



# Minimum dose path planning for facility inspection based on the discrete Rao-combined ABC algorithm in radioactive environments with obstacles

Kwon Ryong Hong<sup>1</sup> · Su Il O<sup>1</sup> · Ryon Hui Kim<sup>1</sup> · Tae Song Kim<sup>1</sup> · Jang Su Kim<sup>2</sup>

Received: 7 December 2022 / Revised: 12 February 2023 / Accepted: 26 February 2023 / Published online: 17 April 2023

© The Author(s), under exclusive licence to China Science Publishing & Media Ltd. (Science Press), Shanghai Institute of Applied Physics, the Chinese Academy of Sciences, Chinese Nuclear Society 2023

## Abstract

Workers who conduct regular facility inspections in radioactive environments will inevitably be affected by radiation. Therefore, it is important to optimize the inspection path to ensure that workers are exposed to the least amount of radiation. This study proposes a discrete Rao-combined artificial bee colony (ABC) algorithm for planning inspection paths with minimum exposure doses in radioactive environments with obstacles. In this algorithm, retaining the framework of the traditional ABC algorithm, we applied the directional solution update rules of Rao algorithms at the employed bee stage and onlooker bee stage to increase the exploitation ability of the algorithm and implement discretion using the swap operator and swap sequence. To increase the randomness of solution generation, the chaos algorithm was used at the initialization stage. The *K*-opt operation technique was introduced at the scout bee stage to increase the exploration ability of the algorithm. For path planning in an environment with complex structural obstacles, an obstacle detour technique using a recursive algorithm was applied. To evaluate the performance of the proposed algorithm, we performed experimental simulations in three hypothetical environments and compared the results with those of improved particle swarm optimization, chaos particle swarm optimization, improved ant colony optimization, and discrete Rao's algorithms. The experimental results show the high performance of the proposed discrete Rao-combined ABC algorithm and its obstacle detour capability.

**Keywords** Minimum dose · Path planning · Nuclear facility inspection · ABC algorithm · Rao algorithms · Swap sequence · *K*-opt operation

## 1 Introduction

The inspection of nuclear power plants and facilities is essential to prevent accidents and provide a stable and safe working environment. With the rapid development of science and technology, improved radiation protection instruments are being developed and utilized; however, radiation exposure remains inevitable for facility inspection workers working in radioactive environments. In particular, the amount of radiation that workers are exposed to is

considerable when nuclear facilities are overhauled or dismantled. Therefore, it is important to plan the path of facility inspections well to ensure that workers are exposed to the least amount of radiation.

Solving the problem of minimum dose path planning relies primarily on three important techniques: construction of the radiation environment, collision detection, and a search algorithm for the optimal path. The radiation environment was constructed using either a forward method or an inversion method. The forward method is typically applicable when the location and strength of the radiation sources are known. In research on the construction of radiation environments using the forward method, several methods have been proposed in consideration of the geometric shape and type of radiation source and the shielding of radiation because of obstacles. Stochastic methods based on the Monte Carlo technique [1–3] and analytical methods based on the Point Kernel technique [4–6] have attracted

✉ Kwon Ryong Hong  
kwonryonghong@163.com

<sup>1</sup> Faculty of Energy Science, Kim Il Sung University, Pyongyang, Democratic People's Republic of Korea

<sup>2</sup> High-Tech Research & Development Centre, Institute of Information Technology, Kim Il Sung University, Pyongyang, Democratic People's Republic of Korea

considerable attention. The inversion method is used to reconstruct the radiation field from the dose rate values measured in specific places when the radiation sources are unknown. Reconstruction methods using a concentration function [7], net function interpolation [8] and Bayesian inference [9] are noteworthy. However, in this paper, we assume simple environments because the main purpose of this study is to develop an algorithm for planning the path of minimum dose inspection considering obstacles. We assume that the radiation sources are point sources whose locations and strengths are known. The radiation environment is two-dimensional, and the shielding effect of the obstacles is overlooked. Such an environment can easily be constructed using Point Kernel technique, which is an analytical method.

Collision detection techniques vary depending on the type of search algorithm used for path planning. Grid-based algorithms such as Dijkstra and A\* consider obstacles by dividing the radiation-contaminated region into grids and excluding the obstructed grids from the path search [2, 10, 11]. Sampling-based algorithms, such as rapidly-exploring random tree (RRT) and probabilistic roadmap (PRM), consider obstacles as a way to exclude samples that occur within an obstacle zone or collide with obstacles when they are connected with other samples [1, 12, 13]. In meta-heuristic algorithms, the entire path consists of a series of straight segments. If a straight segment passes through an obstacle, then the detour curve replaces the straight segment. Because the cumulative dose of the path is determined by how this detour path is created, the obstacle detour technique considerably influences the optimal path search. Xie et al. [14] abstracted obstacles into rectangles of different sizes to plan the facility inspection path with obstacles. If the straight line between two target points collides with the obstacle rectangle, it avoids the obstacle by creating a curved detour path with the smallest dose, including the vertices of the rectangle. Lai and Smith [15] avoided obstacles by abstracting them as one or more rectangles and moving two nodes of a straight line passing through the obstacles to the free vertices of the obstacle rectangles. Hong et al. [16] proposed a technique for determining the detour path with the smallest dose by recursive circulation, considering the case in which there is another obstacle while bypassing one obstacle. This technique, applied to the minimum dose path planning problem with fixed start and goal points, is suitable for optimal path search in radiation environments with narrow routes or complex structural obstacles. Therefore, we applied this technique to solve the problem of facility

inspection path planning in radiation environments with obstacles.

Unlike path planning, which searches for paths with a minimum dose when a start point and a goal point are given, inspection path planning is a problem of determining the tour path (i.e., target sequence) such that the cumulative dose received by a person during the entire inspection process is minimized under the condition that the inspection places (target points) are fixed. Mathematically, the former is a continuous optimization problem, whereas the latter is a combinatorial optimization problem. Therefore, grid-based algorithms or sampling-based algorithms cannot be used for inspection path-planning problems. The inspection path planning problem using meta-heuristic algorithms has been investigated in several studies. Liu et al. [17] explored the inspection path in an obstacle-free radiation environment using an improved particle swarm optimization (IPSO) algorithm combined with a genetic algorithm. Wang and Cai [18] developed a discrete chaos particle swarm optimization (CPSO) algorithm that combines chaos algorithms to plan inspection paths without considering obstacles. Xie et al. [14] proposed an improved ant colony optimization (IACO) algorithm that considers simple rectangular obstacles.

The inspection path planning problem is similar to the traveling salesman problem (TSP), a representative combinatorial optimization problem. In solving the TSP, it is known that the artificial bee colony (ABC) algorithm is superior to other meta-heuristic algorithms in exploration and exploitation abilities [19–21]. Research on combinatorial optimization using the ABC algorithm has been conducted by many researchers. For example, Karaboga and Gorkemli [22] proposed a discrete ABC algorithm for the combinatorial optimization problem using the greedy subtour mutation operator. Khan and Maiti [23] studied the ABC approach using swap operators, swap sequences, and  $K$ -opt operation techniques to solve the TSP. In this study, we propose a discrete ABC algorithm that combines the directional solution update method of Rao algorithms [24, 25] to plan an inspection path with a minimum dose under a radiation environment with complex obstacles. The remainder of this paper is organized as follows. Section 2 describes in detail the radiation dose map construction, obstacle detection technique, and the newly proposed discrete Rao-combined ABC algorithm. A comparative analysis with other meta-heuristic algorithms is described in Sect. 3, and the conclusions are presented in Sect. 4.

## 2 Inspection path planning method

### 2.1 Construction of radioactive environment

We used an interpolation method based on the radiation dose map to determine the radiation dose rate at any point in the radiation-contaminated region when the location and intensity of the radiation sources are given. First, the radiation dose map was constructed in advance by dividing the contaminated region into  $N \times M$  grids and obtaining the radiation dose rate at each grid point  $(i, j)$  ( $i = 1, 2, \dots, N$ ,  $j = 1, 2, \dots, M$ ). If the radiation environment is composed of several point sources, the radiation dose rate at the grid point  $(i, j)$  is expressed by the following equation when shielding by obstacles is ignored:

$$R_{ij} = \sum_s \frac{A_s}{r_{ij,s}^2}, \quad (1)$$

where  $A_s$  is the strength of the  $s$ th radiation point source and  $r_{ij,s}$  is the distance from the grid point  $(i, j)$  to the  $s$ th source. Next, the dose rate at any point in the contaminated region during the path search process can be obtained by interpolation using the dose map. The dose rate at point  $g$  in grid  $(i, j)$  is obtained using the following interpolation equation:

$$R_g = \frac{S_{ij}R_{i-1,j-1} + S_{i,j-1}R_{i-1,j} + S_{i-1,j}R_{i,j-1} + S_{i-1,j-1}R_{i,j}}{S_{ij} + S_{i,j-1} + S_{i-1,j} + S_{i-1,j-1}}, \quad (2)$$

where  $R_{ij}$  ( $i = 2, \dots, N$ ,  $j = 2, \dots, M$ ) is the dose rate at grids  $(i, j)$  and  $S_{ij}$  is the area of the rectangle with grid point  $(i, j)$  and the given point  $g$  as diagonal vertices.

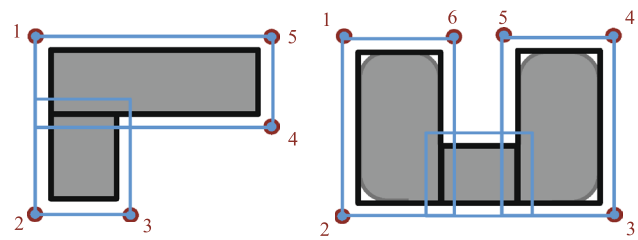
The cumulative dose of a given path can be determined by dividing the path into small segments and summing up the doses in each segment. Here, the dose of each segment was obtained by multiplying the average value of the dose rate at the two nodes of the segment by the duration of the segment:

$$D = \sum_{g=1}^G D_g = \sum_{g=1}^G \left( \frac{R_{g-1} + R_g}{2} \right) \frac{\Delta l_g}{v}, \quad (3)$$

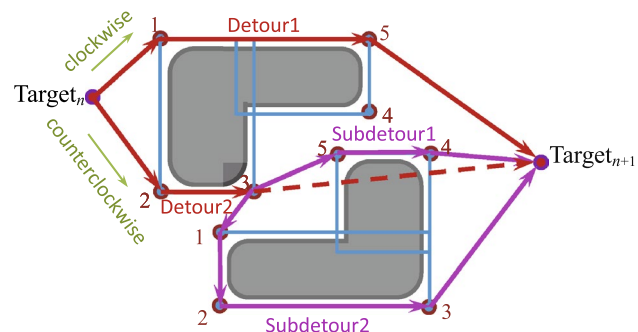
where  $G$  denotes the number of segments,  $D_g$  denotes the cumulative dose in the  $g$ th segment,  $\Delta l_g$  denotes the length of the  $g$ th segment, and  $v$  denotes the human walking speed. In this study, we assumed that the human walking speed is constant at  $v = 1$  m/s.

### 2.2 Obstacle detour technique

If there is an obstacle between two target points, we consider the path between the two target points as a detour curve; otherwise, we consider it as a straight line. The detour curve



**Fig. 1** Modeling obstacles by rectangles and numbering the envelope vertices



**Fig. 2** Obstacle detour scheme

between two target points was obtained using an obstacle detour technique based on Hong et al.'s recursive algorithm [16], considering the case of complex obstacle structures. Any obstacle can be abstracted into one or more rectangles, depending on its shape (black rectangles in Fig. 1). Each rectangle was extended 30 cm outward considering the human body volume (blue rectangles in Fig. 1). Among the four vertices of each extended rectangle, vertices that are not included in the other extended rectangle are identified. These vertices are called envelope points. The envelope points are then numbered counterclockwise for each obstacle.

The obstacle detour path is obtained using the following method (Fig. 2). If an obstacle exists in the path between the two target points, it can be turned in two directions: clockwise and counterclockwise. First, the start point is connected to the nearest envelope point in the direction of detour. Next, the envelope point is connected to the adjacent envelope points in order. This connection continues until it reaches the envelope point nearest to the goal point. The envelope and goal points are connected to complete the detour path. At this time, if there is another obstacle between the target point and the envelope point or between the envelope point and the adjacent envelope point, sub-detour paths are obtained via recursive circulation. Among the detour paths obtained, the path with the

smallest cumulative dose was selected as the detour path between the two target points. For example, in Fig. 2, when the path is detoured clockwise, the detour path (detour1) consists of the start point  $n \rightarrow$  envelope point 1  $\rightarrow$  envelope point 5  $\rightarrow$  goal point  $n + 1$ . Because the counterclockwise detour path (detour2) has another obstacle between envelope point 3 and goal point  $n + 1$ , instead of the straight path, the sub-detour path (subdetour1 or subdetour2) is taken. Eventually, the counterclockwise detour can have two paths. Among the three paths obtained, the path with the smallest dose was selected as the detour path between the target point  $n$  and target point  $n + 1$ . The algorithm for obtaining the obstacle detour path is shown in Algorithm 1 by Hong et al. [16].

### 2.3 Discrete Rao-combined ABC algorithm

In this section, we propose a discrete Rao-combined artificial bee colony (DRABC) algorithm to solve the problem of planning the minimum dose path for nuclear facility inspection. The algorithm combines the framework of traditional ABC algorithms, known for their robustness and convergence, with the Chaos algorithm [26] to further enhance its randomness, Rao's algorithms [24, 25] to further enhance its exploitation, and 3-opt operation [27] to further enhance its exploration. Meanwhile, because the path-planning problem for facility inspection is a combinatorial optimization problem, a discrete meta-heuristic algorithm is required. Therefore, we applied a discrete solution update technique based on the swap operator and swap sequence of Wang et al. [28].

#### 2.3.1 Traditional ABC algorithm and Rao's algorithms

The ABC algorithm is a population-based meta-heuristic algorithm that mimics the foraging characteristics of honeybees [29]. In the ABC algorithm, honeybees are divided into three categories: employed, onlooker, and scout bees. Employed bees search for better food sources around the food source in their memory. Each food source is assigned to a single employed bee. They share their search information with onlooker bees waiting in the hive. Onlooker bees select food sources based on the information provided by employed bees and explore new food sources. Therefore, the more profitable the food source, the more onlooker bees fly in. The employed bee, which has failed to find better food sources for a long time, abandons it and becomes a scout bee, randomly searching for new food sources. The general algorithmic structure of ABC optimization is given below:

Initialization stage  
**Repeat**  
 Employed bee stage  
 Onlooker bee stage  
 Scout bee stage  
 Memorize the best solution achieved so far  
**Until** termination criteria is satisfied

Rao proposed the Jaya algorithm [24] and three Rao algorithms [25] as meta-heuristic algorithms for optimization problems of continuous variables. Rao's algorithms are simple and metaphor-less, especially parameter-less. In Rao's algorithms, each solution in the population is updated by interactions with the best, worst, and randomly chosen solutions in the population. Each solution  $X_p$  in the population is updated according to the following equation:

$$\text{(Jaya)} \quad X'_p = X_p + r_1(X_b - |X_p|) - r_2(X_w - |X_p|), \quad (4)$$

$$\text{(Rao1)} \quad X'_p = X_p + r_1(X_b - X_w), \quad (5)$$

$$\text{(Rao2)} \quad X'_p = X_p + r_1(X_b - X_w) + r_2(|X_p \text{ or } X_q| - |X_q \text{ or } X_p|), \quad (6)$$

$$\text{(Rao3)} \quad X'_p = X_p + r_1(X_b - |X_w|) + r_2(|X_p \text{ or } X_q| - (X_q \text{ or } X_p)), \quad (7)$$

where  $r_1$  and  $r_2$  are random numbers between  $[0,1]$ , and  $X_b$ ,  $X_w$  and  $X_q$  ( $q \in \{1, 2, \dots, SN\}$ ,  $SN$ : population size) are the best, worst, and randomly chosen solutions in the population, respectively. The symbol  $|\cdot|$  indicates the absolute value. If the solution  $X_p$  is better than the solution  $X_q$ , then the term  $(X_p \text{ or } X_q)$  indicates  $X_p$ ; otherwise, it indicates  $X_q$ . As evidenced from Eqs. (4)–(7), the updated solution approaches the best solution and moves away from the worst solution. To increase the exploration, Rao2 and Rao3 algorithms also include interactions with other randomly chosen solutions. If the objective value of the randomly chosen solution is better than that of the original solution, the updated solution approaches the randomly chosen solution, and if worse, moves away from it, ensuring both exploration and exploitation. It is emphasized that Rao2 and Rao3 become the same if all the variables have positive values.

#### 2.3.2 Swap operator and swap sequence

The concepts of the swap operator and swap sequence were proposed by Wang et al. to solve the TSP using the PSO algorithm [28]. Khan and Maiti used these operations in the ABC algorithm to solve TSP [23]. Consider the solution  $X = (x_1, x_2, \dots, x_K)$  of TSP, which consists of  $K$  positive integer elements. Let  $V = \{1, 2, \dots, K\}$ ; then  $x_k \in V$  and

$x_k \neq x_l$  for  $\forall k \neq l$ . The swap operator  $SO(k, l)$  is defined as the swap of element  $x_k$  and element  $x_l$  in solution  $X$ . When the solution obtained by the swap operator is  $X'$ , this operation is denoted as  $X' = X \triangleleft SO(k, l)$ . Here, the symbol  $\triangleleft$  represents a binary swap operator. For example, the new solution obtained when applying the swap operator  $SO(2, 4)$  to solution  $X = (x_1, x_2, x_3, x_4, x_5) = (3, 1, 5, 2, 4)$  becomes  $X' = X \triangleleft SO(2, 4) = (3, 1, 5, 2, 4) \triangleleft SO(2, 4) = (3, 2, 5, 1, 4)$ . That is, the second element of  $X$ , 1, and the fourth element, 2, are interchanged.

A collection of different swap operators on a particular sequence of a solution is called a swap sequence, and is denoted by  $SS = (SO_1, SO_2, \dots, SO_n)$ . Here,  $SO_1, SO_2, \dots, SO_n$  are the swap operators. When a swap sequence acts on a solution, all swap operators in the swap sequence act on the solution in turn. That is,

$$\begin{aligned} X' &= X \triangleleft SS = X \triangleleft (SO_1, SO_2, \dots, SO_n) \\ &= (\dots ((X \triangleleft SO_1) \triangleleft SO_2) \dots \triangleleft SO_n). \end{aligned}$$

When different swap sequences are applied to the same solution, if the same new solution is obtained, the swap sequences are called the equivalent set of swap sequences, and the swap sequence with the smallest number of swap operators is called the basic swap sequence.

The following operators can be defined with respect to the swap sequence.

#### Plus operator of the two swap sequences: $\oplus$

Let solution  $X'$  be obtained when the swap sequence  $SS_1$  is applied; then,  $SS_2$  acts on solution  $X$ . At this time, if there exists a swap sequence  $SS'$  that generates solution  $X'$  when applied to solution  $X$ , the swap sequence  $SS'$  is called the addition of  $SS_1$  and  $SS_2$ , denoted by  $SS' = SS_1 \oplus SS_2$ .

#### Negative operator between the two solutions: $\ominus$

When the basic swap sequence acting on solution  $X$  to obtain solution  $X'$  is denoted as  $SS$ , the swap sequence  $SS$  is called the difference between solution  $X$  and solution  $X'$  and is denoted by  $SS = X' \ominus X$ .

#### Multiply operator of random numbers and swap sequence: $\odot$

Multiplying the swap sequence  $SS$  by a random number  $r$  implies that all swap operators that make up the exchange sequence  $SS$  are maintained with probability  $r$ . When swap sequence  $SS$  acts on solution  $X$ , each swap operator is chosen with probability  $r$ . The resulting swap sequence  $SS'$  is denoted as  $SS' = r \odot SS$ . Each swap operator in swap sequence  $SS'$  is a collection of swap operators chosen with probability  $r$  among the swap operators of  $SS$ .

### 2.3.3 The proposed algorithm

The discrete Rao-combined ABC algorithm proposed in this study is a hybrid algorithm that combines solution generation using the chaos algorithm in the initialization stage, solution update using discrete Rao's algorithms in the employed bee and onlooker bee stages, and solution exploration using the 3-opt perturbation technique in the scout stage while maintaining the framework of traditional ABC algorithms. The proposed DRABC algorithm is shown in **Algorithm 1**.

#### Initialization stage

Population initialization was performed using the chaos algorithm. The chaos algorithm exhibits high randomness, ergodicity, and regularity. Therefore, initializing a population using a chaos map can increase the population diversity and accelerate convergence. The following piecewise logistic mapping equation was used to generate a chaotic sequence [26]:

$$cx^{t+1} = \begin{cases} 4\mu \cdot cx^t \cdot (0.5 - cx^t), & 0 < cx^t < 0.5 \\ 1 - 4\mu \cdot cx^t \cdot (cx^t - 0.5)(1 - cx^t), & 0.5 \leq cx^t < 1 \end{cases} \quad (8)$$

where  $cx^t$  is the  $t$ -th generated chaos variable vector, and  $\mu$  is a constant with a value of 4. The chaotic variables generated by Eq. (8) are distributed with ergodicity, randomness, and regularity in the search space. Using Eq. (8), we generate the initial solutions of the population as follows. First, a  $K$ -dimensional random number vector  $cx^0$  whose elements are placed in the interval  $[0, 1]$ , is generated. Next, chaos variable vectors are generated according to Eq. (8). This process is repeated using the maximum number of iterations of the chaotic algorithm. Next, when arranging each element in incremental order for each chaos variable vector, an index vector consisting of the indices of the elements to be arranged is obtained. For example, if the chaotic variable vector generated for  $K = 5$  is  $cx = (0.4527, 0.9548, 0.5311, 0.0215, 0.1246)$ , the smallest value of 0.0215 becomes the first element when it is arranged in incremental order. Thereafter, the first element of the index vector has an index of four. When the second smallest, 0.1246, is organized, it becomes the second element; thus, the second element of the index vector is given an index of five. If all the elements are organized in this manner, the index vector  $X = (4, 5, 1, 3, 2)$  can be obtained. Next, a tour path of the corresponding targets was constructed for each index vector, and the cumulative dose of the path was obtained. In the example above, the corresponding tour path is  $4 \rightarrow 5 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 4$ . Finally, the smallest cumulative dose of  $SN$  tour paths was taken as the initial solution of the population.



**Algorithm 1:** Discrete Rao-combined ABC algorithm**Input:** positions of targets and obstacles,

positions and strengths of radiation sources

**Output:** target sequence and dose value of best path

---

```

01: generate the radiation dose rate map
02: calculate the doses between any two targets taking into account obstacles
03: chaotically initialize the population using Eq. (8)                                /* Initialization
04: while termination criterion is not satisfied                                    stage */
05:   select one of the solution update rules (9)–(11) using Eq. (12)              /* Employed bee
06:   create a new path according to the selected update rule and calculate its cumulative dose  stage */
07:   if the dose of new path is smaller than old one
08:     update the path with new one
09:   else
10:     keep the old path
11:   end if
12:   evaluate the selection probability of each path using Eqs. (13) and (14)      /* Onlooker bee
13:   select the path probabilistically                                              stage */
14:   select one of the solution update rules (9)–(11) using Eq. (12)
15:   create a new path according to the selected update rule and calculate its cumulative dose
16:   if the dose of new path is smaller than old one
17:     update the path with new one
18:   else
19:     keep the old path
20:   end if
21:   if there is an abandoned path (there is a path that has not been updated limits times) /* Scout bee
22:     generate new nodes using Algorithm 2                                         stage */
23:     if target sequence of new path is equal to old one
24:       create a new path randomly
25:     end if
26:     replace new path with old one and calculate its cumulative dose
27:   end if
28:   store the best path with the smallest dose among the paths in the population
29: end while

```

---

*Employed bee stage*

In the traditional ABC, each solution in the population is updated by interacting with another randomly chosen solution. The probabilities of the randomly chosen solutions being good and bad are the same; it cannot be established that the new solution is better than the original. Consequently, traditional ABC does not have a very high exploitation ability. DRABC uses Rao's solution update methods instead of the previous ABC update method to further enhance exploitation. Jaya algorithm (4) and Rao algorithms (5)–(7) use directional solution updates, such that the new solution approaches the best, moves away from the worst, and if the randomly chosen solution is better than the original, it approaches it, and if it is worse, it moves away from it.

The solution update rules based on Eqs. (4)–(7) are expressed using the swap operator and the swap sequence as follows.

$$(\text{disJaya}) \quad X'_p = X_p \triangleleft (r_1 \odot (X_b \ominus X_p) \oplus r_2 \odot (X_p \ominus X_w)), \quad (9)$$

$$(\text{disRao1}) \quad X'_p = X_p \triangleleft (r_1 \odot (X_b \ominus X_w)), \quad (10)$$

$$(\text{disRao2, 3}) \quad X'_p = X_p \triangleleft (r_1 \odot (X_b \ominus X_w) \oplus r_2 \odot ((X_p \text{ or } X_q) \ominus (X_q \text{ or } X_p))). \quad (11)$$

As each element of the solution is a positive integer, the absolute value signs in Eqs. (4)–(7) become meaningless, and Rao2 and Rao3 algorithms become the same. In the employed bee stage of the DRABC algorithm, all solutions in the population are updated with a uniform probability according to Eqs. (9)–(11). If the obtained new solution  $X'_p$  is better than the original solution  $X_p$ , the  $p$ -th solution in the population is replaced by  $X'_p$ ; otherwise, it maintains the original solution  $X_p$ . In the DRABC algorithm, the solution is updated by selecting one of the three rules in every iteration of Eqs. (9)–(11). The rule selection method is as follows: each rule has a counter with an initial value of 1.

If there is an improvement in the solution after a rule is selected and used to generate a new solution, the counter value of the rule is increased by one. The selection of the rule is performed according to the roulette wheel selection method with its counter value. When the counter value of the  $m$ th rule is  $v_m$ , the selection probability  $\rho_m$  of the rule is obtained by the following equation:

$$\rho_m = \frac{v_m}{\sum_{m=1}^3 v_m}, \quad (m = 1, 2, 3). \quad (12)$$

#### Onlooker bee stage

Each solution in the population is updated/maintained with a uniform probability in the employed bee stage. However, in the onlooker bee stage, the solutions (food sources) are updated/maintained with a probability associated with their fitness values (honey amounts). Therefore, the larger the fitness value, the more it is updated or maintained. The update method for the solution is the same as that in the employed bee stage. For the  $p$ th solution, the probability  $w_p$  selected by the onlooker bee is as follows:

$$w_p = \frac{fit_p}{\sum_{p=1}^{SN} fit_p}, \quad (p = 1, 2, \dots, SN), \quad (13)$$

where  $fit_p$  is the fitness value of the  $p$ -th solution. For the minimization problem, when the objective function is  $f$ , it can be expressed as

$$fit_p = \begin{cases} 1/(1 + f_p), & \text{if } f_p \geq 0 \\ 1 + |f_p|, & \text{otherwise} \end{cases} \quad (14)$$

When the selection probability of each solution is determined, the solution to be updated/maintained by the onlooker bee is selected according to the roulette wheel selection method.

#### Scout bee stage

In the scout bee stage, solutions that are not updated for a preset iteration number (called *limits*) are replaced with randomly generated new solutions to prevent solutions from falling into local optimization and to find global optimization. This guarantees the exploration of the algorithm. For this purpose, each solution of the population has a counter. If the solution improves in the stages of employed bees and onlooker bees, the counter value becomes zero; otherwise, the value increases by one. If this counter value is equal to *limits*, the solution is considered abandoned because it has not been updated *limits* times, and a new solution is generated and replaced. In the proposed DRABC algorithm, we introduce the 3-opt perturbation technique when replacing abandoned solutions, attempting to replace them with new solutions that are better than the original solution. The 3-opt operation is conducted for a preset iteration number (denoted as *MaxOptIter*) in the scout bee stage of DRABC to update the abandoned solution. If any improvement is made, the abandoned solution is replaced with an improved solution, and the counter value of the solution is set to zero. Otherwise, as in traditional ABC, a new solution is randomly generated to replace the abandoned solution, and its counter value is set to zero.

---

#### Algorithm 2: 3-opt operation algorithm

---

**Input:** abandoned path, *MaxOptIter*

**Output:** target sequence of best path

---

01: *iter* = 1

02: **while** *iter* ≤ *MaxOptIter*

03:   remove three edges from target sequence of abandoned path, then it makes 3 sub-paths *subtour1*, *subtour2*, and *subtour3*

04:   reverse the contents of three sub-paths, then it makes three new sub-paths *revtour1*, *revtour2*, and *revtour3*

05:   combine these sub-paths to create new paths as follows:

06:   {*subtour1*, *revtour2*, *subtour3*}

07:   {*subtour1*, *subtour2*, *revtour3*}

08:   {*subtour1*, *revtour3*, *revtour2*}

09:   {*subtour1*, *subtour3*, *revtour2*}

10:   {*subtour1*, *revtour3*, *subtour2*}

11:   {*subtour1*, *revtour2*, *revtour3*}

12:   {*subtour1*, *subtour3*, *subtour2*}

13:   store the best path with the smallest dose among seven new paths and the abandoned path

14:   *iter* = *iter* + 1

15: **end while**

16: output the path with the smallest dose among the *MaxOptIter* stored paths

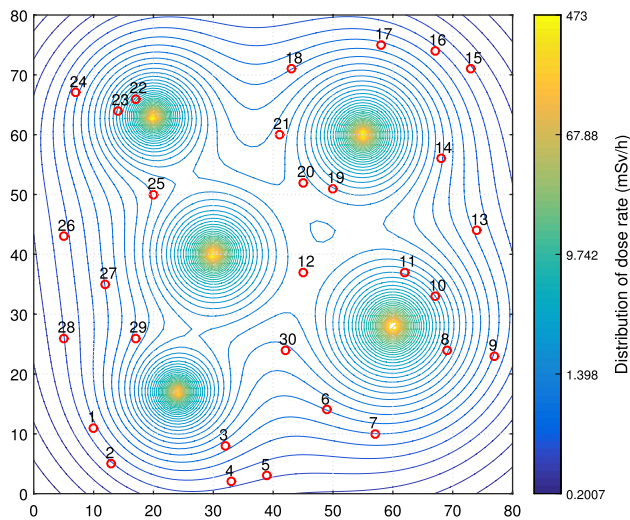
---

**Table 1** The locations and strengths of five radiation point sources in Case 1

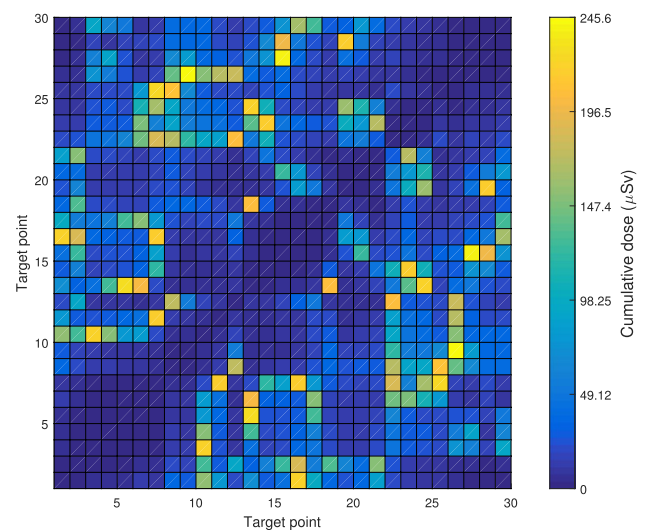
Serial number		1	2	3	4	5
Location	$x$ (m)	24	30	20	60	55
	$y$ (m)	17	40	63	28	60
Strength		$A$ ( $\mu\text{Sv} \cdot \text{m}^2/\text{s}$ )				
		15	30	20	40	30

**Table 2** The locations of 30 targets in Case 1

Target		1	2	3	4	5	6	7	8	9	10
Location	$x$ (m)	10	13	32	33	39	49	57	69	77	67
	$y$ (m)	11	5	8	2	3	14	10	24	23	33
Target		11	12	13	14	15	16	17	18	19	20
Location	$x$ (m)	62	45	74	68	73	67	58	43	50	45
	$y$ (m)	37	37	44	56	71	74	75	71	51	52
Target		21	22	23	24	25	26	27	28	29	30
Location	$x$ (m)	41	17	14	7	20	5	12	5	17	42
	$y$ (m)	60	66	64	67	50	43	35	26	26	24

**Fig. 3** (Color online) The hypothetical environment of Case 1

A new solution generation algorithm using a 3-opt operation is presented in **Algorithm 2**. The abandoned solution (target sequence) is arbitrarily decomposed into three sub-paths, and new solutions are created by recombining these sub-paths and their inverse sub-paths. Among the seven possible new target sequences, the sequence with the smallest cumulative dose was selected as the new solution. If the new solution has a lower cumulative dose than the abandoned solution, the abandoned solution is replaced with the new solution.

**Fig. 4** (Color online) Cumulative doses of straight paths between any two targets in Case 1

### 3 Experimental simulation

We simulated three hypothetical radioactive environments to demonstrate the performance and obstacle detouring capabilities of the proposed algorithm. The first hypothetical case was a relatively simple environment with no obstacles and a small number of targets and radiation sources. In this case, the proposed DRABC algorithm was compared with other inspection path planning algorithms, namely IPSO [17], CPSO [18], and IACO [14]. To further demonstrate the performance of the



Table 3 Optimal values of specific parameters for CPSO, IACO and DRABC

Algorithms	Parameters		Full name	Level			Optimal value
	Symbol			level 1	level 2	level 3	
CPSO	<i>MaxChaosIter</i>		Maximum number of chaos iteration	20	50	100	100
	$\alpha$		Pheromone importance factor	1	2	3	2
	$\beta$		Heuristic function factor	2	3	5	5
	$\rho$		Pheromone volatilization coefficient	0.1	0.3	0.5	0.3
	$Q$		Total pheromone amount	1	5	10	1
DRABC	$k$		Adjustable parameter for local search	3	5	7	5
	<i>Limits</i>		Limits for scout stage	10	20	30	20
	<i>MaxOptIter</i>		Maximum number of 3-opt iteration	5	7	10	10

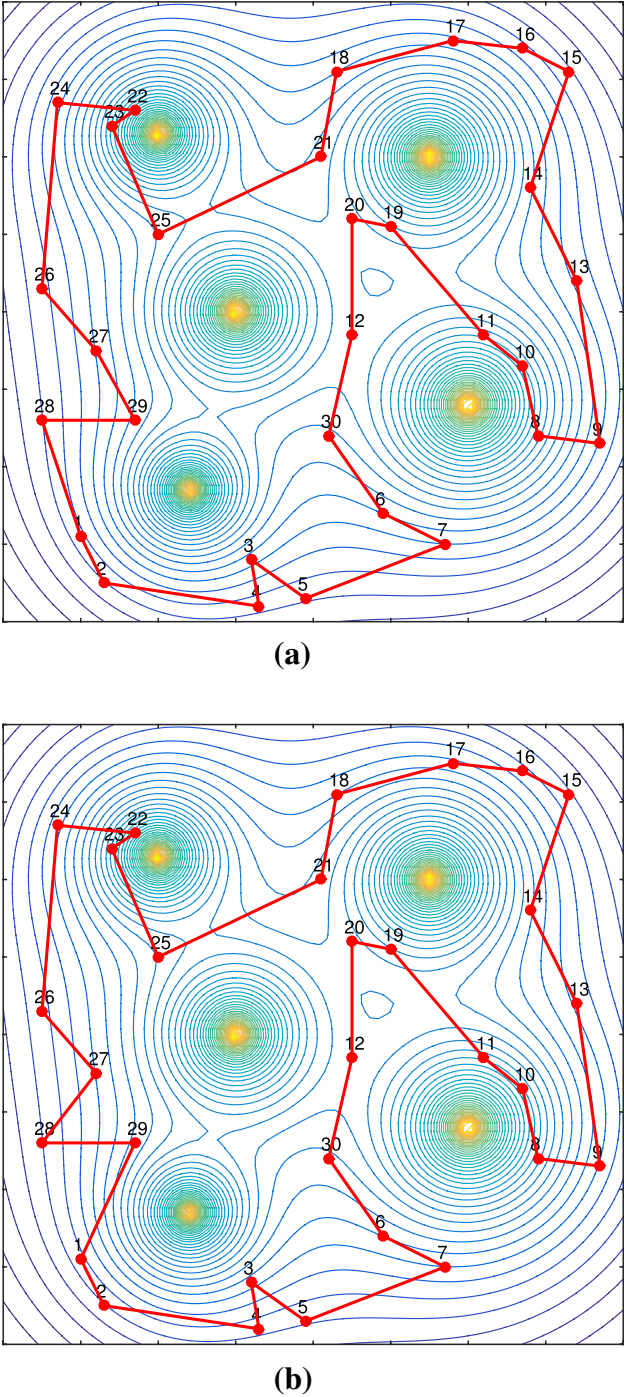


Fig. 5 (Color online) The best paths obtained by four algorithms in Case 1. (a) IPSO, IACO and DRABC; (b) CPSO

algorithm in complex environments, the second hypothetical case considered an environment with several rectangular obstacles and a large number of targets. The proposed algorithm was compared to the IACO algorithm [14]. To demonstrate the high obstacle detour capability of the proposed algorithm, the third hypothetical case simulated an environment with

**Table 4** Experimental results of four meta-heuristic algorithms in Case 1

Algorithm	Worst value ( $\mu\text{Sv}$ )	Average value ( $\mu\text{Sv}$ )	Best value ( $\mu\text{Sv}$ )	Proportion of optimal value (%)	Less than 100 (%)	More than 100 (%)	Average cpu time (sec)
IPSO	149.2232	113.5752	94.8678	4	10	90	20.552
CPSO	155.8636	122.6496	95.8223	2	4	96	20.394
IACO	111.4076	99.7814	94.8678	28	64	36	13.107
DRABC	98.6882	96.1210	94.8678	42	100	0	12.872

many complex structural obstacles and radiation sources. In this case, we also highlighted the superiority of the proposed algorithm via a comparison with the discrete Jaya algorithm (9) and Rao algorithms (10)–(11).

### 3.1 Case 1

Figure 3 shows the  $80\text{ m} \times 80\text{ m}$  hypothetical environment of Case 1. The locations and strengths of the five radiation sources and 30 targets are listed in Tables 1 and 2, respectively. Figure 4 shows the cumulative doses of straight paths between any two targets in Case 1. The radiation map was obtained according to the description in Sect. 2.1. The grid interval for the radiation map was set to 1.1 m.

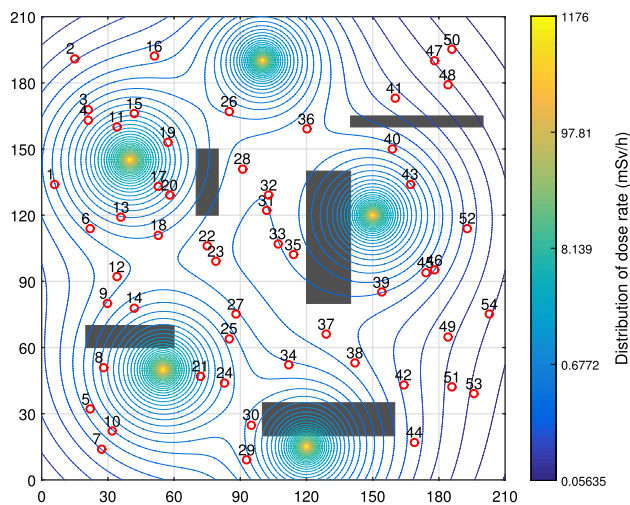
For all algorithms used in the comparison, the population size and termination criterion for the iterations were fixed. The population size ( $SN$ ) is set to 50. We used the maximum number of function evaluations ( $MaxNFEs$ ) as the termination criterion. Here, the number of function evaluations was the number of cumulative dose calculations for the entire inspection path. For Case 1, we set  $MaxNFEs = 150,000$ . To optimally set the algorithm-specific parameters of the CPSO, IACO, and DRABC algorithms, we applied Taguchi's three-level experimental design method [30]. According to the number of specific parameters,  $L^3$  design for CPSO,  $L^9$  for DRABC, and  $L^{27}$  for IACO were performed. The optimal parameter values of the three algorithms obtained by the statistical analysis of 20 iterations for each experiment are listed in Table 3.

**Table 5** The locations and strengths of five radiation point sources in Case 2

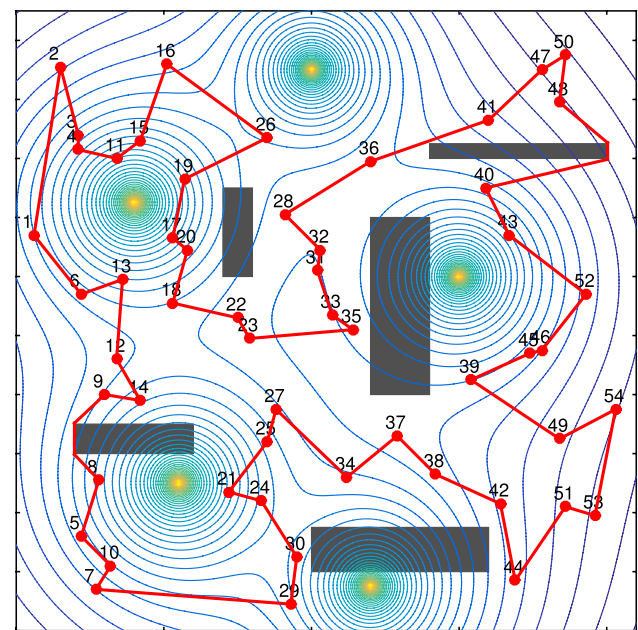
Serial number		1	2	3	4	5
Location	$x$ (m)	150	120	100	40	55
	$y$ (m)	120	15	190	145	50
Strength	$A$ ( $\mu\text{Sv} \cdot \text{m}^2/\text{s}$ )	50	60	40	80	80

**Table 6** The locations of 54 targets in Case 2

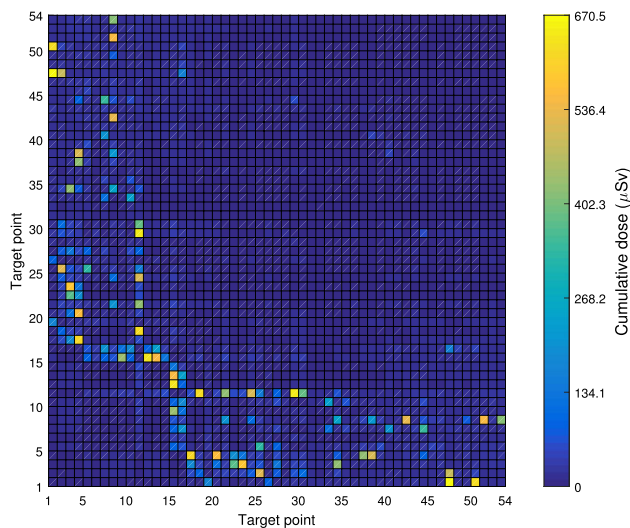
Target		1	2	3	4	5	6	7	8	9
Location	$x$ (m)	6	15	21	21	22	22	27	28	30
	$y$ (m)	134	191	168	163	32	114	14	51	80
Target		10	11	12	13	14	15	16	17	18
Location	$x$ (m)	32	34	34	36	42	42	51	53	53
	$y$ (m)	22	160	92	119	78	166	192	133	111
Target		19	20	21	22	23	24	25	26	27
Location	$x$ (m)	57	58	72	75	79	83	85	85	88
	$y$ (m)	153	129	47	106	99	44	64	167	75
Target		28	29	30	31	32	33	34	35	36
Location	$x$ (m)	91	93	95	102	103	107	112	114	120
	$y$ (m)	141	9	25	122	129	107	52	102	159
Target		37	38	39	40	41	42	43	44	45
Location	$x$ (m)	129	142	154	159	160	164	167	169	174
	$y$ (m)	66	53	85	150	173	43	134	17	94
Target		46	47	48	49	50	51	52	53	54
Location	$x$ (m)	178	178	184	184	186	186	193	196	203
	$y$ (m)	95	190	179	65	195	42	114	39	75



**Fig. 6** (Color online) The hypothetical environment of Case 2



**Fig. 8** (Color online) The optimal path obtained by IACO and DRABC algorithms in Case 2



**Fig. 7** (Color online) Cumulative doses of straight paths between any two targets in Case 2

We obtained the minimum dose inspection path for Case 1 by iterating each algorithm 50 times with the optimal specific parameters. Figure 5 shows the best paths obtained by the four algorithms, and Table 4 shows the experimental results.

The experimental results show that the cumulative dose of the best path generated by CPSO is higher than those generated by the other algorithms. The IPSO, IACO, and DRABC algorithms identify the optimal path with a cumulative dose of 94.8678  $\mu\text{Sv}$  but differ from each other in the search ratio of the optimal path. In terms of calculation time, DRABC is similar to IACO but approximately 1.6 times faster than IPSO or CPSO. From Table 4, it can be observed that the DRABC algorithm is superior to the other three algorithms for all metrics.

### 3.2 Case 2

Figure 6 shows the 210 m  $\times$  210 m hypothetical environment of Case 2 with 5 rectangular obstacles. Table 5 lists the locations and strengths of the five radiation sources, and Table 6 lists the locations of the 54 targets. If there is an obstacle between the two targets, the path is not a straight line; however, a detour path obtained using the technique described in Sect. 2.2. Figure 7 shows the cumulative doses

**Table 7** Experimental results of IACO and DRABC algorithms in Case 2

Algorithm	Worst value ( $\mu\text{Sv}$ )	Average value ( $\mu\text{Sv}$ )	Best value ( $\mu\text{Sv}$ )	Proportion of optimal value (%)	Less than 105 (%)	More than 105 (%)	Average cpu time (sec)
IACO	116.1915	105.9539	100.8149	1	43	57	124.614
DRABC	103.6477	101.2941	100.8149	50	100	0	119.540

**Table 8** The locations and strengths of 19 radiation point sources in Case 3

Serial number		1	2	3	4	5	6	7
Location	$x$ (m)	40	90	190	265	320	90	160
	$y$ (m)	320	320	320	285	290	250	250
Strength	$A$ ( $\mu\text{Sv} \cdot \text{m}^2/\text{s}$ )	6	6	12	18	18	12	6
Serial number		8	9	10	11	12	13	14
Location	$x$ (m)	190	260	30	30	110	175	240
	$y$ (m)	250	250	110	230	210	170	130
Strength	$A$ ( $\mu\text{Sv} \cdot \text{m}^2/\text{s}$ )	6	6	12	18	18	36	12
Serial number		15	16	17	18	19		
Location	$x$ (m)	320	320	100	175	250		
	$y$ (m)	110	230	90	90	90		
Strength	$A$ ( $\mu\text{Sv} \cdot \text{m}^2/\text{s}$ )	24	12	6	6	3		

of the paths between any two targets identified by considering the obstacles.

In Case 2, the performances of the DRABC and IACO algorithms were compared. The common parameters were set as  $SN = 100$  and  $MaxNFEs = 10^6$ . The specific parameters for each algorithm are listed in Table 3. The experiment was performed 100 times for each algorithm. The results are listed in Table 7.

From the experiments, we know that the cumulative dose of the optimal path for Case 2 is  $100.8149 \mu\text{Sv}$ . Figure 8 shows the optimal path obtained. Table 7 shows the difference between the proposed and IACO algorithms. For the DRABC algorithm, 100% of the experiments obtained paths below  $105 \mu\text{Sv}$ , of which 50% provided an optimal path. However, for the IACO algorithm, 43% of the experiments obtained paths below  $105 \mu\text{Sv}$ , and only one provided the optimal path. In terms of the calculation time, the DRABC algorithm was slightly faster than the IACO algorithm. Experiments showed that the superiority of the proposed algorithm is more pronounced in large-scale problems with obstacles.

### 3.3 Case 3

Case 3 shows the high-path search and obstacle detour capabilities of the proposed algorithm in the  $350 \text{ m} \times 350 \text{ m}$  hypothetical environment with many radiation sources and obstacles

of complex structures. A hypothetical environment is shown in Fig. 9. Tables 8 and 9 show the locations and strengths of 19 radiation sources and 33 target points, respectively. The cumulative doses on the path between the two target points considering the obstacle detour are shown in Fig. 10.

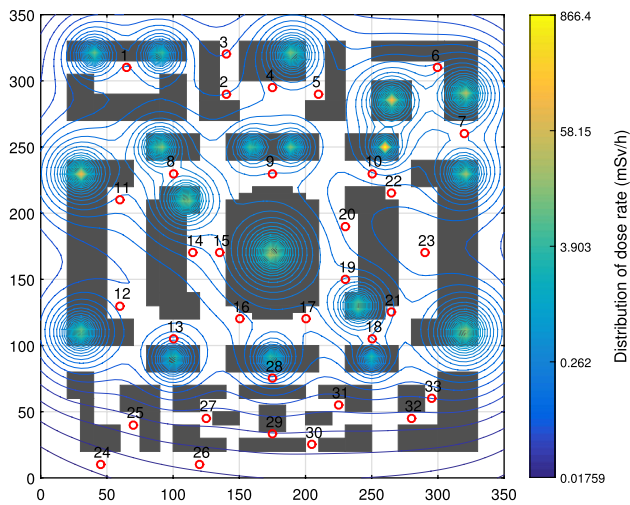
In Case 3, the DRABC algorithm was compared with the disJaya, disRao1, and disRao2 algorithms, which are discrete versions of Jaya and Rao algorithms that are not combined with ABC. The common parameters for Case 3 were set to  $SN = 100$  and  $MaxNFEs = 2 \times 10^6$ . The experiment was repeated 20 times.

The experimental results are listed in Table 10. As shown in Table 10, in the DRABC algorithm, 65% of the simulations yielded an optimal path (with an accumulated dose of  $55.7721 \mu\text{Sv}$ ), and all experiments provided solutions that were very close to the optimal path (standard deviation  $0.2685 \mu\text{Sv}$ ). Figure 11 shows the best and worst paths identified by the DRABC algorithm. From the two figures in Fig. 11, we can see that the obtained paths bypass the various shapes of obstacles that appear in layers very well. Meanwhile, the discrete Jaya and Rao algorithms do not find an optimal path in the set of common parameters. The best result is found with the disRao1 algorithm; however, this result is inferior to the worst result of the DRABC algorithm. In terms of calculation time, the disRao1 algorithm was faster than the other algorithms. Experiments on Case 3

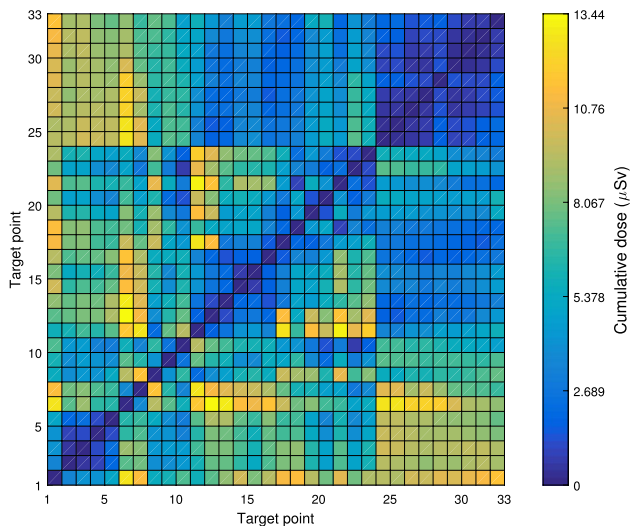
**Table 9** The locations of 33 targets in Case 3

Target		1	2	3	4	5	6	7	8	9	10	11
Location	$x$ (m)	65	140	140	175	210	300	320	100	175	250	60
	$y$ (m)	310	290	320	295	290	310	260	230	230	230	210
Target		12	13	14	15	16	17	18	19	20	21	22
Location	$x$ (m)	60	100	115	135	150	200	250	230	230	265	265
	$y$ (m)	130	105	170	170	120	120	105	150	190	125	215
Target		23	24	25	26	27	28	29	30	31	32	33
Location	$x$ (m)	290	45	70	120	125	175	175	205	225	280	295
	$y$ (m)	170	10	40	10	45	75	33	25	55	45	60





**Fig. 9** The hypothetical environment of Case 3

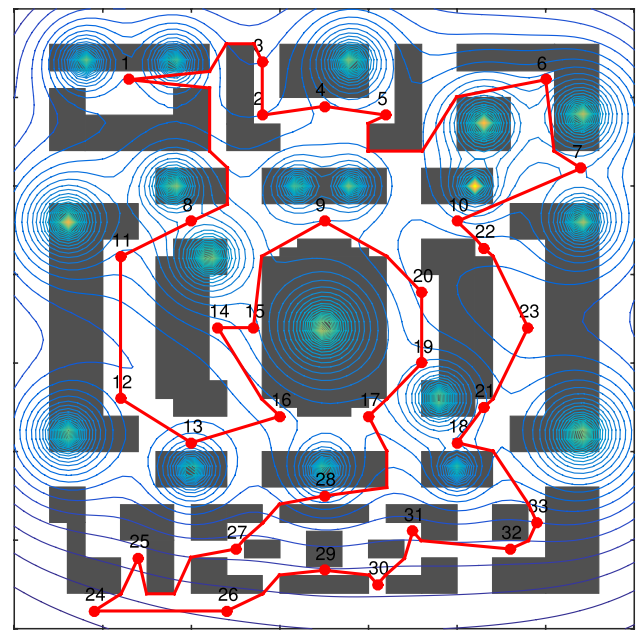


**Fig. 10** Cumulative doses of straight paths between any two targets in Case 3

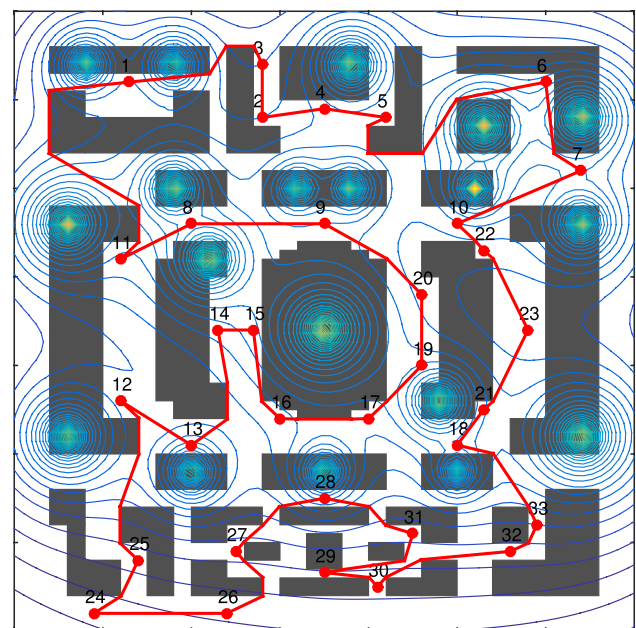
show that the discrete Jaya and Rao algorithms, when combined with the ABC algorithm, improve exploration and exploitation abilities without increasing computational time. Experiments also show that the proposed DRABC algorithm facilitates optimal path exploration, even in radiation environments with complex structural obstacles.

## 4 Conclusion

In this study, a DRABC algorithm was proposed for planning inspection paths in radiation environments with complex structural obstacles. [16] proposed a continuous



**(a)**



**(b)**

**Fig. 11** (Color online) The best and worst paths identified by the DRABC algorithm in Case 3. (a) The best path; (b) The worst path

Rao-combined ABC algorithm that combined Rao's directional solution update rules with the traditional ABC algorithm to enhance its exploitation ability. To plan the inspection path, we studied a discrete version of the algorithm proposed by [16]. To this end, the concepts of the



**Table 10** Experimental results of the disJaya, disRao1, disRao2 and DRABC algorithms in Case 3

Algorithm	Worst value ( $\mu\text{Sv}$ )	Average value ( $\mu\text{Sv}$ )	Best value ( $\mu\text{Sv}$ )	Standard deviation ( $\mu\text{Sv}$ )	Proportion of optimal value (%)	Average cpu time (sec)
disJaya	92.4343	84.5328	70.1031	5.6038	0	130.732
disRao1	74.4286	65.8043	58.0470	4.1845	0	87.999
disRao2	78.1814	66.8099	58.5682	5.7868	0	149.950
DRABC	56.4413	55.9245	55.7721	0.2685	65	147.583

swap operator and swap sequence were applied to develop discrete Jaya and Rao algorithms, and accordingly, the solutions were updated in the employed bee and onlooker bee stages. In addition, the chaos algorithm was used during the initialization stage to increase the randomness of the solution generation, and the 3-opt operation technique was used during the scout bee stage to increase the exploration ability of the algorithm. Three hypothetical radiation environments were simulated to test the performance of the proposed algorithm. In the first hypothetical environment (Case 1), with five radiation sources, 30 target points, and no obstacles, the DRABC algorithm was compared with the IPSO, CPSO, and IACO algorithms. The experimental results showed that three algorithms, excluding CPSO, found the optimal path under the given conditions; however, the DRABC algorithm had the highest search proportion. To further demonstrate the performance of the proposed algorithm, a more complex hypothetical environment (Case 2) with five radiation sources, 54 target points, and five rectangular obstacles was simulated. A comparison with the IACO algorithm showed that the DRABC algorithm was superior in the searchability of the optimal path or convergence of the algorithm. The simulation of a hypothetical environment (Case 3) with 19 radiation sources and 33 target points showed that the proposed algorithm identified optimal paths even in complex radiation environments with various shapes of obstacles, and this ability could be achieved because of the good combination of the discrete Rao's algorithm and the ABC algorithm. The above experiments show that the proposed algorithm is very efficient in solving the problem of planning a facility inspection path.

**Author Contributions** All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Kwon Ryong Hong, Su Il O, Ryon Hui Kim, Tae Song Kim and Jang Su Kim. The first draft of the manuscript was written by Kwon Ryong Hong and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

## References

1. N. Chao, Y. Liu, H. Xia et al., A sampling-based method with virtual reality technology to provide minimum dose path navigation for occupational workers in nuclear facilities. *Prog. Nucl. Energy*. **100**, 22–32 (2017). <https://doi.org/10.1016/j.pnucene.2017.05.024>
2. C.F. Chen, J. Cai, Z. Wang et al., An improved A\* algorithm for searching the minimum dose path in nuclear facilities. *Prog. Nucl. Energy*. **126**, 103394 (2020). <https://doi.org/10.1016/j.pnucene.2020.103394>
3. M.E. Miyombo, Y. Liu, A. Ayodeji, Minimum dose path planning based on three-degree vertex algorithm and FLUKA modeling: radiation source discrimination and shielding considerations. *Ann. Nucl. Energy*. **168**, 108916 (2022). <https://doi.org/10.1016/j.anucene.2021.108916>
4. N. Chao, Y. Liu, H. Xia et al., Adaptive point kernel dose assessment method for cutting simulation on irregular geometries in nuclear facility decommissioning. *Radiat. Phys. Chem.* **150**, 125–136 (2018). <https://doi.org/10.1016/j.radphyschem.2018.04.035>
5. L.Q. Yang, Y. Liu, M. Peng et al., A gamma-ray dose rate assessment method for arbitrary shape geometries based on voxelization algorithm. *Radiat. Phys. Chem.* **158**, 122–130 (2019). <https://doi.org/10.1016/j.radphyschem.2019.02.015>
6. L.Q. Yang, Y. Liu, M. Peng et al., A fast gamma-ray dose rate assessment method for complex geometries based on stylized model reconstruction. *Nucl. Eng. Technol.* **51**(5), 1436–1443 (2019). <https://doi.org/10.1016/j.net.2019.03.004>
7. M. Zhang, S. Zou, Inversion and construction of nuclear radiation field. *Metall. Min. Ind.* **8**, 78–85 (2015)
8. Z. Wang, J. Cai, Inversion of radiation field on nuclear facilities: a method based on net function interpolation. *Radiat. Phys. Chem.* **153**, 27–34 (2018). <https://doi.org/10.1016/j.radphyschem.2018.09.003>
9. Z. Wang, J. Cai, Reconstruction of the neutron radiation field on nuclear facilities near the shield using Bayesian inference. *Prog. Nucl. Energy*. **118**, 103070 (2020). <https://doi.org/10.1016/j.pnucene.2019.103070>
10. Y. Liu, M. Li, C. Xie et al., Minimum dose method for walking-path planning of nuclear facilities. *Ann. Nucl. Energy* **83**, 161–171 (2015). <https://doi.org/10.1016/j.anucene.2015.04.019>
11. Z. Wang, J. Cai, Probabilistic roadmap method for path-planning in radioactive environment of nuclear facilities. *Prog. Nucl. Energy*. **109**, 113–120 (2018). <https://doi.org/10.1016/j.pnucene.2018.08.006>
12. R.M.C. Santiago, A.L.D. Ocampo, A.T. Ubando et al., Path planning for mobile robots using genetic algorithm and probabilistic roadmap. in *9th IEEE International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management*, Manila, Philippines, (2017). <https://doi.org/10.1109/HNICEM.2017.8269498>
13. N. Chao, Y. Liu, H. Xia et al., Grid-based RRT\* for minimum dose walking path-planning in complex radioactive environments. *Ann. Nucl. Energy* **115**, 73–82 (2018). <https://doi.org/10.1016/j.anucene.2018.01.007>
14. X. Xie, Z. Tang, J. Cai, The multi-objective inspection path-planning in radioactive environment based on an improved ant colony optimization algorithm. *Prog. Nucl. Energy*. **144**, 104076 (2021). <https://doi.org/10.1016/j.pnucene.2021.104076>

15. Y.C. Lai, S. Smith, Metaheuristic minimum dose path planning for nuclear power plant decommissioning. *Ann. Nucl. Energy* **166**, 108800 (2022). <https://doi.org/10.1016/j.anucene.2021.108800>
16. K.R. Hong, J.Y. Pak et al., Rao-combined artificial bee colony algorithm for minimum dose path planning in complex radioactive environments. *Nucl. Eng. Des.* **400**, 112043 (2022). <https://doi.org/10.1016/j.nucengdes.2022.112043>
17. Y. Liu, M. Li, C. Xie et al., Path-planning research in radioactive environment based on particle swarm algorithm. *Prog. Nucl. Energ.* **74**, 184–192 (2014). <https://doi.org/10.1016/j.pnucene.2014.03.002>
18. Z. Wang, J. Cai, The path-planning in radioactive environment of nuclear facilities using an improved particle swarm optimization algorithm. *Nucl. Eng. Des.* **326**, 79–86 (2018). <https://doi.org/10.1016/j.nucengdes.2017.11.006>
19. Y. Zhong, J. Lin, L. Wang et al., Hybrid discrete artificial bee colony algorithm with threshold acceptance criterion for traveling salesman problem. *Inform. Sci.* **421**, 70–84 (2017). <https://doi.org/10.1016/j.ins.2017.08.067>
20. A. Chandra, C. Natalia, Performance analysis of optimization methods for solving traveling salesman problem. *Innov. Technol. Sci. Solut. Ind.* **1**(15), 69–75 (2021). <https://doi.org/10.30837/itssi.2021.15.069>
21. A. Chandra, A. Naro, A comparative study of metaheuristics methods for solving traveling salesman problem. *Int. J. Inf. Sci. Technol.* **6**(2), 1–7 (2022)
22. D. Karaboga, B. Gorkemli, Solving traveling salesman problem by using combinatorial artificial bee colony algorithms. *Int. J. Artif. Intell. T.* **28**(01), 1950004 (2019). <https://doi.org/10.1142/s0218213019500040>
23. I. Khan, M.K. Maiti, A swap sequence based Artificial Bee Colony algorithm for traveling salesman problem. *Swarm Evol. Comput.* **44**, 428–438 (2019). <https://doi.org/10.1016/j.swevo.2018.05.006>
24. R.V. Rao, Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int. J. Ind. Eng. Comp.* **7**, 19–34 (2016). <https://doi.org/10.5267/j.ijiec.2015.8.004>
25. R.V. Rao, Rao algorithms: three metaphor-less simple algorithms for solving optimization problems. *Int. J. Ind. Eng. Comp.* **11**, 107–130 (2020). <https://doi.org/10.5267/j.ijiec.2019.6.002>
26. H. Liu, L. Gao, X. Kong et al., An improved artificial bee colony algorithm. in *25th Chinese Control and Decision Conference* (2013)
27. G. Sierksma, Hamiltonicity and the 3-Opt procedure for the traveling salesman problem. *Appl. Math.* **22**(3), 351–358 (1994). <https://doi.org/10.4064/am-22-3-351-358>
28. K.P. Wang, L. Huang, C.G. Zhou et al., Particle swarm optimization for traveling salesman problem. in *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, (2003)
29. D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J. Global Optim.* **39**(3), 459–471 (2007). <https://doi.org/10.1007/s10898-007-9149-x>
30. R.K. Roy, *A primer on the Taguchi method*, 2nd edn. (U.S, SME, 2010), p.329

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.