Progress in Nuclear Energy 146 (2022) 104170

Contents lists available at ScienceDirect

Progress in Nuclear Energy

journal homepage: www.elsevier.com/locate/pnucene

Improved TDV algorithm for three-dimensional space path planning in a complex radioactive environment with obstacles

Miyombo Ernest Miyombo [a,b], Yong-kuo Liu [a,*], Abiodun Ayodeji [c]

[a] Fundamental Science on Nuclear Safety and Simulation Technology Laboratory, Harbin Engineering University, Harbin, 150001, China
[b] Radiation Protection Authority, Exploration House, P. O. Box, 50002, Lusaka, Zambia
[c] State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou, 310027, China

ARTICLE INFO

Keywords:
TDV
3D space
Path planning
Radiation protection
FLUKA
Algorithm

ABSTRACT

Decommissioning of nuclear power plants is a complex process that exposes workers to radiation risks. Recently, many shortest path algorithms have been proposed to find a minimum dose path and the applications are generally based on three concepts: environmental modeling, radiation dose computation, and shortest path algorithm. However, the conventional minimum dose path search methods are sub-optimal because they suffer from semi-empirical approximation, complex geometries restrictions, and lack integrated environmental modeling capability. This study presents an improved three-degree of vertex (iTDV) algorithm, an extension of the novel three-degree of vertex (TDV) shortest path algorithm previously developed. In this work, we enhance its search mechanism to find the shortest path with the least amount of radiation dose in a complex three-dimensional (3D) space. The work also outlines the effective coupling of TDV with FLUKA Monte Carlo environmental modeling, which prevents interpolation uncertainty associated with the conventional path planning methods when dealing with large geometries. To demonstrate its efficiency, the iTDV is used to find the minimum dose walking path in environments with and without obstacles. The performance of the iTDV algorithm is also compared with the original TDV, and the Dijkstra's algorithm. The evaluation results and the improvements of the proposed method are discussed in this paper.

1. Introduction

Nuclear power plant (NPP) decommissioning involves nuclear fuel material transfer, removal of contaminated reactor structures and components and contaminated waste disposal. It is therefore important to develop safety measures for the protection of workers from radiation risks. When safety measures are not adhered to, nuclear facilities, such as NPPs, could be hazardous working environments. Minimum dose path planning has proven to be a reliable technique for controlling radiation exposure during decommissioning. The ability to control the time of exposure and distance of travel during decommissioning of a nuclear facility makes minimum dose path planning an important area of research. Traditionally, path planning methods in nuclear facilities consist of three important concepts: environmental modeling, radiation dose computation, and shortest path algorithm. Each concept has a unique effect on the minimum dose path planning method applied.

Presently, there are several minimum dose navigation methods (Alzalloum, 2009; Pei et al., 2020; Wang and Cai, 2018; Miyombo et al., 2021) used to find optimal paths in radioactive environments. Each method depends on several factors such as accuracy, reliability, and speed of shortest path algorithm, type, and complexity of the nuclear environment and types and forms of radiation sources, etc (Adibeli et al., 2021). Path planning in a nuclear environment is an optimization problem that utilizes an effective path search algorithm. Several methods are employed in various path planning research, and each approach has benefits and limitations, as well as a distinct scope and field of use (McConnell, 2001; Guanglin et al., 2011)-(McConnell, 2001; Guanglin et al., 2011).

Heuristic or probabilistic-based algorithms found in the literature are generally used to obtain the optimal path. A common algorithm for the shortest path problem is Dijkstra's algorithm, proposed in 1959 by the Dutch computer scientist Edger Dijkstra (Dijkstra, 1007). For a given directed graph *G(V, E), V* is the set of vertices (nodes) and *E* is the set of edges connecting these vertices. The algorithm searches for the shortest

Abbreviations: 3D, Three-dimensional; TDV, Three-degree of vertex; NPP, Nuclear power plant; PK, Point-Kernel.

* Corresponding author.
E-mail addresses: emiyombo@gmail.com (M.E. Miyombo), lyk08@126.com (Y.-k. Liu), ayod_abe@yahoo.com (A. Ayodeji).

https://doi.org/10.1016/j.pnucene.2022.104170
Received 8 October 2021; Received in revised form 11 January 2022; Accepted 20 February 2022
Available online 1 March 2022

path *G* from a given node *i* to a single destination node or all other nodes within a graph with non-negative weights (Dijkstra, 2012). The basic idea begins with the initial node, with the distance to the source node set to zero and all other distances set to infinity. All adjacent nodes are relaxed from the current node, and the closest node (shortest distance) is chosen to be the next current node. The relaxing of nodes adjacent to each current node is repeated until the destination is reached.

Another basic heuristic algorithm is A* (A star), a generic extension of Dijkstra's algorithm, used for finding the shortest path between two nodes in a graph. Algorithms such as A* include additional information, or heuristics, derived from the input data. In graph theory, a heuristic is the cost estimate of the path from the current node to the goal for the part of the graph tree that hasn't been explored yet (Fensel and Motta, 2001). As a result of this property, the algorithm can distinguish between more and less promising nodes, so that it can find the solution more efficiently. The algorithm is among the best known in the field of games and robotics; however, large memory requirement is its main drawback (Klancar et al., 2017). To address the memory issue, many improvements of the A* algorithm have been presented (Cai et al., 2015). One more algorithm that computes shortest paths from a single source vertex to all of the other vertices in a weighted digraph is the Bellman-Ford algorithm developed by Richard Bellman and Lester Ford Jr (Bellman, 2013). The algorithm is comparable to Dijkstra's algorithm in its basic structure but slower for the same problem and is more adaptable and computable with negative weights.

D* Lite, a fast-replanning method for robot navigation in unknown terrain that implements the same navigation strategies as focus dynamic A* (D*)(Likhachev et al., 2005) was proposed by (Koenig and Likhachev, 2002). The algorithm searches from the goal vertex towards the current vertex using heuristics to focus the search. As a result of applying lifelong planning A* to robot navigation in complex terrains, such as goal-directed navigation in unknown terrain, and mapping of unknown terrain, D* Lite was developed. The A* (D*) is a clever heuristic search method that maintains speedup of values, and has been extensively used in real robots such as Mars Rover prototypes and tactical mobile robot prototypes for urban reconnaissance (Matthieset al., 2002). Nevertheless, D* Lite has the same navigation strategy as D* but is algorithmically different and shorter.

Rapidly-exploring Random Tree (RRT), a randomized data structure concept and algorithm was designed for a broad class of path planning problems and includes some of the same desirable properties as a probabilistic roadmap (Véras et al., 2019). Although the experiments with RRTs have been successful, many challenging issues remain which require efficient nearest-neighbor techniques a topic of active interest in computational geometry. Compared to probabilistic road-maps, RRTs have the advantage that no connections between states are necessary and as a result, they are uniquely suited for use in nonholonomic and kino-dynamic planning (Chosetet al., 2005). A simple and efficient randomized algorithm was further proposed by (Kuffner and LaValle, 2000) for solving single-query path planning problems in high-dimensional configuration spaces. The method works by incrementally building two Rapidly-exploring Random Bees (RRTs) rooted at the start and the goal configurations. Rapidly Exploring Random Tree Star (RRT*) proposed by Karaman and Frazzoli (2011) is a sampling-based method for solving optimal path problems. The algorithm aggregates a set of tree-like collision-free nodes to develop a feasible walk path. RRT* builds a tree structure by connecting a set of points sampled from the obstacle-free space. Due to their suitability for high-dimensional complex problems, sampling-based algorithms such as RRT and RRT* have gained much success (Noreen et al., 2019). However, since the optimal path between two points in radioactive environments is difficult to predict, RRT* and A* methods have limitations in solving the minimum dose path problem in dynamic environments.

In continuous space, sampling-based algorithms that are less computationally complex (Chao et al., 2018a) are among the most popular (Islam et al., 2012). Rapidly exploring random tree star (RRT*) (Karaman et al., 2011) is one of the sampling-based algorithms demonstrated to be asymptotically optimal. The technique has been successfully applied to a wide range of robotic problems, as explained in (Karaman and Frazzoli, 2011), (Islam et al., 2012). There have been some approaches to improve the slow convergence rate of the RRT*, such as RRT*-Smart (Islam et al., 2012), Sampling-based A* (Persson and Sharf, 2014), and Transition-Based RRT* (Devaurs et al., 2016). While the RRT* plan is efficient for designing an optimal path in a narrow or complex obstacle space, it is inefficient when planning the optimal path in complex obstacle space and narrow areas because much time is spent checking whether local paths and samples are obstacle-free (Chao et al., 2018a).

In general, the sampling-based algorithms described above are limited by computation space requirements. As the number of grid points increases exponentially with the dimension of the state space, so does the worst-case (running) time of the algorithms (Karaman et al., 2011). Moreover, the efficiency of such algorithms is determined by the execution time, memory overhead, and the environment in which the system runs, whether static, dynamic, or real-time. In addition, most of the conventional algorithms have been found to perform better in static environments (Chao et al., 2019).

Path planing in static radioactive environments is the most common shortest path problem found in nuclear facilities because calculation of radiation from static radioactive sources is more simpler than from mobile sources. To solve the shortest path problem in a dynamic environment, Miyombo et al.(Miyombo et al., 2022) proposed an efficient state-aware adaptive pathfinder for nuclear emergency. Moreover, minimum dose path planning in a narrow environment or obstacle free-spaces is easy to perform. However, a better approach is necessary when the environment becomes complex, with multiple sources and obstructing obstacles. The existing minimum dose path-planning techniques based on the shortest path algorithms discussed above use the point-kernel dose calculations and Monte Carlo techniques. Analytical or Point-Kernel (PK) radiation dose calculations are less costly. However, they are generally weak when used in complex geometries, and they rely on a semi-empirical approximation of scattered radiation. The lack of integrated environmental modeling interface also makes these methods inefficient. Moreover, the radiation field computed from either the PK or Monte Carlo method is not continuous, and the graphical representation and visualization of dose rate in other places are usually computed by interpolation.

Firstly, this study presents an improved Three-Degree of Vertex (iTDV) shortest path algorithm, an extension of the novel three-degree of vertex (TDV) shortest path algorithm previously developed by Miyombo et al., (Miyombo et al., 2022). Secondly, this work studies a three-dimensional geometry path search, and the integration with FLUKA Monte Carlo environmental modeling for optimal protection of workers. The enhancement to the TDV in this work is the modification of the search mechanism to reduce computation memory and the subsequent reduction in computation time. Thirdly, the work outlines the adaptability of the iTDV, and the integration with the FLUKA Monte Carlo code to search for a minimum dose walking path in a three-dimensional space. Moreover, the radioactive environment simulation of the work minimizes interpolation uncertainty associated with traditional path planning methods when dealing with large geometries, and the theoretical background and implementation framework of the iTDV is discussed. The study also demonstrates FLUKA modeling of obstacle-cluttered environment, and how the proposed iTDV algorithm considers obstacles and their effect on minimum dose path search. The novelty of this work is summarized below:

1. The work presents an improved Three-Degree of Vertex (iTDV) algorithm for optimal path search in a complex radioactive environment.
2. The work enhances the conventional TDV by modifying the search mechanism to reduce computation memory and time.

3. This study also presents the framework to couple the iTDV with FLUKA for complex environmental simulations to address the interpolation uncertainty of radiation dose caused by the discrete nature of radiation from either the analytical or Monte Carlo methods.
4. To evaluate and validate the proposed iTDV, two different obstacle scenarios in complex geometries are examined for: (a) an obstacle cluttered radioactive environment and (b) a 3D radioactive space. Further, the performance of the iTDV is compared with the original TDV and the conventional Dijkstra' algorithm.

2. Background

2.1. Radiation dose calculations

Minimum dose path-planning in a nuclear environment deals with radiation field calculations. Thus, one of the key elements in path planning in radioactive environments is calculating cumulative dose. Currently, scientific modeling methods and related computation software are used for radiation dose assessment and shielding optimization, which require many mathematical calculations. Two common categories of software used for radiation dose modeling are the analytical-based methods (Prokhorets et al., 2007), (Antoni and Bourgois, 2017) and Monte Carlo methods (He et al., 2016). Visualization of dose rate distribution also plays a significant role when performing path planning in nuclear environments. High-dose risk areas are visually represented to help search for the minimum dose path required for optimized radiation protection. Through graphical representation, minimum dose paths can be identified and verified.

The PK method is based on the idea that radiation sources can be viewed as discrete elementary cells (points), and that the dose at the detecting point can be seen as the sum of all the contributions from these discrete elementary cells (Prokhorets et al., 2007). The total dose rate from a point kernel isotropic with energy, E, to the detecting point is given by:

$$D(r,E) = C(E)S(E)B(E,t(E))\frac{e^{-t(E)}}{4\pi r^2} \tag{1}$$

where r is the distance from the point kernel to the detecting point, C is gamma flux density to dose rate conversion factor, S is the strength of the point kernel, B is the buildup factor usually obtained from tables such as (Trubey, 1988). Other computations use the Berger Equation to evaluate B as:

$$B = 1 - a\mu re^{-b\mu r} \tag{2}$$

where a and b are coefficients, r is the distance, and μ is the total attenuation coefficient of the intervening material. The build-up factor B in the point-kernel method plays a significant role as a multiplier or factor that considers the attenuated radiation dose.

The mean free path $t(E)$ is the path length between the point-kernel and the detecting point given by:

$$t(E) = \sum_{i=1}^{n} \mu_i(E)d_i \tag{3}$$

where i is the index of the space region, n is the number of regions, μ_i is the linear attenuation factor for i-th region and d_i is the section of the line between the detecting point and the point kernel in the i-th region.

The point kernel represents a response at a point r from a source of unit strength. It is used extensively in developing relationships between radiation flux and exposure for various source geometries and absorbing materials. According to equation (1), this method usually deals with the sources of line spectra photons and primary walls based on a deterministic approach. The significance of results obtained from the point-kernel method is estimations of build-up factors that consider scattered radiation (Antoni and Bourgois, 2017).

Generally, analytical-based software uses the PK (Prokhorets et al., 2007; Miyombo et al., 2021), (Suteau et al., 2003) method for dose calculations. Computation with the PK method is less costly but disadvantaged with the semi-empirical approximation of scattered radiation. Even though build-up factors are used to consider the attenuated dose, evaluation of these factors depends on the energy, thickness, and type of material, thereby making the calculations complex. Moreover, the traditional PK programs have many restrictions, such as complicated modeling, complicated source setting, 3D fine mesh results statistics, and large-scale computing efficiency.

In PK codes, numerical calculations are employed to estimate radiometric and dosimetric quantities with approximations that tend to affect more or less the true value of the desired quantity. Additionally, the methods may result in significant bias in results, especially if they involve complex radioactive environments. For complex particle transport calculations, Monte Carlo is preferred. Moreover, computation involving PK is designed to solve the average particle equation while the Monte Carlo method, on the other hand, does not use an explicit equation but rather simulates particles and records aspects of average behavior (Antoni and Bourgois, 2017). The ability to compute complex geometries is the advantage of Monte Carlo methods and errors in the calculations also reduce when there is an increase in the number of particles under investigation. Thus, to compensate for bias results in complex geometries, the Monte Carlo method is an alternative method (Briesmeisterothers, 2000). The technique depends on repeated random sampling to obtained numerical results and provides a smooth definition of physical parameters during particle transport for greater precision in the final results (Antoni and Bourgois, 2017). The method is widely used in radiation transport calculations, shielding calculations, and particle tracking calculations based on simulations without special assumptions (Turner, 2008).

The software-based on Monte Carlo methods such as FLUKA, Geant4, MCNP, etc., requires a substantial amount of execution time, which is a disadvantage (Chao et al., 2018b). When dealing with multiple sources, thick barriers, and complex geometry, etc, computation time increases further. However, the Monte Carlo method offers reliable radiation transport effects computation and yields accurate results, including those performed in complex models. Even though the application of the Monte Carlo method to generate massive 3D fine mesh dose rate results of the whole shielded hall for visualization is still constrained for large-scale nuclear plants (Hendricksetal., 2005)in scientific research, Monte Carlo methods are best suited for transport calculations since their capacity to accurately model particle physics is excellent.

Additionally, it has been observed that general guidance for assessing effective dose requires prediction of organ dose, the relationships regarding the change in intensity with distance and intervening materials, and more especially absorption characteristics of human tissue for the radiation (Kim et al., 2011; Knoll, 2010). Understanding the time pattern of the dose rate (time exposed in various geometries) enables the calculation of the cumulative dose being deposited to important tissues in the body. The majority of work associated with nuclear facilities such as operation and decommissioning activities exposes workers to radiation risks and thus, exposure levels must be assessed to protect workers from undue radiation hazards.

In general, deterministic algorithms described above rely on a discretization of the configuration space to represent all possible navigation. This usually makes the algorithms computationally too expensive and dedicate far too much effort (Stentz, 1997), in time and memory because they depend on fewer nodes to accomplish their tasks, and thus are more effective in narrow areas and static environments. For these reasons, sampling-based motion-planning methods discussed above have become very popular. They alternatively provide probabilistically constructed motion in a graph to serve the purposes of the path planning problem and can thus be controlled to direct resources more effectively. However, the sampling-based algorithms have a longer execution time

when planning the optimal path in complex obstacle space. They also possess similar offline finite discretization characteristics with other traditional search algorithms and their solution suffers from grid point explosion and high computational cost (Chao et al., 2019). On the other hand, PSO-based algorithms are disadvantaged by yielding random results.

3. Method

To effectively search for minimum dose path in nuclear facilities, techniques usually involve radioactive environment modeling, radiation dose calculation and shortest path algorithm. Thus, the proposed method consists of the following elements:

(a) An improved three-degree of vertex (iTDV) shortest path algorithm,
(b) Environmental modeling with FLUKA, and
(c) Radiation dose calculation using FLUKA.

Moreover, to address the uncertainty caused by the interpolation of radiation values, this work uses FLUKA simulated radiation transport code results to generate several data points which cover the wide area of the search environment.

3.1. Environmental modeling with FLUKA and radiation calculations

FLUKA is an integrated particle physics Monte Carlo transport code for particle transport calculations and interaction with matter (Ferrari et al., 2021). It has extensive applications in high energy experimental physics and engineering, shielding, detector, and telescope design, cosmic ray studies, dosimetry, medical physics, and radiobiology (Battistoniet al., 2007; Battistoniet al., 2015; Battistoniet al., 2011; Abdul Haneefaet al., 2014; Taleei and Shahriari, 2009). Many different particles, including photons and electrons, can be simulated with high accuracy for various energies. With an enhanced version of Combinatorial Geometry (CG), FLUKA can handle even the most complex geometries. The FLUKA geometry gives very effective results due to a combination of efficiency, accuracy, consistency, and flexibility. Other tools available include visualization, debugging, and tracking charged particles even when magnetic or electric fields are present. With the introduction of new bodies, rounding accuracy has increased, speed has increased, and input preparation has become more accessible. A text file is used to prepare the input, and other tools are used for plotting results. By using command lines, jobs are submitted to FLUKA for simulations. FLAIR (FLUKA Advanced Interface) is an advanced graphical user interface for particle simulation programs. It is an interface for FLUKA which enables editing of input files, executing code, and viewing output files. In this study, FLUKA is used for nuclear environment modeling because of the enhanced version of Combinatorial Geometry (CG) for complex geometry.

3.2. Shortest path problem based on the degree of vertex

A path with the least cost (weight) from a point A, to another point, B, is traditionally referred to as a shortest path problem. Depending on the application and the algorithm used, the term weight under a graph can be distance, time, radiation exposure levels, or any other measurable property. In this study, the weight is the radiation dose rate computed from FLUKA. The shortest path problem is solved using a two-dimensional square grid consisting of discrete radiation fields generated from FLUKA modeling and simulations. Grid-based pathfinding algorithms are commonly disadvantaged by two limitations. Firstly, grid-based algorithms have a limited ability to find the optimal path, because the path must pass through adjacent grid points, thereby making them computationally expensive. Secondly, grid-based path planning often requires significant memory.

In discrete mathematics, related objects such as nodes (vertices) are joined by lines or edges to form a simple graph $G = (V, E)$ which comprises of V, a set of nodes and E a set of edges (Westothers, 2001). When a number representing cost, length, or capacity is assigned to each edge, a weighted graph is generated. A weighted graph is the same as a simple graph except that a real number (weight) is associated with each edge in the graph. Mathematically, a weighted graph consisting of a graph $G = (V, E)$ and a weight function $w: E \rightarrow \mathbb{R}$. For path-planning studies, weighted graphs are key because radiation dose can be represented as weights of the graph.

To solve these problems, this study proposes a shortest path algorithm that can limit the number of nodes to traverse when searching for an optimal path. This also minimizes the execution time and memory space.

3.3. TDV algorithm input

The input weights are Monte Carlo simulated radiation field results from FLUKA. The results represent radiation field distribution data points of the search environment. The FLUKA output results consist of 4 data columns and several rows of radiation values. The first column and second columns are x and y coordinates respectively. The third column is the radiation field which can be the energy, radiation dose, etc. The last column is the error for each entry. The radiation field can be visualized using the plot functions in the FLUKA interface or other external tools. The only data needed for minimum dose path planning studied in this work is the third column of the radiation field. When extracted, each value of the column vector is a point whose coordinates correspond to the x and y values in the initial 4 column results. These data outputs represent the radiation field distribution of the environment modeled in FLUKA. These values are converted into a two-dimensional array for path planning to represent the designed geometry (environment). The resulting m x n grid is the cost graph in which the least-cost decision algorithm is implemented. The proposed algorithm utilizes the cost graph either by user input (manually) or automatic reading from FLUKA output files. This is another advantage of the proposed algorithm compared with the traditional algorithms such as Dijkstra which requires manual input of the cost graph. Automatic input or reading of FLUKA files without user input makes the proposed algorithm faster. The cost matrix A of size m x n, in Equation (4) consists of weights (radiation weights) representing an environmental model. Rows are denoted by m and columns are denoted by n. In matrix form, A can be represented as follows:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} = A[i,j] \in \mathbb{R}^{m \times n} \quad (4)$$

3.4. Description of the degree of vertex (deg(v))

The Three-Degree of Vertex (TDV) shortest path algorithm is a grid-based shortest path algorithm. The theoretical and fundamental working principle of TDV is in the authors' previous work (Miyombo et al., 2022). The optimal path problem is implemented in matrix notation. Each element in matrix A denotes a vertex (node) and the vertex has other adjacent vertices joined to it according to the position A [i,j] it occupies. The degree of vertex deg(v) is the number of vertex adjacent to a vertex used to control the least-cost path planning. Thus, for any given matrix, nodes are grouped into 6 categories according to the number of adjacent vertices joined to vertex v. Table 1 shows a description of degree of vertices for each category of nodes.

Table 1
Degree of vertex per category.

Category	Description	Matrix notation	deg (v)
1.	Nodes in the corners (4)	*A[i, j], A[i, n], A[m, j], A[m, n],*	3
2.	Nodes along the first row *i* from column *j+1* to column *n-1*	*A[i, j+1:n-1]*	5
3.	Nodes along the last row *m* from column *j+1* to column *n-1*	*A[m, j+1:n-1]*	5
4.	Nodes along the first column *j* from row *i+1* to column *m-1*	*A[i+1:m-1, j]*	5
5.	Nodes along the last column *n* from row *i+1* to column *m-1*	*A[i+1:m-1, n]*	5
6.	Rest of the nodes	*A[i+1:m-1, j+1:n-1]*	8

3.5. The sequence of the path

In A, the path from an initial vertex v_o to the final vertex v_f is obtained from a finite sequence of the form, $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \bullet\bullet\bullet \rightarrow v_f$. Since the navigation deals with a weighted graph, the sequence can be used to obtain the accumulated weights from the initial vertex to the final vertex. The aim is to compute the cost of moving from v_o to v_f of the weighted graph. This is done by summing the sequence of the vertices to obtain the total cost covered as shown in Equation (5).

$$\sum_{i=1}^{N} v_i = v_0 + v_1 + v_2 + \bullet\bullet\bullet + v_f \tag{5}$$

where N is the number of vertices. The proposed algorithm uses the generated matrix A in Equation (6) to search for the least-cost path. Equation (5) is a general rule of finding the total cost at the final vertex. However, because the objective is to solve the least-cost pathfinding problem, certain rules must be applied.

First, only A elements which yield the least cost from the initial vertex v_0 to the final vertex v_f are used in the path search. This is achieved by always redefining A according to v_0 (starting element) and v_f (ending element). Secondly, to define vertices for use in the path sequence, the degree of vertex deg(v) is determined and applied. For example, when the initial vertex in A is *A[i,j]* as shown in Equation (6) and the final vertex is *A[m, n]* the movement of *A[i,j]* entails the sequence of the path is from left to right. This definition automatically means that in the sequence there must be no movement from right to left at any point because this will be an added cost that is sub-optimal. Thus, regardless of the category (1, 2, …, 6) in which an element belongs, the degree of vertex deg(v) is reduced to 3. This is the fundamental framework of the proposed TDV shortest path algorithm. As an example, in A, where a_{ij} is the initial vertex, the adjacent elements added to it in the algorithm are *A[i, j], A [i+1, j], [i, j+1], A[i+1, j+1].*

$$A = \begin{bmatrix} a_{ij} & a_{i,j+1} & \cdots & a_{i,n} \\ a_{i+1,j} & a_{i+1,j+1} & \cdots & a_{i+1,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,j} & a_{m,j+1} & \cdots & a_{mn} \end{bmatrix} \tag{6}$$

To show the elimination of elements from the matrix when an initial vertex is not a_{ij} or *A[i,j],* element *A[i+1,j+1]* can be taken as initial vertex and *A[m, n].* The sequence of elements is also from left to right. However, because the final vertex is to the right, the first row *A[i, j:n]* and the first column *A[i:m, j]* is not significant to the least-cost pathfinding. Including any element from these regions will not yield the least cost path. Therefore, the proposed algorithm solves the shortest path problem by eliminating the first row *A[i, j:n]* and the first column *A[i:m, j]* from A to obtain a new matrix A' with a size equal to the initial vertex and final vertex indices. This component is one of the advantages of the algorithm which makes it less costly especially when dealing with wide areas.

3.6. Improved TDV (iTDV) algorithm

In the previous study (Miyombo et al., 2022) in which TDV is developed, the algorithm consists of two sections, Section I and Section II. Section I involves the computation of the least-cost from initial element to the final element in A and storing the cost into a matrix B. Section II involves the least-cost path search. In both sections, the degree of vertex is applied. Computing least-cost aims to obtain the total sum of the cost for vertices traversed from v_o to v_f yielding the minimum cost of travel. On the other hand, Section II is aimed at evaluating the sequence path of vertices traversed during navigation.

Application and implementation of TDV in two parts has the following shortcomings: more computation memory taken, and higher computation time. TDV takes more computer memory because the execution consists of two stages. Firstly, the computation of minimum cost at the destination while passing through the searching nodes based on two matrices. Secondly, tracking back and storing the nodes passed using two matrices takes up additional space. Consequently, computation time increases. These limitations are disadvantageous to TDV when dealing with complex geometries. Thus, the objective of this work is to simultaneously search and store the nodes or vertices of the weighted graph passed. This way, the minimum dose path is obtained during minimum cost evaluation without tracking back the passed nodes. The use of 4 identical matrices is also avoided, and consequently, computer memory usage and execution time are minimized.

The improved TDV algorithm is executed as follows:

Given an initial vertex (starting point) and final vertex (destination) in matrix *A,* the aim is to obtain matrix *B* shown in Equation (9) whose size is equal to A. Each element *A[i, j]* is a node consisting of a weight or cost to traverse through that vertex. On the other hand, the weights in B, consists of the sums of adjacent elements (nodes) of A, in the direction of motion from the initial vertex (element) to the final vertex (element). This is the principle of dynamic programming in which the already calculated value is saved in extra memory for future reference. Matrix *A* has size *(m, n)* and *consists* of elements (costs) generated from the FLUKA simulation. For a given scenario to obtain the total cost of the path sequence from v_0 or *A[i, j]* to v_f or *A[m, n]* according to the degree of vertex, is the sum of cost from *A[i, j]* to the cost on that path (including both initial vertex and the final vertex *A[m, n]).*

In a weighted graph, the iTDV obtains the optimal path in both left-right and left-right motion. In both cases, regardless of the degree of vertex or deg(v) of an element described in Table 1, the degree of vertex, deg(v) used for the optimal path is 3 as described above. Thus, each element has 3 adjacent elements. For a left-right motion ($\overrightarrow{A}$), and expressing the navigation in terms of travel, only three possibilities of travel are considered: traverse down, right, and diagonally lower element from the current element, i.e., from the current element *A[i, j],* only *A [i+1, j], [i, j+1],* and *A[i+1, j+1]* can be navigated rightwards, as shown in Equation (7). Similarly, for the right-left motion ($\overleftarrow{A}$), only three possibilities of travel are also considered: traverse down, left and diagonally lower element from the current element, i.e., from the current element *A[i, n],* only *A [i+1, j], [i, j-1],* and *A[i+1, n-1]* can be navigated rightwards as shown in Equation (8).

First of all, possible paths are obtained by recursion because the path from a particular element is dependent on the paths from its adjacent element. To calculate the cost from an element, we calculate the cost of the paths from adjacent elements. This process demonstrates quite clearly that it is possible to visit a particular element multiple time. So, the path from an initial vertex to the final vertex from a vertex already calculated can be saved in extra memory to reduce the cost of computation and hence time.

$$\vec{A} = \begin{bmatrix} a_{ij} & a_{i,j+1} & \cdots & a_{i,n} \\ a_{i+1,j} & a_{i+1,j+1} & \cdots & a_{i+1,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,j} & a_{m,j+1} & \cdots & a_{mn} \end{bmatrix} \quad (7)$$

$$\overleftarrow{A} = \begin{bmatrix} a_{ij} & a_{i,j+1} & \cdots & a_{i,n} \\ a_{i+1,j} & a_{i+1,j+1} & \cdots & a_{i+1,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,j} & a_{m,j+1} & \cdots & a_{mn} \end{bmatrix} \quad (8)$$

$$B = \begin{bmatrix} b_{ij} & b_{i,j+1} & \cdots & b_{i,n} \\ b_{i+1,j} & b_{i+1,j+1} & \cdots & b_{i+1,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m,j} & b_{m,j+1} & \cdots & b_{mn} \end{bmatrix} \quad (9)$$

To obtain the travel cost from one node to the other, the weight of the current node is added to the adjacent weight throughout the navigation until the destination is reached. These sums make up matrix B of the same size as A. The criterion for adding two adjacent nodes using TDV is based on the fundamental principle of shortest path evaluation which requires that only the least cost or node with the smallest cost is added to the current node to obtain the optimal results. Consequently, the final vertex has the sum of the cost of navigation with the least possible value. In this work, the path (navigation nodes) is simultaneously evaluated as the sum of the least possible adjacent cost are being computed. The smallest sum in B is evaluated at every iteration, and the corresponding node in A becomes the path which the algorithm stores during the process. As a result, an array of nodes representing the optimal path is obtained without tracking back the process, the procedure followed in the previous study.

If the direction is from left to right, the current cost in B is computed from the addition of the corresponding current node in A to the minimum adjacent nodes of B [i,j] among the three nodes: up, left and diagonally upper elements. The iteration is done throughout the search until the destination is reached. At the same time, the minimum cost (sum) in B at every iteration is obtained from: the current sum, left adjacent, and the upper adjacent. The corresponding position in A of a minimum sum evaluated in B is the position in the weighted graph that denotes the optimal path. If the direction is from right to left, the current cost in B is computed from the addition of the corresponding current node in A to the minimum adjacent nodes of B[i,j] among the three nodes: up, right and diagonally upper elements. In this case, the minimum cost in B at every iteration is obtained from: the current sum, right adjacent, and the upper adjacent. Similarly, the corresponding position in A of a minimum sum evaluated in B is the position in the weighted graph that denotes the optimal path for the search.

Table 2 shows the original TDV, and Table 3 describes the iTDV. In the original TDV in Table 2 the 13th – 26th instructions require more computation space compared to the improved version. The summary flow chart of the improved TDV algorithm is given in Fig. 1.

3.7. Three-dimensional radioactive environment path planning

Traditionally, minimum dose path planning is conducted in 2-D space represented by the horizontal plane. In a radioactive environment, the path search is done between 10 and 20 cm above the ground. This horizontal plane is most suitable to denote the walking path in a radioactive environment. However, in some real situations and more complex radioactive environments, navigation consists of 3-dimensional (3D) space. Such environments can be nuclear facilities with stairs or inclined surfaces. The worker, thus, is expected to walk on both a horizontal and inclined plane. In this work, FLUKA simulation is used to model a 3D space and also compute dose distribution. The study utilizes the capability of FLUKA Monte Carlo simulation to obtain 2D weighted graphs of horizontal and inclined surfaces on which the personnel walks. The two 2D weighted graphs each contained costs (radiation dose) from the simulations. Application of the two categories of 2D weighted graphs to search for the optimal path is generally complex and tedious with the conventional method.

Table 2
TDV algorithm.

S/N	Description
1.	0 ← B (least -cost matrix initialization)
2.	initial cost← A [start_row, start_col]
3.	final cost ← A [end_row, end_col]
	Sum of ith row of cost matrix A for (i = start_row)
4.	**if** j = start_col**, then** B [i, j] ← A [i,j]
5.	**else** B [1, j] ← A [1, j] + B [1,j-1]
	Sum of cost along jth column of the cost matrix (j = start_col)
6.	if i = start_row, B [i, j] ← A [i, j],
7.	**e**lse B [i,j]←A [i, j] + B [i-1,j]
	Compute minimum cost in A [start_row+1:end_row, start_co+1:end_col]
8.	AboveAdj ← B [i-1, j]
9.	DiagonalAdj ← B [i-1, j-1]
10.	LeftAdj ←B [i, j-1]
11.	B [i, j] ← A [i,j] + min ([AboveAdj DiagAdj LeftAdj])
12.	*k*← (mxn)/2, number of iterations
13.	start ← B [end_row,end_col] , search
	TDV minimum cost path
14.	**if** *LeftAdj ←MinAdj*, **then** B [m, n-1], LR or
15.	**else if** *DiagAdj ←MinAdj*, **then** B [m-1, n-1]
16.	**else** *UpAdj←MinAdj,* **then** B [m-1, n]
17.	Path cost = occupied -MinAdj
18.	**if** B [i,1] ← B, **then**
19.	**m**inimum path ←Stored path cost append A [i,1]. **STOP!**
20.	**else** repeat steps 14-17
21.	**if** B [1,j] ← B, **then**
22.	**m**inimum path ←Stored path cost append A [1,j]. **STOP!**
23.	**else** repeat steps steps 14-17
24.	**if** B [1,1]← B, **then**
25.	**m**inimum path ←Stored path cost append A [1,1]. **STOP!**
26.	**else** repeat steps 14-17

Table 3
iTDV algorithm implementation and description.

S/N	Description	Execution
1	A, 0 ← B	*# Initialization* of A and B
2	initial cost← A [start_row, start_col]	*# initial node*
3	final cost ← A [end_row, end_col]	*# final node*
	Sum of ith row of cost matrix A for (i = start_row)	
4	**if** j = start_col**, then** B [i, j] ← A [i,j]	
5	**else** B [1, j] ← A [1, j] + B [1,j-1]	*# sum of* ith *row in A*
	Sum of cost along jth column of the cost matrix (j = start_col)	
6	**if** i = start_row, B [i, j] ← A [i, j],	
7	**else** B [i,j]←A [i, j] + B [i-1,j]	*# sum of* jth *column of A*
8	path = zeros (1, (m-1)*(n-1))	*# initialize number of elements in path to the maxima*
9	**while** u ≤ length (path)	*# set iteration equal to the number of elements in path*
10	AboveAdj ← B [i-1, j]	*# adjacent cost above occupied*
11	DiagonalAdj ← B [i-1, j-1]	*#diagonal adjacent cost of occupied*
12	LeftAdj ←B [i, j-1]	*#left adjacent cost of occupied*
13	B [i, j] ← A [i,j] + min ([AboveAdj DiagAdj LeftAdj])	*# minimum cost*
14	upper_cost = B [i-1,j]	*#upper sum*
15	left_cost = B [i,j-1]	*#left sum*
16	current_cost = B [i,j]	*#current sum*
17	min_cost = min ([upper_cost left_cost current_cost])	*# least possible cost path*
18	path.append (min_cost)	*#store the path*
	STOP!	

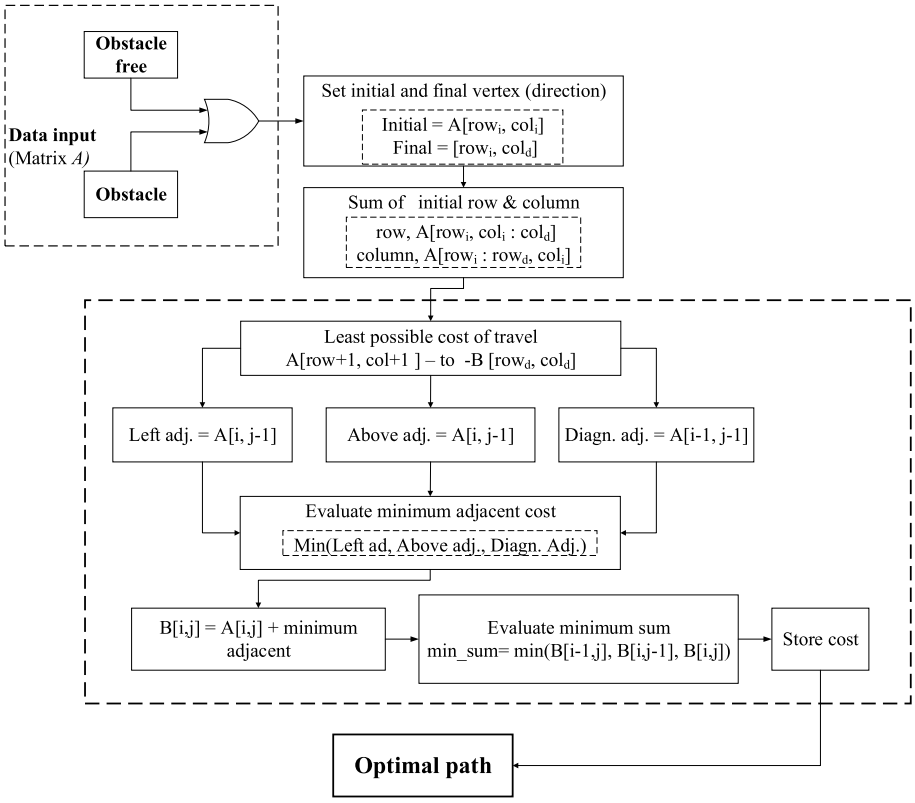

Fig. 1. Summary flow chart of the iTDV algorithm.

Thus, the iTDV algorithm built from the TDV is adaptable to huge data points and complex geometry applications with FLUKA. This integration enables the coupling of horizontal and inclined surface path planning in a 3D radioactive environment. The radioactive scene in Fig. 2 shows an example of a 3D radioactive scene in which nuclear personnel is exposed to radiation when walking on the horizontal floor as well as while walking up the inclinded surface. The objective of this scenario is to obtain an optimal path of the worker walking on the horizontal plane and the inclined plane. Path planning on the inclined surface with a grid-based algorithm such as the iTDV is achieved by firstly computing the dose distribution on the inclined surface. The radiation field is then used as the cost of a 2D weighted graph which then replaces the equivalent projection on the horizontal surface to obtain a combined 2D weighted graph for the implementation path planning. Figs. 2 and 3 show an illustration of the coupling of the two different surfaces into a single surface in which the corresponding 2D grid graph is used for the implementation of the iTDV path planning. The 3D coupling path planning flow chart is shown in Fig. 4.

4. Simulations and results

The iTDV is evaluated in two simulations: an obstacle cluttered radioactive environment and a 3D radioactive space. In the obstacle-filled space, the improved TDV is compared with the original version of TDV and a Dijkstra algorithm and its further application is analyzed in 3D radioactive environment path planning. Environmental modeling and radiation dose calculations are performed using FLUKA installed on Ubuntu (Linux distribution) operating system, whereas the iTDV, and other shortest path algorithms are all implemented using Python programming.

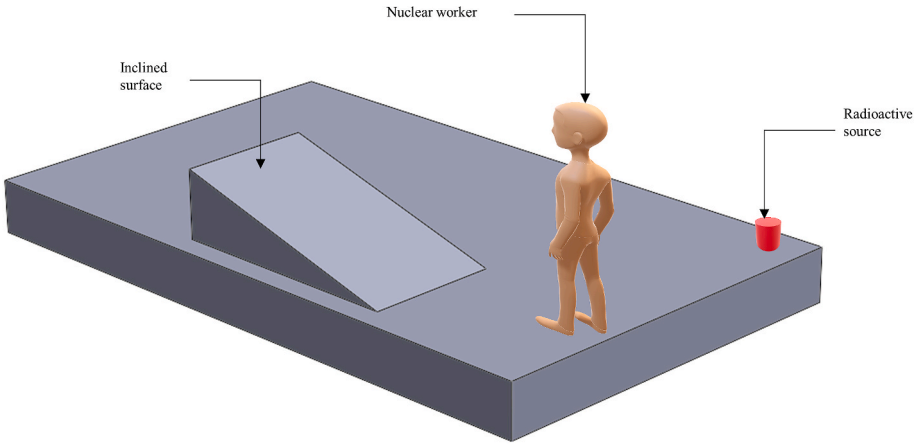

Fig. 2. Path planning in 3D space. Nuclear personnel walking up an inclined surface of a radioactive environment.

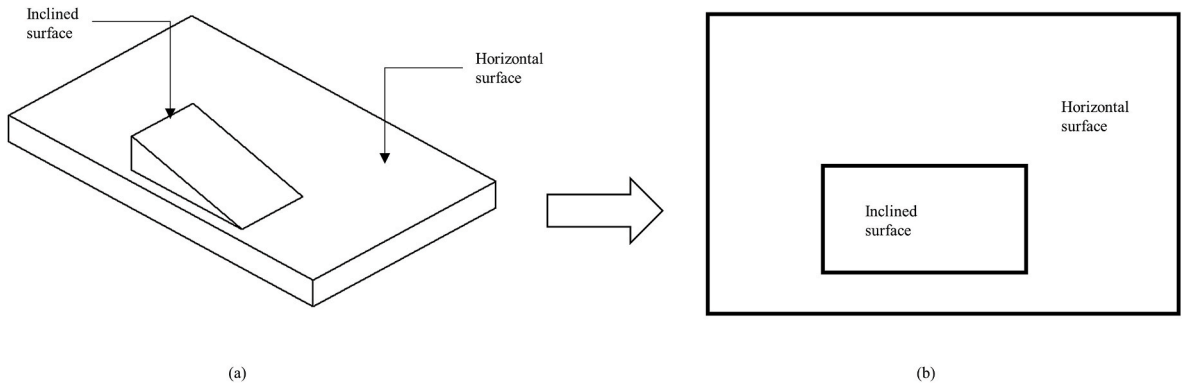

Fig. 3. Coupling of inclined surface and horizontal surface. (a) walking surfaces; (b) projection of the inclined surface onto the horizontal surface. The coupling effect yields one weighted graph consisting of the included surface weighted graph and horizontal weighted graph for path planning implementation.

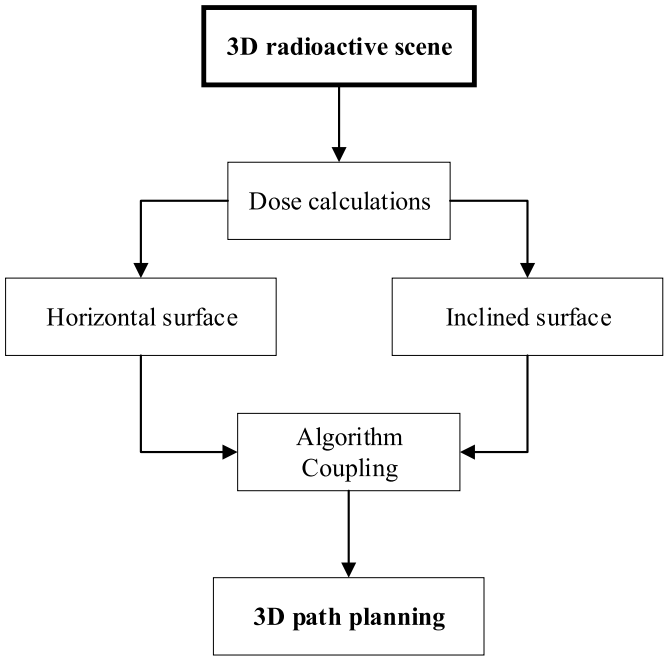

Fig. 4. Flow chart of the 3D path planning.

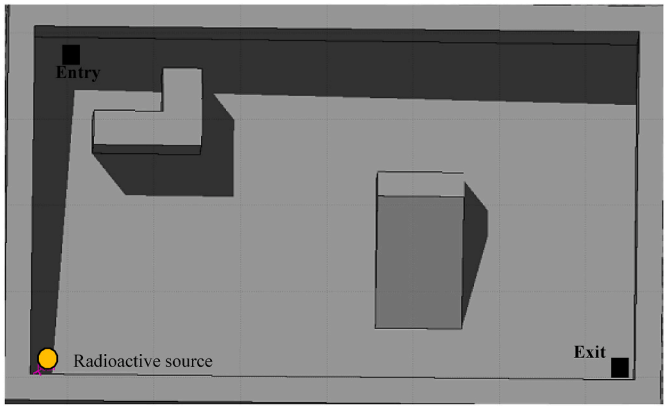

Fig. 5. Obstacle filled radioactive environment.

4.1. Simulations

4.1.1. Obstacle cluttered environment

The purpose of this simulation is to evaluate the performance of the improved TDV algorithm to search for an optimal path in an obstacle-cluttered radioactive environment. The simulations are conducted in a hypothetical rectangular radioactive room of dimensions 4m × 8m x 1.5m in which a cylindrical Cobalt-60 (^{60}Co) is temporarily stored and located at the bottom left corner of the room as shown in Fig. 5. The gamma spectrum of ^{60}Co undergoing spontaneous decay has two significant peaks. It emits two gamma rays with energies of 1.17 and 1.33 MeV. The activity of the source is designed at 2.8 GBq for an average of 2.5 MeV photon energy. Thus, the corresponding gamma intensity in this work is also designated at the intensity of 2.24 × 10E9 γ/s and used in the FLUKA dose calculations. The radioactive environment is filled with two obstacles; an L-shaped and a wedge.

The simulation aims to use the improved TDV algorithm to search for the shortest path that will yield the minimum dose from the entry point to the exit point. The algorithm is also tested on how the two obstacles are avoided. Path planning is determined as follows: First, the environmental modeling (dose calculations and geometry) is performed in FLUKA with 2 million particles for 5 cycles. With the designated radioisotope details, FLUKA commands are used for scoring and conversion of intensity to dose equivalent and further normalization from the gamma intensity.

The resulting radiation field output from FLUKA is converted into a 2D weighted graph (grid) in which the iTDV is implemented. Fig. 6 shows the radiation field. The optimal path is also determined using the original TDV algorithm and Dijkstra's algorithm to compare the results of the iTDV algorithms. Fig. 7 shows the path obtained from all three algorithms. Computation speed and accumulated dose obtained from the algorithms for fewer and more nodes in the environment are given in Table 4 and Table 5 respectively.

4.1.2. Three-dimensional space path planning

To simulate a three-dimensional path planning, a hypothetical temporal storage room for radioactive materials is developed. The room has an inclined walking surface as shown in Fig. 8 with a cylindrical Cobalt-60 (^{60}Co) emitting gamma radiation of 1.17 and 1.33 MeV. The source is located on one side of the room near the bottom of the inclined surface. Similar to the previous simulation, 2 million particles and 5 cycles are simulated to compute the radiation field distribution in the 3D scene. The task of the nuclear personnel requires walking from the horizontal surface and climbing the walking surface. For this scene, the radiation dose distribution is computed 10 cm above the horizontal floor and also 10 cm above and parallel to the inclined surface. Fig.9 shows the horizontal radiation dose distribution. For the inclined surface, the dose distribution in Fig.10 is obtained from the projection along the vertical height of the inclined surface. As described in Section 3.7, inclined surface dose and horizontal are coupled to generate a single 2D weighted graph shown in Fig.11 for path planning implementation. Fig. 11 and

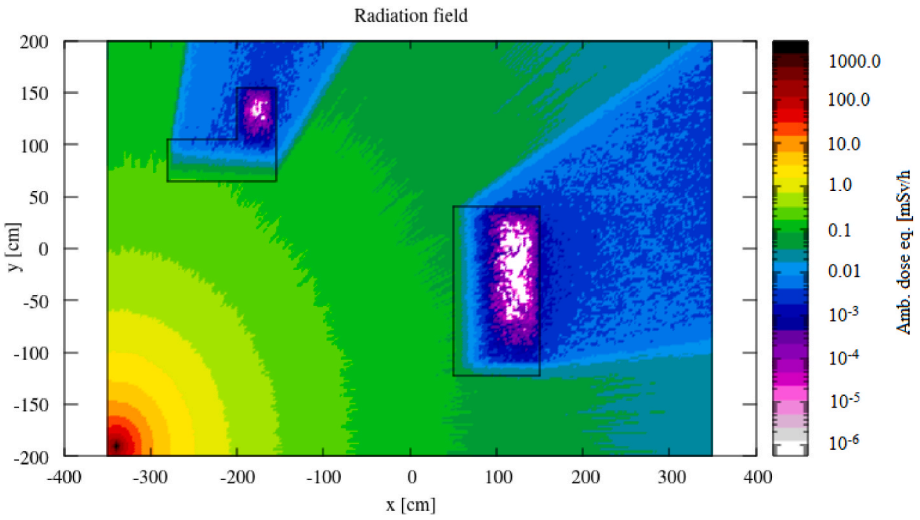

Fig. 6. Radiation field distribution in an obstacle cluttered radioactive environment.

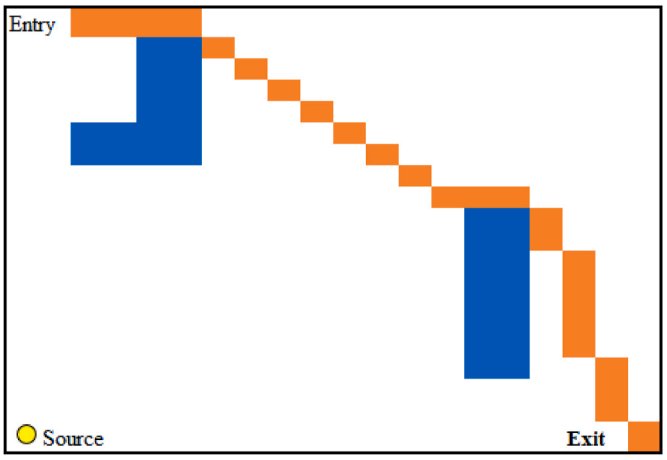

Fig. 7. Optimal path obtained from iTDV, TDV and Dijkstra.

Table 4
Comparison of the execution time of iTDV algorithm with TDV and Dijkstra's algorithm.

	Computation speed (s)				
Scenario	TDV	Dijkstra	iTDV	iTDV/TDV	iTDV/Dijkstra
Obstacle-filled scene	0.39	0.42	0.29	0.74	0.69

Table 5
Accumulated minimum dose at the exit.

	Accumulated dose, mSv		
Algorithm	400 nodes	160801	[a]Ratio
iTDV	0.38	0.22	1.73
TDV	0.38	0.22	1.73
Dijkstra's	0.38	N/A	N/A

[a] Ratio = 400 nodes (dose)/160801 nodes (dose).

Fig. 12 illustrate coupling of the optimal path of the person navigating from the starting point A on the horizontal floor to the top (point B) of the inclined surface. The corresponding minimum dose path is shown in both Fig.12 and Fig. 13. Table 6 shows the average statistical errors from the output files.

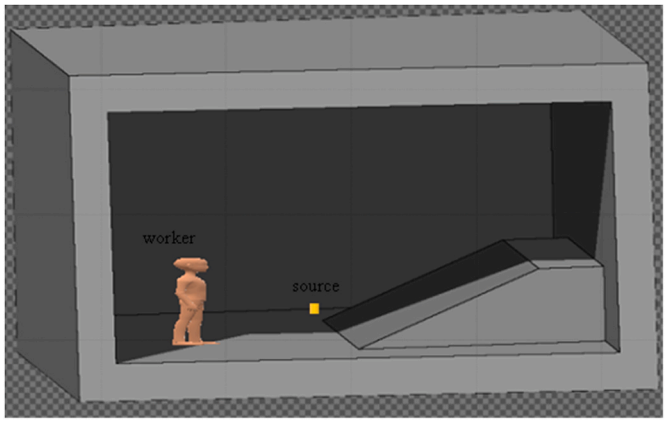

Fig. 8. 3D scene of a radioactive environment with an inclined walking surface.

5. Discussion

As opposed to the earlier vision of the TDV, a grid-based pathfinding algorithm that is based on forward and backward search mechanisms, the improved version of TDV is proposed to reduce the lengthy search process. Combining the two fundamental search processes of TDV into one makes the improved TDV more adaptable to complex geometry due to less computation memory and time. The extension of TDV in this work is to enhance the computation memory and computation speed of the original version of TDV as shown in Table 4. The computation speed of the improved TDV is faster than TDV by a factor of 0.74 and by a factor of 0.69 more than Dijkstra's algorithm. In addition, the improved TDV algorithm solves three different problems limiting other analytical methods: the semi-empirical approximation problem, restrictions in geometric complexity, and lack of integrated environmental modeling capability.

Generally, computation time of grid based shortest path algorithms is mainly dependent on three factor: computation memory, size of the navigation space and number of nodes. Computation memory is affected by the memory occupied by the algorithm. The higher the memory the higher the longer the time. Larger and complex environment increase computation time. Thus, in the complex geometry simulation considered in this work, more nodes are needed accurately search for the shortest path. In grid-based algorithms, the number and size of nodes determine both the computation time and accuracy. Consequently, both TDV and Dijkstra require more time than iTDV to search for the optimal path as

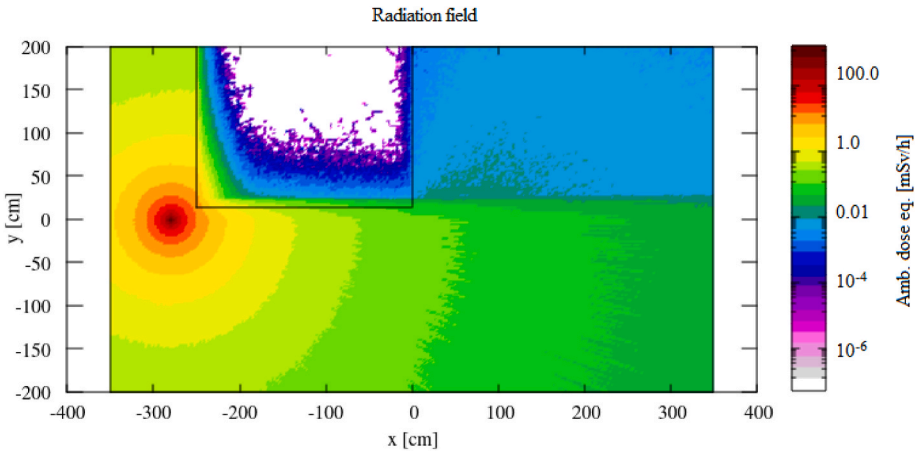

Fig. 9. Horizontal floor radiation field.

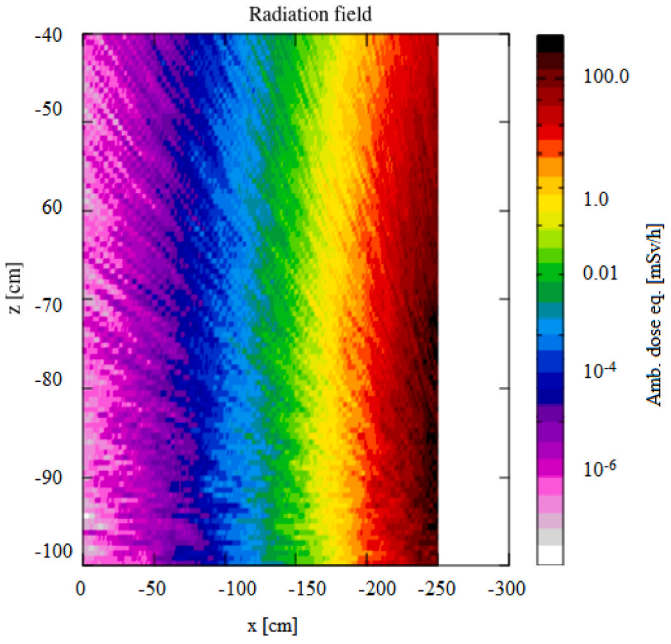

Fig. 10. Vertical projection radiation field.

shown in Table 4.

The TDV, iTDV, and Dijkstra algorithms obtained identical optimal paths (Fig. 7) and dose (Table 5) when using the same number of nodes, in this case, 400 nodes. These paths are also comparable to the path obtained from 160,000, nodes using the TDV and iTDV algorithms, but different accumulated doses. As shown in Table 5 the accumulated dose from TDV, iTDV and Dijkstra at destination in a 400 node space is 0.38 mSv. This dose is overestimated by a factor of 1.73 from 160,000 nodes

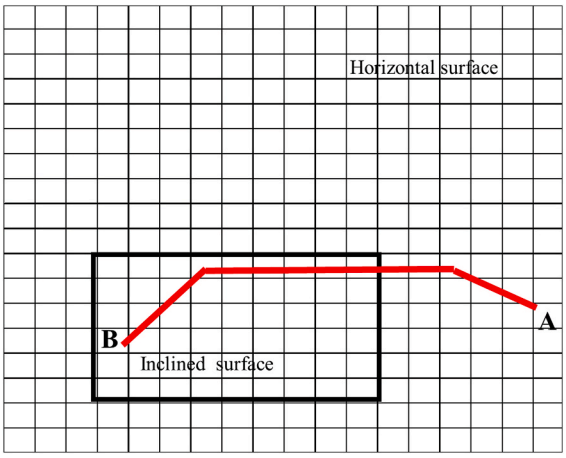

Fig. 12. Optimal path in on coupled 2D grid.

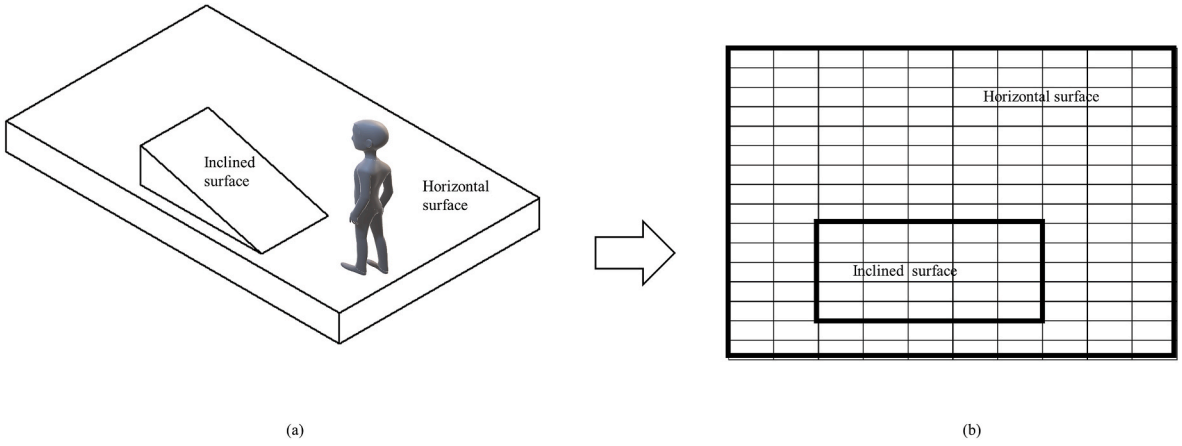

Fig. 11. (a) 3D scene (b) Coupling into a single 2D weighted graph for path planning.

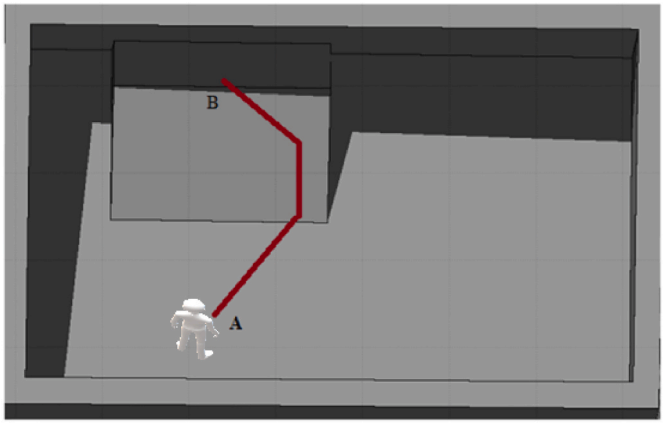

Fig. 13. 3D scene path planning.

Table 6
Statistical errors.

Simulation	Average statistical error per 5 cycles
Obstacle -cluttered environment	6.7%
3D path planning	5.5%

implemented with the TDV and iTDV algorithm which yielded 0.22mSv. Higher nodes are more accurate than a few nodes of the same environment because every space can be associated with a radiation field. On the other hand, the few nodes consist of interpolation uncertainties and summation approximations resulting from the segmentation of the searching area into fewer nodes.

In addition, FLUKA Monte Carlo is preferred in this work because it is capable of computing radiation dose and generating a huge number of radiation field distributions within the area of interest. For example, in the obstacle-cluttered environment, 160,000 nodes within the environment are generated corresponding to 400 by 400 grid. The iTDV algorithm uses less computer memory. The algorithm can therefore utilize the huge number of nodes for path planning to obtain the optimal path shown in Fig. 7. On the other hand, searching for an optimal path with 160, 000 nodes with Dijkstra's algorithm requires more computer space and is cumbersome. Therefore, to evaluate the performance of the developed TDV algorithm with the traditional Dijkstra' algorithm, the number of nodes is reduced to 400 nodes to enable comparison of the TDV with Dijkstra.

As demonstrated in Fig. 11, the integration of the complex geometry of FLUKA with the iTDV can solve the shortest path problem in a 3D radioactive environment. 3D path planning is made possible from the capability of FLUKA simulations to estimate dose along the vertical height of an inclined surface. Consequentrly, by projecting this radiation field over the inclined surface, and coupling the two planes (horizontal surface and inclined), 3D path planning is achieved to obtain a minimum dose path.

6. Conclusion

Several algorithms have been used to optimize paths in radioactive environments. However, the conventional path search algorithms are sub-optimal, because they use finite discretization techniques to compute optimal paths in simple and narrow geometry, which is not robust enough for obstacle-cluttered environments or large navigation area that requires significant computation. This work presents an improved Three-Degree of Vertex (iTDV) algorithm capable of searching for an optimal path in complex geometries and in 3D navigation space by coupling a horizontal and inclined walking planes.

Although FLUKA provides three-dimensional radioactive environment modeling, as well as radiation field computation along a vertical plane, projection of the vertical dose along an inclined surface has some uncertainty caused by variation of distance from the radioactive source. Therefore, to accurately compute the radiation field along the inclined surface, more segmentation is required, which is also computationally costly. Nevertheless, the methods presented in this study, and the demonstration therein, are useful for nuclear facility decision-makers to improve workers' safety during nuclear decommissioning.

Credit author statement

Miyombo Ernest Miyombo: Conceptualization, Methodology, Data curation, Software, Validation, Writing – original draft. Abiodun Ayodeji: Writing- Reviewing and Editing, Visualization, Data curation. Yong-kuo Liu: Supervision, Project administration

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the project of the National Natural Science Foundation of China (NO.61803285) and (NO. U21B2083), the China Ship Development and Design Center (Grant NO.XX519-2019-A001-08 (A40102)_03 and the International Atomic Energy Agency (IAEA) project INT2018 under the fellowship code EVT1901086-0007-ZMB.

References

Abdul Haneefa, K., et al., 2014. FLUKA Monte Carlo for basic dosimetric studies of dual energy medical linear accelerator. Journal of Radiotherapy 2014.
Adibeli, J.O., Liu, Y.-K., Ayodeji, A., Awodi, N.J., 2021. Path Planning in Nuclear Facility Decommissioning: Research Status, Challenges, and Opportunities. Nuclear Engineering and Technology. https://doi.org/10.1016/J.NET.2021.05.038.
Alzalloum, A.-Q., 2009. Application of Shortest Path Algorithms to Find Paths of Minimum Radiation Dose. University of Illinois at Urbana-Champaign.
Antoni, R., Bourgois, L., 2017. Applied Physics of External Radiation Exposure. Springer.
Battistoni, G., et al., 2007. The FLUKA Code: Description and Benchmarking. https://doi.org/10.1063/1.2720455.
Battistoni, G., et al., Dec. 2011. Applications of FLUKA Monte Carlo code for nuclear and accelerator physics. Nucl. Instrum. Methods Phys. Res. Sect. B Beam Interact. Mater. Atoms 269 (24), 2850–2856. https://doi.org/10.1016/J.NIMB.2011.04.028.
Battistoni, G., et al., Aug. 2015. Overview of the FLUKA code. Ann. Nucl. Energy 82, 10–18. https://doi.org/10.1016/J.ANUCENE.2014.11.007.
Bellman, R., 2013. Dynamic Programming. Courier Corporation.
Briesmeister, J.F., others, 2000. In: MCNPTM-A General Monte Carlo N-Particle Transport Code, vol. 2. Los Alamos National Laboratory. Version 4C, LA-13709-M.
Cai, Y., Abd Algfoor, Z., Sunar, M.S., Kolivand, H., 2015. A comprehensive study on pathfinding techniques for robotics and video games. International Journal of Computer Games Technology 2015, 736138. https://doi.org/10.1155/2015/736138 [Online]. Available:
Chao, N., Liu, Y., Xia, H., Ayodeji, A., Bai, L., 2018a. Grid-based RRT*for minimum dose walking path-planning in complex radioactive environments. Ann. Nucl. Energy 115, 73–82 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306454918300069.
Chao, N., Liu, Y., Xia, H., Ayodeji, A., Yang, H., 2018b. Adaptive point kernel dose assessment method for cutting simulation on irregular geometries in nuclear facility decommissioning. Radiat. Phys. Chem. 150, 125–136 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0969806X18300033.
Chao, N., Liu, Y., Xia, H., Peng, M., Ayodeji, A., 2019. DL-RRT* algorithm for least dose path Re-planning in dynamic radioactive environments. Nucl. Eng. Technol. 51 (3), 825–836 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1738573318305564.
Choset, H.M., et al., 2005. Principles of Robot Motion: Theory, Algorithms, and Implementation. MIT press.
Devaurs, D., Siméon, T., Cortés, J., 2016. Optimal path planning in complex cost spaces with sampling-based algorithms. IEEE Trans. Autom. Sci. Eng. 13 (2), 415–424.
E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," Numer. Math., vol. 1, no. 1, pp. 269–271–269–271, [Online]. Available: https://doi.org/10.1007/BF01386390.

Dijkstra, S., 2012. The four-component instructional design model for training complex cognitive skills. Instructional Design: International Perspectives: Volume I: Theory, Research, and Models: volume Ii: Solving Instructional Design Problems 427.
Fensel, D., Motta, E., 2001. Structured development of problem solving methods. IEEE Trans. Knowl. Data Eng. 13 (6), 913–932–913–932.
Ferrari, A., Sala, P.R., Fasso, A., Ranft, J., Siegen, U., others, 2021. FLUKA: a Multi-Particle Transport Code.
Guanglin, Z., Xiaomei, H., Jianfei, C., Lei, Z., Tao, Y., 2011. Summary of path planning algorithm and its application. Modern Machinery 5 (5), 85–90 [Online]. Available: www.scopus.com.
He, S., Zang, Q., Zhang, J., Zhang, H., Wang, M., Chen, Y., May 2016. Development and validation of an interactive efficient dose rates distribution calculation program arshield for visualization of radiation field in nuclear power plants. Radiat. Protect. Dosim. 174 (2), 159–166. https://doi.org/10.1093/rpd/ncw127 [Online]. Available:
Hendricks, J.S., et al., 2005. MCNPX Extensions Version 2.5. 0, vol. 15. Los Alamos National Laboratory.
Islam, F., Nasir, J., Malik, U., Ayaz, Y., Hasan, O., 2012. RRT*-Smart: rapid convergence implementation of RRT* towards optimal solution. In: 2012 IEEE International Conference on Mechatronics and Automation, pp. 1651–1656.
Karaman, S., Frazzoli, E., 2011. Sampling-based algorithms for optimal motion planning. Int. J. Robot Res. 30 (7), 846–894. https://doi.org/10.1177/0278364911406761 [Online]. Available:
Karaman, S., Walter, M.R., Perez, A., Frazzoli, E., Teller, S., 2011. Anytime motion planning using the RRT*. In: 2011 IEEE International Conference on Robotics and Automation, pp. 1478–1483.
Kim, C.H., Cho, S., Jeong, J.H., Bolch, W.E., Reece, W.D., Poston Sr, J.W., 2011. Development of new two-dosimeter algorithm for effective dose in ICRP publication 103. Health physics 100 (5), 462–467–462–467.
Klancar, G., Zdesar, A., Blazic, S., Skrjanc, I., 2017. Wheeled Mobile Robotics: from Fundamentals towards Autonomous Systems. Butterworth-Heinemann.
Knoll, G.F., 2010. Radiation Detection and Measurement. John Wiley & Sons.
Koenig, S., Likhachev, M., 2002. D* lite. Aaai/iaai 15.
Kuffner, J.J., LaValle, S.M., 2000. RRT-connect: an efficient approach to single-query path planning. In: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), vol. 2, 995–1001 vols.2–1001 2.
Likhachev, M., Ferguson, D.I., Gordon, G.J., Stentz, A., Thrun, S., 2005. Anytime dynamic A*: an anytime, replanning algorithm. ICAPS 5, 262–271–262–271.
Matthies, L., et al., 2002. A portable, autonomous, urban reconnaissance robot. Robot. Autonom. Syst. 40 (2), 163–172 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889002002415.
McConnell, J.J., 2001. Analysis of Algorithms: an Active Learning Approach. Jones and Bartlett Publishers, Inc.
Miyombo, M.E., kuo Liu, Y., Ayodeji, A., Nov. 2021. An analytical approach to organ equivalent dose estimation based on material composition photon attenuation coefficient. Prog. Nucl. Energy 141, 103955. https://doi.org/10.1016/J.PNUCENE.2021.103955.
Miyombo, Miyombo Ernest, Liu, Yong-kuo, Ayodei, Abiodun, 2022. A state-aware adaptive pathfinder for dynamic minimum dose path planning during an emergency in a complex radioactive environment. Prog. Nucl. Energy 146 (2022). https://doi.org/10.1016/j.pnucene.2022.104154.
Miyombo, Miyombo Ernest, Liu, Yong-kuo, Ayodeji, Abiodun, 2021. An Optimized Dynamic Algorithm With Photon Attenuation Coefficient for Path-Planning in Radioactive Environments. Am. Soc. Mech. Eng. 85260 https://doi.org/10.1115/ICONE28-64958. In press.
Miyombo, Miyombo Ernest, Liu, Yong-kuo, Ayodeji, Abiodun, 2022. Minimum dose path planning based on three-degree vertex algorithm and FLUKA modeling: radiation source discrimination and shielding considerations. Annals of Nuclear Energy 168. https://doi.org/10.1016/J.ANUCENE.2021.108916.
Noreen, I., Khan, A., Asghar, K., Habib, Z., 2019. A path-planning performance comparison of RRT*-AB with MEA* in a 2-dimensional environment. Symmetry 11 (7), 945.
Pei, Q.-Y., Hao, L.-J., Chen, C.-H., Zheng, X.-L., He, T., 2020. Minimum collective dose based optimal evacuation path-planning method under nuclear accidents. Ann. Nucl. Energy 147, 107644 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030645492030342X.
Persson, S.M., Sharf, I., 2014. Sampling-based A* algorithm for robot path-planning. Int. J. Robot Res. 33 (13), 1683–1708. https://doi.org/10.1177/0278364914547786 [Online]. Available:
Prokhorets, M.I., Prokhorets, S.I., Khazhmuradov, M.A., v Rudychev, E., v Fedorchenko, D., 2007. Point-Kernel method for radiation fields simulation," Вопросы атоМной науки и техники.
Stentz, A., 1997. Optimal and efficient path planning for partially known environments. In: Intelligent Unmanned Ground Vehicles. Springer, 203–220–203–220.
Suteau, C., Chiron, M., Luneville, L., Berger, L., Huver, M., 2003. MERCURE: a 3D Industrial Code for Gamma Rays Transport by Straight Line Attenuation Method Shielding Applications [Online]. Available: http://inis.iaea.org/search/search.aspx?orig_q=RN:37021955.
Taleei, R., Shahriari, M., Feb. 2009. Monte Carlo simulation of X-ray spectra and evaluation of filter effect using MCNP4C and FLUKA code. Appl. Radiat. Isot. 67 (2), 266–271. https://doi.org/10.1016/J.APRADISO.2008.10.007.
Trubey, D.K., Sep. 1988. New Gamma-Ray Buildup Factor Data for Point Kernel Calculations: ANS-6. 4. 3 Standard Reference Data [Online]. Available: https://www.osti.gov/biblio/7032799.
Turner, J.E., 2008. Atoms, Radiation, and Radiation Protection. John Wiley & Sons.
Véras, L.G.D.O., Medeiros, F.L.L., Guimaráes, L.N.F., 2019. Systematic literature review of sampling process in rapidly-exploring random trees. IEEE Access 7, 50933–50953–50933–50953.
Wang, Z., Cai, J., 2018. Probabilistic roadmap method for path-planning in radioactive environment of nuclear facilities. Prog. Nucl. Energy 109, 113–120 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0149197018301884.
West, D.B., others, 2001. Introduction to Graph Theory, vol. 2. Prentice hall, Upper Saddle River.