IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 30, 2020, accepted November 9, 2020, date of publication November 16, 2020, date of current version November 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3037679

Area Decomposition Algorithm for Large Region Maritime Search

SHENGWEI XING, RENDA WANG, AND GANG HUANG
Navigation College, Dalian Maritime University, Dalian 116026, China

Corresponding author: Shengwei Xing (xingsw@dlmu.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 51709032, and in part by the Natural Science Foundation of Liaoning Provence under Grant 2019-ZD-0149 and Grant 2020-HYLH-27.

ABSTRACT In this paper, an area decomposition algorithm is presented that is suitable for multiple vessels and multiple aircraft participating in maritime searches over a large region. The algorithm can decompose the entire sea region to be searched (deemed as polygonal area) into nonoverlapping subpolygons (search subareas) according to the sizes of the areas covered by various search facilities while considering their search capabilities as well as their corresponding commence search points. The algorithm draws on the concept of a polygon division algorithm in computational geometry. The main novelty in this study is the optimization of the classic polygon division algorithm by introducing a ''maximizing-minimum-angle'' strategy, which can effectively compensate for the deficiency of the traditional algorithm, as reflected in the area decomposition result. This improved algorithm can produce rectangle-like subareas, especially for a rectangular search region, which is commonly used in maritime search operations. For nonrectangular search regions, a right-angle division can be achieved so that the shapes of the search subareas are more conducive to planning specific search routes for search facilities. Fast maritime search coverage over a large region can be achieved. The effectiveness of the algorithm is validated by comparing decomposition results before and after the improvement.

INDEX TERMS Maritime search, area decomposition, marine safety, search facility, search subarea.

I. INTRODUCTION

Because the speed of aircraft far exceeds the speed of ships, when an airplane crash occurs on a vast ocean, the sea region involved in search and rescue operations will be enormous. In the Malaysia Airlines Flight 370 accident, the approximate maximum flight radius of the airplane reached 5250 kilometers, and the theoretical search area exceeded 86 million square kilometers, which was the largest search operation at sea in history [1]–[4]. Fig.1 shows this theoretical search region. To make scientific and reasonable search decisions, we must first accurately identify all available search facilities. Timely, accurate, and comprehensive access to all available search facility information is the basis for crafting a search plan. At present, the acquisition technology of marine search facilities is mature, especially information acquisition technology for marine search facility dynamics, which has experienced revolutionary breakthroughs such as long range identification and tracking (LRIT) and universal shipborne automatic identification system (AIS) [5]–[8]. These rich data types of search and rescue facilities provide basic conditions for scientific methods for search and rescue. To ensure the efficiency of a search operation, the search region should be reasonably divided into subareas when it is large, and all available search facilities should be assigned to cooperate in their respective search subareas to achieve complete coverage of the region. During a maritime search, polygons are usually used to describe the search region, and the area of the polygon region is equal to the size of the search region. In recent years, several studies have been conducted on how to carry out collaborative work between mobile robots such as unmanned aerial vehicles (UAVs) and autonomous underwater vehicles (AUVs). Many of these studies examined the question of region division and coverage [9]–[23]. Polygon division is a classic problem in the field of computational geometry, and research has been extensive [24]–[29]. References [30]–[35] examined triangulation algorithms for polygons. References [36]–[41] provided useful algorithms for the rectangular and trapezoidal division of polygons. References [42]–[44] introduced algorithms for dividing polygons

The associate editor coordinating the review of this manuscript and approving it for publication was F. R. Islam.

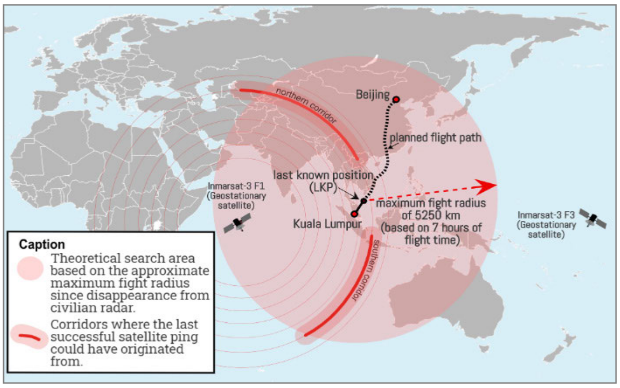

northern corridor
Beijing
planned flight path
last known position (LKP)
maximum fight radius of 5250 km (based on 7 hours of flight time)
Kuala Lumpur
Inmarsat-3 F1 (Geostationary satellite)
Inmarsat-3 F3 (Geostationary satellite)
southern corridor
Caption
Theoretical search area based on the approximate maximum fight radius since disappearance from civilian radar.
Corridors where the last successful satellite ping could have originated from.

FIGURE 1. The theoretical search area of Malaysia Airlines Flight 370.

according to a given area size. Reference [42] proposed a "divide and conquer sweep line algorithm" for polygon area division, which provides an important reference value for solving the search region decomposition problem. As this algorithm was originally aimed at the division of a multiple-robot workspace, its result is not suitable for maritime search in practice. The main contributions can be summarized as follows:

(1) Determining the subareas of search facilities when a search region is large is expressed as a polygon area decomposition problem, which is solved by computational geometric algorithms.

(2) The traditional polygon decomposition algorithm is optimized by introducing a "maximizing-minimum-angle" strategy that can effectively compensate for the deficiency of the traditional algorithm, as reflected in the area decomposition results that are more conducive to search operations at sea.

(3) The computational complexity of the proposed algorithm is analyzed. The division procedure requires $O(n+v) = O(m)$ time, which means this algorithm can be computed in constant time, so it can meet actual application needs.

The paper is organized as follows: Section 2 introduces the search area decomposition problem; Section 3 outlines the polygon area decomposition algorithm; Section 4 gives an example of the traditional decomposition algorithm; Section 5 presents the improvement of the algorithm; Section 6 concludes the paper and proposes the future focus of additional work.

II. PROBLEM DESCRIPTION

If the area of the sea region to be searched is very large, and there is no specific information showing that the possibility of the target in distress in some positions of the sea region is greater than other positions, then the position of the target is generally considered to be uniformly (equal probability) distributed in this region. Therefore, it is necessary to carry out complete search coverage, typically using a parallel sweep search (PS) pattern. Since multiple search facilities are acting in a simultaneously coordinated fashion, the entire region to be searched should be divided into several partitions of appropriate size and assigned to each search facility. By taking the size of the area that can be covered by each facility as the metric for dividing the search region and comprehensively considering the commence search points (CSPs), the problem of decomposing the maritime search region can be described as

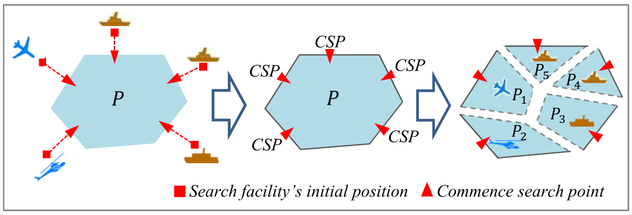


FIGURE 2. Diagram of the search area decomposition problem.

(1) Polygon P represents the sea region to be searched, R represents the area of P;

(2) There are m vessels and n aircraft assigned to carry out search operations in this region;

(3) The search area allocated to vessel i is $A_i^v, i \in \{1, \ldots, m\}$, and the search area allocated to aircraft j is $A_j^a, j \in \{1, \ldots, n\}$, and satisfies $\sum_{i=1}^m A_i^v + \sum_{j=1}^n A_j^a = R$;

(4) All the search facilities drive from the initial position to P at full speed and reach P at each particular point (called site) $S_k, k \in \{1, \ldots, m+n\}$ on the edge of the polygon. This point is commonly known as the CSP;

(5) All search facilities will immediately start searching operations once they reach their respective CSP, as shown in Fig.2.

Polygon P is decomposed into a set of nonoverlapping subpolygons $P_1, \cdots, P_n$, each of which has a specified area. Note that each polygon P_i also has a CSP on its edge; this kind of polygon area decomposition problem is also known as anchored area decomposition on sites $S_1, \cdots, S_n$.

III. POLYGON AREA DECOMPOSITION ALGORITHM

A. GENERAL DEFINITION AND NOTATION

In this paper, "polygon" refers to its internal area (including its boundary). The boundary of a polygon will be explicitly referred to as edges.

Defining the following notations:

- P - the polygon corresponding to the search area;
- CP - the convex polygon (common search region at sea);
- $Area\ (P_i)$ - the area of a polygon P_i;
- $AreaRequired\ (S_i)$ - the area required for each site S_i, which specifies the desired area of each search facility;
- $S\ (CP)$ - the list of sites to be assigned to CP, together with their area requirements;
- $|S\ (CP)|$ - the quantity of sites for CP;
- $V\ (CP)$ - the vertices of the polygon CP;

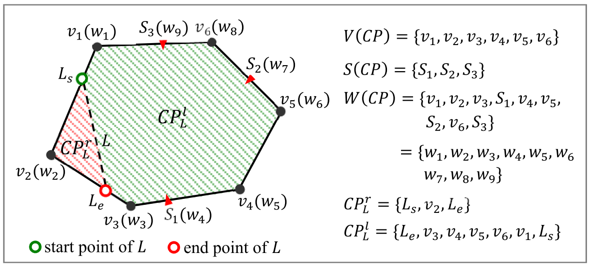

$v_1(w_1)$
$S_3(w_9)$
$v_6(w_8)$
$S_2(w_7)$
$v_5(w_6)$
L_s
CP_L^l
CP_L^r
L
$v_2(w_2)$
L_e
$v_3(w_3)$
$S_1(w_4)$
$v_4(w_5)$
$V(CP) = \{v_1, v_2, v_3, v_4, v_5, v_6\}$
$S(CP) = \{S_1, S_2, S_3\}$
$W(CP) = \{v_1, v_2, v_3, S_1, v_4, v_5, S_2, v_6, S_3\}$
$= \{w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9\}$
$CP_L^r = \{L_s, v_2, L_e\}$
$CP_L^l = \{L_e, v_3, v_4, v_5, v_6, v_1, L_s\}$
○ start point of L
○ end point of L

FIGURE 3. The 6-vertex polygon P is divided into two pieces by sweep line $L(L_e, L_s)$, and the results are CP_L^r (a triangle) and CP_L^l (a 7-vertex polygon).

- $W(CP)$ - the sequence of vertices and sites for CP (each element in the sequence is represented as $w_1, \cdots, w_m$). In this paper, $W(CP)$ is ordered counterclockwise (CCW).

Let $L = (L_s, L_e)$ denote a line segment oriented from one endpoint L_s to another endpoint L_e, which are both on the edge of CP. Polygon CP_L^r is the portion of CP to the right of L; CP_L^l is the portion of CP to the left of L, and $CP_L^l = CP - CP_L^r$, as shown in Fig.3.

A polygon P for which $|S(CP)| = q$ is called a q-site polygon. A polygon P is also called:

- area-complete if $AreaRequired(S(P)) = Area(P)$;
- area-incomplete if $AreaRequired(S(P)) > Area(P)$;
- site-incomplete if $AreaRequired(S(P)) < Area(P)$.

Using the definitions and notations, the anchored area decomposition problem on sites $S_1, \cdots, S_n$ can be restated as follows:

Given an n-site area-complete polygon P divided into a set of convex pieces $CP_j, j = 1, \cdots, p$ and site $S(P) = \{S_1, \cdots, S_n$ on P, construct n 1-site area-complete polygons $P_1, \cdots, P_n$ with $S(P_i) = \{S_i\}$.

B. CONVEX DIVIDE ALGORITHM

The anchored area decomposition problem can be solved by an algorithm called *ConvexDivide,* which uses a divide-and-conquer sweep-line approach to construct a convex polygon area partition [42]. A given polygon P is divided into two subpolygons using one or more sweep lines, and each smaller subpolygon is recursively divided until the entire partition has been constructed. Fig.4 shows this algorithm procedure.

C. ALGORITHM COMPLEXITY ANALYSIS

The procedure *ConvexDivide* takes time linear in the size of the list $W(P)$. Each vertex and site in $W(P)$ is considered at most twice in this procedure: once to add it to either P_L^r or P_L^l and possibly once to subtract it if the area of P_L^r is initially too large. Vertices can be added to or subtracted from polygons in constant time, and as each vertex is added to P_L^r, the change in the area (the area of the triangle added to the polygon) can be computed in constant time. Thus, in total, this procedure requires $O(n + v) = O(m)$ time, where $v = |V(CP)|$ and $m = |W(CP)|$. In the worst case, *ConvexDivide* will partition a q-site polygon, $q > 1$ with v vertices, into a 1-site polygon with 3 vertices and a $(q-1)$-site polygon with $v+1$ vertices in $O(q + v)$ time. In this case, the next call to *ConvexDivide* to partition the $(q-1)$-site polygon will also require $O(q + v) = O((q-1) + (v+1))$ time. To compute the complete partition of a convex n-site polygon P with v vertices, *ConvexDivide* will be called exactly $(n-1)$ times. Thus, in the worst case, $O((n-1) + (n+v))$ time is required to compute the complete partition into n 1-site polygons [42].

```
Input:
convex polygon $P$;
the list $W(P) = w_k, k = 1, \cdots, m$ of vertices and sites in CCW order;
the set of sites $S(P) = S_1, \cdots, S_q$ numbered according to their order in the list $W(P)$.
Output:
Two polygons $P_L^l$ and $P_L^r$ and their sites $S(P_L^l)$ and $S(P_L^r)$.
Procedure:
1. Assume $S_1 = w_k$; $L \leftarrow (w_1, w_k)$
2. $S(P_L^r) \leftarrow \{S_1\}$
3. while $Area(P_L^r) < AreaRequired(S(P_L^r))$ and $L_e \neq S_n$ do
      if (k > 1) and ($w_{k-1} \in S(P)$) then
     $S(P_L^r) \leftarrow S(P_L^r) \cup w_{k-1}$
      $k \leftarrow k + 1$
      $L_e \leftarrow w_k$
4. if $L_e = S_1$ and $Area(P_L^r) > AreaRequired(S(P_L^r))$ then
      Move $L_s$ CCW along $P_L^r$ until $Area(P_L^r) = AreaRequired(S(P_L^r))$
   else if $L_e = S_n$ and $Area(P_L^r) < AreaRequired(S(P_L^r))$ then
      Move $L_s$ CW along $P$ until $Area(P_L^r) = AreaRequired(S(P_L^r))$
   else
      Interpolate to find the point $t$ on edge $(w_{k-1}, w_k)$ such that
      if $L_e = t$ then $Area(P_L^r) = AreaRequired(S(P_L^r))$
      $L_e \leftarrow t$
5. $P_L^l = P - P_L^r$
6. $S(P_L^l) = S \setminus S(P_L^r)$
```

FIGURE 4. The procedure for the *ConvexDivide* algorithm.

IV. ALGORITHM EXAMPLE

Giving a rectangular sea region (CP) to be searched whose area is 2000 $nmile^2$, four facilities (two aircraft and two vessels) are assigned to search within this region. Fig.5(a)

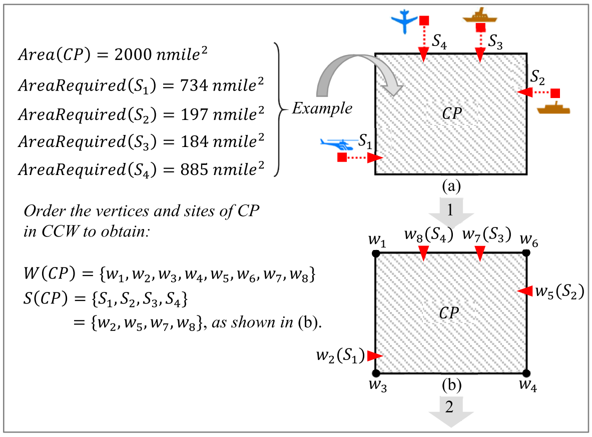
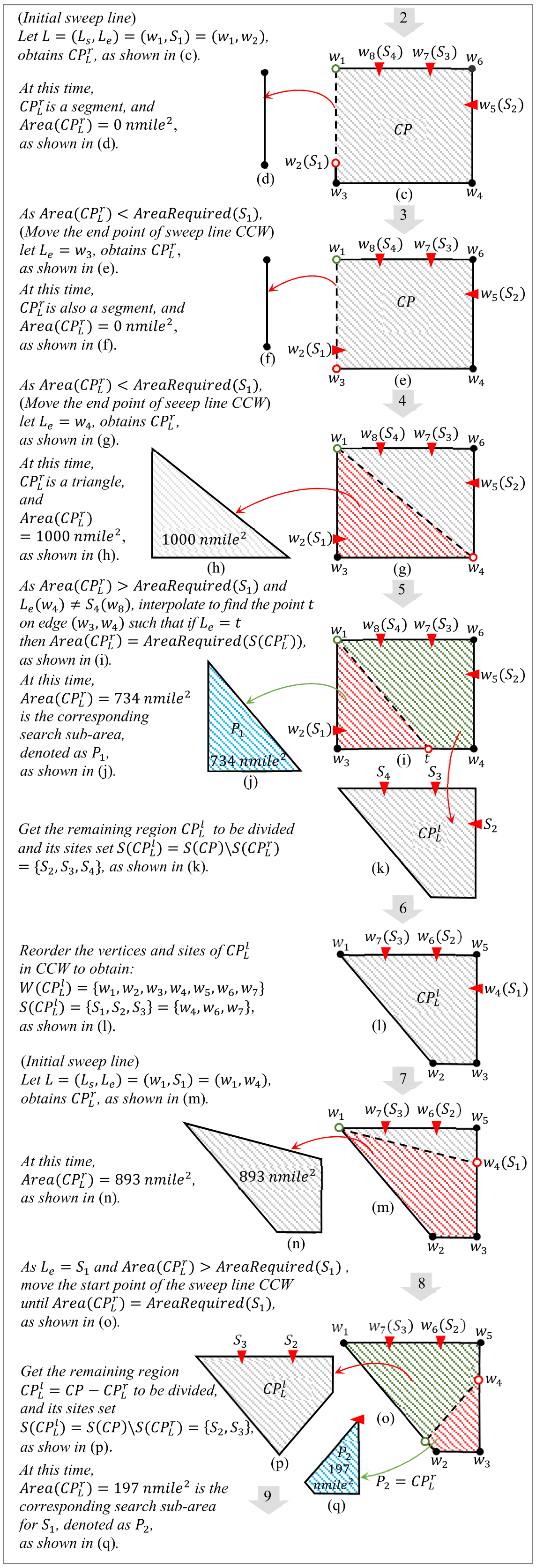

$Area(CP) = 2000\ nmile^2$
$AreaRequired(S_1) = 734\ nmile^2$
$AreaRequired(S_2) = 197\ nmile^2$
$AreaRequired(S_3) = 184\ nmile^2$
$AreaRequired(S_4) = 885\ nmile^2$
Example
S_4
S_3
S_2
S_1
CP
(a)
1
Order the vertices and sites of CP
in CCW to obtain:
$W(CP) = \{w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8\}$
$S(CP) = \{S_1, S_2, S_3, S_4\}$
$= \{w_2, w_5, w_7, w_8\}$, as shown in (b).
w_1
$w_8(S_4)$
$w_7(S_3)$
w_6
$w_5(S_2)$
CP
$w_2(S_1)$
w_3
(b)
w_4
2

FIGURE 5. Algorithm example.



(Initial sweep line)
Let $L = (L_s, L_e) = (w_1, S_1) = (w_1, w_2)$, obtains CP_L^r, as shown in (c).

At this time, CP_L^r is a segment, and $Area(CP_L^r) = 0\ nmile^2$, as shown in (d).

As $Area(CP_L^r) < AreaRequired(S_1)$, (Move the end point of sweep line CCW) let $L_e = w_3$, obtains CP_L^r, as shown in (e).

At this time, CP_L^r is also a segment, and $Area(CP_L^r) = 0\ nmile^2$, as shown in (f).

As $Area(CP_L^r) < AreaRequired(S_1)$, (Move the end point of seeep line CCW) let $L_e = w_4$, obtains CP_L^r, as shown in (g).

At this time, CP_L^r is a triangle, and $Area(CP_L^r) = 1000\ nmile^2$, as shown in (h).

As $Area(CP_L^r) > AreaRequired(S_1)$ and $L_e(w_4) \neq S_4(w_8)$, interpolate to find the point t on edge (w_3, w_4) such that if $L_e = t$ then $Area(CP_L^r) = AreaRequired(S(CP_L^r))$, as shown in (i).

At this time, $Area(CP_L^r) = 734\ nmile^2$ is the corresponding search sub-area, denoted as P_1, as shown in (j).

Get the remaining region CP_L^l to be divided and its sites set $S(CP_L^l) = S(CP) \backslash S(CP_L^r) = \{S_2, S_3, S_4\}$, as shown in (k).

Reorder the vertices and sites of CP_L^l in CCW to obtain:
$W(CP_L^l) = \{w_1, w_2, w_3, w_4, w_5, w_6, w_7\}$
$S(CP_L^l) = \{S_1, S_2, S_3\} = \{w_4, w_6, w_7\}$,
as shown in (l).

(Initial sweep line)
Let $L = (L_s, L_e) = (w_1, S_1) = (w_1, w_4)$, obtains CP_L^r, as shown in (m).

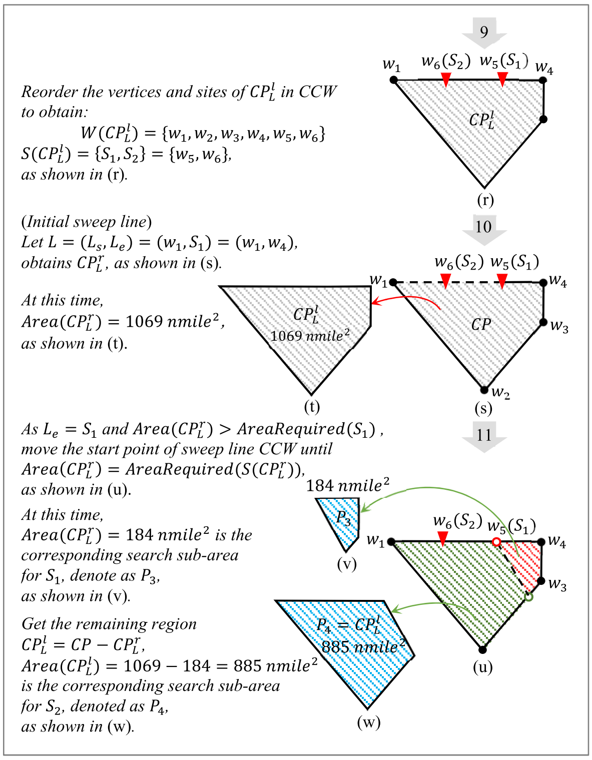
At this time, $Area(CP_L^r) = 893\ nmile^2$, as shown in (n).

As $L_e = S_1$ and $Area(CP_L^r) > AreaRequired(S_1)$, move the start point of the sweep line CCW until $Area(CP_L^r) = AreaRequired(S_1)$, as shown in (o).

Get the remaining region $CP_L^l = CP - CP_L^r$ to be divided, and its sites set $S(CP_L^l) = S(CP) \backslash S(CP_L^r) = \{S_2, S_3\}$, as show in (p).

At this time, $Area(CP_L^r) = 197\ nmile^2$ is the corresponding search sub-area for S_1, denoted as P_2, as shown in (q).

FIGURE 5. *(continued.)* Algorithm example.



Reorder the vertices and sites of CP_L^l in CCW to obtain:
$W(CP_L^l) = \{w_1, w_2, w_3, w_4, w_5, w_6\}$
$S(CP_L^l) = \{S_1, S_2\} = \{w_5, w_6\}$,
as shown in (r).

(Initial sweep line)
Let $L = (L_s, L_e) = (w_1, S_1) = (w_1, w_4)$, obtains CP_L^r, as shown in (s).

At this time, $Area(CP_L^l) = 1069\ nmile^2$, as shown in (t).

As $L_e = S_1$ and $Area(CP_L^r) > AreaRequired(S_1)$, move the start point of sweep line CCW until $Area(CP_L^r) = AreaRequired(S(CP_L^r))$, as shown in (u).

At this time, $Area(CP_L^r) = 184\ nmile^2$ is the corresponding search sub-area for S_1, denote as P_3, as shown in (v).

Get the remaining region $CP_L^l = CP - CP_L^r$, $Area(CP_L^l) = 1069 - 184 = 885\ nmile^2$ is the corresponding search sub-area for S_2, denoted as P_4, as shown in (w).

FIGURE 5. *(continued.)* Algorithm example.

shows their initial positions and CSPs (corresponding to sites S_1, S_2, S_3, S_4), which are located on CP's edges. According to their different search capabilities, the search areas covered by these four search facilities are 734, 197, 184, and 885 $nmile^2$. The decomposition procedure is shown in Fig.5.

V. ALGORITHM IMPROVEMENT

A. ANALYSIS OF DECOMPOSITION RESULT

The *ConvexDivide* algorithm can divide the entire search region into search subareas according to each search facility's capability (coverage area) and its corresponding CSP. Fig.6 shows the result of the above example.

To achieve full coverage of each search subarea, search planners need to further specify the search pattern and specific routes of these facilities in their respective search subareas. The abovementioned parallel sweep search (PS) pattern is normally used when the uncertainty in the survivor's location is large, requiring a large region to be searched with uniform coverage. It is most effective when used over water or reasonably flat terrain. A parallel sweep search pattern usually covers a rectangular area. It is almost always used when a large search region must be divided into subareas for assignment to individual search facilities that will be on-scene at the same time. The parallel sweep search pattern generally requires the subareas to be rectangular. However, Fig.6 shows that the decomposition result of

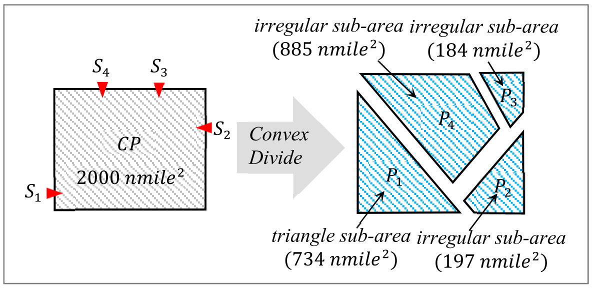


FIGURE 6. Decomposition result of example.

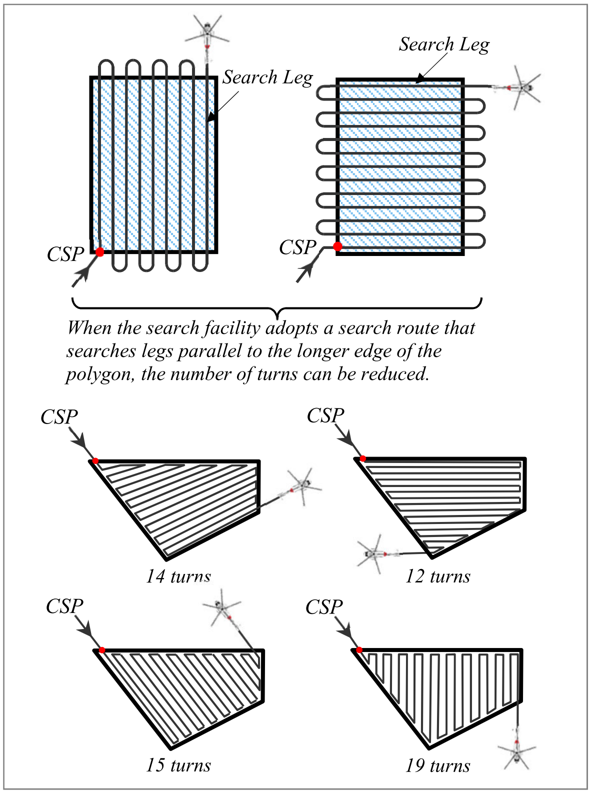

Search Leg
Search Leg
CSP
CSP
When the search facility adopts a search route that searches legs parallel to the longer edge of the polygon, the number of turns can be reduced.
CSP
CSP
14 turns
12 turns
CSP
CSP
15 turns
19 turns

FIGURE 7. The parallel sweep search within the search subarea.

the *ConvexDivide* algorithm does not fulfill this requirement, which is not conducive to the implementation of a parallel sweep search.

For a rectangular subarea, the specific search route is easier to plan. By making the search leg parallel to the longer edge of the rectangle, the number of turns of search facilities can be reduced, thereby increasing the search efficiency, as shown in Fig.7. For a nonrectangular subarea, the search facility can reach the minimum turns only when the search legs are parallel to a specific edge of the polygon. Since the direction of search legs cannot be judged intuitively relative to the rectangular area, it is necessary to calculate the number of turns parallel to each edge separately and finally choose the search leg direction to generate the minimum number of turns. The irregular subarea not only increases the OSC (on-scene coordinator, a person designated to coordinate search and rescue operations within a specific area) difficulty in specifying each search facility's search route but is also not good for carrying out search operations. Therefore, improving the *ConvexDivide* algorithm to generate rectangular search subareas as much as possible is a valuable goal.

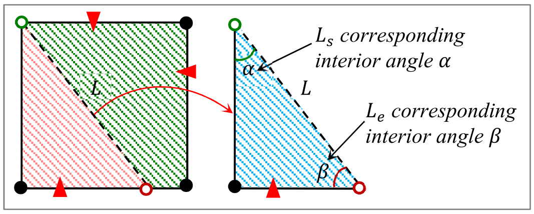

L
L_s corresponding interior angle α
α
L
L_e corresponding interior angle β
β

FIGURE 8. The same side interior angles are generated by *L*.

B. WAYS TO IMPROVE ALGORITHM

The *ConvexDivide* algorithm uses a sweep line as a dividing tool to decompose a convex polygon into two smaller subpolygons. The shape of each subpolygon fully depends on the position of the sweep line. Therefore, we can optimize the shapes of the subpolygons by adding a strategy to control the position of the sweep line. Fig.8 shows a rectangular region to be divided since the two endpoints of the sweep line $L = (L_s, L_e)$ are located on the edge of the polygon, and two interior angles are generated within the same adjacent edge. The interior angle with the start point L_s is denoted as α, and the other angle on the same side with the end point L_e is denoted as β. To make the shape of subareas generated by the sweep line rectangular, the condition $\alpha = \beta = 90^\circ$ is required, that is, the two interior angles on the same side are both right angles. For convenience, this paper refers to the sweep line that satisfies the above condition as a "right-angle sweep line", and the division is called a "right-angle division". Next, we discuss how to generate a "right-angle sweep line" by improving the original algorithm by adding a control strategy.

The sizes of interior angles α and β are determined by the position of sweep line L, which is further determined by the position of its start point L_s and end point L_e. Therefore, the interior angles α and β can be altered by moving the positions of L_s and L_e. When moving L_s and L_e, the size of α and β is changed as follows:

When the number of edges of the polygon CP_L^r is unchanged, if we fix L_s and move L_e CCW, angle β will decrease and angle α will increase, as shown in Fig.9(a); if we fix L_s and move L_e CW, angle β will increase and angle α will decrease, as shown in Fig.9(b); if we fix L_e and move L_s CCW, angle β will decrease and angle α will increase, as shown in Fig.9(c); if we fix L_e and move L_s CW, angle β will increase and angle α will decrease, as shown in Fig.9(d). This rule can be proven by the sum of the interior angles of the polygon formula.

1) SUM OF THE INTERIOR ANGLES OF THE POLYGON FORMULA

If a convex polygon has n sides, then the sum of its interior angle is given by the formula $S = (n - 2)180^\circ$.

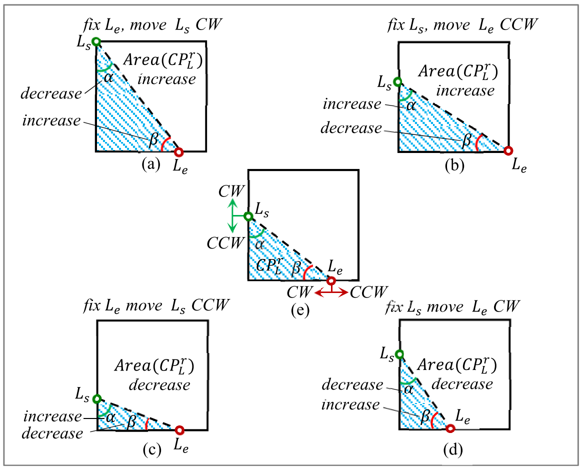


FIGURE 9. The alteration in interior angles and area of CP_L^r.

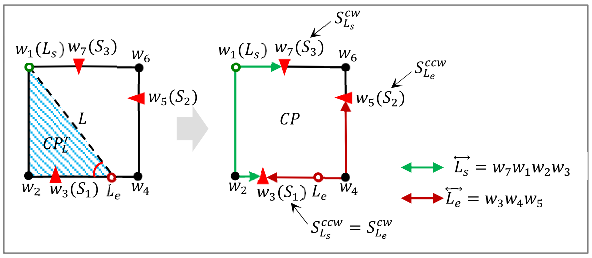

$w_1(L_s)$ $w_7(S_3)$ w_6
$w_5(S_2)$
L
CP_L^r
w_2 $w_3(S_1)$ L_e w_4
$S_{L_s}^{cw}$
$w_1(L_s)$ $w_7(S_3)$ w_6
$S_{L_e}^{ccw}$
$w_5(S_2)$
CP
w_2 $w_3(S_1)$ L_e w_4
$S_{L_s}^{ccw} = S_{L_e}^{cw}$
$\overleftrightarrow{L_s} = w_7w_1w_2w_3$
$\overleftrightarrow{L_e} = w_3w_4w_5$

FIGURE 10. The movable range of L_s and L_e.

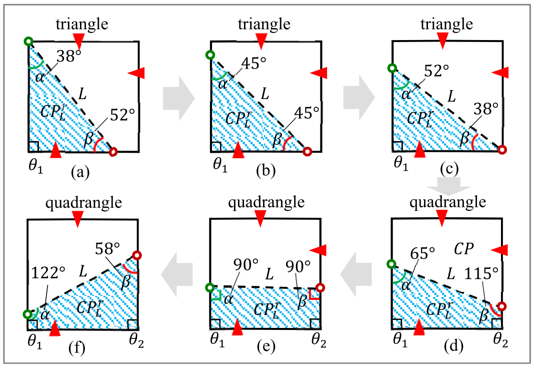


FIGURE 11. The variation of interior angles α and β when the sweep line performs "CCW equal-area movement".

The above formula shows that if the sides of a convex polygon are fixed, then the sum of its interior angles is a constant value. Therefore, considering any two adjacent interior angles, when one of them increases (or decreases), the other one has to decrease (or increase). When L_s and L_e are moving, the area of CP_L^r changes as follows:

If we keep L_e still and move L_s CCW, then $Area(CP_L^r)$ will decrease; if we move L_s CW, then $Area(CP_L^r)$ will increase;

If we keep L_s still and move L_e CCW, then $Area(CP_L^r)$ will increase; if we move L_s CW, then $Area(CP_L^r)$ will decrease, as shown in Fig 9(e).

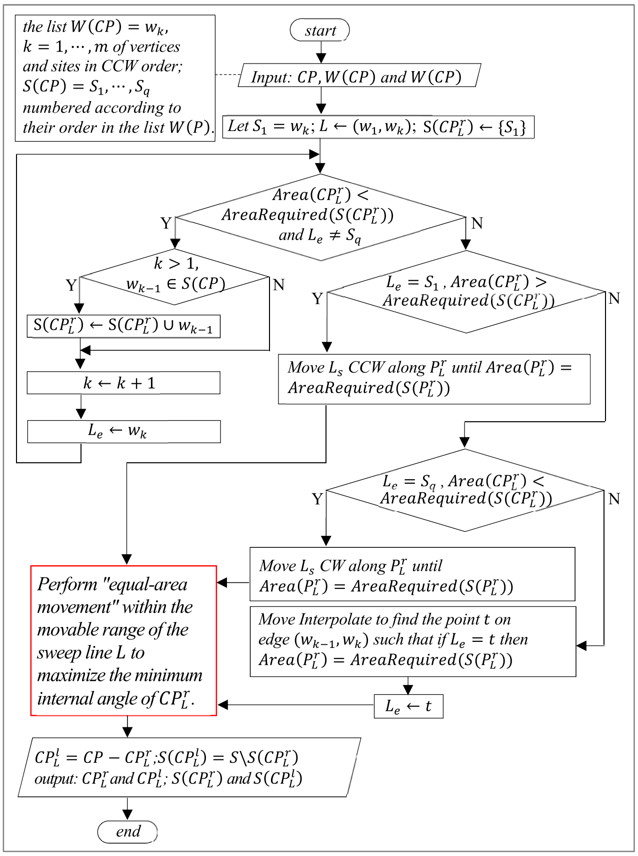

the list $W(CP) = w_k$, $k = 1, \cdots, m$ of vertices and sites in CCW order; $S(CP) = S_1, \cdots, S_q$ numbered according to their order in the list $W(P)$.
start
Input: $CP, W(CP)$ and $W(CP)$
Let $S_1 = w_k$; $L \leftarrow (w_1, w_k)$; $S(CP_L^r) \leftarrow \{S_1\}$
$Area(CP_L^r) <$ $AreaRequired(S(CP_L^r))$ and $L_e \neq S_q$
Y
N
$k > 1$, $w_{k-1} \in S(CP)$
Y
N
$S(CP_L^r) \leftarrow S(CP_L^r) \cup w_{k-1}$
$k \leftarrow k + 1$
$L_e \leftarrow w_k$
$L_e = S_1$, $Area(CP_L^r) >$ $AreaRequired(S(CP_L^r))$
Y
N
Move L_s CCW along P_L^r until $Area(P_L^r) =$ $AreaRequired(S(P_L^r))$
$L_e = S_q$, $Area(CP_L^r) <$ $AreaRequired(S(CP_L^r))$
Y
N
Move L_s CW along P_L^r until $Area(P_L^r) = AreaRequired(S(P_L^r))$
Move Interpolate to find the point t on edge (w_{k-1}, w_k) such that if $L_e = t$ then $Area(P_L^r) = AreaRequired(S(P_L^r))$
$L_e \leftarrow t$
Perform "equal-area movement" within the movable range of the sweep line L to maximize the minimum internal angle of CP_L^r.
$CP_L^l = CP - CP_L^r$;$S(CP_L^l) = S \backslash S(CP_L^r)$
output: CP_L^r and CP_L^l; $S(CP_L^r)$ and $S(CP_L^l)$
end

FIGURE 12. The flowchart of the improved *ConvexDivide* algorithm.

TABLE 1. The interior angles of CP_L^r while sweep line L performs "equal-area movement".

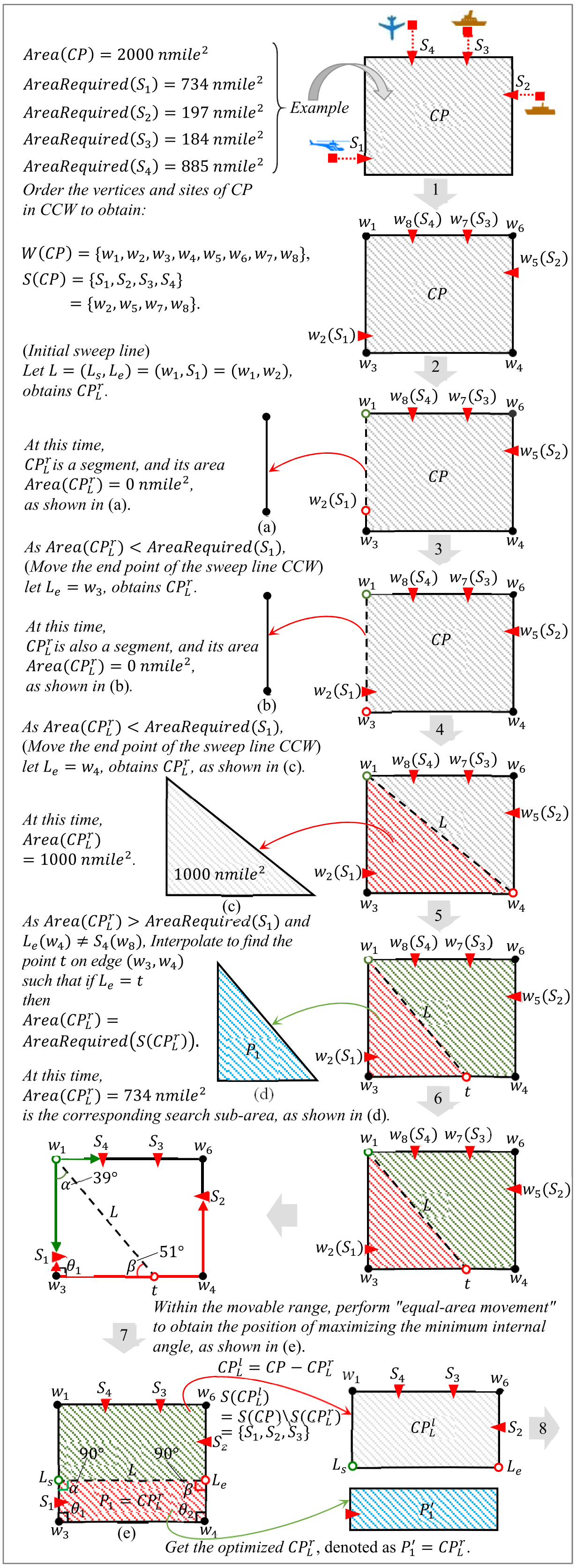
CP_L^r	Triangle (a)	 (b)	 (c)	Quadrangle (d)	 (e)	 (f)
α	38	45	52	65	90	122
β	52	45	38	115	90	58
θ_1	90	90	90	90	90	90
θ_2				90	90	90
sort in ascending order	$\alpha\beta\theta_1$	$\overline{\alpha\beta}\theta_1$	$\beta\alpha\theta_1$	$\alpha\overline{\theta_1\theta_2}\beta$	$\overline{\alpha\beta\theta_1\theta_1}$	$\beta\overline{\theta_1\theta_2}\alpha$

Through the above analysis, the following conclusion can be drawn.

2) CONCLUSION

To change the sizes of the two interior angles of CP_L^r while keeping $Area(CP_L^r)$ unchanged, both L_s and L_e must be moved simultaneously in the same direction (CW or CCW).

For convenience, the movement of the sweep line under the condition of keeping $Area(CP_L^r)$ unchanged is called "equal-area movement". If L_s and L_e move in the clockwise direction, then it is called "CW equal-area movement"; if L_s and L_e move in the counterclockwise direction, then it is called "CCW equal-area movement". In the case of "equal-area movement", let us further consider the rule of alteration in α and β if the number of polygon edges changes.



$Area(CP) = 2000\ nmile^2$

$AreaRequired(S_1) = 734\ nmile^2$

$AreaRequired(S_2) = 197\ nmile^2$

$AreaRequired(S_3) = 184\ nmile^2$

$AreaRequired(S_4) = 885\ nmile^2$

Example

Order the vertices and sites of CP in CCW to obtain:

$W(CP) = \{w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8\}$,
$S(CP) = \{S_1, S_2, S_3, S_4\}$
$= \{w_2, w_5, w_7, w_8\}$.

(Initial sweep line)
Let $L = (L_s, L_e) = (w_1, S_1) = (w_1, w_2)$, *obtains* CP_L^r.

At this time, CP_L^r *is a segment, and its area* $Area(CP_L^r) = 0\ nmile^2$, *as shown in* (a).

As $Area(CP_L^r) < AreaRequired(S_1)$, *(Move the end point of the sweep line CCW) let* $L_e = w_3$, *obtains* CP_L^r.

At this time, CP_L^r *is also a segment, and its area* $Area(CP_L^r) = 0\ nmile^2$, *as shown in* (b).

As $Area(CP_L^r) < AreaRequired(S_1)$, *(Move the end point of the sweep line CCW) let* $L_e = w_4$, *obtains* CP_L^r, *as shown in* (c).

At this time, $Area(CP_L^r)$ $= 1000\ nmile^2$.

As $Area(CP_L^r) > AreaRequired(S_1)$ *and* $L_e(w_4) \neq S_4(w_8)$, *Interpolate to find the point t on edge* (w_3, w_4) *such that if* $L_e = t$ *then* $Area(CP_L^r) = AreaRequired(S(CP_L^r))$.

At this time, $Area(CP_L^r) = 734\ nmile^2$ *is the corresponding search sub-area, as shown in* (d).

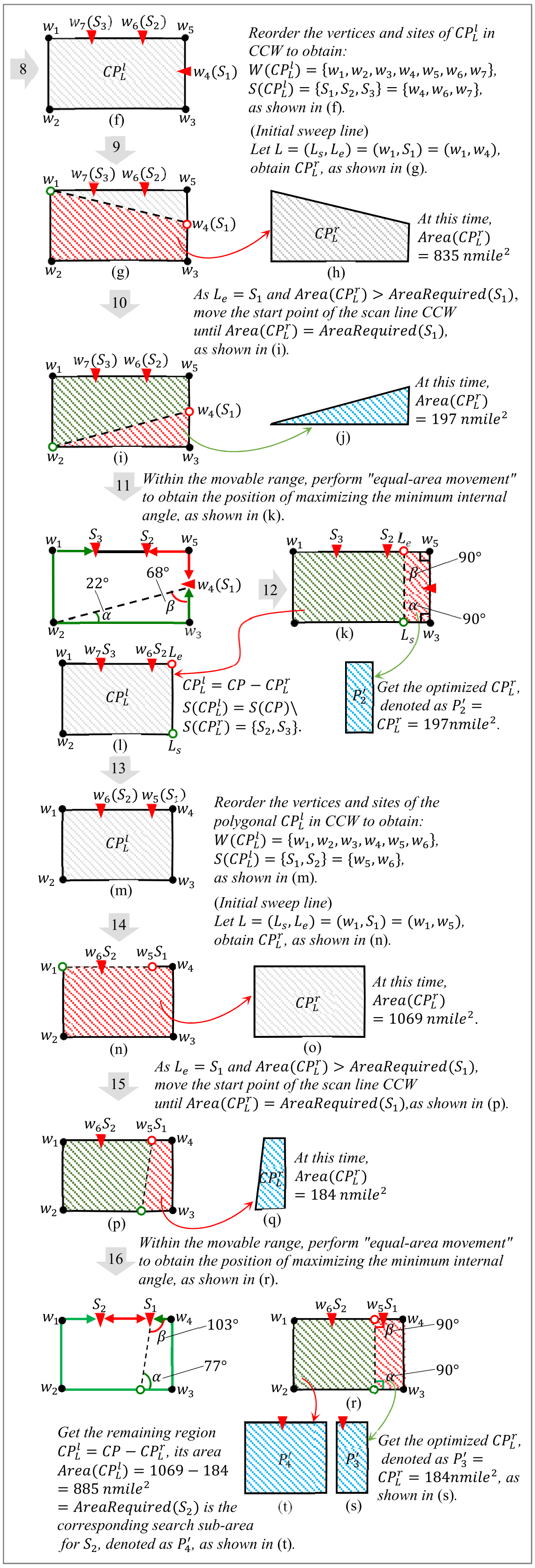
Within the movable range, perform "equal-area movement" to obtain the position of maximizing the minimum internal angle, as shown in (e).

$CP_L^l = CP - CP_L^r$

$S(CP_L^l) = S(CP) \backslash S(CP_L^r) = \{S_1, S_2, S_3\}$

Get the optimized CP_L^r, denoted as $P_1' = CP_L^r$.

FIGURE 13. Improved *ConvexDivide* algorithm example.



Reorder the vertices and sites of CP_L^l *in CCW to obtain:*
$W(CP_L^l) = \{w_1, w_2, w_3, w_4, w_5, w_6, w_7\}$,
$S(CP_L^l) = \{S_1, S_2, S_3\} = \{w_4, w_6, w_7\}$,
as shown in (f).

(Initial sweep line)
Let $L = (L_s, L_e) = (w_1, S_1) = (w_1, w_4)$, *obtain* CP_L^r, *as shown in* (g).

At this time, $Area(CP_L^r)$ $= 835\ nmile^2$

As $L_e = S_1$ *and* $Area(CP_L^r) > AreaRequired(S_1)$, *move the start point of the scan line CCW until* $Area(CP_L^r) = AreaRequired(S_1)$, *as shown in* (i).

At this time, $Area(CP_L^r)$ $= 197\ nmile^2$

Within the movable range, perform "equal-area movement" to obtain the position of maximizing the minimum internal angle, as shown in (k).

$CP_L^l = CP - CP_L^r$
$S(CP_L^l) = S(CP) \backslash$
$S(CP_L^r) = \{S_2, S_3\}$.

Get the optimized CP_L^r, *denoted as* $P_2' =$ $CP_L^r = 197nmile^2$.

Reorder the vertices and sites of the polygonal CP_L^l *in CCW to obtain:*
$W(CP_L^l) = \{w_1, w_2, w_3, w_4, w_5, w_6\}$,
$S(CP_L^l) = \{S_1, S_2\} = \{w_5, w_6\}$,
as shown in (m).

(Initial sweep line)
Let $L = (L_s, L_e) = (w_1, S_1) = (w_1, w_5)$, *obtain* CP_L^r, *as shown in* (n).

At this time, $Area(CP_L^r)$ $= 1069\ nmile^2$.

As $L_e = S_1$ *and* $Area(CP_L^r) > AreaRequired(S_1)$, *move the start point of the scan line CCW until* $Area(CP_L^r) = AreaRequired(S_1)$,*as shown in* (p).

At this time, $Area(CP_L^r)$ $= 184\ nmile^2$

Within the movable range, perform "equal-area movement" to obtain the position of maximizing the minimum internal angle, as shown in (r).

Get the remaining region $CP_L^l = CP - CP_L^r$, *its area* $Area(CP_L^l) = 1069 - 184$ $= 885\ nmile^2$ $= AreaRequired(S_2)$ *is the corresponding search sub-area for* S_2, *denoted as* P_4', *as shown in* (t).

Get the optimized CP_L^r, *denoted as* $P_3' =$ $CP_L^r = 184nmile^2$, *as shown in* (s).

FIGURE 13. *(Continued.)* Improved *ConvexDivide* algorithm example.

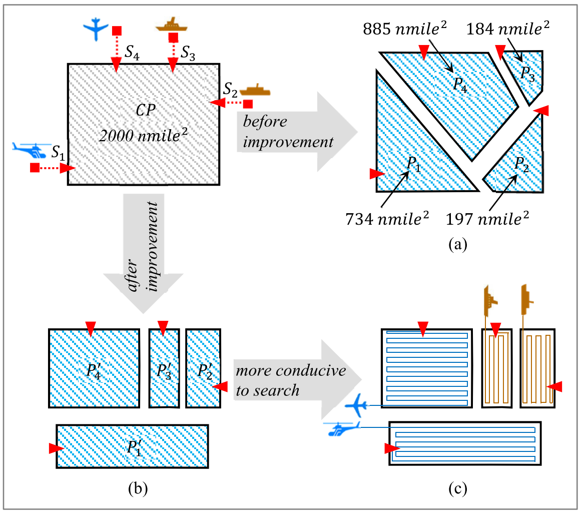


FIGURE 14. Decomposition results contrasted before and after improving the *ConvexDivide* algorithm.

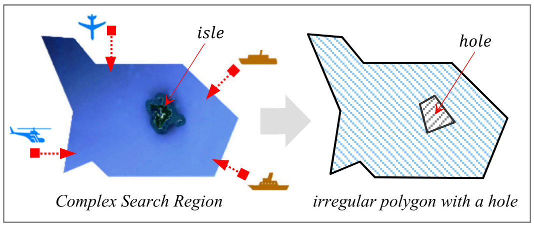


FIGURE 15. The arbitrary search area with an isle inside.

It should be noted that the movable ranges of L_s and L_e are limited by the position of sites. The movable range of L_s is denoted as $\overleftrightarrow{L_s}$, and the movable range of L_e is denoted as $\overleftrightarrow{L_e}$. Therefore, the movable range of sweep line L can be determined by $\overleftrightarrow{L_s}$ and $\overleftrightarrow{L_e}$. $\overleftrightarrow{L_s}$ and $\overleftrightarrow{L_e}$ can be obtained by the following two steps:

Step 1: Starting from L_s, move along the edge of CP to find the nearest site in the CCW direction and record it as $S_{L_s}^{ccw}$; then find the nearest site in the CW direction and record it as $S_{L_s}^{cw}$. The "movable range of L_s" is the edge between $S_{L_s}^{ccw}$ and $S_{L_s}^{cw}$;

Step 2: Starting from L_e, move along the edge of CP to find the nearest site in the CCW direction and record it as $S_{L_e}^{ccw}$; then find the nearest site in the CW direction and record it as $S_{L_e}^{cw}$. The "movable range of L_e" is the edge between $S_{L_e}^{ccw}$ and $S_{L_e}^{cw}$, as shown in Fig.10.

The number of edges may change when sweep line L moves within its movable range. The alteration of the number of edges can further affect the interior angles α and β, as shown in Fig.11.

Fig.11 shows that the shape of the polygon CP_L^r has changed from triangular to quadrangular when the sweep line L moves in the CCW direction. Table 1 presents the corresponding angle values of α and β in six positions (Fig.11(a)-(f)), while the sweep line L performs "CCW equal-area movement".

In position Fig.11(a), CP_L^r is a triangle, and its interior angles are α, β, and θ_1 in ascending order ($\alpha = 38°$ is the minimum interior angle). If L moves in the CCW direction, the size of angle α gradually increases, and the size of angle β gradually decreases. When L is in the position shown in Fig.11(b), the ascending order of these interior angles is $\overline{\alpha, \beta}, \theta_1$ ($\overline{\alpha, \beta}$ indicates that α and β are equal in size). At this point, both α and β are minimum interior angles. If L keeps moving in the CCW direction, α is no longer the minimum interior angle; β instead of α becomes the minimum angle until L is in the position shown in Fig.11(c). If L moves in the CCW direction, CP_L^r will change from a triangle to a quadrangle, and angle α will become the minimum angle again, as shown in Fig.11(d) ($\alpha = 65°$). Next, L moves in the CCW direction, and the size of α continues to increase. When L is in position shown in Fig.11 (e), the four interior angles ($\alpha = \beta = \theta_1 = \theta_2 = 90°$) are all the minimum interior angles. Next, L moves in the CCW direction, the size of α gradually increases, and the size of angle β gradually decreases. Angle α is no longer the minimum interior angle, and β replaces α again to become the minimum interior angle until L is in the position shown in Fig.11(f) ($\beta = 58°$).

From the above analysis, it can be seen that the minimum interior angle of CP_L^r changes when the sweep line L performs "equal-area movement". A special position for L can be found at which the minimum interior angle of CP_L^r reaches its maximum value (see Fig. 11(e)). The four interior angles (α, β, θ_1, θ_2) of CP_L^r are all the minimum interior angles and are all right angles when sweeping line$L$ at this position. This means that the "right-angle division" of the polygon is achieved. This method of making the minimum interior angle of CP_L^r reach its maximum value when the sweep line L makes an "equal-area movement" is called the "maximizing-minimum-angle" strategy. That is, if $\alpha_1, \alpha_2, \cdots, \alpha_n$ are the interior angles of CP_L^r, we make the sweep line perform "equal-area movement" until it reaches a position that satisfies the following condition:

$$Min\ \alpha_i \rightarrow Max, \quad i \in\ 1, \cdots, n.$$

By importing the "maximizing-minimum-angle" strategy into the original *ConvexDivide* algorithm, the decomposition result can be optimized to form a "right-angle sweep" line as much as possible, which is beneficial when planning search routes. Fig.12 shows the improved *ConvexDivide* algorithm procedure.

It can be seen in Fig.12 that adding the "equal-area movement" step does not increase the complexity of the original algorithm.

C. EXAMPLE OF APPLIED IMPROVED ALGORITHM

Fig.13 shows the improved algorithm compared to the previous example.

D. COMPARISON BETWEEN ORIGINAL ALGORITHM AND IMPROVED ALGORITHM

Fig.14 shows the results of the original algorithm and the improved algorithm. The original algorithm can divide the rectangular search area (2000 $nmile^2$) into four search subareas based on the sizes of the area covered by four search facilities (two aircraft and two vessels), namely, search subarea P_1 (734 $nmile^2$), search subarea P_2 (197 $nmile^2$), search subarea P_3 (184 $nmile^2$) and search subarea P_4 (885 $nmile^2$). The shape of P_1 is a triangle, and the shapes of P_2, P_3, and P_4 are all irregular quadrangle (see Fig.14(a)). Such a decomposition result is not conducive to planning specific search routes (e.g., parallel sweep search patterns) for search facilities. Therefore, this is not suitable for a large region maritime search. However, the improved algorithm can obtain four rectangular search subareas, namely, P'_1, P'_2, P'_3, and P'_4, while keeping these CSPs unchanged, and the partition area is the same as the original partition area (see Fig.14(b)). Rectangular subareas are very helpful for planning parallel sweep search routes as they can improve the efficiency of large-scale search coverage (see Fig.14(c)).

After improvement, the ''right-angle division'' of the original area is realized, which is more suitable for planning a search route for each facility within its search subarea.

VI. CONCLUSION AND FUTURE WORK

After maritime distress occurs, it is of great significance to save lives and reduce property losses by immediately carrying out efficient search operations to find the target in distress early and then implement effective rescue procedures. Maritime search operations are carried out within a specific geographic region, which is the search area most likely to contain targets in distress as determined by the joint rescue coordinator center (JRCC). To ensure the efficiency of the search, when the search region is large and the probability distribution of the target position in this region is evenly distributed, it should be reasonably decomposed into search subareas. Then, the subareas should be assigned to each search facility to coordinate operations and quickly achieve complete coverage of the entire region.

The area decomposition algorithm for a large region maritime search proposed in this paper implements a scientific search task allocation method (determining the search subareas for the search facilities) under the condition that the search region is large and its shape is a regular polygon, particularly a rectangular region. When the search region is determined by the line datum or area datum, the shape may be an arbitrary polygon, especially when the search area contains an isle, and there will exist a ''hole'' inside the polygon, as shown in Fig.15. In future work, to achieve efficient search coverage of such irregular regions (corresponding to complex polygons), nonconvex or nonsimple connected polygon decomposition algorithms will be investigated to increase the efficiency of search coverage.

ACKNOWLEDGMENT

The authors would like to thank the Maritime Search and Rescue Center of the People's Republic of China for providing theoretical guidance and practical expertise.

REFERENCES

[1] K. Picard, B. P. Brooke, P. T. Harris, P. J. Siwabessy, M. F. Coffin, M. Tran, M. Spinoccia, J. Weales, M. Macmillan-Lawler, and J. Sullivan, ''Malaysia airlines flight MH370 search data reveal geomorphology and seafloor processes in the remote southeast Indian ocean,'' *Mar. Geol.*, vol. 395, pp. 301–319, Jan. 2018, doi: 10.1016/.j.margeo.2017.10.014.

[2] M. McNutt, ''The hunt for MH370,'' *Science*, vol. 344, no. 6187, p. 947, May 2014, doi: 10.1126/science.1255963.

[3] R. Parnell-Turner, S. J. Sim, and J. Olive, ''Time-dependent crustal accretion on the Southeast Indian ridge revealed by Malaysia airlines flight MH370 search,'' *Geophys. Res. Lett.*, vol. 47, no. 12, Jun. 2020, doi: 10.1029/2020GL087349.

[4] N. Mironova and P. Butterworth-Hayes, ''Learning fast from MH370,'' *Aerosp. Amer.*, vol. 52, no. 7, pp. 20–25, Jul./Aug. 2014.

[5] Y. Huang, Y. Li, Z. Zhang, and R. W. Liu, ''GPU-accelerated compression and visualization of large-scale vessel trajectories in maritime IoT industries,'' *IEEE Internet Things J.*, early access, Apr. 21, 2020, doi: 10.1109/.JIOT.2020.2989398.

[6] R. W. Liu, J. Nie, S. Garg, Z. Xiong, Y. Zhang, and M. S. Hossain, ''Data-driven trajectory quality improvement for promoting intelligent vessel traffic services in 6G-enabled maritime IoT systems,'' *IEEE Internet Things J.*, early access, Oct. 5, 2020, doi: 10.1109/JIOT.2020.3028743.

[7] T. Yang, H. Feng, C. Yang, Y. Wang, J. Dong, and M. Xia, ''Multivessel computation offloading in maritime mobile edge computing network,'' *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4063–4073, Jun. 2019, doi: 10.1109/JIOT.2018.2876151.

[8] T. Yang, H. Feng, S. Gao, Z. Jiang, M. Qin, N. Cheng, and L. Bai, ''Two-stage offloading optimization for Energy–Latency tradeoff with mobile edge computing in maritime Internet of Things,'' *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5954–5963, Jul. 2020, doi: 10.1109/JIOT.2019.2958662.

[9] T. Yang, Z. Jiang, R. Sun, N. Cheng, and H. Feng, ''Maritime search and rescue based on group mobile computing for unmanned aerial vehicles and unmanned surface vehicles,'' *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7700–7708, Dec. 2020, doi: 10.1109/TII.2020.2974047.

[10] M. Coombes, T. Fletcher, W.-H. Chen, and C. Liu, ''Decomposition-based mission planning for fixed-wing UAVs surveying in wind,'' *J. Field Robot.*, vol. 37, no. 3, pp. 440–465, Dec. 2020, doi: 10.1002/rob.21928.

[11] P. Yuan, M. Su, Y. Li, and Q. Zhang, ''Timely and full coverage algorithm for region area with UAVs-assisted small satellite constellation,'' in *Proc. IEEE/CIC Int. Conf. Commun. Workshops China (ICCC Workshops)*, Aug. 2019, pp. 104–108, doi: 10.1109/ICCChinaW.2019.8849946.

[12] Y. Wu, J. Zhu, and K. Gao, ''Multi-UAVs area decomposition and coverage based on complete region coverage,'' in *Proc. SAMSE*, Shanghai, China, Dec. 2018, Art. no. 062009, doi: 10.1088/1757-899X/490/6/062009.

[13] K. Vinh, S. Gebreyohannes, and A. Karimoddini, ''An area-decomposition based approach for cooperative tasking and coordination of UAVs in a search and coverage mission,'' in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2019, pp. 1–8, doi: 10.1109/AERO.2019.8741565.

[14] Q. Pang, Y. Hu, and W. Li, ''UAV coverage track planning based on decomposition along the direction of perpendicular to the width of the area,'' *J. Syst Eng Electron.*, vol. 41, no. 11, pp. 2550–2558, Nov. 2019, doi: 10.3969/j.issn.1001-506X.2019.11.19.

[15] L. D. Nielsen, I. Sung, and P. Nielsen, ''Convex decomposition for a coverage path planning for autonomous vehicles: Interior extension of edges,'' *Sensors*, vol. 19, no. 19, p. 4165, Sep. 2019, doi: 10.3390/s19194165.

[16] P. Qiangwei, H. Yongjiang, and L. Wenguang, ''Research on multi-UAVs coordinated coverage reconnaissance strategy,'' in *Proc. ACM-RCAE*, Beijing, China, Dec. 2018, pp. 117–121, doi: 10.1145/3303714.3303725.

[17] F. Balampanis, I. Maza, and A. Ollero, ''Area decomposition, partition and coverage with multiple remotely piloted aircraft systems operating in coastal regions,'' in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Arlington, VA, USA, Jun. 2016, pp. 275–283, doi: 10.1109/ICUAS.2016.7502602.

[18] S. Yu, ''Decomposition and coverage of multi-UAV cooperative search area,'' *J. Beiing Univ. Aeronaut. Astronaut.*, vol. 41, no. 1, pp. 167–173, 2015. Jan. 2015, doi: 10.13700/j.bh.1001-5965.2014.0056.

[19] J. F. Araujo, P. B. Sujit, and J. B. Sousa, ''Multiple UAV area decomposition and coverage,'' in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl. (CISDA)*, Apr. 2013, pp. 30–37, doi: 10.1109/CISDA.2013.6595424.

[20] V. Yordanova and B. Gips, ''Coverage path planning with track spacing adaptation for autonomous underwater vehicles,'' *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4774–4780, Jul. 2020, doi: 10.1109/LRA.2020.3003886.

[21] B. Sun, D. Zhu, C. Tian, and C. Luo, "Complete coverage autonomous underwater vehicles path planning based on glasius bio-inspired neural network algorithm for discrete and centralized programming," *IEEE Trans. Cognit. Develop. Syst.*, vol. 11, no. 1, pp. 73–84, Mar. 2019, doi: 10.1109/TCDS.2018.2810235.

[22] R. Miao, S. Pang, D. Jiang, and Z. Dong, "Complete coverage path planning for autonomous marine vehicle used in multi-bay areas," *J. Acta Geodaetica et Cartographica Sinica*, vol. 48, no. 2, pp. 256–264, Feb. 2019, doi: 10.11947/j.AGCS.2019.20180385.

[23] J. Li, J. Zhang, L. Yang, C. Huang, and B. Zhang, "Target search of multiple autonomous underwater vehicles in an unknown environment," *hebgcdxxb*, vol. 40, no. 12, pp. 1951–1957 and 1972, Dec. 2019, doi: 10.11990/jheu.201808094.

[24] O. Aichholzer, C. Huemer, S. Kappes, B. Speckmann, and C. D. Toth, "Decompositions, partitions, and coverings with convex polygons and pseudo-triangles," *Graph Combinatoring*, vol. 23, no. 5, pp. 481–507, Oct. 2007, doi: 10.1007/11821069_8.

[25] Y.-L. Chang and I.-L. Tseng, "A parallel dual-scanline algorithm for partitioning parameterized 45-degree polygons," *ACM Trans. Design Autom. Electron. Syst.*, vol. 18, no. 4, pp. 1–18, Oct. 2013, doi: 10.1145/2505015.

[26] M. S. Floater and T. Lyche, "Divided differences of inverse functions and partitions of a convex polygon," *Math. Comput.*, vol. 77, no. 264, pp. 2295–2308, Jun. 2008, doi: 10.1090/S0025-5718-08-02144-3.

[27] Y. Q. Lu, Q. Su, and J. Kawa, "An innovative Steiner tree based approach for polygon partitioning," in *Proc. Asia South Pacific Design Automat. Conf.*, Mar. 2008, pp. 293–298, doi: 10.1109/ASPDAC.2008.4483974.

[28] R. Nandakumar and N. Ramana Rao, "Fair partitions of polygons: An elementary introduction," *Proc. Math. Sci.*, vol. 122, no. 3, pp. 459–467, Aug. 2012, doi: 10.1007/s12044-012-0076-5.

[29] X. Z. Wei, A. Joneja, and D. M. Mount, "Optimal uniformly monotone partitioning of polygons with holes," *Comput. Aided Des.*, vol. 44, no. 12, pp. 1235–1252, Dec. 2012, doi: 10.1016/j.cad.2012.06.005.

[30] C. Freeman, "Multiple methods beyond triangulation: Collage as a methodological framework in geography," *Geografiska Annaler, B, Hum. Geography*, vol. 102, pp. 1–13, Aug. 2020, doi: 10.1080/04353684.2020.1807383.

[31] V. An, Z. Qu, F. Crosby, R. Roberts, and V. An, "A triangulation-based coverage path planning," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 6, pp. 2157–2169, Jun. 2020, doi: 10.1109/TSMC.2018.2806840.

[32] A. Dumitrescu and C. D. Tóth, "Convex polygons in geometric triangulations," *Combinatorics, Probabability Comput.*, vol. 26, no. 5, pp. 641–659, Sep. 2017, doi: 10.1017/S0963548317000141.

[33] A. Asinowski, C. Krattenthaler, and T. Mansour, "Counting triangulations of some classes of subdivided convex polygons," *Eur. J. Combinatories*, vol. 62, pp. 92–114, May 2017, doi: 10.1016/j.ejc.2016.12.002.

[34] J. Yu and Z. L. Tang, "Generating strictly binary trees at random based on convex polygon triangulations," *Int. J. Comput. Math.*, vol. 93, no. 3, pp. 445–452, Mar. 2016, doi: 10.1080/00207160.2014.1002780.

[35] K. J. Böröczky, M. Matolcsi, I. Z. Ruzsa, F. Santos, and O. Serra, "Triangulations and a discrete Brunn–Minkowski inequality in the plane," *Discrete Comput. Geometry*, vol. 64, no. 2, pp. 396–426, Sep. 2020, doi: 10.1007/s00454-019-00131-9.

[36] Y.-I. Tseng, I.-L. Tseng, T. H. Huang, and A. Postula, "Fast partitioning of parameterized 45-degree polygons into parameterized trapezoids," in *Proc. IEEE 11th Int. New Circuits Syst. Conf. (NEWCAS)*, Paris, France, Jun. 2013, pp. 1–4, doi: 10.1109/NEWCAS.2013.6573570.

[37] F. Li and R. Klette, "Decomposing a simple polygon into trapezoids," in *Proc. Int. Conf. CIAP*, Vienna, Austria, Aug. 2007, pp. 726–733, doi: 10.1007/978-3-540-74272-2_90.

[38] B. Zalik, A. Jezernik, and K. R. Žalik, "Polygon trapezoidation by sets of open trapezoids," *Comput. Graph-Uk*, vol. 27, no. 5, pp. 791–800, Oct. 2003, doi: 10.1016/S0097-8493(03)00151-1.

[39] Z. Li, Z. Zhang, H. Liu, and L. Yang, "A new path planning method based on concave polygon convex decomposition and artificial bee colony algorithm," *Int. J. Adv. Robotic Syst.*, vol. 17, no. 1, Jan. 2020, Art. no. 172988141989478, doi: 10.1177/1729881419894787.

[40] P. S. Stanimirović, P. V. Krtolica, M. H. Saračević, and S. H. Mašović, "Decomposition of catalan numbers and convex polygon triangulations," *Int. J. Comput. Math.*, vol. 91, no. 6, pp. 1315–1328, Jun. 2014, doi: 10.1080/00207160.2013.837894.

[41] F. Assous and M. Michaeli, "Hodge decomposition to solve singular static Maxwell's equations in a non-convex polygon," *Appl. Numer. Math.*, vol. 60, no. 4, pp. 432–441, Apr. 2010, doi: 10.1016/j.apnum.2009.09.004.

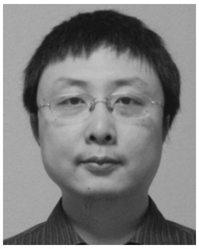
[42] S. Hert and V. Lumelsky, "Polygon area decomposition for multiple-robot workspace division," *Int. J. Comput. Geometry Appl.*, vol. 8, no. 4, pp. 437–466, Aug. 1998, doi: 10.1142/S0218195998000230.

[43] D. Adjiashvili and D. Peleg, "Equal-area locus-based convex polygon decomposition," *Theor. Comput. Sci.*, vol. 411, nos. 14–15, pp. 1648–1667, Mar. 2010, doi: 10.1016/j.tcs.2010.01.012.

[44] P. Dulio, "Convex decomposition of U-polygons," *Theor. Comput. Sci.*, vol. 406, nos. 1–2, pp. 80–89, Oct. 2008, doi: 10.1016/j.tcs.2008.06.008.

SHENGWEI XING was born in Tianjin, China, in 1982. He received the degree in marine technology, the M.S. degree in traffic information engineering and control, and the Ph.D. degree in traffic information engineering and control from Dalian Maritime University, Dalian, China, in 2005, 2008, and 2012, respectively.

He is currently a Lecturer with the Navigation College, Dalian Maritime University. His research interests include navigation science and technology, maritime search and rescue, electronic chart display and information system (ECDIS), geographic information system (GIS), and intelligent transportation system (ITS).



RENDA WANG was born in Liaoning, China, in 1984. He received the Ph.D. degree in traffic information engineering and control from Dalian Maritime University, Dalian, China, in 2014.

He is currently a Lecturer with the Navigation College, Dalian Maritime University. His research interests include navigation simulation systems and electronic chart display and information system (ECDIS).

GANG HUANG was born in Hebei, China, in 1981. He received the degree in marine technology and the M.S. degree in navigational science and technology from Dalian Maritime University, Dalian, China, in 2006 and 2008, respectively.

He is currently an Associate Professor with Dalian Maritime University. His research interests include navigation science and technology, cargo transportation by sea, ship intact stability, and marine conventions.