J Intell Robot Syst (2015) 77:391–409
DOI 10.1007/s10846-013-9901-z

Geometric Reinforcement Learning for Path Planning of UAVs

Baochang Zhang · Zhili Mao · Wanquan Liu · Jianzhuang Liu

Received: 17 August 2013 / Accepted: 11 September 2013 / Published online 18 October 2013

Abstract We proposed a new learning algorithm, named Geometric Reinforcement Learning (GRL), for path planning of Unmanned Aerial Vehicles (UAVs). The contributions of GRL are as: (1) GRL exploits a specific reward matrix, which is simple and efficient for path planning of multiple UAVs. The candidate points are selected from the region along the Geometric path from the current point to the target point. (2) The convergence of calculating the reward matrix is theoretically proven, and the path in terms of path length and risk measure can be calculated. (3) In GRL, the reward matrix is adaptively updated based on the Geometric distance and risk information shared by other UAVs. Extensive experimental results validate the effectiveness and feasibility of GRL on the navigation of UAVs.

Keywords Path planning · UAV · Geometric

B. Zhang (✉) · Z. Mao
Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University, Beijing, China
e-mail: bczhang@buaa.edu.cn

Z. Mao
e-mail: joelleo.mao@gmail.com

W. Liu
Department of Computing, Curtin University, Perth, WA 6102, Australia

J. Liu
Shenzhen Key Lab for CVPR, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Beijing, China

J. Liu
Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China

1 Introduction

Unmanned Aerial Vehicles (UAVs) in military fields has been given much attention over the past decade [1–5]. The increasing demand has brought into focus on several challenges associated with multiple UAV operation. Reducing UAV dependence on human pilots is a major concern. Currently, UAVs such as Predators require the full attention of two human operators. In order to successfully carry out a complex mission, these UAVs need to share information and to cooperate one another to improve the overall group performance. In the battle fields, UAVs are flying in a highly dynamic and challenging environment. They must react quickly to sudden changes both on the ground and in the air.

UAVs often fly in a complicated environment. Many threats, such as hills, trees, enemy's

UAVs, and enemy's aircrafts, can be fatal in causing the UAVs to crash. These threats can only be detected within a limit range from a single UAV; however, by sharing information with other UAVs, these threats can be detected over a longer distance. Furthermore, an effective path for navigation should be smooth, and provide an escape route, and must be computationally efficient. These are hot issues and attract much attention in recent years.

In previous work on path planning for a single UAV, Voronoi Graph Search [1] and Visibility Graph Search [6] are among the earliest algorithms, which have been proven to be effective only in a simple environment. They are not real-time, and also lead to fatal failure when the map information is not entirely available, such as when some obstacles are not detected. A_{2D} [7] is an efficient and real-time algorithm, and generally can find a valid route for UAVs. However, it cannot find a valid path in a locally complicated area, and have to design a path to escape the dangerous zone, particularly when the turning corner at the dangerous area is too large. Evolutionary Algorithms have been used as a viable candidate to effectively solve path planning problems and provide feasible solutions within a short time [8]. A Radial Basis Functions Artificial Neural Network (RBF-ANN) assisted Differential Evolution (DE) algorithm is used to design an off-line path planner for UAVs coordinated navigation in known static maritime environments. The Behavior Coordination and Virtual (BCV) goal method [9] proposes a real-time path planning approach based on the coordination of the global and local behaviors. This approach realizes the path planning by controlling the local behavior and the global behavior. A fuzzy logic controller (FLC) for controlling the local behavior is designed to achieve the threat-avoidance.

Different from single UAV, the path planning of multiple UAVs concentrates on the collaborative framework, collaborative strategies and consistency, etc. The Voronoi Graph Search [1] and the A* algorithm (or Dijkstra algorithm) [2] plan a global path for multiple UAVs to simultaneously reach the target in an exhaustive procedure. Researchers in [6] use the Dubins path to plan a global cooperative path for multiple UAVs to avoid collision, which suffers from sudden change in local region. In [10], the authors propose a path planning algorithm based on a map of the probability of threats, which can be built from priori surveillance data. In [11, 12], the researchers develop a novel hybrid model, and design consensus protocols for the management of information. They further synthesize local predictive controllers through a distributed, scalable and suboptimal neuro-dynamic programming (NDP) algorithm. An explicit feedback mechanism [13, 14], the so-called feedback based CRI (FBCRI), is embedded into optimal fuzzy reasoning methods to solve the path planning of multiple UAVs. By embedding virtual subgoal into the FBCRI based approach, a new cooperative path planning approach based on virtual subgoal (CPVS) is proposed for further solving the path planning problem. However, to the best of our knowledge, the real-time path planning on multiple UAVs and information sharing is not well studied.

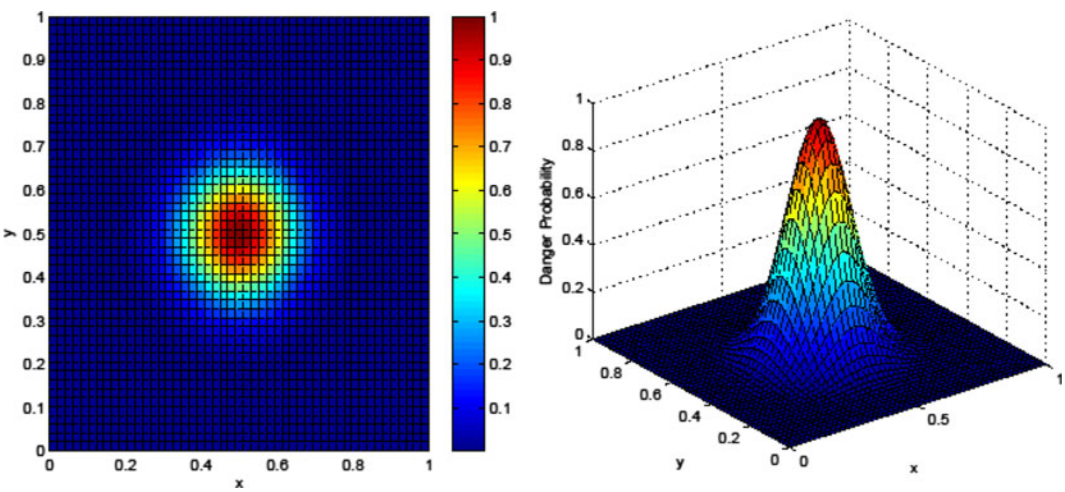
In this paper, we address the path planning of multiple UAVs from the perspective of reinforcement learning [15–18]. Q-Learning as a kind of classical reinforcement learning method is a traditional way to solve the path planning problem. The basic idea of Q-Learning is to obtain the optimal control strategy from the delayed rewards according to the observed state of the environment in a learning map and to make a control strategy to select the action to achieve the purpose. But the method is actually designed for the entire map of the environment known to planners. Q-Learning fails to use the geometric distance information which is a very valuable element for path planning when only partial information of the map is available. Moreover, for multiple UAVs, the shared information from other UAVs cannot be well exploited as there are a lot of unnecessary calculations to propagate from one point to others in Q-Learning. Also some special points such as the start and target points are not well considered. We propose a new algorithm, Geometric Reinforcement Learning, to utilize the geometric distance and risk information from detection sensors and other UAVs, and build a general path planning model. By dividing the map into a series of lattice, path planning for UAVs is formulated as the problem of the optimal path planning. A continuous threat function is used in the paper to simulate the real situation where UAVs fly. To reduce the complexity of calculation, we finely modulate the parameter to control the size of the map. Moreover, we generalize the algorithm to multiple UAVs by using the information shared from other UAVs, which provides an effective solution for the path planning and avoids local optimums. In our approach, the UAVs detect threatening objects in real-time and share the information with each other. Collisions are avoided by the virtual UAVs created from a UAV, which is considered as new obstacles for the other UAVs. Further we can change the reward matrix in terms of an exposure risk function. The target planner gets the final path according to all the known threat information and UAV's real-time position. Finally, Extensive experiments have shown that the proposed approach performs very well in terms of the path length and the integral risk measure.

The rest of the paper is organized as follows: Section 2 describes the UAV threat environment modeling. Sections 3 and 4 present the main components of the GCL algorithm. The extensive experimental results are given in Section 5, and Section 6 concludes the paper.

2 Modeling of Probabilistic Risk Exposure to Obstacle

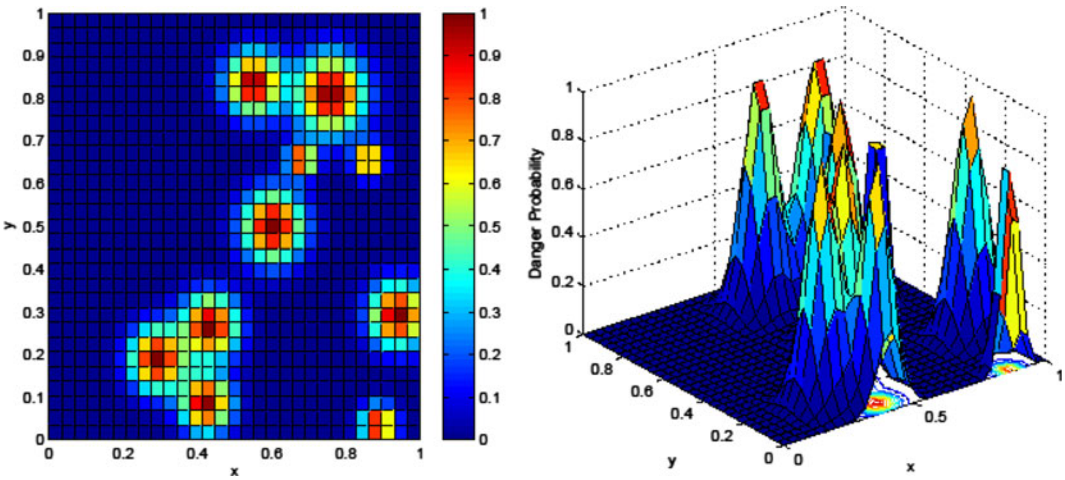
The environment of UAVs often fly in needs to be well simulated to evaluate performances of different kinds of path panning methods. Furthermore, UAVs are vulnerable to attack from the ground, or other UAVs. It is necessary for UAVs to keep a certain distance from regions of high risk to ensure a safe flying. So the measure of probabilistic risk exposure to obstacle can be seen as a continuous distribution function as shown in Figs. 1 and 2. For example, considering the case where an obstacle is at position (x_i, y_i), the measure of the risk is denoted by f_d, in which the parameters are related to the dimension of the planning space. In the two-dimensional space, f_d is represented as $f_i\ (x,y)$. In this paper, we consider it as a normal distribution:

$$f_i\,(x,y) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{\left(-\frac{d_i^2}{2\sigma_i}\right)},\ d_i = \sqrt{(x-x_i)^2 + (y-y_i)^2} \tag{1}$$



(a) 2D (b) 3D

Fig. 1 The probabilistic risk of exposure to obstacle in the normal distribution



(a) 2D (b) 3D

Fig. 2 The probabilistic risk of exposure to multiple obstacles

The probabilistic risk of the area where UAVs could not fly over can be represented as a very big value. Furthermore, when more than one obstacle exists on the map, the probabilistic risk at position (x, y) can be calculated as:

$$F(x, y) = 1 - \prod_{i=1}^{M} \left[1 - f_i(x, y)\right] \tag{2}$$

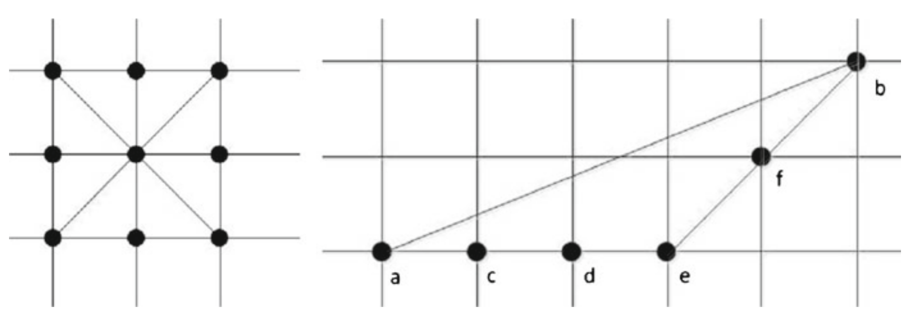
3 The Weight Matrix Based on the Geometric Distance and Integral Risk Measure

The weight matrix is very important for a path planning problem. In order to find the next point for the current point in a given path, we need to know the relationship or weight between any two points. The element in the weight matrix for path planning needs to contain both the distance and the risk information for any two points. In the following, we first discuss the state update method in the traditional Q-learning method.

3.1 Drawbacks of Q-Learning

In our study, we also try Q-Learning method in path planning of UAVs [18]. The decision in Q-Learning depends on the current state and the weight of the next states. In our simulation results, we find that the flight direction of a UAV is limited to eight directions as shown in Fig. 3a, as the next state is confined to the eight states of the neighborhood. So it affects the feasibility

Fig. 3 Illustration of Geodisc distance **a** Eight directions for Q-Learning from the central point to neighbors, **b** An example of the Geometric distance



(a) (b)

of the UAV flight path, and also the complexity of the trajectory of the UAVs. As shown in Fig. 3b, the geometric distance results in a more efficient path from position *a* to *b*, while the corresponding one in Q-learning has to pass *c, d, e, f*.

Moreover, it is reasonable that the reward matrix should contain both geometric distance and threatening object information to get the state transition. In Q-Learning, the weight of the reward at the target point have to be very large, and consequently the final path may be unreasonable. In this paper, we calculate the integral distance between every two points on the map, and further improve the reward matrix update mechanism.

3.2 The New Weight Matrix Update Scheme

In this paper, the discrete map is used to verify our method. It is well known that the discrete map is an effective method to reduce the calculation complexity. We create a complete graph, or the weight matrix, in which a weight is set to two points of the map is set to a weight. In Q-Learning, the action in each step is restricted to a set of actions decided by the state matrix. Generally the state matrix cannot be complicated, the further action in each step is in eight directions as shown in [18].

In this paper, the map is divided into $N \times N$ lattice ($N = 20$), and the weight for the path between any two points on the map is computed by the geometric distance and the integral risk measure:

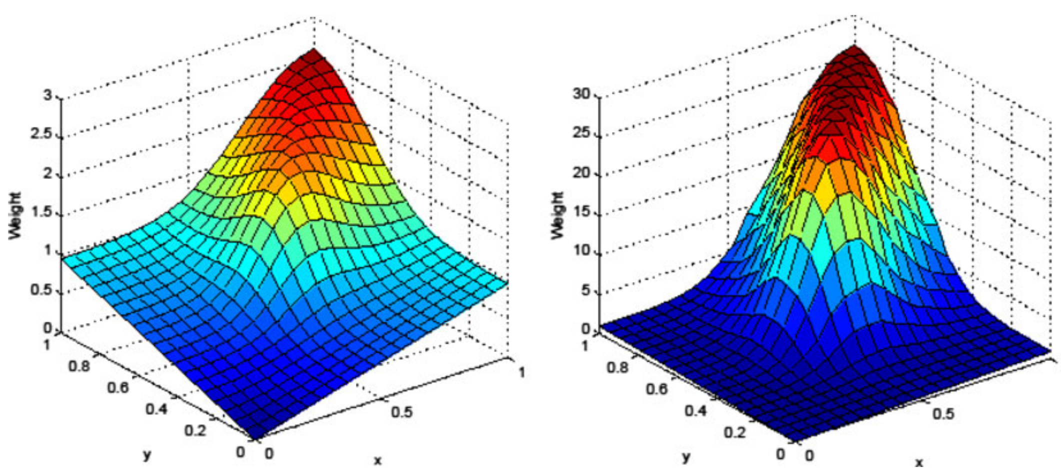
$$A_{p_1,p_2} = d_{p_1,p_2} + K \times \int_C F(x, y)\, ds \qquad (3)$$

Where C is the point set on the path from p_1 to p_2. dp_1,p_2 is the distance between p_1 and p_2, and K is the parameter related to the degree of threat, which affects the weight between two points. Considering a single threat as in Fig. 1, we visualize an example of threat at (0.5, 0.5) in the given map. As shown in Fig. 4, we illustrate an example of the weight values calculated based on Eq. 3 from (0,0) to other points when $K = 5$ and 100.

Considering that the calculation of A matrix consume much time, we improve the efficiency by the following method as:

We UAV is in the position p_{r_1} with p_T as the target point, and S_{map} as the point set in the map, S_{ab} is the point set in the rectangular with the diagonal line from the point a to the point b. The definition of $S_{P_T P_{r_1}}$ is similar to that of S_{ab}. The procedure is described as follows:

1. We random select p_{rm}, and $m \subset S_{map}$;
2. We random choose on point p_{rn}, with $n \subset S_{P_T P_{r_1}}$



(a) *K*=5 (b) *K*=100

Fig. 4 Illustration of the weight values from (0,0) to other points with different *K*

3. And we calculate $A_{p_{rm},p_{rn}} = d_{p_{rm},p_{rn}} + K \times \int_C F(x, y)ds$, according to Eq. 3 to get A matrix
4. Repeat above steps 1–3.

The advantages of the proposed reward matrix update scheme are as following:

(1) Different from Q-learning, our method is not limited to the eight directions. The distance between any two points in GRL is calculated based on the geometric distance, which is useful for path planning when only partial information is available.
(2) The integral risk measure is embedded into a specific reward matrix, which contributes to a more reasonable path for UAVs.
(3) The single reward matrix can lead to an efficient algorithm for real time path planning, which is suitable to the navigation task of UAV.

4 Geometric Reinforcement Learning

GRL is designed for path planning of both single and multiple UAVs. For a single UAV, GRL is executed once the matrix A is updated when a new threatening object is detected. In the case of multiple UAVs, GRL is performed when A is changed by both the detected risk information from all UAVs.

4.1 Basic Idea of the Algorithm

In this paper, we aim to minimize both the integral risk and the length (time) of the path. The risk of the path can be computed by:

$$M = \int_C F(x, y)\, ds \tag{4}$$

where C is the path. The time or length is denoted by:

$$T = \int_C v ds \tag{5}$$

Where v is the speed of a UAV. Assuming that the speed v is fixed, and the time spent is simply proportional to the length of the path.

In this paper, we study the combination of the risk measure and path length for path evaluation, and define the optimal path in terms of Eqs. 4 and 5 as:

In this paper, the optimization objective is:

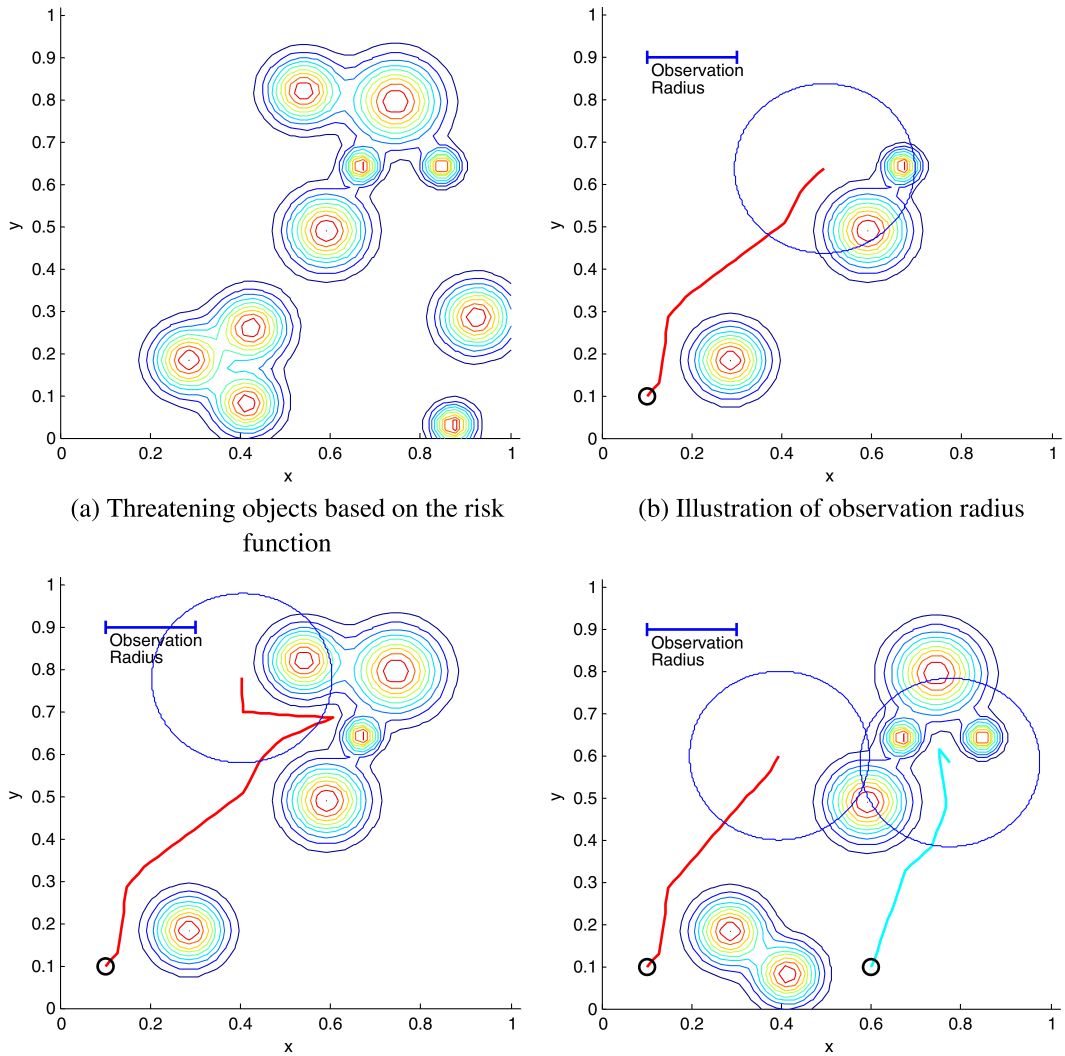
$$C^* = \arg\min_C (T + K \times M) \tag{6}$$

where c is any point set containing a sequence of points from the beginning to the target point on the map. If c* is with the smallest value in terms of both integral risk measure and path length, it is considered as an optimal one. And K is the parameter as mentioned in Eq. 3. The larger the K is, the less risky path the algorithm finds. To find the optimal path from the above weight matrix $T + K\text{x}M$, the dynamic programming method is generally used to search exhaustively for an optimal path. But it is not effective for path planning of UAV in real-time as the map information or the weight matrix is partially available. So this paper presents a reward matrix for such a purpose.

The UAVs only have a limited view scope, so they can just know the information from the threatening objects in the scope within their detection radius as shown in Fig. 5b. While multiple UAVs can share the information with each other, the view scope of UAVs can thus be greatly expanded.

In order to avoid the collision among UAVs, we need to handle its nearby UAVs. An UAV can be a threat object when it flies into the radius of other UAVs. Let the Observation Rradius be represented as *OR*. Considering motion of the UAVs, we set the threaten area in front of each UAV in its flight direction. The safety distance between two UAVs is denoted as *SD*, which is the minimum distance that UAVs must maintain. The ratio γ of OR to SD is empirically set to 0.1, which can result in a good path planning result. According to *OR*, we can set a reasonable value for *SD* in the practical application.

The view scope can be definitely expanded when multiple UAVs cooperate together to complete a task. When new dangerous objects are visible to any UAV, the risk information on the map and the weight matrices are accordingly updated. Moreover, if the distance between two UAVs is



(a) Threatening objects based on the risk function

(b) Illustration of observation radius

(c) Local optimum in real-time planning

(d) Information sharing

Fig. 5 Effect of information sharing for path planning

smaller than *SD*, each UAV has to be set to be an virtual risk object for other UAVs, and then the weight matrices will be dynamically updated (Fig. 6).

4.2 Description of GRL

The key idea of GRL is how to calculate the reward matrix *G*, which can be used to find the optimal path from a given point to the target point with optimal distance and integral risk.

4.2.1 Threatening Object Detection

(1) We first check whether there is any threatening object entering the circle area of the detection radius. If any, the threat information will be updated according to Eq. 3.

(2) We further check whether there is any UAV entering the circle area with high risk. If any, the threat information in the map will be updated according to the position of the threatening UAV.

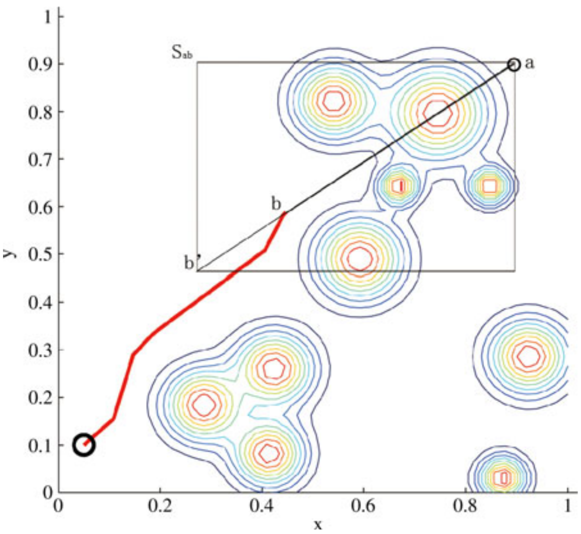


Fig. 6 The region of calculating G

4.2.2 Calculating the Matrix G for Each UAV

(1) Assign G_{p_t} on the end point p_t with 0, and other points with $+\infty$ (a big value).
(2) Update the matrix G for each UAV using the same procedure: randomly pick a point $p_r|r \in (1, N^2)$ on the map. And update G_{p_i} by the other points as:

$$G^{t+1}_{p_i} = \min\left\{G^t_{p_i}, A_{p_i,p_r} + G^t_{p_r}\right\},$$
$$r, i \in \left\{1, 2, \ldots, N^2\right\} \tag{7}$$

We provide the proof that the above process can be used to find the optimal path from a given point to the target point.

When we choose first point p_{r1}, the value of $G^t_{p_{r1}}$ is determined by p_t as only $G^t_{p_t} = 0$ and the values on other points are very large. Then $G^t_{p_{r1}} = A_{p_{r1}p_t} + G^t_{p_t} = A_{p_{r1}p_t}$.

For the point p_{r2} in the second time, we have two cases. One is that p_t leads to a small $G^t_{p_{r2}}\left(G^t_{p_{r2}} = A_{p_{r2}p_t} + G^t_{p_t} = A_{p_{r2}p_t}\right)$, and another is that p_{r1} makes $G^t_{p_{r2}}$ smaller $\left(G^t_{p_{r2}} = A_{p_{r2}p_{r1}} + G^t_{p_{r1}} = A_{p_{r2}p_{r1}} + A_{p_{r1}p_t}\right)$. Note that the final point is the target point p_t.

Supposed the mth point p_{rm} leads to an optimal path with p_t as the target, now we prove that the m+1th point leads to the same target point in our process. Similar to the above procedure, we can find the point p_{rk} with the smallest G value for the $(m+1)$th point, $G^t_{p_{r(m+1)}} = A_{p_{rk}p_{r(m+1)}} + G^t_{p_{rk}}$, as $G^t_{p_{rk}}$ and $k <= m$ is already calculated for any other points, we can find that the final point is the target point p_t. So the procedure can find the optimal path from the beginning to the target point.

(3) Repeat step (2) until the matrix G of any UAV is stable.

We can easily know that $G^{t+1}_{p_i} <= G^t_{p_i}$ from Eq. 7, and so $G^{t+1}_{p_i}$ will converge to a stable point, if t is large enough.

G cannot affect the calculation of the weight matrix, so the convergence of the whole algorithm depends on that of calculating G. We decrease the searching region of calculating G, and only the rectangle from the target to the UAV are considered as shown in Fig. 6.

4.2.3 Path Planning

Let a UAV start at p_s and the target point be p_t, all the points $p^0, p^1, p^2, \cdots p^M$ $(p^0 = p_s, p^M = p_t)$ on the path can be found as following:

$$p^{n+1} = \arg\min_{p^n}(G) \tag{8}$$

The above procedure is based on the greedy algorithm to find each point of a path. It will be redone when the weight matrix is updated or the scheduled path is not safe enough for the UAV to fly.

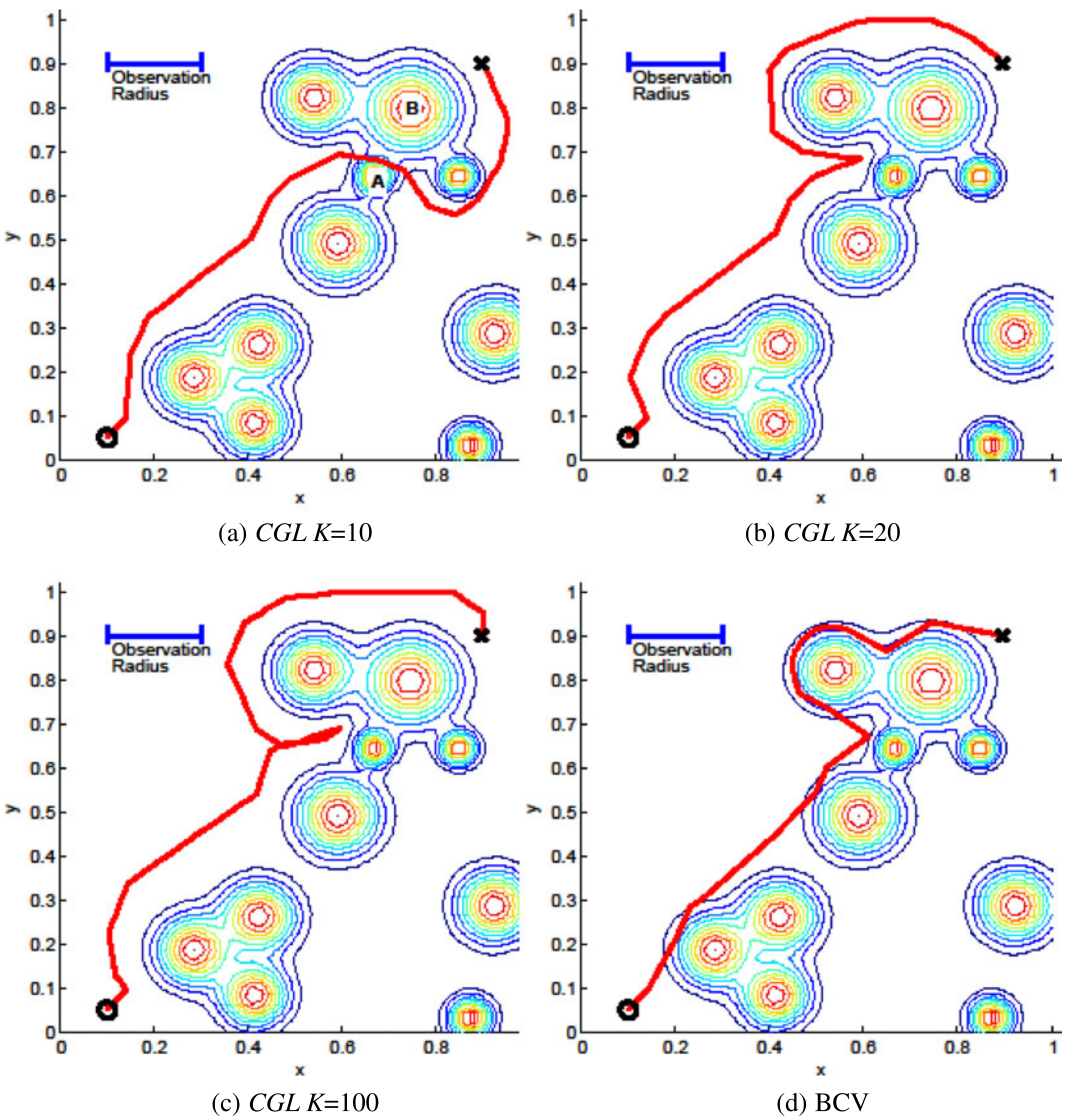
5 Experiments

The experiments are conducted for both single UAV and multiple UAVs. The compared methods include BCV, CPVS, and FBCRI (with information sharing for multiple UAVs), and Cooperative and Geometric Learning (CGL) method [19]. To measure the performance, we choose the same map for all the methods. The path length and the integral risk are used for quantitative evaluation.

While traditional methods consider only the path length, we propose to include the integral risk for better measurement of path planning.

5.1 The Comparative Experiments for Path Planning of a Single UAV

In this section, we conduct the experiments to evaluate the feasibility and adaptability of the proposed method. The threatening regions for a UAV are modeled as the exposure risk function, which is illustrated in Fig. 3. The threatening region is gradually detected when the distance between the UAV and the threat is less than the detection radius. The results of GRL for path planning for a single UAV with different values of K are shown in Fig. 7a–c; the path starts from the point (0.1,0.1) and end at the point (0.9,0.9). As shown in Fig. 5, UAV is likely to choose a shorter-term higher-risk path when K is smaller. As shown in Fig. 7a, the



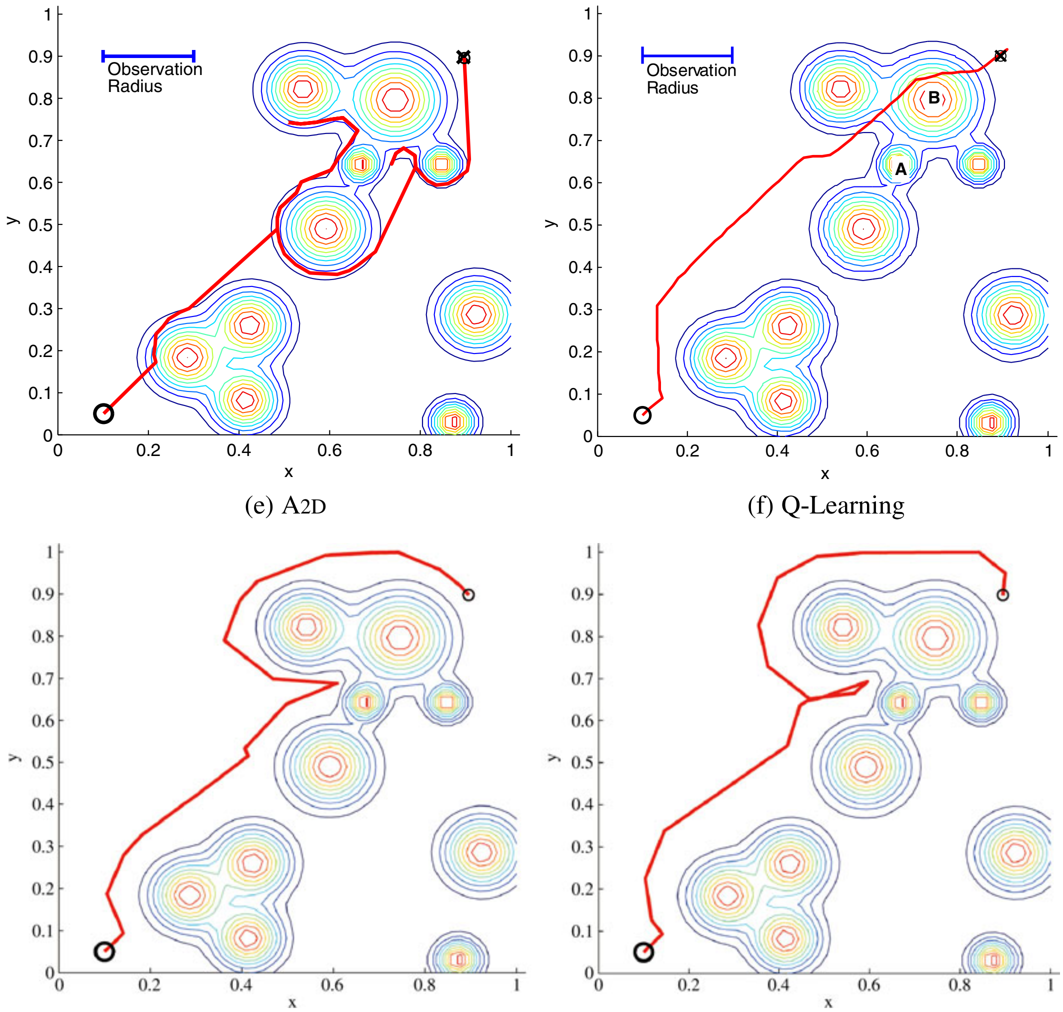
(a) *CGL K*=10

(b) *CGL K*=20

(c) *CGL K*=100

(d) BCV

Fig. 7 Examples of different values for K



(e) A2D

(f) Q-Learning

(g) GRL K=20

(h) GRL K=100

Fig. 7 (continued)

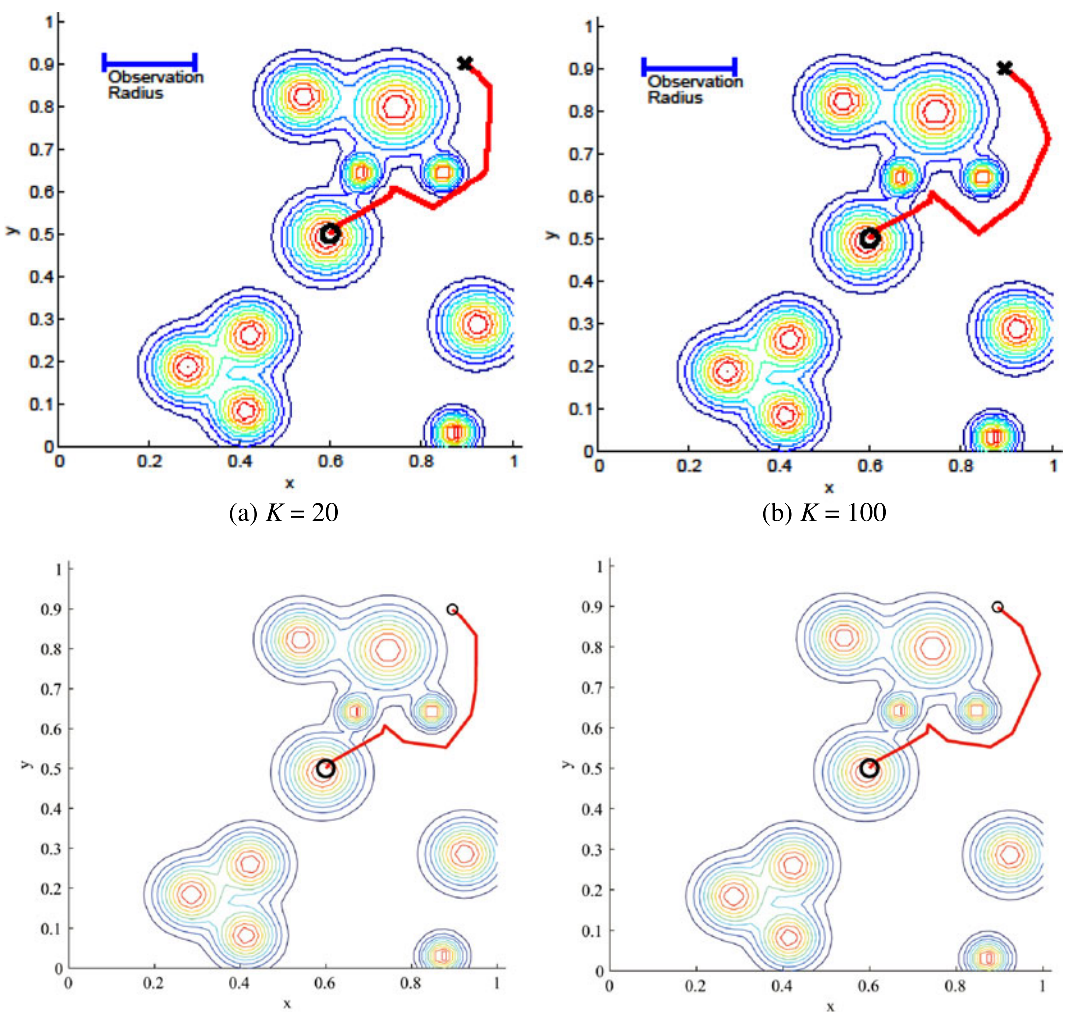
UAV passes through regions A and B to reach the target point when we choose a small K. This kind of characteristic is favorite for tasks like sharing risk information with other UAVs. When a large value is set to K, a UAV will fly away from the threat obstacles to reduce possible risks.

The comparative results with BCV are shown in Fig. 7d. Different from GRL, the BCV method suffers from local minima, and the path length and risk measure cannot be flexibly adjusted. This can be further confirmed in Fig. 7 and Table 1.

Table 1 Comparison of GRL and other methods in the risk measure for a single UAV

Planning method	Risk measure	Path length
BCV	10.1658	256
A_{2D}	24.3638	431
Q-Learning	27.7908	204.8
CGL, $K = 10$	2.4318	243
CGL, $K = 20$	0.9256	278
CGL, $K = 100$	0.4254	304
GRL, $K = 20$	0.8424	288
GRL, $K = 100$	0.4128	306

Moreover, as only the simple quantization of the map in the BCV algorithm is exploited, the UAV threats during the flight cannot be fully investigated. For example, if the threatening object flies for a long time around a threat region, the cost on time should be much higher than that of the object fast passing through the high-threat areas to the target point. The GRL learning is better than other methods in terms of the risk measure and path length, as it finds the path considering both the risk measure and the path length together. Comparative evaluation of planning path of BCV and GRL is also done with different K values. It can be seen from Table 1, in the case of $K = 10$, GRL is superior in terms of the risk measure and path length. When the path length is similar to that of BCV, the risk is only 1/4. Furthermore, a manual set of rules used by BCV needs to be well designed. But it flies in the region of large risk for a long time. Compared with the A2D and Q-Learning methods, GRL achieves much better performance. Specifically, the Q-Learning



(a) $K = 20$

(b) $K = 100$

(c) GRL $K = 20$

(d) GRL $K = 100$

Fig. 8 Illustration of path planning for a sudden threat

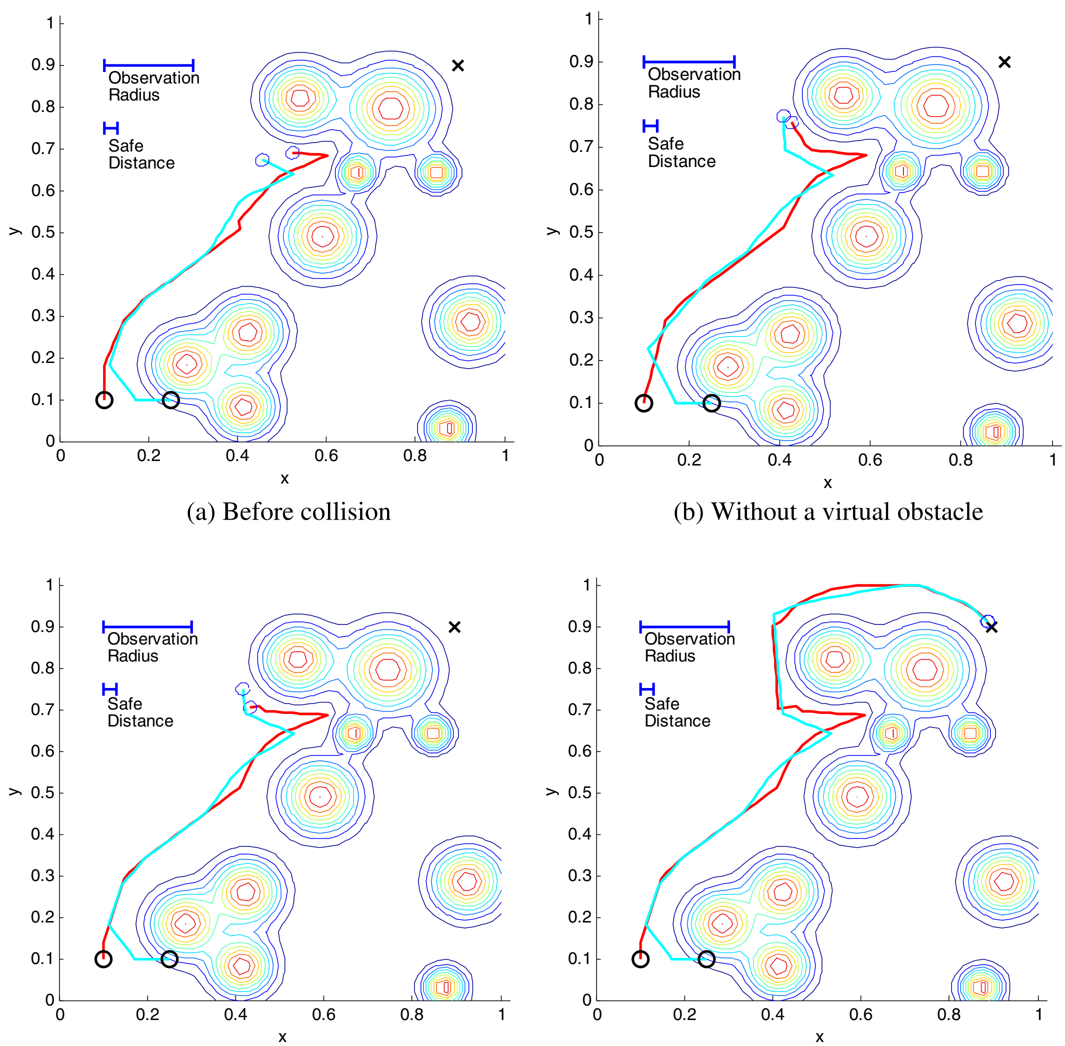
Table 2 Risk measure and path length for sudden threat testing

Planning method	Risk measure	Path length
CGL, $K = 20$	1.8940	105
CGL, $K = 100$	1.5217	113
GRL, $K = 20$	1.5956	108.8
GRL, $K = 100$	1.5411	106.3

method is much worse than other methods as it is restricted by eight directions in the further action.

To sum up, this paper uses a measure of risk and path length for performance comparison. The results are shown in Fig. 7 and Table 1. Noted that the path lengths of the comparative methods are re-scaled into the same unit as shown in BCV and other competing methods, while the value of risk measure is remained as its original value. The risk measure, introduced by us for the first time, is shown to be a good criterion of path evaluation.

The pilotless vehicle, UAV, should immediately respond to sudden changes in path planning



(a) Before collision

(b) Without a virtual obstacle

(c) With a virtual obstacle

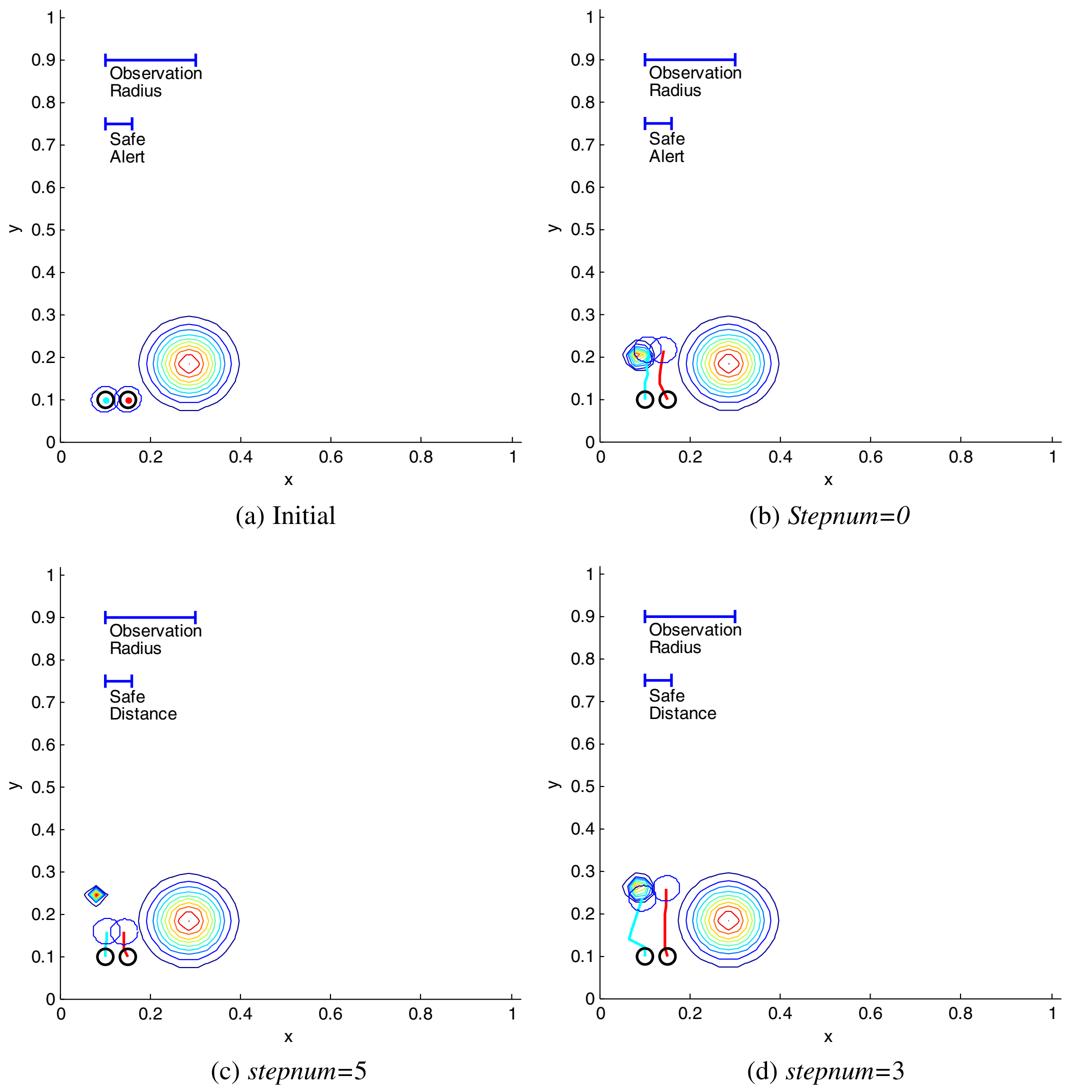
(d) The final result

Fig. 9 Illustration of a virtual obstacle model for path planning and avoid unexpected threats in route to its destination, in real time. In contrast to other methods, GRL is simple, reliable and robust for the sudden threats. The simulation results are shown in Fig. 8a and b with different K. From Table 2, we can see that the UAV can be easier to flee from the threatening area when a larger value of K is given, as a large value of K leads to a smaller risk and a longer distance.

5.2 Multiple UAV Path Planning Simulation

5.2.1 Multi-UAV Path Planning

Path planning for multiple UAVs needs to consider the problem of collisions among UAVs. UAVs affect each other during the flight, and probably collide with each other. Therefore, it is obvious that UAVs have to keep a safe distance



(a) Initial

(b) *Stepnum=0*

(c) *stepnum*=5

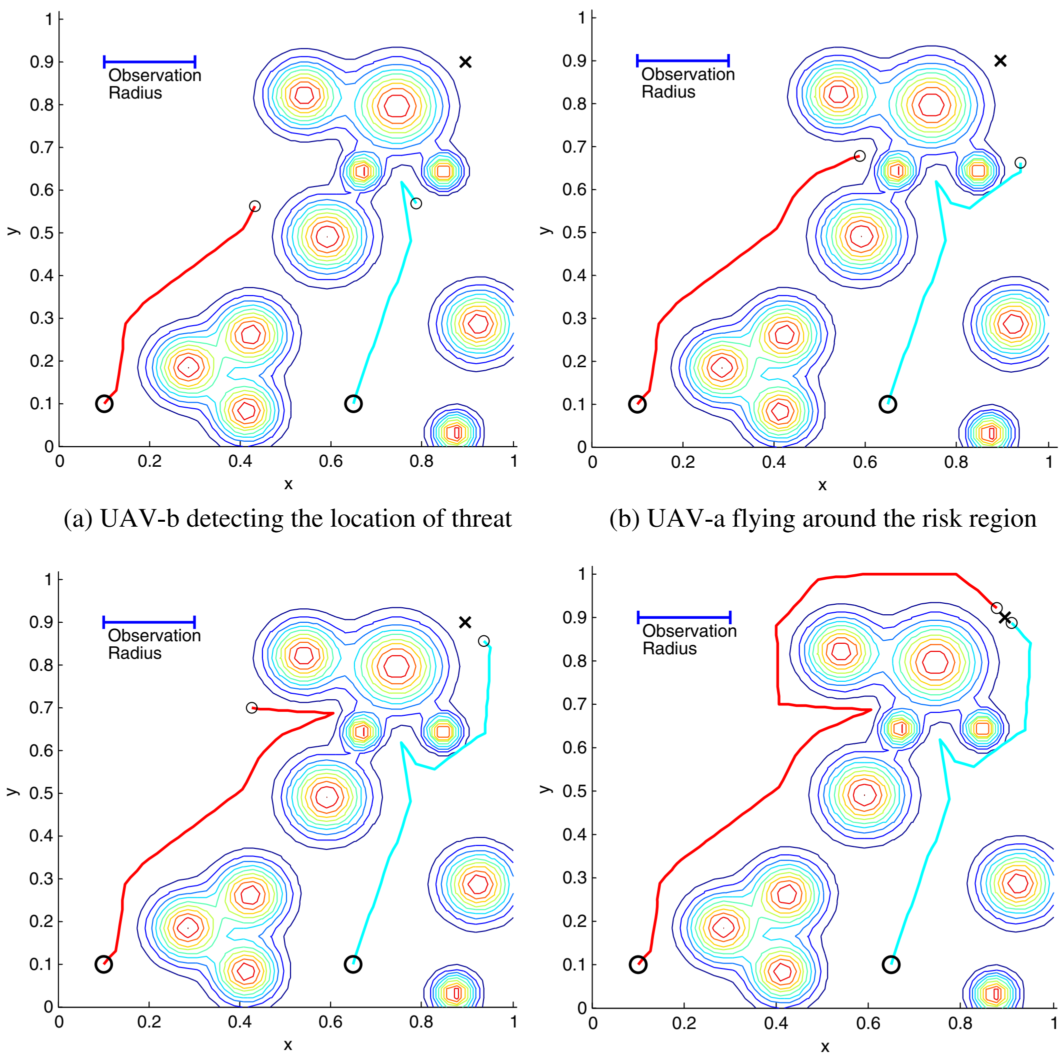
(d) *stepnum*=3

Fig. 10 The effect of *Stepnum* on path planning of UAV from others. We introduce a virtual obstacle for each UAV. The idea is that one UAV is considered as a threat object or obstacle for others, when the distance between two UAVs is too close.

In Fig. 9a, UAV a (the red UAV) starts from (0.1,0.1), while UAV b (the blue UAV) starts from (0.1,0.25), and the target for both UAVs is the point (0.9,0.9). As shown in Fig. 9b, the UAVs fly at the same speed, and will cause a collision near location (0.5, 0.7). If the virtual obstacle is set for each UAV in the same situation, UAV-a will adaptively change the path to avoid a collision as shown in Fig. 9c. In the following subsection, we will discuss how to set a virtual obstacle for each UAV.

5.2.2 The Setting of a Virtual Obstacle

UAVs are affected by any other UAV, which can be considered as the threatening objects in the procedure of path planning. Different from the real risk represented by Eq. 1, the threatening UAV is a virtual one and generally set to the front of other UAVs. We obtain an empirical



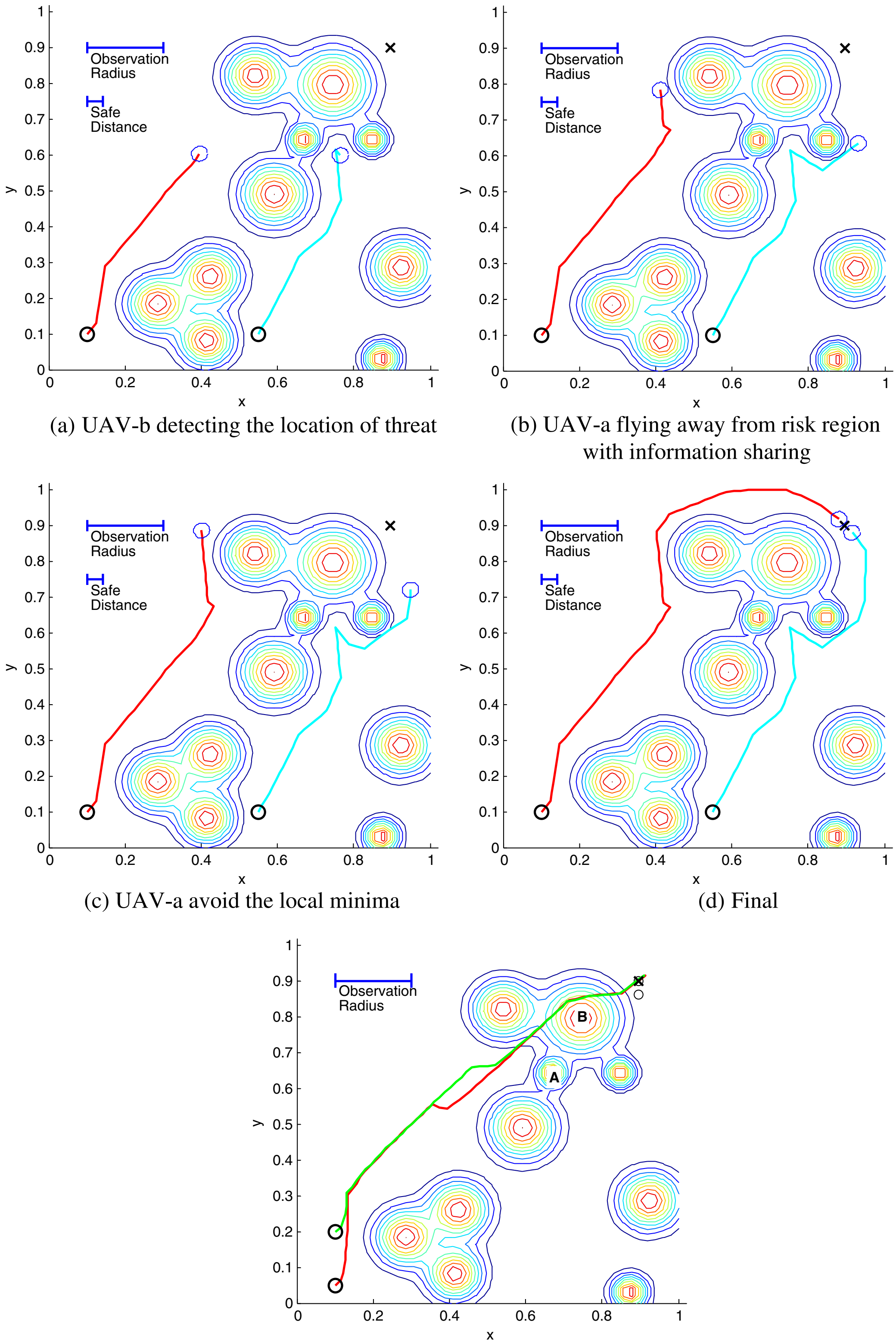
(a) UAV-b detecting the location of threat

(b) UAV-a flying around the risk region

(c) UAV-a continue flying until finding the threat

(d) Final result

Fig. 11 Illustration of path planning without information sharing



(a) UAV-b detecting the location of threat

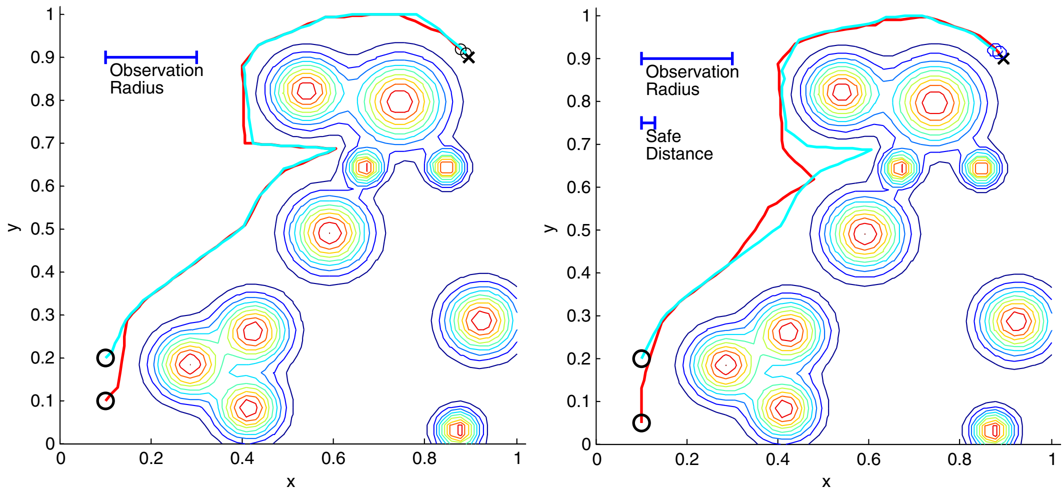
(b) UAV-a flying away from risk region with information sharing

(c) UAV-a avoid the local minima

(d) Final

(e) The result of Q-Learning

Fig. 12 Illustration of path planning with information sharing



(a) Without information sharing (b) With information sharing

Fig. 13 Effect of information sharing on path planning

value from the extensive and quantitative experiments; the smallest step is 40 % of the unit of the map. The number of the front step is denoted by *stepnum*, which denotes the number of the smallest step. The value of *stepnum* represent the distance, between any UAV and the virtual obstacle.

Figure 10 shows the effect of *stepnum* to the threat area. When the *stepnum* is 3, UAV a can well finish the task to avoid collision to reach the target point as shown in Fig. 10d. We can also see that UAVs will collide with each other, when *stepnum* is set to 0 or 5.

5.2.3 Avoiding Local Optima

We share the risk information in the UAVs for the collaborative path planning. After that, UAVs have a more broad view scope, and can effectively avoid local minima. Simulation results for two UAVs are shown in Figs. 11, 12 and 13. Without information sharing, two threat domains located at (0.6 0.7) and (0.7 0.8) detected by one UAV cannot be used by other UAVs. Two UAVs still fly toward the area of high risk, and result in local minima. It is an obvious waste of valuable operational time and energy, and also greatly increases the risk of the journey.

Investigation about performance with or without information sharing is shown in Figs. 11 and 12, in terms of the path length and the risk measure. The quantitative comparison of the information sharing effects for multi-UAV path planning are shown in Table 3. By using the information from other UAVs, GRL has a better performance as the view scope is greatly expanded and produces a much better path planning in terms of the degree of risk and the path length.

We also show the result, when two UAVs start at (0.1 0.1) and (0.1 0.2), as in Fig. 13 and Table 4. The result of path planning with information sharing is obviously better than that without information sharing. We also see that the result

Table 3 The evaluation of information sharing on path planning

Planning methods	Risk measure	Path length
Without information sharing	0.8972	278
With information sharing	0.3387	234
Q-Learning	22.8151	179.2

Table 4 The evaluation of information sharing on path planning

Planning methods	Risk measure	Path length
Without information sharing	0.9610	275
With information sharing	0.4539	253

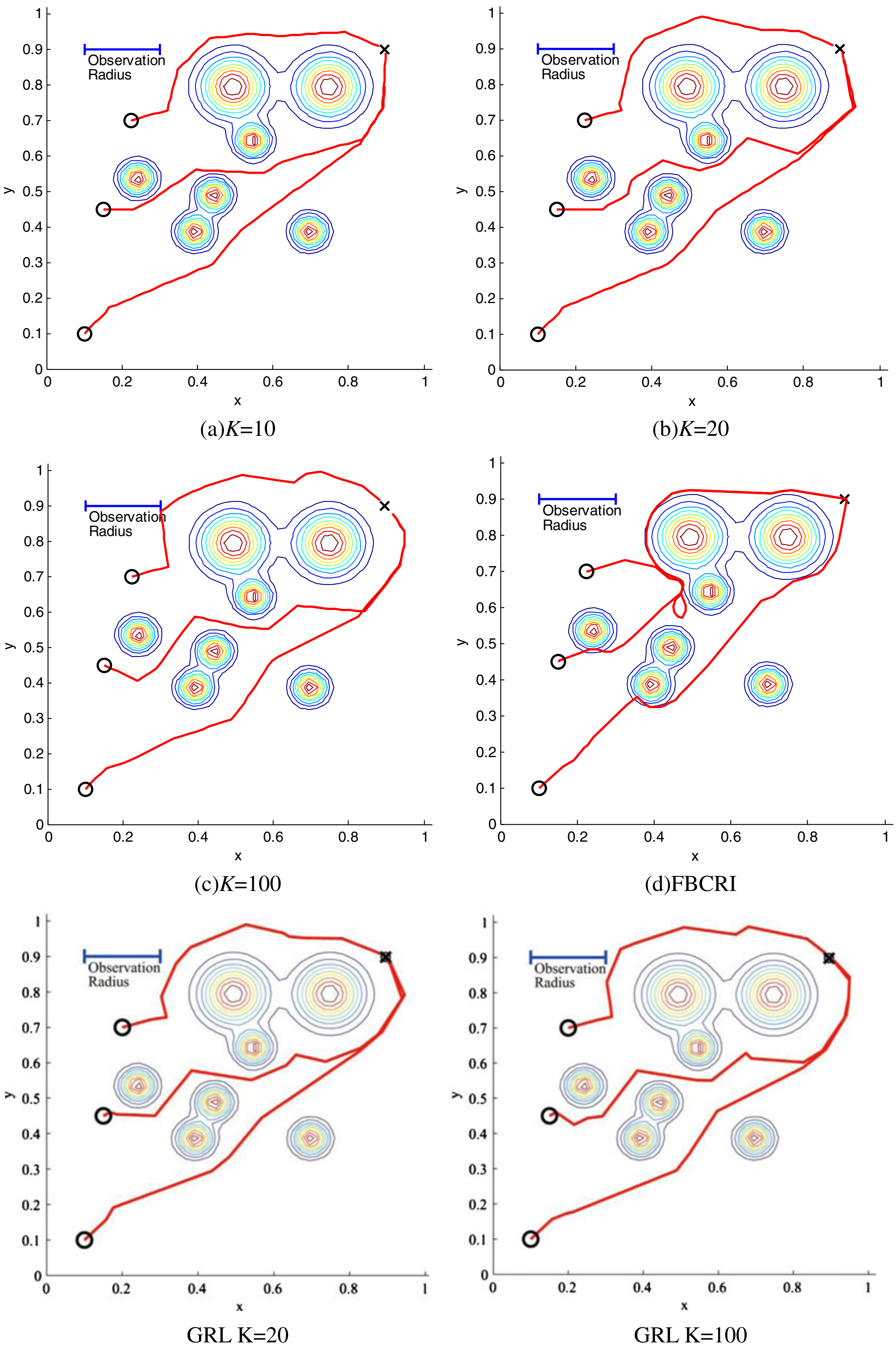


Fig. 14 Results of multiple UAV path planning with different K and the result of FBCRI

Table 5 Comparison of GRL and other methods in risk measure

Plan method	UAV1	UAV2	UAV3
FBCRI	3.5326	4.5282	4.5576
CGL, $K = 10$	0.2096	0.8282	0.4095
CGL, $K = 20$	0.0749	0.4551	0.1761
CGL, $K = 100$	0.0215	0.1523	0.0592
GRL, $K = 20$	0.0670	0.3300	0.1590
GRL, $K = 100$	0.0137	0.2048	0.0666

of Q-Learning is not effective, which suffers from much more risk than other GRL.

5.2.4 Multi-UAV Collaborative Path Planning

We choose the feedback based CRI (FBCRI) as the comparative method. FBCRI is for multi-UAV collaborative planning, which also uses the technology of virtual targets to avoid non-convergence for performance improvement. When $K = 10$, $K = 20$ and $K = 100$, the simulation results calculated by GRL for UAV flying from (0.1, 0.1) (0.15, 0.45) and (0.225, 0.7) to (0.9, 0.9) are shown in Fig. 14a–c. Figure 14d is the result of FBCRI, which needs manual selected rules for further action, and not flexible to adapt to various threatening objects.

In order to further compare two methods, we still use the risk measure and path length for a quantitative evaluation, and the results are shown in Tables 5 and 6. It can be seen that GRL achieves a much better performance than FBCRI, when the path lengths of GRL and FBCRI are similar. The risk measure for GRL is quite small, which shows that GRL is much superior to the FBCRI method.

Table 6 Comparison of GRL to other methods in path length

Plan method	UAV1	UAV2	UAV3
FBCRI	195	244	211
CPVS	188	234	199
CGL, $K = 10$	186	157	125
CGL, $K = 20$	192	173	131
CGL, $K = 100$	195	192	141
GRL, $K = 20$	192	166.4	137.6
GRL, $K = 100$	198.4	179.2	140.8

6 Conclusions

This paper proposes a new Geometric Learning method to solve the path planning for UAVs. Compared with other methods, GRL leads to a very single Path Planning algorithm. The parameter K in GRL can balance between the safety and economy of the paths. K can be adjusted to be suitable to different kinds of task of UAVs, which are designed to find the path with more flexibility. GRL is also designed for multi-UAV collaborative planning. The collisions of UAVs can be avoided each other by the specific predefined parameter (*stepnum*) and virtual obstacle. Meanwhile, we also verify that information sharing among UAVs is very effective for navigation. By the scheme of information sharing, the view scope can be expanded, and a shorter path with smaller risk can also be obtained.

Out future work will focus on setting adaptive K for more efficient path planning. We will also try to apply the algorithm to a real UAV to test its effectiveness in real applications.

Acknowledgement This work was supported in part by the Natural Science Foundation of China, under Contracts 60903065, 61039003 and 61272052, in part by the Ph.D. Programs Foundation of Ministry of Education of China, under Grant 20091102120001, in part by the Fundamental Research Funds for the Central Universities, and by the Program for New Century Excellent Talents in University of Ministry of Education of China.

References

1. Bortff, S.A.: Path planning for UAVs. In: IEEE Proceedings of the American Control Conference, pp. 364–36 (2008)
2. Chandler, P.R., Rasmussen, S.: UAV cooperative path planning. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, pp. 1–11 (2000)
3. Zheng C., Li L., Xu F., et al.: Evolutionary route planner for unmanned air vehicles. IEEE Trans. Robot. **21**(4), 609–620 (2005)
4. Rab bath, C.A., Gagnon, E., Lauzon, M.: On the cooperative control of multiple unmanned aerial vehicles.

In: IEEE Canadian ReviewSpring/Printemps, pp. 15–19 (2004)
5. Chandler, P.R., Pachter, M.: Complexity in UAV cooperative control. In: The Proceedings of the American Conference, pp. 1831–1836 (2002)
6. Bellingham, J., Richards, A., How J.P.: Receding horizon control of autonomous aerial vehicles. In: IEEE Proceedings of the American Control Conference, pp. 3741–3746 (2002)
7. Kim Y., Gu D.-W., Postlethwaite, I.: Real-time path planning with limited information for autonomous unmanned air vehicles. Automatica **44**, 696–712 (2008)
8. Nikolos, I.K., Zografos, E.S., Brintaki, A.N.: UAV path planning using evolutionary algorithms. Stud. Comput. Intell. **70**, 77–111 (2007)
9. Wu, S., Zheng, Z., Cai, K.-Y.: Real-time path planning for unmanned aerial vehicles using behavior coordination and virtual goal. Control Theory & Applications **28**(1), 131–136 (2011)
10. Dogman, A.: Probabilistic approach in path planning for UAVs. In: Proceedings of IEEE International Symposium on Intelligent, pp. 608–613 (2003)
11. Beard, R.W., Stepanyan, V.: Information consensus in distributed multiple vehicle coordinated control. IEEE Conf. Decis. Control **2**, 1359–1364 (2003)
12. Bauso, D., Giarrè, L., Pesenti, R.: Multiple UAV cooperative path planning via neuro-dynamic programming. In: Proceedings of the 43th IEEE Conference on Decision and Control (2004)
13. Zhao, L., Zheng, Z., Liu, W., Cai, K.-Y., Lin, S.: Real-time path planning for multi-UAVs with share of threats information. In: 6th IEEE Conference on Industrial Electronics and Applications, pp. 1359–1364. Beijing (2011)
14. Zheng, Z., Wu, S., Liu, W., Cai, K.-Y.: A feedback based CRI approach to fuzzy reasoning. Appl. Soft Comput. **11**(1), 1241–1255 (2011)
15. Christopher, J.C.H., Watkins, C.J.C.H.: Learning from delayed rewards. Ph.D. thesis, King's College, Cambridge (1989)
16. Watkins, C.J.C.H., Christopher, J.C.H.: Learning from Delayed Rewards. Cambridge University, Cambridge (1989)
17. Watkins, C.J.C.H., Dayan, P., Christopher, J.C.H.: Q-learning. Mach. Learn. **8**, 279–292 (1992)
18. Mao, Z., Zhang, B., Xu, B.: Path Planning of UAV Based on Q-learning. http://www.paper.edu.cn/releasepaper/content/201203-807 (2012). Accessed 29 Mar 2013
19. Zhang, B., Mao, Z., Liu, W., Liu, J.: Cooperative and geometric learning for path planning of UAVs. J. Intell. Robot. Syst. (2013). doi:10.1007/s10846-013-9901-z