Combining Meta-Heuristics and K-Means++ for Solving Unmanned Surface Vessels Task Assignment and Path Planning Problems

Weiyu Tang, Kaizhou Gao, Minglong Gao and Zhenfang Ma

November 2, 2023

Combining Meta-heuristics and K-Means++ for Solving Unmanned Surface Vessels Task Assignment and Path Planning Problems

Weiyu Tang
Macau Institute of System Engineering
Macau University of Science and Technology
Taipa, Macao
tangweiyu2000@126.com

Kaizhou Gao*
Macau Institute of System Engineering
Macau University of Science and Technology
Taipa, Macao
kzgao@must.edu.mo

Minglong Gao
Macau Institute of System Engineering
Macau University of Science and Technology
Taipa, Macao
gaominglong525@foxmail.com

Zhenfang Ma
Macau Institute of System Engineering
Macau University of Science and Technology
Taipa, Macao
mazhenfang@live.com

***Abstract*—This study addresses Unmanned Surface Vessels (USVs) task assignment and path planning problems with minimizing the maximum completion time of USVs. First, a mathematical model is developed for the concerned problems. Second, an unsupervised learning algorithm, K-Means++, is employed to assign multi-tasks to USVs. According to the assignment results, five meta-heuristics are used to solve path planning problems for USVs. Finally, experiments are executed to solve 10 cases with different scales. The effectiveness of K-Means++ for task assignment is verified. The results of five meta-heuristics for path planning are reported and analyzed. The harmony search algorithm has the strongest competitiveness among all compared algorithms for solving the concerned problems.**

Keywords*—*Unmanned surface vessel, Task assignment, Path planning, Meta-heuristics, K-Means++

I. Introduction

In recent years, with the development of autonomous control technology, more and more scholars have focused on the research related to Unmanned Surface Vessels (USVs). USV is an intelligent vessel that can autonomously accomplish multiple tasks in complex scenarios, and is widely concerned in scientific research, environmental tasks, surface search and rescue, marine survey and other fields [1]. With the popularity of multi-USV systems, for the convenience of research, it is divided into two sub-problems: multi-task assignment and path planning.

There are many methods that have been proven to be effective in solving task assignment problems, such as exact algorithms [2], meta-heuristics [3], etc. Unsupervised learning algorithms [4] as machine learning methods also have some advantages in solving task assignment problems. Unsupervised learning algorithms learn patterns from unlabeled data. The two main methods used in unsupervised learning are principal component analysis and cluster analysis. Cluster analysis is a branch of machine learning that divides similar objects into different groups or more subsets by static classification. The objects in the same subset have some similar properties [5].

Path planning problems are generally classified as global path planning and local path planning. Among the solution methods, the exact algorithms can obtain accurate results but consumes high computation cost. The A* algorithm can obtain the global optimal path in a limited search time for small-scale problems [6]. Q-Learning and DQN algorithms based on reinforcement learning can also be used for intelligent vessel path planning [7]. Solving global path planning problems are taken as traveling salesman problems, which is a typical NP-hard problem [8]. Meta-heuristics [9]-[14] have some advantages for solving such problems and can obtain approximate optimal solutions for path planning problems.

In this study, a combination of K-Means++ and meta-heuristics is proposed to solve the task assignment and path planning problems of multiple USVs. K-Means++ is used to assign tasks for USVs, while five meta-heuristics are improved to solve the multi-USV path planning problems. The experimental results show that K-Means++ combined with the meta-heuristics have better effects and advantages in solving the task assignment and path planning problems of USVs.

II. Problem Description

When multiple USVs are performing tasks, they are required to reach multiple task coordinate points. The task completion time includes exploration time and travel time. The task assignment among USVs and the execution sequence of tasks on each USV are what we need to study. The framework of the task assignment and path planning system for multiple USVs is shown in Fig. 1.

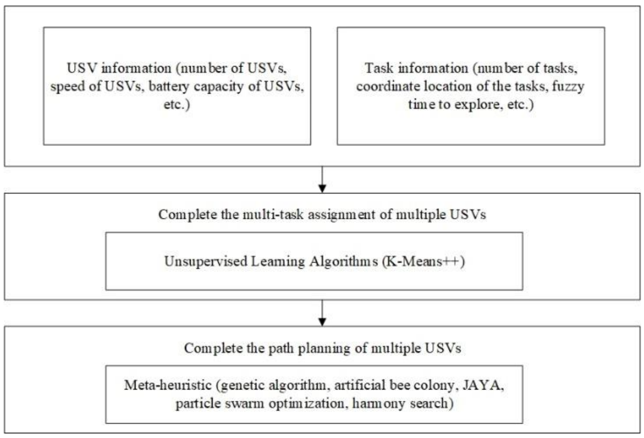

Fig. 1. Framework of task assignment and path planning for multiple USVs.

In the multi-USV task assignment and path planning problems, n tasks are assigned to $m(m \geq 2)$ USVs. we define the problems as an undirected graph, $G = \{V, E\}$, where the set of vertices is denoted as $V = \{0,1,2, \dots, n\}$ and the set of edges is denoted as $E = \{(i,j)|i,j \in V, i \neq j\}$. Vertex 0 represents the departure point and the other vertices represent the task points. The set $K = \{1,2,\dots,m\}$ represents the route of USVs. The coordinate of the departure point is set to $(0,0)$, and when a USV travels from task i to task j, the traveling distance d_{ij} and traveling time t_{ij} are respectively calculated as follows:

$$d_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{1}$$

$$t_{ij} = \frac{d_{ij}}{v} \tag{2}$$

In this study, the exploration time is considered as a triangular fuzzy number. For a triangular fuzzy number, the addition and ranking are the most important operations to calculate the USVs' completion time. In the concerned problems, the addition operation is used to compute the fuzzy exploration time while the ranking operation is used to compare the completion time and obtain the maximum fuzzy completion time.

For two exploration times $\bar{t}_i = (\bar{t}_{i1}, \bar{t}_{i2}, \bar{t}_{i3})$ and $\bar{t}_j = (\bar{t}_{j1}, \bar{t}_{j2}, \bar{t}_{j3})$, the addition operation is shown as follows.

$$\bar{t}_i + \bar{t}_j = (\bar{t}_{i1} + \bar{t}_{j1}, \bar{t}_{i2} + \bar{t}_{j2}, \bar{t}_{i3} + \bar{t}_{j3}) \tag{3}$$

In the ranking operation, in order to compare two exploration times $\bar{t}_i = (\bar{t}_{i1}, \bar{t}_{i2}, \bar{t}_{i3})$ and $\bar{t}_j = (\bar{t}_{j1}, \bar{t}_{j2}, \bar{t}_{j3})$, three ranking criteria for the fuzzy exploration time are set as follows.

$$If\ \ \frac{(\bar{t}_{i1} + 2\bar{t}_{i2} + \bar{t}_{i3})}{4} > (<) \frac{\left(\bar{t}_{j1} + 2\bar{t}_{j2} + \bar{t}_{j3}\right)}{4}, then\ \ \bar{t}_i > (<) \bar{t}_j \tag{4}$$

$$If\ \ \frac{(\bar{t}_{i1} + 2\bar{t}_{i2} + \bar{t}_{i3})}{4} = \frac{\left(\bar{t}_{j1} + 2\bar{t}_{j2} + \bar{t}_{j3}\right)}{4}\ and\ \ \bar{t}_{i2} > (<) \bar{t}_{j2},\ \ then\ \ \bar{t}_i > (<) \bar{t}_j \tag{5}$$

$$If\ \ \bar{t}_{i2} = \bar{t}_{j2}\ and\ \bar{t}_{i3} - \bar{t}_{i1} > (<) \bar{t}_{j3} - \bar{t}_{j1},\ then\ \ \bar{t}_i > (<) \bar{t}_j \tag{6}$$

The arrival time for task j is calculated by Equation (7). When a USV does not have enough residual energy to complete the next task, it needs to return to the departure point for battery replacement. A round trip time is represented by Equation (8). The total time required to travel and explore is shown in Equation (9). The total round-trip time required for USV k is shown in Equation (10).

$$T_j = T_i + \bar{t}_i + t_{ij}, \forall i \in V, j \in V \backslash \{0\}, i \neq j \tag{7}$$

$$t_b = \beta * \frac{\sum_{i=0}^{n} (\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2})}{n * v} \tag{8}$$

$$W_k = \sum_{i=0}^{n} \sum_{j=0}^{n} x_{ijk} \left(t_{ij} + \bar{t}_i\right), \forall k \in K, \forall i, j \in V \tag{9}$$

$$R_k = N_k * 2t_b, \forall k \in K, \forall i, j \in V \tag{10}$$

The objective of this study is to minimize the maximum completion time of USVs. To elaborate on this objective, a USV that takes the longest time to complete its own task is selected and then its completion time is minimized. Thus, the objective function and corresponding constraints for the multi-USV task assignment and path planning problems can be expressed as follows.

$$\min C_{max} = Max(C_1, C_2, \dots, C_k) \tag{11}$$

$$C_k = W_k + R_k, \forall k \in K \tag{12}$$

$$\sum_{k=1}^{m} \sum_{i=0}^{n} x_{ijk} = 1, \forall j \in V \backslash \{0\} \tag{13}$$

$$\sum_{k=1}^{m} \sum_{j=0}^{n} x_{ijk} = 1, \forall i \in V \backslash \{0\} \tag{14}$$

$$\sum_{i=0}^{n} x_{ijk} - \sum_{i=0}^{n} x_{jik} = 0, \forall k \in K, j \in V \backslash \{0\} \tag{15}$$

$$\sum_{i=1}^{n} x_{i0k} = \sum_{j=1}^{n} x_{0jk} = 1, \forall k \in K \tag{16}$$

$$(N_k + 1) * B_k \geq \sum_{i=0}^{n} \sum_{j=0}^{n} x_{ijk} \left(t_{ij} + \bar{t}_i\right) + N_k * 2t_b, \forall k \in K, \forall i, j \in V \tag{17}$$

$$m \geq K \tag{18}$$

$$x_{ijk} \in \{0,1\}, \forall i, j \in V, i \neq j, \forall k \in K \tag{19}$$

$$x_{ijk} = 0, \forall i, j \in V, i = j, \forall k \in K \tag{20}$$

$$\bar{t}_i = 0, i \in \{0\} \tag{21}$$

where C_k is the completion time of USV k. Constraints (13)-(15) indicate that each task must be accessed once by a USV. Constraint (16) denotes that the number of visits to the task area is equal to the number of departures from the task area. Constraint (17) ensures the capacity constraint of a USV. Constraint (18) limits the number of USVs. Constraints (19)-(20) set the definition of decision variables. Constraint (21) means that the required exploration time for the departure point is 0.

III. Algorithm Design

A. Solution Representation

In the task assignment sub-problems, we use a clustering algorithm, K-Means++, to assign tasks to USVs based on the information related to USVs and tasks.

In the path planning of USVs, we encode one solution as a vector, denoted as $L = (0, S_1, 0, S_2, 0, \dots, 0, S_k)$, where S_k denotes the task list of USV k and 0 is used to separate the tasks between two USVs. The task sequence of USV k is denoted as $S_k = (S_k(1), S_k(2), \dots, S_k(q))$, where q is the number of tasks assigned to USV k. To describe the solution representation method more clearly, Fig. 2 shows a simple example. We assume that there are two USVs performing 10 tasks. The assignment of 10 tasks on two USVs is denoted as $S_1 = (2,4,5,7)$, $S_2 = (1,3,6,8,9,10)$, respectively. The corresponding solution can be denoted as $L = (0,2,4,5,7,0,1,3,6,8,9,10)$.

	S_1				S_2					
Solution	2	4	5	7	1	3	6	8	9	10

Representation	0	2	4	5	7	0	1	3	6	8	9	10

Fig. 2. Illustration of the solution.

B. Meta-heuristics

As population-based meta-heuristics, swarm and evolutionary algorithms [15]-[20] use mechanisms inspired by biological evolution and swarm behaviors, such as crossover, mutation, selection, flying and predation. The candidate solutions to the optimization problems play the role of individuals in the population of swarm and evolutionary algorithms, and the fitness function determines the quality of the solutions. After repeated execution of the above operators, the population evolves. Swarm and evolutionary algorithms aim to find, generate, adapt, or select heuristics that can provide sufficiently good solutions to an optimization problem. Especially, swarm and evolutionary algorithms are usually employed to solve the problems, where the problem-specific information is incomplete or imperfect, and computational power is limited. The basic flow chart of the swarm and evolutionary algorithms is shown in Fig.3.

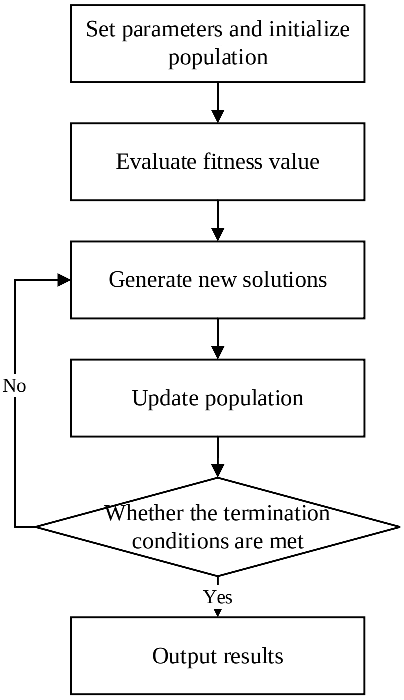

Fig. 3. The basic flow chart of swarm and evolutionary algorithms.

We use five swarm and evolutionary algorithms to solve the USVs path planning problems, namely genetic algorithm (GA), artificial bee colony (ABC), JAYA, particle swarm optimization (PSO), harmony search (HS).

GA [21] is based on the evolutionary laws of organisms in nature and is commonly used to generate high-quality solutions to optimization and scheduling problems, relying on biologically inspired operators including selection, crossover and mutation. ABC [22], [23] is an optimization algorithm based on the intelligent foraging behavior of bee colonies. In ABC, the position of a food source represents a possible solution to an optimization problem, the amount of nectar from a food source corresponds to the quality (fitness) of the associated solution, and each food source is utilized by only one employed bee. JAYA [24], [25] has a very simple structure and is few of meta-heuristics without algorithm-specific parameters. Many its variants have been proposed to solve different types of optimization problems. PSO [26] is a swarm intelligence-based meta-heuristic that can efficiently and globally optimize a problem with a large solution space and find candidate solutions without knowing much information about the problem. Inspired by the musical performance process, HS [27] consists of three operators: random search, harmonic memory consideration rule and pitch adjustment rule. The ways of handling exploration and exploitation with the three operators make the HS as a unique meta-heuristic.

C. K-Means++ algorithm

K-Means++ [28] is an algorithm for selecting initial values for the K-Means clustering algorithm. A disadvantage of the K-Means algorithm is that it is sensitive to the initialization of the prime or mean points, while the K-Means++ algorithm ensures a smarter initialization of the prime and improves the quality of the clusters. Algorithm 1 shows the basic steps of the K-Means++ algorithm. Since the data set is small, the data dimensionality is not high, and the location selection of the initialized centroid is more important. This study mainly uses K-Means++ algorithm to complete the clustering for USVs' task assignment.

Algorithm 1 K-Means++ algorithm

Input: X (n datapoints), k (number of centers)

Output: C (set of initial centers)

```
Sample a point c_1 independently and uniformly at random from X.
Let C = {c_1}
for i = 2 to k do
    for x ∈ X do
        p(x) = min_{c∈C} ||x−c||^2 / Σ_{y∈X} min_{c∈C} ||y−c||^2
    end for
    Sample a point c_i from X, where every x ∈ X has probability p(x).
    Update C = C ∪ {c_i}.
end for
Run Lloyd's algorithm initialized with center set C and output the result.
```

$$p(x) = \frac{\min_{c\in C}\|x-c\|^2}{\sum_{y\in X}\min_{c\in C}\|y-c\|^2}$$

D. Ensemble meta-heuristics and K-Means++

Based on the K-Means++ algorithm, it is possible to obtain the USVs' task assignment results, which will be used in combination with the meta-heuristics to solve path planning problems for multiple USVs. The flow chart for the ensemble meta-heuristics and K-Means++ is shown in Fig. 4.

In Fig. 4, the dashed boxed section on the left represents the process of completing the clustering using the K-Means++ algorithm and deriving the results of the task assignment. Based on the results of the task assignment, the operations in the right dashed box are performed. It represents the process of solving for multi-USV path planning problems using meta-heuristics.

IV. EXPERIMENTS AND DISCUSSION

A. Experimental Setup

The model for this study is involved in a collaboration with an USV company in China, and the data is obtained from the company's simulation platform. In this section, 10 cases with different scales are solved, each case consists of information about the USVs and tasks. The USV information includes the number of USVs, speeds and battery capacity limitation, while the task information includes the number of tasks, the location of the task coordinates and the fuzzy time of exploration. The case scales are 2×20, 2×40, 2×80, 4×20, 4×40, 4×80, 8×20, 8×40, 8×80, 8×120, where 2, 4 and 8 represent the number of USVs, and 20, 40, 80 and 120 represent the number of tasks, respectively.

In the experiments, all algorithms are coded in Python, and the experimental design is implemented on a computer with Intel Core i7-10750H @2.60 GHz, 16 GB memory, and Windows 10 operating system.

In the clustering algorithm selection experiments, all the algorithms are repeated ten times to ensure the stability of the algorithm when performing task assignment. In the path planning experiment, all meta-heuristics are also repeated 10 times, and the population size is set to 10 with the same computation time to ensure the fairness of the comparisons.

B. Comparison of Clustering Algorithms

First, we need to complete the task assignment using unsupervised learning algorithms. Suitable clustering algorithms help to obtain better quality scheduling results in path planning. Therefore, based on the algorithmic advantages, this study chooses the K-Means++ algorithm to complete the task assignment and compare it with five clustering algorithms, including K-Means, Mini Batch K-Means, Hierarchical Agglomerative Clustering (HAC), Spectral Clustering (SC), and Gaussian Mixed Model (GMM).

In this experiment, the largest case with the 8 USVs and 120 tasks is tested. After clustering the tasks and USVs several times, the maximum completion time of all USVs is obtained by using meta-heuristics with the same termination condition. Finally, the mean values and coefficient of variation (CV) are calculated as shown in TABLE I. The calculation method of the CV is as follows.

$$CV = \frac{\sigma}{\mu} \times 100\% \tag{22}$$

where σ is the standard deviation of 10 repeated experiments and μ is the mean values.

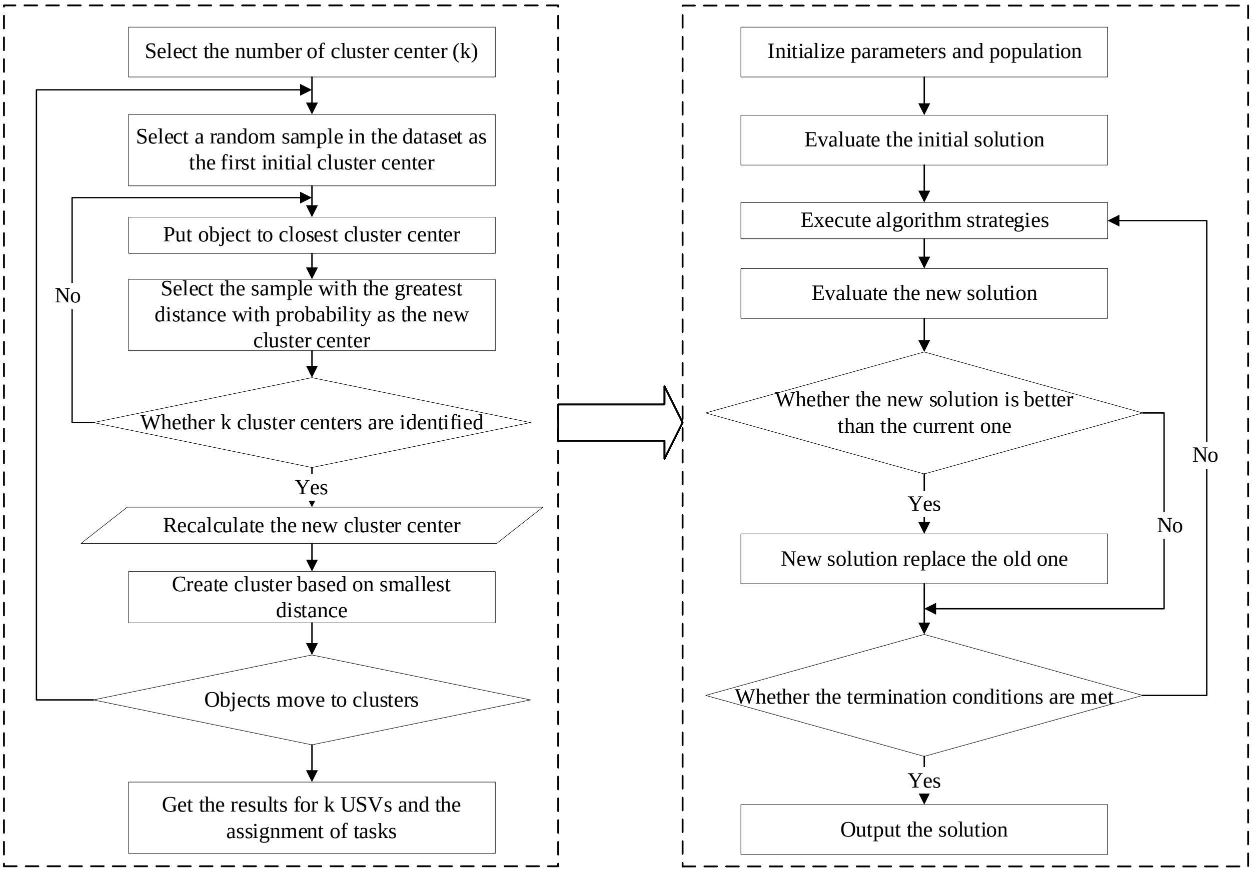

Fig. 4. The flow chart for the ensemble of meta-heuristics and K-Means++.

TABLE I. COMPARISON OF MEAN AND CV OF CLUSTERING ALGORITHMS COMBINED WITH META-HEURISTICS

Algorithm	K-Means++		K-Means		Mini Batch K-Means		HAC		SC		GMM	
	Mean (s)	CV (%)	Mean (s)	CV (%)	Mean (s)	CV (%)	Mean (s)	CV (%)	Mean (s)	CV (%)	Mean (s)	CV (%)
GA	**1055.6**	0.22	1259.4	0.13	1346.3	0.16	1488.4	0.17	1185.2	0.21	2078.8	0.24
ABC	**1057.0**	0.19	1261	0.17	1348.9	0.24	1486.4	0.43	1184.6	0.32	2076.1	0.27
JAYA	**1052.2**	0.21	1254.2	0.20	1339.5	0.26	1478.8	0.28	1177	0.52	2057.6	0.22
PSO	**1064.6**	0.48	1269.8	0.29	1362.3	0.52	1503.6	0.26	1204.7	0.68	2096.2	0.48
HS	**1051.4**	0.18	1254.5	0.22	1339.5	0.23	1473.3	0.44	1174.6	0.25	2059.1	0.35

It can be seen that K-Means++ combined with all meta-heuristics has achieved the best results and the CV of all algorithms are similar. Therefore, K-Means++ is chosen as the algorithm for task assignment in the following experiments.

C. Comparisons of Meta-heuristics with K-means++

We solve the maximum completion time for all cases using five meta-heuristics based on the task assignment results by K-Means++ and perform 10 repeated experiments. Finally, the mean values and CV are calculated, which are shown in TABLE II.

TABLE II. COMPARISON OF MEAN AND CV OF ALL INSTANCES IN THE K-MEANS++ ALGORITHM COMBINED WITH THE FIVE META-HEURISTICS

Instance	GA + K-Means++		ABC + K-Means++		JAYA + K-Means++		PSO + K-Means++		HS + K-Means++	
	Mean (s)	CV (%)	Mean (s)	CV (%)	Mean (s)	CV (%)	Mean (s)	CV (%)	Mean (s)	CV (%)
2×20	841.0	1.30	851.2	1.38	839.5	2.28	937.5	5.02	**828.0**	1.42
2×40	2414.1	1.68	2377.4	2.48	**2267.6**	1.62	2520.9	3.62	2294.2	2.14
2×80	5350.3	1.04	5373.6	0.60	**5158.0**	1.48	5475.1	3.90	5192.3	1.24
4×20	**352.9**	0.00	353.0	0.05	355.4	1.12	359.6	0.67	353.0	0.09
4×40	913.2	0.43	916.2	0.62	908.8	0.80	938.8	1.23	**886.9**	1.92
4×80	1504.3	1.18	1521.2	0.71	**1470.0**	0.53	1569.6	1.39	1476.3	1.21
8×20	137.3	0.65	137.9	0.64	139.6	1.79	142.5	1.61	**136.1**	0.30
8×40	875.9	0.41	881.3	0.36	875.7	0.61	894.3	0.67	**867.9**	0.43
8×80	891.4	0.14	891.0	0.33	882.8	0.34	897.7	0.62	**882.0**	0.31
8×120	1055.6	0.07	1056.1	0.18	**1051.7**	0.23	1063.2	0.36	1052.0	0.25
Average		0.69		0.73		1.08		1.91		0.93

It can be seen from TABLE II that the JAYA with K-Means++ and the HS with K-Means++ obtain the best results in most cases, while the results of the GA and ABC algorithms with K-Means++ are relatively general. The results by PSO with K-Means++ are poor. Next, we perform a Friedman test on the average values of all meta-heuristics to test significant differences among these algorithms. The results are shown in TABLE III. The significance value (Asymp.Sig) obtained by the Friedman test is far less than the setting significance level (0.05). Therefore, there is a significant difference in the competitiveness of the five meta-heuristics with K-Means++.

TABLE III. THE STATISTICAL RESULTS OF THE FRIEDMAN TEST

Test Statistics	
N	10
Chi-Square	29.809
df	4
Asymp.Sig	0.000

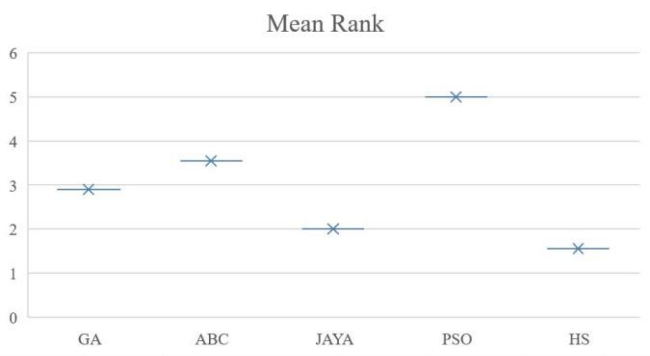

Fig. 5. The Nemenyi post-hoc test ranking.

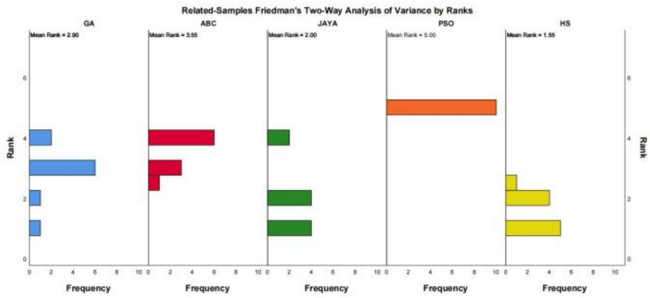

Fig. 6. Friedman's two-way analysis of variance by ranks.

The five algorithms are ranked across the 10 cases as shown in the Fig. 5. The meta-heuristics with smaller ranking values have better competitiveness. Among the five meta-heuristics, the HS with K-Means++ outperforms the others with the minimum mean rank value (1.55). JAYA follows closely (2.00). GA (2.90) has a similar performance to ABC (3.55), and PSO has the worst mean rank value (5.00).

Fig. 6 presents Friedman's two-way analysis of variance by ranks. As shown in Fig. 6, the HS with K-Means++ has the best performance on 5 out of 10 cases and does not obtain the worst results for any instance. The JAYA with K-Means++ ranks second, GA and ABC have similar competitiveness and are better than PSO.

V. CONCLUSION AND FUTURE WORKS

This study presents an unsupervised learning algorithm, K-Means++, combined with meta-heuristics to solve the task assignment and path planning problems for multiple USVs. K-Means++ is used to cluster USVs and tasks, and then the meta-heuristics are used to optimize the completion time of USVs according to the clustering results. Comparative experiments on clustering effects are designed and the performance of K-Means++ is verified. Further experiments are conducted on 10 cases with different scales, yielding the best results for HS with K-Means++ among the five meta-heuristics. Based on the results of several experiments, it is demonstrated that the K-Means++ combined with meta-heuristics has strong performance and advantages in solving the USVs scheduling problems.

In the future, we will consider more real-life constraints for the concerned problems, and employ reinforcement learning algorithms, such as the Q-Learning, to further improve the performance of the meta-heuristics. In addition, this research can be generalized to be applied to task assignment and path planning for unmanned devices such as unmanned vehicles, e.g., for loading and unloading of goods by unmanned vehicles. The research ideas in this study can also be applied to various multiple traveling salesman problems.

ACKNOWLEDGMENTS

This study is partially supported by the Zhuhai Industry University Research Project with Hongkong and Macao under Grant ZH22017002210014PWC, the Science and Technology Development Fund (FDCT), Macau SAR, under Grant 0019/2021/A, the National Natural Science Foundation of China under Grant 62173356, and the Guangdong Basic and Applied Basic Research Foundation (2023A1515011531).

REFERENCES

[1] W. Zhang, K. Wang, S. Wang, and G. Laporte, 'Clustered coverage orienteering problem of unmanned surface vehicles for water sampling', Naval Research Logistics, vol. 67, no. 5, pp. 353–367, Aug. 2020.

[2] K. Xue, Z. Huang, P. Wang, and Z. Xu, 'An Exact Algorithm for Task Allocation of Multiple Unmanned Surface Vehicles with Minimum Task Time', JMSE, vol. 9, no. 8, p. 907, Aug. 2021.

[3] S. Sun, B. Song, P. Wang, H. Dong, and X. Chen, 'An adaptive bi-level task planning strategy for multi-USVs target visitation', Applied Soft Computing, vol. 115, p. 108086, Jan. 2022.

[4] S. Ma, W. Guo, R. Song, and Y. Liu, 'Unsupervised learning based coordinated multi-task allocation for unmanned surface vehicles', Neurocomputing, vol. 420, pp. 227–245, Jan. 2021.

[5] A. Sancho, J. C. Ribeiro, M. S. Reis, and F. G. Martins, 'Cluster analysis of crude oils with k-means based on their physicochemical properties', Computers & Chemical Engineering, vol. 157, p. 107633, Jan. 2022.

[6] P. Hart, N. Nilsson, and B. Raphael, 'A Formal Basis for the Heuristic Determination of Minimum Cost Paths', IEEE Trans. Syst. Sci. Cyber., vol. 4, no. 2, pp. 100–107, 1968.

[7] X. Zhou, P. Wu, H. Zhang, W. Guo, and Y. Liu, 'Learn to Navigate: Cooperative Path Planning for Unmanned Surface Vehicles Using Deep Reinforcement Learning', IEEE Access, vol. 7, pp. 165262–165278, 2019.

[8] C. Archetti, D. Feillet, M. Gendreau, and M. Grazia Speranza, 'Complexity of the VRP and SDVRP', Transportation Research Part C: Emerging Technologies, vol. 19, no. 5, pp. 741–750, Aug. 2011, doi: 10.1016/j.trc.2009.12.006.

[9] Y. Pan, K. Gao, Z. Li, and N. Wu, 'Improved Meta-Heuristics for Solving Distributed Lot-Streaming Permutation Flow Shop Scheduling Problems', IEEE Trans. Automat. Sci. Eng., vol. 20, no. 1, pp. 361–371, Jan. 2023.

[10] Y. An, X. Chen, K. Gao, L. Zhang, Y. Li, and Z. Zhao, 'A hybrid multi-objective evolutionary algorithm for solving an adaptive flexible job-shop rescheduling problem with real-time order acceptance and condition-based preventive maintenance', Expert Systems with Applications, vol. 212, p. 118711, Feb. 2023.

[11] Y. Pan, K. Gao, Z. Li, and N. Wu, 'Solving Biobjective Distributed Flow-Shop Scheduling Problems With Lot-Streaming Using an Improved Jaya Algorithm', IEEE Trans. Cybern., pp. 1–11, 2022.

[12] F. Zhao, Z. Wang, L. Wang, T. Xu, N. Zhu, and Jonrinaldi, 'An exploratory landscape analysis driven artificial bee colony algorithm with maximum entropic epistasis', Applied Soft Computing, vol. 137, p. 110139, Apr. 2023.

[13] K. Z. Gao, Z. M. He, Y. Huang, P. Y. Duan, and P. N. Suganthan, 'A survey on meta-heuristics for solving disassembly line balancing, planning and scheduling problems in remanufacturing', Swarm and Evolutionary Computation, vol. 57, p. 100719, Sep. 2020.

[14] K. Li et al., 'Meta-Heuristic Optimization Based Two-stage Residential Load Pattern Clustering Approach Considering Intracluster Compactness and Inter-cluster Separation', IEEE Trans. on Ind. Applicat., pp. 1–1, 2020.

[15] K. Gao, Z. Cao, L. Zhang, Z. Chen, Y. Han, and Q. Pan, 'A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems', IEEE/CAA J. Autom. Sinica, vol. 6, no. 4, pp. 904–916, Jul. 2019.

[16] P. W. Shaikh, M. El-Abd, M. Khanafer, and K. Gao, 'A Review on Swarm Intelligence and Evolutionary Algorithms for Solving the Traffic Signal Control Problem', IEEE Trans. Intell. Transport. Syst., vol. 23, no. 1, pp. 48–63, Jan. 2022.

[17] R. Li, W. Gong, L. Wang, C. Lu, and S. Jiang, 'Two-stage knowledge-driven evolutionary algorithm for distributed green flexible job shop scheduling with type-2 fuzzy processing time', Swarm and Evolutionary Computation, vol. 74, p. 101139, Oct. 2022.

[18] F. Zhao, X. He, and L. Wang, 'A Two-Stage Cooperative Evolutionary Algorithm With Problem-Specific Knowledge for Energy-Efficient Scheduling of No-Wait Flow-Shop Problem', IEEE Trans. Cybern., vol. 51, no. 11, pp. 5291–5303, Nov. 2021.

[19] Q.-K. Pan, L. Gao, and L. Wang, 'An Effective Cooperative Co-Evolutionary Algorithm for Distributed Flowshop Group Scheduling Problems', IEEE Trans. Cybern., vol. 52, no. 7, pp. 5999–6012, Jul. 2022.

[20] Z. Liang, H. Dong, C. Liu, W. Liang, and Z. Zhu, 'Evolutionary Multitasking for Multiobjective Optimization With Subspace Alignment and Adaptive Differential Evolution', IEEE Trans. Cybern., vol. 52, no. 4, pp. 2096–2109, Apr. 2022.

[21] G. Zhang, Y. Hu, J. Sun, and W. Zhang, 'An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints', Swarm and Evolutionary Computation, vol. 54, p. 100664, May 2020.

[22] B. Akay and D. Karaboga, 'A modified Artificial Bee Colony algorithm for real-parameter optimization', Information Sciences, vol. 192, pp. 120–142, Jun. 2012.

[23] H. Li, K. Gao, P.-Y. Duan, J.-Q. Li, and L. Zhang, 'An Improved Artificial Bee Colony Algorithm With Q-Learning for Solving Permutation Flow-Shop Scheduling Problems', IEEE Trans. Syst. Man Cybern, Syst., vol. 53, no. 5, pp. 2684–2693, May 2023.

[24] Y. Zhang, A. Chi, and S. Mirjalili, 'Enhanced Jaya algorithm: A simple but efficient optimization method for constrained engineering design problems', Knowledge-Based Systems, vol. 233, p. 107555, Dec. 2021.

[25] K. Gao, F. Yang, M. Zhou, Q. Pan, and P. N. Suganthan, 'Flexible Job-Shop Rescheduling for New Job Insertion by Using Discrete Jaya Algorithm', IEEE Trans. Cybern., vol. 49, no. 5, pp. 1944–1955, May 2019.

[26] F. Wang, H. Zhang, and A. Zhou, 'A particle swarm optimization algorithm for mixed-variable optimization problems', Swarm and Evolutionary Computation, vol. 60, p. 100808, Feb. 2021.

[27] J. H. Kim, 'Harmony Search Algorithm: A Unique Music-inspired Algorithm', Procedia Engineering, vol. 154, pp. 1401–1405, 2016.

[28] J. Hämäläinen, T. Kärkkäinen, and T. Rossi, 'Improving Scalable K-Means++', Algorithms, vol. 14, no. 1, p. 6, Dec. 2020, doi: 10.3390/a14010006.