Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Robust global route planning for an autonomous underwater vehicle in a stochastic environment*

Jiaxin ZHANG[1,2], Meiqin LIU‡[1,2,3], Senlin ZHANG[1,2], Ronghao ZHENG[1,2]

[1]*State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China*

[2]*College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China*

[3]*Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China*

E-mail: zhangjiaxin@zju.edu.cn; liumeiqin@zju.edu.cn; slzhang@zju.edu.cn; rzheng@zju.edu.cn

Received Jan. 20, 2022; Revision accepted Mar. 7, 2022; Crosschecked Mar. 23, 2022; Published online Apr. 29, 2022

Abstract: This paper describes a route planner that enables an autonomous underwater vehicle to selectively complete part of the predetermined tasks in the operating ocean area when the local path cost is stochastic. The problem is formulated as a variant of the orienteering problem. Based on the genetic algorithm (GA), we propose the greedy strategy based GA (GGA) which includes a novel rebirth operator that maps infeasible individuals into the feasible solution space during evolution to improve the efficiency of the optimization, and use a differential evolution planner for providing the deterministic local path cost. The uncertainty of the local path cost comes from unpredictable obstacles, measurement error, and trajectory tracking error. To improve the robustness of the planner in an uncertain environment, a sampling strategy for path evaluation is designed, and the cost of a certain route is obtained by multiple sampling from the probability density functions of local paths. Monte Carlo simulations are used to verify the superiority and effectiveness of the planner. The promising simulation results show that the proposed GGA outperforms its counterparts by 4.7%–24.6% in terms of total profit, and the sampling-based GGA route planner (S-GGARP) improves the average profit by 5.5% compared to the GGA route planner (GGARP).

Key words: Autonomous underwater vehicle; Route planning; Genetic algorithm; Orienteering problem; Stochastic path cost

https://doi.org/10.1631/FITEE.2200026 **CLC number:** TP242.2

1 Introduction

Monitoring the environment in an all-round way is often necessary in marine-related industries, such as submarine pipelines, oil exploration, harbor industry, and aquaculture (Cheng et al., 2021). Autonomous underwater vehicles (AUVs) that are maneuverable and can be equipped with multiple sensors have attracted great attention from researchers. AUV is a kind of underwater robot which is able to reach areas that are inaccessible to human beings and can finish complex tasks automatically. An AUV can adjust its actions timely based on environmental variations; hence, the route-planning strategy significantly affects the reliability and efficiency of the operation. An AUV is expected to carry out as many tasks as possible with limited battery energy (Han et al., 2021). Due to the complexity of the ocean, a comprehensive route-planning strategy is required to address the route-planning problem when the path cost is stochastic.

Many noteworthy works have been reported to deal with the autonomous vehicle route-planning

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China and Zhejiang Joint Fund for the Integration of Industrialization and Informatization (Nos. U1809212 and U1909206), the Fundamental Research Funds for the Zhejiang Provincial Universities (No. 2021XZZX014), and the National Natural Science Foundation of China (No. 62088102)

ORCID: Jiaxin ZHANG, https://orcid.org/0000-0003-3358-2206; Meiqin LIU, https://orcid.org/0000-0003-0693-6574

problem. The AUV route-planning problem has been represented as a combination of the travel salesman problem (TSP) and the knapsack problem (KP), where the vehicle is required to autonomously maximize the efficiency, i.e., using the limited battery capacity to obtain as much profit as possible (Mahmoud Zadeh et al., 2018). The problem has also been introduced as an orienteering problem (OP) (Chou et al., 2021), and it has been discussed in many fields, such as unmanned aerial vehicle (UAV) mission planning (Royset and Reber, 2009; Dorling et al., 2017) and tourist trip design problem (Vansteenwegen and van Oudheusden, 2007; Schilde et al., 2009). However, the methods proposed in the above studies do not perform well when facing the complex ocean environment. Given that OP is a non-deterministic polynomial (NP) hard problem (Bagagiolo et al., 2021), heuristic methods are supposed to be promising to deal with large and variant instances. A bi-level task planning strategy has been proposed to use metaheuristics to address the issue of team OP of autonomous surface vehicles (Sun et al., 2022). To solve OP with heterogeneous task characteristics, a two-phase heuristic approach has been proposed (Ji et al., 2021). Besides, the evolutionary algorithm has been combined with a greedy randomized adaptive search procedure to find the optimal route of OP (Marinakis et al., 2015). Different from UAVs (Lan et al., 2021), an AUV often faces a highly random ocean environment. Dealing with the randomness of the local path cost in the AUV route-planning problem is full of challenges, and swarm intelligence based evolutionary algorithms are powerful for solving this problem. Despite the excellent performance of existing evolutionary algorithms (Mahmoud Zadeh et al., 2018; Abbasi et al., 2020; Sun et al., 2022), the individuals in these methods are often of low efficiency due to the strict constraint of the total cost. The optimization is often seriously hindered because there is usually no clear boundary between the feasible and infeasible domains in the space of definition.

Fruitful achievements based on various theories exist in AUV path planning. Traditional algorithms, such as the Dijkstra algorithm (Kirsanov et al., 2013), A* algorithm (Duchoň et al., 2014), the rapidly exploring random tree algorithm (RRT) (Xue et al., 2019), and the fast marching (FM) algorithm (Yu and Wang, 2014), are all effective in handling the AUV path-planning problem. To improve the universality and efficiency, heuristic algorithms, such as the particle swarm optimization (PSO) algorithm (Zhuang et al., 2016), the ant colony algorithm (ACA) (Yan, 2021), and the differential evolution (DE) algorithm (Zhang JX et al., 2022), have been introduced and testified to be reliable. However, the uncertainty of the local path cost, which comes from the inconsistency between the plan and the actual operation, is often not considered. The replanning strategy is effective in dealing with the cost fluctuation in path and route planning (Zeng et al., 2015; Mahmoud Zadeh et al., 2019), but it is only a remedial measure. Ocean model based prediction has been introduced to enhance the performance of planning (Zeng et al., 2020), but it is rarely useful when the problem is small-scale in space and time. To settle the problem with stochastic path cost, a recourse model which describes the constraint as a soft one is designed, where a penalty is proportional to timeout (Teng et al., 2004). Regrettably, this method may bring risks to the AUV. Linearization has also been introduced to model the total profit based on the two-stage recourse model (Evers et al., 2014), but it is unsatisfactory when dealing with large instances because of the high computational cost. Inspired by these works, in this paper we propose a sampling strategy to settle the existing challenges.

To improve the reliability and efficiency of route planning when the task set is large and the local path cost is stochastic, an improved genetic algorithm (GA) based route planner equipped with a sampling-based route-cost estimator is designed in this paper. The GA is modified with the greedy strategy to be more suitable for solving the AUV route-planning problem compared with traditional methods. The DE local path planner is used to draw out the deterministic path cost, and the total deterministic route cost is replaced by the average sampled multiple times from the local paths' probability density functions (PDFs). The structure of the proposed route planner is illustrated in Fig. 1. The contributions of this study can be summarized as follows:

1. An exclusive GA with efficient evolutionary operators is devised to settle the AUV route-planning problem. To address the individual feasibility problem in heuristic methods neglected by reported works (Mahmoud Zadeh et al., 2015, 2018), a novel greedy strategy based rebirth operator is

proposed. It can effectively solve the problem that individuals in the infeasible domain contribute little under the total time constraint, thereby tremendously improving the efficiency of the optimization.

2. Most existing works regarding AUV route planning consider planning with deterministic local path cost (Mahmoud Zadeh et al., 2019). Taking the ocean complexity into consideration, we model the stochastic local path cost as the superposition of normal and Poisson distributions, based on the fact that the randomness of the path cost comes mainly from the path cost estimation error and the maneuvers caused by dynamic obstacles.

3. The route cost is obtained by sampling from the PDFs of the local paths. The sampling-based route-cost estimator is integrated into the route planner to evaluate the fitness of each feasible route by sampling local paths in the optimization.

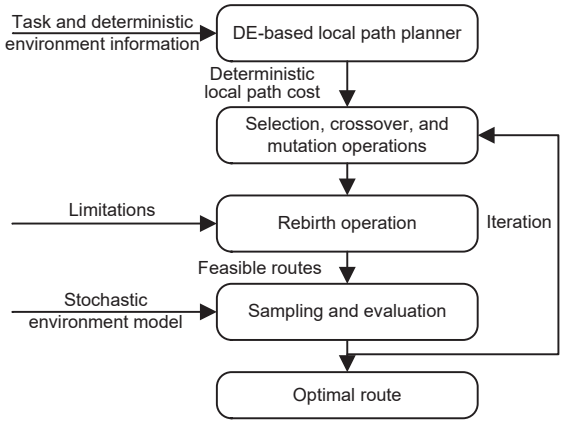

Fig. 1 Structure of the introduced route planner

2 Problem formulation

An AUV is expected to perform as many tasks as possible with limited energy. The tasks to be performed include water quality measurement, underwater photography, terrain detection, data loading and unloading with sensor nodes, etc. Each of the tasks can be reckoned as a task spot with a certain profit according to its importance. The mission of the planner is to find a feasible route that maximizes the total profit of one AUV execution under the circumstance that the battery capability is limited and the cost of each sub-path is stochastic. A route refers to the sequence of a series of tasks to be completed, while a path is the physical path to be followed when an AUV operates between two task spots. A typical scenario is shown in Fig. 2, where a feasible route is marked. In this section, the AUV route-planning problem with stochastic path cost is formulated mathematically.

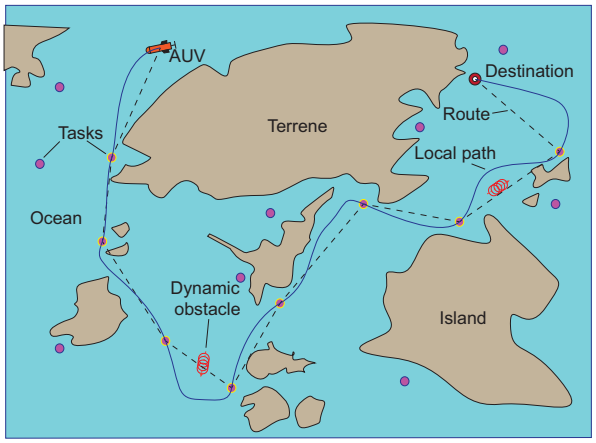

Fig. 2 A sample of an autonomous underwater vehicle (AUV) following the planned path in the mission region

2.1 AUV global route-planning problem

Assume that N is the task set. An AUV departs from N_{s} (launch position) and finally arrives at N_{d} (destination) where the AUV is supposed to be recovered by the ship. For convenience, we define the set of all the spots as follows:

$$N' = N \cup \{N_{\mathrm{s}}, N_{\mathrm{d}}\}. \tag{1}$$

The profit p, which represents the degree of priority of the task is endowed to each of the tasks in N. Define (i, j) as the arc connecting $N_i \in N'$ and $N_j \in N'$, and t_{ij} is the deterministic cost from task N_i to task N_j. Assume that all the arcs $(i, j) \in A$, where A is the set of arcs connecting spots in N', are accessible. Thus, we can formulate the AUV global route-planning problem on the complete graph $G = \{N', A\}$.

Let $s_{ij} \in \{1, 0\}$ be the binary selection variable. If arc (i, j) is chosen to be part of the route, $s_{ij} = 1$; otherwise, $s_{ij} = 0$. Thus, the total profit of one optional route is given by

$$J = \sum_{N_i \in N \cup \{N_{\mathrm{s}}\},\ N_j \in N} s_{ij} p_j, \tag{2}$$

where p_j is the profit of task N_j. The starting and destination spots must be included in the route.

Thus, it follows that

$$\sum_{N_j \in N} s_{\mathrm{s}j} = \sum_{N_i \in N} s_{i\mathrm{d}} = 1. \tag{3}$$

Each task can be carried out only once; i.e., for $N_k \in N$, there exists

$$\sum_{N_i \in N \cup \{N_\mathrm{s}\}} s_{ik} = \sum_{N_j \in N \cup \{N_\mathrm{d}\}} s_{kj} \leq 1. \tag{4}$$

The duration of travel of the AUV is usually limited by the battery capacity. For simplicity, we assume that the thrust of the power is constant, and thus the limitation of the energy is equivalent to that of the time. A route should satisfy

$$\sum_{(i,j) \in A} s_{ij} t_{ij} \leq T_{\max}, \tag{5}$$

where t_{ij} is the local cost of path $P_{i,j}$, while $T_{\max}$ is the time of battery duration. In this study, the goal of the optimization is to maximize Eq. (2) subject to conditions (3)–(5).

2.2 Local path with stochastic cost

The total cost of route $R = [N_\mathrm{s}, \ldots, N_i, N_j, \ldots, N_\mathrm{d}]$ is the sum of all the local path costs. The local cost t_{ij} is defined as the sum of the traveling time from N_i to N_j and the task execution time t_j as follows:

$$t_{ij} = \frac{d_{ij}}{v_\mathrm{G}} + t_j, \tag{6}$$

where v_G is the AUV's ground-referenced speed, d_{ij} is the length of the most time-saving path between N_i and N_j, and t_j is the time going to be spent in carrying out task N_j. The beeline path is the shortest way, but it is often not considered because the AUV needs to avoid collision with any obstacles and make full use of the ocean current. The most energy-saving path is given by the path planner embedded in the planning system.

Eq. (6) gives the deterministic form of the local path cost. However, the AUV's future motion is often uncertain, which results in uncertainty of the time cost. The uncertainty comes from the measurement error, trajectory tracking error, and unplanned maneuvers performed to avoid collision with unpredictable obstacles.

2.2.1 Measurement and trajectory tracking errors

The working area is covered by the ocean current field which is often complex and subject to changes. The AUV uses on-board sensors to measure its speed, location, and other information in real time, and then controls the actuator to track the planned trajectory. The measurement and trajectory tracking errors are hard to estimate or eliminate under limitations of the battery, sensors, and computing resources. Consequently, it is rational to assume that the impact of the measurement and tracking errors is statistically uniform; the additional time can be described as follows:

$$\Delta t_{ij}^\mathrm{e} \sim \mathcal{N}(0, \sigma^2), \tag{7}$$

where $\sigma \propto t_{ij}$ indicates that the longer the path $P_{i,j}$ is, the more the additional time Δt_{ij}^e may be.

2.2.2 Unpredictable obstacles

The on-board collision avoidance system protects the AUV from collisions. However, the detection range of the obstacle avoidance sonar is usually tens to hundreds of meters, and this is not always enough to cover the entire mission area. Besides, the positions of some moving obstacles are hard to ascertain in advance. These facts lead to the uncertainty of the motion: unplanned maneuvers have to be performed to avoid collisions. We assume that the AUV adopts the consistent maneuvering strategy during the movement, and each maneuver will introduce the same additional traveling time Δt^m. Thus, the additional traveling time of the travel from N_i to N_j caused by unplanned maneuvers is given by

$$\Delta t_{ij}^\mathrm{m} = n_{ij} \Delta t^\mathrm{m}, \tag{8}$$

where $n_{ij} \sim \pi(\lambda)$ is the number of maneuvers performed to avoid collisions, which follows a Poisson distribution, and $\lambda \propto (t_{ij} + \Delta t_{ij}^\mathrm{e})$ indicates that the longer the vehicle travels on its way, the more likely it will encounter obstacles.

Thus, the AUV's running time of each local path in one route is described as follows:

$$t = t_{ij} + \Delta t_{ij}^\mathrm{e} + \Delta t_{ij}^\mathrm{m}. \tag{9}$$

The cumulative distribution function (CDF) of the stochastic running time t can be written as follows:

$$\begin{aligned} F(t) &= \sum_{n=0}^{\infty} P\left(n_{ij} = n\right) P\left(\Delta t_{ij}^\mathrm{e} < t - n\Delta t^\mathrm{m}\right) \\ &= \sum_{n=0}^{\infty} \left[\frac{\mathrm{e}^{-\lambda} \lambda^n}{n!} F_\mathrm{e}(t - n\Delta t^\mathrm{m}) \right], \end{aligned} \tag{10}$$

where $F_{\mathrm{e}}(\cdot)$ is the CDF of $\Delta t_{ij}^{\mathrm{e}}$. It then follows that the PDF of the running time satisfies

$$\begin{aligned} f(t) &= \frac{\mathrm{d}F(t)}{\mathrm{d}t} \\ &= \sum_{n=0}^{\infty}\left[\frac{\mathrm{e}^{-\lambda}\lambda^n}{n!}f_{\mathrm{e}}(t-n\Delta t^{\mathrm{m}})\right] \\ &= \frac{\mathrm{e}^{-\lambda}}{\sqrt{2\pi}\sigma}\sum_{n=0}^{\infty}\left[\frac{\lambda^n}{n!}\exp\left(\frac{-(t-n\Delta t_{ij}-t_{ij})^2}{2\sigma^2}\right)\right], \end{aligned} \tag{11}$$

where $f_{\mathrm{e}}(\cdot)$ is the PDF of the normal distribution. Consequently, the mathematical expectation of the stochastic running time is

$$\begin{aligned} E(t) =& \int_{-\infty}^{+\infty} t\frac{\mathrm{e}^{-\lambda}}{\sqrt{2\pi}\sigma}\sum_{n=0}^{\infty}\left[\frac{\lambda^n}{n!}\mathrm{e}^{\frac{-(t-n\Delta t_{ij}-t_{ij})^2}{2\sigma^2}}\right]\mathrm{d}t \\ =& \sum_{n=0}^{\infty}\frac{\mathrm{e}^{-\lambda}\lambda^n}{n!}\int_{-\infty}^{+\infty}\left[\frac{t}{\sqrt{2\pi}\sigma}\mathrm{e}^{\frac{-(t-n\Delta t_{ij}-t_{ij})^2}{2\sigma^2}}\right]\mathrm{d}t \\ =& \sum_{n=0}^{\infty}\frac{\mathrm{e}^{-\lambda}\lambda^n}{n!}\left[\int_{-\infty}^{+\infty}\frac{t}{\sqrt{2\pi}\sigma}\mathrm{e}^{\frac{-(t-t_{ij})^2}{2\sigma^2}}\mathrm{d}t\right. \\ &\left.+\int_{-\infty}^{+\infty}\frac{n\Delta t_{ij}}{\sqrt{2\pi}\sigma}\mathrm{e}^{\frac{-(t-t_{ij}-n\Delta t_{ij})^2}{2\sigma^2}}\mathrm{d}t\right] \\ =& \sum_{n=0}^{\infty}\frac{\mathrm{e}^{-\lambda}\lambda^n}{n!}(t_{ij}+n\Delta t_{ij}) \\ =& \mathrm{e}^{-\lambda}\left[t_{ij}\sum_{n=0}^{\infty}\frac{\lambda^n}{n!}+\lambda\Delta t_{ij}\sum_{n=1}^{\infty}\frac{\lambda^{n-1}}{(n-1)!}\right] \\ =& t_{ij}+\lambda\Delta t_{ij}. \end{aligned} \tag{12}$$

The above inference holds based on the scenario defined by Eqs. (7)–(9). Note that the formulation of the stochastic path cost is decoupled from the planner. In other words, the model described can be replaced by others that coincide with specific environments better.

3 Greedy strategy based genetic algorithm (GGA): solver for the orienteering problem

The large-scale OP described by expressions (2)–(5) is nonlinear due to the stochastic path cost (Evers et al., 2014). The GA is a powerful heuristic tool to settle such a problem. It uses chromosomes to represent possible solutions and uses a variety of evolutionary operators to iteratively find the optimal solution of the objective function. Generally, these operators include selection, crossover, and mutation. Based on the improved operators of the GA, a rebirth operator is proposed and thus a new route planning method is generated, namely the greedy strategy based genetic algorithm (GGA).

3.1 Solution space and encoding

Every $N_i \in N$ is possible to be visited in any order during the AUV's motion. If we put aside the prohibition of task repetition, the solution space of the optimization problem will be an l-dimensional discrete space:

$$S = \{0, 1, \ldots, l\}^l, \tag{13}$$

where $l = |N|$ is the cardinal number of the task set. The population space of GA is hence defined as

$$\begin{aligned} S^K = \{\mathbb{X} | \mathbb{X} = \{X_1, X_2, \ldots, X_K\}, \\ X_i \in S\ (1 \le i \le K)\}, \end{aligned} \tag{14}$$

where K is the population size and X_i is the individual solution in population $\mathbb{X}$. The optimization goal is to maximize the total profit J and find the corresponding task sequence $X^* \in S$. Notice that because of the restrictions defined by conditions (3) and (4), not all the points in S are feasible to be chosen as possible solutions.

In the chromosome of one solution, each task is represented by its sequence number, which is an integer, and the sequence numbers of the tasks are arranged in order. Notice that the number of tasks to be carried out is unknown before the departure, so the optimization process should be carried out simultaneously in the l-dimensional space and all of its arbitrary-dimensional subspaces. Therefore, the chromosome length is fixed to l. If the number of planned tasks is fewer than l, the vacancies will be filled with zeros. In Fig. 3, an example of one feasible route in a certain task set and the corresponding chromosome are presented.

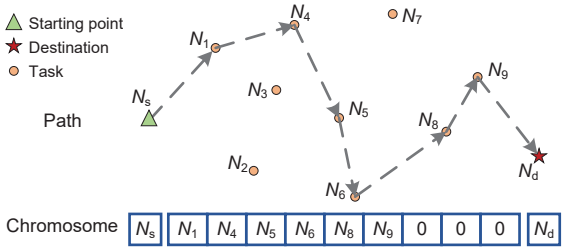

Fig. 3 An example route and its corresponding chromosome

3.2 Selection, crossover, and mutation

3.2.1 Selection operator

The selection operator $T_{\mathrm{s}} : S^K \to S$ is a random mapping, which selects one individual from the population.

The total profit is used as the fitness to assess each individual in the population, and the roulette wheel strategy is adopted by the selection operator. Specifically, the probability of each individual being selected to be a parent is related to its fitness and the population's fitness, which satisfies the following expression:

$$P(X_j^{\circ} = X_i) = \frac{J^{\alpha}(X_i)}{\sum_{q=1}^{K} J^{\alpha}(X_q)}, \tag{15}$$

where J is the fitness function defined in Eq. (2), X_j° is one individual in the parent population $\mathbb{X}^{\circ}$, and $0 < \alpha < \infty$ (typically, $\alpha = 1$). The whole $\mathbb{X}^{\circ}$ is generated by repeating the selection operation.

Additionally, we adopt the elite retention policy (Zhang H et al., 2020), which allows the most adaptable individuals in each generation being inherited directly in the next generation. In each generation, a certain proportion of the best-performing individuals will be reproduced to the next generation directly.

3.2.2 Crossover operator

Crossover is a fundamental aspect of evolution, which enables individuals in the population to exchange information by exchanging chromosome segments. The crossover operator $T_{\mathrm{c}} : S^2 \to S^2$ is also described as a random mapping.

On account of that a radical evolution strategy may lead individuals into the infeasible space frequently, the single-point crossover is therefore adopted. Assume that the parents to be operated are X_{p1} and X_{p2}. They are divided into two segments at the crossover point which is generated arbitrarily, as follows:

$$\begin{cases} X_{\mathrm{p1}} = [X_{\mathrm{p1},1}, X_{\mathrm{p1},2}], \\ X_{\mathrm{p2}} = [X_{\mathrm{p2},1}, X_{\mathrm{p2},2}]. \end{cases} \tag{16}$$

The offsprings $X_{\mathrm{o1}} = [X_{\mathrm{p1},1}, X_{\mathrm{p2},2}]$ and $X_{\mathrm{o2}} = [X_{\mathrm{p2},1}, X_{\mathrm{p1},2}]$ are generated by exchanging chromosome segments. The optimization has to be executed in subspaces with different dimensions; hence, the two crossover points in X_{p1} and X_{p2} are not required to be the same, which is different from the situation in traditional methods. Because task repetition may exist in X_{o1} and X_{o2}, task deletion is contained in the crossover operator to satisfy inequality (4). Thereafter, zeros will be used to fill the vacancies if any offspring's length is shorter than l. The crossover operator is intuitively illustrated in Fig. 4.

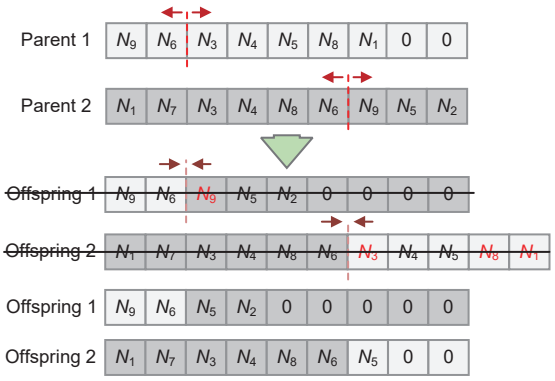

Fig. 4 An example of the crossover operation

3.2.3 Mutation operator

Mutation is a vital mechanism to ensure that the GA converges to the optimal solution set in probability. The mutation operator $T_{\mathrm{m}} : S \to S$ carried out on a single individual is a random mapping.

Traditional mutation operators can hardly cope with such a complex optimization problem. Therefore, we design four optional suboperators based on the existing methods. For X, the individual to be operated on, these suboperators are defined as follows:

1. Replacement: Replace task $N_i \in X$ by task $N_j \notin X$. Both N_i and N_j are selected with an equal probability from all the candidates. This operator is designed to perform a small-range optimization near the previous solution.

2. Insertion: Select a new task $N_j \notin X$ and generate the insertion position $i < l$ randomly; then, insert N_j between x_i and x_{i+1}, where $x_i, x_{i+1} \in X$ are two adjacent genes in X. Since gene x_i corresponds to a certain task, in what follows we use task x_i in X for simplicity. By this operation, a new task will be added into the route without changing the existing ones.

3. Swap: Switch the positions of tasks $x_i, x_j \in X$, whose indexes are selected randomly. This operation explores whether the fitness can be improved by changing the order of two tasks.

4. Inversion: The randomly selected task sequence $[x_i, x_{i+1}, \ldots, x_j] \in X$ is inverted to replace itself. This operator adjusts the order of tasks more efficiently.

These suboperators are illustrated in Fig. 5. One of the suboperators is chosen randomly each time the mutation is performed.

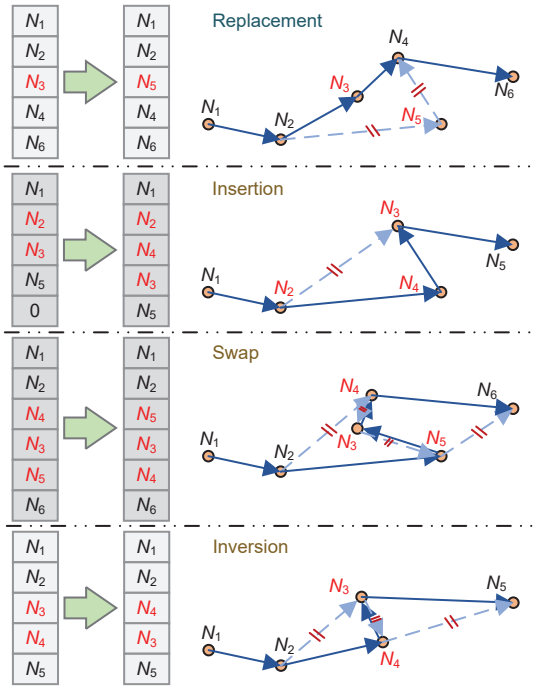

Fig. 5 Optional suboperators in the mutation operation (the dotted and solid lines represent the routes before and after being operated on, respectively)

3.3 Rebirth operator with greedy strategy

Due to the limitation of the AUV's durability, some newly generated individuals do not satisfy inequality (5). To settle this problem, discarding the infeasible individuals or giving up the final tasks (Evers et al., 2014) is a practical method. However, these processes are nearly equal to discarding part of the information acquired during the evolution. To improve the efficiency of the optimization using infeasible individuals, a novel rebirth operator is designed. The rebirth operation maps an individual from the infeasible solution space to the feasible solution space:

$$T_{\mathrm{r}} : \complement_S \Omega \to \Omega, \tag{17}$$

where $\Omega = \{X | X \in S, T_X \le T_{\max}\}$ is the feasible solution space and $\complement_S \Omega$ is the complement space of Ω.

This operator aims at minimizing the profit loss when some tasks have to be abandoned to satisfy the total cost restriction. Therefore, the problem can be formulated as follows:

$$\begin{aligned} &\min\ f(y_1, y_2, \ldots, y_l) = \sum_{i=1}^{l} p_i y_i \\ &\text{s.t.} \begin{cases} \sum_{i=1}^{l} \Delta T_i y_i \ge T_X - T_{\max}, \\ y_i \in \{0, 1\}\ (1 \le i \le l), \end{cases} \end{aligned} \tag{18}$$

where $X = [x_1, x_2, \ldots, x_n]$ is the individual to be optimized, y_i determines whether task x_i will be deleted, and ΔT_i is the running time that can be saved by deleting task x_i. This problem can be regarded as a variant KP, which is NP-complete. Problem (18) has to be solved several times in each iteration during the evolution, and hence a huge amount of computation is required to find the global optimal solution. The greedy algorithm is an effective tool to find the local optimal solution of KP. The proposed rebirth operator is inspired by the greedy strategy, which makes full use of the domain knowledge. The basic idea of the greedy strategy is to first get rid of the tasks with low profits and high costs.

Following the crossover and mutation operations, the rebirth operation will be performed if $T_X > T_{\max}$. The cost effectiveness of x_i is defined as ρ_i, satisfying the following expression:

$$\rho_i = p_i \Delta T_i^{-1},\ i = 1, 2, \ldots, l. \tag{19}$$

Following the greedy strategy, the task to be deleted is determined by

$$x_{\mathrm{d}} = \arg\min_{x_i \in X} \rho_i, \tag{20}$$

and the new individual X' is formed by the remaining tasks. To ensure that the route defined by X' can be completed within the battery life, the operation will be repeated until

$$\sum_{j=1}^{k} \Delta T_j \ge T_X - T_{\max}, \tag{21}$$

where k is the number of times that the deletion is executed.

Based on the greedy strategy, the newly designed rebirth operator can save those individuals who were destined to be weeded out. As a result, the average fitness of the population is improved,

giving the evolution a boost. Meanwhile, because the new information does not come from the existing feasible individuals, the diversity of the population is ensured, which allows the optimization to continue without premature convergence.

The algorithm of the whole GGA is given in Algorithm 1.

4 Global route planning with stochastic local path cost

The stochastic path cost leads to a situation as follows: the AUV may have to give up subsequent tasks because it has already spent too much time in completing the early tasks, and the remaining energy is not enough to finish the plan completely. If the stochastic feature of the path cost is not considered, the optimization result may be less robust, which causes a high probability of obtaining an unsatisfactory result in practice.

Algorithm 1 Greedy strategy based genetic algorithm (GGA) for route planning

Input the geographical map, ocean current map, and information of task set N'
Set the maximum number of iterations $i_{\max}$ and the proper population size K
Generate the initial population $\mathbb{X}_0$
for $i = 0$ to $i_{\max}$ **do**
  $\mathbb{X}_{i+1} = \varnothing$
  Calculate the fitness of each $X \in \mathbb{X}_i$
  for $j = 1$ to K **do**
    Select the parents X_{p1} and X_{p2} from $\mathbb{X}_i$ by roulette, and generate the crossover points randomly. $X' = T_{\mathrm{c}}(X) = [X_{\mathrm{p1},1}, X_{\mathrm{p2},2}]$ is the first intermediate individual
    if There exist repetitive nodes $\{N_{\mathrm{r}}\}$ **then**
      Delete repetitive tasks in X'
    end if
    Select one mutation suboperator to operate on X'. $X'' = T_{\mathrm{m}}(X')$ is noted as the second intermediate individual
    while $T_i > T_{\max}$ **do**
      Renew the list of ΔT
      for $k = 1$ to $|X|$ **do**
        $\rho_k = p_k \Delta T_k^{-1}$
      end for
      $N_{\mathrm{d}} = \underset{x_k \in X''}{\arg\min}\, \rho_k$, $X'' = X'' - N_{\mathrm{d}}$
    end while
    $X_{\mathrm{new}} = X''$
    Add X_{new} into $\mathbb{X}_{i+1}$
  end for
end for
$X_{\mathrm{optimal}} = \underset{X \in \mathbb{X}_{i_{\max}}}{\arg\max}\, J(X)$
return X_{optimal}

4.1 Local path planning

The optimal local path is the physical path $P_{i,j}$ from N_i to N_j. By following it, the AUV departing from N_i and targeting N_j can accomplish its journey using the minimum time. The AUV's water-referenced velocity v_{A} is constant if we assume that its thrust power is constant (Zeng et al., 2020). The AUV's ground-referenced velocity v_{G} is the resultant of v_{A} and the water velocity v_{C}. The energy consumption is consequently proportional to the time consumption. Therefore, the optimal path from N_i to N_j satisfies the following expression:

$$P_{i,j} = \arg\min_{P} \int v_{\mathrm{G}}^{-1} \mathrm{d}P, \tag{22}$$

under the constraints:

$$P \cap \mathbb{T} = \varnothing, \tag{23}$$

and

$$M_{\mathrm{AUV}} = 0, \tag{24}$$

where $\mathbb{T}$ is the forbidden region occupied by land and other obstacles, and M_{AUV} is the kinematic model of the AUV.

The map of the operation water area and the detectable ocean current and obstacles will be sent to the local path planner before the departure. The planner will output the optimal path, by following which the AUV can avoid any collision and finally arrive at N_j. Moreover, the ocean current can be used effectively and thus the energy is saved.

The DE-based path planner is used to act as the local path planner. DE is an improved version of GA, and it is usually used for multi-dimensional real-valued functions. A path in DE is defined by a series of way points (WPs), and the chromosome that represents this path is formed by the coordinates that are arranged in order. For example, $P = [N_1, \mathrm{WP}_1, \mathrm{WP}_2, N_2]$ forms a unique path from N_1 to N_2. However, such a path composed of a couple of straight lines is difficult for the AUV to follow, because sharp maneuvers are certainly tough and uneconomical. The B-spline method is an efficient tool for path smoothing (Cai and Yao, 2020); therefore, it is adopted to smooth the local path to satisfy the constraint of the AUV's kinematic model. This process is shown in Fig. 6.

The standard evolutionary operations (Mahmoud Zadeh et al., 2019) are included in the planner. Chromosomes are composed of genes, which

are the coordinates of the control points arranged in sequence. During the optimization, the evaluator assesses each path, and the result will be deemed as the gist of the evolution. In the global route-planning problem, the task set N contains l elements. The possibility of navigation exists between any two of the tasks in the absence of strong prior knowledge (which is common). Consequently, the $l \times l$ triangle path cost matrix $\boldsymbol{C}$ is formed and sent to the route planner to form the basis of the subsequent global optimization.

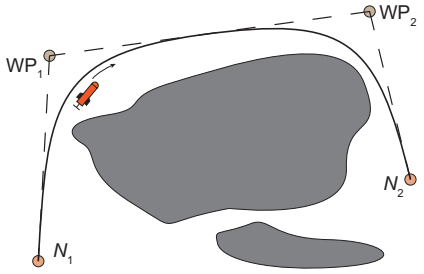

Fig. 6 A local path from N_1 to N_2

4.2 Sampling-based route evaluator

Taking the uncertainty of the path cost into consideration, the AUV may have to give up some tasks for the sake of its safety. Different from the planning, when the AUV finds that it has spent too much time on the initial tasks, which have already been completed, it has to discard the tasks at the end of the route. Therefore, the total profit in one test is represented as follows:

$$J_{\mathrm{r}} = \sum_{i=1}^{n} p_{x_i} - V = \sum_{i=1}^{n} p_{x_i} - \sum_{i=n-k+1}^{n} p_{x_i}, \quad (25)$$

where V is the profit shortage caused by the deletion of the k tasks at the end of the sequence.

Assume that $X = [x_1, x_2, \ldots, x_n]$ is one potential solution of the instance. The total profit loss brought about by the task deletion is naturally the superposition of the deleted tasks' profits. For simplicity, we define the Euclidean distance $\|\boldsymbol{x}_i - \boldsymbol{x}_{i+1}\|$ as the cost from x_i to x_{i+1} (for simplicity, we use the normal form to represent a point (e.g., x_i) while the bold form to represent the corresponding position for calculation (e.g., $\boldsymbol{x}_i$)). The time saved by deleting x_j $(2 < j < n)$ will be

$$\Delta T_j = -\|\boldsymbol{x}_{j+1} - \boldsymbol{x}_{j-1}\| + \sum_{i=j-1}^{j} \|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|. \quad (26)$$

If x_j and x_{j-1} are deleted simultaneously, the time saved will be

$$\Delta T_{\{j,j-1\}} = -\|\boldsymbol{x}_{j+1} - \boldsymbol{x}_{j-2}\| + \sum_{i=j-2}^{j} \|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|. \quad (27)$$

Because

$$\begin{aligned}\Delta T_j + \Delta T_{j-1} =& \|\boldsymbol{x}_j - \boldsymbol{x}_{j-1}\| - \|\boldsymbol{x}_{j+1} - \boldsymbol{x}_{j-1}\| \\ & - \|\boldsymbol{x}_j - \boldsymbol{x}_{j-2}\| + \sum_{i=j-2}^{j} \|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|,\end{aligned} \quad (28)$$

it follows that $\Delta T_{\{j,j-1\}} \neq \Delta T_j + \Delta T_{j-1}$, which indicates that the system does not satisfy the superposition principle (i.e., the problem is nonlinear).

Consequently, in spite of the mathematical formulation of the stochastic local path cost presented in Eqs. (9)–(12), the nonlinearity of the problem leads to the difficulty in describing the time consumption of one route analytically in a large instance, because a slight difference in one local path cost may dramatically affect the whole route. Therefore, the sampling-based method is adopted to estimate the cost of possible routes.

The expectation of the time cost of a local path is given by Eq. (12). However, the total time expectation of a route is not the superposition of that of its local paths. Estimation is implemented to replace the deterministic local path cost with the average value of the sample distribution (SD), which is obtained from m Monte Carlo simulations. Specifically, for route X, we have

$$\overline{T} = \frac{1}{m} \sum_{i=1}^{m} T_i, \quad (29)$$

where T_i is given by $T_i = \sum_{j=1}^{|X|+1} t_j$, and t_j is the time cost of the j^{th} local path of X, which is sampled from Eq. (11). Meanwhile, if $T_i > T_{\max}$, the tasks at the end of X will be discarded one by one until the time constraint is met. A larger m leads to a more exact approximation and costs more computing resources. An example of the PDF and SD of one local path where $m = 100$ is presented in Fig. 7. The expectation of the path cost calculated according to Eq. (12) is 830.98 s, while that obtained from the Monte Carlo simulations is 833.33 s. The approximation is satisfactory, and our subsequent tests will prove that the additional computational expense

caused by the sampling is acceptable. It is hence feasible to replace the PDF with SD in route planning. Algorithm 2 illustrates this process in pseudo-codes.

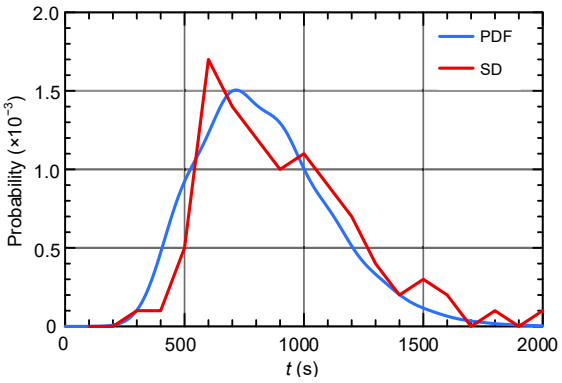

Fig. 7 Probability density function (PDF) and sample distribution (SD) of the local path $P_{i,j}$

Algorithm 2 Evaluation of route X

Input route X and path cost matrix $\boldsymbol{C}$
Input time limitation $T_{\max}$
Set the number of Monte Carlo simulations m
for $i = 1$ to m **do**
  for $j = 1$ to $|X| + 1$ **do**
    Look up $\boldsymbol{C}$ to obtain the deterministic path cost t_j^{d}
    Generate the stochastic path cost t_j^{s} based on SD
  end for
  $T_i = \sum_{j=1}^{|X|+1} t_j^{\mathrm{s}}$
  while $T_i > T_{\max}$ **do**
    $X = X - N_{\mathrm{end}}$
    Generate $t_{|X|+1}^{\mathrm{s}}$ based on SD
    $T_i = T_i - t_{\mathrm{end}}^{\mathrm{s}} - t_{\mathrm{end}-1}^{\mathrm{s}} + t_{|X|+1}^{\mathrm{s}}$
  end while
end for
$\overline{T} = \frac{1}{m} \sum_{i=1}^{m} T_i$
return $\overline{T}$

5 Simulation results and discussion

In this section, the GGA and the sampling-based route evaluator are tested to clarify their superiority. The GGA containing the newly designed operators is included in the planning system, where a DE-based path planner is used as the local path planner. The site of the tests is set in the sea around Luxi Island, Zhejiang Province, China, covering an area of 5 km × 7 km. The kinematic parameters of the AUV are set according to the AUV TH-B050R produced by Tianhe Maritime, Xi'an, China. A test instance of the route-planning problem with deterministic path cost is solved using the GGA planner proposed in Section 3. Furthermore, the sampling-based route evaluator described in Section 4 is tested when the stochastic path cost is taken into account. The simulations are implemented in MATLAB R2017b on a computer with an Intel i7-8700 CPU and 16 GB RAM.

5.1 Route planning with deterministic path cost

The AUV works off the coast of Luxi Island. Forty tasks whose importance is quantified as different dimensionless integers are scattered in this area, making up the task set. The AUV launched at the starting spot is urged to complete some of these tasks and arrive at the recovery station to be picked up. In this subsection, the local path cost is considered as deterministic, so the local paths and their costs are determined by the DE-based path planner.

The AUV travels at a constant ground-referenced speed of 4 kn (about 2.06 m/s), and the battery duration is set as 10 800 s. The GGA is tested and the result is compared with that from three GAs that use traditional operators. The differences between the GGA and the others are described as follows:

1. GA_1: Traditionally, there is only one form of mutation in the GA. For a chromosome to be operated on, the mutation operator selects one of the tasks and turns it into another. In other words, only the replacement is adopted in the mutation process. By contrast, the GGA adopts four kinds of mutation suboperators as designed in Section 3.2.

2. GA_2: Uniform crossover is deemed to be advanced in many pieces of research. We compare the GA where the uniform crossover is used with our algorithm, which applies the single-point crossover strategy.

3. GA_3: The greedy strategy based rebirth operator is newly proposed in this research. Following the traditional method, GA_3 deletes those individuals that are in the infeasible space and generates new ones to fill the gaps. We investigate the improvement due to the rebirth operator.

As for the items not mentioned, the test algorithm (GGA) is consistent with the others. In other words, there is only one difference between the test algorithm and each control algorithm. The parameters of these algorithms are listed in Table 1.

The planning result of the GAA in one test is presented in Fig. 8: the AUV departs from the starting station (marked by the triangle), passes through the task spots marked by circles (task profits are marked next to these circles), and finally reaches the destination (marked by the square). The route given by the planner is represented by a set of dashed lines, and the task spots to be visited are marked by crosses. Note that the dashed lines indicate only routes but not physical paths, which explains why they intersect the terrestrial area painted in black.

To draw a convincing conclusion, we perform 60 Monte Carlo simulations and list the results in Table 2, wherein the values of the residual time, total profit, and CPU time are the average of the Monte Carlo simulations. In Fig. 9, the variation of the total profit of the algorithms during the evolution is illustrated. Finally, the GGA earns the profit of 1051.2, which is 4.7% higher than that of the algorithm with monotonous mutation operator (GA_1) (Ferreira et al., 2014) and 19.9% higher than that of the algorithm with crossover strategy (GA_2) (Mahmoud Zadeh et al., 2018). The operations of insertion, inversion, and swap are essentially equivalent to repeatedly performing the replacement, which is the most classical mutation operation. However, they are more purposeful, which enhances the local optimization capability of the algorithm. The uniform crossover is efficient in many cases, but the sparsity of the feasible solution of the route-planning problem undermines its validity severely. In other words, too many individuals in the new generation will be infeasible if uniform crossover is adopted. Furthermore, the GGA outperforms GA_3, which does not adopt the rebirth operator, by 24.6% in terms of total profit. The stability of the algorithms is shown in Fig. 10. In addition to the more satisfactory median, the deviation of the GGA results is significantly lower than that of its competitors.

Table 1 Parameter setting of GAs

Parameter	Value
Population size	100
Number of iterations	1000
Elite proportion	0.05
Proportion of new individuals	0.05
Mutation probability	0.05
Crossover probability	0.8

5.2 Route planning with stochastic path cost

In this subsection, the performance of the sampling-based route evaluator is discussed when the

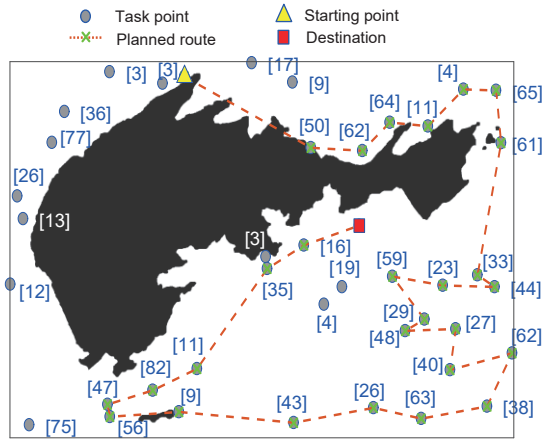

Fig. 8 Working region and the route planned by the GGA (expected profit: 1108)

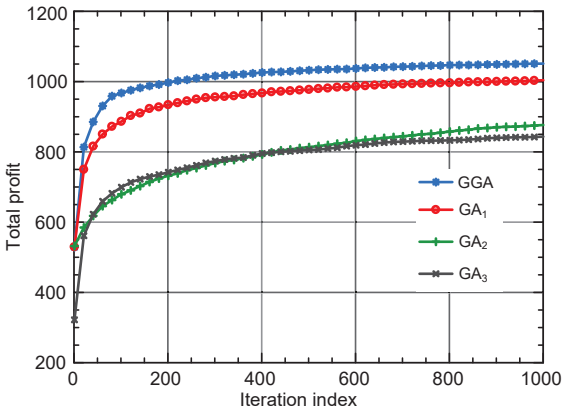

Fig. 9 Variation of the total profit in 1000 iterations (average of 60 Monte Carlo simulations)

Table 2 Results of 60 Monte Carlo simulations

Algorithm	Available time (s)	Residual time (s)	Total profit	Minimum profit	Maximum profit	CPU time (s)
GGA	10 800	29.3	1051.2	984	1128	12.94
GA_1	10 800	41.8	1004.0	809	1074	19.54
GA_2	10 800	39.8	876.5	691	1062	9.84
GA_3	10 800	71.8	843.4	692	996	7.60

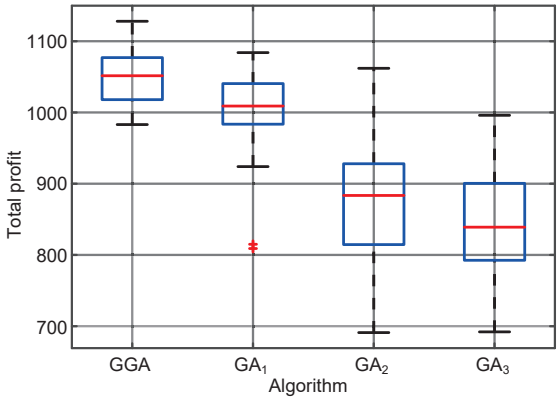

Fig. 10 Comparison of the stability of GGA, GA_1, GA_2, and GA_3 (the central mark, bottom and top edge marks on each box indicate the 50^{th}, 25^{th}, and 75^{th} percentiles, respectively, and the whiskers extend to the most extreme data points that do not consider outliers, while outliers are noted individually by "+")

local path cost is stochastic. The GGA route planner (GGARP) and the sampling-based GGARP (S-GGARP) are tested. The AUV carries out the same tasks as in the previous tests in the same ocean region, and its speed follows the previous setting. The battery duration is set as 7200 s, and the stochastic local path cost is set as described in Section 2.2. We set $\sigma = 0.2t_{ij}$ and $\lambda = 2 \times 10^{-3}t_{ij}$, where t_{ij} is calculated by the DE-based local path planner beforehand. Besides, we assume that each unplanned maneuver takes additional 60 s to avoid the collision, i.e., $\Delta t^{\mathrm{m}} = 60$ s.

One typical planning result is presented in Fig. 11. Apparently, most of the tasks in the paths given by the two planners are the same, while the differences lie in the order of these tasks and the selection of certain tasks in the initial and final moving stages. S-GGARP tends to arrange the tasks with low profits at the end of the route. By doing so, when the running time exceeds its plan so much that the AUV has to give up the last tasks, the profit shortage can be minimized. For example, S-GGARP gives up the task with profit 9 at the beginning of the voyage and includes the task with profit 19 at the end of the route. GGARP does not take the stochastic path cost into consideration; the expected returns given by GGARP and S-GGARP are close. It is obvious from Fig. 12 that GGARP maintains its advance during the evolution. It suggests that the expected profit of GGARP is steadily higher than that of S-GGARP.

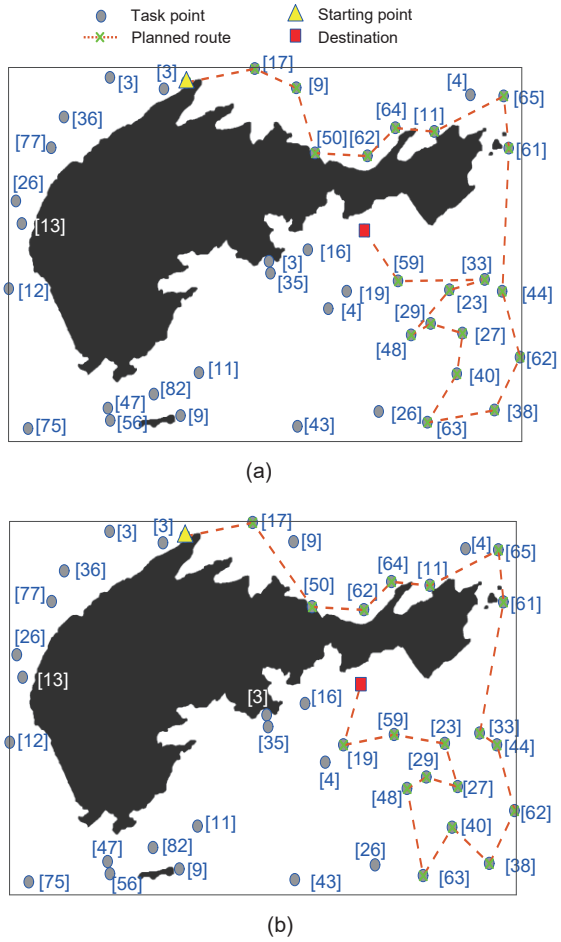

Fig. 11 Routes given by GGARP (a) and S-GGARP (b)

However, the actual profit that the AUV can achieve by following the GGARP route is often lower than expected. The reason is that S-GGARP is more discreet, which means that it inclines to ensure that the AUV can complete the planned tasks as much as possible when unplanned events occur. By contrast, GGARP tries to use every second available to improve the total profit and ignores the future uncertainty. If the situation deviates from the plan, the AUV may have to give up some tasks with high profits. For example, the deterministic running time values of the routes in Fig. 11 are 7140.5 s (GGARP) and 6721.5 s (S-GGARP). If an obstacle appears and it takes the AUV 60 s to avoid the collision, the value of total time consumption will be 7200.5 s (GGARP) and 6781.5 s (S-GGARP). The AUV following the GGARP route has to give up the task with the profit of 59 to avoid running out of energy on its way, while the AUV following the S-GGARP route has to give

up nothing. Finally, the obtained profit of the former (746) is lower than that of the latter (815). To testify the performance of the planners, each planning result of the 60 Monte Carlo simulations is tested in 100 different environments, where the ocean current and moving obstacles are generated randomly according to the preset parameters. The results show that the tested profit of the GGARP route is obviously lower than the expected one, while that of the S-GGARP route is almost consistent with the expected profit (Fig. 13).

Despite the lower expected profit, S-GGARP dramatically outperforms its counterpart in the tests. Table 3 lists the performances of the two algorithms after implementing 100 tests. It is difficult for the AUV to complete the GGARP route because the measurement and trajectory tracking errors and unpredictable obstacles are not taken into account. This leads to the abandonment of some high-profit tasks when the vehicle takes too much time in the early moving stage. Consequently, the average profit of the GGARP route in the tests is 6.4% lower than the expected one. Meanwhile, S-GGARP takes the localization and trajectory tracking uncertainty into consideration. Therefore, the total profits in the plan are coincident with those in the test, and S-GGARP is 5.5% more profitable than GGARP. Besides, the lower standard deviation reflects that S-GGARP is more robust and that its result is more predictable. S-GGARP spends more computing time than GGARP, because the sampling method consumes considerable computing resources. However, limited additional computing is acceptable, because the computing speed can be dramatically improved using the evolutionary algorithm's parallel version with system-on-a-programmable-chip (SoPC) (Tsai et al., 2011), and the route planning does not need to be completed online in most cases.

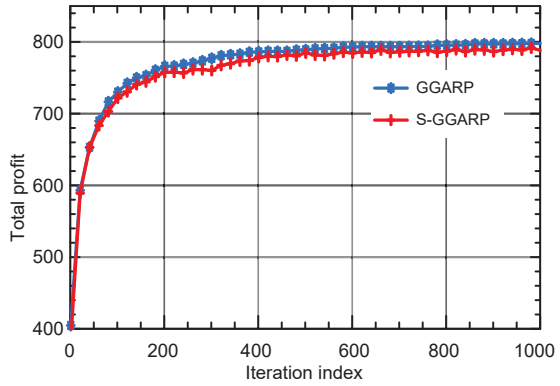

Fig. 12 Expected profits of GGARP and S-GGARP (average of 100 Monte Carlo simulations)

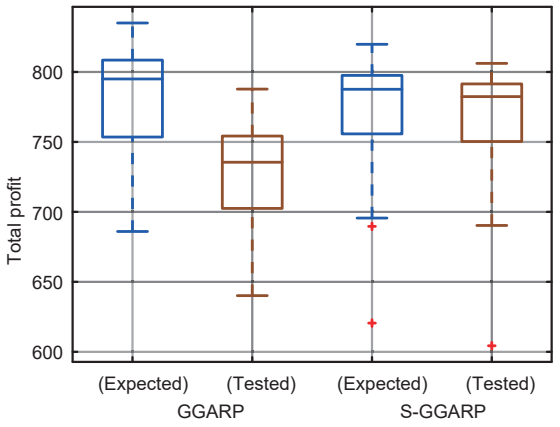

Fig. 13 Expected and tested profits of GGARP and S-GGARP in Monte Carlo simulations

Table 3 Performance of GGARP and S-GGARP when the path cost is stochastic

Route planner	Average profit (expected)	Average profit (tested)	Standard deviation	CPU time (s)
GGARP	798	747.15	64.92	10.72
S-GGARP	790	788.45	16.17	40.40

To further clarify the superiority of the proposed planner, planners using the three GAs with unimproved operators (GA_1RP, GA_2RP, and GA_3RP) are also tested in the scenario in this subsection. The results of 40 Monte Carlo tests are listed in Table 4. The results indicate that the proposed S-GGARP significantly improves the optimization results of the existing methods.

Table 4 Expected and tested profits of route planners when the path cost is stochastic

Route planner	Total profit (expected)	Total profit (tested)
GA_1RP	749	700.20
GA_2RP	674	606.55
GA_3RP	663	599.80
GGARP	789	746.85
S-GGARP	791.5	790.15

6 Conclusions

In this study, a GA-based AUV route planner has been proposed using the novel rebirth operator and the sampling-based route evaluator. The

developed planner leads the AUV to carry out a series of tasks selectively within the limited battery life and maximize the total profit when the measurement and trajectory tracking errors are not neglectable and dynamic obstacles are unpredictable. The deterministic local path cost between any two tasks has been calculated by the DE-based planner and then delivered to the global route planner. In the proposed GGA, traditional evolutionary operators have been improved to enhance the algorithm's capability for the AUV route-planning problem. Besides, the novel greedy strategy based rebirth operator has been integrated into the evolution process to improve the performance of the algorithm when dealing with large instances. Based on the deterministic local path cost, the stochastic local path cost has been considered as a random variable influenced by the moving uncertainty and undiscovered dynamic obstacles. The evaluation of each possible route has been executed by sampling from the PDFs of the local path costs, and the sampling-based route-cost estimator has been embedded in the route planner. The simulation results demonstrated that the planner performs effectively in the uncertain ocean environment, and the superiority of the proposed planner over traditional GA-based route planners has been testified.

Future work will focus on extending the proposed algorithms to multi-AUV collaborative operation, including task assignment, online coordination, and low-communication information sharing.

Contributors

Jiaxin ZHANG designed the research. Jiaxin ZHANG and Meiqin LIU processed the data. Jiaxin ZHANG drafted the paper. Senlin ZHANG helped organize the paper. Ronghao ZHENG revised and finalized the paper.

Compliance with ethics guidelines

Jiaxin ZHANG, Meiqin LIU, Senlin ZHANG, and Ronghao ZHENG declare that they have no conflict of interest.

References

Abbasi A, Mahmoud Zadeh S, Yazdani A, 2020. A cooperative dynamic task assignment framework for COTSBot AUVs. *IEEE Trans Autom Sci Eng*, early access. https://doi.org/10.1109/TASE.2020.3044155

Bagagiolo F, Festa A, Marzufero L, 2021. The orienteering problem: a hybrid control formulation. *IFAC-PapersOnLine*, 54(5):175-180. https://doi.org/10.1016/j.ifacol.2021.08.494

Cai CY, Yao XL, 2020. Trajectory optimization with constraints for alpine skiers based on multi-phase nonlinear optimal control. *Front Inform Technol Electron Eng*, 21(10):1521-1534. https://doi.org/10.1631/FITEE.1900586

Cheng CX, Sha QX, He B, et al., 2021. Path planning and obstacle avoidance for AUV: a review. *Ocean Eng*, 235:109355. https://doi.org/10.1016/J.OCEANENG.2021.109355

Chou XC, Gambardella LM, Montemanni R, 2021. A Tabu Search algorithm for the probabilistic orienteering problem. *Comput Oper Res*, 126:105107. https://doi.org/10.1016/j.cor.2020.105107

Dorling K, Heinrichs J, Messier GG, et al., 2017. Vehicle routing problems for drone delivery. *IEEE Trans Syst Man Cybern Syst*, 47(1):70-85. https://doi.org/10.1109/TSMC.2016.2582745

Duchoň F, Babinec A, Kajan M, et al., 2014. Path planning with modified A star algorithm for a mobile robot. *Procedia Eng*, 96:59-69. https://doi.org/10.1016/j.proeng.2014.12.098

Evers L, Glorie K, van der Ster S, et al., 2014. A two-stage approach to the orienteering problem with stochastic weights. *Comput Oper Res*, 43:248-260. https://doi.org/10.1016/j.cor.2013.09.011

Ferreira J, Quintas A, Oliveira JA, et al., 2014. Solving the team orienteering problem: developing a solution tool using a genetic algorithm approach. In: SnáŠel V, Krömer P, Köppen M, et al. (Eds.), Soft Computing in Industrial Applications. Springer, Cham, p.365-375. https://doi.org/10.1007/978-3-319-00930-8_32

Han GJ, Gong AN, Wang H, et al., 2021. Multi-AUV collaborative data collection algorithm based on Q-learning in underwater acoustic sensor networks. *IEEE Trans Veh Technol*, 70(9):9294-9305. https://doi.org/10.1109/TVT.2021.3097084

Ji HP, Zheng WM, Zhuang XY, et al., 2021. Explore for a day? Generating personalized itineraries that fit spatial heterogeneity of tourist attractions. *Inform Manage*, 58(8):103557. https://doi.org/10.1016/j.im.2021.103557

Kirsanov A, Anavatti SG, Ray T, 2013. Path planning for the autonomous underwater vehicle. Proc 4th Int Conf on Swarm, Evolutionary, and Memetic Computing, p.476-486. https://doi.org/10.1007/978-3-319-03756-1_43

Lan ML, Lai SP, Lee TH, et al., 2021. A survey of motion and task planning techniques for unmanned multicopter systems. *Unmanned Syst*, 9(2):165-198. https://doi.org/10.1142/S2301385021500151

Mahmoud Zadeh S, Powers D, Sammut K, et al., 2015. Optimal route planning with prioritized task scheduling for AUV missions. Proc IEEE Int Symp on Robotics and Intelligent Sensors, p.7-14. https://doi.org/10.1109/IRIS.2015.7451578

Mahmoud Zadeh S, Powers DMW, Sammut K, et al., 2018. A novel versatile architecture for autonomous underwater vehicle's motion planning and task assignment. *Soft Comput*, 22(5):1687-1710. https://doi.org/10.1007/s00500-016-2433-2

Mahmoud Zadeh S, Powers DMW, Atyabi A, 2019. UUV's hierarchical DE-based motion planning in a semi dynamic underwater wireless sensor network. *IEEE Trans*

Cybern, 49(8):2992-3005. https://doi.org/10.1109/TCYB.2018.2837134

Marinakis Y, Politis M, Marinaki M, et al., 2015. A memetic-GRASP algorithm for the solution of the orienteering problem. In: Le HA, Dinh TP, Nguyen NT (Eds.), Modelling, Computation and Optimization in Information Systems and Management Sciences. Springer, Cham, p.105-116. https://doi.org/10.1007/978-3-319-18167-7_10

Royset JO, Reber DN, 2009. Optimized routing of unmanned aerial systems for the interdiction of improvised explosive devices. *Mil Oper Res*, 14(4):5-19.

Schilde M, Doerner KF, Hartl RF, et al., 2009. Metaheuristics for the bi-objective orienteering problem. *Swarm Intell*, 3(3):179-201. https://doi.org/10.1007/s11721-009-0029-5

Sun SQ, Song BW, Wang P, et al., 2022. An adaptive bi-level task planning strategy for multi-USVs target visitation. *Appl Soft Comput*, 115:108086. https://doi.org/10.1016/j.asoc.2021.108086

Teng SY, Ong HL, Huang HC, 2004. An integer L-shaped algorithm for time-constrained traveling salesman problem with stochastic travel and service times. *Asia-Pac J Oper Res*, 21(2):241-257. https://doi.org/10.1142/S0217595904000229

Tsai CC, Huang HC, Chan CK, 2011. Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation. *IEEE Trans Ind Electron*, 58(10):4813-4821. https://doi.org/10.1109/TIE.2011.2109332

Vansteenwegen P, van Oudheusden D, 2007. The mobile tourist guide: an OR opportunity. *OR Insight*, 20(3):21-27. https://doi.org/10.1057/ori.2007.17

Xue ZB, Liu JX, Wu ZX, et al., 2019. Development and path planning of a novel unmanned surface vehicle system and its application to exploitation of Qarhan Salt Lake. *Sci China Inform Sci*, 62(8):84202. https://doi.org/10.1007/s11432-018-9723-5

Yan SK, 2021. Research on path planning of AUV based on improved ant colony algorithm. Proc 4th Int Conf on Artificial Intelligence and Big Data, p.121-124. https://doi.org/10.1109/ICAIBD51990.2021.9458959

Yu H, Wang YJ, 2014. Multi-objective AUV path planning in large complex battlefield environments. Proc 7th Int Symp on Computational Intelligence and Design, p.345-348. https://doi.org/10.1109/ISCID.2014.118

Zeng Z, Sammut K, Lammas A, et al., 2015. Efficient path re-planning for AUVs operating in spatiotemporal currents. *J Intell Robot Syst*, 79(1):135-153. https://doi.org/10.1007/s10846-014-0104-z

Zeng Z, Zhou HX, Lian L, 2020. Exploiting ocean energy for improved AUV persistent presence: path planning based on spatiotemporal current forecasts. *J Mar Sci Technol*, 25(1):26-47. https://doi.org/10.1007/s00773-019-00629-0

Zhang H, Xin B, Dou LH, et al., 2020. A review of cooperative path planning of an unmanned aerial vehicle group. *Front Inform Technol Electron Eng*, 21(12):1671-1694. https://doi.org/10.1631/FITEE.2000228

Zhang JX, Liu MQ, Zhang SL, et al., 2022. AUV path planning based on differential evolution with environment prediction. *J Intell Robot Syst*, 104(2):23. https://doi.org/10.1007/s10846-021-01533-9

Zhuang YF, Sharma S, Subudhi B, et al., 2016. Efficient collision-free path planning for autonomous underwater vehicles in dynamic environments with a hybrid optimization algorithm. *Ocean Eng*, 127:190-199. https://doi.org/10.1016/j.oceaneng.2016.09.040