

## Article

# Research on Multiple AUVs Task Allocation with Energy Constraints in Underwater Search Environment

Hailin Wang<sup>1,2,3,\*</sup>, Yiping Li<sup>1,2,3</sup>, Shuo Li<sup>1,2,3</sup> and Gaopeng Xu<sup>2,3</sup><sup>1</sup> University of Chinese Academy of Sciences, Beijing 100049, China; lyp@sia.cn (Y.L.); shuoli@sia.cn (S.L.)<sup>2</sup> State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; xugaopeng@sia.cn<sup>3</sup> Key Laboratory of Marine Robotics, Shenyang 110016, China

\* Correspondence: wanghailin@sia.cn; Tel.: +86-1800-248-1021

**Abstract:** The allocation of tasks among multiple Autonomous Underwater Vehicles (AUVs) with energy constraints in underwater environments presents an NP-complete problem with far-reaching consequences for marine exploration, environmental monitoring, and underwater construction. This paper critically examines the contemporary methodologies and technologies in the task allocation for multiple AUVs, with a particular focus on strategies that optimize navigation time with energy consumption constraints. By conceptualizing the multiple AUVs task allocation issue as a Capacitated Vehicle Routing Problem (CVRP) and addressing it using the SCIP solver, this study seeks to identify effective task allocation strategies that enhance the operational efficiency and minimize the mission duration in energy-restricted underwater settings. The findings of this research provide valuable insights into efficient task allocation under energy constraints, providing useful theoretical implications and practical guidance for optimizing task planning and energy management in multiple AUVs systems. These contributions are demonstrated through the improved solution quality and computational efficiency.

**Keywords:** multiple AUVs; task allocation; underwater search; constraint programming



**Citation:** Wang, H.; Li, Y.; Li, S.; Xu, G. Research on Multiple AUVs Task Allocation with Energy Constraints in Underwater Search Environment. *Electronics* **2024**, *13*, 3852. <https://doi.org/10.3390/electronics13193852>

Academic Editor: Paul Mitchell

Received: 20 August 2024

Revised: 18 September 2024

Accepted: 27 September 2024

Published: 28 September 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Autonomous Underwater Vehicles (AUVs) have emerged as essential instruments in oceanographic research, underwater surveillance, and resource exploration. Their capacity to function autonomously in hostile and hard-to-reach environments renders them indispensable for a variety of tasks, including scientific data collection and military reconnaissance [1]. However, an underwater mission poses distinct challenges, particularly regarding energy limitations and communication barriers. Efficient task allocation among multiple AUVs is important for optimizing their performance and fulfilling the mission objectives within these constraints [2].

In recent years, the task allocation problem for multiple AUVs has garnered significant attention and has been modeled as a Capacitated Vehicle Routing Problem (CVRP), which has been proven to be NP-complete. Several approaches have been proposed to address this problem, including optimization algorithms such as Quantum Particle Swarm Optimization (QPSO), Reinforcement Learning (RL), and Multi-Objective Discrete Particle Swarm Optimization combined with Simulated Annealing (MODPSO-SA). These methods have effectively improved the computational efficiency and reduced the energy consumption in task allocation and path planning.

In practical applications, task allocation in multiple AUVs systems is influenced by several environmental factors, including interference from ocean currents, the dynamic constraints of AUVs, and energy consumption. To address these challenges, researchers have proposed various algorithms and models. For example, Pang et al. (2021) [3] introduced a QPSO-based task allocation method that can find globally optimal task allocation plans for

multiple AUVs systems in a constant ocean current environment, significantly improving the task execution efficiency by optimizing the path length and energy consumption. Similarly, Lu et al. (2020) [4] developed an MODPSO-SA algorithm that addresses local extrema in task allocation and improves the global search capabilities and energy efficiency.

The task allocation problem for multiple AUVs has also been approached as a time-constrained optimization issue. Wang et al. (2023) [5] proposed a constraint-driven optimal control method that uses a minimum input norm objective to achieve energy-aware task execution. This method enhances computational efficiency through quadratic programming and avoids task conflicts by adjusting the task priorities, ensuring reliable and energy-efficient task execution.

Reinforcement Learning has also gained traction in multiple AUVs systems, particularly in optimizing task allocation in ocean current environments. Ding and Zheng (2022) [6] designed a reward function to optimize task allocation and introduced the Automatic Policy Amendment Algorithm (APAA) to overcome the slow convergence issue in the traditional RL approaches. The algorithm leverages cumulative task rewards to rapidly identify optimal task allocation strategies under complex environmental conditions.

Additionally, research has focused on workload balance and path planning in multiple AUVs systems. Chen and Zhu (2019) [7] proposed a workload-balanced algorithm that dynamically assigns tasks through a self-organizing map (SOM) neural network, ensuring rational task distribution and avoiding energy wastage due to suboptimal path selection. Qu et al. (2019) [8] also developed an optimized SOM algorithm that reduces computational time and energy consumption by addressing invalid task assignments.

In the energy-constrained environment of multiple AUVs task execution, optimizing task allocation and path planning to minimize task execution time and maximize energy efficiency remains a significant challenge. Bychkov et al. (2019) [9] introduced a two-level dynamic mission planner that combines evolutionary algorithms with path planning techniques to manage energy and task allocation during long-term missions. Sarkar et al. (2018) [10] presented a nearest-neighbor clustering and routing algorithm (nCAR) that significantly improves the task allocation execution efficiency in large-scale problems and reduces the energy consumption.

In summary, the underwater search environment is defined by limited energy resources, communication challenges, and dynamic environmental conditions [11]. These obstacles demand innovative task allocation strategies to enable multiple AUVs to collaborate effectively and efficiently. Energy constraints are especially crucial as they significantly influence the operational range and duration of AUV missions. The application of a CVRP framework to multiple AUVs task allocation provides useful theoretical implications and practical guidance for optimizing task planning and energy management. By combining techniques such as Particle Swarm Optimization, Reinforcement Learning, and Simulated Annealing, researchers have significantly improved the quality of the solutions and computational efficiency, providing strong support for the application of multiple AUVs systems in complex underwater environments.

This research addresses the multiple AUVs task allocation problem by modeling it as a Capacitated Vehicle Routing Problem (CVRP) and utilizing the SCIP solver to propose a strategy that optimizes both navigation time and energy consumption. While several approaches have been proposed in the field of multiple AUVs task allocation, few studies have comprehensively addressed the combination of global optimality, energy consumption optimization, and mission duration. By integrating the mathematical model of a CVRP, this study offers a novel theoretical framework to tackle this complex problem. Furthermore, the use of the SCIP solver significantly enhances the computational efficiency of task allocation solutions, enabling faster and more accurate identification of optimal task plans in real-world multiple AUVs systems. The contributions of this research not only improve the solution quality but also provide practical guidance for enhancing the operational efficiency of multiple AUVs systems in complex underwater environments. The findings hold important theoretical implications and offer actionable insights for optimizing task

planning and energy management in future marine applications, supporting sustainable ocean resource exploration.

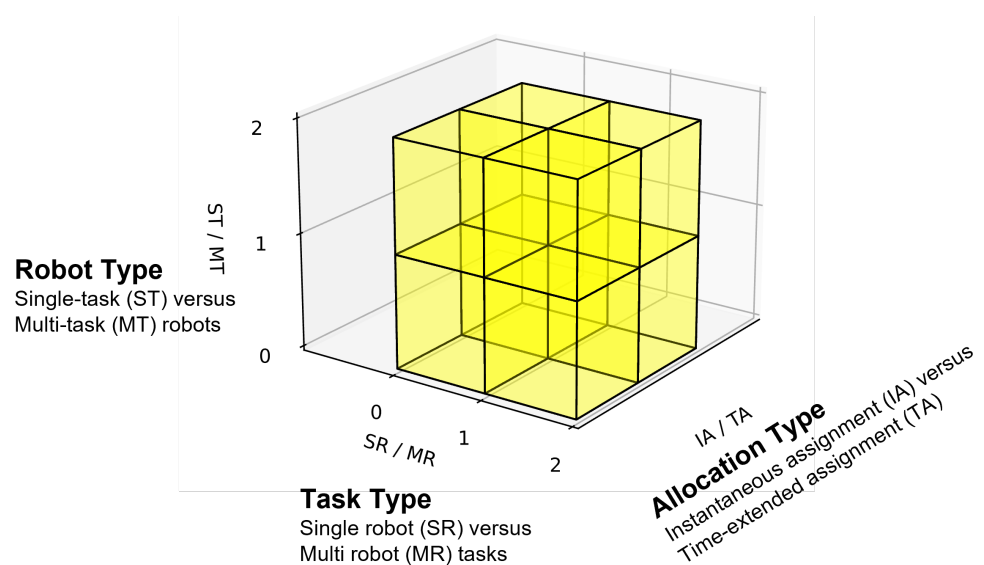
In Section 2, we provide in-depth background on the multiple AUVs task allocation problem, detailing the challenges and complexity of energy-constrained underwater environments. We explore different task allocation algorithms and introduce the SCIP solver, which plays a key role in solving the optimization problem. Section 3 focuses on the methodology, including the problem formulation, modeling approach, and the algorithms and frameworks used in our study. We also analyze the optimality and existence of solutions, ensuring the effectiveness of our proposed method. Section 4 discusses the simulation setup and results, along with a comparative analysis between our algorithm and the Particle Swarm Optimization (PSO) algorithm, comparing performance, computational time, and machine configuration. Finally, in Section 5, we summarize the practical implications of our findings and suggest potential directions for future research.

## 2. Background

### 2.1. Multiple AUVs Task Allocation Problem

Multiple Robots Task Allocation (MRTA) is a vital component of multi-robot systems, aiming to allocate tasks among robots to maximize the overall efficiency and effectiveness. In the context of multiple AUVs, task allocation involves assigning tasks to optimize mission performance while accounting for the capabilities and constraints of each AUV. Depending on the nature of the tasks, AUVs with similar functions may be needed to perform specific tasks, while more complex missions may necessitate the cooperation of heterogeneous AUV clusters.

In 2004, Gerkey and Mataric [12] introduced a taxonomy for multi-robot task allocation according to Figure 1, distinguishing the task type, assignment type, and robot type. According to this taxonomy, task allocation in underwater search environments is categorized as SR-MT-IA, meaning that each task can be completed by a single robot, each robot can handle multiple tasks, and the task assignments are allocated instantaneously.



**Figure 1.** Visual representation of the three axes of Gerkey and Mataric's taxonomy.

Wang et al. [13] conducted a comprehensive analysis of the existing literature on task allocation models and algorithms for multiple AUVs systems, providing a macro-level overview to help researchers enhance these algorithms more thoroughly.

In this paper, the modeling of task allocation for multiple AUVs leverages mathematical methods such as combinatorial optimization and operations research. After establishing the mathematical model, we propose corresponding algorithms tailored to it.

Miloradovic et al. [14] formulated MRTA using a Mixed-Integer Linear Programming (MILP) approach, capitalizing on the potential parallelism of the robots.

## 2.2. Task Allocation Algorithms

The approaches to task allocation include centralized and decentralized algorithms, heuristic methods, and optimization techniques [13].

Centralized algorithms involve a central controller assigning tasks to AUVs, optimizing the task allocation on a global scale. However, they are often constrained by the communication range and reliability [15]. In contrast, decentralized algorithms enable AUVs to make independent decisions based on local information, offering greater robustness against communication failures but potentially failing to achieve globally optimal solutions.

In this study, we examine an underwater search environment where AUVs are initially positioned near shore, with the mission and task information disseminated by an information center. At this stage, the AUVs are on the water surface and can communicate via radio. To achieve optimal task allocation, we assume ideal communication conditions and employ a centralized task allocation method.

## 2.3. SCIP Solver

The SCIP (Solving Constraint Integer Programs) solver is a state-of-the-art software tool designed to tackle mixed-integer linear programming (MILP) and constraint integer programming (CIP) problems [16]. Developed by the Zuse Institute Berlin (ZIB), SCIP incorporates features from branch-and-bound algorithms, cutting planes, heuristics, and domain propagation techniques, making it one of the most versatile and powerful solvers available.

SCIP's capability to handle various problem structures, including linear and nonlinear constraints, renders it particularly suitable for complex optimization challenges across fields such as logistics, energy systems, and telecommunications. A key strength of SCIP is its flexibility, allowing users to customize the solving process by implementing specific heuristics and branching strategies tailored to their unique problem instances.

In the context of multiple AUVs task planning, SCIP plays a critical role in optimizing task allocation under energy constraints. The task allocation problem for AUVs can be effectively modeled as a Capacitated Vehicle Routing Problem (CVRP), where the goal is to determine the most efficient routes for a fleet of AUVs to complete a set of tasks while minimizing the energy consumption and ensuring that each AUV's energy capacity constraints are met.

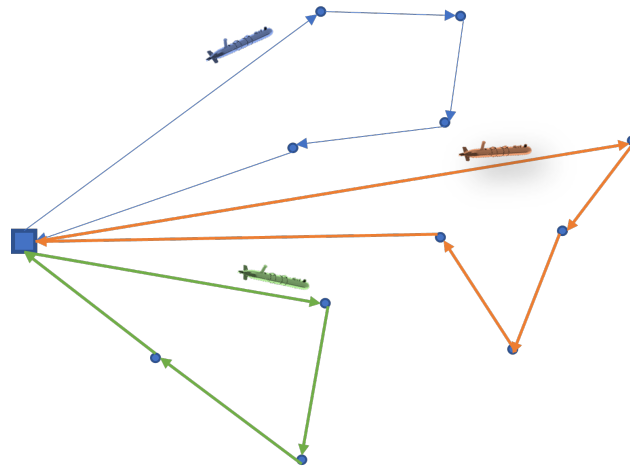
By employing SCIP to solve the CVRP, researchers can leverage its advanced algorithmic capabilities to discover optimal task allocation strategies [17]. SCIP's branch-and-cut framework, combined with its robust cutting planes and heuristics, facilitates efficient exploration of the solution space, even for the large-scale and complex problem instances typical of underwater environments.

## 3. Methodology

### 3.1. Problem Formulation and Modeling

The task allocation problem is formulated as an optimization problem with the objective of minimizing the total execution time of AUVs. The constraints include the power energy limits of the AUVs, the energy requirements of the tasks, and the maximum traveling distance of the AUVs.

The task allocation for multiple AUVs under energy constraints in an underwater search environment is modeled as a Capacitated Vehicle Routing Problem (CVRP). As illustrated in Figure 2, three AUVs, each with varying power energy capacities, must ensure sufficient power for their assigned task nodes. The limited energy capacity of each AUV restricts the number of task nodes it can serve. This energy constraint also prevents highly capable AUVs from completing all task nodes in a single mission.



**Figure 2.** Example of 11 task nodes searched and identified by 3 AUVs.

To model the problem, we need to define some variables. The set of task nodes is  $X = \{t_0, t_1, \dots, t_n\}$  ( $X' = X \setminus \{t_0\}$ ), where the first node  $t_0$  represents the starting point of the AUVs, assuming all AUVs depart from the shore base. The set of AUVs is  $V = \{v_1, v_2, \dots, v_m\}$ . In this model, the energy consumption for an AUV to service task node  $t_j$  is determined by the energy demand of the task node, which is  $Pu(j)$ . The position of task nodes  $t_j$  is  $x_j$ , provided by its 2-dimensional coordinates  $x_j^x, x_j^y$ , and the Euclidean distance between task nodes  $x_i$  and  $x_j$  is calculated as  $d(x_i, x_j) = \sqrt{(x_i^x - x_j^x)^2 + (x_i^y - x_j^y)^2}$ .

In practice, different AUVs carry different amounts of energy. For simplicity, the energy capacity of AUV  $v_i$  is modeled as  $Au(i)$ , indicating that AUV  $v_i$  carries  $Au(i)$  units of energy. And the speed of AUV  $v_i$  is represented by  $Sv(i)$ .

We denote the time required to serve  $t_i$  as  $Tp(i)$ . Consequently, the energy carried by each AUV limits its capacity to serve task nodes. The model employs two objective functions: one aims to minimize the maximum AUV traveling time (min-max) and the other seeks to minimize the total AUV traveling time. Given the complexity of energy consumption along the path, the optimization objective inherently includes minimizing the total task completion time, which implies finding the shortest path.

$$\min \max_{v \in V} \left( \sum_{i \in X} \sum_{j \in X} \left( \frac{d(x^i, x^j)}{Sv(v)} + Tp(i)w_{ij}^v \right) \right) \tag{1}$$

$$\min \sum_{v \in V} \sum_{i \in X} \sum_{j \in X} \left( \frac{d(x^i, x^j)}{Sv(v)} + Tp(i)w_{ij}^v \right) \tag{2}$$

In this model,  $w_{ij}^v$  is a binary variable, where  $w_{ij}^v = 1$  indicates that AUV  $v$  moves from task node  $i$  to task node  $j$  and serves both, and  $w_{ij}^v = 0$  otherwise.

In summary, the task allocation problem for a multiple AUVs system involves a heterogeneous fleet of AUVs departing from the shore base, with each AUV selecting subset of task nodes. The AUVs must satisfy certain constraints, ensuring that the energy carried by each AUV can sequentially complete the tasks at each task node position and return to the shore base. Based on these requirements, our constraints are as follows:

$$\sum_{v \in V} \sum_{j \in X} w_{ij}^v = 1, \quad i \in X' \quad (3)$$

$$w_{ij}^v - w_{jk}^v = 0, v \in V, \quad j \in X, i \in X, i \neq j \neq k \quad (4)$$

$$\sum_{i \in X} \sum_{j \in X} w_{ij}^v P_u(i) \leq A_u(v), \quad v \in V \quad (5)$$

$$\sum_{i \in X} \sum_{j \in X} w_{ij}^v d(x_i, x_j) \leq \text{MaxPath}(v), \quad v \in V \quad (6)$$

$$\sum_{i \in X} w_{ij}^v - \sum_{jk} w_{jk}^v = 0, \quad v \in V, i, j, k \in X', i \neq j \neq k \quad (7)$$

$$w_{ij}^v = \{0, 1\}, \quad v \in V, i \in X, j \in X \quad (8)$$

Constraints (3) and (4) ensure that each task node is visited only once and that each path is completed by the same AUV. Constraint (5) ensures that the energy demands are met before visiting a task node. Constraint (6) ensures that the distances traveled by each AUV do not exceed its present power capacity. Constraint (7) guarantees the continuity of traveling different task nodes.

### 3.2. Algorithms and Frameworks

This Algorithm 1 tackles the problem of task allocation challenges for multiple Autonomous Underwater Vehicles (AUVs) during their missions. It establishes a constraint model based on the CVRP and solves this constrained model to ensure optimal task allocation of each AUV while adhering to energy and path length limitations.

#### Input Parameters

- $X = \{x_i\}_{i=0}^n$ : Set of task nodes, where  $x_0$  is the starting point, and the remaining nodes are target points.
- $V = \{v_i\}_{i=1}^m$ : Set of AUVs.
- $P_u[j]$ : Energy required to complete task  $j$ .
- $A_u[v]$ : Energy capacity of AUV  $v$ .
- $dist[i, j]$ : Distance between node  $i$  and node  $j$ .
- $MaxPathLength[v]$ : Maximum path length for AUV  $v$ .

#### Output

Optimal paths for each AUV.

#### Algorithm Steps

1. **Initialize Remaining Energy:** Initialize the remaining energy variable for all AUVs.
2. **Calculate Node Distances:** Calculate the Euclidean distance between each pair of nodes and compute the energy required to reach each target point based on the energy consumption model.
3. **Set Constraints:**
  - **No Immediate Return Constraint:** Ensure that an AUV does not immediately return to the same node after completing a task.
  - **One AUV per Task Constraint:** Ensure that each target point is serviced by exactly one AUV.
  - **Path Continuity Constraint:** Ensure path continuity for each AUV between consecutive nodes.
  - **Energy Limit Constraint:** Ensure that the total energy consumed by each AUV does not exceed its energy capacity after completing all assigned tasks.
  - **Path Length Limit Constraint:** Ensure that the total travel distance for each AUV does not exceed its maximum path length.

4. **Optimize Solution:** Call the optimizer to solve the above constraint model and obtain the optimal path for each AUV.

---

**Algorithm 1:** SCIP-based MRTA-CVRP algorithm
 

---

**Input:**  $X = \{x_i\}_{i=0}^n$  (Task Nodes),  $V = \{v_i\}_{i=1}^m$  (AUVs),  $P_u[j]$  (Energy required for task  $j$ ),  $A_u[v]$  (Energy capacity of AUV  $v$ ),  $dist[i, j]$  (Distance between  $i$  and  $j$ ),  $MaxPathLength[v]$  (Maximum path length for AUV  $v$ )

**Output:** Optimal paths for each AUV

```

foreach  $i \in X$  do
  | foreach  $j \in X$  do
  | | if  $i \neq j$  then
  | | |  $model.addCons(w[i, j, v] + w[j, i, v] \leq 1, No\_Immediate\_Return\_)$ 
  | | end
  | end
end
foreach  $j \in list(X)[1 : ]$  do
  |  $model.addCons(\sum_{v \in V} \sum_{i \in X, i \neq j} w[i, j, v] = 1, One\_AUV\_per\_Task\_)$ 
end
foreach  $v \in V$  do
  | foreach  $j \in list(X)[1 : ]$  do
  | |  $model.addCons(\sum_{i \in X, i \neq j} w[i, j, v] - \sum_{k \in X, j \neq k} w[j, k, v] = 0, Continuity\_)$ 
  | end
end
foreach  $v \in V$  do
  |  $model.addCons(\sum_{i \in X} \sum_{j \in X, i \neq j} w[i, j, v] \cdot P_u[j] \leq A_u[v], Power\_Limit\_)$ 
end
foreach  $v \in V$  do
  |  $model.addCons(\sum_{i \in X} \sum_{j \in X, i \neq j} w[i, j, v] \cdot dist[i, j] \leq MaxPathLength[v],$ 
  |  $Max\_Distance\_Limit\_)$ 
end
 $model.optimize();$ 
Result:  $travel\_dict$ 

```

---

### 3.3. Analysis of the Optimality and Existence of Solutions

In this section, we analyze both the optimality and the existence of solutions when solving the multiple AUVs task allocation problem using the SCIP solver. The SCIP solver, known for its robustness in handling Mixed-Integer Linear Programming (MILP) problems, was employed to ensure that the solutions obtained are either globally optimal or, in the case of infeasibility, report a lack of feasible solutions.

#### 3.3.1. Optimality of Solutions

The SCIP solver guarantees optimality for solvable MILP problems through a combination of branch-and-bound algorithms, cut-plane methods, and other optimization techniques. For problems within a reasonable scale, such as the multiple AUVs task allocation problem studied here, the solver is capable of finding global optimal solutions within acceptable computational times. For instance, in the case of a problem involving 4 AUVs and 10 tasks, the solver successfully identified the optimal solution with the output in Table 1.

In this example, the primal and dual bounds converged perfectly (Primal Bound = 9680.47; Dual Bound = 9680.47), with a gap of 0.00, confirming that the solution is globally optimal. The results were obtained within a computational time of just 1 s, demonstrating the solver's efficiency for small-scale task allocation problems. This reinforces the viability of using the SCIP solver in scenarios where global optimality is required under operational constraints.

**Table 1.** SCIP solver results for 4 AUVs and 10 tasks.

Parameter	Value
SCIP Status	problem is solved [optimal solution found]
Solving Time (s)	1.00
Solving Nodes	1 (total of 2 nodes in 2 runs)
Primal Bound	$+9.68047032967033 \times 10^3$ (6 solutions)
Dual Bound	$+9.68047032967033 \times 10^3$
Gap	0.00%
Objective value	9680.47032967033

### 3.3.2. Existence of Solutions

While the SCIP solver guarantees the optimality of solutions for feasible problems, it also effectively identifies cases where no feasible solution exists. This is particularly important when dealing with real-world operational constraints, such as battery life or maximum path lengths for AUVs. For instance, when solving a problem with 4 AUVs and 20 task nodes, the solver output the following in Table 2.

In this case, the SCIP solver determined that the problem was infeasible under the current constraints. This infeasibility likely arises from excessive limitations imposed by the problem, such as the AUVs' battery capacities or task allocation requiring paths beyond feasible operational distances. The solver's ability to identify these infeasible scenarios is crucial for adjusting the model or constraints to ensure that feasible solutions can be found in practical applications.

**Table 2.** SCIP solver results for 4 AUVs and 20 tasks.

Parameter	Value
SCIP Status	problem is solved [infeasible]
Solving Time (s)	0.00
Solving Nodes	0
Primal Bound	$+1.0000000000000000 \times 10^{20}$ (0 solutions)
Dual Bound	$+1.0000000000000000 \times 10^{20}$
Gap	0.00%
Objective value	infeasible

## 4. Results and Discussion

### 4.1. Simulation Setup and Results

Simulations are conducted to evaluate the performance of our method, which is based on the SCIP solver under various AUV fleets and different numbers of task nodes.

The multiple AUVs system studied in this paper is deployed underwater, where AUVs are typically assigned to different depths. Given this predefined depth allocation, the likelihood of AUVs occupying the same physical space is minimized. Therefore, the issue of potential collisions between AUVs is not considered in this study.

The AUVs have different power capacities, and the task nodes require specific amounts of energy for execution. We simulated six scenarios, as illustrated in Figure 3. From the results of the task allocation, Figure 3a—three AUVs and eleven task nodes, Figure 3b—four AUVs and eight task nodes, Figure 3c—four AUVs and ten task nodes, Figure 3e—eight AUVs and sixteen task nodes, and Figure 3f—eight AUVs and twenty task nodes achieved optimal task allocation within 10 s. However, Figure 3d—four AUVs and twenty task nodes did not yield a feasible solution as it failed to meet the energy requirements of all the task nodes, suggesting the need to increase the number of AUVs.



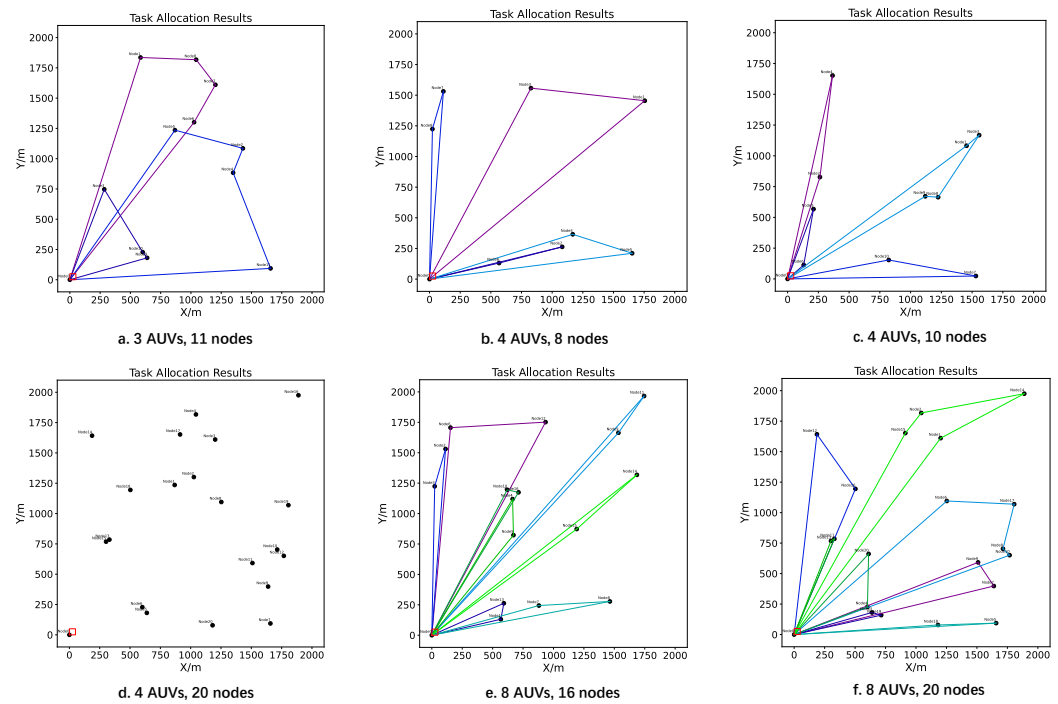


Figure 3. Different AUVs and different task nodes’ allocation results.

#### 4.2. Comparative Analysis of Our Algorithm and PSO Algorithm

##### 4.2.1. Introduction to PSO Algorithm

The Particle Swarm Optimization (PSO) algorithm is a heuristic optimization method widely used for solving complex combinatorial optimization problems such as the Capacity-Constrained Vehicle Routing Problem (CVRP). While the CVRP is classified as an NP-hard problem, PSO emulates the behavior of bird flocks searching for food to iteratively optimize solutions. This makes it suitable for large-scale or complex problems.

##### 4.2.2. Performance Comparison

To compare the performance of both algorithms, we conducted several experiments across various problem instances. The results are summarized in Table 3, where the objective values obtained by both the SCIP solver and the PSO algorithm are compared.

Table 3. Comparison of objective values between SCIP solver and PSO.

Experiment No.	a	b	c	d	e	f
SCIP Solver (s)	10,040.6	11,220.9	9680.5	0	22,480.7	21,783.9
PSO (s)	12,007.5	11,258.8	10,308.7	inf	24,776.6	26,993.1

From the results in Table 3, it is evident that, while PSO can provide a feasible solution, the SCIP solver consistently finds the optimal solution. For example, in the a, e, and f experiment, the SCIP solver outperforms the PSO algorithm by a significant margin in terms of the objective values. This indicates that the SCIP solver is more effective in terms of achieving the globally optimal solution.

The corresponding visualization of the objective values is shown in the figure below (Figure 4):

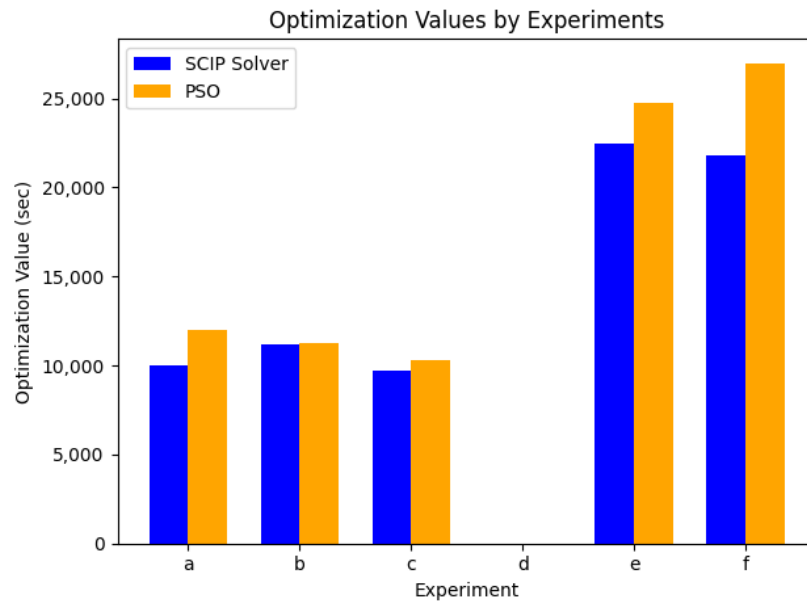


Figure 4. Objective value comparison between our algorithm and PSO.

#### 4.2.3. Computational Time Comparison

The following Figure 5 presents a comparison of the computational time for both algorithms across the same set of problem instances. As shown in Figure 5, PSO has an advantage in terms of the computational time, especially for smaller problem instances. Importantly, for most of the problem instances, the SCIP-solver-based algorithm solves the problems in less than 10 s, demonstrating that it is practical for real-world applications. In half of the instances, the SCIP solver completes the task in less than 1 s, further emphasizing its computational efficiency.

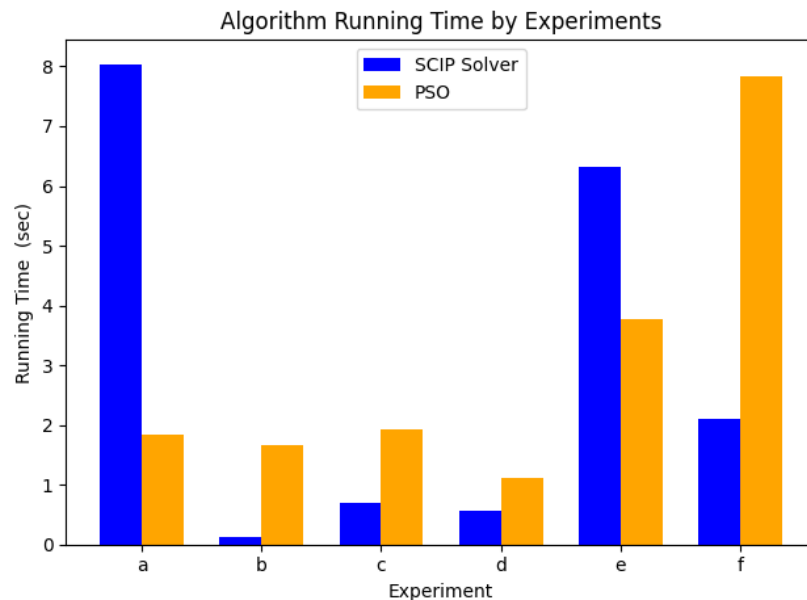


Figure 5. Algorithm running time comparison between our algorithm and PSO.

#### 4.2.4. Machine Configuration

The experiments were conducted on a machine with the following configuration in Table 4:

**Table 4.** Machine configuration.

Parameter	Value
Processor	Core i7-10750H @ 2.60 GHz
Memory	16 GB RAM
Operating System	Windows 10 64-bit
Solver Version	8.0.4

This configuration was sufficient to efficiently handle both the SCIP solver and PSO algorithm for the problem instances considered in this study. The computation times reported are consistent with the expectations for small- and medium-scale problems, making our method applicable in practical scenarios.

#### 4.2.5. Summary

Through this comparative analysis, it is evident that, while the PSO algorithm can generate feasible solutions quickly, our method based on the SCIP solver consistently finds the globally optimal solution in reasonable time frames, making it a more robust choice for complex optimization problems like the CVRP. Furthermore, the computational time for our algorithm is competitive, especially when considering larger problem instances, where the solver can provide solutions efficiently. The combination of optimality guarantees and practical computational times makes our algorithm highly applicable for real-world problems.

While the proposed methodology based on the SCIP solver demonstrates clear advantages in finding globally optimal solutions for complex optimization problems like the CVRP, it has certain limitations. First, the approach may struggle with scalability for large problem instances. Although the computational time is competitive for larger cases compared to PSO, the solver's performance could degrade when handling significantly larger or more complex datasets, potentially leading to longer processing times or increased resource consumption. Additionally, the solver's requirements for precise input and constraints might make it less flexible in real-time dynamic environments where the problem parameters can change frequently, requiring constant recalibration or re-optimization. Furthermore, the current model lacks the inclusion of key real-world constraints relevant to underwater applications. Critical factors such as 3D problem treatment, inertial effects when changing course, and marine currents are not yet accounted for.

#### 4.3. Practical Implications

The diversity in the battery power capacities among different AUVs and the specific energy requirements of the task nodes significantly impact task allocation. Our simulations, which encompass eight distinct scenarios, provide valuable insights into the practical challenges and solutions in multiple AUVs systems.

The findings indicate that, in the scenarios represented by Figure 3a–c,e,f, the AUVs successfully achieved optimal task allocations. These scenarios demonstrate the feasibility of effectively allocating tasks among AUVs with varying energy capacities, ensuring that the mission objectives are met within the given constraints.

However, the scenario depicted in Figure 3d highlights a critical limitation. The failure to obtain a feasible solution in this scenario underscores the importance of adequate energy provisioning for all the task nodes. This infeasibility suggests that the current fleet size and battery capacities were insufficient to meet the energy demands of the task nodes. Therefore, one practical implication is the necessity of increasing the number of AUVs or enhancing their energy capacities to ensure successful mission completion in similar contexts.

These insights are crucial for the design and operation of multiple AUVs systems, emphasizing the need for the careful consideration of the energy management and fleet composition to achieve efficient and effective task allocation.

## 5. Conclusions and Future Work

In this study, we propose a novel task allocation model for multiple AUVs based on the CVRP, optimizing the navigation time under battery constraints and offering both theoretical and practical insights into task allocation and energy management in energy-constrained underwater environments.

To address these limitations, future work could explore hybrid approaches that combine the strengths of the SCIP solver with heuristic or metaheuristic methods like PSO. Such a combination could maintain the solver's optimality guarantees for smaller and mid-sized instances while using heuristic methods for quicker solutions in larger-scale or time-sensitive scenarios. Additionally, implementing parallelization techniques or utilizing distributed computing could further enhance the scalability of the proposed method, allowing it to handle larger datasets more efficiently. Incorporating mechanisms for real-time adaptability would make the algorithm more applicable to dynamic real-world environments, where the problem constraints and parameters evolve during the optimization process.

While the centralized approach proved to be effective in controlled simulation environments, practical applications necessitate a shift towards distributed strategies. The real-world deployment of multiple AUVs systems involves dynamic and uncertain environments where centralized control may not be feasible or efficient. The future work will focus on developing distributed task allocation algorithms that can operate effectively under weak communication conditions. These algorithms will leverage local decision-making capabilities and robust coordination mechanisms to ensure resilience and adaptability in diverse operational scenarios.

By transitioning to distributed approaches, we aim to enhance the scalability and reliability of multiple AUVs systems, ultimately facilitating their deployment in real-world applications where robust performance under varying communication conditions is critical.

**Author Contributions:** Conceptualization, H.W. and Y.L.; methodology, H.W.; software, H.W.; validation, H.W., Y.L. and G.X.; formal analysis, H.W.; investigation, H.W.; resources, H.W.; data curation, H.W.; writing—original draft preparation, H.W.; writing—review and editing, H.W.; visualization, H.W.; supervision, Y.L. and S.L.; project administration, S.L.; funding acquisition, G.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by National Natural Science Foundation of China (42176194).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

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

## Abbreviations

The following abbreviations are used in this manuscript:

AUV	Autonomous Underwater Vehicle
CVRP	Capacitated Vehicle Routing Problem
SCIP	Solving Constraint Integer Programs
MRTA	Multiple Robots Task Allocation
MILP	Mixed-Integer Linear Programs
CIP	Constraint Integer Programming
PSO	Particle Swarm Optimization

## References

1. Zhou, Z.; Liu, J.; Yu, J. A Survey of Underwater Multi-Robot Systems. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 1–18. [[CrossRef](#)]
2. Chakraa, H.; Guérin, F.; Leclercq, E.; Lefebvre, D. Optimization techniques for Multi-Robot Task Allocation problems: Review on the state-of-the-art. *Robot. Auton. Syst.* **2023**, *168*, 104492. [[CrossRef](#)]
3. Pang, T.; Song, Y.; Zhu, D. Task allocation for multi-AUV system under ocean current environment. In Proceedings of the 2021 China Automation Congress (CAC), Beijing, China, 22–24 October 2021; pp. 4844–4849. [[CrossRef](#)]
4. Lu, Q.; Pan, C.-S.; Ding, Y.; Hu, R.-X. Task Allocation of Multi-AUV Based on Multi-objective Discrete Particle Swarm Optimization and Simulated Annealing. In Proceedings of the 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 565–571. [[CrossRef](#)]
5. Wang, C.; Zhu, S.; Li, B.; Song, L.; Guan, X. Time-Varying Constraint-Driven Optimal Task Execution for Multiple Autonomous Underwater Vehicles. *IEEE Robot. Autom. Lett.* **2023**, *8*, 712–719. [[CrossRef](#)]
6. Ding, C.; Zheng, Z. A Reinforcement Learning Approach Based on Automatic Policy Amendment for Multi-AUV Task Allocation in Ocean Current. *Drones* **2022**, *6*, 141. [[CrossRef](#)]
7. Chen, M.; Zhu, D. A Workload Balanced Algorithm for Task Assignment and Path Planning of Inhomogeneous Autonomous Underwater Vehicle System. *IEEE Trans. Cogn. Dev. Syst.* **2019**, *11*, 483–493. [[CrossRef](#)]
8. Qu, Y.; Zhu, D.; Chen, M. Optimized SOM Algorithm to Solve Problem of Invalid Task Allocation. In Proceedings of the International Conference on Intelligent Robotics and Applications, Shenyang, China, 8–11 August 2019; pp. 218–224. [[CrossRef](#)]
9. Bychkov, I.; Kenzin, M.; Maksimkin, N. Two-Level Evolutionary Approach to Persistent Surveillance for Multiple Underwater Vehicles with Energy Constraints. *Inform. Autom.* **2019**, *18*, 267–301. [[CrossRef](#)]
10. Sarkar, C.; Paul, H.; Pal, A. A Scalable Multi-Robot Task Allocation Algorithm. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 1–9. [[CrossRef](#)]
11. Wei, C.; Hindriks, K.V.; Jonker, C.M. Dynamic task allocation for multi-robot search and retrieval tasks. *Appl. Intell.* **2016**, *45*, 383–401. [[CrossRef](#)]
12. Gerkey, B.P.; Mataric, M.J. A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. J. Robot. Res.* **2004**, *23*, 939–954. [[CrossRef](#)]
13. Wang, C.Q.; Mei, D.; Wang, Y.; Yu, X.W.; Sun, W.; Chen, J.Q. Task allocation for Multi-AUV system: A review. *Ocean Eng.* **2022**, *266*, 112911. [[CrossRef](#)]
14. Miloradovic, B.; Curuklu, B.; Ekstrom, M.; Papadopoulos, A.V. Exploiting Parallelism in Multi-Task Robot Allocation Problems. In Proceedings of the IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Santa Maria da Feira, Portugal, 28–29 April 2021; pp. 197–202. [[CrossRef](#)]
15. Raja, S.; Habibi, G.; How, J.P. Communication-Aware Consensus-Based Decentralized Task Allocation in Communication Constrained Environments. *IEEE Access* **2022**, *10*, 19753–19767. [[CrossRef](#)]
16. Booth, K.E.C.; Nejat, G.; Beck, J.C. A Constraint Programming Approach to Multi-Robot Task Allocation and Scheduling in Retirement Homes. In Proceedings of the 22nd International Conference on the Principles and Practice of Constraint Programming (CP), CHAM, Toulouse, France, 5–9 September 2016; Lecture Notes in Computer Science; Volume 9892, pp. 539–555. [[CrossRef](#)]
17. Aziz, H.; Pal, A.; Pourmiri, A.; Ramezani, F.; Sims, B. Task Allocation Using a Team of Robots. *Curr. Robot. Rep.* **2022**, *3*, 227–238. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.