Journal of Information and Communication Engineering (JICE), 6 (1): 321-327, June 1, 2020
e-ISSN: 2432-5465, p-ISSN: 2186-9162
Applied Science and Computer Science Publications

Research on Robot Path Planning Based on Dijkstra and Ant Colony Optimization

Zhen Nie
School of Electrical and Electronic Engineering
Shanghai Institute of Technology
Shanghai, China
hbniezhen@foxmail.com

Huailin Zhao
School of Electrical and Electronic Engineering
Shanghai Institute of Technology
Shanghai, China
zhao_huailin@yahoo.com

Abstract: This paper studied on the path planning problem in known environments. According to Dijkstra algorithm and ant colony optimization (ACO), a hybrid algorithm to search the path was designed. Based on the environment model, constructed by using visual graph method, Dijkstra algorithm was used for initial path planning. Then the ACO was improved and used to optimize the initial path to minimize the path of the robot. The simulation on MATLAB showed that the path planning algorithm based on Dijkstra-ACO has higher efficiency of path search and good effect of path planning, and the algorithm is effective and feasible.

Keywords: Dijkstra Algorithm, Ant Colony Optimization (ACO), Path Planning

I. INTRODUCTION

The path planning [1-2] for mobile robots is a foreword topic of intelligent robots. Its task is to plan a safe collision-free path from the starting position to the target position. There are some evaluation criteria to achieve the task (e.g., the shortest distance, the best time, the lowest energy consumption, etc.). The robot path planning method can be divided into two situations. The process in a known environment called global path planning The process in an unknown environment is called local path planning. For the time being, domestic and foreign scholars have put forward many algorithms after a larger number of researches on the path planning. Dijkstra algorithm [3-5], A* algorithm [6-7], artificial potential field method [8-10], ant colony optimization algorithm [11-13], etc. are commonly used for path planning algorithms.

The Dijkstra algorithm is a greedy algorithm, which only considers the distance from the current node to the next node, and uses the method, traversal search, to find the shortest path. The obtained path has high reliability and good robustness, but the complexity is high and it is easy to fall into local optimum. Based on Dijkstra, the A* algorithm introduces a heuristic function to guide the path searching. However, the traditional A* algorithm gradually determines the next path by comparing the neighborhood heuristic function F . When the heuristic function value has multiple minimum values, the A* algorithm cannot guarantee the best solution for path searching. The artificial potential field method introduces a virtual field into the motion space. The target shows gravity to the robot and the obstacles show repulsion to the robot. The robot adjusts its motion state through the action of gravity and repulsion. However, the artificial potential field method cannot guarantee the optimal path and is prone to deadlock phenomenon. The ant colony optimization algorithm is an intelligent method of random search based on group foraging.

It has strong robustness and can well combines with other algorithms. However, it has a long searching time and easy to fall into local optimum.

In this paper, aiming at the problem of robot path planning. We designed a hybrid algorithm combined Dijkstra with ACO. First an environment model was built by visual graph method, and then the initial path planning is performed by Dijkstra algorithm. Last

Corresponding Author: Zhen Nie. School of Electrical and Electronic Engineering, Shanghai Institute of Technology, China. E-mail: hbniezhen@foxmail.com

optimizing the initial path by improved ACO to get the optimal mobile robot moving path. Finally, the feasibility of the hybrid algorithm is verified by MATLAB (Mathworks Inc., United States) simulation, and the shortest path of the mobile robot is found.

II. ENVIRONMENTAL MODELING

The construction of environment model [14-15] is the primary solution to be solved in the study of path planning, and it is also the basic condition to achieve path planning. The environmental model refers to a model that describes specific things in a complex environmental system in an abstract form. It generally including two models − mathematical models and graphical models. It is mainly used to describe the exact location of various space objects in the environment where the mobile robot is located, and to help mobile robots to search for an optimal path from the initiation node to the target node in the established space model with obstacles. At present, methods of environmental modeling for mobile robot path planning mainly include geometric feature graph, visual graph and raster graph [16-19].

In order to facilitate the research, in this paper the visual graph method was used to construct the environment model, including the initial position, target position, obstacles, etc. The environment model constructed in this paper is shown in Figure 1.

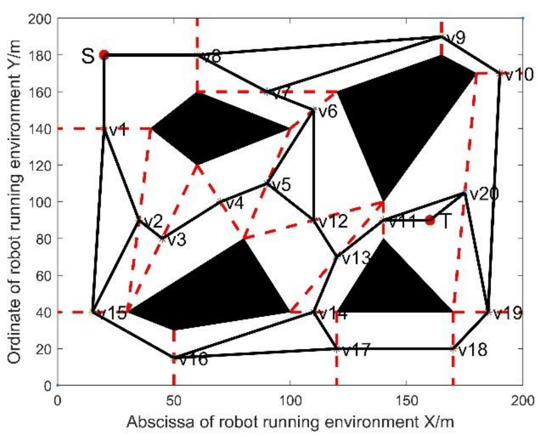

Fig.1. Environmental Model

Fig. 1 shows an environment model designed by visual graphic method in MATLAB. The size of the environment is $200m \times 200m$, where four black polygons represent four obstacles, and red dashed lines called free link lines[20], which indicate the lines between the boundary vertices of obstacles, the initiation node and the target node. The black solid lines indicate the path that the robot can go through. The nodes S and T indicate the initiation node and the target node of the robot respectively, and the serial numbers at the beginning of V indicate the midpoint of each link line.

The task of robots is to search for an effective path without collision quickly in the path planning, and to ensure that the path length is the shortest while avoiding obstacles effectively. The objective function can be expressed as follows:

$$L(S,T) = \min\left[L\left(N_c, m\right)\right] \tag{1}$$

Where $L(S,T)$ is the path length. N_c is the number of iterations. $L(N_c, m)$ is the path length that the m^{th} ant travels in the Nc^{th} iteration in the process of path planning.

III. INITILA PATH PLANNING

Dijkstra algorithm was designed by Dijkstra. E. W. in 1959. It can effectively solve the problem of the shortest path in graph theory. It has high reliability and robustness. It is often used to solve the problem of the shortest path in path planning. Therefore, Dijkstra algorithm is used in the initial path planning. So that the shortest path from the initiation node *S* to the target node *T* in the feasible path in the built environment model would be selected. The idea of initial path planning is as follows:

According to the feasible path in the environment model that constructed before, the distance matrix L between nodes in each link is constructed. The construction rule is that the distance between adjacent nodes is obtained by formula (2), and the distance between non-adjacent nodes is set to ∞ .

$$L\left(V_i, V_j\right) = \left\|V_i, V_j\right\| \tag{2}$$

Where V_i and V_j is the i^{th} and j^{th} midpoints.

Set up two sets *M* and *N*. The function of *M* is to record the vertices of the shortest path and the corresponding length. The function of *N* is to record the vertices which have not yet found the shortest path and the distance from the vertex to the initiation node *S*. Initialize the parameters and put all the nodes into the set *M* or *N* respectively $M = \{S\}$, $N = \{V_1, V_2, \cdots, V_n, T\}$. Finding node V_k from set N makes

$L(S,V_k) = \min\{L(S,V_j)\}$. Where S is the initiation node and V_k is the k^{th} node, $V_j \in N$. Placed the V_k into the set M, and update the set M and N after V_k is selected. At the same time, update the distance from each vertex in N to the initiation node S. The reason why the distance from each vertex to the initiation node S in set N is updated is that V_k is the node of the shortest path in the upper step. Therefore the distance from other nodes to the initiation node can be updated by V_k. Continuous selection of nodes and updating of sets until the set N is an empty set, the iteration is end, and the initial path planning is completed, the initial path can be obtained. As shown by the blue solid line in Figure 2 is the initial path achieved by Dijkstra.

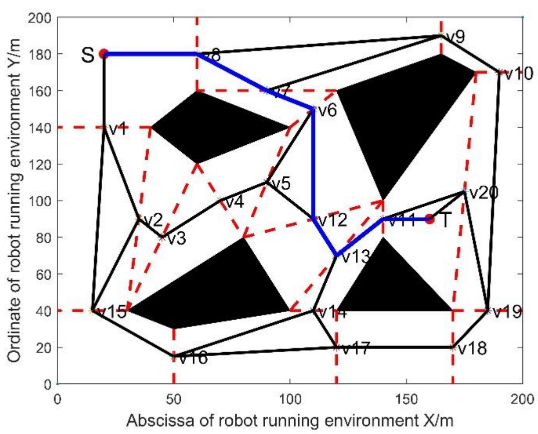

Fig.2. Initial Path.

IV. PATH OPTIMIZATION

According to Dijkstra algorithm, the initial robot path can be planned, but the path is not the most optimal. Next, ACO is used to optimize the path. A new optimal path can be obtained by using ACO. The purpose is to solve some optimal parameters $(\lambda_1, \lambda_2, \cdots, \lambda_n)$ on the link through which the initial path passes, so that the coordinates of each node satisfy the formula (3).

$$Q_i(\lambda_i) = Q_i^0 + (Q_i^1 - Q_i^0) \times \lambda_i \qquad (3)$$
$$\lambda_i \in [0,1], i = 1, 2, \cdots, n$$

Where Q_i^0 and Q_i^1 is two endpoint coordinates of the i^{th} link. λ_i is the proportional parameters of link.

A. Traditional ACO

The traditional ACO is an intelligent method of random search based on group foraging. It mainly uses the information transmission between ant colonies and solves the optimal path through positive feedback. The basic principle is that in the process of searching for food, the probability of next path selection is determined by pheromone concentration and heuristic information left on the path between ants. And then the path selection is determined by this probability. The formula of path transition probability is as follows:

$$P_{ij}^k = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum\limits_{s \in allowed_k} [\tau_{ir}(t)]^\alpha [\eta_{ir}(t)]^\beta} & s \in allowed_k \\ 0 & otherwise \end{cases} \qquad (4)$$

$$\eta_{ij}(t) = \frac{1}{d_{ij}} \qquad (5)$$

Where $allowed_k$ is the set of nodes that the k^{th} ant can choose next. $\tau_{ij}(t)$ is the pheromone concentration on the path from the current node to the next node at time t. α is the pheromone heuristic factor. β is the expectation heuristic factor. $\eta_{ij}(t)$ is the heuristic function on the path from the current node to the next node at time t. d_{ij} is the distance from the i^{th} node to the j^{th} node.

Since ants depend on pheromone concentration and heuristic information to select path, pheromone will gradually decrease with time. To avoid the influence of pheromone changes on node selection, all ants need to update and adjust pheromone according to formula (6) after completing a search.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t,t+1) \qquad (4)$$

$$\Delta\tau_{ij}(t,t+1) = \begin{cases} \dfrac{Q}{L_k} & ij \in L_k \\ 0 & otherwise \end{cases} \qquad (5)$$

Where ρ is the volatilization rate of pheromone. $\Delta\tau_{ij}(t,t+1)$ is the pheromone concentration increment. Q is the pheromone intensity and is a constant greater than zero. L_k is the path length of the k^{th} ant in this search.

B. Improved ACO

1) Improvement of Heuristic Function

The traditional heuristic function of ACO only considers the distance from the current node to the next node. It will reduce the calculation speed and the accuracy of the path search. In this paper, the idea of the valuation function of the A* algorithm is used to improve the calculation speed and path search accuracy. The valuation function is shown in formula (8).

$$f(n) = d_{ij} + \left(d_{jT} + \frac{1}{N_j} \right) \tag{8}$$

Where $f(n)$ is the valuation function of the current node. d_{jT} is the distance from the node *j* to the target node. N_j is the number of nodes that node *j* can select. It can be seen from equation (8) that the smaller the d_{jT} is, the better the orientation of the next node to the target node is, and the smaller the cost value is. The larger the N_j is, the more the subsequent node selection is, and the smaller the cost value is. However, considering the interference of obstacles on path planning, when the optional node is far away from the target node, the actual path length is much larger than the estimated value, so the estimated path length should be appropriately expanded. Therefore, a weighting coefficient $\mu(\mu \geq 1)$ is introduced, and the weighting coefficient will decrease as the decreases of d_{jT} until it is reduced to 1. The improved heuristic function is shown as the formula (9).

$$\eta_{ij}(t) = \frac{1}{f(n)} = \frac{1}{d_{ij} + \mu \cdot d_{jT} + \frac{1}{N_j}} \tag{9}$$

2) Improvement of Node Selection

In order to avoid ants falling into a local optimum situation in the process of moving, the obstacle avoidance strategy is adopted at the beginning of searching, and an obstacle avoidance factor γ is added. The probability formula of path selection is as follows:

$$P_{ij}^{k} = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}(t)\right]^{\beta} \gamma_{jb}{}^{\varepsilon}}{\sum\limits_{s \in allowed_k} \left[\tau_{ir}(t)\right]^{\alpha}\left[\eta_{ir}(t)\right]^{\beta}} & s \in allowed_k \\ 0 & otherwise \end{cases} \tag{10}$$

$$\gamma_{jb} = \frac{1}{d(j,b)} \tag{11}$$

Where ε is an obstacle avoidance coefficient that generally taking a positive number.

After adding the obstacle avoidance strategy, a random selection mechanism is used to select the nodes on the next link line, as shown in the equation (12).

$$j = \begin{cases} \arg\max\left\{\tau_{ij}(t)^{\alpha}\eta_{ij}(t)^{\beta}\gamma^{\varepsilon}\right\} & q \leq q_0 \\ p_{ij}^{k} & q > q_0 \end{cases} \tag{8}$$

$$q_0 = \delta \cdot \frac{N_c - N_m}{N_c} \tag{9}$$

Where N_c is the number of iterations. δ is the adjustment coefficient, $\delta \in [0.5,1]$. N_m is the current iterations

3) Improvement of Pheromone Updating

Aiming at the problem that local pheromone updating will reduce the convergence of the algorithm and global pheromone updating cannot guide the ant colony to find the optimal solution in time, the updating methods of these two pheromones are improved respectively.

a. Local Pheromone Updating

In the process of path searching, pheromone updating formula (14) is used to update the local pheromone whenever ant passes through a link line.

$$\tau_{ij}(t+1) = (1-\lambda)\tau_{ij}(t) + \lambda\tau_0 \tag{10}$$

Where τ_0 is the pheromones under initial conditions. λ is the volatilization rate of local pheromone.

b. Global Pheromone Updating

After the ant finishes a path search, there will generate two paths. The one is the best path and the other is the worst. The path nearer to the optimal solution is selected by these two solutions, and the global pheromones are updated. The pheromone concentration update formula is as follows:

$$\Delta\tau_{ij}(t,t+1) = \begin{cases} \dfrac{Q}{L_k} \cdot \dfrac{L_g - L_G}{L_G} \cdot \dfrac{L_g + L_b}{2} & ij \in L_k \\ 0 & otherwise \end{cases} \tag{11}$$

Where L_g is the length of the best path in this searching. L_b is the length of the worst path in this searching. L_G is the length of optimal path which has been searched.

V. PROCESS PATH PLANNING

As mentioned earlier, the path planning algorithm has two parts, initial path planning and path optimization. Firstly, the initial path planning is carried out according to Dijkstra algorithm. Then the improved ACO is used to optimize the initial path.

A. Initial Path Planning Process

According to the requirement of the shortest path of Dijkstra algorithm, the initial path is planned. The specific process is as follows:

Step 1: Building the environment model by visual graph. Constructing the distance matrix *L* between each link midpoint according to the feasible path in the model;

Step 2: Initialize the parameters and put all nodes into the sets *M* and *N* respectively;

Step 3: Selecting the shortest distance node *k* from the set *N*, and adding the node *k* to the set *M*. Meanwhile, removing the node *k* from the set *N*;

Step 4: Update the distance from each node of set *N* to the initiation node *S*, and judge $N = \varnothing$. If true, the algorithm ends. Otherwise, repeat step 3 until the set *N* is empty. The path from the initiation node *S* to the target node *T* can be obtained

B. Path Optimization Process

After the initial path planning is completed, using the improved ACO to optimize the initial path. The specific process is as follows:

Step 1: Initializing the parameters of improved ACO;

Step 2: Start the path searching, and select the next node according to the current node information and the next node selection principle;

Step 3: After the next node is selected, update the local pheromone on the path that the ant has just traveled;

Step 4: Judging whether the ant has reached the target node, if true, skip to next step, otherwise repeat the step 2;

Step 5: Searching the optimal path that is currently searched and update the global pheromone;

Step 6: Judging whether the number of iterations $N_m \le N_c$, if true, end the search, otherwise repeat the step 2.

VI. SIMULATION ANALYSIS

According to the path planning algorithm designed in the previous section, the simulation was carried out on the MATLAB. Firstly, the environment model was established by using the visual graph. The simulation environment model is $200m \times 200m$, including obstacles and initiation node and target node. Then the initial parameters of the improved ACO are configured. The parameters are given in Table 1.

TABLE I. INITIAL PARAMETERS OF THE IMPROVED ACO

Parameters	Numerical Value
The number of iterations: N_c	500
Number of ants: m	40
pheromone heuristic factor: α	1.2
expectation heuristic factor: β	3.5
volatilization rate of pheromone: ρ	0.6
pheromones under initial conditions: τ_0	1.5
volatilization rate of local pheromone: λ	0.8
pheromone intensity: Q	1
obstacle avoidance coefficient: ε	2.3
adjustment coefficient: δ	0.7

According to the designed algorithm, the optimal path of the mobile robot can be achieved by path planning in the established environment model. Figure 3 shows the path of the mobile robot. The blue solid line is the path obtained by Dijkstra algorithm. And the path length is 229.0611m. The green solid line is the optimal path obtained by the design algorithm. The path length is 173.8837m. The path is shortened by 24%. It can be seen that the algorithm designed in this paper can avoid obstacles very well and ensure the optimization of the path. Figure 4 is the iteration graph of the algorithm, in which the black solid line is the iteration graph obtained by the Dijkstra algorithm. And the black dotted line is the iteration graph obtained by the designed algorithm. The optimal path is obtained by 98 iterations. It can be seen that the algorithm designed in this paper has high efficiency and improves the efficiency of path planning.

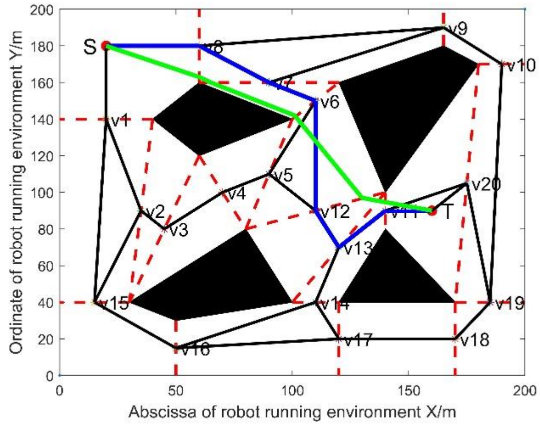

Fig.3 Path of Robot.

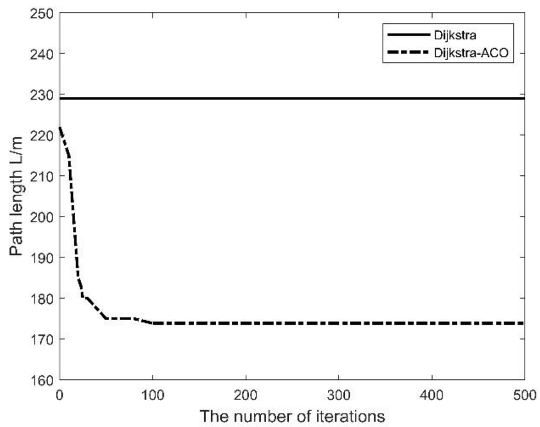

Fig.4 Iteration Graph of the Algorithm.

VII. CONCLUSION

In this paper, a hybrid algorithm based on Dijkstra algorithm and ACO is proposed to achieve the path planning in complex environments. It realizes the fast search of the path of robots and gets the shortest path. MATLAB simulation shows the superiority of the algorithm. It can be obtained that the algorithm can effectively avoid obstacles and achieve the optimal path planning. The length and quality of the obtained path are obviously superior than the path which is obtained by Dijkstra algorithm. The algorithm has better rapidity and convergence, and better improves the efficiency of path planning.

REFERENCES

[1] Huailin Zhao, Zheng Wu, Xiaoxing Wang. A Mul-ti-robot Rescuing System. The 2018 International Conference on Artificial Life and Robotics (ICAROB 2018), 2018, Japan, pp. 582-586;

[2] Huailin Zhao, Zhen Nie. Design and Analysis of Multi-robot grouping aggregation algorithm. Journal of Robotics, Networking and Artificial Life, 2019, Vol. 6, pp. 60-65.

[3] Chen Yalin, Zhuang Liyang, Zhu Longbiao, Shao Xiaojiang, Wang Heng. Research on path planning of parking system based on the improved Dijkstra algorithm. Modern Manufacturing Engineering. 2017, pp. 63-67;

[4] Wang Xu, Liu Yi, Li Guoyan. Path Planning for Mobile Robot Based on Improved Dijkstra Algorithm. Journal of Tianjin Chengjian University. 2018, Vol. 24, pp. 378-386;

[5] Yizhou Chen, Shifei Shen, Tao Chen, Rui Yang. Path Optimization Study for Vehicles Evacuation Based on Dijkstra algorithm. Procedia Engineering. 2014, pp. 159-165;

[6] Xue Fenghui, Zhou Ruitao. Using Dijkstra A* algorithm to implement ship navigation arithmetic. ship science and technology. 2017, Vol. 39, pp. 81-83;

[7] Li Siliang, Yuan Qingni, Hu Lai, Huang Xin, Pan Zhufeng. Research on Path Planning Based on Improved Human-m Computer Interaction A* Algorithm. Modular Machine Tool & Automatic Manufacturing Technique. 2019, pp. 38-44;

[8] Luo Qiang, Wang Haibao, Cui Xiaojin, He Jingchang. Autonomous Mobile Robot Path Planning Based on Improved Artificial Potential Method. Control Engineering of China. 2019, Vol. 26, pp. 1091-1098;

[9] Jin Wang, Xiaozhu Liu. Obstacle avoidance of mobile robots using modified artificial potential field algorithm. EURASIP Journal on Wireless Communications and Networking, 2019, Vol.2019 (1), pp. 1-19;

[10] Liang Xianxia, Liu Chaoying, Song Xueling, Zhang Yingkun. Research on Improved Artificial Potential Field Approach in Local Path Planning for Mobile Robot. Computer Simulation. 2018, pp. 291-361;

[11] Jiang Ming, Wang Fei, Ge Yuan, Sun Longlong. Research on path planning of mobile robot based on improved ant colony algorithm. Chinese Journal of Scientific Instrument. 2019, Vol. 40, pp. 113-121;

[12] Zhan Wei, Qu Junsuo, Lu Xin, Hou Leichao. Global path planning based on improved ant colony algorithm for mobile robot. Modern Electronics Technique. 2018, Vol. 41, pp. 170-173;

[13] Yan Jie, Qin Feizhou, Zhai Shuaihua. An Improved Ant Colony Algorithm for Mobile Robot Path Planning. Software Guide. 2019, Vol. 18, pp. 5-8;

[14] Huanpu Shuyun, Tang Shoufeng, Tong Ziyuan, Song Bin, Tong Minming. A survey of path planning methods for autonomous mobile robots. Software Guide. 2018, Vol. 17, pp. 1-5;

[15] Pan Ruoyu, Zhu Wei, Yang Shanlin. Metropolitan bus lines optimization and evaluation method based on Dijkstra-PD-ACO algorithm. Chinese journal of management science. 2015, Vol. 23, pp. 106-115;

[16] Yuan Shizhao, Li Jun. Summary of Research on Path Planning of Driverless Vehicles. Auto Engineer. 2019, pp. 11-14;

[17] Li Gen, Li Hang, Zhang Shuaiyang, Luo Qiuhui, Yang Hongdan. Optimal path planning and parameter analysis based

on ant colony algorithm. China Science paper. 2018, Vol. 13, pp. 1909-1914;

[18] Wang Hongjun, Xu Jun, Zhao Hui, Yue Youjun. Mobile robot path planning based on smoothing ant colony algorithm. Journal of Yanshan University. 2017, Vol. 41, pp. 278-282;

[19] Li Jun, Shu Zhibing, Wang Suzhou. Robot trajectory planning based on spline function and improved genetic algorithm. Manufacturing Technology & Machine Tool. 2017, pp. 91-95;

[20] Jiang Haiyang, Yan Zhaoru, Guo Qi. Path planning of robot based on improved genetic algorithm. Journal of natural science of Heilongjiang university. 2017, Vol. 34, pp. 601-607.