Path Planning for Reconfigurable hTetro Robot Combining Heat Conduction-Based and Discrete Optimization

HUY DO¹, ANH VU LE¹, OLIVER WEEGER², RAJESH ELARA MOHAN³, YUJIE GUO³, NGUYEN NHAT TAN⁴, MINH BUI VU⁵, AND PHAN VAN DUC⁶

¹ROAR Laboratory, Engineering Product Development Pillar, Singapore University of Technology and Design, Singapore 487372
²Cyber-Physical Simulation Group, Department of Mechanical Engineering, Technical University of Darmstadt, 64289 Darmstadt, Germany
³College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
⁴Wireless Communications Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam
⁵Faculty of Automotive, Mechanical, Electrical and Electronic Engineering, Nguyen Tat Thanh University, Ho Chi Minh City 70000, Vietnam
⁶Faculty of Automobile Technology, Van Lang University, Ho Chi Minh City 70000, Vietnam

Corresponding author: Nguyen Nhat Tan (nguyennhattan@tdtu.edu.vn)

This work was supported in part by the National Robotics Program under its Robotics Enabling Capabilities and Technologies under Project 192 25 00051, in part by the National Robotics Program under its Robot Domain Specific under Project 192 22 00058, and in part by the Agency for Science, Technology and Research.

Abstract
Self-reconfigurable robots present advanced solutions for various automation applications in domains, e.g., planetary exploration, rescue missions, cleaning, and maintenance. These robots have the ability to change their morphology according to given requirements or adapt to new circumstances, which, for example, can overcome constraints while navigating within a working environment. However, the autonomous navigation of self-reconfigurable robots is more complex than that of robots with fixed shape because of the intrinsic complexity of robot motions, especially in complicated obstacle environments. To address this challenge, we present a novel path planning method for reconfigurable robots in this study. The technique is inspired by the similarity between a robot motion path and a heat conduction path at the steady-state. In the heat transfer analysis domain, feasible moving locations are modeled as materials with high conductivity, while obstacles are considered thermal insulators, and the initial and destination positions are assigned as heat sink and heat source, respectively. The temperature profile and gradient calculated by finite element analysis are used to indicate the possible moving directions from the heat sink to the heat source. Based on the temperature gradient ascent, a step-wise conductivity reaching algorithm is developed to optimize robot paths using customized multi-objective functions that take the costs of morphology changes, path smoothness, and safety into account. The proposed path planning method is successfully applied to the hinged-tetro self-reconfigurable robot and demonstrated on several virtual environments and a real-world testbed environment.

Index Terms
Self-reconfigurable robot, tiling robotics, multi-objective path planning, heat conduction, mobile robot.

I. INTRODUCTION
With technological advances, autonomous mobile robots have been offering broad applications in many sectors, such as cleaning and maintenance services [1], [2], transportation, agriculture, healthcare, surveillance, and exploration [3]. To navigate effectively in their working environment, those robots are required to find safe and feasible routes.

To execute autonomous navigation, mobile robots are required to have essential control units, sensors, and intelligent path planning strategies [4]. Therefore, path planning (PP) is a crucial component of research and development in autonomous mobile robots. The goal of PP is to search for a collision-free path from the initial position to the target position while optimizing specific performance criteria. Common target criteria in multi-objective PP algorithms include path length, safety, smoothness, or time and energy efficiency [5], [6].
This work focuses on a novel PP method for such self-reconfigurable robots and demonstrates it in particular application to the Tetris-inspired, hinged-tetro (hTetro) robot proposed by [7], [8]. Specifically, the hTetro is a self-reconfigurable, tiling robot with the ability to shift between seven shapes by rearranging its four tiles, see Fig. 1. Due to the complexity of shapeshifting robots, smooth locomotion among available configurations is required while finding the shortest travel path. Since the reconfigurable robot has several degrees of freedom and additional constraints due to the base footprint size, the shortest path planning techniques applied for fixed form robots needed to be modified to find appropriate solutions.

In this work, based on analogies between a robot motion path and a heat transfer path, we present a novel PP method for reconfigurable robots, which combines heat conduction-based method and discrete optimization. Thus, the major contributions of this work are:

- This is the first time that a heat conduction-based approach is applied to path planning for a polyomino-based reconfigurable robot platform. To leverage the shapeshifting of the reconfigurable robot while navigating through confined narrow space, the proposed method exploits the principle of conduction heat flow to search for robot moving paths on the grid-based workspace rapidly.
- A harmonic function is used to simulate the temperature field, which guarantees that the path solutions are global optimal and avoids the deadlock problem.
- The proposed PP enables the robot to change its shape morphology during navigation to pass through tight areas where fixed shape robots are unable to pass through.
- Simulation and real-world validations on the hTetro robot demonstrated the PP efficiency of saving navigation time and energy spent.
- The presented multi-objective optimization technique is generic and could be applied to other reconfigurable robot platforms.

The rest of the paper is organized as follows. Section II presents the related works about state of the art PP techniques by dividing them into categories. Section III describes the reconfigurable robot platform and the workspace that is being modeled. Section IV introduces the application of the steady-state heat conduction to the robot path planning problem. In Section V, the proposed heat conduction method for hTetro path planning is presented. Section VI presents the results and discussion on the proposed method’s performance. Finally, Section VII are conclusions and recommendations for the future research.

II. RELATED WORKS

Research on PP so far mainly focused on fixed morphology robots and considered robots as a single point. Representative PP algorithms can be broadly divided into four groups: classical graph search algorithms, bio-inspired search methods, geometric algorithms, and virtual potential field methods.

In graph search algorithms, continuous environments of mobile robots are represented by a finite number of vertexes that are connected by edges with their assigned distances. For a given source vertex in the graph, the algorithms find the shortest path between that vertex and every other. Different PP algorithms can be classified according to specific search mechanisms. The most classical graph search algorithm is Dijkstra’s method [9], which is a special form of dynamic programming. However, its computational cost is high for complex and high-dimensional problems. The A* algorithm [10] extends Dijkstra’s method with a heuristic estimation of the remaining cost to the goal state. Thus, it reduces the total number of searching states and accelerates the convergence rate. Nevertheless, the heuristic estimation function must be close to the real cost, which is crucial for the method’s overall performance. Other graph search methods include D* [11], D* Lite [12] and Lifelong Planning A* (LPA) [12], etc., where specific improvements are proposed to enhance the performance of the A* algorithm. In general, the results of graph search algorithms for robot path planning are configuration dependent, i.e., they depend on the structure of the generated grids and nodes.

In bio-inspired algorithms, two main sub-categories can be distinguished, namely Evolutionary Algorithms (EA) and Neural Networks (NN), which mimic the behaviors of natural creatures or human beings. A typical bio-inspired EA is the Genetic Algorithm (GA) [5], [13], [14], which is able to solve the NP-hard problems with a large number of variables. However, the performance of GA depends on the diversity of
the population. Ant Colony Optimization (ACO) [15], [16] is another well-known EA method, which mimics the movements of a group of ants from their colony to the destination. ACO is able to deal with multi-objectives and continuous planning problems. However, it may suffer from the high computational effort for solving complex problems. Other EA methods such as Particle Swarm Optimization (PSO) [17], [18], Memetic Algorithm [19] and Shuffled Frog Leaping Algorithm [20] share almost the same advantages and disadvantages with the GA and ACO. Neural Networks mimic the way neural circuits process information and are able to deal with dynamic environments. Nevertheless, NN also suffers from high computational complexity and relies much on the chosen rules and organisms [21]–[23].

Geometric shortest path algorithms search for the optimal path based on geometric algorithms and sampling strategies. The Rapid-exploring Random Tree (RRT) [24], [25] algorithm, proposed by LaValle [26], searches for a feasible path based on the configuration space where obstacles can be predefined. The RRT algorithm is able to deal with multi-DOF problems and has the fast searching ability, but it pays no attention to the quality of the results [27]. Another popular algorithm is the Probabilistic Road Map (PRM) method [28], which considers different choices for the set of states to which connections are attempted. However, the collision check may become expensive with the expansion of the exploration graph. Other well-known geometric algorithms include the Voronoi graph [29], visibility graph [30] and cell decomposition [31] methods, etc. In general, the performance of these methods relates much to the sampling strategies adopted, but they can be adapted to various environments and are computationally efficient, which is desirable for online implementation in robot PP.

Virtual potential field methods, e.g., [32], [33], define a potential function over the free domain of the robot configuration space using an attractive potential that pulls the robot toward the goal position and a repulsive potential that pushes the robot away from obstacles [34]–[36]. These methods can achieve a fast and reactive response to the dynamic environment and can be applied to higher-dimensional problems [37], [38]. However, they often suffer from the drawback that the robot may get trapped in local minima and oscillations. A possible remedy to the local minimum problem is to reformulate the artificial field function [39] or compute the potential with constraints [40], [41].

Based on a virtual potential field method, Wang and Chirikjian [42] first presented a steady-state heat conduction approach applied to robot PP problems. Obstacles and feasible areas are identified by their thermal conductivity, and the optimal path from the heat source (start position) to the heat sink (goal position) is considered as the heat flow with minimal thermal resistance. Although their work is of original importance, the method cannot ensure that all feasible paths avoid obstacles. Ryu et al. [43] formulated PP as a topology optimization problem by minimizing the thermal compliance, in which obstacles were modeled as regions of zero-thermal conductivity. The approach is a typical pixel-based topology optimization framework and can be directly used in PP. Inspired by this research direction, Li et al. [44] developed a conductivity spreading method using cooling channels to create a heat transfer path. Their method can remove the dependency of the solution on resolving the computational grid and provide an optimal path with explicit geometrical representation.

In general, each of these algorithms has its advantages and shortcomings in finding the most efficient solution. Thus, a combination of certain algorithms may perform rather well to achieve a global optimum and cost minimum simultaneously. Furthermore, mobile robots commonly operate in complex and dynamic environments rather than in predefined ones. For instance, a pavement sweeping robot operates in a dynamic environment with moving pedestrians, and objects [45]–[47]. The requirement of implementing precisely autonomous tasks in uncertain environments is crucial for the development of next-generation robots. In such circumstances, self-reconfigurable robots [22], [48], [49] are perfect candidates since they have the ability to deliberately change their shape or morphology by rearranging the connectivity of their parts to adapt to the dynamic environments. Although self-reconfigurable robotics have attracted a significant amount of attention over the past thirty years, the development of autonomous systems mostly focused on autonomous motion control and mechanism design, and there are only few works on PP for self-reconfigurable robots, e.g., [50]–[53]. Recently, Cheng et al. [54] combined a novel graph theory-based model and a dynamic programming method to simulate the coverage path planning of reconfigurable robots. Furthermore, the multi-objective path planning problem of the hinged-tetro reconfigurable tiling robot was studied based on the Genetic Algorithm [5]. Other related works utilized a modified A-star algorithm [55], and reinforced machine learning [56].

III. HINGED-TETRO RECONFIGURABLE ROBOT

The hTetro applied the idea of a chain-type inter-reconfigurable mechanism to enable the shape-shifting of the robot base (see Figures 1 and 2). This idea enables hTetro to maximize the area cover and provide a feasible solution to find the optimal path to overcome the environmental constraints.

The hTetro robot platform that was first developed by Prabakaran et al. [7] and extended for floor cleaning purposes in [8], [53], [55]. It is based on the principle of Tetris and consists of four isosceles right-angle triangular blocks connected with active hinges. We chose the right angle isosceles poly-form as our robot structure to achieve maximum area coverage by changing its defined morphologies among I, L, O, J, Z, T, and S shapes, as shown in Figure 1.

The robotic device is categorized into several subsystems such as locomotion, reconfigurable mechanism, structural design, and electronic circuits. This subsystem acts as an essential component that combines to achieve environment
adaptation and obstacle detection. The structural dimension of each isosceles triangular block was developed with a dimension of 210 mm in adjacent and 294 mm in hypotenuse. The block’s vertices are positioned as chambers to skip the edge collision between the blocks during reconfiguration. The robot’s walls and base were constructed with an acrylic sheet with 2mm of thickness. The robot is equipped with a set of Herkulex motor and Pololu DC motor in each locomotion module. The herkulex motor act as a steering motor, and the dc motor drives the robots as in Figure 1. With such the arrangement, the robot could achieve holonomic locomotion. Each locomotion motor set was powered with 7.4 VDC. Concerning the reconfiguration, we again equipped two Herkulex servo motors housed in block 1 and block 2.

For localization, the Lidar’s range information and the IMU data are fused in the robot localization package of Operation System (ROS) [57]. Then, using the robot’s global position, the proposed navigation algorithm will generate the appropriate path. This global path will be passed to the ROS navigation stack wherein the local path planner generates the command velocity for the robot that passes to the local controller (Arduino). The local controller later passes the PWM values to the motor.

### A. hTetro ROBOT MODEL

We use a grid-based workspace $\mathcal{W} \subset \mathbb{R}^2$ in the 2-D Cartesian space as the hTetro working environment (see Figure 2). The grid width equals the hTetro single block width, $d_{\text{grid}} = dB$. The geometries of the four hTetro blocks are represented as $B_n \subset \mathbb{R}^2 (n = \{1, 2, 3, 4\})$ of the same width $dB$. The angle between the block local frames in the workspace are depicted as $\theta_{B_n} \in \mathbb{R} (n = \{1, 2, 3, 4\})$ and the counterclockwise is set as the positive direction. The hinges are represented as $H_n (n = \{1, 2, 3\})$, and the hinge angles between two different blocks are denoted as $\theta_{H_n} \in \mathbb{R} (n = \{1, 2, 3\})$.

![FIGURE 2. The hTetro system model.](image)

### TABLE 1. Angles of hinges for the seven shapes of hTetro.

| Morphology | $[\theta_{H_1}, \theta_{H_2}, \theta_{H_3}]$ | Morphology | $[\theta_{H_1}, \theta_{H_2}, \theta_{H_3}]$ |
|------------|---------------------------------|------------|---------------------------------|
| I-shape    | $[\theta, \theta, \theta]$      | O-shape    | $[\theta, \theta, \theta]$      |
| L-shape    | $[\theta, \theta, \theta]$      | O-shape    | $[\theta, \theta, \theta]$      |
| Z-shape    | $[\theta, \theta, \theta]$      | S-shape    | $[\theta, \theta, \theta]$      |
| T-shape    | $[\theta, \theta, \theta]$      | S-shape    | $[\theta, \theta, \theta]$      |

The rotation values of $\theta_{H_n}$ are presented as:

$$\frac{\pi}{2} \leq \theta_{H_1} = \frac{\pi}{2} + \theta_{B_1} - \theta_{B_2} \leq \frac{3\pi}{2}$$
$$\frac{\pi}{2} \leq \theta_{H_2} = \frac{\pi}{2} + \theta_{B_2} - \theta_{B_1} \leq \frac{3\pi}{2}$$
$$\frac{\pi}{2} \leq \theta_{H_3} = \frac{\pi}{2} + \theta_{B_3} - \theta_{B_4} \leq \frac{3\pi}{2}$$

Each hinge is allowed to rotate freely as long as all three hinge angles fulfill the constraints above. Several combinations of the hinge angles $\theta_{H_n} (n = \{1, 2, 3\})$ form shapes that simulate those of the 7 one-sided tetrominoes, as shown in Figure 1. Table 1 depicts the hinge angle combinations to shift the robot between seven available shapes of hTetro. The ability to freely transform will allow the robot to find ideal configurations to avoid obstacles and efficiently reach the destinations. However, the trade-offs of performing reconfiguration are the increased time and energy consumption during the process.

### B. MODEL OF THE MOTION OF hTetro ON THE WORKSPACE

The hardware architecture of the hTetro is the self-reconfigurable platform where the robot’s block makes use of four omnidirectional wheels as its moving mechanism (instead of differential wheels) and utilizes hinge motors to shapeshift the robot forms. During navigation, we define the translation ($T$), rotation ($R$), and shape-shift ($S$) motions.

By using omnidirectional wheels, the hTetro robot is capable of performing an instant change of its moving direction, while robots equipped with differential drive mechanisms are required to perform a U-turn to reverse their moving direction. A single translation motion command ($T$) moves the hTetro robot to any of the four linear directions ($\pm x$ or $\pm y$ in $\mathcal{W}$) for a distance of $d_{\text{grid}}$ and to any of the four diagonal directions ($\pm x$ and $\pm y$ in $\mathcal{W}$) for a distance of $\sqrt{2}d_{\text{grid}}$.

The robot platform’s stability also allows the hTetro robot to perform pivot rotation concerning any point on the workspace. When a hTetro robot rotates, it rotates around the axis that passes through the center of $B_2$ and perpendicular to the x-y plane of $\mathcal{W}$. Thus, a single rotation motion command ($R$) drives the robot for 90° in the direction of either clockwise or counterclockwise.

The shapeshift motions depicted as ($S$) transform the robot to the desired shape $M$ from the initial hTetro block angles $\theta_{B_n}$. The required hinge angle for shape $M$ presented in Table 1 ($\theta_{H_n}^M$) determine the heading angle offset ($\Delta \theta_{H_n}$).
of each robot block as follows:
\[
\Delta \theta_{B_1} = \theta_{B_1}^i - \theta_{B_1}^i - \theta_M - \frac{\pi}{2} \\
\Delta \theta_{B_2} = \theta_{B_2}^i - \theta_{B_2}^i - \theta_M - \frac{\pi}{2} \\
\Delta \theta_{B_3} = \theta_{B_3}^i - \theta_{B_3}^i - \theta_M - \frac{\pi}{2} \\
\Delta \theta_{B_4} = \theta_{B_4}^i - \theta_{B_4}^i - \theta_M - \frac{\pi}{2}
\]

Therefore, here, a single shapeshift command \((S)\) will directly transform the hTetro robot into the desired shape.

With the definitions of each hTetro motion above, the configuration motion table is listed in Table 2.

The shortest path planning algorithms developed for fixed morphologies platforms attempt to search for the best series of translational movement commands that navigate the robot to its destination with the shortest distance or travel time. However, this route optimization for reconfigurable robots such as the hTetro is more complex since it could perform three different motions during the navigation. Thus, defining minimum distance traveled as the sole optimization goal for the hTetro would completely omit the possibilities and costs of rotation and shapeshifting motions. Therefore, an alternative optimization goal must be defined, which will be introduced later in Section V-A, where the multi-objective evaluation technique is being implemented.

**IV. HEAT CONDUCTION-BASED PATH PLANNING**

In the past decades, artificial potential fields based on harmonic functions have been described by different physical analogies, e.g., electrostatics, incompressible fluids dynamics, and mechanical stress. The artificial potential field approach has been widely used in robot path planning owning to the property of harmonic functions that overcome the deadlock problem, i.e., the presence of local minima in the potential function [36], [42]. In this study, the temperature is utilized as the artificial potential field to identify the desired paths; thus, path planning is formulated as a steady-state heat transfer problem [42]. The analogy between a heat transfer path and a robot motion path is indicated by Table 3.

The robot travel space \(C \subseteq \mathbb{R}^2\) (also called workspace) consists of four different kinds of regions: the feasible moving region \(C_F\), obstacles \(C_i, i = 1, \ldots, n_0\), a start position \(C_S\), and a goal position \(C_G\) [43] as shown in Figure 3a:

\[
C = C_F \cup \bigcup_{i=1}^{n_0} C_i \cup C_S \cup C_G \quad (1)
\]

In equation (1), we assume that the configuration space can be discretized into a grid of squares, of which \(C_S\) and \(C_G\) coincide with the start and goal positions of the hTetro block 2.

Analogously, the heat analysis domain \(\Omega \subseteq \mathbb{R}^2\) consists of four kinds of subdomains: the conductive region \(\Omega_F\), insulated regions \(\Omega_i, i = 1, \ldots, n_0\), a heat sink \(\Omega_S\), and a heat source \(\Omega_G\), as shown in Figure 3b:

\[
\Omega = \Omega_F \cup \bigcup_{i=1}^{n_0} \Omega_i \cup \Omega_S \cup \Omega_G \quad (2)
\]

In the mapping relationship, shown in Figure 3, feasible regions and obstacles correspond to high thermal conductivity and zero conductivity regions, and start and goal locations are considered as heat sink and heat source regions, respectively:

\[
C \leftrightarrow \Omega, \quad C_\alpha \leftrightarrow \Omega_\alpha (\alpha = F, S, G), \quad \times C_i \leftrightarrow \Omega_i (i = 1, \ldots, n_0) \quad (3)
\]

In the heat analysis domain \(\Omega\), heat always flows from the heat source towards the heat sink and is not conducted through the region with zero thermal conductivity. Thus, the heat flux is a vector field pointing from the high-temperature region to the low-temperature region. This phenomenon is known as thermal conduction and is described by Fourier’s law:

\[
q(x) = -k(x) \nabla u(x), \quad x \in \Omega, \quad (4)
\]
where \( q \) is the heat flux vector for a given temperature profile \( u : \Omega \rightarrow \mathbb{R} \) and \( k \in \mathbb{R} \) is the thermal conductivity. The minus sign means that heat flows down the temperature gradient, i.e., from high temperature to low temperature.

The temperature profile \( u \) within the heat analysis domain \( \Omega \) depends on the rate of its internally generated heat, its capacity to store some of this heat, and the rate of thermal conduction on its boundaries. The steady-state of the temperature field \( u \) in the heat analysis domain \( \Omega \) is described by the generalized Poisson’s equation:

\[
\text{div}(-k(x)\nabla u(x)) = Q(x) \quad \text{in } \Omega, \tag{5}
\]

where \( Q \) is the rate of heat generated per unit volume. The elliptic boundary differential equation (5) is completed as a boundary value problem with boundary conditions that prescribe either the temperature \( u \) on the boundary \( \partial \Omega_1 \) or the heat flux \( q \) on \( \partial \Omega_2 \) as:

\[
u(x) = \bar{u}(x) \quad \text{on } \partial \Omega_1, \tag{6}
\]

or

\[-q(x) \cdot n(x) = \bar{q}(x) \quad \text{on } \partial \Omega_2, \tag{7}\]

where \( \partial \Omega = \partial \Omega_1 \cup \partial \Omega_2 \) and \( n \) is the outer normal on \( \partial \Omega_2 \). To approximate the solution of the boundary value problem defined by equations (5)–(7), i.e., to obtain the temperature field \( u \), its weak form can be discretized using the finite element method (see, e.g., [58]), which ultimately leads to a linear system of equations:

\[KU = F, \tag{8}\]

where \( K \), \( U \), and \( F \) denote the stiffness matrix, the nodal temperature vector, and the thermal load vector, respectively. The values in the solution vector \( U \) describe the temperatures at the nodes of the discretized domain \( \Omega \) and can be used to evaluate \( u \), \( \nabla u \), or \( q \) anywhere in \( \Omega \).

V. PATH PLANNING METHOD FOR RECONFIGURABLE ROBOTS

Now, the proposed PP method for reconfigurable robots is presented. As summarized in the flowchart in Figure 4, the grid-based search method consists of three main phases. In the first phase, the 2D Cartesian workspace of the hTetro robot is discretized into a set of square-shaped elements with grid size \( d_{\text{grid}} \). In the second phase, the PP problem is transformed to a corresponding heat transfer problem using a mapping relationship, as shown in Section IV. Then, a finite element analysis on the heat conduction domain is performed to calculate the temperature profile that indicates the direction of the heat flow from the heat source to the heat sink, i.e., the shortest path from the start location to the goal location. Finally, in the third phase, the robot moving path is determined using a grid-based gradient ascent approach that considers not only the path length but also the necessities and costs of the involved motions, i.e., translations, rotations, and shapeshifts.

A. COST FUNCTIONS

For reconfigurable robots, several types of costs can be included in the definition of the objective function of the PP problem. Here, we consider cost terms for time consumption,
path smoothness, and path safety. In single-objective optimization, either of these three terms will be regarded as, while in multi-objective optimization, a weighted linear combination of all three terms is used to define the cost function to be maximized.

For a path \( p = (p_1, \ldots, p_{l_p}) \), i.e., a sequence, of \( l_p \) motion commands \( p_i \in m_c = \{T_x, T_y, T_x, T_y, T_x, T_y, T_x, T_y, T_x, T_y, T_x, T_y, R_x, R_y, S_M\} \), see Table 2, the three cost functions selected for the hTetro are introduced in detail in the following:

1) TIME CONSUMPTION COST \( (f_t) \)
The total time consumption cost of the entire path is calculated as:

\[
f_t(p) = 1 - \frac{\sum_{i=1}^{l_p} t_{p_i}}{t_{max}},
\]

where \( t_{p_i} \) is the time consumption for a hTetro motion command \( p_i \), as shown in Table 2. \( t_{max} \) is the consumption time that the hTetro robot in the O-shape performs 50 percent of a complete coverage path on the free-obstacle environment. For instance, in an environment with a size of \( 24 \times 24 \) cells, \( t_{max} \) is set to 39.6 seconds.

2) PATH SMOOTHNESS COST \( (f_{sm}) \)
The path smoothness cost term is calculated as:

\[
f_{sm}(p) = 1 - \frac{\sum_{i=1}^{l_p} \text{diff}(p_i, p_{i+1})}{l_p},
\]

where:

\[
\text{diff}(p_i, p_{i+1}) = \begin{cases} 1, & \text{if } p_i \neq p_{i+1} \\ 0, & \text{if } p_i = p_{i+1} \end{cases}
\]

Here, a smaller change between consecutive motion commands during navigation leads to a better path smoothness cost. Thus, this objective function promotes paths with high stabilities of the robot’s motion commands.

3) PATH SAFETY COST \( (f_{sf}) \)
The path safety cost term measures the security of an entire robot path during navigation and is calculated as:

\[
f_{sf}(p) = 1 - \frac{1}{l_p} \sum_{i=1}^{l_p} \sum_{j=1}^{4} \sum_{(x,y) \in \Omega_{sp}} W_{obs}(P_{p_{ij}} + (x,y)) \| (x,y) \|_2^{-2},
\]

where:

\[
W_{obs}(x,y) = \begin{cases} 1, & \text{if } (x,y) \in \Omega_0^k \\ 0, & \text{otherwise} \end{cases},
\]

and \( \Omega_{sp} \) is the searching pattern, i.e., the surrounding domain in terms of circles with predefined radii \( r \), here \( r = 2 \cdot d_{grid} \), and center \( P_{p_{ij}} \) at each hTetro block \( j \), in which \( (x,y) \in \Omega_{sp} \) are position vectors with respect to that center. Basically, in equation (11), the too-close approach of any robot block to an obstacle domain is penalized.

Thus, obstacles that present in the obstacle searching circles of a path will decrease its safety value \( f_{sf} \), and the robot’s moving paths would be less desired during the gradient-driven grid search process. Figure 5 visualizes a searching pattern for O- and I- shaped morphologies with radius of \( 2 \cdot d_{grid} \). The value in a cell represents the number of the circles (the center at one of the 4 robot blocks with a radius of \( 2 \cdot d_{grid} \) that occupy that cell. Therefore, a cell marked by 3 is closer to the robot than a cell marked by 2 and 1.

For safety, the robot will move on the obstacle-free path with the lowest summation of all the values marked in obstacle cells near the robot during navigation. For instance, Fig. 5a shows a searching pattern including 8 cells adjacent to the robot with the value of 3 and the next surrounding 12 cells with a value of 1. Within this searching pattern, the obstacle grids include 2 cells with a value of 3 and 4 cells with a value
of 1. Therefore, the accumulated value of $W_{obs}$ is 10 and the $f_{sf}$ value for this particular motion command is $1 - 10/20 = 0.5$. In the same way, safety cost evaluation for the case of Fig. 5b, $f_{sf}$ is $1 - 14/24 = 0.417$. Implementing this cost function favors the moving paths that remain at a certain distance to the obstacles.

4) MULTI-OBJECTIVE COST FUNCTION ($f_{mul}$)

For multi-objective optimization, the overall cost function $f$ of a path $p$ is calculated as the weighted sum of time consumption $f_t(p)$, smoothness $f_{sm}(p)$, and safety $f_{sf}(p)$ cost terms:

$$f(p) = \frac{w_t f_t(p) + w_{sm} f_{sm}(p) + w_{sf} f_{sf}(p)}{w_t + w_{sm} + w_{sf}}, \quad (12)$$

where $w_t, w_{sm}, w_{sf} \geq 0$ are the weights corresponding to time consumption, smoothness, and safety cost functions, respectively.

**B. PATH FINDING USING GRID SEARCH WITH GRADIENT ASCENT**

The path-finding method uses the temperature gradient from the calculated heat map to search for the cost-optimal path $p$ that connects the start cell $C_S$ (referring to block 2 of hTetro) at the heat sink $\Omega_S$ to the goal cell $C_G$ at the heat source $\Omega_S$. For reference, a list of terminologies used in the method is shown in Table 4 and a flowchart of the path-finding process for hTetro navigation is shown in Figure 6.

Using the cost function $f(p)$ as defined (12), the method aims to minimize the travel distance of paths between the start cell $S$ and all the other cells (for instance, cells $U, V$).
TABLE 5. Numerical inputs used by the thermal conduction-based method.

| Term                                      | Value         | Note                                      |
|-------------------------------------------|---------------|-------------------------------------------|
| Number of finite elements (grid cells)    | 24 × 24       | Workspace resolution                      |
| Element edge length d_{grid}              | 1 m           | Arbitrary positive value                  |
| Thermal conductivity of the conductors \( \kappa_c \) (feasible moving regions \( \Omega_P \)) | 1 W/(m²K)     | Arbitrary positive value                  |
| Thermal conductivity of the insulators \( \kappa_i \) (obstacles \( \Omega_i \)) | \( 1 \times 10 \) W/(m²K) | Arbitrary positive value                  |
| Heat flux at 4 nodes at the heat source element (the goal cell \( \Omega_G \)) | \( 24^4/4 \) W/m² | Arbitrary positive value                  |
| Heat flux at 4 nodes at the heat sink element (the start cell \( \Omega_S \)) | \(-24^4/4 \) W/m² | Arbitrary negative value                  |
| Prescribed temperature at the boundaries (all nodes on \( \partial \Omega \)) | 0 K           | Standard Dirichlet boundary condition      |

The method initializes some cost values of cells and tries to improve them step-wise as follows:

1) Initialize a tentative cost value to every cell: set it to zero for start cell \( S \) and infinity for all other cells \( V \). Add the start cell \( S \) into the current reaching list \( list \).

2) For every cell \( U \) member of the current list, consider all of its equal or higher temperature neighbors, e.g., \( V \), and calculate their tentative costs through cell \( U \). Compare the newly calculated tentative cost \( f(U) + f(U, V) \) to the current assigned value \( f(V) \) and assign the smaller one. If \( f(V) \) is reassigned, Add cell \( V \) into \( listNext \) and previous cell of \( V \) assigned to \( U \) by \( prev(V) = U \).

3) Update the current list \( list \) by the next one \( listNext \).

4) If the goal cell \( G \) has been reached (\( G \) in \( list \)), (then output the optimal path by tracking from node \( G \) back to cell \( S \) using the cell array \( prev() \) ) or \( G \) cannot be reached if the current list is empty and \( G \) is not in \( list \) (when the reaching process has been completed but there is still no connection between the start cell \( S \) and remaining cells including \( G \) ), then stop. The reaching process has stopped.

5) Otherwise, go back to step 2.

VI. RESULTS AND DISCUSSION

A. PATH PLANNING OF hTetro IN VIRTUAL ENVIRONMENT

The performance of the proposed path planning method is evaluated through simulations on six benchmark virtual environments. The scenarios with different obstacles regarding sizes, shapes, and location distributions are chosen for comprehensive evaluations of the method. In all cases, both start and goal configurations are in O-shaped morphology.

Although the path planning problems are formulated as heat conduction problems to find robot moving directions by showing the temperature gradients, it is not necessary to directly associate the results with heat transfer phenomena [43], [44]. Therefore, heat conductivity input values for the finite element analysis are quite arbitrary without influence on the temperature gradients. In all the tests, the environments are meshed into 24 × 24 elements, and the specific input values are listed in Table 5. The simulations are performed on a personal laptop with an Intel Core i7-3520M CPU (2.9 GHz, 2 cores, 4 threads) and 8 GB RAM. Both the finite element analysis and path-finding procedures are implemented in MATLAB R2019a, and one simulation takes a second to give the optimized path solutions.

Figures 7–10 show four virtual obstacle environments, temperature maps, and optimized path results for best time.
FIGURE 8. H-shaped obstacle environment and optimized path results based on: time consumption cost ($f_t$), smoothness cost ($f_{sm}$), safety ($f_{sf}$), and multi-objective ($f$).

FIGURE 9. Spiral obstacle environment and best path results based on: time consumption cost ($f_t$), smoothness cost ($f_{sm}$), safety ($f_{sf}$), and multi-objective ($f$).

costs. The temperature fields obtained by the finite element analysis using 4-noded, bilinear, quadrilateral element are shown in Figures 7b, 8b, 9b, and 10b. These figures show that the heat transfers from the heat source to other feasible moving locations via conductor paths. The temperature field $u$ has the highest value at the heat source and gradually decreases at locations farther from the heat source. The temperatures surrounding the insulators are almost zero due to their low thermal conductivity. In the figures, the arrows represent the gradient directions $\nabla u$ of the temperature field at finite element nodes, which indicates the shortest moving directions of the robot for one step during the optimization process.

The cost function values of the optimized paths for all six simulation environments are listed in Table 6. It is noted here that the cost function is formulated such that it is maximized, i.e., values closer to 1 are more optimal. From equation (12), the specific optimal cost functions $f_t$, $f_{sm}$, $f_{sf}$, and $f$ are calculated by the path finding algorithm using variations of weight factors $w_t$, $w_{sm}$, and $w_{sf}$ as shown in the third column. For example, the path with best time consumption
$f_t$ is calculated as $f$ with $w_t = 1$ and $w_{sm} = w_{sf} = 0$. For a comprehensive comparison, each optimal path is minimized based on a criterion, and its best cost value and other cost values are listed in the same row of the table, even though they might not be considered in $f$. Thus, it can be seen that, for each environment, the maximum cost value in a column of a criterion is the value of the element in the row having the same name of that criterion. For example, in the random obstacle environment, the best safety cost is $0.945$ in the row and column of “Safety”, and the best multi-objective cost is $0.892$ in the row and column of “Multi-objective”.

**TABLE 6.** Cost function values of optimized paths.

| Environment | Best path of | Weights $w_t$, $w_{sm}$, $w_{sf}$ | Time consumption $f_t$ | Smoothness $f_{sm}$ | Safety $f_{sf}$ | Multi-objective $f$ |
|-------------|--------------|-----------------------------------|------------------------|-------------------|-------------------|---------------------|
| Random      | Time consumption | 1, 0, 0                          | 0.894                  | 0.800             | 0.865             | 0.853               |
|             | Smoothness   | 0, 1, 0                          | 0.867                  | 0.941             | 0.849             | 0.886               |
|             | Safety       | 0, 0, 1                          | 0.879                  | 0.615             | 0.945             | 0.813               |
|             | Multi-objective | 1, 1, 1                          | 0.894                  | 0.880             | 0.902             | 0.892               |
| H-shape     | Time consumption | 1, 0, 0                          | 0.869                  | 0.875             | 0.766             | 0.837               |
|             | Smoothness   | 0, 1, 0                          | 0.869                  | 0.875             | 0.875             | 0.837               |
|             | Safety       | 0, 0, 1                          | 0.851                  | 0.722             | 0.926             | 0.833               |
|             | Multi-objective | 1, 1, 1                          | 0.864                  | 0.813             | 0.914             | 0.864               |
| Spiral      | Time consumption | 1, 0, 0                          | 0.667                  | 0.908             | 0.702             | 0.759               |
|             | Smoothness   | 0, 1, 0                          | 0.664                  | 0.920             | 0.704             | 0.763               |
|             | Safety       | 0, 0, 1                          | 0.652                  | 0.864             | 0.862             | 0.793               |
|             | Multi-objective | 1, 1, 1                          | 0.652                  | 0.864             | 0.862             | 0.793               |
| 3-slit      | Time consumption | 1, 0, 0                          | 0.832                  | 0.684             | 0.773             | 0.763               |
|             | Smoothness   | 0, 1, 0                          | 0.814                  | 0.777             | 0.773             | 0.788               |
|             | Safety       | 0, 0, 1                          | 0.814                  | 0.777             | 0.773             | 0.788               |
|             | Multi-objective | 1, 1, 1                          | 0.814                  | 0.777             | 0.773             | 0.788               |
| Zigzag      | Time consumption | 1, 0, 0                          | 0.806                  | 0.816             | 0.529             | 0.717               |
|             | Smoothness   | 0, 1, 0                          | 0.798                  | 0.827             | 0.551             | 0.725               |
|             | Safety       | 0, 0, 1                          | 0.766                  | 0.649             | 0.596             | 0.670               |
|             | Multi-objective | 1, 1, 1                          | 0.798                  | 0.827             | 0.551             | 0.725               |
| Maze        | Time consumption | 1, 0, 0                          | 0.857                  | 0.686             | 0.680             | 0.741               |
|             | Smoothness   | 0, 1, 0                          | 0.848                  | 0.682             | 0.659             | 0.786               |
|             | Safety       | 0, 0, 1                          | 0.857                  | 0.686             | 0.680             | 0.741               |
|             | Multi-objective | 1, 1, 1                          | 0.848                  | 0.827             | 0.641             | 0.787               |
1) RANDOM

An environment with six “randomly” distributed obstacles is shown in Figure 7a. All the optimized paths are perfectly “smooth” for this environment, moving only from starting to goal configurations in the O-shaped configuration. This is the case since translational motions in sparse obstacle environments are better than shapeshifts or rotations in terms of saving time consumption cost. Thus, the optimized time consumption costs based on all the four criteria are varied in a good range from 0.879 to 0.894. On the contrary, the path of best smoothness navigates to another direction, and its time consumption and safety are lower with values of 0.867, and 0.615, respectively. Another remark on this result set is that the best safety path (with a very good \( f_{st} = 0.945 \)) is better than other paths in keeping a certain distance from the obstacles.

2) H-SHAPE

An example with a H-shaped obstacle environment is shown in Figure 8. Here, start, goal and obstacles are symmetrically located, which results in the symmetry of temperature distribution and temperature gradient (see Figure 8b). All the
four optimized paths form a C-shape bounding the left side of the H-shaped obstacles as shown in Figure 8a. The optimized paths for best time consumption and best smoothness are completely overlapping and they have the same cost values in comparisons of all cost functions ($f_t = 0.869, f_{sm} = 0.875, f_{st} = 0.766$). Compared to those paths, the optimized path for best multi-objective has almost the same cost values of time consumption ($f_t = 0.864$) and a lower smoothness value ($f_{sm} = 0.813$), however, it has a better safety value ($f_{st} = 0.914$) as it keeps some distance from the left side of the H-shaped obstacle. The best safety path in this environment has a longer time of travel ($f_t = 0.851$) in a larger C-shaped to reduce collision probabilities with the obstacles ($f_{st} = 0.926$).

3) SPIRAL
In the spiral obstacle environment example shown in Figure 9, the temperature gradient (see Figure 9b) forms a heating channel between starting and goal positions. Therefore, as shown in Figure 9a, all the four optimized paths remain in the O-shape with a good smoothness range of $f_{sm}$ from 0.862 to 0.920 and have very similar time consumption values with $f_t$ varying from 0.652 and 0.667. The optimized paths for best time consumption and best smoothness are almost overlapping and close to the obstacles with bad safety values of $f_{st} = 0.702$ and 0.704, while the optimized paths for best safety and best multi-objective are overlapping each other and have much better safety values of $f_{st} = 0.862$ by keeping some distance from the obstacle to minimize collision probabilities.

4) 3-SLIT
Different from the previous cases, in which the path planners are successful from the initial location to the final location using only O-shaped morphology, in a 3-slit environment, as shown in Figure 10, the hTetro robot also requires the I-shaped morphology in the navigation process. The optimized path solutions, shown in Figure 10a, demonstrate that the algorithm can assign free locations where shapeshift and rotation commands are executable to navigate through narrow channels created by the obstacles. A common point of the optimized paths is that after passing the narrow channels, the robot remains in the I-shape and uses translational motions to reduce some time cost $f_t$ and to increase consistency of motion commands $f_{sm}$. While the optimized paths based on smoothness, safety, and multi-objective are the same, the optimized path for best time consumption has a slightly better time consumption value due to utilizing some diagonal translational motions. It can also be seen that all shapeshift positions of the paths are far enough from the obstacles to improve safety costs.

5) ZIGZAG
In the Zigzag environment, as shown in Figure 11a, three long obstacles are arranged in parallel to each other to form a zigzag path with three tight spaces. As expected, the temperature gradient shown in Figure 11b indicates a feasible moving channel from the start to the goal positions. Again, like in the 3-slit environment, the optimized path solutions shown in Figure 11a demonstrate that the algorithm can determine valid positions for the robot to perform shapeshifts to the I state to pass through the tight spaces. It can be seen that the optimized paths for best smoothness and best multi-objective overlap and that they are similar to the optimized path for best time consumption. In a different way, the optimized path for best safety uses the I shape in its rotated position parallelled to the moving channel to increase the safety values. However, this optimized path is longer and less smooth than the other paths and has a lower score value in overall.

6) MAZE
In the Maze environment shown in Figure 12a, several U- and L-shaped obstacles are scattered to form some U-shaped traps. Furthermore, it can be seen in Figure 12b that the temperature gradient indicates two feasible moving channels from the start to the goal positions. This also shows the heat conduction approach’s benefit in guaranteeing that the robot is not trapped during navigation. On the first channel, the optimized paths for best time consumption and best safety are entirely overlapping each other. On these paths, the robot uses I-shape and I-rotation to navigate through two tight spaces. On the second channel, the optimized paths for best smoothness and best multi-objective are almost the same. Although the robot uses only O-shape to keep the motion command consistent, it leads to lower scores in time consumption and safety compared to the other paths.

7) REACTIVE NAVIGATION IN AN UNKNOWN OBSTACLE ENVIRONMENT
Information about the environment is not always completely known before the robot motion begins. In such situations, the online navigation algorithm of the proposed method is used. During the navigation, the data of the environment are feed from robot on-board sensors and are then used to update the simulation workspace for each sampling interval of one motion command. Here, the observation range of the sensors is set as $D_S = 6 \cdot d_{\text{grid}}$.

For another Maze environment example shown in Figure 13a, the proposed method is performed to seek optimal paths for the shortest distance in two obstacle scenarios: fully known and partially known environments. In the first scenario, there is no environmental change, and all the obstacles are fully known. As can be seen in Figure 13b, the temperature gradient indicates two feasible moving channels from the start position (2,2) to the goal position (22,21). Within the first moving channel, the optimized path for the shortest distance is marked red in Figure 13a. In the second scenario, the partially known environment consists of an unknown obstacle at cells (5,11) and (6,11) and the known obstacles. As the initial plan, the robot moves on the same path from the start position (2,2) toward the cell (5,7). Then, the robots’ sensors detect environment changes, i.e., that some new obstacles occupy the path. Then, from the current
cell (5,7), the re-planning process based on the updated environment begins. The updated temperature gradient as shown in Figure 13c indicates only one moving channel from the current cell to the goal position. The new optimized path marked green in Figure 13a, demonstrates that the re-planning algorithm is successful.

8) COMPARATIVE STUDY
This section provides a performance comparison between the proposed method and the Genetic Algorithm. Additional path planning simulations using GA are carried out for the first four benchmark virtual environments. The simulation on each environment with a population size of 100 is executed 50 times. The cost values of the optimized paths from the Genetic Algorithm and the proposed method and their runtimes are listed in Table 7. It can be seen that the proposed method outperforms the genetic algorithm in all the categories and all the environments. Furthermore, the proposed method performs much faster than the GA.

B. DISCUSSION
It can be remarked for all the seven virtual environments that the thermal gradients (see Figures 7b, 8b, 9b, 10b, 11b, 12b, 13b, 13c) indicate clearly feasible paths for the robot moving from the start locations to the goal locations. In other words, by following any gradient ascent path from the start location, the robot must end up at the goal location without getting trapped at an undesired destination. Because the governing thermal equation (5) is a harmonic function that satisfies a minimum/maximum principle, it attains its
TABLE 7. Comparative study of optimized path results from the proposed method and genetic algorithm.

| Environment | Method            | Time consumption | Smoothness | Safety | Run time (s) |
|-------------|-------------------|------------------|------------|--------|--------------|
|             | Proposed method   | 0.894            | 0.941      | 0.945  | 0.93         |
|             | Genetic algorithm | 0.886            | 0.722      | 0.940  | 41.18        |
| H-shape     | Proposed method   | 0.869            | 0.875      | 0.926  | 1.03         |
|             | Genetic algorithm | 0.861            | 0.750      | 0.920  | 154.23       |
| Spiral      | Proposed method   | 0.667            | 0.920      | 0.862  | 0.99         |
|             | Genetic algorithm | 0.643            | 0.852      | 0.860  | 34.92        |
| 3-slit      | Proposed method   | 0.832            | 0.777      | 0.773  | 1.10         |
|             | Genetic algorithm | 0.832            | 0.733      | 0.773  | 100.30       |

minimum/maximum only on the boundary and not in the interior domain [36]. The thermal gradients in H-shape and Maze environments, see Figures 8b and 12b, show the advantage of this conduction-based approach over the artificial potential field method in avoiding “U” trap sites caused by local minima. Furthermore, Figure 12b also indicates that the proposed method can overcome the goal non-reachable problem when the goal is very close to an obstacle [59].

In the first three environments with both start and goal positions in O-shaped morphology, the algorithm prioritizes commanding sequences of translation motions without any shapeshifts or rotations. However, in the 3-split, Zigzag, and Maze environments with some narrow spaces, shapeshift and rotation commands are required to transform between O- and I-shaped morphologies during the navigation. Although there are seven shapes available for hTetro, O- and I-shapes are the most used. This finding is in good agreement with a previous study [5]. It is also suggested from practices that the best navigation scheme for hTetro is to use the O-shape to travel in open areas, which maximizes the path safety value, whereas it uses I-shape to travel through narrow channels that are inaccessible by O-shape. The other five morphologies are less competitive than O- and I-shapes in maximizing the path safety value and the capability to arrive at the destination. This implies that the PP simulations can be used to analyze, design, and optimize shaped morphologies of reconfigurable robots.

The optimized paths based on best time consumption tend to be close to the obstacles to shorten travel distances, while the optimized routes based on best safety tend to be longer and keep some distance from the obstacles. Also, the optimized paths based on the best smoothness often take a longer travel time than the best time consumption paths, as they emphasize a high consistency in the translation motion commands. Since the PP is always guided by the thermal gradients, optimized paths for different cost functions can also be completely overlapping with each other.

It should be noted that the definitions of the cost functions influence path solutions. Thus, the search algorithm may yield different path solutions if the cost functions are defined differently. Furthermore, it can also be noted that the optimized path based on multi-objectives is a trade-off solution path, which is influenced not only by the three customized cost criteria but also by the choice of their weighting factors.

The running time for all the case studies of $24 \times 24$ cells is within a second on a personal computer and the method can handle environments decomposed by grids with millions of cells. In general, the computational complexity of the method is the same as that of the linear finite element procedure for the heat transfer problem, which is more computationally expensive than the potential field approach, e.g. [37]–[39].

C. PATH PLANNING OF hTetro IN A REAL-WORLD ENVIRONMENT

We evaluate the proposed path planning method in comparison to the conventional A* and Dijkstra global path planning techniques in a complex real-world testbed environment in terms of run time and energy consumption of hTetro. Hector SLAM [60] is used to map an indoor room with a complex obstacle layout as presented in Figure 14. After having obtained the environment discretization, the temperature field is generated from the finite element analysis with $128 \times 128$ elements, as shown in Figure 14b. Then, the predefined global plans of each method with locations and the desired shapes of the robot are generated with the ROS move_base package for navigation as shown in Figure 14a. The trajectory for our proposed PP method with reconfiguration ability of hTetro within the temperature gradient ensures the feasible shapes...
to navigate through the narrow passage. The actual robot transformation actions while navigating through the tight spaces are shown in Figure 15.

The energy consumption and travel time results averaged over 5 trials for each method are reported in Table 8. Specifically, with only the option of keeping the O-shape, the A* method consumes the most energy since the robot needs to travel a longer path to navigate from the source to destination, and close behind is the Dijkstra with the advance of diagonal navigation directions. As a result, without reconfiguration abilities, the proposed method with multi-objective function consumes less time and energy than the A* method by 18.03% and 15.69%.

### VII. CONCLUSION

This paper presents a novel path planning method for reconfigurable robots, combining heat conduction analysis and the grid-based optimization technique. The proposed method combines the strengths of both techniques: On one hand, the thermal conduction analysis can rapidly identify feasible moving paths, and the temperature gradient guarantees that the navigation overcomes the deadlock problem. On the other hand, the grid-based optimization technique handles the size and morphologies of the reconfigurable robot. The proposed method has been successfully demonstrated for robot path planning in various virtual environments and a real-world testbed, where different settings required the reconfigurability of the robot.

The method has shown a strong capability of determining optimal paths based on different criteria, i.e., time consumption, safety, smoothness, and their combined multi-objectives. Therefore, this feature can be used as an optimization tool in shaped morphologies design for reconfigurable robots. With the re-planning algorithm, the method has also been found suitable for path planning in dynamic environments. The proposed path planner is more stable, faster, and provides better path solutions than established methods such as the A*, Dijkstra, or Genetic Algorithm methods.

Future research directions can be as follows: (1) **Application on real terrains:** The proposed method can be extended to deploy path planning on real terrains. The heat conductivity values of finite elements of the workspace should be assigned to reflect the well movability levels of the robot in a real terrain environment. (2) **Moving obstacles:** The proposed path planner is suitable for unknown dynamic environments with moving obstacles in the workspace. In such a case, the online path planning procedure as presented in the last simulation example will need to be reactivated over a given time interval. (3) **Multi-robot path planning:** The proposed method would be extended for multi-robot path planning problems. This can be implemented by utilizing the fact that robots always move along the temperature gradient and avoid heat sinks. If a dynamic heat sink is attached to each robot, possible collisions between robots could be avoided. (4) **Application to other robot platforms:** As the proposed method is a generic path planning framework, it is straightforward to apply it to other reconfigurable robot platforms with different footprint sizes such as pavement sweeping, staircase cleaning, or vertical climbing robots.

### REFERENCES

[1] A. V. Le, P. T. Kyaw, P. Veerajagdheswar, M. A. V. J. Muthuga, M. R. Elara, M. Kumar, and N. H. K. Nhan, “Reinforcement learning-based optimal complete water-blasting for autonomous ship hull corrosion cleaning system,” Ocean Eng., vol. 220, Jan. 2021, Art. no. 108477.

[2] A. V. Le, P. Veerajagdheswar, P. T. Kyaw, M. A. V. J. Muthuga, M. R. Elara, M. Kuma, and N. H. K. Nhan, “Towards optimal hydro-blasting in reconfigurable climbing system for corroded ship hull cleaning and maintenance,” Expert Syst. Appl., vol. 170, May 2021, Art. no. 114519.

[3] M. B. Alatise and G. P. Hancke, “A review on challenges of autonomous mobile robot and sensor fusion methods,” IEEE Access, vol. 8, pp. 39830–39846, 2020.

[4] Y. D. V. Yasuda, L. E. G. Martins, and F. A. M. Cappabianco, “Autonomous visual navigation for mobile robots: A systematic literature review,” ACM Comput. Surv., vol. 53, no. 1, pp. 1–34, 2020.

[5] K. P. Cheng, R. E. Mohan, N. H. Khanh Nhan, and A. V. Le, “Multi-objective genetic algorithm-based autonomous path planning for hinged-tetro reconfigurable tiling robot,” IEEE Access, vol. 8, pp. 121267–121284, 2020.

[6] P. Veerajagdheswar, K. Ping-Cheng, M. R. Elara, A. V. Le, and M. Iwase, “Motion planner for a tetris-inspired reconfigurable floor cleaning robot,” Int. J. Adv. Robot. Syst., vol. 17, no. 2, 2020, Art. no. 1729881420914441.

[7] V. Prabakan, M. R. Elara, T. Pathmakumar, and S. Nansai, “hTetro: A Tetris inspired shape shifting floor cleaning robot.” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2017, pp. 6105–6112.

[8] A. V. Le, P.-C. Ku, T. T. Tun, N. H. K. Nhan, Y. Shi, and R. E. Mohan, “Realization energy optimization of complete path planning in differential drive based self-reconfigurable floor cleaning robot,” Energies, vol. 12, no. 6, p. 1136, 2019.

[9] E. W. Dijkstra, “A note on two problems in connexion with graphs,” Numer. Math., vol. 1, no. 1, pp. 269–271, 1959.

[10] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” IEEE Trans. Syst. Sci. Cybern., vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.
[11] A. Stenz, “Optimal and efficient path planning for partially-known environments,” in Proc. IEEE Int. Conf. Robot. Autom., May 1993, pp. 3310–3317.

[12] S. Koenig and M. Likhachev, “Past replanning for navigation in unknown terrain,” IEEE Trans. Robot., vol. 21, no. 3, pp. 354–363, Jun. 2005.

[13] M. Nazarahari, Enfazi, no. 5, pp. 77–84, Apr. 2018.

[14] M. A. P. Garcia, O. Montiel, R. Castilho, R. Sepulveda, and P. Melin, “Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation,” Appl. Soft Comput., vol. 9, no. 3, pp. 1102–1110, 2009.

[15] M.-R. Zeng, L. Xi, and A.-M. Xiao, “The free step length ant colony optimization algorithm in mobile robot path planning,” Adv. Robot., vol. 30, no. 23, pp. 1509–1514, Dec. 2016.

[16] S. Thabit and A. Mohades, “Multi-robot path planning based on multi-objective particle swarm optimization,” IEEE Access, vol. 7, pp. 2138–2147, 2018.

[17] P. K. Das, H. S. Behera, and B. K. Panigrahi, “A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning,” Swarm Evol. Comput., vol. 26, pp. 14–28, Jun. 2016.

[18] P. Moscati and M. G. Norman, “A memetic approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems,” Parallel Comput., vol. 25, no. 3, pp. 176–187, 1999.

[19] M. M. Eusuff and K. E. Lansey, “Optimization of water distribution network design using the shuffled frog leaping algorithm,” J. Water Resources Planning Management, vol. 129, no. 3, pp. 210–225, 2003.

[20] V. Kroumov, J. Yu, and K. Shihyama, “3D path planning for mobile robots using simulated annealing neural network,” Int. J. Innov. Comput. Inf. Syst., vol. 6, no. 7, pp. 2885–2899, 2010.

[21] A. V. Le, R. Parween, P. T. Kayaw, R. E. Moohan, T. H. Q. Minh, and C. S. S. Borusu, “Reinforcement learning-based energy-aware area coverage for reconfigurable hRombo tiling robot,” IEEE Access, vol. 8, pp. 209750–209761, 2020.

[22] P. T. Kayaw, A. Paing, T. T. Thu, R. E. Moohan, A. V. Le, and P. Veerajagadheswar, “Coverage path planning for decomposition reconfigurable grid-maps using deep reinforcement learning based travelling salesman problem,” IEEE Access, vol. 8, pp. 225945–225956, 2020.

[23] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” Int. J. Robot. Res., vol. 30, no. 7, pp. 846–894, 2011.

[24] A. H. Qureshi and Y. Y. Kim, “A hybridization of independent algorithm for optimal path planning and efficient path planning in complex cluttered environments,” Robot. Auton. Syst., vol. 68, pp. 1–11, Jun. 2015.

[25] J.-C. Latombe, “Rapidly-exploring random trees: A new tool for path planning,” IEEE Comput. Graph. Appl., vol. 15, no. 6, pp. 66–77, Nov. 1995.

[26] A. H. Qureshi and Y. Y. Kim, “A hybrid mobile robot path planning algorithm by equivalent conduction heat flow topology optimization,” Struct. Multidiscip. Optim., vol. 45, no. 5, pp. 703–715, May 2012.

[27] B. Li, H. Liu, and W. Su, “Topology optimization techniques for mobile robot path planning,” Appl. Soft Comput., vol. 78, pp. 528–544, May 2019.

[28] A. V. Le, A. A. Hayat, M. R. Elara, N. H. K. Nhan, and K. Prathap, “Reconfigurable pavement sweeping robot and pedestrianhabitant framework by vision techniques,” IEEE Access, vol. 7, pp. 159402–159419, 2019.

[29] L. Yi, A. V. Le, B. Ramalingam, A. A. Hayat, M. R. Elara, T. H. Q. Minh, B. F. Gomez, and L. K. Wen, “Locomotion with pedestrian aware from perception sensor by pavement sweeping reconfigurable sensor,” Sensors, vol. 21, no. 5, p. 1745, Mar. 2021.

[30] L. Yi, A. V. Le, A. A. Hayat, C. S. S. Borusu, R. E. Moohan, N. H. K. Nhan, and P. Kandasamy, “Reconfiguration during locomotion by pavement sweeping robot with feedback control from vision system,” IEEE Access, vol. 8, pp. 113355–113370, 2020.

[31] A. V. Le, N. H. K. Nhan, and R. E. Moohan, “Evolutionary algorithm-based complete coverage path planning for tetriamond tiling sensors,” Sensors, vol. 20, no. 2, p. 445, Jan. 2020.

[32] K. G. S. Apuroop, A. V. Le, M. R. Elara, and B. J. Sheu, “Reinforcement learning-based complete area coverage path planning for a modified hTrihex robot,” Sensors, vol. 21, no. 4, p. 1067, Feb. 2021.

[33] Y. Yoshida, S. Mori, T. Kaminuma, K. Tomita, H. Kurokawa, and S. Kokaji, “A motion planning method for a self-reconfigurable modular robot,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Expanding Soc. Role Robot. Next Millennium, Oct./Nov. 2002, pp. 590–597.

[34] E. Guan, Z. Fu, W. Yan, D. Jiang, and Y. Zhao, “Self-reconfiguration path planning design for M-lattice robot based on genetic algorithm,” in Intelligent Robotics and Applications (Lecture Notes in Computer Science), Berlin, Germany: Springer, 2011, pp. 505–514.

[35] L. Protzer, M. Staehler, A. Hermann, A. Roennau, and R. Dillmann, “KAIRO 3: Moving over stairs & unknown obstacles with reconfigurable snake-like robots,” in Proc. Euc. Conf. Mobile Robots (ECMR), Sep. 2015, pp. 1–6.

[36] A. Le, M. Arunmozhi, P. Veerajagadheswar, P.-C. Ku, T. H. Minh, V. Sivananthan, and R. Mohan, “Complete path planning for a tetrinspired self-reconfigurable robot by the genetic algorithm of the traveling salesman problem,” Electronics, vol. 7, no. 12, p. 344, 2018.

[37] K. P. Cheng, R. E. Moohan, N. H. K. Nhan, and A. V. Le, “Graph theory-based approach to accomplish complete coverage path planning tasks for reconfigurable robots,” IEEE Access, vol. 7, pp. 94642–94657, 2019.

[38] A. Le, V. Prabakan, V. Sivananthan, and R. Mohan, “Modified a-star algorithm for efficient coverage path planning in tetrinspired self-reconfigurable robot with integrated laser sensor,” Sensors, vol. 18, no. 8, p. 2585, 2018.

[39] A. Krishna Lakshmanan, R. Elara Mohan, B. Ramalingam, A. V. Le, P. Veerajagadeshwar, K. Tiwari, and M. Ilyas, “Complete coverage path planning using reinforcement learning for tetrinos based cleaning and maintenance robot,” Autom. Constr., vol. 112, Apr. 2020, Art. no. 103078.
RAJESH ELARA MOHAN received the B.E. degree from Bharathiar University, India, in 2003, and the M.Sc. and Ph.D. degrees from Nanyang Technological University, in 2005 and 2012, respectively. He is currently an Assistant Professor with the Engineering Product Development Pillar, Singapore University of Technology and Design. He is also a Visiting Faculty Member with the Optoelectronics Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam. His current research interests include robotics vision, robot navigation, human detection, action recognition, feature matching, and 3D video processing.

OLIVER WEEMBERG received the Diploma (master’s) degree in techno-mathematics from TU Munich, in 2011, and the Ph.D. degree in mathematics from TU Kaiserslautern, in 2015. He is currently an Assistant Professor of cyber-physical simulation with the Department of Mechanical Engineering, TU Darmstadt, Germany. His research interests include computational mechanics and engineering, in particular in the development of advanced discretization methods, multiscale modeling and simulation, design optimization, and additive manufacturing applications.

RAJESH ELARA MOHAN received the B.E. degree from Bharathiar University, India, in 2003, and the M.Sc. and Ph.D. degrees from Nanyang Technological University, in 2005 and 2012, respectively. He is currently an Assistant Professor with the Engineering Product Development Pillar, Singapore University of Technology and Design. He is also a Visiting Faculty Member with the Optoelectronics Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam. His current research interests include robotics vision, robot navigation, human detection, action recognition, feature matching, and 3D video processing.

YUJIE GUO received the M.S. degree in mechanical engineering from Southeast University, China, in 2011, and the Ph.D. degree from the Faculty of Aerospace Engineering, Delft University of Technology, The Netherlands, in 2015. He is currently working as an Associate Professor with the College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, China. His current research interests include computational mechanics, structural optimization, composite structures, and advanced numerical methods for multi-physics problems.

OLIVER WEEMBERG received the Diploma (master’s) degree in techno-mathematics from TU Munich, in 2011, and the Ph.D. degree in mathematics from TU Kaiserslautern, in 2015. He is currently an Assistant Professor of cyber-physical simulation with the Department of Mechanical Engineering, TU Darmstadt, Germany. His research interests include computational mechanics and engineering, in particular in the development of advanced discretization methods, multiscale modeling and simulation, design optimization, and additive manufacturing applications.

HUY DO received the B.E. degree in civil engineering from Hanoi Architectural University, Vietnam, in 2001, the M.Sc. degree from the University of Stuttgart, Germany, in 2009, and the Ph.D. degree from the Ecole Polytechnique Federale de Lausanne, Switzerland, in 2013. He is currently working as a Postdoctoral Research Fellow with the ROAR Laboratory, Singapore University of Technology and Design. His current research interests include robot autonomy, optimization techniques, and numerical analysis.

ANH VU LE received the B.S. degree in electronics and telecommunications from Hanoi University of Technology, Vietnam, in 2007, and the Ph.D. degree in electronics and electrical from Dongguk University, South Korea, in 2015. He is currently working as a Postdoctoral Research Fellow with the ROAR Laboratory, Singapore University of Technology and Design. He is also with the Optoelectronics Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam. His current research interests include robotics vision, robot navigation, human detection, action recognition, feature matching, and 3D video processing.

MINH BUI VU was born in Dong Nai, Vietnam, in March 1991. He graduated in electrical and electronic engineering from Nguyen Tat Thanh University, Ho Chi Minh City, Vietnam, in 2014. He received the master’s degree in electrical engineering from Ho Chi Minh City University of Technology and Education, Ho Chi Minh City, in 2019. He joined the Faculty of Mechanical, Electrical, Electronic and Automotive Engineering, Nguyen Tat Thanh University, as a Laboratory-Practice Management, in 2014, where he was a Lecturer, in 2017. His research interests include artificial neural networks and wireless networks.

NGUYEN NHAT TAN received the B.S. and M.S. degrees in electronics and telecommunications engineering from Ho Chi Minh University of Natural Sciences, a member of Vietnam National University, Ho Chi Minh City, Vietnam, in 2008 and 2012, respectively, and the Ph.D. degree in computer science, communication technology and applied mathematics from the VSB Technical University of Ostrava, Czech Republic, where he is currently pursuing the Ph.D. degree in electrical engineering. He joined the Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Vietnam, in 2013, where he has been working as a Lecturer since then. His research interests include robotics vision and image processing.

PHAN VAN DUC was born in Long An, Vietnam, in 1975. He received the M.S. degree from the Department of Electric, Electrical and Telecommunications Engineering, Ho Chi Minh City University of Transport, Vietnam, and the Ph.D. degree from the Department of Mechanical and Automation Engineering, Da-Yeh University, Taiwan, in 2012. His current research interests include sliding mode controls, non-linear systems and active magnetic bearings, flywheel energy storage systems, power system optimization, optimization algorithms, renewable energies, and energy harvesting (EH) enabled cooperative networks.