Constrained Multi-objective Task Assignment for UUVs using Multiple Ant Colonies System

Zhenzhen Xu\textsuperscript{1,2}, Yiping Li\textsuperscript{1,2}, Xisheng Feng\textsuperscript{1}

\textsuperscript{1}State Key laboratory of Robotics, Shenyang Institute of Automation, CAS
Shenyang, Liaoning Province, China
\textsuperscript{2}Graduate School of the Chinese Academy of Sciences
Beijing 100039, China
{xzz & lyp & fxs}@sia.cn

Abstract
The purpose of this research is to develop an effective task assignment algorithm for multiple Unmanned Underwater Vehicles (UUVs) to reacquaint multiple targets. This algorithm is specifically designed for the underwater environment where vehicles typically have dissimilar starting and ending locations. Besides the objective of minimizing the total distance of multiple vehicles, the objectives of minimizing the total turning angle and the constraint of balancing the targets number visited by each vehicle are also considered. This problem is modeled as a constrained multi-objective MTSP. The different measurement units and order of magnitudes of multiple objectives significantly increase the difficulty to generate an effective solution. The proposed algorithm consists of two phases: task number assignment and task assignment using Multiple Ant Colony System (MACS) which is extended from the classical Ant Colony System (ACS). In the first phase, the target number is assigned to each vehicle. Afterwards, MACS is used to solve constrained multi-objective MTSP, in which multiple ant colonies work separately to optimize dissimilar objectives, the ideal solution is generated according to the result of each colony, and the output is the best solution which has the smallest deviation from the ideal solution in the set of Pareto optimal solutions. The computational results show that the output of the proposed algorithm can satisfy the constrained multi-objective requirement and can be applied to underwater application scenario.

1. Introduction
The multiple UUVs system has been widely used in the oceanography research, seafloor survey, marine archaeology and so on. A typical mission is searching for the presence of targets produced from an initial survey of the area that must now be reacquired and identified. This research introduces a task assignment algorithm for multiple UUVs to reacquaint multiple targets.

The existing research on task assignment problem for reacquisition is limited and immature. The waypoint reacquisition algorithm provides a computationally efficient task assignment method [1]. This approach is accomplished by employing a cluster-first, route-second heuristic technique with no feedback or iterations between the clustering and route building steps. However, it will generally not find the high-quality solutions. It also may leave some waypoints (i.e. targets) unvisited. The planning of underwater vehicle group is modeled as a MTSP and implemented by genetic algorithm [2]. Both the proposed algorithms only consider the objective of minimizing the travel distance and do not take into account the constrained multiple objectives. Due to the maneuverability of UUV, when it changes its heading too much with a velocity forwards, it will produce a curve track which will waste fuel or even can not reach the planned position. Thus the optimality criteria of minimizing the turning angle should be considered. Due to the requirement of loading balancing, the constraint of balancing the targets number visited by each vehicle should also be taken into account.

In this paper, the task assignment problem for multiple UUVs is modeled as a constrained multi-objective MTSP. The MTSP is an NP-hard problem in combinatorial optimization. It is a generalization of the well-known TSP where more than one salesman is allowed to be used in the solution. Although there exists a wide body of the literature for the TSP, the MTSP has not received the same amount of attention. To find a guaranteed optimal solution to the MTSP using exhaustive searches is only feasible for very small number of cities. Many of the proficient MTSP solution techniques are heuristic such as evolutionary algorithm [3], simulated annealing[4], tabu search[5], genetic algorithms[6], neural networks[7] and ant systems[8]. However, all these methods can not deal with multi-objective optimization problem directly. This research will show how the ant colony algorithm can be applied to constrained multi-objective MTSP.

A new Multiple Ant Colonies System (MACS) is presented to solve the constrained multi-objective MTSP. The classical methods to solve multi-objective problem is to combine multiple objectives into one objective by weighting method or constraint method. These methods depend on the subjective bias of decision-maker and can not guarantee Pareto optimality. What is more important is that the multiple objectives often have different measurement units and order of magnitudes, so it is difficult and inappropriate to combine them by weighting method. This paper extends the Ant Colony System (ACS) into Multiple Ant Colonies System (MACS). The purpose is to construct
multiple ant colonies to optimize multiple objectives respectively, and to find the solution closest to the ideal solution from the set of Pareto optimal solutions through the interaction among different ant colonies.

The reminder of this paper is organized as follows. First, the task assignment problem for multiple UUVs is formulated by integer linear programming formulation. In Section 3, we give a detail description of the task assignment algorithm. Finally, Computational results show that the proposed algorithm can find competitive solutions which can satisfy constrained multiple objectives.

2. UUV Task Assignment Problem Statement

The task assignment problem for multiple UUVs can be stated as follows: Given n targets with random locations in a certain area, let there be m vehicles which typically have dissimilar starting and ending locations outside the area. The m vehicles must visit and reacquisition the n targets, and each target must be visited exactly once by only one vehicle. In order to saving energy and time, the total distance of visiting all targets must be minimized. Considering the maneuverability of UUV, the total turning angle also must be minimized to reduce the energy loss and tracking error. In addition, the targets number visited by each vehicle should be average due to the requirement of load balancing.

We propose the following integer linear programming formulation for the constrained multiple-objective MTSP defined above. The distance objective function \( f_1(x) \) and the turning angle objective function \( f_2(x) \) can be described as follows:

\[
\begin{align*}
  f_1(x) &= \sum_{i=1}^{m} \sum_{k=1}^{n} d(T_i^k, T_i^{k+1}) + d(S_i, T_i^1) + d(T_i^n, E_i) \\
  f_2(x) &= \sum_{i=1}^{m} \sum_{k=1}^{n} h(T_i^k, T_i^{k+1}) + h(S_i, T_i^1) + h(T_i^n, E_i)
\end{align*}
\]

Where \( T_i^k \) is the \( k \)-th target visited by vehicle \( i \), \( d(T_i^k, T_i^{k+1}) \) is the distance between \( T_i^k \) and \( T_i^{k+1} \), \( h(T_i^k, T_i^{k+1}) \) is the turning angle from \( T_i^k \) to \( T_i^{k+1} \), \( S_i \) is the starting depot of vehicle \( i \), \( E_i \) is the ending depot of vehicle \( i \), \( n_i \) is the target number assigned to vehicle \( i \).

The task number constraint \( g_i(x) \) is defined as follows:

\[
g_i(x) = \max(n_i) - \min(n_i) \in \{0,1\}
\]

The constrained multi-object task assignment problem for multiple UUVs is formulated as follows:

\[
\begin{align*}
  \min f(x) &= \{f_1(x), f_2(x)\} \\
  \text{subject to } g(x) &= \{g_1(x)\}
\end{align*}
\]

3. UUV Task Assignment Algorithm

The task assignment algorithm is divided into two phases: task number assignment and task assignment using MACS. The first phase assigns the target number for each vehicle to visit. Then the second phase presents an extended ant colony algorithm named MACS to provide optimized result about task assignment which can satisfy constrained multiple objectives.

3.1. Task number assignment

The task number assignment phase is employed to realize the task number constraint. Here, the task number is equal to the number of targets assigned to each vehicle to visit. This phase focuses on assigning the number of task while the task assignment and the optimal ordering of tasks will be performed at the second phase. If the number of targets \( n \) can be divided exactly by the number of vehicles \( m \), the task number \( n_i \) for vehicle \( i (i=1,2,\cdots,m) \) should be \( n/m \). Otherwise, let \( q \) be the quotient and \( r \) be the remainder, then \( n_i \) can be defined mathematically as:

\[
n_i = \begin{cases} 
  q + 1, & 1 \leq i \leq r \\
  q, & r < i \leq m 
\end{cases}
\]

After this phase, the task number for each vehicle to visit is clear and we can see \( \sum_{i=1}^{m} n_i = n \).

3.2. Task assignment using MACS

In this section, a Multiple Ant Colonies System (MACS) algorithm for solving UUV task assignment is proposed. In MACS, multiple ant colonies are constructed to optimize multiple objectives respectively. The number of ants in each colony is the same. We update the set of Pareto optimal solutions after each iteration through the interaction among different ant colonies. At the end of optimization, the optimal solution for each objective can be obtained. We combine the multiple optimal solutions into an ideal solution which is probably not a feasible solution. Then we can find the Pareto optimal solution which is closest to the ideal solution as the output of this algorithm. The ideal solution is defined as \( x^* \):

\[
f_q(x^*) = \min f_q(x^+) = \min f_q(x^{k+1}), k = 1,2,\cdots,a, q = 1,2,\cdots,b
\]

Where \( a \) is the number of ants, \( b \) is the number of objectives (i.e. the number of ant colonies). The deviation between solution \( x \) and the ideal solution is defined as \( u \):

\[
u = \sum_{q=1}^{b} w_q u_q
\]

Where \( w_q \) is the weight of objective \( q \). The goal is to find the solution with the smallest deviation from the ideal solution from the set of Pareto optimal solutions.

At first, all ant colonies search at the same space with respective optimality criteria. Each ant colony runs the steps of solution construction and pheromone updating according
to the theory of ant colony system (ACS) [9]. Due to the characteristic of the UUV task assignment, the ACS in this paper is a little different from the classical ACS in solution construction. The task number constraint should be considered.

As defined above, the ant number is $a$, the vehicle number is $m$, and the target number for each vehicle to visit is $n_i (i = 1, 2, \ldots, m)$ . Initially, we put all ants on the $S_i (i = 1, 2, \ldots, m)$ randomly. Each ant starts from the initial position, and then selects unvisited targets to construct a route. Suppose that an ant starts from $S_i , p \in \{1, 2, \ldots, m\}$ , if the number of targets which the ant has visited is equal to $n_p$ , then the ant returns to $E_p$ . After that, the ant chooses an unvisited $S_i , p \in \{1, 2, \ldots, m\}$ , departs from that starting depot and selected unvisited targets iteratively. The targets which have been visited by an ant are recorded in the ant’s tabu table. A valid solution is constructed by an ant until all the targets, starting depots and ending depots are visited. Thus there are $a$ solutions constructed by $a$ ants.

Every ant selects the next city independently. The rules of moving from target $i$ to target $j$ for ant $k$ can be formulated as follows:

$$j = \begin{cases} \arg \max_{l \in N_i^k} \left\{ \eta_{ij}^{\alpha} \left( \frac{1}{q_{ij}} \right)^{\beta} \right\} & \text{if } q \leq q_0 \\ J_i^k & \text{otherwise} \end{cases} \quad (10)$$

If $q$ is larger than $q_0$ , the probability of moving from target $i$ to target $j$ for ant $k$ can be formulated as follows:

$$p_{ij}^k = \frac{\left( \frac{\tau_{ij}}{\tau_{ij}^\text{best}} \right)^\beta}{\sum_{j \in N_i^k} \left( \frac{\tau_{ij}}{\tau_{ij}^\text{best}} \right)^\beta}, \quad \text{if } j \in N_i^k \quad (11)$$

Where $\tau_{ij}$ is equal to the amount of pheromone on the path between the current target $i$ and possible target $j$ . $\eta_{ij}$ is the heuristic information which is defined as the inverse of the cost (e.g. distance or turning angle) between the two targets. $q$ is a random uniform variable $[0,1]$ and the value $q_0$ is a parameter to determine the relative influence between exploitation and exploration. $\alpha$ and $\beta$ are the parameters whose value determine the relation between pheromone and heuristic information. $N_i^k$ denotes the targets which ant $k$ has not visited.

In order to improve future solutions, the pheromone on the best solution found so far must be updated after all ants have constructed their valid routes. The global pheromone updating is done using the following equation:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho \Delta \tau_{ij}^\text{best}, \quad \forall (i, j) \in T_{\text{best}} \quad (12)$$

Where $\rho (0 < \rho \leq 1)$ is a parameter that controls the speed of global pheromone evaporation. $T_{\text{best}}$ denotes the best solution found so far, and $\Delta \tau_{ij}^\text{best}$ is defined as the inverse of the objective value of $T_{\text{best}}$ . In addition, each ant should execute local pheromone updating as soon as it selects a new target $j$ according to the following equation:

$$\tau_{ij} = (1 - \xi)\tau_{ij} + \xi \tau_{ij}^0 \quad (13)$$

$$\tau_{ij} = 1/nC_{\text{nn}} \quad (14)$$

Where $\xi (0 < \xi < 1)$ represents the speed of local pheromone evaporation. $\tau_{ij}$ is the initial value of pheromone. $C_{\text{nn}}$ is the objective value computed with the Nearest Neighbor Algorithm. The global updating belongs to a positive feedback mechanism, while the local updating can increase the opportunity to choose the targets which have not been explored and avoid falling into the state of stagnation.

When an iteration is finished, the set of Pareto optimal solutions is generated from the solutions of each ant colony according to the theory of Pareto optimality. And the pheromone on the dominated solutions must be decreased as a punishment. The punishment pheromone updating is done using the following equation:

$$\tau_{ij} = (1 - \rho)\tau_{ij} - \rho \Delta \tau_{ij}^\text{bs}, \quad \forall (i, j) \in T_{\text{bs}} \quad (15)$$

$$\Delta \tau_{ij}^\text{bs} = 1/C^\text{bs} \quad (16)$$

Where $T_{\text{bs}}$ denotes the dominated solutions generated at this iteration, and $C^\text{bs}$ represents the objective value of $T_{\text{bs}}$.

After all ant colonies have met the end condition of ACS, each ant colony will obtain the best solution found so far. Thus, the objective function value computed from the best solution of each colony is combined into the objective function value of the ideal solution. As far as the deviation from the ideal solution is concerned, the final output of the best solution is the one in the set of Pareto optimal solutions which has the smallest deviation from the ideal solution.

According to the procedure described above, the MACS can be summarized as follows:

**Step 1:** Read the input data;

**Step 2:** Initialize the parameters and pheromone matrixes of all ant colonies;

**Step 3:** For ant colony $q$ , assign one objective to it, execute the steps of ACS to generate valid routes of $a$ ants;

**Step 3.1** Put all ants on the starting depots of vehicles randomly;

**Step 3.2** Construct route for each ant by using the rules (10) and (11), the target number constraint is considered;

**Step 3.3** Carry out the local pheromone updating by using Eq.13;

**Step 3.4** Return to Step 3.2 until each ant has visited all targets, the starting depots and the ending depots, i.e. each ant constructs a valid route;

**Step 3.5** Compute the objective function values of the routes constructed by all ants, and sort them in increasing order;

**Step 3.6** Select the best route found by this ant colony so far, and carry out the global pheromone updating according to Eq.12;

**Step 3.7** Interact among different ant colonies, combine the solutions of each colonies into the set of Pareto optimal solutions;
Step 3.8 Do the punishment pheromone updating on the new dominated solutions according to Eq.15;
Step 3.9 Repeat Step 3.1-3.8 until the end condition (e.g. the pre-specified number of iterations) is met;
Step 4: Repeat Step 3 until all ant colonies meet the end condition;
Step 5: Generate the final output of MACS;
Step 5.1 Combine the best solutions of all colonies to the ideal solution;
Step 5.2 Compute the deviation $u$ between the ideal solution and the solutions in the set of Pareto optimal solutions, respectively;
Step 5.3 Select the best solution with the smallest $u$ as the output of the algorithm.

4. Computational Experiments

In this section, we discuss the parameter settings for the proposed task assignment algorithm and present computational results. According to the objectives $f_1(x)$ and $f_2(x)$, two ant colonies ($b = 2$) are used. For each ant colony, the parameters are set as follows: $a = n$, $\alpha = 1$, $\beta = 2$, $q_0 = 0.9$, $\rho = \xi = 0.1$. The algorithm was tested in two different scenarios. Scenario $A$ contains 10 targets ($n = 10$) and 2 vehicles ($m = 2$), and scenario $B$ contains 20 targets ($n = 20$) and 3 vehicles ($m = 3$). In both scenarios, the targets positions are generated randomly. The respective starting and ending depots coordinates of the vehicles are $S_1 = (30,0)$, $S_2 = (70,0)$, $E_1 = (45,100)$, $E_2 = (55,100)$ in scenario $A$, and $S_1 = (20,0)$, $S_2 = (50,0)$, $S_3 = (80,0)$, $E_1 = (40,100)$, $E_2 = (50,100)$, $E_3 = (60,100)$ in scenario $B$. We set $w_1 = 0.7$, $w_2 = 0.3$ and the pre-specified number of iterations of each ant colony is equal to 200. The results of MACS applied to the two scenarios with four different targets layout are illustrated in Figure 1 and Figure 2. As we have seen, each vehicle obtains a valid route, and the task number constraint is satisfied.

The proposed MACS as well as ACS only considering distance objective (denoted by ACS$^D$) were implemented in order to compare their performances. The coordinates of targets are shown in Table 1. The algorithm runs 10 times in each scenario and Table 2 gives the comparison results of averages. Because the distance objective is the unique optimality criteria in ACS$^D$, ACS$^D$ outperforms MACS in terms of the distance objective value $f_1(x)$, but it is poor at turning angle objective value $f_2(x)$. In addition, the deviation from the ideal solution in ACS$^D$ is larger than the deviation in MACS. MACS can find the satisfying solution which has the smallest deviation from the ideal solution through the cooperative optimization among multiple ant colonies. This experimental results show that MACS is better than ACS$^D$ according to the actual applications requirements of UUV task assignment problem. The same comparison has been done for many times with different targets layout, and the comparison results are similar.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>{30,0},{70,0},{45,100},{55,100}</td>
</tr>
<tr>
<td>$B$</td>
<td>{20,0},{30,0},{40,55},{30,75},{32,70},{30,15},{45,35},{60,60},{70,80}</td>
</tr>
</tbody>
</table>

Figure 1. Results in scenario $A$

Figure 2. Results in scenario $B$

Table 1. Coordinates of targets

In Table 3, the outputs of MACS are compared to the solutions of ACS using weighting method (denoted by ACS$^W$). We still use the coordinates of targets in Table 1 and compare the averages of running 10 times. This comparison results show that both distance objective values $f_1(x)$ and turning angle objective values $f_2(x)$ using MACS are superior to corresponding values using ACS$^W$. Moreover, the deviation from the ideal solution in ACS$^W$ is larger than the deviation in MACS. The reason for this results is that the multiple optimality criteria have different measurement units and order of magnitudes. Thus, the single optimality criteria computed by weighting method is lack of guidance and direction during the searching procedure of ant colony.
5. Conclusion

(1) In this paper, we considered the situation which is more appropriate in the real UUV task assignment problem for reacquisition. Besides the objective of minimizing the total distance of multiple vehicles, we also took into account the objectives of minimizing the total turning angle and the constraint of balancing the targets visited by each vehicle.

(2) The task assignment problem for multiple UUVs was modeled as a constrained multi-objective MTSP. As many proficient heuristic methods to solve MTSP can not deal with multi-objective optimization problem directly, this research made the first attempt to apply the ant colony algorithm to solve constrained multi-objective MTSP.

(3) The proposed Multiple Ant Colonies System (MACS) method can solve the problem with multiple objectives which have different measurement units and order of magnitudes. It is inappropriate to combine the distance objective and turning angle objective with weighting method. However, it is reasonable to optimize multiple objectives by multiple ant colonies respectively. The punishment pheromone updating on dominated solutions is designed to improve the quality of solutions. The ideal solution and the deviation from the ideal solution are defined, and the best one with the smallest deviation is selected from the set of Pareto optimal solutions as the output of algorithm.

(4) The simulation results show that the output of the UUV task assignment algorithm can satisfy the constrained multi-objective requirement of the problem discussed. And the quality of the solutions generated by MACS are better than or equal to the quality of solutions obtained by ACS only considering distance objective and ACS using weighting method.

6. Acknowledgment

This research is supported by 863 Program of China (2006AA04Z262) and National Nature Science Foundation of China (60775061). And thanks to the fellowship of Underwater Vehicle Research Center, Shenyang Institute of Automation, CAS.

7. References