



# A two-stage optimization method for unmanned aerial vehicle inspection of an oil and gas pipeline network

Yamin Yan<sup>1</sup> · Yongtu Liang<sup>1</sup> · Haoran Zhang<sup>1</sup> · Wan Zhang<sup>1</sup> · Huixia Feng<sup>1</sup> · Bohong Wang<sup>1</sup> · Qi Liao<sup>1</sup>

Received: 7 May 2018 / Published online: 13 March 2019  
© The Author(s) 2019

## Abstract

Oil and gas pipeline networks are a key link in the coordinated development of oil and gas both upstream and downstream. To improve the reliability and safety of the oil and gas pipeline network, inspections are implemented to minimize the risk of leakage, spill and theft, as well as documenting actual incidents. In recent years, unmanned aerial vehicles have been recognized as a promising option for inspection due to their high efficiency. However, the integrated optimization of unmanned aerial vehicle inspection for oil and gas pipeline networks, including physical feasibility, the performance of mission, cooperation, real-time implementation and three-dimensional (3-D) space, is a strategic problem due to its large-scale, complexity as well as the need for efficiency. In this work, a novel mixed-integer nonlinear programming model is proposed that takes into account the constraints of the mission scenario and the safety performance of unmanned aerial vehicles. To minimize the total length of the inspection path, the model is solved by a two-stage solution method. Finally, a virtual pipeline network and a practical pipeline network are set as two examples to demonstrate the performance of the optimization schemes. Moreover, compared with the traditional genetic algorithm and simulated annealing algorithm, the self-adaptive genetic simulated annealing algorithm proposed in this paper provides strong stability.

**Keywords** Pipeline network · Unmanned aerial vehicle inspection · Mixed-integer nonlinear programming · Two-stage solution

## Abbreviations

UAV	Unmanned aerial vehicle
MINLP	Mixed-integer nonlinear programming
GA	Genetic algorithm
SA	Simulated annealing algorithm
AGASA	Self-adaptive genetic simulated annealing algorithm
PSO	Particle swarm optimization
ACO	Ant colony optimization

## List of symbols

$F$	Total length of inspection path, m
$\alpha$	Conversion coefficient of distance and time

$I = \{0, 1, \dots, N\}$	Number of inspection nodes including the start node
$H = \{1, \dots, N\}$	Number of inspection nodes without start node
$L_{i,h}$	Distance between inspection node $i$ and node $h$ , m
$B_{i,h,k}$	(A binary variable): If the inspection mission from node $i$ to node $h$ is performed by UAV $k$ , it is 1; otherwise it is 0
$R_{i,k}$	(A binary variable): If node $i$ is inspected by UAV $k$ , it is 1; otherwise it is 0
$\theta$	Maximum turning angle of the UAV
$E_k$	Endurance of UAV $k$ , h
$d$	Minimum distance between the UAVs, m
$d_s$	Minimum safe distance between the UAVs, m
$x_0, y_0$	Coordinate of the UAV base, m
$x_i, y_i$	Coordinate of node $i$ , m
$L_m$	Chromosome of individual $m$ in the species

Edited by Xiu-Qin Zhu

✉ Haoran Zhang  
zhang\_ronan@csis.u-tokyo.ac.jp

<sup>1</sup> Beijing Key Laboratory of Urban Oil and Gas Distribution Technology, China University of Petroleum-Beijing, Beijing 102249, China

$P(L_m)$	Probability of selection of an individual
$p_m$	Probability of mutation
$T_0$	Initial temperature, °C
$w$	Temperature update coefficient
$M$	Size of population
$G_{\max}$	Maximum evolution generation
$T_{\text{end}}$	Termination temperature, °C

## 1 Introduction

Oil and gas pipelines have features of high pressure and flammability, sometimes accompanied by leakage, fire and explosion (Zhang et al. 2018a). Accidents could result in great loss of lives and high economic costs, so it is of great significance to inspect the pipeline regularly to minimize the risks. In recent years, unmanned aerial vehicles (UAV) have attracted much attention due to the low cost, high efficiency as well as safety. There has been extensive research regarding the civil UAV applications for pipeline inspection (Tu and Yang 2003; Patle et al. 2018; Reddy et al. 2011; Sedighi et al. 2004; Tsai et al. 2011; Hu and Yang 2004). UAVs can carry relevant detection equipment (Gómez and Green 2017) for different targets, including pipeline infrastructure, leak detection and pipeline environmental condition monitoring. UAV inspection paths of pipeline networks should be optimized to reduce manpower and to use material resources effectively. Previous research on optimization of inspection path focused on power grids, railway networks (Zhang et al. 2018b) and other networks. However, oil and gas pipeline network topology is more complicated (Wang et al. 2018a, b; Zhang et al. 2017b, d) and difficult to work out. Even though previous algorithms are applied, the optimal solution cannot be obtained in many cases.

Currently, there are three approaches to plan the inspection path for pipeline networks. The first one is the graph-based algorithm, such as the Voronoi diagram, the probabilistic roadmap and the Dijkstra's algorithm. The Voronoi diagram has been widely used in robot path planning (Bhattacharya and Gavriloa 2007; Candeloro et al. 2017; Chen and Chen 2014; Garrido et al. 2006). The inspection area is divided into points and these points are used to generate the Voronoi diagram. Then the Voronoi diagram divides the inspection area into many convex polygons which contain only one inspection point. When the graph contains the initial position and destination position, the optimal inspection path can be obtained. A probabilistic roadmap (Akbaripour and Masehian 2017; Geraerts and Overmars 2006; Wang and Cai 2018) is to convert the continuous space into discrete space and adopts a search algorithm to find the path on the roadmap to improve search efficiency. The Dijkstra's algorithm is one of the shortest path algorithms from one vertex

to the rest of the vertices, solving the shortest path problem in the directed graph. The main feature of this algorithm is to extend the outer layer centering on the starting point until it reaches the end. Chen et al. (2015) proposed a multi-objective optimization model for a wireless sensor network mobile agent problem, and the improved Dijkstra's algorithm was applied to solve the model. The optimal mobile agent path between any two nodes could be obtained according to the network environment. However, it is very difficult to combine the motion constraints of a UAV with the graph-based algorithm, and the number of sampling points also has a great effect on the path search results.

The second approach is the classical heuristic search algorithms (Gammell et al. 2015; Li et al. 2017; Yu and LaValle 2016), such as the A\* algorithm and Sparse A\* Search (SAS). The A\* algorithm was first described by Hart et al. (1968) and determined the optimal path from an initial node to a target node which evaluated each search position in the state space. In order to reduce the search space and realize the real-time path planning for the UAV (Szczerba et al. 2000), the constraints of the drone flight process are taken into account. However, the A\* algorithm and the SAS algorithm can only plan the path when all environmental information is known. What is worse, the search space will be larger with an increase in the number of inspection points, and the computation time of the classical heuristic search algorithm will increase exponentially.

The third approach is the modern heuristic search algorithm that mainly includes a genetic algorithm (GA) (Tu and Yang 2003; Nazarahari et al. 2019; Patle, et al. 2018; Sedighi et al. 2004; Tsai, et al. 2011; Hu and Yang 2004), particle swarm optimization (PSO) (Zhang et al. 2017c) and an ant colony optimization (ACO) (Zhang et al. 2017a). Shen et al. (2016) developed a novel method for the path planning for an electricity distribution network patrol. The vehicle routing problem (VRP) model was established, and the improved ACO was adopted to obtain the optimum patrol path. In this way, the patrol programs became more scientific, reasonable and efficient. Guo et al. (2017) established an optimization model of the logistics network to minimize the overall cost of the circulation-type distribution vehicle routing, and a genetic algorithm and a particle swarm optimization algorithm are implemented to solve the model. However, the proposed optimization issue is so complex that the computation can easily fall into a local optimum in the evolutionary process, which will lead to premature convergence and miss the optimal solution.

Given that oil and gas pipeline networks are more complex than electricity distribution networks, the self-adaptive genetic simulated annealing algorithm (AGASA) is introduced in this paper to improve the solution quality and efficiency of path planning. Moreover, based on the inspection demand of the pipeline network, taking the minimization

of the total length of the inspection path as the objective, and through the establishment of constraints including the mission scenario and the safety performance of UAVs, a mixed-integer nonlinear programming (MINLP) model is proposed. By adopting the two-stage solution methodology, the optimal inspection path is obtained.

## 2 Methodology

### 2.1 Preliminaries

In our path planner, all of the candidate routes are evaluated in the workspace. The pipelines are usually distributed linearly, so the pipeline is divided into multiple nodes and the two-dimensional coordinates of these nodes are identified as deterministic parameters in the model. It should be noted that the flight path consists of straight-line segments, e.g., a sequence of segments connecting the way nodes from the starting node to the goal node, and the starting node and goal node are the same.

### 2.2 Objective function

Suppose that there are  $K$  UAVs implemented in the oil and gas pipeline network inspection mission. Taking the total inspection path length as the objective, which can be defined as follows:

$$\min F = \sum_i \sum_h \sum_k (L_{i,h} B_{i,h,k}) \quad i \in I; h \in H; k \in K \quad (1)$$

$$L_{i,h} = \sqrt{(x_i - x_h)^2 + (y_i - y_h)^2} \quad (2)$$

where  $L_{i,h}$  defines the distance between the inspection node  $i$  and node  $h$ .  $B_{i,h,k}$  is a binary variable, if the inspection mission from node  $i$  to node  $h$  is performed by UAV  $k$ , it is 1. Otherwise it is 0,  $(x_i, y_i)$  and  $(x_h, y_h)$  are the coordinates of node  $i$  and node  $h$ .

### 2.3 Constraints

Each UAV starts from the base and returns to the base after completing the inspection mission. Also, it should be noted that the UAV is not allowed to return on the original path.

$$\sum_h B_{0,h,k} = 1 \quad h \in H; \quad k \in K \quad (3)$$

$$\sum_i B_{i,0,k} = 1 \quad i \in I; \quad k \in K; \quad i \neq h \quad (4)$$

Each node must be inspected by the UAV.

$$\sum_h B_{i,h,k} = R_{i,k} \quad i \in I; \quad h \in H; \quad k \in K \quad (5)$$

The overall flight time of each UAV cannot exceed its endurance.

$$\sum_i \sum_h \alpha L_{i,h} R_{i,k} \leq E_k \quad i \in I; \quad h \in H; \quad k \in K \quad (6)$$

Set  $a_i = (x_i - x_{i-1}, y_i - y_{i-1})$ , the flight angle of the UAV cannot exceed the maximum turning angle.

$$\frac{a_i a_{i+1}^T}{|a_i| |a_{i+1}|} \geq \cos(\theta) \quad i \in I \quad (7)$$

All of the UAVs perform the inspection mission according to the set path and the distance between UAVs cannot be less than the minimum safe distance  $d_s$ .

$$d \geq d_s \quad (8)$$

## 3 Two-stage solution methodology

In this paper, the two-stage solution methodology is proposed to optimize the UAV inspection path. In the first stage, the pipeline is divided into some nodes according to the vision of UAV. In the second stage, the AGASA is adopted to solve the model and obtain the optimal patrol path for an oil and gas pipeline network.

### 3.1 First-stage solution

Figure 1 shows the schematic diagram of the inspection area: The flight altitude is  $H$ , the angle of view of UAV is  $a$ , then the inspection radius can be calculated to be  $R = H \tan a$ . In the first stage, the pipeline is divided into some nodes according to the inspection radius. When the UAV is located at the top of the node, the whole pipeline of length  $2R$  can be detected. Similarly, the UAV only needs to inspect the nodes, and the whole pipeline can be detected, which is shown in Fig. 2. In this way, the size of the model is greatly reduced, and the workload of the inspection task can be minimized, thereby improving the inspection efficiency.

### 3.2 Second-stage solution

Genetic algorithm (GA) can search for the global optimal solution easily, but the local search optimization ability is poor and premature convergence could take place easily. As a stochastic optimization technique that simulates the annealing process of heating and melting metals proposed

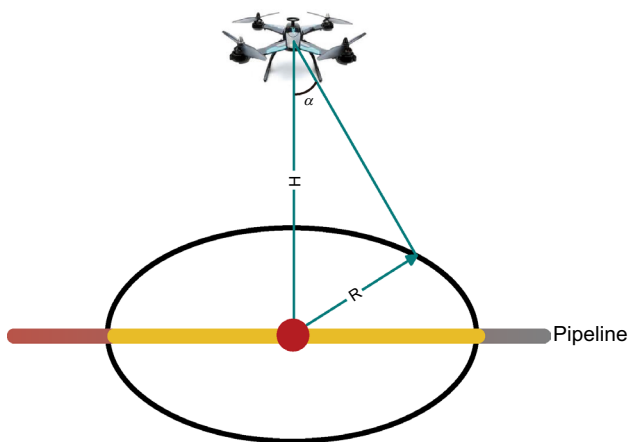


Fig. 1 Schematic diagram of the inspection area

by Metropolis, the simulated annealing algorithm (SA) is able to get rid of the local optimal solution and inhibit the precocity of a genetic algorithm, but it evolves slowly. In this section, the global parallel search ability of GA and the strong local serial search ability of SA are combined. By introducing heuristic rules in the process of generating the initial population and crossover operation, the optimal solution is expected to be found.

### 3.2.1 Self-adaptive genetic algorithm

#### (1) Generation of the initial population

The first step is to generate a set of flight routes for each UAV, and the starting node and terminal node of flight path are both the UAV base. To generate a large number of feasible solutions quickly, the pipeline is divided into segments, and the order of the nodes on the pipe segment remains unchanged. After that, the inspection path of each UAV is checked to see whether the constraints are satisfied. If the constraints are satisfied, the scheme would be feasible; otherwise, a new inspection path is generated, and the test is performed again. In this way, each inspection node is assigned to UAV, completing the layout of the initial solution. Better yet, the initial path has a lower fitness, which is

beneficial to fundamentally improve the convergence speed of the algorithm and optimize the convergence result.

The encoding of the path can be expressed as follows:

$$L_m = (L_1, L_2, \dots, L_k) = \left( x_0, y_0, x_{11}, y_{11}, \dots, x_{1s}, y_{1s}, x_0, y_0, x_{21}, y_{21}, \dots, x_{2t}, y_{2t}, x_0, y_0, x_{k1}, y_{k1}, \dots, x_{kw}, y_{kw}, x_0, y_0 \right) \quad (9)$$

where  $L_k$  defines the inspection path of the UAV  $k$ ,  $x_{1s}, y_{1s}$  is segment  $s$  of the pipeline inspected by the first UAV,  $x_{2t}, y_{2t}$  is segment  $t$  of the pipeline inspected by the second UAV. Similarly,  $x_{kw}, y_{kw}$  defines segment  $w$  of the pipeline inspected by UAV  $k$ .

#### (2) Fitness function

Fitness is a measure of the ability of an individual in a group to reach or come close to the optimal solution. The probability of individuals with higher fitness to evolve to the next generation is larger, while the probability of individuals with lower adaptation to reach the next generation is relatively small. From the analysis above, we can conclude that path planning is to find the minimum of the objective function under some constraints. If the inspection mission can be completed by one UAV, only four constraints are involved. If the inspection mission needs multiple UAVs, all five constraints should be included.

The objective function is chosen as a fitness function:

$$\min f = L_m = \sum_i \sum_h \alpha L_{i,h} B_{i,h,k} \quad (10)$$

#### (3) Selection of genetic operator

Selection operators are used to simulate the survival of the fittest in the process of biological and genetic evolution. Crossover operators are used to imitate the production of new biological individuals or species, and mutation operators are used to imitate the accidental errors resulting from cell division and replication. Also, the selection operator generates a next generation of the population by replicating individuals, and crossover and mutation operators are responsible for generating new individuals.

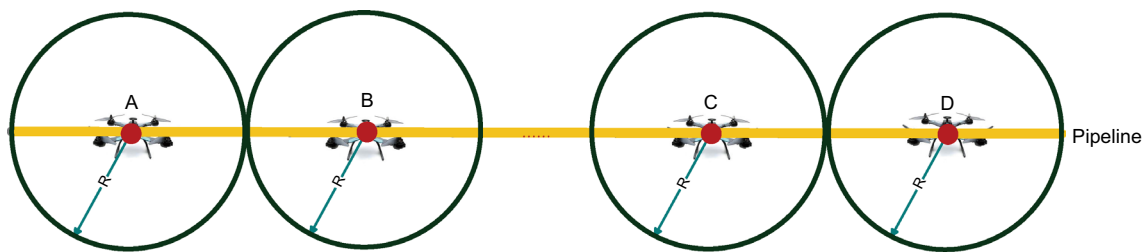


Fig. 2 Top view of UAV implements inspection mission

(A) Selection operator

The algorithm adopts the roulette wheel selection and the elitist model to select the individual.

(1) Roulette wheel selection

The probability of an individual being selected is proportional to the fitness value. This can be expressed as follows:

$$P(L_m) = \frac{f(L_m)}{\sum_{m=1}^M f(L_m)} \tag{11}$$

The specific implementation of the algorithm is as follows:

- Summing up the fitness function values of all individuals in the population and getting the  $f_{total}$ ;
- A random number  $f_{random}$  is generated in the interval  $[0, f_{total}]$ ;
- The fitness function values are incremented one by one starting from the first individual. The chromosome  $L_m$  will be chosen if  $q_m \geq f_{random}$ ;

$q_m$  is named as the accumulated probability of the chromosome  $L_m$ , whose formula of calculation is expressed in Eq. (12)

$$q_m = \sum_{k=1}^m P(L_k) \tag{12}$$

(2) Elitist model

The genetic algorithm continuously generates new individuals by performing crossover and mutation operations. Although more and more good individuals will be produced as the evolutionary process of the population increases, they may also destroy the individuals with the best fitness in the current population due to the randomness of genetic operations such as replication, crossover and mutation. Therefore, the elitist model is used to perform the survival of the fittest, the individuals with the highest fitness in the current population do not participate in the crossover and mutation operation. Instead, it is used to replace the individuals with the lowest fitness value after genetic manipulation in the current generation.

The specific operation steps of the elitist model are as follows:

- Identify the individuals with the highest fitness value  $f_{best}^m$  and the lowest fitness value  $f_{worst}^m$  in the current population.

- If  $f_{best}^m > f_{best}^{all}$ , take the individual with the highest fitness value  $f_{best}^m$  as the best individuals in all populations, i.e.,  $f_{best}^{all} = f_{best}^m$ .
- Replace the individual with the lowest fitness value  $f_{worst}^m$  in the current population with the best individuals  $f_{best}^m$  in all populations, i.e.,  $f_{worst}^m = f_{best}^{all}$ .

(B) Crossover

The individuals generated by the parent generation will perform the multi-segment crossover operation, during which the nodes on the segment remain unchanged. In this way, the UAVs are restricted to fly along the pipeline, avoiding the redundant path in the process of crossover operation.

It can be described as follows:

- (1) Set the number of intersections to  $W$  and select the intersection range  $p \in [p, p + W]$  randomly. Meanwhile, set  $r = 1$ .
- (2) Search the position of the intersection in the individuals  $C_1$  and  $C_2$ , and record it as  $x$  and  $y$ .
- (3) Switch the position of individuals  $C_1(1, p)$  and  $C_2(1, p)$ .
- (4) Switch the position of individuals  $C_1(1, x)$  and  $C_2(1, y)$ ,  $r = r + 1$ .
- (5) If  $r < W$ , go back to Step 2, otherwise, the crossover operation is finished.

$$C_1 = L_m^{C_1} = [S_1^{C_1}, S_2^{C_1}, \dots, S_k^{C_1}]$$

$$= \begin{bmatrix} x_0^{C_1}, y_0^{C_1}, x_{11}^{C_1}, y_{11}^{C_1}, \dots, x_{1s}^{C_1}, y_{1s}^{C_1}, x_0^{C_1}, y_0^{C_1}, x_{21}^{C_1}, y_{21}^{C_1}, \dots, \\ x_{2t}^{C_1}, y_{2t}^{C_1}, x_0^{C_1}, y_0^{C_1}, x_{k1}^{C_1}, y_{k1}^{C_1}, \dots, x_{kw}^{C_1}, y_{kw}^{C_1}, x_0^{C_1}, y_0^{C_1} \end{bmatrix} \tag{13}$$

$$C_2 = L_m^{C_2} = [S_1^{C_2}, S_2^{C_2}, \dots, S_k^{C_2}]$$

$$= \begin{bmatrix} x_0^{C_2}, y_0^{C_2}, x_{11}^{C_2}, y_{11}^{C_2}, \dots, x_{1s}^{C_2}, y_{1s}^{C_2}, x_0^{C_2}, y_0^{C_2}, x_{21}^{C_2}, y_{21}^{C_2}, \dots, \\ x_{2t}^{C_2}, y_{2t}^{C_2}, x_0^{C_2}, y_0^{C_2}, x_{k1}^{C_2}, y_{k1}^{C_2}, \dots, x_{kw}^{C_2}, y_{kw}^{C_2}, x_0^{C_2}, y_0^{C_2} \end{bmatrix} \tag{14}$$

After performing the crossover operation, two new individuals  $D_1 D_2$  can be obtained:

$$D_1 = L_m^{D_1} = [S_1^{D_1}, S_2^{D_1}, \dots, S_k^{D_1}]$$

$$= \begin{bmatrix} x_0^{D_1}, y_0^{D_1}, x_{11}^{D_1}, y_{11}^{D_1}, \dots, x_{1s}^{D_1}, y_{1s}^{D_1}, x_0^{D_1}, y_0^{D_1}, x_{21}^{D_1}, y_{21}^{D_1}, \dots, \\ x_{2t}^{D_1}, y_{2t}^{D_1}, x_0^{D_1}, y_0^{D_1}, x_{k1}^{D_1}, y_{k1}^{D_1}, \dots, x_{kw}^{D_1}, y_{kw}^{D_1}, x_0^{D_1}, y_0^{D_1} \end{bmatrix} \tag{15}$$

$$D_2 = L_m^{D_2} = [S_1^{D_2}, S_2^{D_2}, \dots, S_k^{D_2}]$$

$$= \begin{bmatrix} x_0^{D_2}, y_0^{D_2}, x_{11}^{D_2}, y_{11}^{D_2}, \dots, x_{1s}^{D_2}, y_{1s}^{D_2}, x_0^{D_2}, y_0^{D_2}, x_{21}^{D_2}, y_{21}^{D_2}, \dots, \\ x_{2t}^{D_2}, y_{2t}^{D_2}, x_0^{D_2}, y_0^{D_2}, x_{k1}^{D_2}, y_{k1}^{D_2}, \dots, x_{kw}^{D_2}, y_{kw}^{D_2}, x_0^{D_2}, y_0^{D_2} \end{bmatrix} \quad (16)$$

(C) Mutation

The individuals that will undergo a mutation operation can be expressed as follows:

$$E_1 = L_m^{E_1} = [S_1^{E_1}, S_2^{E_1}, \dots, S_k^{E_1}]$$

$$= \begin{bmatrix} x_0^{E_1}, y_0^{E_1}, x_{11}^{E_1}, y_{11}^{E_1}, \dots, x_{1s}^{E_1}, y_{1s}^{E_1}, x_0^{E_1}, y_0^{E_1}, x_{21}^{E_1}, y_{21}^{E_1}, \dots, \\ x_{2t}^{E_1}, y_{2t}^{E_1}, x_0^{E_1}, y_0^{E_1}, x_{k1}^{E_1}, y_{k1}^{E_1}, \dots, x_{kw}^{E_1}, y_{kw}^{E_1}, x_0^{E_1}, y_0^{E_1} \end{bmatrix} \quad (17)$$

$$E_2 = L_m^{E_2} = [S_1^{E_2}, S_2^{E_2}, \dots, S_k^{E_2}]$$

$$= \begin{bmatrix} x_0^{E_2}, y_0^{E_2}, x_{11}^{E_2}, y_{11}^{E_2}, \dots, x_{1s}^{E_2}, y_{1s}^{E_2}, x_0^{E_2}, y_0^{E_2}, x_{21}^{E_2}, y_{21}^{E_2}, \dots, \\ x_{2t}^{E_2}, y_{2t}^{E_2}, x_0^{E_2}, y_0^{E_2}, x_{k1}^{E_2}, y_{k1}^{E_2}, \dots, x_{kw}^{E_2}, y_{kw}^{E_2}, x_0^{E_2}, y_0^{E_2} \end{bmatrix} \quad (18)$$

The specific mutation operation is described as follows:

- (1) A random mutation probability  $p_m$  is generated in the interval  $[0, 1]$ .
- (2) A random two elements on the chromosome  $L_m$  will switch position if  $0 \leq p_m \leq 0.5$ , otherwise, a random three elements on the chromosome  $L_m$  will switch position if  $0.5 < p_m \leq 1$ .
- (3) The new individuals  $F_1$  and  $F_2$  are generated after a mutation operation.

$$F_1 = L_m^{F_1} = [S_1^{F_1}, S_2^{F_1}, \dots, S_k^{F_1}] \quad (19)$$

$$F_2 = L_m^{F_2} = [S_1^{F_2}, S_2^{F_2}, \dots, S_k^{F_2}] \quad (20)$$

3.2.2 Simulated annealing algorithm

(1) Initial temperature

The initial temperature needs to be high enough to ensure that the algorithm has strong ergodicity from the beginning to avoid falling into the local optimal solution.  $T_0$  is selected as the initial temperature which can be denoted as:

$$T_0 = 10 \times N \quad (21)$$

(2) Temperature update function

The temperature update function is used to modify the temperature value in the external cycle. It is generally required that the temperature drops to nearly zero and then converges to the global optimal solution. Equal proportion temperature is chosen as the temperature update function since it is effective and easy to implement:

$$T_{n+1} = wT_n, n \geq 0, 0 \leq w \leq 1 \quad (22)$$

(3) Acceptance function

Generally, the Metropolis criterion is selected as the acceptance function for generating a solution to a combinatorial optimization problem. At the same time, it is used to discriminate against the probability  $P_t$  that the new solution  $q_{new}$  will replace the old solution  $q_{old}$ .  $t$  is the temperature of the isothermal process. The formula for calculating the probability is as follows:

$$P_t = \begin{cases} 1 & f(q_{old}) \geq f(q_{new}) \\ \exp\left(\frac{f(q_{old}) - f(q_{new})}{t}\right) & f(q_{old}) < f(q_{new}) \end{cases} \quad (23)$$

3.2.3 Path planning steps

The steps to plan the patrol path of UAVs by AGASA are shown in Fig. 3 and described as follows:

- (1) Initialize a group of paths with the size of  $M$  and set the parameters of the initial temperature  $T_0$ , maximum evolution generations  $G_{max}$ , and termination temperature  $T_{end}$ .
- (2) Initialize the genetic algebra counter  $g = 0$  and evaluate the fitness of each path.
- (3) According to the crossover probability and mutation probability, the individuals selected from each subpopulation to undergo the operations of selection, crossover as well as mutation are decided, then the fitness value of each new individual is calculated.
- (4) Decide whether to replace the old individual with the new individual according to the Metropolis criterion.
- (5) If  $g < G_{max}$ ,  $g = g + 1$ , go back to Step 3, Otherwise, go to Step 6.
- (6) If  $T < T_{end}$ , update the temperature and go back to Step 2, otherwise, the optimal or near-optimal routes are found.



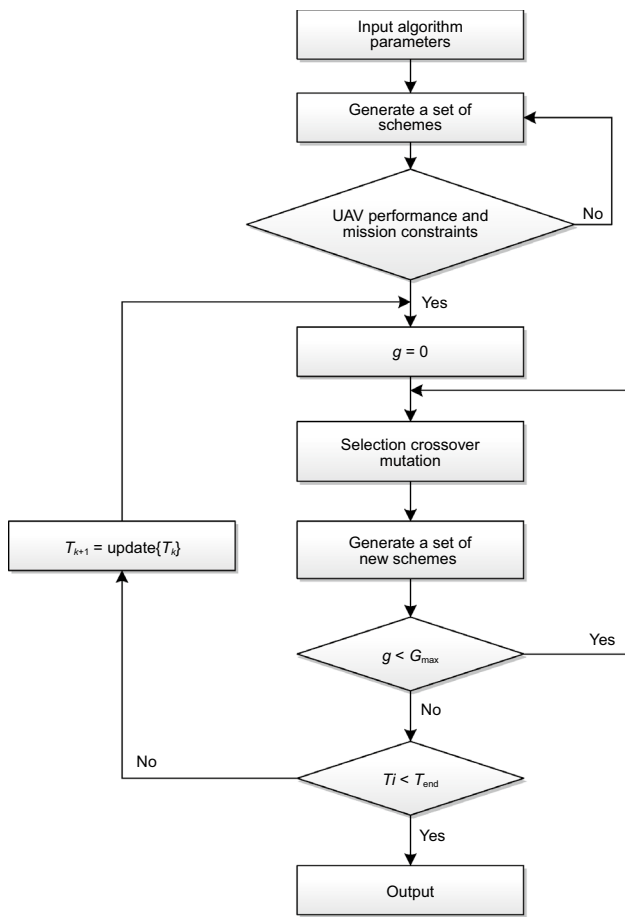


Fig. 3 Flowchart of the second-stage solution

### 4 Computational studies

UAVs are equipped with high-resolution cameras and sensors that transmit information in real time, through which they can detect changes in the environment around the

**Table 1** The parameters and configuration of UAV

Flying altitude	Low (< 100 m)
Velocity	5–15 km/h
Endurance	5 h
Payload	< 7 kg
Platform	Fixed wing
Sensor	Optical or IR camera, LIDAR

pipeline and identify any pipeline leaks. Table 1 presents the parameters and configuration of UAV.

#### 4.1 Example 1

A 5 × 5 km virtual pipeline network is set as a research example. In this paper, we propose a two-stage solution method for the inspection path optimization as shown in Fig. 4. The proposed method shows a relatively universal frame and realizing methodology for the comprehensive solution of optimization. The coordinates of production wells and valves are known. The base was chosen to be built at the valve to facilitate the operation of UAVs. The method is decomposed into two parts: One is the pipeline segmentation according to the vision capability of the UAV, which can select an effective inspection range and greatly reduce the scale of the model. In the second part, the self-adaptive genetic simulated annealing algorithm is introduced to deal with the inspection path optimization problem for the oil and gas pipeline network. Finally, the optimal solution shows that only one UAV is required for the inspection task, and the total length of the inspection path is 16,250 m.

In this paper, three algorithms (AGASA, GA and SA) are applied to solve the model. During initialization, the population size is set to 500, the maximum evolution generations are 2000, the temperature update coefficient is 0.97, and the

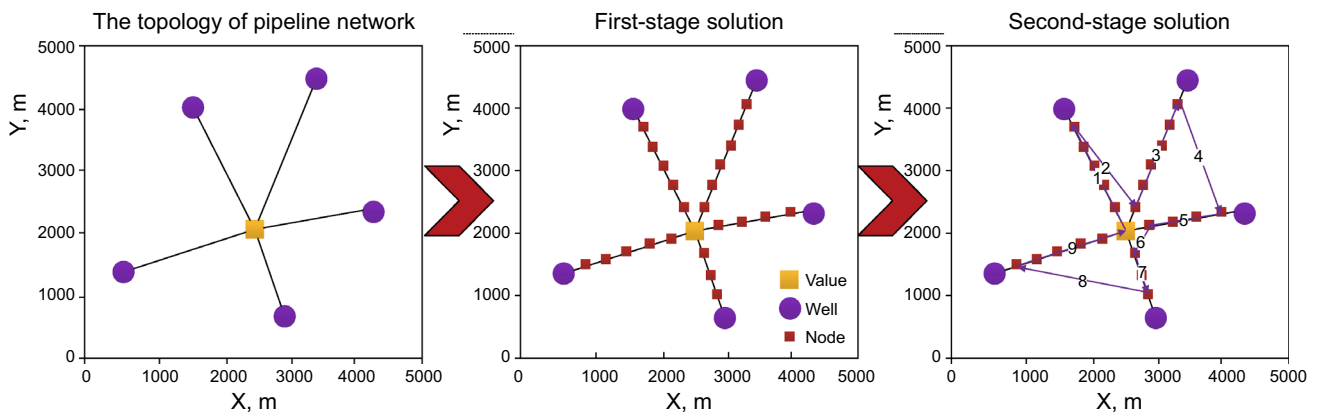
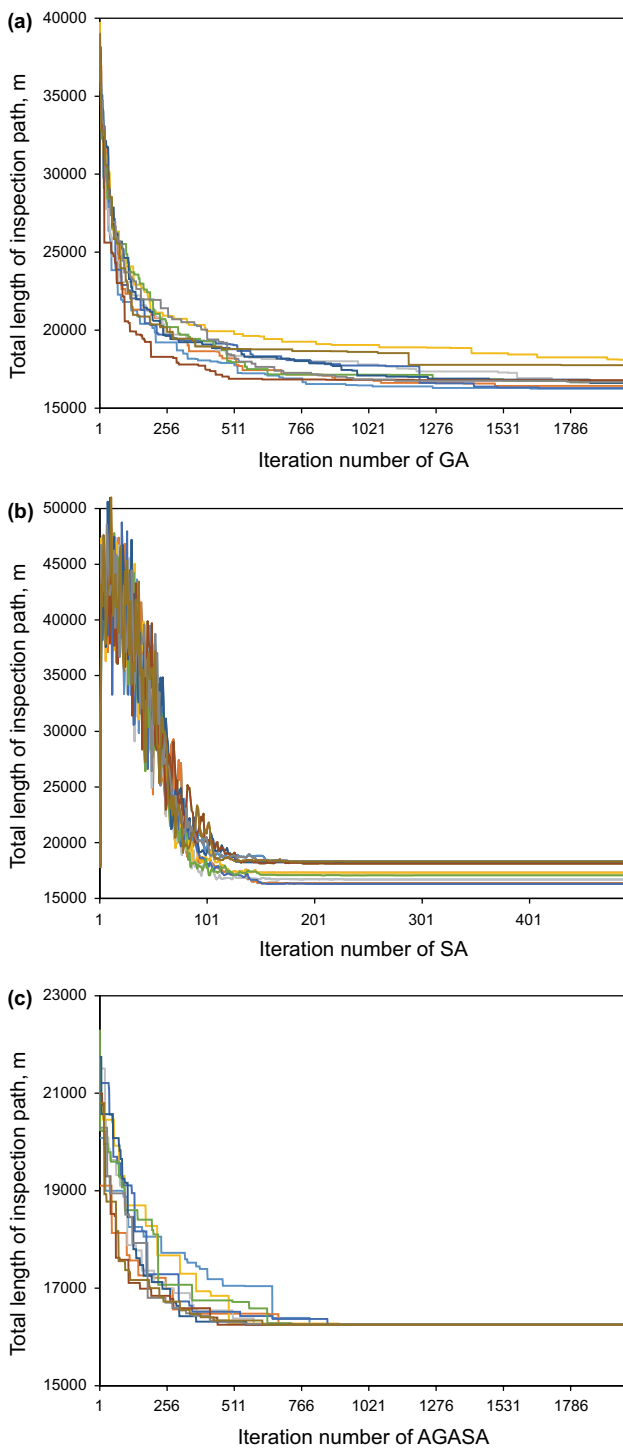


Fig. 4 The frame of the two-stage solution method



**Fig. 5** Calculation results of different algorithms

termination temperature is set to 0.001. To test the stability of each algorithm, the calculation was repeated 10 times, and the results are shown in Fig. 5. It can be noted that the calculation stability of GA and SA is unsatisfactory. As for GA, the solution is not tending to be stable after 2000 iterations. SA has a better performance than GA; it tends to be

**Table 2** Influence of population size on calculation results

Population size	Best fitness value
50	19,811
100	19,787
150	18,937
200	18,891
250	18,696
300	17,819
350	16,250
400	16,250
450	16,250
500	16,250

**Table 3** Influence of temperature update coefficient on calculation results

Temperature update coefficient	Best fitness value
0.90	18,932
0.91	18,067
0.92	17,406
0.93	16,858
0.94	16,826
0.95	16,250
0.96	16,250
0.97	16,250
0.98	16,250
0.99	16,250

stable after 150 iterations. However, the calculation results are not consistent each time, which means that there is a serious risk that the algorithm missed the optimal solution and fell into a local optimum in the calculation process. On the contrary, the calculation results of AGASA are consistent each time after 900 iterations, which can be attributed to the treatment of the initial solution and introduction of heuristic rules in the algorithm. This further verified that the AGASA proposed in this paper has a great advantage in solving the problem of UAV inspection path optimization for oil and gas pipeline networks.

Also, the influence of algorithm parameters on the optimal solution is tested in this paper. Generally, two main parameters affecting the performance of the algorithm include population size and temperature update coefficient. According to that, the experiments are conducted, respectively, with the specified  $w$  value of 0.95 and then continued with the specified  $M$  value of 500. Tables 2 and 3 show the calculation results under different population and temperature update coefficients. From Table 2, it can be noted that the performance of genetic optimization is not very good if the population size is too small. Conversely, selecting a



larger number of initial populations can process more solutions at the same time, thereby finding the global optimal solution easily. In detail, the optimal solution can be found when the population size reaches 350. Meanwhile, there is a similar phenomenon in the temperature update coefficient. When the value of the temperature update coefficient is closer to 1, the chance of finding the global optimal solution is greater.

### 4.2 Example 2

A real case in North China is studied as a research example, including a central processing facility (CPF), four valves and twenty-nine production wells. Figure 6 shows the topology of the gathering and transportation network. The base was chosen to be built at the CPF.

During the generation of the initial solution and crossover operation, the nodes on the pipeline segment remain

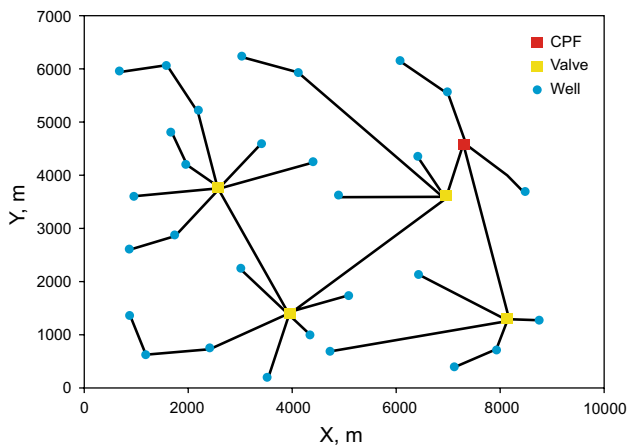


Fig. 6 The topology of the gathering and transportation network

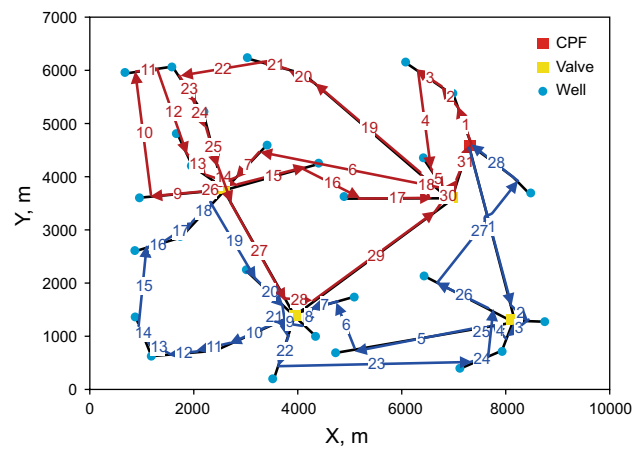


Fig. 8 Inspection path of UAVs for the oil and gas pipeline network (\*\*brown and blue in the paths represent the first and second UAV)

unchanged, which means that the size of the model is largely related to the number of segments divided. For the sake of simplification, pipelines of less than 1.5 km in length will not be divided and considered as segments directly. In contrast, pipelines with a length greater than 1.5 km will be divided into segments at a distance of 1.5 km. This can greatly reduce the model scale and accelerate the convergence speed, thereby finding the optimal solution quickly.

Similarly, the population size is set to be 500, the maximum evolution generation is 3000, the temperature update coefficient is 0.97, and the termination temperature is set to be 0.001. Meanwhile, the calculation is repeated 10 times by adopting SA and AGASA, and the results are shown in Fig. 7. As seen from the table, the stability and convergence of SA are unsatisfactory as the model scale increases. Differently from that, AGASA showed great stability and good convergence owing to the reduction in model scale

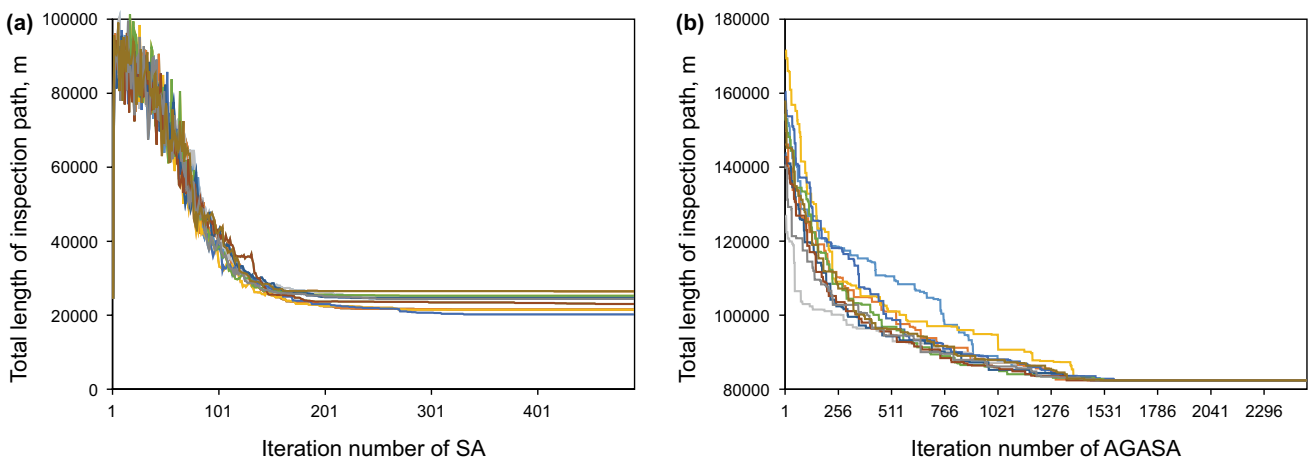


Fig. 7 Calculation results of SA and AGASA

and the introduction of heuristic rules. This shows that the AGASA proposed in this paper has a great advantage in solving the problem of UAV inspection for oil and gas pipeline networks. Figure 8 shows the inspection paths obtained by AGASA, and the total length of the inspection path is 82,389 km.

## 5 Conclusions

In this paper, a new inspection path optimization method for oil and gas pipeline networks is proposed. A mixed-integer nonlinear programming model is established by setting the minimum length of the inspection path as the objective function, which also takes the mission scenario and the safety performance of UAVs into account. In model solving part, a two-stage solution methodology is proposed. In the first stage, the pipeline is divided into some nodes according to the vision capability of the UAV. In the second stage, the AGASA that introduces heuristic rules is adopted to solve the model. Finally, three algorithms (GA, SA and AGASA) are employed for calculation, and the results show that the AGASA proposed in this paper has great stability and convergence. What is more, it can be applied to the inspection path planning optimization problem of most oil and gas pipeline networks. This paper considered the importance of each inspection node is the same, ignoring the difference in the inspection of high-leakage danger zones, densely populated areas and no-man's land. Therefore, further research can add the requirement for the number of inspection times in each area and solve the optimization problem of inspection path planning.

**Acknowledgements** This work was part of the Program of “Study on Optimization and Supply-side Reliability of Oil Product Supply Chain Logistics System” funded under the National Natural Science Foundation of China, Grant Number 51874325. The authors are grateful to all study participants.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

- Akbaripour H, Masehian E. Semi-lazy probabilistic roadmap: a parameter-tuned, resilient and robust path planning method for manipulator robots. *Int J Adv Manuf Technol.* 2017;89(5):1401–30. <https://doi.org/10.1007/s00170-016-9074-6>.
- Bhattacharya P, Gavrilova ML. Voronoi diagram in optimal path planning. In: 4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007); 2007. p. 38–47. <https://doi.org/10.1109/isvd.2007.43>.
- Candeloro M, Lekkas AM, Sørensen AJ. A Voronoi-diagram-based dynamic path-planning system for underactuated marine vessels. *Control Eng Pract.* 2017;61:41–54. <https://doi.org/10.1016/j.conengprac.2017.01.007>.
- Chen C, Rickert M, Knoll A. Path planning with orientation-aware space exploration guided heuristic search for autonomous parking and maneuvering. In: 2015 IEEE Intelligent Vehicles Symposium (IV); 2015. p. 1148–53. <https://doi.org/10.1109/ivs.2015.7225838>.
- Chen X, Chen X. The UAV dynamic path planning algorithm research based on Voronoi diagram. In: Proceedings of the 26th Chinese Control and Decision Conference (2014 CCDC). Changsha, China; 2014. p. 1069–71. <https://doi.org/10.1109/ccdc.2014.6852323>.
- Gammell JD, Srinivasa SS, Barfoot TD. Batch Informed Trees (BIT\*): sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs. In: Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA). Seattle, WA, USA; 2015. p. 3067–74. <https://doi.org/10.1109/icra.2015.7139620>.
- Garrido S, Moreno L, Abderrahim M, et al. Path planning for mobile robot navigation using Voronoi diagram and fast marching. In: Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. Beijing, China; 2006. p. 2376–81. <https://doi.org/10.1109/iroso.2006.282649>.
- Geraerts R, Overmars MH. Sampling and node adding in probabilistic roadmap planners. *Robot Auton Syst.* 2006;54:165–73. <https://doi.org/10.1016/j.robot.2005.09.026>.
- Gómez C, Green DR. Small unmanned airborne systems to support oil and gas pipeline monitoring and mapping. *Arab J Geosci.* 2017;10(9):202. <https://doi.org/10.1007/s12517-017-2989-x>.
- Guo J, Wang X, Fan S, et al. Forward and reverse logistics network and route planning under the environment of low-carbon emissions: a case study of Shanghai fresh food E-commerce enterprises. *Comput Ind Eng.* 2017;106:351–60. <https://doi.org/10.1016/j.cie.2017.02.002>.
- Hart PE, Nilsson NJ, Raphael B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans Syst Sci Cybern.* 1968;4(2):100–7. <https://doi.org/10.1109/TSSC.1968.300136>.
- Hu Y, Yang SX. A knowledge based genetic algorithm for path planning of a mobile robot. In: Robotics and Automation, 2004. Proceedings. ICRA '04. Proceedings of 2004 IEEE International Conference on Robotics and Automation. New Orleans, LA, USA; 2004;(2). p. 4350–55. <https://doi.org/10.1109/robot.2004.1302402>.
- Li J, Huang Y, Xu Z, et al. Path planning of UAV based on hierarchical genetic algorithm with optimized search region. In: Proceedings of 2017 13th IEEE International Conference on Control & Automation (ICCA). Ohrid, Macedonia; 2017. p. 1033–8. <https://doi.org/10.1109/icca.2017.8003203>.
- Nazarahari M, Khanmirza E, Doostie S. Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Syst Appl.* 2019;115:106–20. <https://doi.org/10.1016/j.eswa.2018.08.008>.
- Patle BK, Parhi DRK, Jagadeesh A, et al. Matrix-Binary Codes based Genetic Algorithm for path planning of mobile robot. *Comput Electr Eng.* 2018;67:708–28. <https://doi.org/10.1016/j.compeleceng.2017.12.011>.
- Reddy HP, Narasimhan S, Bhallamudi SM, et al. Leak detection in gas pipeline networks using an efficient state estimator. Part-I: theory and simulations. *Comput Chem Eng.* 2011;35(4):651–61. <https://doi.org/10.1016/j.compchemeng.2010.10.006>.
- Sedighi KH, Ashenayi K, Manikas TW, et al. Autonomous local path planning for a mobile robot using a genetic algorithm. In: Proceedings of the 2004 Congress on Evolutionary Computation (IEEE

- Cat. No. 04TH8753). Portland, OR, USA; 2004. p. 1338–45. <https://doi.org/10.1109/cec.2004.1331052>.
- Shen X, Sang J, Sun Y, et al. Application of improved ant colony algorithm in distribution network patrol route planning. In: Proceedings of 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS). Beijing, China; 2016. p. 560–63. <https://doi.org/10.1109/icse.2016.7883132>.
- Szczerba RJ, Galkowski P, Glicktein IS, et al. Robust algorithm for real-time route planning. *IEEE Trans Aerosp Electron Syst.* 2000;36(3):869–78. <https://doi.org/10.1109/7.869506>.
- Tsai CC, Huang HC, Chan CK. Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation. *IEEE Trans Ind Electron.* 2011;58(3):4813–21. <https://doi.org/10.1109/TIE.2011.2109332>.
- Tu J, Yang SX. Genetic algorithm based path planning for a mobile robot. In: Proceedings of 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422). Taipei, Taiwan (China); 2003. p. 1221–6. <https://doi.org/10.1109/robot.2003.1241759>.
- Wang B, Liang Y, Zheng J, et al. A methodology to restructure a pipeline system for an oilfield in the mid to late stages of development. *Comput Chem Eng.* 2018a;115:133–40. <https://doi.org/10.1016/j.compchemeng.2018.04.008>.
- Wang B, Yuan M, Zhang H, et al. An MILP model for optimal design of multi-period natural gas transmission network. *Chem Eng Res Des.* 2018b;129:122–31. <https://doi.org/10.1016/j.cherd.2017.11.001>.
- Wang Z, Cai J. Probabilistic roadmap method for path-planning in radioactive environment of nuclear facilities. *Prog Nucl Energy.* 2018;109:113–20. <https://doi.org/10.1016/j.pnucene.2018.08.006>.
- Yu J, LaValle SM. Optimal multirobot path planning on graphs: complete algorithms and effective heuristics. *IEEE Trans Robot.* 2016;32(5):1163–77. <https://doi.org/10.1109/TRO.2016.2593448>.
- Zhang H, Liang Y, Liao Q, et al. A hybrid computational approach for detailed scheduling of products in a pipeline with multiple pump stations. *Energy.* 2017a;119:612–28. <https://doi.org/10.1016/j.energy.2016.11.027>.
- Zhang H, Liang Y, Ma J, et al. An MILP method for optimal offshore oilfield gathering system. *Ocean Eng.* 2017b;141:25–34. <https://doi.org/10.1016/j.oceaneng.2017.06.011>.
- Zhang H, Liang Y, Ma J, et al. An improved PSO method for optimal design of subsea oil pipelines. *Ocean Eng.* 2017c;141:154–63. <https://doi.org/10.1016/j.oceaneng.2017.06.023>.
- Zhang H, Liang Y, Zhang W, et al. A unified MILP model for topological structure of production well gathering pipeline network. *J Pet Sci Eng.* 2017d;152:284–93. <https://doi.org/10.1016/j.petro.2017.03.016>.
- Zhang H, Liang Y, Zhang W, et al. Improved PSO-based method for leak detection and localization in liquid pipelines. *IEEE Trans Ind Inf.* 2018a;14(7):3143–54. <https://doi.org/10.1109/TII.2018.2794987>.
- Zhang H, Yuan M, Liang Y, et al. A risk assessment based optimization method for route selection of hazardous liquid railway network. *Saf Sci.* 2018b;110:217–29. <https://doi.org/10.1016/j.ssci.2018.04.003>.