symmetry

MDPI

*Article*

# Event Coverage Hole Repair Algorithm Based on Multi-AUVs in Multi-Constrained Three-Dimensional Underwater Wireless Sensor Networks

**Yaoming Zhuang [1,*], Chengdong Wu [1], Hao Wu [2], Zuyuan Zhang [3], Hongli Xu [1], Qingyong Jia [4] and Li Li [5]**

1 Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China; wuchengdong@ise.neu.edu.cn (C.W.); xuhongli@mail.neu.edu.cn (H.X.)
2 Engineering Faculty, University of Sydney, Sydney, NSW 2006, Austria; hawu1598@uni.sydney.edu.au
3 School of Compute Science, University of Oklahoma at Norman, Norman, OK 73070, USA; drzuyuanzhang@hotmail.com or Zuyuan.Zhang-1@ou.edu
4 Peng Cheng Laboratory, Shenzhen 518055, China; jiaqy@pcl.ac.cn
5 JangHo School of Architecture, Northeastern University, Shenyang 110819, China; lili1118@mail.neu.edu.cn
* Correspondence: zhuangyaoming524@163.com or zhuangyaoming@mail.neu.edu.cn

Received: 27 October 2020; Accepted: 14 November 2020; Published: 16 November 2020

check for updates

**Abstract:** It is important for underwater wireless sensor networks (UWSNs) to satisfy the diverse monitoring demands in harsh and perilous three-dimensional underwater environments. After the monitoring missions and demands transform, a large number of underwater event coverage holes will appear. Traditional network repair strategies cannot be applied to the ever-changing underwater monitoring missions and the harsh multi-constrained three-dimensional underwater environments. Multiple autonomous underwater vehicles (multi-AUVs) have strong adaptability and flexibility in perilous and harsh three-dimensional underwater environments. First, an underwater event coverage hole (UECH) repair model under various constraints is proposed. Next, a multi-agent event coverage hole repair algorithm (MECHR), which combines multi-agent strategy with diversity archive strategy, is proposed to repair UECHs in UWSNs. The presented algorithm symmetrically completes subtasks through information exchange and interactive operations with other agents. Unlike existing repair strategies, the MECHR algorithm can effectively repair a large number of UECHs resulted by the transformations in underwater monitoring scenes and demands. The MECHR algorithm can adapt to a wide range of harsh scenes and multi-constrained three-dimensional underwater environments. Eventually, the effect of the MECHR algorithm is verified through underwater repair simulation experiments, which can adapt to the constantly changing three-dimensional underwater monitoring environments.

**Keywords:** multi-AUVs; event coverage holes; multi-constrained situations; route planning; underwater wireless sensor networks

## 1. Introduction

Due to the consumption of land resources and the maturity of event monitoring technology, event monitoring research has progressively shifted from the land to the ocean [1,2]. Event monitoring in the underwater wireless sensor networks (UWSNs) has gradually become a novel research focus. With the continuous changes in underwater monitoring missions and demands, a large number of underwater event coverage holes (UECHs) will be generated in the UWSNs. Traditional research on coverage hole repair in sensor networks is mostly aimed at repairing coverage holes in a fixed two-dimensional unconstrained scene [3]. In the real world scene, it is essential to apply multiple types of sensors to detect events in the underwater environments with the optimal repair strategy according to various event monitoring demands, such as military intrusion monitoring, environmental monitoring, fish monitoring, and underwater rescue [4]. At the same time, due to the particularity of the underwater environments, the repair process based on multi-AUVs in the underwater environments will be limited by multiple types of harsh constraints. The traditional repair methods mostly use manpower redeployment repair and aircraft airdrop repair. Underwater sensor networks are mostly applied in perilous and harsh environments. Repairing the underwater event coverage holes by divers is costly, time-consuming, and highly perilous. The repair accuracy when being carried out by random airdrops is not satisfactory. Not only are the initial UECHs hard to repair, but also superfluous sensors waste a large number of network resources in UWSNs. With the maturity of underwater robot technology, multiple autonomous underwater vehicles (multi-AUVs) have been widely applied in UWSNs. Multi-AUVs can not only work in harsh and perilous underwater environments, but also quickly sail to accurate locations. Therefore, multi-AUVs are applied to accurately repair the UECHs in the underwater environments.

Traditional sensor network coverage holes are always resulted by sensor failure. The UECHs are usually resulted by the variations in monitoring missions and demands. For instance, in different periods of marine monitoring, different monitoring missions such as military intrusion event monitoring, environmental monitoring, fish monitoring, and underwater rescue will be performed. According to different monitoring events, UWSNs need to deploy various types and amounts of sensors in different locations in the underwater environments. After the underwater monitoring mission changes, lots of UECHs will occur in the UWSNs. Existing event deployment strategies are two-dimensional deterministic deployment, ignoring deployment height and depth. After deployment, they can only be used in a certain application scene. When the monitoring mission changes or sensors fail in UWSNs, the adjustment and repair are costly and time-consuming that cannot satisfy the real-time monitoring demands in the UWSNs.

Traditional coverage hole repair only applies to a single type of sensor [5]. In real-world applications, for the purpose of adapting to different monitoring environments and ever-changing monitoring demands, a single type of sensor can no longer effectively repair event coverage holes to satisfy multiple types of harsh monitoring requirements. Different from the two-dimensional scene repair, the underwater event coverage hole repair needs to consider the spatial location and spatial constraints. When faced with a large number of underwater event coverage holes, it is essential to effectively allocate heterogeneous sensor resources during the repair process of UECHs. By balancing the diversity and convergence of the solution set of the repair strategy, the optimal repair strategy is planned for the multi-AUVs. The repair strategy needs to plan the optimal repair strategy for multi-AUVs by balancing the diversity and convergence of the solution set.

Repairing event coverage holes in underwater perilous and harsh environments will be subject to multiple constraints: for example, repair time constraints, space constraints, repair cost constraints, repair energy constraints, and repair error constraints [6]. In particular, it is difficult to supplement energy in the underwater environments. It is essential to satisfy the demanding energy constraints under the premise of satisfying other underwater constraints. Even in the identical underwater monitoring scene, according to different monitoring missions, the UECH repair is restricted by diverse constraints. For example, in underwater environment monitoring, it is usually requested by diverse monitoring demands for underwater space in different times. In times of war, submarine intrusion monitoring will be conducted in the underwater environments. In peacetime, the underwater environment will be monitored for fish schools and environmental pollution. In the face of different monitoring missions and limited resources, the UECH repair is restricted by diverse constraints. Under the mission of submarine intrusion monitoring, it is essential to quickly and accurately repair the UECHs. As a result, the UECH repair is subject to strict time constraints and error constraints;

under the mission of fish monitoring, it is essential to control repair costs and reduce repair energy expenditure as much as possible. Thus, the repair of underwater event coverage holes will be subject to strict cost constraints and energy consumption constraints; in underwater environmental pollution monitoring, it is essential to quickly monitor and accurately determine the position of the oil spill with the lowest cost. Therefore, the repair of underwater event coverage holes will be restricted by rigorous time, error, and cost constraints.

Based on the above analysis, how to efficaciously allocate sensor resources in the harsh underwater environments with multiple constraints and apply multi-AUVs to efficiently repair event coverage holes in UWSNs is an urgent problem to be solved. The contributions can be summarized:

- It is the first time that the issue of repairing a large number of UECHs is carried out. In response to the rapidly changing underwater event monitoring demands in three-dimensional underwater practical applications, multiple types of sensors are applied to effectively repair a large number of UECHs in the three-dimensional UWSNs.
- It is the first time that the multi-constraint problem during the repair process of underwater event coverage hole is considered. Aiming at the space constraints and resource constraints of the underwater event coverage hole repair, an underwater event coverage hole repair model under multiple constraints is proposed.
- It is the first time to use multi-AUVs to repair a large number of underwater event coverage holes. Multi-AUVs are used to take various kinds of underwater sensors to repair UECHs in the underwater space with the optimal route. Then, a multi-agent event coverage hole repair algorithm (MECHR) is proposed that combines a multi-agent strategy with a diversity archive strategy. The MECHR algorithm efficaciously allocates underwater sensor resources with multi-AUVs to carry out the event coverage hole repair in harsh underwater environments to adapt to changing monitoring demands.

The structure of the paper is organized as follows. Related works are elaborated in Section 2. The UECH repair model is built under multiple constraints in Section 3. The multi-agent event coverage hole repair algorithm (MECHR) is elaborated in Section 4. In Section 5, underwater repair simulation experiments and performance analysis are implemented. In the end, the conclusion is presented in Section 6.

## 2. Related Works

Underwater wireless sensor networks (UWSNs) are widely used in event monitoring in underwater environments [7]. The event monitoring in UWSNs is based on concepts from event-based control and event-based signal processing [8]. Wang et al. propose a sensor virtualization algorithm to solve the event detection issue in the UWSNs [9]. Due to the constantly changing monitoring missions and demands in underwater event monitoring process, a large number of UECHs will be generated, which greatly affects the quality of underwater event monitoring. Different from conventional coverage holes that resulted in sensor invalidation, the UECHs usually resulted in the flexible surveillance missions. For example, in the ocean surveillance, there are different surveillance demands for the ocean at different periods. It is essential to apply different types of sensors to detect events in the underwater environments according to various event monitoring missions and demands, such as military intrusion monitoring, environmental monitoring, fish monitoring, and underwater rescue. When the monitoring missions and demands change, the types, numbers, and positions of sensors that need to be deployed are all different. Therefore, a large number of event coverage holes will be generated. At this time, multi-AUVs need to be used to repair UECHs to meet the needs of new monitoring missions and demands.

Jing et al. design a coverage holes recovery strategy to repair the coverage holes in UWSNs. The movement of sensors is divided into several processes, during each movement process; on the basis of the balance, distance, and position, relations move sensors to distinguish the aggregate sensors and achieve the maximum coverage of the monitoring region [10]. Jiang et al. put forward a distributed and energy-efficient event K-coverage algorithm (DEEKA) in UWSNs, which first considers the influence of terrible underwater environments on data collection and transmission. It takes into account the remaining energy of sensors and situations in which sensors are chosen by some other events [2]. Yu et al. propose a pigeon-based self-deployment algorithm (PSA) aiming at event coverage for UWSNs [11]. In the research on the coverage hole repair in sensor networks, only a single kind of sensor is used to repair coverage holes. With the constantly changing monitoring demands in UWSNs, a single kind of sensor can no longer satisfy the monitoring demands of the underwater environments. In [12], the author proposes an event coverage scheme to manage sensing scope dynamically. This scheme can only optimize the quality of event coverage temporarily, which is unable to repair all the event coverage holes completely.

Autonomous underwater vehicles are able to adapt to harsh and perilous underwater monitoring environments. In [13], the authors propose a system involving autonomous underwater vehicles (AUVs) and lots of static underwater sensor nodes (USN) together visually and audibly for data upload. In [14], the authors propose an AUV path-planning algorithm-based efficient routing aiming at UWSNs. In [15], Han et al. propose an AUV location prediction (ALP)-based information collection strategy to surmount high and imbalanced energy consumption aiming at UWSNs. Based on the above related works, in [16], the authors propose a high-efficiency information collection mechanism with multi-AUVs to optimize the performance of the underwater sensor networks and assure the high efficiency of the information collection service. The above research does not consider the repair problem in the case of a large number of UECHs.

The underwater event coverage hole repair in the harsh underwater environments will be restricted by multiple constraints. In [2], aiming at the complex environment of UWSNs, Jiang et al. put forward a distributed and energy-efficient event K-coverage algorithm (DEEKA) that first considers the influence of terrible underwater environments on event coverage and data collection. In real-world applications, underwater sensor networks will be constrained by multiple harsh conditions at the same time. Latif et al. address the energy hole creation issue in UWSNs and design a scheme to surmount the shortages in existing methods. The proposed scheme benefits from superfluous overlapping and repairs coverage holes in the process of network operation [17]. In response to the energy constraints in UWSNs, Azam et al. present a Balanced Load Distribution (BLOAD) strategy to shun energy holes generated by imbalanced energy consumption in the UWSNs [18]. In addition to energy constraints, underwater sensor networks will also be limited by the underwater three-dimensional space constraint. Wang et al. propose a novel node sinking strategy aiming at three-dimensional coverage in the UWSNs [19]. During the repair process of underwater event coverage holes, underwater sensor networks are not only restricted by one-fold energy or space constraint, but they are also restricted by multiple constraints simultaneously.

In underwater sensor networks, how to plan the route of multi-AUVs efficaciously on the basis of various monitoring missions and demands is a pressing problem that needs to be settled. Khan presents a greedy route planning strategy for maximizing data value in the UWSNs [20]. In [21], the authors propose an optimal strategy of coverage enhancing aiming at three-dimensional UWSNs on the basis of an improved fruit fly algorithm, which realizes the global optimal coverage according to foraging behavior. In [22], the authors take into account the energy consumption and cruising time of the AUVs. The authors propose an improved field mower scheme route algorithm to boost the long-range AUVs and the sensors that are balanced energy consumption. Based on the above related works, in [23], the authors present a Routing Void Prediction and Repairing (RVPR) strategy based on AUVs in the UASNs. This strategy applies AUVs to take sensors to repair the routing holes when predicting the emergence of holes. The above research does not consider the multiple constraints of the repair route. In [24], an inexact Multi-Objective Genetic Algorithm (iMOGA) is proposed to settle Constrained Multi-Objective Solid Travelling Salesman Problems (CMOSTSPs) in fragile, stochastic, and fuzzy scenes. In [25], the authors present a Symbiotic Organisms Search (SOS) with Simulated Annealing algorithm (SOS-SA) to solve multi-robot patrolling problems. In [26], the authors present a Joint Event Coverage Hole Repair algorithm (JECHR), which combines global repair with local repair to take advantage of multiple robots to repair event coverage holes in WSRNs.

The above related works analyze the research of event coverage hole repair, underwater sensor networks, multi-constrained underwater environments, and underwater robot route planning. Based on the above analysis, the underwater robot can effectively adapt to the harsh environment in the UWSNs. However, current research studies are aimed at route planning of underwater multi-robot systems. There is a big research gap in applying multiple types of sensors with multi-AUVs to repair UECHs in multi-constrained underwater scenes. In response to the above problems, we present an event coverage hole repair model and a multi-agent event coverage hole repair algorithm (MECHR) to repair UECHs in harsh and multi-constrained underwater environments.

## 3. Underwater Event Coverage Hole Repair Model

### *3.1. Main Idea*

Unlike the current coverage hole repair issue, underwater event monitoring is often put into use in perilous and harsh multi-constrained environments. In the process of repairing UECHs, it is generally restricted by multiple types of harsh constraints. Thus, the multi-constrained UECH repair issue is converted to the underwater multi-objective unconstrained repair issue. The converted multi-objective unconstrained repair issue is figured out by efficient multi-objective optimization algorithms that are able to accelerate the efficiency of problem-solving.

### *3.2. Problem Formulation*

The underwater event coverage hole (UECH) repair issue under multi-constraints is formulated as follows:

$$\begin{aligned} &\max F(\alpha) = f(\alpha_1, \alpha_2, \ldots, \alpha_n) \\ s.t.\ &g_i(\alpha) = g_i(\alpha_1, \alpha_2, \ldots, \alpha_n) \le 0 \quad (i = 1, 2, \ldots, p) \\ &h_i(\alpha) = h_i(\alpha_1, \alpha_2, \ldots, \alpha_n) = 0 \quad (i = p+1, \ldots, q) \end{aligned} \tag{1}$$

$F(\alpha)$ denotes the UECH repair objective function. $i$ represents the amount of constraint functions. $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ denotes various types of sensors, in which $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n) \in T \subset R^n$ represents an $n$-dimensional decision variable. $g_i(\alpha)$ and $h_i(\alpha)$ represent the inequality and equality constraint function, respectively.

**Definition 1.** *(Feasible Region): $\Omega = \{\alpha \middle| \alpha \in T, g_i(\alpha) \le 0, h_i(\alpha) = 0, i = 1, \ldots, q\}$ is the feasible region of Problem (1). The points in the feasible region are defined as feasible solutions.*

The feasible region of the underwater event coverage hole repair issue with equality constraints is smaller than the target space. Therefore, the equality constraints are usually converted into inequality constraints.

$$|h_i(\alpha)| - \mu \le 0 \quad (i = p+1, \ldots, q) \tag{2}$$

In Formula (2), $\mu$ represents the constraint conversion tolerance.

The multi-constraint UECH repair problem is converted into a multi-objective unconstrained repair issue.

$$F(\alpha) = (f(\alpha), V(\alpha)) \tag{3}$$

In Formula (3), $V(\alpha)$ represents the degree of constraint violation.

$$V(\alpha) = \sum_{i=1}^{i=q} V_i(\alpha) \tag{4}$$

where $V_i(\alpha)$ denotes the constraint violation value.

$$V_i(\alpha) = \begin{cases} \max(0, g_i(\alpha)), & i = 1, 2, \ldots, p \\ \max(0, |h_i(\alpha)| - \mu), & i = p+1, \ldots, q \end{cases} \tag{5}$$

**Definition 2.** *(Pareto Domination): If two solutions $\alpha, \beta \in R^n$, $\alpha$ dominates $\beta(\alpha \prec \beta) \Leftrightarrow \forall i \in \{1, 2, \ldots, n\}, \alpha_i \leq \beta_i$, and $\exists i \in \{1, 2, \ldots, n\}, \alpha_i < \beta_i$.*

**Definition 3.** *(Pareto Optimal Solution): $\alpha \in T$ is a Pareto optimal solution $\Leftrightarrow$ There is no other solution $\alpha' \in T$ can Pareto dominate $\alpha$.*

**Definition 4.** *(Pareto Set): The Pareto Set $\Psi$ consists of all Pareto optimal solutions.*

$$\Psi = \{\alpha \in T | \neg \exists \alpha' \in T : \alpha' \prec \alpha\} \tag{6}$$

**Definition 5.** *(Pareto Frontier): The mapping of the Pareto solution set in the target space is Pareto Frontier P.*

$$\mathrm{P} = \left\{ F(\alpha) \middle| \alpha \in \Psi \right\} \tag{7}$$

## 4. Multi-Agent Event Coverage Hole Repair Algorithm

### *4.1. Main Idea*

The current event coverage coverage hole repair algorithm uses non-inferior dominated sorting based on Pareto dominance as a preliminary selecting strategy. Since most of the individuals in the population are non-dominated solutions, non-inferior dominated sorting cannot effectively select individuals, which takes a long time. When the Pareto dominance relationship fails, the selection criteria based on diversity dominates the environmental selection stage. Most of the diversity maintenance mechanisms tend to select individuals that distribute in sparse areas. These individuals may not be able to enhance the convergence of the population. It even caused some degree of degradation in the evolutionary process. Therefore, balancing the convergence and diversity of the solution set is a key issue that needs to be solved for an underwater event coverage hole repair algorithm.

### *4.2. Multi-Agent Strategy*

The multi-agent strategy is a divide and conquer model. Each agent symmetrically and autonomously completes subtasks through information exchange and interactive operations with other agents. The multi-agent strategy has the characteristics of symmetry, autonomy, coordination, and distribution. Therefore, it is introduced into the underwater event coverage hole repair algorithm, which can effectively accelerate the repair efficiency and optimize the repair quality. In the underwater event coverage hole repair algorithm based on the multi-agent strategy, each agent has subpopulations of a certain number of individuals and evolves autonomously. By exchanging information with other agents, multiple agents co-evolve.

Aiming at the inefficiency of Pareto dominance and the difficulty in balancing the convergence and diversity of the solution set in the underwater event coverage hole repair problem, this paper proposes a MECHR algorithm. The multi-agent strategy is combined with the diversity archive strategy to solve the underwater event coverage hole repair problem with multi-AUVs.

### *4.3. Multi-Agent Event Coverage Hole Repair Algorithm*

The multi-agent event coverage hole repair algorithm (MECHR) first sets reference vectors that uniformly distribute in the target space. Then, the population will be divided into multiple subpopulations. Each agent performs the task of guiding the subpopulation to evolve toward the real Pareto frontier along the specified reference vector. The evolutionary behavior of agents are autonomous. Therefore, the scope of subpopulation genetic operations is limited to this agent. When the agent selects individuals, it can effectively distinguish the pros and cons of individuals based on the specified scalar index. Through the process of gathering individuals who perform different tasks every few generations into the diversity archive and updating the archive, the diversity of the population can be effectively maintained. The individuals in each agent are excellent individuals who perform each subtask. The updated diversity archive redistributes individuals to different agents. The migration of good genes between different tasks in this process can effectively improve the performance of the algorithm.

The MECHR algorithm is denoted in Algorithm 1. The MECHR algorithm first inputs parameters. Then, the MECHR algorithm sets the size of the population, the maximum evolutionary generation, the size of the subpopulation, and the frequency of archive updates. In steps 2 to 3 of Algorithm 1, the population, reference vector, and diversity archive are initialized. The fifth step is to allocate the individuals of the population to each agent, so that each agent has a certain number of subpopulations. The fifth step is to perform the task of guiding the subpopulation to evolve to the real Pareto front along the specified reference vector. Steps 6 to 7 describe the autonomous evolution process of agents. The scope of genetic operations of each agent is limited to this agent. In the process of selecting, scalar indicators are used to distinguish between the pros and cons of individuals, which does not depend on the Pareto relationship. Not only can they reduce the time complexity, but they also can effectively select individuals. Steps 9 to 11 describe that when the agent autonomously evolves to a designated generation $U$, all the individuals in the agent gather in the diversity archive and update the diversity archive to maintain the diversity of the population. The individuals in the updated diversity archive are redistributed to each agent. The above process can realize the migration of good individuals between different tasks and effectively strengthen the convergence of the algorithm.

**Algorithm 1** The Multi-Agent Event Coverage Hole Repair Algorithm (MECHR)

**Input**: population maximum evolutionary generation $s_{\max}$, population size $M$, subpopulation size $W$, archive update frequency $U$.
**Output**: population final solution set $P_{s_{\max}}$.
1: **Begin**
2: Initialization: initial population $P_0$ and diversity archive $D$, current iteration number $s = 0$;
3: Generate reference vector set $R$;
4: **while** $s \le s_{\max}$ do
5: Divide individuals in population $P_s$ into $R$ agents $\{K_{s,1}, K_{s,2}, \ldots, K_{s,R}\}$;
6: **for** $i = 1 : R$ // Each agent evolves autonomously.
7: $K_{s,i}$ generates offspring $W_{s,i}$, and selects individuals to update $K_{s,i}$;
8: **end for**
9: **if** $s$ mod $U$ is 0
10: $P_s$ is used to update the diversity archive $D$, and the updated archive is $D'$;
11: $P_{s+1} = D'$;
12: **end if**
13: $s = s + 1$;
14: **end while**
15: **end**

*4.4. Population Initialization*

First, the MECHR algorithm initializes the population. The initial population $P_0$ is composed of $M$ individuals that are randomly generated in the target space.

$$\alpha_m = rand(0,1) * (H^m_{\max} - H^m_{\min}) + H^m_{\min} \tag{8}$$

where $H^m_{\min}$ is the lower limit of $\alpha_m$, $H^m_{\max}$ is the upper limit of $\alpha_m$, and $rand(0,1)$ is a random value uniformly distributed in the interval [0,1].

*4.5. Reference Vector Generation*

So as to maintain the diversity of the population in the evolution process, reference vectors that are uniformly distributed in the decision space need to be preset to guide the evolution of the agent. The process of generating reference vectors is as follows:

The systematic sampling method can generate uniformly distributed reference points in the hyperplane of the target space [27]. For an optimization problem with $N$ objectives, if the [0,1] interval on each objective problem is equally divided into $\lambda + 1$ parts, the number of generated reference points is $R^T = \begin{pmatrix} N+\lambda-1 & \lambda \end{pmatrix}^T$.

The coordinates of each reference point are:

$$(r_i, r_i^j) = \begin{cases} r_i = (r_i^1, r_i^2, \ldots, r_i^N) \\ r_i^j \in \left\{0, \frac{1}{\lambda}, \frac{2}{\lambda}, \ldots, 1\right\}, \sum\limits_{j=1}^{N} r_i^j = 1 \end{cases} . \tag{9}$$

The ideal point of the population is $c^* = c_1^{\max}, c_2^{\max}, \ldots, c_N^{\max}$, where $c_i^{\max}$ is the maximum value on the $i$-th objective function. The nadir point of the population is $c' = c_1^{\min}, c_2^{\min}, \ldots, c_N^{\min}$, where $c_i^{\min}$ is the minimum value on the $i$-th objective function. The lines connecting the ideal point and each reference point constitute the reference vector set $R$.

*4.6. Multi-Agent Evolution*

The algorithm divides the population into multiple subpopulations. Then, each agent performs the task of guiding the subpopulation to evolve toward the real Pareto frontier along the specified reference vector. The evolutionary behavior of agents is autonomous. Therefore, the scope of the subpopulation genetic operations is limited to this agent.

(a) Agent Partition

The standardized population $G$ is divided into $R$ agents $K_1, K_2, \ldots, K_R$. Each agent has its own subpopulation and evolves in the direction of the specified reference vector. So as to maintain the diversity of the population, each subpopulation needs to maintain a certain number of individuals. The standard of population division is to assign $W$ individuals with the smallest angle between the target vector and the reference vector to the corresponding agents. The formula for calculating the cosine of the angle between the individual target vector and the reference vector is as follows:

$$\cos \varphi_j = \frac{F_i^* \cdot r_j}{||F_i^*||} \tag{10}$$

where $\varphi_j$ is the angle between the target vector $F_i^*$ and the reference vector $r_j$.

(b) Genetic Operation

When solving the multi-objective optimization problem, due to the huge target space, the individuals in the population are far apart. The offspring individuals generated by the crossover may also be far away from the parent individuals, and the reorganization operator fails. Since the evolutionary behavior of each agent is autonomous, the scope of selecting parent individuals is limited to this agent. Individuals are close together, which can guarantee the effectiveness of the recombination operator. The parent individuals are randomly selected from this agent to enter the mating pool. Then, the simulated binary crossover operator and polynomial mutation operator are applied to generate offspring.

(c) Agent Update Algorithm

Before updating the agent, it is necessary to standardize the individuals in the agent. First, the offspring individuals are used to update $c^*$ and $c'$. Then, the individuals are standardized in the agent as follows:

$$F'_j(\alpha) = \frac{1}{c'_j - c^*_j}(F_j(\alpha) - c^*) \tag{11}$$

where $j \in \{1, 2, \ldots, N\}$, $F_j(\alpha)$ is the function value of the individual before normalization. $F'_j(\alpha)$ is the function value of the individual after normalization in the $j$-th target problem.

The penalty boundary intersection (PBI) method is able to deal with continuous multi-objective optimization issues. The PBI method uses two distances $l_1(\alpha)$ and $l_2(\alpha)$ to measure the pros and cons of individuals.

$$\max g(\alpha) = l_1(\alpha) + \xi l_2(\alpha) \tag{12}$$

where $\alpha \in T$, $\xi$ is the preset parameter. $l_1(\alpha)$ denotes the projection distance of the individual $\alpha$ in the direction of the reference vector $r$ to the origin that represents the convergence information of the individual. $l_2(\alpha)$ denotes the distance from the individual $\alpha$ to the reference vector $r$ corresponding to the agent that represents the diversity information of the individual. The calculation formulas for the two distances $l_1(\alpha)$ and $l_2(\alpha)$ are as follows:

$$l_1(\alpha) = \frac{\left\| r(F'(\alpha) - c^*)^T \right\|}{\|r\|} \tag{13}$$

$$l_2(\alpha) = \left\| F(\alpha) - (l_1 \frac{r}{\|r\|} + c^*) \right\| \tag{14}$$

According to Formula (12), the individual penalty boundary intersection (PBI) value is calculated as the individual fitness. As the individual PBI value is smaller, its convergence and diversity are better. Therefore, the agents are sorted in the ascending order of the PBI value, and the top-ranked individuals are selected into the next generation population. The basis for each agent to evaluate the individual's pros and cons is a quantified index, which does not depend on the Pareto dominance relationship. Therefore, the algorithm can effectively select individuals. Algorithm 2 describes the agent update process.

**Algorithm 2** Agent Update Algorithm

**Input**: agent $K_{s,i}$, the offspring $W_{s,i}$ produced by the agent, the reference vector $r_i$ corresponding to the agent.
**Output**: the updated agent $K_{s+1,i}$.
1: **Begin**
2: $H = K_{s,i} \cup W_{s,i}$;
3: **for** $i = 1 : H$ do
4: calculate the PBI value of the individual in $H$;
5: **end for**
6: Individuals in $K_{s,i}$ are sorted in ascending order of PBI value;
7: $K_{s+1,i} = \left\{H_1, H_2, \ldots, H_{K_{s,i}}\right\}$;
8: **end**

### 4.7. Diversity Archive Update Algorithm

After each agent autonomously evolves for a specified number of generations, it is necessary to update the diversity archive with the individuals in all agents. The diversity archive update is mainly carried out in accordance with the principle of individual priority with the largest angle [28]. The algorithm first merges the individuals in all agents and the individuals in the diversity archive. Then, non-inferior dominance sorting is used to perform stratification $\{F_1, F_2, \ldots, F_{\max}\}$. Individuals of the first $(q-1)$ level are selected into the updated diversity archive $D'$, where $|D'| + |F_{q-1}| < |M|$ and $|D'| + |F_q| \geq |M|$. When selecting individuals in the $q$-th layer $F_q$, the minimum angle between each individual and the individual target vector in $D'$ is first calculated in order to select the largest individual from the individuals with the smallest included angle value into $D'$, and this individual is deleted from $F_q$. Then, the minimum included angle between the individual in $F_q$ and the individual target vector in $D'$ is calculated to select a pair of individuals $\alpha^* \in F_q$ and $\alpha_{\min} \in D'$ with the smallest minimum included angle. If the angle between the pair of individual target vectors is less than the threshold $0.5\pi/(M+1)$, and the convergence of $\alpha^*$ is better than $\alpha_{\min}$, then $\alpha^*$ enters $D'$ instead of $\alpha_{\min}$, and this individual is deleted from $F_q$. The specific process of diversity archive update is shown in Algorithm 3.

By using individuals in different agents to update the diversity archive, the diversity of the population can be effectively maintained. Gupta proposes that the migration of good genes between different tasks during the evolution of the population can effectively accelerate the convergence of the algorithm [29]. The updated diversity archive stores the good individuals of all tasks. These individuals can be redistributed to different agents according to Formula (11) to realize the migration of good genes between different tasks.

---

**Algorithm 3** Diversity Archive Update Algorithm (DAU)

---

**Input**: population $P_s$, diversity archive $D$.
**Output**: updated diversity archive $D'$.
**Begin**
  $D = D \cup P_s$;
  Divide $D$ into different non-dominated layers $\{F_1, F_2, \ldots, F_{\max}\}$ by non-inferior dominance sorting;
  $H = \varnothing, q = 1$;
  **while** $|Q| + |H| < M$ do
    $H = H \cup F_q, q = q + 1$;
  **end while**
  $D' = H/Q$; // $D'$ is the remaining individuals of subpopulation $W$ after removing the last layer of non-inferior dominance sorting.
  $flag(\alpha_j) = flase, \alpha_j \in F_q, j = 1, 2, \ldots, U$;
  Calculate the minimum angle $\psi(\alpha_j)$ between the target vector of all individuals in $\alpha_j \in H \wedge flag(\alpha_j) == flase$ and $D'$;
  **while** $|D'| < M$ do
    $\alpha_{\max} = \operatorname{argmax}_{j=1}^{U}\{\psi(\alpha_j) | \alpha_j \in H \wedge flag(\alpha_j) == flase\}$; //Find the individual with the largest minimum angle value.
    $D' = D' \cup \alpha_{\max}, flag(\alpha_{\max}) = true$;
    Calculate the minimum angle $\psi(\alpha_j)$ between the target vector of all individuals in $\alpha_j \in H \wedge flag(\alpha_j) == flase$ and $D'$;
    $\alpha_{\min} = \operatorname{argmin}_{j=1}^{U}\{\psi(\alpha_j) | \alpha_j \in H \wedge flag(\alpha_j) == flase\}$; //Find the individual with the smallest minimum included angle value.
    $\alpha^* = \operatorname{argmin}_{j=1}^{U}\{\psi(\alpha_{\min}, \alpha_j) | \alpha_j \in H \wedge flag(\alpha_j) == flase\}$; //Find the individual with the smallest angle between $H$ and $\alpha_{\min}$.
    **if** $\psi(\alpha^*, \alpha_{\min}) < 0.5\pi/(M+1)$ and $\sum_{i=1}^{N} f_i(\alpha^*) < \sum_{i=1}^{N} f_i(\alpha_{\min})$
      $D' = D'/\alpha_{\min}, D' = D' \cup \alpha^*, flag(\alpha_i) = true$;
    **end if**
  **end while**
**end**

---

### *4.8. The Complexity Analysis of the MECHR Algorithm*

For the event coverage hole repair problem with $M$ targets, the algorithm is iterated once, and then $M$ individuals need to be selected from the $2M$ population into the next generation population. The time complexity for the algorithm to assign individuals to different agents is $O(NM^2)$. In the process of agent autonomous evolution, the time complexity for each agent to select individuals is $O(NW \log W)$, where $W$ is the preset subpopulation size, which is a small integer. There are $R$ agents in total. The time complexity of agent autonomous evolution in one generation is $O(N|R|W \log W)$. The size of $|R|$ is close to the population size $M$. Therefore, the time complexity of the agent autonomous evolution process is close to $O(NM)$. Every designated $S$ generations, it is necessary to update the diversity archive with the individuals in each agent. The time complexity of the diversity archive update is $\max\left\{O(NM^2), O(M \log^{N-2} M)\right\}$ [28]. In summary, the time complexity of the proposed algorithm is $\max\left\{O(NM^2), O(M \log^{N-2} M)\right\}$.

## 5. Performance Evaluation

Repairing UECHs in the harsh underwater environments is subject to multiple constraints. In this section, the effect of the MECHR algorithm is tested. Multi-AUVs are applied to repair the UECHs in the three-dimensional multi-constrained underwater sensor networks. The six AUVs respectively carry six different kinds of sensors setting out at the base position to repair the UECHs on the optimal route. According to various monitoring missions in four underwater situations, the amount of sensors loaded by each AUV is different respectively.

### *5.1. Experimental Settings*

In the underwater repair simulation experiments, six different types of sensors are used, including water pressure sensors, eddy current sensors, flow velocity sensors, water quality combination sensors, laser displacement sensors, and water temperature sensors. These sensors have different repair time, radius, cost, energy, and error, respectively, which depend on the sensor properties. Each type of sensor is carried by an AUV, respectively. The parameters of six types of sensors are shown in Table 1.

**Table 1.** The sensor parameters.

| Symbol | Sensors | T/S | R/M | C | En/J | Er/CM |
|---|---|---|---|---|---|---|
| $\alpha_1$ | Water Pressure Sensor | 3 | 35 | 16 | 5 | 15 |
| $\alpha_2$ | Eddy Current Sensor | 5 | 10 | 12 | 15 | 25 |
| $\alpha_3$ | Flow Velocity Sensor | 4 | 30 | 24 | 25 | 10 |
| $\alpha_4$ | Water Quality Combination Sensor | 2 | 25 | 8 | 10 | 20 |
| $\alpha_5$ | Laser Displacement Sensor | 6 | 15 | 4 | 20 | 30 |
| $\alpha_6$ | Water Temperature Sensor | 1 | 20 | 20 | 30 | 5 |

The abbreviations are defined: T: repair time; R: repair radius; C: repair cost; En: repair energy; Er: repair error.

### *5.2. Performance Evaluation*

During the repair process of the UECHs, the repair algorithms have to meet multiple constraints including repair time, space, cost, energy, and error constraints. At the same time, the multi-AUVs have to repair UECHs on the most efficient route. In the performance comparison experiment, the MECHR algorithm makes comparisons with the iMOGA [24], SOS-SA [25], and JECHR [26] algorithms.

In the underwater repair simulation experiments, four underwater situations with multiple constraints are tested to show the performance of the MECHR algorithm. Each underwater situation will be restricted by different repair time, space, cost, energy, and error constraints. The parameters of the experiments are shown in Table 2. Different experimental situations are set to simulate various constraints in real-world applications. Simulation experiments are performed to verify whether the MECHR algorithm is able to efficaciously allocate limited sensor resources in the multi-constrained

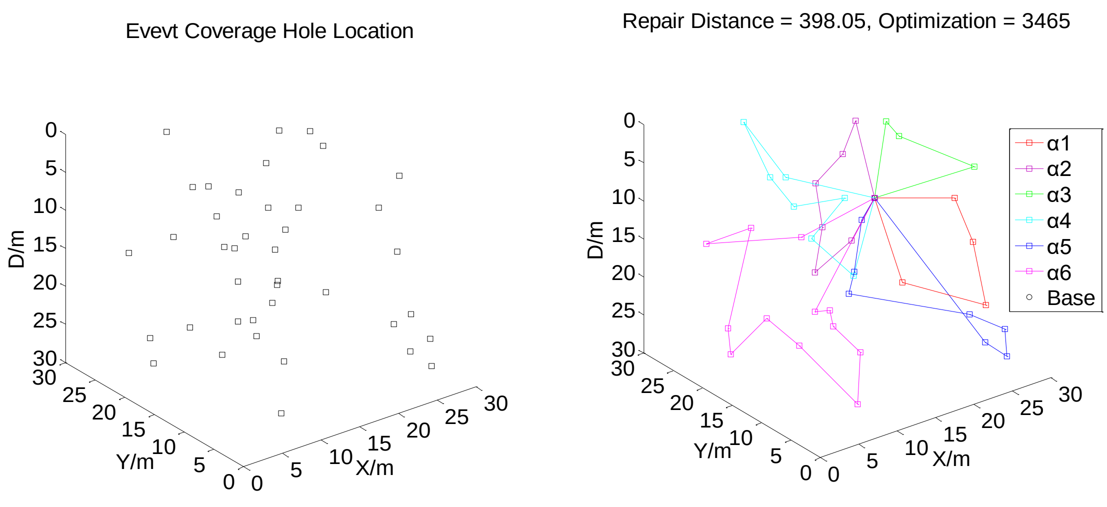
UWSNs. In real-world applications, the repair process of UECHs faces unknown multi-constrained underwater environments. The experiment situations are a portion of multi-constrained underwater environments in real-world applications.

**Table 2.** The experimental parameters.

| Situation | T/S | X | Y | Dep | C | En/J | Er/CM | Dis/M |
|---|---|---|---|---|---|---|---|---|
| 1 | 96 | 950 | 1295 | 1025 | 328 | 510 | 490 | 398.05 |
| 2 | 146 | 860 | 905 | 895 | 588 | 585 | 450 | 406.03 |
| 3 | 163 | 890 | 945 | 990 | 624 | 455 | 575 | 445.21 |
| 4 | 119 | 1145 | 1390 | 1195 | 564 | 595 | 545 | 482.56 |

The abbreviations are defined: T: repair time constraints; X: horizontal repair constraints; Y: vertical repair constraints; Dep: depth repair constraints; C: repair cost constraints; En: repair energy constraints; Er: repair error constraints; Dis: repair route distance.

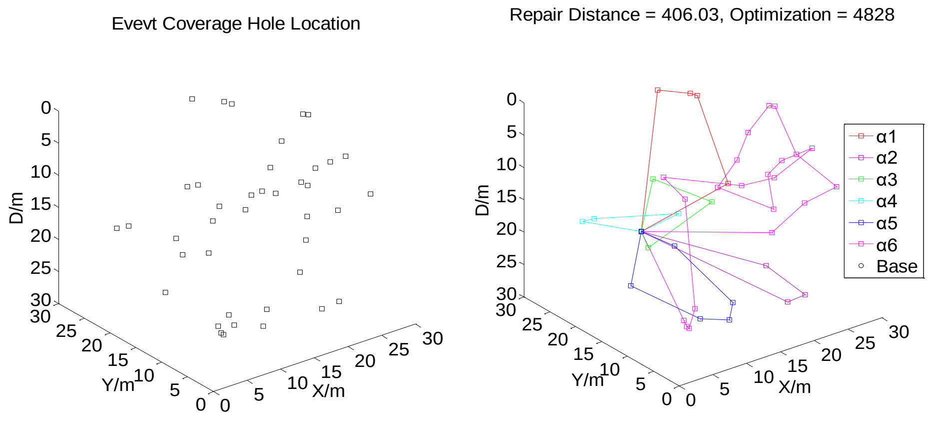
Figures 1–4 present the distribution of underwater event coverage holes and multi-AUV repair strategies under four multi-constraint situations. Figures 1a, 2a, 3a and 4a present the allocation of the UECHs in four different situations. The squares in the figure indicate the position of the UECHs. Figures 1b, 2b, 3b and 4b present the multi-AUV repair strategy aiming at four different underwater situations. Six AUVs set off from the base position at the same time. Each AUV carries a type of sensor to repair a total of forty UECHs in the UWSNs. The multi-color routes in Figures 1b, 2b, 3b and 4b present the repair route of six AUVs.




(a) (b)

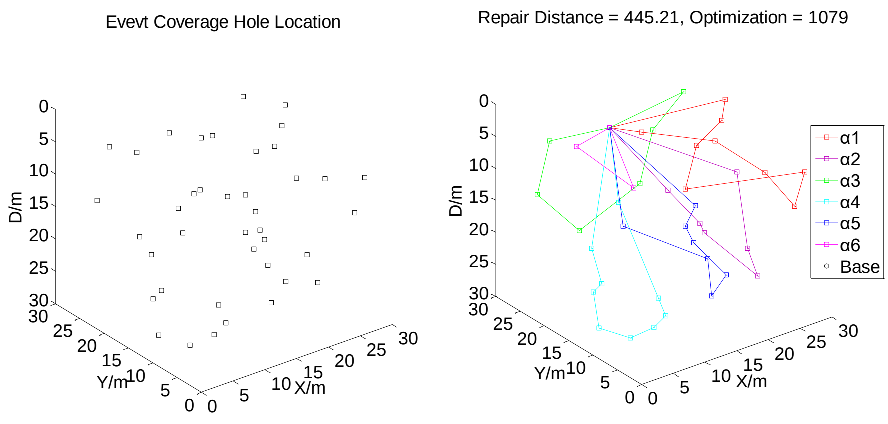
**Figure 1.** Situation 1: The underwater robot repair performance and distribution of various types of sensors in a multi-constrained underwater environment: (**a**) The distribution of underwater event coverage holes; (**b**) The repair strategy and route.

In real-world applications, when underwater monitoring missions and monitoring demands transform, the UECHs will appear. The UECHs will appear anywhere in UWSNs according to different monitoring missions and monitoring demands. The position of the UECHs is unpredictable and arbitrary. Thus, the position of the UECHs is assumed to appear discretionarily to verify if the MECHR algorithm is able to satisfy the needs of real-world applications. In the experiments, the underwater space location coordinates of the forty UECHs are discretionarily generated to verify if the MECHR algorithm is able to repair the UECHs in any underwater situation.



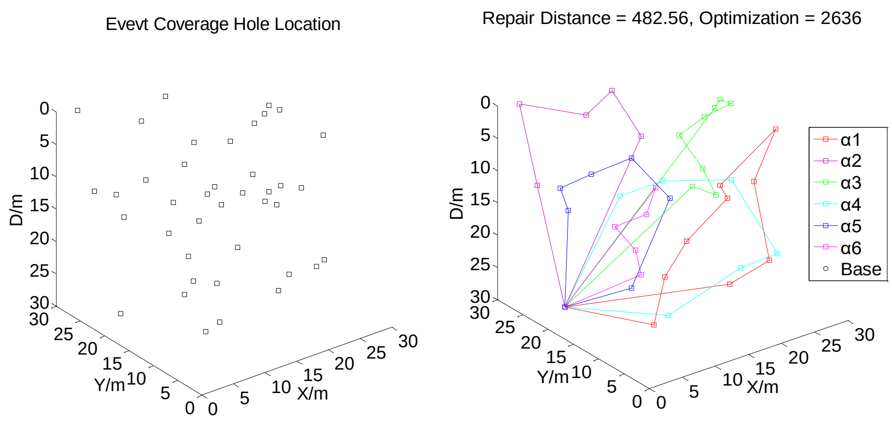

(**a**) (**b**)

**Figure 2.** Situation 2: The underwater robot repair effect and distribution of various types of sensors in a multi-constrained underwater environment: (**a**) The distribution of underwater event coverage holes; (**b**) The repair strategy and route.




(**a**) (**b**)

**Figure 3.** Situation 3: The underwater robot repair effect and distribution of various types of sensors in a multi-constrained underwater environment: (**a**) The distribution of underwater event coverage holes; (**b**) The repair strategy and route.




(**a**) (**b**)

**Figure 4.** Situation 4: The underwater robot repair performance and distribution of various types of sensors in a multi-constrained underwater environment: (**a**) The distribution of underwater event coverage holes; (**b**) The repair strategy and route.

When repairing the UECHs in the four situations, multiple underwater repair constraints need to be met at the same time. As can be seen from Figure 1, the 1st underwater situation for the repair time constraint is more stringent in four underwater situations. Meanwhile, the repair process is not merely restricted by a stringent time constraint, but it is limited by the repair space, repair cost, repair energy, and repair error constraints. Thus, the water temperature sensors that take less repair time in the repair process of the UWSNs are applied a lot. The number of water temperature sensors is 12. It can be seen from Figure 2 that the 2nd underwater situation for the repair error constraint is more rigorous in four underwater situations. The repair process of UECHs is not merely restricted by rigorous error constraints, but it is limited by other four underwater repair constraints. Therefore, the water temperature sensors with smaller repair error are used a lot. The number of water temperature sensors is 21. As can be seen from Figure 3, the 3rd underwater situation for the repair energy constraint is stricter in four underwater situations in the repair process of the UWSNs. In the meantime, the repair time, repair space, repair cost, and repair error constraints must also be met. Therefore, the water pressure sensors and water quality combination sensors that consume less repair energy are applied a lot. The number of both types of sensors is nine.

In Figure 4, the repair space constraints are the most restrictive of the four situations. The fourth underwater situation needs to satify the other four constraints under the premise of strict horizontal repair constraints, vertical repair constraints, and depth repair constraints, respectively, in the underwater environment. Additionally, the horizontal, vertical, and height constraints increase greatly. The underwater monitoring space that needs to be repaired has almost tripled compared to situation 2. However, the repair error constraint is more stringent than that in situation 2, which put forward a higher requirement to the effect of underwater event coverage hole repair. Therefore, the water pressure sensors and flow velocity sensors with a bigger repair range and lower repair error are more used in the fourth underwater situation.

In the repair process of UECHs, when there is a large number of UECHs that must be repaired, the Pareto dominance relationship will lose its effectiveness. The MECHR algorithm introduces the multi-agent strategy, allowing each agent to perform the task of guiding the subpopulation to evolve toward the real Pareto frontier. When the agent selects offspring, the fitness of the individual is a scalar index that does not need to rely on the Pareto dominance relationship. So, it can effectively distinguish the pros and cons of individuals. The MECHR algorithm introduces the diversity archive strategy to sustain the diversity of the population, and through the process of gathering the individuals in each agent to archiving and redistributing them to each agent every specified number of generations, the migration of good individuals between different tasks is realized. Therefore, the MECHR algorithm can effectively repair a large amount of UECHs in the multi-constrained underwater environments.

It can be seen from Figure 5 that the muiti-AUVs repair distance decreases rapidly in the initial optimization stage. With the continuous repair process, the muiti-AUVs repair distance continues to slow down and stabilize. Then, the repair route distance of UECHs decreases stably. There is no problem of local optimum and continuous oscillation. This is because it is often difficult to balance the diversity and convergence of the solution set when repairing a large amount of UECHs. In response to this problem, the multi-agent strategy and diversity archive strategy are combined in the MECHR algorithm. In the multi-agent strategy, each agent is responsible for the task of guiding the subpopulation to evolve toward the real Pareto frontier along the specified reference vector, which effectively enhances the convergence of the population, and the reference vectors are evenly distributed in the target space. The subpopulations in different agents can maintain better diversity. Through the process of gathering individuals who perform different tasks every few generations into the diversity archive and updating the archive, the diversity of the population can be effectively maintained. The updated diversity archive redistributes individuals to different agents. The migration of good individuals between different tasks in this process can effectively improve the performance of the algorithm. Finally, in Figure 5a, the first situation is optimized 3465 times. The multi-AUVs repair distance is 398.05 m; in Figure 5b, the second situation passes through 4828 optimizations. The repair route distance is 406.03 m; in Figure 5c, the third situation goes through 1079 optimizations. The repairc route distance is 445.21 m; in Figure 5d, after 2636 optimizations, the repair route distance of the fourth situation is 482.56 m.

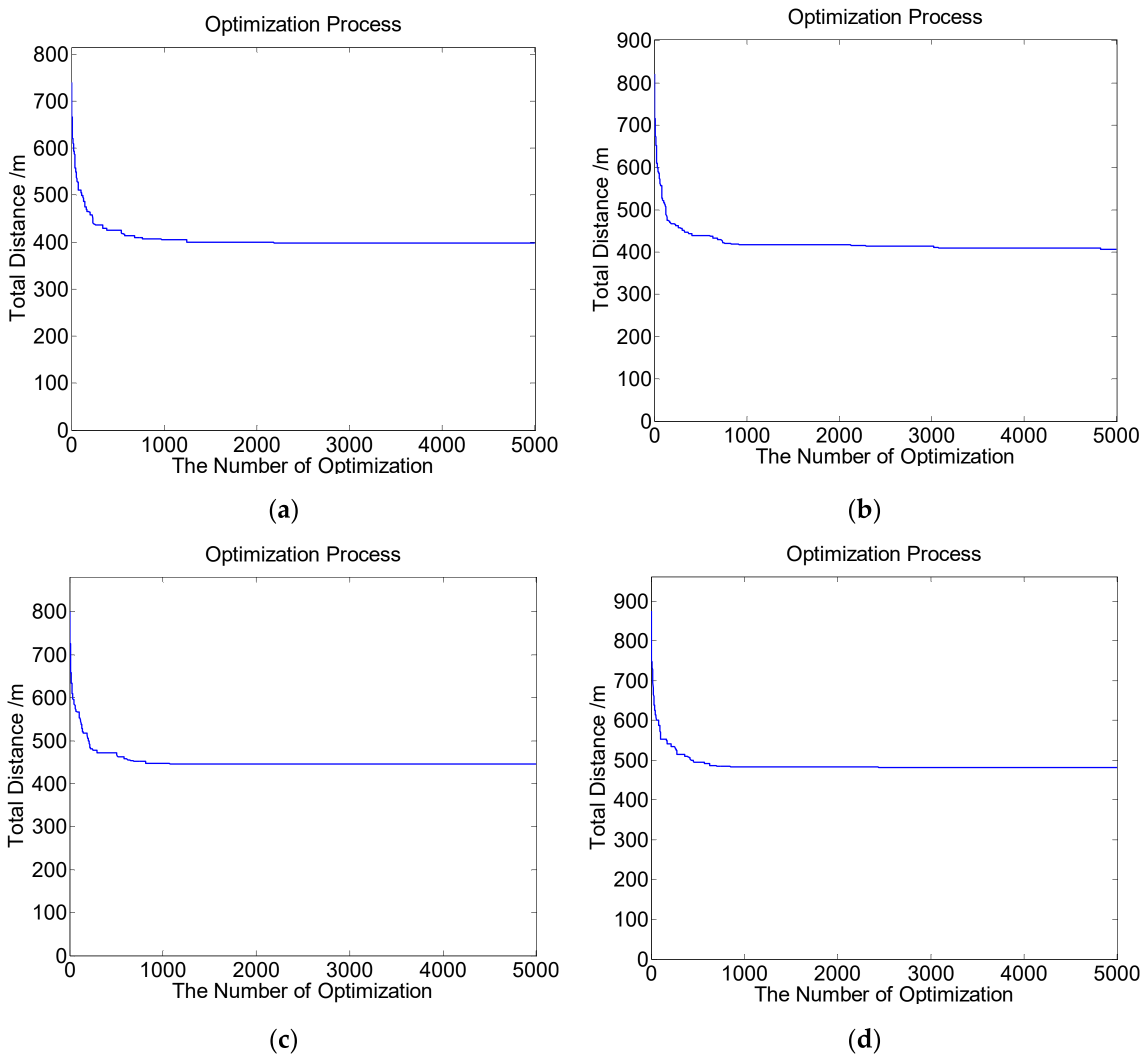



**Figure 5.** The optimization process of the multiple autonomous underwater vehicles (multi-AUVs) repair distance in four multi-constrained underwater situations. (**a**) Situation 1, (**b**) Situation 2, (**c**) Situation 3, (**d**) Situation 4.

### *5.3. Comparative Experiments*

In order to test the effect of the MECHR algorithm, the MECHR algorithm made a comparison with the latest iMOGA [24], SOS-SA [25], and JECHR algorithms [26] in the same experimental condition. Figures 6 and 7 show the performance comparison of MECHR, iMOGA, SOS-SA, and JECHR algorithms under different amounts of UECHs and energy constraints. Since there are no relevant works on a large number of UECH repairs in multi-constrained UWSNs, the MECHR algorithm made a comparison with the latest iMOGA, SOS-SA, and JECHR algorithms in the same experimental condition. These three latest multi-objective optimization algorithms are all used to solve the planning problem. Therefore, it can be seen from the experimental results that the MECHR algorithm is superior to the above algorithms in terms of repair route distance in the underwater environments with multiple event coverage holes and energy constraints.

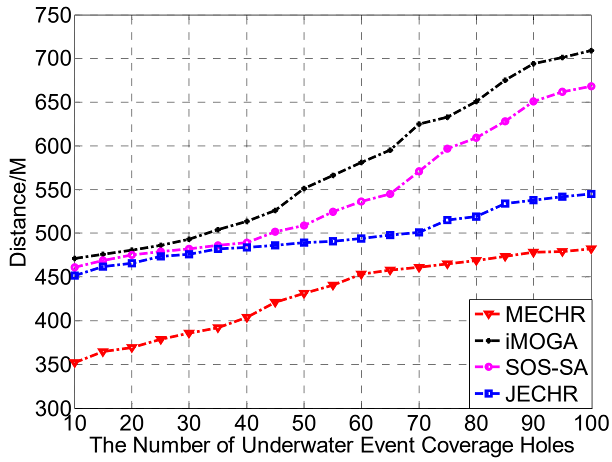
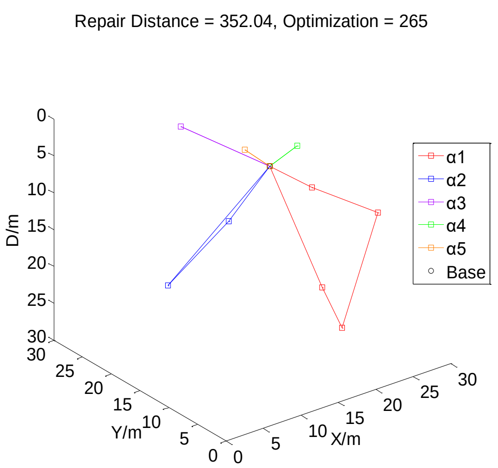
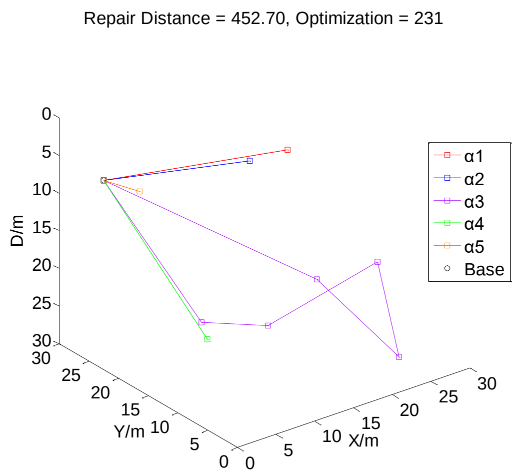


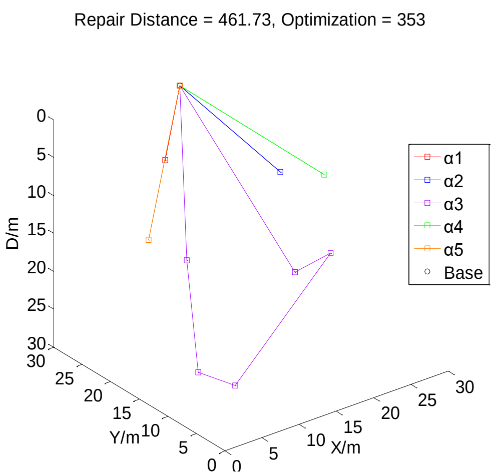

**Figure 6.** The multi-agent event coverage hole repair algorithm (MECHR) algorithm made a comparison with the inexact Multi-Objective Genetic Algorithm (iMOGA), Symbiotic Organisms Search (SOS) with Simulated Annealing algorithm (SOS-SA), and Joint Event Coverage Hole Repair algorithm (JECHR) algorithms in terms of repair route distance under different amounts of underwater event coverage hole (UECHs).



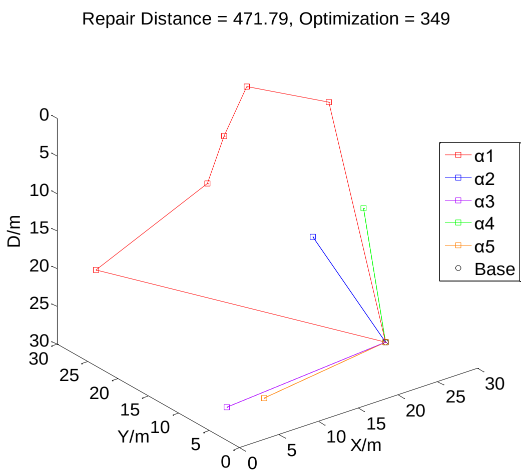

(**a**) The MECHR Algorithm




(**b**) The JECHR Algorithm




(**c**) The SOS-SA Algorithm




(**d**) The iMOGA Algorithm

**Figure 7.** The original experimental results at $N = 10$ of all four algorithms.

It can be seen from Figure 6 that at the initial stage, as the amount of UECHs continues to increase, the repair route distance of the above algorithms increases fast. When the number of underwater event coverage holes (UECHs) in the UWSNs is between 10 and 70, the density of UECHs is very low. There will be less room for further improvement. Therefore, at this stage, the selection of the base position largely determines the repair distance of multi-AUVs. The base position is the optimal starting position selected by multi-AUVs, which greatly affects the repair distance in the initial stage. Multi-AUVs set off from the base position at the same time. Each AUV carries a type of sensor to repair UECHs in the UWSNs. The MECHR algorithm introduces the multi-agent strategy, which ensures that multi-AUVs can select the best base position to start. As the amount of UECHs exceeds 70, the density of UECHs increases. The repair route distance will depend on the choice of base position and repair strategy.

In Figure 7, the original experimental results at $N$ = 10 of all four algorithms have been provided. It can be seen the MECHR algorithm is able to select the best base position, which is in the center of all UECHs. The other three algorithms tend to choose the base position at the edge, which will significantly increase the repair path distance of multi-AUVs. When the number of holes $N$ = 10, the repair distance of the MECHR algorithm is 352.04 m. The repair distance of the JECHR algorithm is 452.70 m. The repair distance of the SOS-SA algorithm is 461.73 m. The repair distance of the iMOGA algorithm is 471.79 m.

As the amount of UECHs exceeds 70, the growth rate of the repair route distance of the iMOGA and SOS-SA algorithms is still increasing, while the growth rate of the MECHR and JECHR algorithms tends to slow down and stabilize. This is because when the number of UECHs is small, the intensity of UECHs in the underwater environment is low. The AUVs have to sail a longer distance to repair UECHs in the underwater environment. With the increasing number of UECHs, the intensity of UECHs in the UWSNs is gradually increasing. In the end, the UWSNs tend to reach saturation. The MECHR and JECHR algorithms continually optimize the repair route, so that the growth rate of the underwater repair route distance gradually slows down. It can be seen that the repair performance of the MECHR algorithm is significantly better than the iMOGA, SOS-SA, and JECHR algorithms. The effect of the iMOGA algorithm is the most unsatisfactory. This is because the iMOGA algorithm is essentially a genetic algorithm that depends on the selection of the initial population to some extent. Meanwhile, the iMOGA algorithm tends to mature prematurely. The iMOGA algorithm is restricted to its own search performance in new spaces, which is apt to converge to a local optimal solution. Thus, when there is a large amount of UECHs in the UWSNs, the iMOGA algorithm cannot provide a reasonable underwater repair solution in the UWSNs. The SOS-SA algorithm is an improved simulated annealing algorithm that is essentially a greedy algorithm. Thus, the parameters are hard to control. In addition, it cannot be guaranteed to converge to the optimal value at one time, which generally requires multiple attempts to be obtained. In most circumstances, it is apt to fall into a local optimal solution. When repairing a large amount of UECHs, the MECHR algorithm combines the multi-agent strategy with a diversity archive strategy to effectively balance the diversity and convergence of the solution set. In the multi-agent strategy, each agent is responsible for the task of guiding the subpopulation to evolve toward the real Pareto frontier along the specified reference vector, which effectively enhances the convergence of the population. Since the reference vectors are evenly distributed in the target space, the subpopulations in different agents can maintain good diversity. Through the process of gathering individuals who perform different tasks every few generations into the diversity archive and updating the archive, the diversity of the population can be effectively maintained. The updated diversity archive redistributes individuals to different agents. The migration of good individuals between different tasks in this process can effectively improve the algorithm performance.

In order to test the effect of the MECHR algorithm under different underwater repair energy constraints, the MECHR algorithm made a comparison with iMOGA, SOS-SA, and JECHR algorithms. So far, there is no relevant work in the field of UECH repair under multi-constraints. Compared with the ground environments, it is very hard to supplement energy in the underwater environments. Thus, the repair of underwater event coverage holes will be subject to strict energy constraints. The AUVs need to sail the shortest repair route distance under limited energy constraints to repair all the UECHs in the underwater environments. It can be seen from Figure 8 that as energy constraints continue to grow, more sensors have to be deployed to repair the UECHs in the UWSNs. The AUVs have to sail a greater repair route distance to complete the repair of the UWSNs. It shows that the repair performance of the MECHR algorithm is significantly better than the iMOGA, SOS-SA, and JECHR algorithms in terms of the repair route distance. The iMOGA algorithm depends on the selection of the initial population to some extent. At the same time, the multi-objective genetic algorithm tends to mature prematurely. It is also apt to converge to local optimum. The JECHR algorithm is unable to adapt to different underwater repair energy constraints in the UWSNs, and it is greatly affected by the penalty factor. The SOS-SA algorithm has inferior global optimization competences. Furthermore, it is easily affected by parameters.

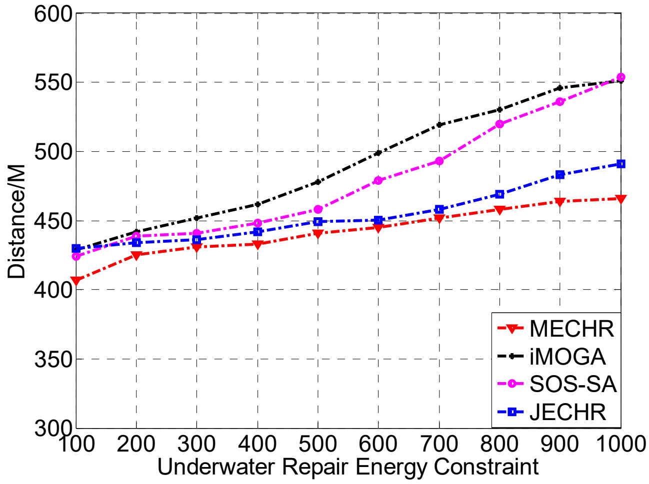



**Figure 8.** The MECHR algorithm made a comparison with the iMOGA, SOS-SA, and JECHR algorithms in terms of repair route distance under different underwater repair energy constraints.

The MECHR algorithm introduces the multi-agent strategy, allowing each agent to perform the task of guiding the evolution of the subpopulation toward the real Pareto frontier. When the agent selects the offspring, the fitness of the individual is a scalar index, which does not need to rely on Pareto dominance. Therefore, it is possible to effectively distinguish the pros and cons of individuals. The MECHR algorithm adopts the diversity archive strategy to maintain population diversity. Then, through the process of gathering the individuals in each agent to archives and redistributing them to each agent every specified number of generations, the migration of good individuals between different tasks is realized. The MECHR algorithm is able to effectively surmount the influence of repair energy constraints on the repair route distance in the UWSNs. Therefore, the MECHR algorithm can more efficiently repair UECHs under different repair energy constraints.

## 6. Conclusions

In this paper, a multi-agent event coverage hole repair algorithm that combines a multi-agent strategy with a diversity archive strategy is proposed with the multi-AUVs to repair underwater event coverage holes in harsh underwater environments. In the multi-agent strategy, each agent is responsible for the task of guiding the subpopulation to evolve toward the real Pareto frontier along the specified reference vector, which effectively strengthens the convergence of the repair strategy. Through the process of gathering individuals who perform different tasks every few generations into the diversity archive and updating the archive, the diversity of repair strategies can be effectively maintained. The MECHR algorithm efficiently solves UECH repair problems by combining the multi-agent strategy and diversity archive strategy to repair the UECHs in multi-constrained UWSNs.

Finally, the underwater repair simulation experiments demonstrate that the MECHR algorithm can efficaciously repair the UECHs in harsh multi-constrained underwater environments.

**Author Contributions:** Y.Z. and H.X. put forward the idea of the paper; Y.Z., C.W., and H.W. designed and performed the experiments; Y.Z., C.W., and H.W. analyzed the data; Z.Z., Q.J., and L.L. contributed analysis tools; Y.Z. wrote the paper. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the China Postdoctoral Science Foundation 2020M670778; the Northeastern University Postdoctoral Research Fund 20200308; the Scientific Research Foundation of Liaoning Provincial Education Department LT2020002; the Fundamental Research Funds for the Central Universities N2026006, N2011001, N2026005, N181602014, N2026004, N2026001, and N172604004; the China Scholarship Council; the National Natural Science Foundation of China under Grants 61701101, 61973093, U1713216, 61901098, 61971118, and 61973063; the National Key Robot Project grant number\2017YFB1300900; Shenyang NEU New Industrial Technology Research Institute (17-500-8-01); Liaoning Key R&D Project 2020JH2/10100040. And The APC was funded by the China Postdoctoral Science Foundation 2020M670778, and the Fundamental Research Funds for the Central Universities N2011001.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zhou, Z.; Xing, R.; Duan, Y.; Zhu, Y.; Xiang, J. Event Coverage Detection and Event Source Determination in Underwater Wireless Sensor Networks. *Sensors* **2015**, *15*, 31620–31643. [CrossRef] [PubMed]
2. Jiang, P.; Xu, Y.; Liu, J. A distributed and energy-efficient algorithm for event k-coverage in underwater sensor networks. *Sensors* **2017**, *17*, 186. [CrossRef]
3. Xiang, Y.; Xuan, Z.; Tang, M.; Zhang, J.; Sun, M. 3D space detection and coverage of wireless sensor network based on spatial correlation. *J. Netw. Comput. Appl.* **2016**, *61*, 93–101. [CrossRef]
4. Wang, H.; Li, Y.; Chang, T.; Chang, S.; Fan, Y. Event-Driven Sensor Deployment in an Underwater Environment Using a Distributed Hybrid Fish Swarm Optimization Algorithm. *Appl. Sci.* **2018**, *8*, 1638. [CrossRef]
5. Deng, X.; Tang, Z.; Yang, L.T.; Lin, M.; Wang, B. Confident Information Coverage Hole Healing in Hybrid Industrial Wireless Sensor Networks. *IEEE Trans. Ind. Inform.* **2018**, *14*, 2220–2229. [CrossRef]
6. Wang, Q.; Kong, F.; Wang, M.; Wang, H. Optimized Charging Scheduling with Single Mobile Charger for Wireless Rechargeable Sensor Networks. *Symmetry* **2017**, *9*, 285. [CrossRef]
7. Wu, H.; Meng, Q.; Xian, J.; Mei, X.; Claramunt, C.; Cao, J. An Information Entropy Based Event Boundary Detection Algorithm in Wireless Sensor Networks. *Symmetry* **2019**, *11*, 537. [CrossRef]
8. Miskowicz, M. *Event-Based Control and Signal Processing*; CRC Press: Boca Raton, FL, USA, 2016.
9. Wang, Z.; Liu, M.; Zhang, S.; Qiu, M. Sensor virtualization for underwater event detection. *J. Syst. Archit.* **2014**, *60*, 619–629. [CrossRef]
10. Jing, H. Coverage holes recovery algorithm based on nodes balance distance of underwater wireless sensor network. *Int. J. Smart Sens. Intell. Syst.* **2014**, *7*, 1890–1907. [CrossRef]
11. Yu, S.; Xu, Y.; Jiang, P.; Wu, F.; Xu, H. Node self-deployment algorithm based on pigeon swarm optimization for underwater wireless sensor networks. *Sensors* **2017**, *17*, 674. [CrossRef]
12. Alam, K.M.; Kamruzzaman, J.; Karmakar, G.; Murshed, M. Dynamic adjustment of sensing range for event coverage in wireless sensor networks. *J. Netw. Comput. Appl.* **2014**, *46*, 139–153. [CrossRef]
13. Dunbabin, M.; Corke, P.; Vasilescu, I.; Rus, D. Data muling over underwater wireless sensor networks using an autonomous underwater vehicle. In Proceedings of the Proceedings 2006 IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 2091–2098.
14. Ahmed, Z.; Bakar, K.A.; Abbas, M.Z.; Ayaz, M.; Aggoune, E.-H.M. Auv path planning based efficient routing for underwater linear sensor networks. *J. Theor. Appl. Inf. Technol.* **2019**, *97*, 4–30.
15. Han, G.; Long, X.; Zhu, C.; Guizani, M.; Bi, Y.; Zhang, W. An AUV location prediction-based data collection scheme for underwater wireless sensor networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6037–6049. [CrossRef]
16. Han, G.; Long, X.; Zhu, C.; Guizani, M.; Zhang, W. A high-availability data collection scheme based on multi-AUVs for underwater sensor networks. *IEEE Trans. Mob. Comput.* **2019**, *19*, 1010–1022. [CrossRef]
17. Latif, K.; Javaid, N.; Ahmad, A.; Khan, Z.A.; Alrajeh, N.; Khan, M.I. On energy hole and coverage hole avoidance in underwater wireless sensor networks. *IEEE Sens. J.* **2016**, *16*, 4431–4442. [CrossRef]
18. Azam, I.; Javaid, N.; Ahmad, A.; Abdul, W.; Almogren, A.; Alamri, A. Balanced load distribution with energy hole avoidance in underwater WSNs. *IEEE Access* **2017**, *5*, 15206–15221. [CrossRef]

19. Wang, Z.; Wang, B. A novel node sinking algorithm for 3D coverage and connectivity in underwater sensor networks. *Ad Hoc Netw.* **2017**, *56*, 43–55. [CrossRef]
20. Khan, F.A.; Khan, S.A.; Turgut, D.; Bölöni, L. Greedy path planning for maximizing value of information in underwater sensor networks. In Proceedings of the 39th Annual IEEE Conference on Local Computer Networks Workshops, Edmonton, AB, Canada, 8–11 September 2014; pp. 610–615.
21. Zhang, Y.; Wang, M.; Liang, J.; Zhang, H.; Chen, W.; Jiang, S. Coverage enhancing of 3D underwater sensor networks based on improved fruit fly optimization algorithm. *Soft Comput.* **2017**, *21*, 6019–6029. [CrossRef]
22. Nam, H. Data-gathering protocol-based AUV path-planning for long-duration cooperation in underwater acoustic sensor networks. *IEEE Sens. J.* **2018**, *18*, 8902–8912. [CrossRef]
23. Jin, Z.; Zhao, Q.; Luo, Y. Routing Void Prediction and Repairing in AUV-Assisted Underwater Acoustic Sensor Networks. *IEEE Access* **2020**, *8*, 54200–54212. [CrossRef]
24. Maity, S.; Roy, A.; Maiti, M. An imprecise multi-objective genetic algorithm for uncertain constrained multi-objective solid travelling salesman problem. *Expert Syst. Appl.* **2016**, *46*, 196–223. [CrossRef]
25. Ezugwu, A.E.-S.; Adewumi, A.O.; Frîncu, M.E. Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem. *Expert Syst. Appl.* **2017**, *77*, 189–210. [CrossRef]
26. Zhuang, Y.; Wu, C.; Wu, H.; Chu, H.; Gao, Y.; Li, L. The Repair Strategy for Event Coverage Holes Based on Mobile Robots in Wireless Sensor and Robot Networks. *Sensors* **2019**, *19*, 5045. [CrossRef] [PubMed]
27. Das, I.; Dennis, J.E. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J. Optim.* **1998**, *8*, 631–657. [CrossRef]
28. Xiang, Y.; Zhou, Y.; Li, M.; Chen, Z. A vector angle-based evolutionary algorithm for unconstrained many-objective optimization. *IEEE Trans. Evol. Comput.* **2016**, *21*, 131–152. [CrossRef]
29. Gupta, A.; Ong, Y.-S.; Feng, L. Multifactorial evolution: Toward evolutionary multitasking. *IEEE Trans. Evol. Comput.* **2015**, *20*, 343–357. [CrossRef]