}\right]_{3x3}, \gamma_{ii} = 1, N^g = \begin{bmatrix} 1 & 0 & 0 \\ \beta_{yx} & 1 & 0 \\ \beta_{zx} & \beta_{zy} & 1 \end{bmatrix}$,

$S^g = diag\left(s_{gx}, s_{gy}, s_{gz}\right)$.

2.2 Data Acquisition

The dataset of ACCE and GYR is collected from a motion processing unit (MPU), MPU9250, which is a typical LIMU. The specifications of MPU9250 are shown in **Table 1**.

The MPU9250 is attached to a 3D-printed device as shown in **Figure 1**. The device contains two main parts: platform and base. The Platform is used to fix MPU9250. The shaft surface of this platform is designed as a regular 12-sided polygon. The base has two pairs of parallel supports. They differ in the section that connects to the platform's shaft. The coordinate system of the first pair is parallel to the base coordinate system, whereas the second pair is rotated 15 degrees relative to the x-axis of the base's coordinate system. When

Table 1: The specification of MPU9250

Features	Parameter	Typical	Units
ACCE 3-axis	Full-Scale Range	±16	g
	Initial tolerance	±3	%
	Nonlinearity	±0.5	%
	Cross-axis	±2	%
	Zero-G	X, Y: 60 Z: 80	mg
	Noise PSD	300	mg/Hz$^{1/2}$
GYR 3-axis	Full-Scale Range	±2000	°/s
	Initial tolerance	±3	%
	Nonlinearity	±0.1	%
	Cross-Axis	±2	%
	Initial ZRO	±5	%
	Rate Noise SD	0.01	°/s/Hz$^{1/2}$

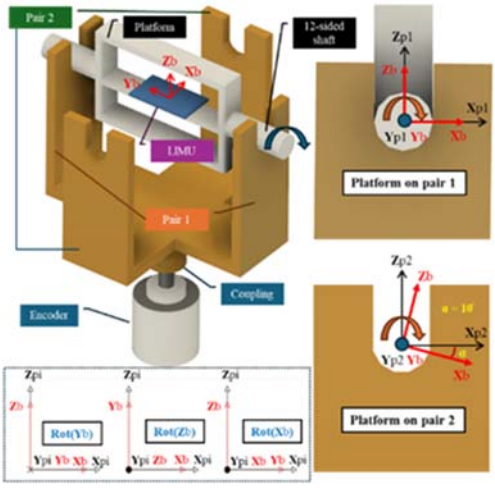

Figure 1: A 3D-printed device and the LIMU placement positions

connecting the platform to each supporting pair, 24 different positions are provided. The dataset of 72 positions for ACCE calibration are distributed equally for X, Y and Z axes of the MPU9250. **Figure 1** illustrates the MPU9250 placement positions used for collecting ACCE measurements. The initial position of MPU9250 is chosen so that the body frame of sensor C_b is parallel to the coordinate frame of the platform C_p. By rotating each face of the platform's shaft and placing it on the two pairs of supports as shown in **Figure 1**, a dataset of 24 positions of MPU9250's Y-axis is collected as illustrated in **Figure 2**. Then, the MPU9250 placement is sequentially changed so that its X-axis and Z-axis are aligned with the Y-axis of pairs 1 and 2, respectively. At each position, MPU9250 is kept stably in 5 seconds to obtain roughly 500 samples. To reduce the effect of varying errors, it is necessary to take the average of each dataset.

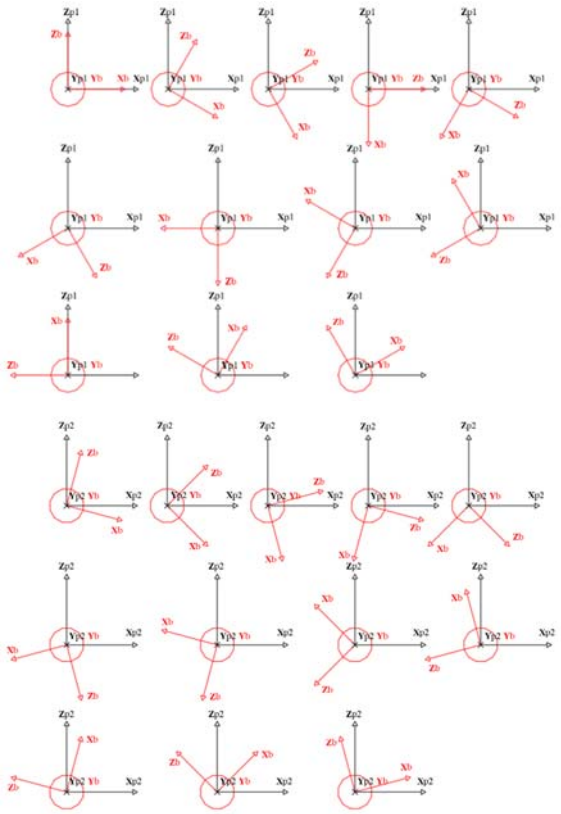

Figure 2: A set of 24 positions of Y-axis for ACCE calibration

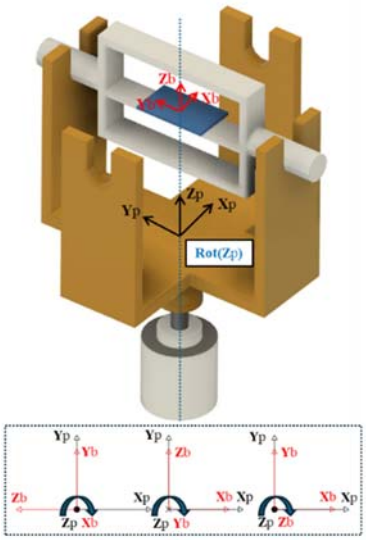

Figure 3: The MPU9250 placement for GYR calibration

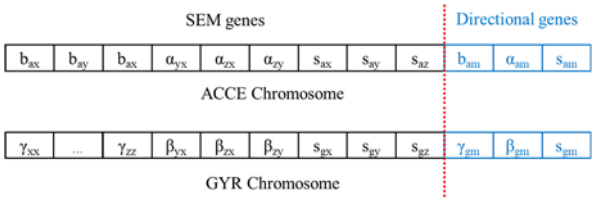

Figure 4: The chromosomes of ACCE and GYR

The non-compensated matrix of 72 positions for ACCE calibration is formed $\overline{F^a} = \left[\overline{f_{ul}^a}\right]_{72x3}$. The data set of GYRs contains three separate rotation measurements of X, Y, and Z axes. **Figure 3** illustrates the position of MPU9250 placement for GYR calibration. Before applying rotation, 100 samples of the stable state are measured to eliminate B^g element. Around 1000 rotation samples are used to estimate K^g element.

3. Integrations of Adaptive Pressure Genetic Algorithms with A Long Immigration Strategy Based on Directional Selection

3.1 A Long Immigration Strategy Based on Directional Selection

3.1.1 Directional Selection

According to classical natural selection, there are three types of selection in population evolution: directional, stabilizing, and disruptive selection. Stabilizing selection focuses on individuals with average genes, those that are not particularly outstanding in any specific pattern. This type limits the diversity of the population and the ability to adapt to an ever-changing environment. Disruptive selection favors the development of individuals at two opposite extremes and eliminates the intermediate group. The population diversity is very high, but the population becomes fragmented. Directional selection causes the population to evolve in a beneficial direction according to environmental conditions. **Figure 4** shows the chromosomes of ACCE and GYR. Each chromosome contains two gene series. SEM genes are the parameters of SEM that have to be estimated. Directional genes (DE) are sequences of traits that are beneficial to the population. These traits are calculated as follows:

$$d1_{nm} = mean(d1_{ni}),$$
$$d2_{nm} = mean(d2_{ij}) \quad (3)$$

where $d1_{nm}$ and $d2_{nm}$ are the DEs, $d1 \in \{b, s\}$, $d2 \in \{\alpha, \gamma\}$, $n \in \{a, g\}$.

3.1.2 A Long Immigration Strategy

IS has been developed for many years to increase the performance of GA. The general principle of IS is to insert or replace some weak individuals in a recent population with special individuals. The special individuals can be random individuals **[33]** for enhancing

exploration or elitist individuals **[34]** of the nearest iteration for increasing exploitation. In a dynamic environment, it is necessary to insert both random and elitist individuals **[35,36,37,38]** to serve an ever-changing environment. In LIMU calibration, the measurements were collected over a short and stable period, so it can be assumed that it is a slowly changing environment. Therefore, LIS in this work has inherited a part of EI. That means N elitist individuals of the current iteration are stored in a p^{th} pool. But unlike traditional EI, the weaker individuals in the i^{th} iteration are replaced by N elitist individuals of the $(i-1)^{th}$ iteration; these individuals will be inserted from the x^{th} iteration as calculated in **Equation (4).** This interval has increased with the current number of iterations.

$$x = round(i - \frac{i_{max} * i}{i_{max} + i}) \tag{4}$$

where i is the current iteration, i_{max} is the maximum iteration.

3.1.3 A Long Immigration Strategy Based on Directional Selection

Figure 5 illustrates the principle of LIS based on directional selection. N elitist individuals are randomly distributed into three groups. Each group uses a DE as the criterion for evaluating the quality of individuals. Half of the weaker individuals in each group are eliminated. N/3 remaining individuals are ready for immigration. This operator plays an essential role in accelerating convergence and saving computational resources. Since the elitist individuals have immigrated from a long iteration with the current iteration, the diversity of the population is preserved.

3.2 Adaptive Pressure Genetic Algorithm

GA is a typical candidate for M-D problems. GA contains many algorithms known as operators, such as the generator, selection, crossover, and mutation operators. Traditional GA uses a fixed parameter set, which requires a significant amount of time to tune in order to obtain a good set. Therefore, AGA is more popular than traditional GA. AGA automatically balances the ratio between exploration and exploitation over time. This leads to a significant improvement in convergence speed, flexibility, and

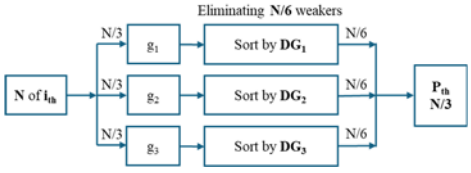

Figure 5: The principle of LIS based on directional selection

characteristic dependence of FF. **Figure 6** illustrates the principle of the APGA with LIS based on directional selection. This proposed algorithm starts from the initialization operator, which generates a random population with I number of individuals. Each individual is represented by a gene series known as a chromosome. There are two typical ways to represent genes: binary numbers and real numbers. Binary encoding brings many advantages in crossover and mutation because its structure is easy to implement. However, when the dataset is real value, the performance of binary encoding is limited. Real values require many bits to be represented to achieve high precision. Additionally, the computational speed is significantly decreased when converting between binary numbers and real numbers. Therefore, real-coded encoding is used for chromosome representation. The chromosome is shown in **Figure 4**. After that, the quality of each individual is evaluated by FF, which is the root mean square error (RMSE) between measurements and references. It is assumed that in the non-motion state, only the local Earth gravity $g_e = (0,0,g)^T$, where g = 9.81 m/s^2, exists in the measurements of ACCE. When C_b is not aligned with the local frame, ACCE will measure the gravity components distributed across the axes according to the tilt angle. However, since g_e remains constant, the norm of g_e can be used for reference of ACCE. The reference for GYR calibration in this work is measurements of a high-revolution encoder, because the Earth rate is approximately 15°/hour, which is much lower than random noise and bias instability of GYR. From **Equation (1)** and the reference signal, FF is calculated as **Equation (5).**

$$RMSE_a = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\left\|\widehat{f_n^a}\right\| - \|g_e\|\right)^2},$$
$$RMSE_g = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(\widehat{F_m^g} - W_m\right)^2} \tag{5}$$

where $W_m = \left[\omega_{ij}\right]_{3x3}$ is the encoder measurement, N is the total number of ACCE measurements, M is the total number of GYR measurements.

From the 2nd iteration, N/3 elitist individuals stored in the p_x pool are inserted into the population to replace the N/3 weakest individuals. After that, the new population has evolved by the key operators of AGA. This work has designed adaptive rules for all three operators: selection, crossover, and mutation. The adaptive strategy for selection based on the roulette wheel **[39]** involves promoting a group of individuals, referred to as the encouraged group (EG), whereas diminishing the remainder, known as the "diminished group" (DIG).

It is hypothesized that the EG accounts for 75% of the total probability of the entire population. It is worth noting that the population's diversity must be preserved to avoid getting trapped in a local optimum. Therefore, the EG should constitute no more than 90% of the total probability of the population. A selection pressure coefficient, denoted as η ranging from 0.2 to 2, is responsible for ensuring this condition is met. When there is a large gap between the highest value $RMSE_{max}$ and lowest value $RMSE_{min}$ in the population, the convergence rate has been increased by raising η. Conversely, η is reduced to improve exploration capability when approaching the global optimum. The variation of η is calculated as **Equation (6)**.

$$\eta_i = e^{\frac{i_{max}-i}{i_{max}+i}\left(\frac{RMSE_{imax}-RMSE_{imin}}{RMSE_{imax}+RMSE_{imin}}\right)},$$
$$\eta_{min} \le \eta_i \le \eta_{max},$$
$$\eta_i \ge \eta_{max} \rightarrow \eta_i = \eta_{max},$$
$$\eta_i \le \eta_{min} \rightarrow \eta_i = \eta_{min} \quad (6)$$

where $\eta_{min} = 0.2$ and $\eta_{max} = 2$ are the minimum and maximum values of η at iteration i.

By combining this adaptive rule with the traditional probability equation of the roulette wheel [39], we obtain **Equation (7)**.

$$\rho_{pr} = \exp\left(-\eta_i \frac{f_i}{\sum_{i=1}^{I} f_i}\right) \quad (7)$$

where ρ_{pr} is the probability after applying selection pressure, f_i is the FF value of the i^{th} individual.

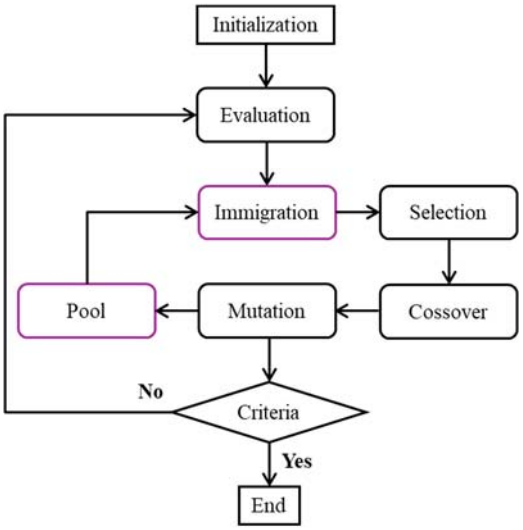

Figure 6: The principle of the APGA with LIS based on directional selection

Blend crossover (BLX) and Gaussian mutation are used to operate real-valued encoding smoother. BLX produces a new individual with its gene series c_i, which is a combination of two parent gene series, c_{i1} and c_{i2}, as calculated in **Equation (8)**. The part $\phi \in [-b, 1+b]$, where $b \le 0.15$, increases the evolved ability of offspring. Then c_i has been mutated by adding a generated random value of their original gene series using normal distribution $\mathcal{N}(0,\sigma)$, where $\sigma \in [-0{,}05, 0.05]$. the mutated offspring c_i' is calculated in **Equation (9)**.

$$c_i = c_{i1} \cdot \phi + c_{i2} \cdot (1-\phi) \quad (8)$$
$$c_i' = c_i + c_i \cdot \mathcal{N}(0,\sigma) \quad (9)$$

Saving computational resources is also a criterion that this work has focused on; therefore, the adaptive rules for crossover coefficient p_c and mutation coefficient p_m depend not only on the ratio of the FF values but also on the number of iterations. P_c and p_m are calculated in **Equations (10)** and **(11)**, respectively. After mutation stages, N elitist individuals are filtered by directional selection to remain N/3 elitist individuals, which have been stored in the pool. The process is repeated until the criteria are satisfied.

$$p_c = p_{cmax} - k_c \frac{i}{i_{max}} \exp\left(\frac{RMSE_{imax}-RMSE_{imin}}{RMSE_{imax}+RMSE_{imin}}\right),$$
$$p_{cmin} \le p_c \le p_{cmax},$$
$$p_c \ge p_{cmax} \rightarrow p_c = p_{cmax},$$
$$p_c \le p_{cmin} \rightarrow p_c = p_{cmin} \quad (10)$$

$$p_m = (1+k_m) \frac{i}{i_{max}} \exp\left(\frac{RMSE_{imax}-RMSE_{imin}}{RMSE_{imax}+RMSE_{imin}}\right),$$
$$p_{mmin} \le p_m \le p_{mmax},$$
$$p_m \ge p_{mmax} \rightarrow p_m = p_{mmax},$$
$$p_m \le p_{mmin} \rightarrow p_m = p_{mmin} \quad (11)$$

where p_{cmin}, p_{cmax} and p_{mmin}, p_{mmax} are the minimum and maximum values of p_c and p_m, respectively, k_c and k_m are the adaptive parameters.

4. Results and Discussion

4.1 Sensor Error Model Parameter Estimation

We perform the estimation using the proposed algorithm and other algorithms, including TAGA, APGA, EAPGA. Each algorithm is tested 30 independent times with the same conditions as shown in **Table 2**.

Figure 7 shows the RMSE between calibrated ACCE measurements and the reference signal. The best result of TAGA is 1.76e-3 g on the 13th time. However, among the 30 trials, only the results of the 6th and 10th times converge to this value. The maximum deviation is around 6.2e-4 g. It can be seen that the output across trials is not stable if there are only two adaptive rules for crossover and mutation. Increasing pressure in selection operators has significantly improved the results of APGA. The RMSE in the APGA test is 1.52e-3 g, whereas there are 14 of 30 times in which the RMSE approximately reached this value. The deviation is reduced by 25.7%, down to 4.63e-4 g. The EI strategy enhances EAPGA's exploitation, as most of the results converge to 1.47e-3 g. However, this strategy also increases the risk of getting trapped in local optima. APGA with a LIS selects elitist individuals through directional selection from iterations far earlier and inserts them into the current population. Therefore, population diversity is still maintained in the latter half of the process. Over 30 trials, the RMSE estimated by APGA with a LIS consistently remained around 1.41e-3 g, with the deviation not exceeding 2.44e-5 g.

Figure 8 displays the RMSE between calibrated GYR measurements and encoder values. A decreasing trend from left to right is observed in all columns: Best RMSE, mean, and worst RMSE.

Table 2: The parameters of algorithm

Parameters	Values
Number of individuals	150
Number of iterations	200
Pressure Coefficient	[0.2, 2]
P_c	[0.2, 0.8]
P_m	[0.05, 0.08]
k_c	0.6
k_m	0.02
Number of runs	30

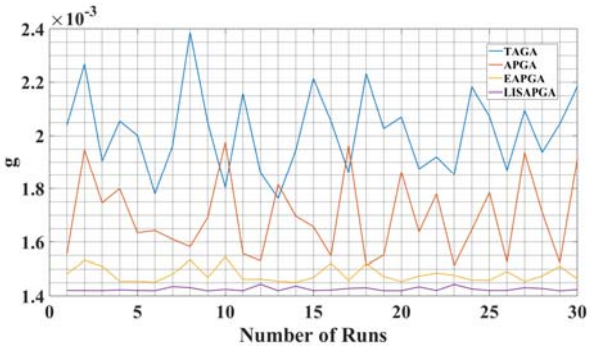

Figure 7: The best RMSE over 30 trials in the case of ACCE error estimation

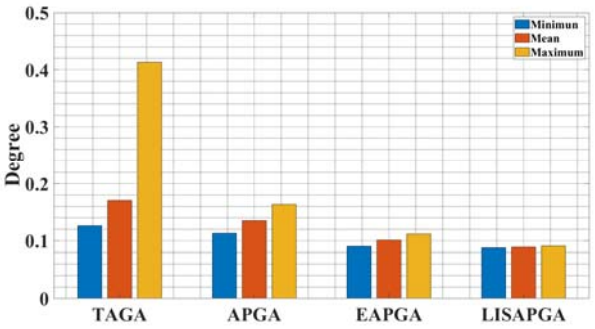

Figure 8: The best RMSE over 30 trials in the case of GYR error estimation

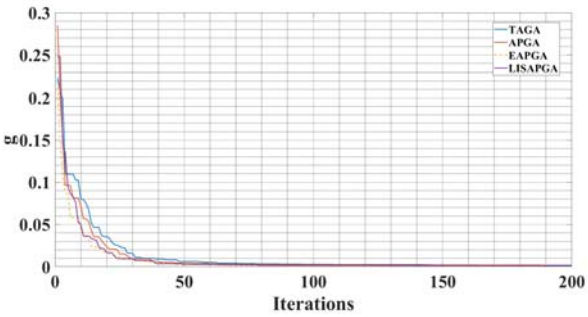

Figure 9: The evolution of RMSE over 200 iterations in the case of ACCE error estimation

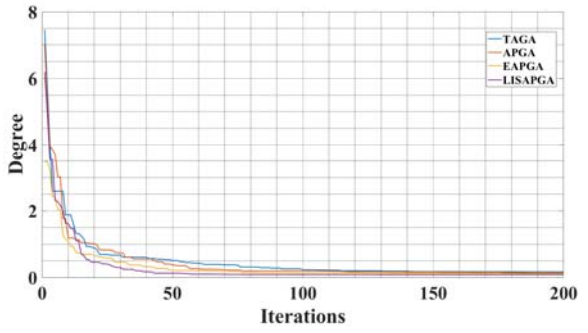

Figure 10: The evolution of RMSE over 200 iterations in the case of GYR error estimation

Figures 9 and **10** show the evolution of RMSE over 200 iterations of ACCE and GYR, respectively. Referring to **Figure 9**, in 50 early iterations, EAPGA has the best performance in convergence rate and then slows down in the middle and final stages of the process. APGA with a LIS consistently follows closely behind EAPGA in 170 early iterations. From the 171st iteration, APGA with a LIS takes the lead due to better exploration than the other algorithms. Referring to **Figure 10**, as the number of model parameters in the case of GYR error estimation increases more than ACCE error estimation, APGA with

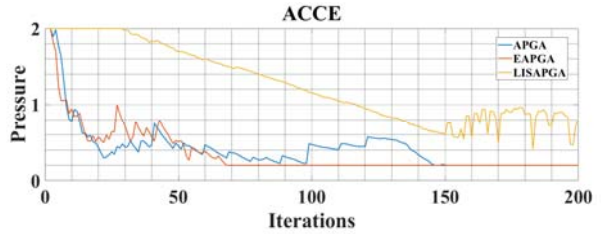

Figure 11: The variation of pressure coefficient in the case of ACCE error estimation

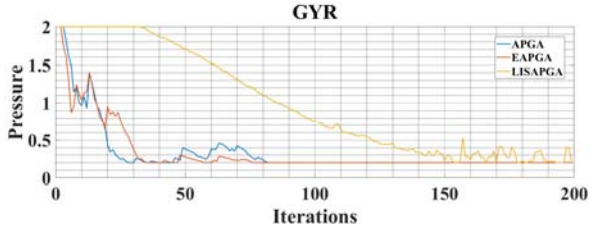

Figure 12: The variation of pressure coefficient in the case of GYR error estimation

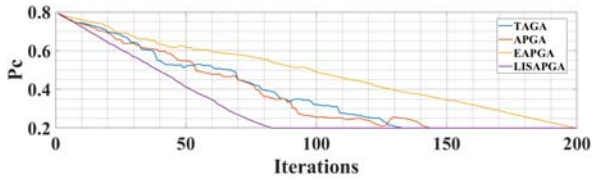

Figure 13: The variation of p_c in the case of ACCE error estimation

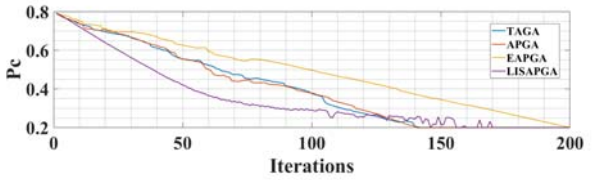

Figure 14: The variation of p_c in the case of GYR error estimation

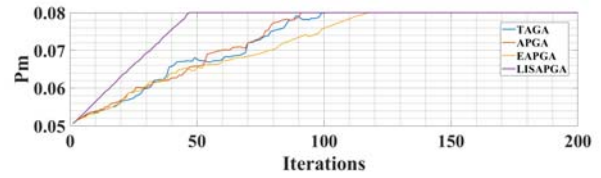

Figure 15: The variation of p_m in the case of ACCE error estimation

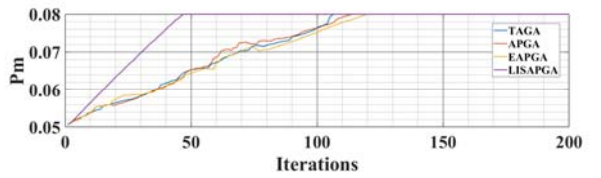

Figure 16: The variation of p_m in the case of GYR error estimation

a LIS demonstrates greater superiority in exploitation and exploration, becoming the lead as early as the 16th iteration.

Figures 11 and **12** illustrate the variation of pressure coefficient over 200 iterations in the case of ACCE and GYR, respectively. Referring to **Figure 11**, the pressure coefficient of EAPGA has significantly decreased in the early 20th iteration. In the next 47 iterations, the rate of decrease is slow to reach the minimum value at the 68th iteration. That means at this point, EAPGA has started to focus on exploitation instead of exploration. APGA has experienced similar trends to EAPGA. However, the convergence speed of APGA is much slower than EAPGA; it takes more iterations to reach the minimum pressure value. In APGA with a LIS result, this parameter reaches its maximum value at 2 within the first 30 iterations, then gradually decreases over the following 120 iterations. At the 150th iteration, there is a significant fluctuation, with a deviation ranging from 0.5 to 1. This situation means that APGA with a LIS has found a better solution area.

Figures 13 and **14** show the variation of P_c over 200 iterations. Referring to **Figure 13**, it is seen that the value of P_c of APGA with a LIS performs a steeper decline compared to other algorithms. P_cs of TAGA, APGA, EAPGA, and APGA with a LIS have reached the minimum value at 132nd, 144th, 200th, and 82nd, respectively.

Figures 15 and **16** illustrate the variation of P_m over 200 iterations.

Referring to **Figures 11** and **13**, it can be seen that APGA effectively leverages the adaptive rules in the selection operator and LIS with the directional operator to increase the convergence rate instead of relying on P_c as other algorithms do. A considerable number of computational resources is saved as the dependence on P_c decreases, because P_c decides the number of parent couples that undergo crossover and mutation in each iteration.

4.2 Pendulum Test Results

An inverted pendulum set is used for testing the performance of MPU9250 estimated error parameters by APGA with a LIS. This set contains three main parts: an inverted pendulum, shaft, and motor part, as shown in **Figure 17**. The motor is connected to the pendulum through a crank mechanism to convert the rotary motion into oscillatory motion. An encoder is connected to the shaft of the pendulum by a coupling to measure the angle of the pendulum. The MPU9250 is attached to the top of the pendulum. The oscillatory motion of the pendulum corresponds to the rotational motion around the X-axis. To limit the effect of varying errors and smooth measurements, a

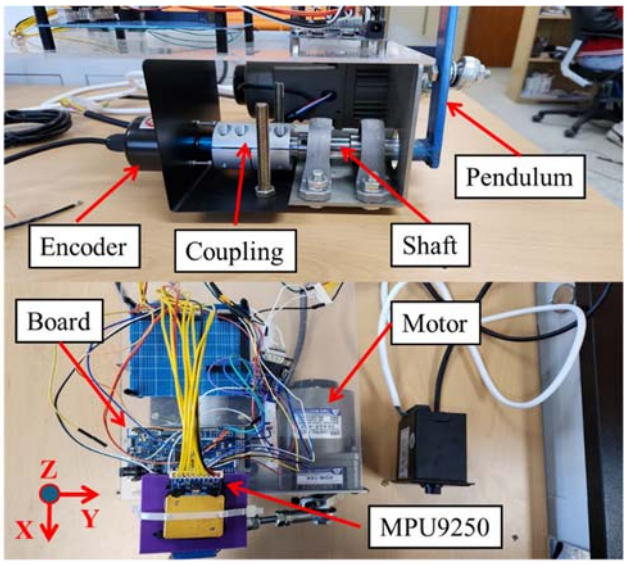

Figure 17: An inverted pendulum set

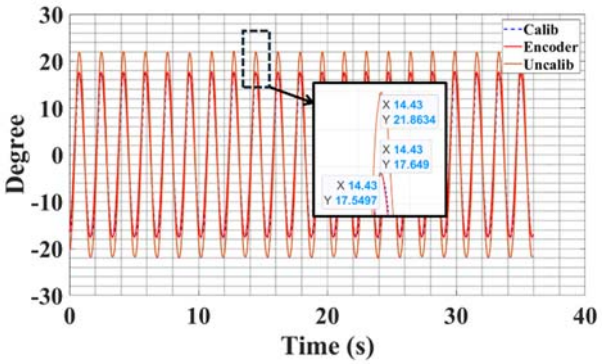

Figure 18: The measurements of MPU9250 before and after calibration compared to the encoder

Kalman filter is used to fuse the output of ACCE and GYR. Parameters of a priori estimate covariance P_0, covariance of process noise Q_0, and covariance of observation noise R_0 are 10e-3, 10e-5, and 5e-5, respectively.

Figure 18 displays the measurements of MPU9250 before and after calibration compared to the encoder. The largest deviation is observed near the peak region of both halves of the motion. This value between the uncalibrated measurement and the encoder reached 4.21°, whereas it was just 0.01° in the case of the calibrated measurement. Throughout the entire experiment, the calibrated value closely follows the encoder, with the error kept within approximately 0.5°.

5. Conclusion

In this work, LIMU errors are estimated by the APGA with LIS based on directional selection.

The DIY device is designed to obtain measurements of ACCE and GYR. The dataset of ACCE includes 72 different positions in a stable state of 3 axes, while the dataset of GYR consists of rotational measurements for each axis. To save costs, this device is manually driven. In future work, we will develop a compact automated device capable of wireless data transmission to facilitate more convenient data collection.

LIS based on directional selection is proposed. This strategy inherits part of the principle of the EI strategy by inserting elitist individuals from previous iterations into the current iteration, therefore enhancing the exploitation of high-quality solutions. Directional selection adds different criteria to select elitist individuals to increase the probability of evolving into new species.

An adaptive rule for selection operators is proposed to increase the pressure selection, thereby boosting the convergence rate up. This adaptive rule also contributes to reducing computational costs by indirectly declining the crossover rate.

The datasets are estimated by four algorithms: TAGA, APGA, EAPGA, and APGA with a LIS. TAGA is enhanced with adaptive selection to become APGA. EAPGA is a combination of APGA and EI. The EI of EAPGA is improved into LIS with directional selection to obtain APGA with a LIS. The analysis results showed that APGA with a LIS exhibits higher stability across multiple trials, fast convergence speed, and stronger exploration capability compared to the other algorithms.

Finally, the quality of the MPU9250 before and after estimating errors by APGA with a LIS has been tested in the inverted pendulum set. The results showed that the measurement of MPU9250 had a great performance compared to the encoder.

This work has contributed to improving the APGA with LIS based on directional selection for precisely estimating errors of LIMU. In the future, optimizing computational costs will be the main work to enable the deployment of this solution in various small-scale applications.

Acknowledgement

This research was supported by the 2024 Marine Police Field-Customized Research and Development (Ocean Lab 2.0) (No. RS-2024-00411499) funded by the Korea Coast Guard.

Author Contributions

Conceptualization, M. T. Pham and K.K Yoon; Methodology, M. T. Pham; Software, M. T. Pham; Validation, M. T. Pham and K.K Yoon; Formal Analysis, M. T. Pham; Investigation, M. T. Pham; Resources, M. T. Pham; Data Curation, M. T. Pham;

Writing—Original Draft Preparation, M. T. Pham; Writing—Review & Editing, K.K Yoon; Visualization, M. T. Pham; Supervision, K.K Yoon; Project Administration, K.K Yoon; Funding Acquisition, K.K Yoon.

References

[1] J. Lu, L. Ye, J. Zhang, W. Luo, and H. Liu, "A new calibration method of MEMS IMU plus FOG IMU," IEEE Sensors Journal, vol. 22, no. 9, pp. 8728-8737, 2022.

[2] J. Lv, A. A. Ravankar, Y. Kobayashi, and T. Emaru, "A method of low-cost IMU calibration and alignment," in 2016 IEEE/SICE International Symposium on System Integration (SII), pp. 373-378, 2016.

[3] M. Dong, G. Yao, J. Li, and L. Zhang, "Calibration of low cost IMU's inertial sensors for improved attitude estimation," Journal of Intelligent & Robotic Systems, vol. 100, no. 3, pp. 1015-1029, 2020.

[4] R. Zhang, F. Hoflinger, and L. M. Reind, "Calibration of an IMU using 3-D rotation platform," IEEE Sensors Journal, vol. 14, no. 6, pp. 1778-1787, 2014.

[5] U. Qureshi and F. Golnaraghi, "An algorithm for the in-field calibration of a MEMS IMU," IEEE Sensors Journal, vol. 17, no. 22, pp. 7479-7486, 2017.

[6] C. C. Peng, J. J. Huang, and H. Y. Lee, "Design of an embedded icosahedron mechatronics for robust iterative IMU calibration," IEEE/ASME Transactions on Mechatronics, vol. 27, no. 3, pp. 1467-1477, 2021.

[7] J. Rohac, M. Sipos, and J. Simanek, "Calibration of low-cost triaxial inertial sensors," IEEE Instrumentation & Measurement Magazine, vol. 18, no. 6, pp. 32-38, 2015.

[8] Z. Zou, L. Li, X. Hu, Y. Zhu, B. Xue, J. Wu, and M. Liu, "Robust equipment-free calibration of low-cost inertial measurement units," IEEE Transactions on Instrumentation and Measurement, vol. 73, pp. 1-12, 2024.

[9] H. A. Jlailaty, A. Celik, M. M. Mansour, and A. M. Eltawil, "IMU hand calibration for low-cost MEMS inertial sensors," IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1-16, 2023.

[10] F. Ghanipoor, M. Hashemi, and H. Salarieh, "Toward calibration of low-precision MEMS IMU using a nonlinear model and TUKF," IEEE Sensors journal, vol. 20, no. 8, pp. 4131-4138, 2020.

[11] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," Chemometrics and intelligent laboratory systems, vol. 149, pp. 153-165, 2015.

[12] A. G. Gad, "Particle swarm optimization algorithm and its applications: A systematic review," Archives of computational methods in engineering, vol. 29, no. 5, 2022.

[13] M. Pant, H. Zaheer, L. Garcia-Hernandez, and A. Abraham, "Differential evolution: A review of more than two decades of research," Engineering Applications of Artificial Intelligence, vol. 90, 103479, 2020.

[14] C. Li, G. Sun, L. Deng, L. Qiao, and G. Yang, "A population state evaluation-based improvement framework for differential evolution," Information Sciences, vol. 629, pp. 15-38, 2023.

[15] Q. Sui, Y. Yu, K. Wang, L. Zhong, Z. Lei, and S. Gao, "Best-worst individuals driven multiple-layered differential evolution," Information Sciences, vol. 655, 119889, 2024.

[16] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," Multimedia Tools and Applications, vol. 80, no. 5, pp. 8091-8126, 2021.

[17] M. S. Umam, M. Mustafid, and S. Suryono, "A hybrid genetic algorithm and tabu search for minimizing makespan in flow shop scheduling problem," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 9, pp. 7459-7467, 2022.

[18] W. Xu, Y. Hu, W. Luo, L. Wang, and R. Wu, "A multi-objective scheduling method for distributed and flexible job shop based on hybrid genetic algorithm and tabu search considering operation outsourcing and carbon emission," Computers & Industrial Engineering, vol. 157, 107318, 2021.

[19] T. Kaya and H. Guler, "A hybrid genetic algorithm for analog active filter component selection," AEU-International Journal of Electronics and Communications, vol. 86, pp. 1-7, 2018.

[20] R. K. Arakaki and F. L. Usberti, "Hybrid genetic algorithm for the open capacitated arc routing problem," Computers & Operations Research, vol. 90, pp. 221-231, 2018.

[21] X. Li, L. Gao, Q. Pan, L. Wan, and K. M. Chao, "An effective hybrid genetic algorithm and variable neighborhood search for integrated process planning and scheduling in a packaging machine workshop," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 10, pp. 1933-1945, 2018.

[22] X. Dong, H. Zhang, M. Xu, and F. Shen, "Hybrid genetic algorithm with variable neighborhood search for multi-scale multiple bottleneck traveling salesmen problem," Future Generation Computer Systems, vol. 114, pp. 229-242, 2021.

[23] D. Gribel and T. Vidal, "HG-means: A scalable hybrid genetic algorithm for minimum sum-of-squares clustering," Pattern Recognition, vol. 88, pp. 569-583, 2019.

[24] S. Jo, J. Y. Oh, J. Lee, S. Oh, H. S. Moon, C. Zhang, R. Gadh, and Y. T. Yoon, "Hybrid genetic algorithm with k-nearest neighbors for radial distribution network reconfiguration," IEEE Transactions on Smart Grid, vol. 15, no. 3, pp. 2614-2624, 2024.

[25] K. I. Lee, H. S. Oh, S. H. Jung, and Y. S. Chung, "Moving least square-based hybrid genetic algorithm for optimal design of W-band dual-reflector antenna," IEEE Transactions on Magnetics, vol. 55, no. 6, pp. 1-4, 2019.

[26] Y. Yun, "Hybrid genetic algorithm with adaptive local search scheme," Computers & Industrial Engineering, vol. 51, no. 1, pp. 128-141, 2006.

[27] V. H. Iyer, S. Mahesh, R. Malpani, M. Sapre, and A. J. Kulkarni, "Adaptive range genetic algorithm: a hybrid optimization approach and its application in the design and economic optimization of shell-and-tube heat exchanger," Engineering Applications of Artificial Intelligence, vol. 85, pp. 444-461, 2019.

[28] N. Sindhwani and M. Singh, "Comparison of adaptive mutation genetic algorithm and genetic algorithm for transmit antenna subset selection in MIMO-OFDM," International Journal of Computer Applications, vol. 97, no. 22, 2014.

[29] A. V. Mokshin, V. V. Mokshin, and L. M. Sharnin, "Adaptive genetic algorithms used to analyze behavior of complex system," Communications in Nonlinear Science and Numerical Simulation, vol. 71, pp. 174-186, 2019.

[30] J. Zhou, Q. Wu, M. Zhou, J. Wen, Y. Al-Turki, and A. Abusorrah, "LAGAM: A length-adaptive genetic algorithm with Markov blanket for high-dimensional feature selection in classification," IEEE Transactions on Cybernetics, vol. 53, no. 11, pp. 6858-6869, 2023.

[31] L. Wu and J. Qu, "AIMD rule-based duty cycle scheduling in wireless sensor networks using quartile-directed adaptive genetic algorithm," IEEE Sensors Journal, vol. 23, no. 5, pp. 4905-4921, 2023.

[32] C. Hua, "Solving the problem of vehicle routing problem with time window via dual adaptive genetic algorithm," IEEE Access, vol. 13, pp. 96535-96543, 2025.

[33] R. Tinós and S. Yang, "A self-organizing random immigrants genetic algorithm for dynamic optimization problems," Genetic Programming and Evolvable Machines, vol. 8, no. 3, pp. 255-286, 2007.

[34] S. Yang, "Genetic algorithms with elitism-based immigrants for changing optimization problems," In Workshops on Applications of Evolutionary Computation Springer Berlin Heidelberg, pp. 627-636, 2007.

[35] S. Yang and R. Tinós, "A hybrid immigrants scheme for genetic algorithms in dynamic environments," International Journal of Automation and Computing, vol. 4, no. 3, pp. 243-254, 2007.

[36] A. Panizo-Lledot, M. Pedemonte, G. Bello-Orgaz, and D. Camacho, "Addressing evolutionary-based dynamic problems: a new methodology for evaluating immigrants strategies in MOGAs," IEEE Access, vol. 10, pp. 27611-27629, 2022.

[37] P. Karthikeyan, "Genetic algorithm with self adaptive immigrants for effective virtual machine placement in cloud environment," International Journal of Intelligent Networks, vol. 4, pp. 155-161, 2023.

[38] P. Karthikeyan and S. Baskar, "Genetic algorithm with ensemble of immigrant strategies for multicast routing in Ad hoc networks," Soft Computing, vol. 19, no. 2, pp. 489-498, 2015.