Multi-UAVs Area Decomposition and Coverage Based on Complete Region Coverage

Yue Wu¹, Jiang Zhu¹ and Kai Gao¹*  
¹School of Electronic Science, National University of Defense Technology, Changsha,  
China  
*Corresponding author e-mail: gaokai000@hotmail.com

Abstract. Using unmanned aerial vehicles (UAVs) to collect data from sensors can improve the performance of the sensor network. If the locations of the sensors are unknown or the mobility of sensors is unpredictable, the UAVs should cover the area where the sensors are located. Area coverage involves two stages, including area decomposition into subareas and path planning inside the subareas. This paper proposes a multi-UAVs area decomposition coverage method based on complete region coverage, referred to ADCCRC. First, the target area is transformed into honeycomb and the centers of the cells constitute a connected graph, so area decomposition is transformed into the problem of dividing the connected graph into sub-connected graphs. Then, find a Hamiltonian cycle inside the sub-connected graph as each UAV’s path. The algorithm also has the advantage that the subarea can be combined with UAVs’ capabilities assessment.

1. Introduction
Sensors are widely deployed in various scenarios to collect data. The sensors usually form a wireless sensor network (WSNs), and by means of self-organization, wireless sensor networks can transmit data to the destination [1]. Because the sensors are usually small and the battery duration is not long enough, if the sensors are disturbed by external environment or the batteries of sensors are exhausted, the data transmission of WSN will be severely affected. Owing to flexibility and small size, UAVs can be used to collect data from sensors, which will effectively extend the life of the sensors network [2].

There are three different classification methods for UAVs to collect sensors’ data. First, according to the number of UAVs, existing researches can be divided into single UAV scenario and multi-UAVs scenario. Second, according to different targets, existing researches are divided into region coverage and point-to-point coverage [3, 4]. The latter means finding the best path to cover all the nodes when position of nodes is known by UAVs. Third, if the number of UAVs is large, UAVs can cover all targets without moving. This situation is called static data collection [5-8], else it’s called as dynamic data collection.

This paper is devoted to studying multi-UAVs’ collaborative area coverage method. Current researches mainly focus on area decomposition into subareas and planning path inside the subareas.

In [9], Li decomposed the concave region into a convex sub-region, and determined the flight direction on the basis of the width of the convex polygon. In order to avoid unnecessary repetitive movement, some adjacent sub-areas were merged. In [3], the cost model UAVs’ mission execution was established and a new approach relative to UAVs’ capabilities assessment was proposed using hierarchical fuzzy inference. Literature [10] believed that any polygon could be circumscribed as a
rectangle. Considering the UAVs’ minimum turning radius, multi-UAVs performed Zigzag cruise in parallel, improving the traditional Zigzag cruise. In [11], the idea of polygon region segmentation was adopted. In [12], the area was decomposed using sweeping technique. For path inside the subareas, to minimize the number of UAV turns, a new method was used to generating optimal number of lanes. In [13], UAVs’ the parallel search strategy was analyzed in detail, and the arbitrary convex polygon was segmented according to the initial position and UAVs’ subarea. In [14], the formation coverage search method applied to moving targets was proposed. The researchers of [15] argued that area coverage problem of dividing the target area into multiple UAVs could be considered as an MTSP (multiple traveling salesman problem). The clustering and graph Method (CGM), which was originally used for image segmentation and compression, was used to area decomposition and correction. In [16], the influence of information fusion and communication delay on multi-UAVs’ collaborative search was discussed, which is also a hot topic in recent years. Similarly, a decentralized local algorithm to control UAVs’ the mobility was proposed [17].

In this paper, the ultimate goal of UAVs’ area coverage is to achieve data collection and transmission. To reduce delay in the process of UAVs collecting data, it is necessary to adopt an efficient method for UAVs’ area decomposition and path planning inside subareas. Therefore, this paper proposes a multi-UAV area cooperative area coverage method, that is, multi-UAVs area decomposition and coverage based on complete region coverage, referred to as ADCCRC algorithm. First, the target area is transformed into honeycomb and the center points of the cells constitute a connected graph, so area decomposition is transformed into the problem of dividing the connected graph into sub-connected graphs. Then, in order to minimize the sensors’ maximum waiting time in the process of data transmission and the time that UAVs complete a coverage for subareas, it is excellent to finding the Hamiltonian cycle [18] as the path for each UAV in theirs corresponding sub-connected graph.

The article structure consists of 5 parts. In Section 2, we describe complete region coverage and Hamiltonian cycle which are the basis of ADCCRC algorithm. The ADCCRC algorithm is introduced in detail in Section 3. Section 4 presents the simulation and we conclude the paper in Section 5.

2. Complete Region Coverage and Hamiltonian Cycle

In the multi-UAVs collaborative area coverage problem, subarea partition is a difficult problem. Because it is difficult to define dividing line between subareas. This chapter introduces the complete region coverage and Hamiltonian cycle, which are the basis to achieve the subarea partitioning and path planning. For the convenience of analysis, record the target area as rectangular area $W$, where the length and width of the rectangle is $xw \times yw$, and assume the UAVs’ coverage radius is $r$.

2.1. Complete Regional Coverage

The best way to cover a given area with circles of radius $r$ is to place the centers of the circles on an equilateral triangle network, i.e., to circumscribe the circles about the hexagons of a regular hexagon network or honeycomb [19], as shown in Fig. 1. At this time, the proportion of overlapping is the smallest, meaning that the number of coverage circles is the least, so the cruise distance of the UAVs can also be the shortest. Accordingly, placement pattern of the circles to achieve the minimum number was proposed in [20]. The circles with radius $r$ in area $W$ are called r-disks, and their centers are called r-centers. R-strips is a vertical line composed of r-centers. The minimum spacing between two r-centers is $\sqrt{3}r$ and the spacing between two adjacent r-strips is $1.5r$.

Figure 1. Placing the circles on an equilateral triangle and hexagon network
By drawing on and improving the method in [20], in order to completely cover the target area with minimum number of disks, the number of disks in each row, odd column and even column, \( m, n1, n2 \), is calculated as:

\[
m = \begin{cases} 
\text{Int}(\frac{xw - r}{1.5r}) + 1, & \text{if } \text{rem}(\frac{xw - r}{1.5r}) = 1.5r \\
\text{Int}(\frac{xw - r}{1.5r}) + 2, & \text{if else}
\end{cases}
\]  \hspace{1cm} (1)

\[
n1 = \text{Int}(\frac{yw}{\sqrt{3}r} + \frac{\sqrt{3}}{2}) + 2
\]  \hspace{1cm} (2)

\[
n2 = \begin{cases} 
n1, & \text{if } \text{rem}(\frac{yw}{\sqrt{3}r}) \leq 0.5 \\
n1 - 1, & \text{if else}
\end{cases}
\]  \hspace{1cm} (3)

Where \( \text{Int}(x) \) is the integer part of \( x \), and \( \text{rem}(x) = x - \text{Int}(x) \). The location of r-center in the \( k \text{th} \) row and \( l \text{th} \) column disk is:

\[
[x^{kl}, y^{kl}] = \begin{cases} 
[(1.5l - 1)r, (k - 1)\sqrt{3}r], & \text{if } l \text{ is odd} \\
[(1.5l - 1)r, (k - 1)\sqrt{3}r + \frac{\sqrt{3}}{2}r], & \text{if } l \text{ is even}
\end{cases}
\]  \hspace{1cm} (4)

We place r-disks to completely cover the rectangular area \( W \) and number the r-centers in a zigzag shape, as is shown Fig. 2.

![Figure 2. Placing the r-disks to completely cover the rectangular area](image)

If any two r-centers with spacing of \( \sqrt{3}r \) are considered as connected points, the undirected graph \( G = (V, E) \) shown in Fig. 3 and the corresponding connected matrix \( D \) can be obtained, where \( V = \{v_1, v_2, \ldots, v_n\} \) is the \( n \) r-centers, and \( E = \{v_i,v_j \mid v_i,v_j \in V, \sqrt{3}r \leq \sqrt{3}r \} \) is the edges in \( G \). These r-centers are called vertexes in \( G \). Because the UAVs’ coverage radius is exactly \( r \), the UAVs can achieve complete coverage of the area \( W \) as long as the UAVs go through all the vertexes in \( G \).

2.2. Hamiltonian Cycle

The spacing between two adjacent vertexes in \( G \) is minimal. If there are \( k \) vertexes in a UAV’s subarea, then if and only if when there is a Hamiltonian path in these \( k \) vertexes, the shortest path covering \( k \) vertexes can be gotten and its length is \( (k - 1)\cdot \sqrt{3}r \). Further, if these \( k \) vertexes assigned to this UAV contains a Hamiltonian cycle, the sensors’ maximum waiting time between two data transmission will be minimized, that is \( k\sqrt{3}r / v \), where \( v \) is the UAVs’ flight speed. Therefore, if there exists a Hamiltonian cycle in each subgraph, the UAVs fleet’s overall time to complete a cruise and sensors’ waiting time between two data transmissions can be as short as possible.
As is known to all, the necessary and sufficient conditions for the Hamiltonian cycle of the ordinary graph is one of the unsolved problems in the graph theory.

**Theorem 1.** \( \{v_1, \ldots, v_i, w, v_j, \ldots, v_N \} \) is a Hamiltonian cycle in the undirected graph \( H = (V, E) \), if \( w \notin H \) is a vertex respectively adjacent to a pair of adjacent vertexes \( v_i, v_j \) in \( H \) \( (v_i, v_j \notin E) \), then the new graph \( H_{\text{new}} \) composed of \( H \) and \( w \) contains a Hamiltonian loop, namely \( \{v_1, \ldots, v_i, w, v_j, \ldots, v_N\} \).

**Proof.** Assume that \( \{v_1, \ldots, v_i, w, v_j, \ldots, v_N\} \) is not a Hamiltonian cycle in the undirected graph \( H_{\text{new}} \). \( wv_i, wv_j \) and \( v_i v_j \) are edges in \( H_{\text{new}} \), so \( \{v_1, \ldots, v_i, v_j, \ldots, v_N\} \) in the subgraph of \( H_{\text{new}} \) is not cycle, implying the hypothesis is wrong. Therefore theorem 1 is proved.

![Figure 3. Connected graph of r-centers](image)

Combining the connected graph \( G \) generated in the chapter 2 with Theorem 1, the inference 1 is obtained and further the inference 2 is obtained.

**Inference 1.** When the number of r-centers on each r-strip in the undirected graph \( G \) are bigger than 1 and the number of r-strips is bigger than 1, \( G \) contains a Hamiltonian cycle. And the graph composed of any two adjacent r-strips also contains a Hamiltonian loop, and the graph consisting of every two adjacent r-strips also contains a Hamiltonian cycle.

**Definition 1.** Suppose that the connected component of the undirected graph \( H \) is 1 after the isolated vertexes (IVs) in \( H \) are removed, and \( H_0 \) is a subgraph by removing IVs and pending vertexes (PVs) in \( H \). Define the generalized cut vertexes (GCVs) in \( H \) is the cut vertexes in \( H_0 \).

**Inference 2.** \( G_0 \) is a subgraph with a connected component of 1 in \( G \), and \( G_0 \) does not contain any IV, PV, and GCV, therefore \( G_0 \) contains Hamiltonian cycle.

### 3. ADCCRC Algorithm

Assume that \( U = \{U_1, U_2, \ldots, U_N\} \) represents UAVs fleet containing \( N \) UAVs. The main purpose of the ADCCRC algorithm is to assign vertexes set \( S_{a_i} = \{v_{i1}, v_{i2}, \ldots, v_{in}\} \subset V \) to \( U_i \)'s subarea and plan path \( P_i = \{p_{i1}, p_{i2}, \ldots, p_{in}\} \) inside \( U_i \)'s subarea (where \( P_i \) is an ordered series of all elements in \( S_{a_i} \)), which involves three stages: compute the number of vertexes in each subarea, assign vertexes to the subarea and plan path inside each subarea. The general idea is shown in pseudocode, where \( C = \{c_1, c_2, \ldots, c_N\} \) is UAVs’ capabilities assessment.

Step 1 to 2 computes the number of vertexes in each UAV’s subgraph based on the UAVs’ capabilities assessment.

After initializing related variables in step 3, step 5 to 20 divides the sub-graphs one by one according to the Theorem 1 and Inference 2. First, step 5 assigns \( n_i \) vertexes to \( S_{a_i} \) one by one according to Theorem 1. The vertexes are assigned into \( S_{a_i} \) from \( RV \) must follow three criteria: (1) The first vertex takes precedence over the vertex with minimum serial number. (2) The second vertex is the vertex with minimum serial number among the neighbors of the first vertex. (3) Find the vertexes in \( RV \) adjacent to any pair of adjacent vertexes in \( S_{a_i} \), and among these vertexes, the vertex with smallest degree in \( G \).
and then with minimum serial number will be next vertex. The commonality of these three criteria is that when dividing a vertex into a sub-area, vertex with minimum serial number is preferred. Thus, when the remaining vertexes set \( RV \) remove IVs, its connected component is 1.

**ADCCRC Algorithm**

**Input:**
- \( U \), \( G \) and its connected matrix \( D \), \( C \)

**Output:**
- Subareas = \( \{S_1, S_2, \ldots, S_n\} \), \( P = \{P_1, P_2, \ldots, P_n\} \)

1. For \( i = 1 : N \) do
   - \( n_i = c_i / \sum_{r=1}^{n} c_i \)
   - End For
2. \( RV \leftarrow V, RD \leftarrow D, S_i \leftarrow \emptyset, P_i \leftarrow \emptyset, i = 1, 2, \ldots, n_i \)
3. For \( i = 1 : N \) do
4. Find \( n_i \) vertexes for \( S_i \) in \( RV \)
5. \( RV \leftarrow RV / S_i, RD \leftarrow \) the connected matrix of \( RV \)
6. \([I_s, P_s, GCs] \leftarrow IVs, PVs and GCVs in RV\)
7. \( RV_{new} \leftarrow \emptyset \), representing new vertexes reassigned to \( RV \)
8. While \( |I_s \cup P_s \cup GCs| \neq 0 \) do
   - If \( |I_s| \neq 0 \) then find a maximum \( v \notin S_i \) to make first vertex in \( I_s \) no longer a IV
   - ElseIf \( |P_s| \neq 0 \) then find a maximum \( v \notin S_i \) to make first vertex in \( P_s \) no longer a PV
   - Elseif \( |GCs| \neq 0 \) then find a maximum \( v \notin S_i \) to make first vertex in \( GCs \) no longer a GCV
   - \( RV_{new} \leftarrow RV_{old} \cup |v|, S_i \leftarrow S_i / RV_{old}, RV \leftarrow RV \cup RV_{new} \)
   - IPGVs \( \leftarrow IVs, PVs and GCVs in S_i \)
   - While \( |IPGVs| \neq 0 \) do
     - newRV \( \leftarrow \emptyset \), representing new vertexes reassigned to \( RV \)
     - Update the \( IPGVs \) of \( S_i \)
   - End While
   - Find \( (n_i - |S_i|) \) vertexes for \( S_i \) in \( (RV \cup S_i) / RV_{new} \)
   - Update \( [I_s, P_s, GCs] \), \( RV \leftarrow V / \sum_{i=1}^{n} S_i \)
9. End While
10. Find \( (n_i - |S_i|) \) vertexes for \( S_i \) in \( (RV \cup S_i) / RV_{new} \)
11. While \( |I_s \cup P_s \cup GCs| \neq 0 \) do
   - IPGVs \( \leftarrow IVs, PVs and GCVs in S_i \)
   - While \( |IPGVs| \neq 0 \) do
     - newRV \( \leftarrow \emptyset \), representing new vertexes reassigned to \( RV \)
     - Update the \( IPGVs \) of \( S_i \)
   - End While
12. End While
13. While \( |I_s \cup P_s \cup GCs| \neq 0 \) do
   - newRV \( \leftarrow \emptyset \), representing new vertexes reassigned to \( RV \)
   - Update the \( IPGVs \) of \( S_i \)
14. End While
15. End While
16. End For

Second, to guarantee the remaining vertexes set \( RV \) also contains Hamiltonian cycle according to Inference 2, step 6 to 21 check whether \( RV \) contains any IV, PV and GCV and repair these vertexes. Each “While” loop in step 9 to 21 only performs “repair” operation on the first vertex (the order of priority from high to low is respectively IV, PV, and GCV). Step 10 to 12 show how to repair these vertexes: (1) If there exists any IV, find a \( v \notin S_i \) with maximum serial number and put \( v \) back into \( RV \) so that the first IV is no longer an IV. (2) If there is no IV but there exists any PV, find a \( v \notin S_i \) with maximum serial number and put \( v \) back into the \( RV \) so that the first PV is no longer a PV. (3) If there only exist GCVs, find a \( v \notin S_i \) with maximum serial number and put \( v \) back into the \( RV \) so that the first GCV is no longer a GCV.

Third, since step 10 to 12 may lead to IVs, PVs or GCVs in \( S_i \), step 14 to 18 remove all these three kinds of vertexes in \( S_i \) to \( RV \) till all these three kinds of vertexes disappear in \( S_i \). At this time, the number of vertexes in \( S_i \) is smaller than \( n_i \). Similar to step 5, step 19 refills the vertexes for \( S_i \) from the remaining vertexes set \( RV \) but all the vertexes that have been deleted from \( S_i \) previously can no longer be selected. After updating the IVs, PVs and GCVs in \( RV \) again, repeat step 9 to 21 to repair these three kinds of vertexes until they all disappear in \( RV \).
Figure 4. (a) Devide nine vertexes to the second subarea $Sa_2$, there is a PV in $RV$. (b) Repair the PV in $RV$ and a PV appears in $Sa_2$. (c) Move the PV in $Sa_2$ into $RV$. (d) Reassign missing vertexes for $Sa_2$; (e) Complete the division of all subareas. (f) Plan the path for each subarea.

Reference to Theorem 1, step 22 to 26 plan path for $U_i$ by find a Hamiltonian cycle in subgraph composed of vertexes in $Sa_i$. First, take any pair of adjacent vertexes in $Sa$ as the first two vertexes of the path. Assuming that the planned path $P_i$ is $\{v_1, v_2, \ldots, v_n\}$. Any vertex $v \in Sa / P_i$ adjacent to a pair of adjacent vertexes $\{p_a, p_b\}$ on the path $P_i$ is found, and then inserting $v$ between $p_a$ and $p_b$, namely, the path inserting $v$ is $P = \{v, v_1, v_2, \ldots, v_n\}$.

Fig. 4 is an example of ADCCRC algorithm.

4. Conclusion

4.1. Simulation settings

The performance of ADCCRC algorithm is assessed using MATLAB. We have considered 5, 8, 11, 14, 17, 20, 23, 26 and 29 UAVs respectively in the rectangular area $W$ of which length and width is $200 \times 100 km$. UAVs’ flight speed is $v = 0.1 km / s$ and coverage radius is $r = 1 km$.
order to simplify the comparison with other algorithms, assuming that all UAVs’ capabilities assessment are equal. In addition, the UAVs’ turning angle is not considered in the algorithm.

The ADCCRC algorithm is compared with Zigzag algorithm [10] and CGM algorithm [15]. CGM algorithm plans closed paths but Zigzag algorithm doesn’t. Therefore, assume that the UAVs returns along the original path in the process of covering the area when Zigzag algorithm is called.

We have chosen two below mentioned metrics for the evaluation of ADCCRC algorithm:

- Maximum search time and minimum search time: Search time means the time that one UAV takes to complete a coverage on its subarea. Due to the shortage of each algorithm, the size of the subarea allocated to each UAV is different even if all UAVs’ capabilities assessment are equal. So there exists maximum search time and minimum search time among the UAVs fleet. The path planning of the CGM algorithm is random in some degree, so only the minimum search time is calculated there.

- The sensors’ maximum waiting time: The sensors in the target area can transmit the data to the UAV when they are within the coverage of a UAV. If the sensors need to continuously transmit data for a period of time, the sensors’ waiting time from this data transmission to next data transmission varies according to the path planning of the UAV and area size. This paper takes the sensors’ maximum waiting time in the area as a metric. If and only if UAVs’ path in the sub-area is a loop, the maximum waiting time of the sensors is the smallest.

![Figure 5](image)

**Figure 5.** (a) Maximum search time and minimum search time for each UAV in the UAV fleet to complete a search; (b) The sensors’ maximum waiting time from this data transmission to next data transmission.

### 4.2. Simulation results

Figure 5 above shows the simulation results of these three algorithms. Maximum search time and minimum search time are as shown in Figure 5. (a). It is obvious that calling ADCCRC and Zigzag algorithm saves at least half of the searching time compared to CGM. And the searching time of calling Zigzag is a little shorter than ADCCRC, but the difference between maximum search time and minimum search time of calling Zigzag algorithm is a little larger than ADCCRC algorithm, which means the searching time of calling Zigzag is unstable. Fig. 5(b) shows the sensors’ maximum waiting time of calling ADCCRC algorithm is much less than calling the other two algorithms. In a word, the performance of ADCCRC algorithm is good.

### 5. Conclusion

In this paper, we propose a multi-UAVs area decomposition coverage method based on complete region coverage, referred to as ADCCRC algorithm. First, the target rectangular area is transformed into connected graph, so area decomposition is transformed into the problem of dividing the connected graph into sub-connected graphs. When dividing the subgraphs, ensure that there are Hamiltonian
cycles in the subgraphs and regard them as the UAVs’ paths inside subareas. Simulation analysis for ADCCRC is performed using MATLAB, where performance of ADCCRC is evaluated against CGM algorithm and Zigzag algorithm. The analysis shows that ADCCRC has a little longer searching time than Zigzag. However, ADCCRC outperforms CGM and Zigzag in the sensors’ maximum waiting time.

References
[1] F. Y. Ren, H. Huang, C. Lin, Wireless sensor networks, Journal of Software. 14.14 (2003) 1513 - 1525.
[2] Zafar, Muhammad Aamir, et al, Design and development of effective manual control system for unmanned air vehicle, International Conference on Computer Research and Development IEEE, 2011, pp. 349-353.
[3] H. Peng, L. C. Shen, X. H. Huo, Research on Multiple UAV Cooperative Area Coverage Searching, Journal of System Simulation. 19.11 (2007) 2472-2476. (in Chinese)
[4] M. Wan, D. Zhong, W. K. Chu, Turning path planning for UAV in region coverage with scanline. Systems Engineering & Electronics. 36.9 (2014) 1750-1754. (in Chinese)
[5] Mozaffari, Mohammad, et al, Efficient Deployment of Multiple Unmanned Aerial Vehicles for Optimal Wireless Coverage, IEEE Communications Letters. 20.8 (2018) 1647-1650.
[6] Srinivas, Anand, G. Zussman, E. Modiano, Construction and Maintenance of Wireless Mobile Backbone Networks. IEEE/ACM Transactions on Networking. 17.1 (2009) 239-252.
[7] Lyu, Jiangbin, et al, Placement Optimization of UAV-Mounted Mobile Base Stations, IEEE Communications Letters. 21.3 (2017) 604-607.
[8] W. T. Yuan, Y. G. Li, Deployment Algorithm for UAV Network, Computer Knowledge & Technology (2017).
[9] Y. Li, H. Chen, J. E. Meng, et al, Coverage path planning for UAVs based on enhanced exact cellular decomposition method. Mechatronics 21.5 (2011) 876-885.
[10] Q. P. Wu, S. L. Zhou, G. Y. Yin, et al, Improvement of Multi-UAV Cooperative Coverage Searching Method. Electronics Optics & Control (2016).
[11] Y. Guo, Z. Qu, Coverage control for a mobile robot patrolling a dynamic and uncertain environment, Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on IEEE, Vol.6 (2004) 4899-4903
[12] J. F. Araujo, P. B. Sujit, J. B. Sousa, Multiple UAV area decomposition and coverage, IEEE, (2013) 30-37.
[13] S. N. Yu, R. Zhou, J. Xia, et al, Decomposition and coverage of multi-UAV cooperative search area, Journal of Beijing University of Aeronautics & Astronautics. 41.1 (2015) 167-173.
[14] Y. B. Xuan, C. Q. Huang, W. C. Wu, Coverage search strategies for moving targets using multiple unmanned aerial vehicle teams, Systems Engineering & Electronics. 35.3 (2013) 539-544.
[15] S. Ališauskas, Area Allocation Algorithm for Multiple UAVs Area Coverage Based on Clustering and Graph Method, Ifac Papersonline. 48.9 (2015) 204-209.
[16] M. Mirzaei, B. W. Gordon, C. A. Rabbath, et al, Cooperative multi-UAV search problem with communication delay, AIAA Guidance, Navigation, and Control Conference, Toronto, 2010, pp.519-524.
[17] J. Schleich, A. Panchapakesan, P. Bouvry, UAV fleet area coverage with network connectivity constraint, ACM International Symposium on Mobility Management and Wireless Access ACM, (2013) 131-138.
[18] F. Harari, Graph theory, Shanghai Scientific & Technical Publishers, Shanghai, 1980, pp.77-80
[19] R. Kershner, The Number of Circles Covering a Set, American Journal of Mathematics. 61.3 (1939) 665-671.
[20] Y. Guo, Z. Qu, Coverage Control for A Mobile Robot Patrolling A Dynamic and Uncertain Environment, Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on IEEE, 2004, pp.4899-4903 Vol.6.