Mobile Robot Path Planning using a Teaching-Learning-Interactive Learning-Based Optimization

Yu-Huei Cheng, *Member, IAENG*, Pei-Ju Chao, and Che-Nan Kuo

***Abstract*—In many automated industrial environments, mobile robots have been widely used for performing exclusive tasks. Collision-free path planning is one of the most basic requirements for the application of mobile robots. In order to find a collision-free path in a known static environment for a mobile robot, a Teaching-Learning-Interactive Learning-Based Optimization (TLILBO) is proposed. The proposed method is a novel stochastic search algorithm modelled based on the process of natural selection. The proposed method is designed based on the three concepts of "teaching", "learning", and "interactive learning" to effectively search for a feasible and collision-free path. Two obstacle environmental maps retrieved from the literature were verified in this study. Simulation results showed that the proposed method was effective for path planning.**

***Index Terms*—mobile robot, collision-free, path planning, Teaching-Learning-Interactive Learning -Based Optimization (TLILBO)**

I. INTRODUCTION

MOBILE robot path planning technology is an important branch of intelligent mobile robot research. In the past, many methods have been developed to solve mobile robot path planning, such as the C-space method [1], cell decomposition [2], roadmap [3], and potential field method [4]. However, most of these methods are based on the concept of spatial configuration. In addition, these technologies show a lack of adaptability and unhealthy behavior.

Path planning is the task of finding a feasible path from the beginning to the goal in a workspace according to some optimization criteria, such as lowest cost, shortest time, and shortest length. According to the robot's understanding of the environment, path planning can be divided into global path planning, in which the environmental information is completely known, and local path planning, in which the environmental information is completely unknown or partially unknown.

There are many algorithms available for various applications, such as the neural network method [5, 6], the GA method [7, 8], PSO method [9, 10], and ant colony optimization [11, 12]. Currently, path planning methods generally include neural network method [13], Differential Evolution (DE) [14], genetic algorithm (GA) [15], and the ant colony algorithm [11], artificial bee colony algorithm (ABC) [16], and particle swarm optimization (PSO) [17]. These methods each have advantages, but there are also some deficiencies, such as poor adaptability to the path diagram, high computational complexity, long search time, low convergence accuracy, and easy to fall into local optimization. Therefore, these methods may limit the ability of mobile robot path planning.

In recent years, Rao *et al.* proposed a Teaching-Learning-Based Optimization (TLBO) method [12, 13]. This algorithm has the advantages of high convergence speed and high precision, and is very suitable for solving path optimization problems. Therefore, this algorithm can provide a new solution for the global path planning of mobile robots. The TLBO is an algorithm with no specific parameters [14]. It only requires common control parameters such as the size of the population and the number of generations, without the burden of adjusting the control parameters. This makes the TLBO algorithm simpler, more efficient, and has a relatively low computational cost. Therefore, TLBO has been successfully applied in various optimization fields such as production job shop scheduling [18], heat pipe optimization design [19], automatic voltage regulation [20], and primer design of biotechnology [21]. Recently, various TLBO variants have been proposed in the literature to improve the performance of the TLBO. Rao *et al.* proposed the ETLBO algorithm [22] to solve the optimization of complex constraints. In addition, there have also been proposals for improved TLBO algorithms to solve global function optimization problems [23, 24] and multi-objective optimization problems [25, 26]. In this study, we propose a Teaching-Learning-Interactive Learning-Based Optimization (TLILBO) to solve the optimization problem of the global path planning of mobile robots.

Manuscript received August 18, 2018; revised December 4, 2018; revised February 16, 2019. This work was supported in part by the Ministry of Science and Technology (MOST) in Taiwan under grant MOST107-2622-E-324-002-CC3, MOST107-2221-E-324-020, MOST107-2821-C-324-001-ES, MOST107-2218-E-005-023, and the Chaoyang University of Technology (CYUT) and Higher Education Sprout Project, Ministry of Education, Taiwan, under the project name: "The R&D and the cultivation of talent for Health-Enhancement Products."

Y.-H. Cheng is with the Department of Information and Communication Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan (e-mail: yuhuei.cheng@gmail.com).

P.-J. Chao is with the International School, Duy Tan University, 254 Nguyen Van Linh, Danang, Vietnam (e-mail: chao.peiju@duytan.edu.vn).

C.-N. Kuo is with Department of Business Administration, CTBC Financial Management College, Tainan 709, Taiwan (corresponding author. e-mail: fkikimo@hotmail.com).

II. Methods

A. Problem definition

The definition of the mobile robot path planning problem for this study is described as follows:

"Given a mobile robot and an environmental description, plan the paths between the two designated locations, one start position and one end position. The planned path cannot have collisions and must meet some optimization criteria."

Based on the above definition, the mobile robot path planning in this study is classified as an optimization problem. In mobile robot path planning, the method for solving the path planning problem can be differentiated according to the following two factors:

(1) Static or dynamic environment type [27]

A static environment is defined as an environment that does not contain any moving objects other than a navigation robot, and a dynamic environment has a dynamic moving object that includes people, moving machines, and moving robots.

(2) Global or local path planning algorithm [28]

Global path planning algorithms require complete knowledge of the search environment and all terrain should be static. On the other hand, local path planning means that path planning is being performed while the robot is moving. In other words, this type of algorithm can produce a new path that responds to environmental changes.

B. Mobile Robot Path Planning

In this study, we proposed a new, improved TLBO algorithm based on the interactive learning mechanism between learners, called the TLILBO algorithm. This algorithm exchanges information gathered by learners with each other to achieve information exchange between learners. This interactive learning mechanism was mainly added after the "learning phase", and interactive learning was achieved through group discussions. The advantage of this approach is that, after the stages of "teaching" and "learning", learners can exchange knowledge with each other through interactive discussion and discussion among group learners to achieve better learning.

The steps of the TLILBO algorithm proposed in this study are described below, and in the flowchart in Fig. 1.

Step 1. Learner coding

Learner coding was first used for the path planning problem of mobile robots. The learner, L, expressed the solution of the path, and each variable, s, in the learner, L, represented the learned subject. Therefore, we defined the learner coding as shown in equation (1).

$$L = \{s_1, s_2, s_3, \dots, s_m\} \tag{1}$$

where L denote the learner, namely the solution of the path, s denoted the learning subject for L and contained the value of the moving direction of each node, and m was the number of learning subjects, and the dimension size of the environmental map.

Step 2. Import environmental map

Next, the environmental map was imported for use in mobile path planning. The imported environmental map was

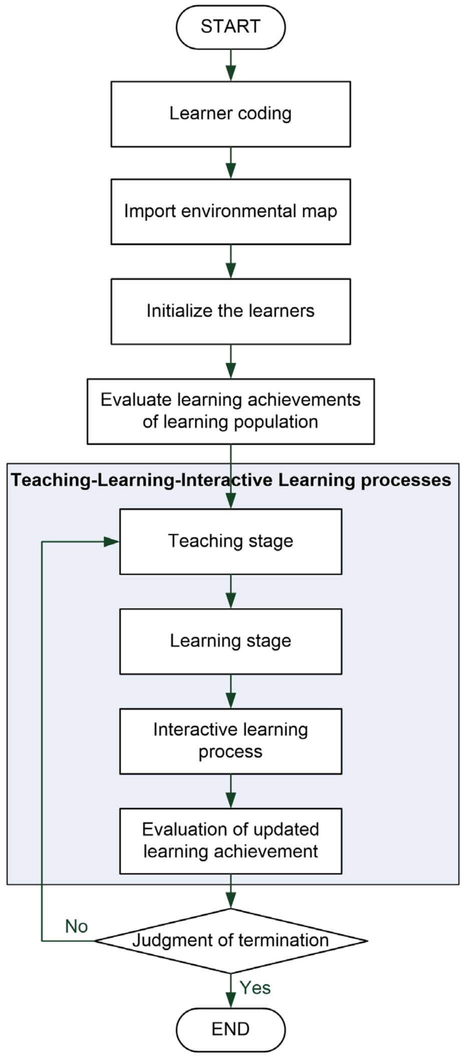

Fig. 1. Flowchart of TLILBO algorithm for mobile robot path planning.

dominated by a 2-dimensional matrix, where each position in the matrix represented a path node. Therefore, when the robot appears in the real environment, the robot should have moved step-by-step on the proposed path nodes.

Step 3. Initialize the learners

At the beginning of the algorithm, a random number of learners, were generated as the initial learning population. Each learner was a solution to the planned path. Therefore, the initial learning population of size n could be expressed as:

$$P = \{L_1, L_2, L_3, \dots, L_n\} \tag{2}$$

where P was a learning population, L was a learner, and n was a learning population size.

Step 4. Evaluate learning achievements of learning population

Each learner was evaluated in turn for their learning achievements. The evaluation of this learning achievement was calculated through the learning achievement function. The preferred planned path for the mobile robots of this study was the shortest path between the start position and the end position. Therefore, the learning achievement function had to be responsible for finding this moving path. The shortest path allowed us to calculate the total number of steps required for the mobile robot to reach the end. Thus, the learning achievement function was designed as shown in equation (3) below.

$$\text{Archievement}(L) = \frac{1}{d_1 + d_2 + \cdots + d_t} \quad (3)$$

where t represented the total number of steps required to move from the start point to the end point. d_1, d_2, …, d_t were the distances between one node and next node, respectively. The calculation was based on the Euclidean distance formula, as shown in following equation (4).

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (4)$$

where A was (x_1, y_1) and B was (x_2, y_2).

Step 5. Teaching stage

Under normal circumstances, the teacher is usually considered to be a person with a high degree of learning ability to train learners so they can have better learning achievement. Therefore, at this stage, we first looked for the best learner from the learning population as the teacher. In accordance with the teacher's abilities, the teacher tried to increase the average learning achievements of learners in the subjects they taught.

At any iteration *i*, we assumed there were "m" number of subjects (i.e. design variables), "n" number of learners (i.e. learning population size, k = 1, 2, ..., n), and $M_{j,i}$ were the learner's average learning achievement for a particular subject "j" (j = 1, 2, ..., m). The best overall learning achievement $X_{total-kbest,i}$ considered the best learning achievement for all subjects, with *kbest* being the best learner. The difference between the existing average learning result and the corresponding learning result for each subject was given as follows:

$$Difference_{mean_{j,k,i}} = r_i \times (X_{j,kbest,i} - T_F M_{j,i}) \quad (5)$$

where $X_{j,kbest,i}$ was the learning result of the best learner (i.e. teacher) in subject j.

T_F was a teaching factor that determined the change in the mean; r_i was a random number in [0,1]. The value of T_F was either 1 or 2 and was randomly determined with the same probability as follows:

$$T_F = round[1 + rand(0,1)\{2 - 1\}] \quad (6)$$

Here, the T_F value was not an input to the algorithm, and its value was determined by equation (6) randomly. It was pointed out in the literature that the TLBO performed well when the T_F value was between 1 and 2 after performing many benchmark function simulation experiments. However, the algorithm also showed that the T_F value of 1 or 2 was more suitable for solving problems based on simulation experiments. Therefore, in order to simplify the algorithm, the teaching factor was suggested as 1 or 2.

Step 6. Update learning achievement

Based on $Difference\_mean_{j,k,i}$, the existing solution was updated at the teaching stage according to the following formula:

$$X'_{j,k,i} = X_{j,k,i} + Difference\_mean_{j,k,i} \quad (7)$$

where $X'_{j,k,i}$ was the updated value of $X_{j,k,i}$. The algorithm would accept $X'_{j,k,i}$ if it provided a better function value. All accepted function values were retained at the end of the teaching period and these values were input into the learner interactive learning phase, which depended on the teaching phase.

Step 7. Learning stage

After a certain amount of iterations, learners could interact with other learners in the learning population at random. If other learners had more knowledge than themselves, the learner will learn new things to improve the learner's own knowledge. We expressed the learning phenomenon at that stage as follows:

Two learners P and Q, were randomly selected so that $X'_{total-P,i} \neq X'_{total-Q,i}$, where $X'_{total-P,i}$ and $X'_{total-Q,i}$ was the updated value of $X_{total-P,i}$ and $X_{total-Q,i}$ at the end of the teacher's period, respectively. The learning updates were shown in equations (8) and (9).

$$X''_{j,P,i} = X'_{j,P,i} + r_i \times (X'_{j,P,i} - X'_{j,Q,i}), \quad if\ X'_{total-P,i} < X'_{total-Q,i} \quad (8)$$

$$X''_{j,P,i} = X'_{j,P,i} + r_i \times (X'_{j,Q,i} - X'_{j,P,i}), \quad if\ X'_{total-P,i} > X'_{total-Q,i} \quad (9)$$

Step 8. Interactive learning process

Because the learning stage was to randomly select learners and interact with learners in the learning population, the learners themselves were not always to learn new things. In this study, an interactive learning process was added. This process was mainly to group learning populations so that learners could exchange knowledge with each other through interactive discussion among learners in the group to achieve better learning. We expressed the learning phenomenon of the interactive learning process as follows:

$$n_{g\_avg} = \lfloor n/g \rfloor \quad (10)$$

where u_{g_avg} was the number of average learners in groups, which required Floor operation, *u* was the learning group size, g was the number of groups.

In the interactive learning process, the learner set in each group, g_x, was expressed as follows:

$$g_x = \{L_1, L_2, \ldots, L_{u_g}\} \quad (11)$$

where u_g was the number of learners in each group.

Two learners, L_c and L_d, were then randomly selected from the g_x group to learn from each other and exchange information by exchanging some of the learning outcomes from each other. The interactive learning update was shown by equations (12) and (13).

$$X'''_{j,L_c,i} = X''_{j,L_d,i}, \quad if\ X''_{total-L_c,i} < X''_{total-L_d,i} \quad (12)$$

$$X'''_{j,L_d,i} = X''_{j,L_c,i}, \quad if\ X''_{total-L_c,i} > X''_{total-L_d,i} \quad (13)$$

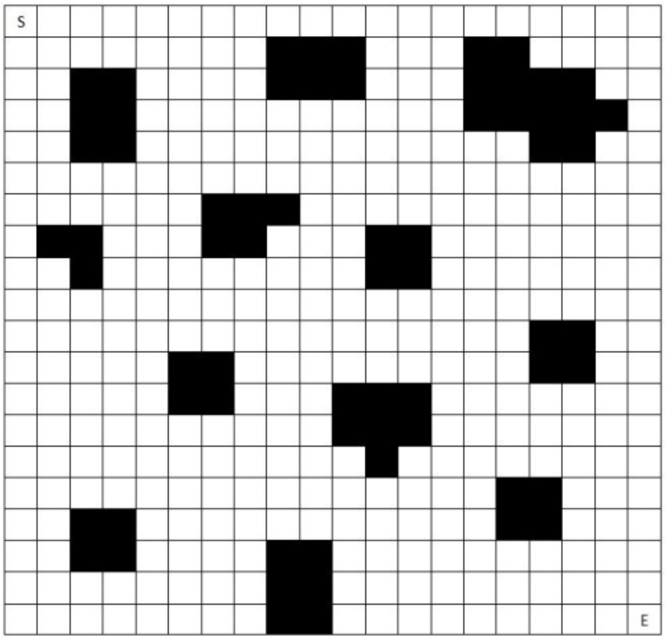

Fig. 2. Map 20×20 retrieved from the literature of Nianyin Zeng *et al.* S represents the start position and the E represents the end position.

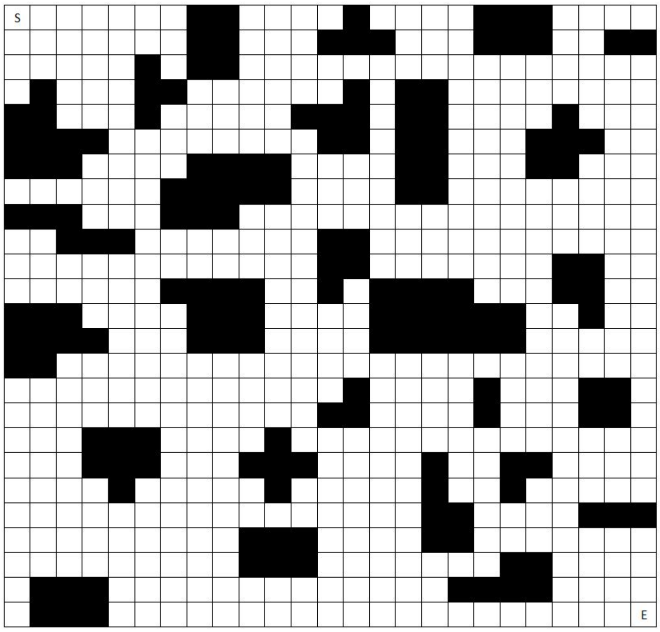

Fig. 3. Map 25×25 retrieved from the literature of Nianyin Zeng *et al.* S represents the start position and the E represents the end position.

Step 9. Evaluation of updated learning achievement

Each updated learner was evaluated through the designed learning achievement function. Each updated learner had a corresponding learning achievement value.

Step 10. Judgment of termination

In the iterative process, the algorithm judged whether the current learner's learning achievement value reached the preset number of iterations, and then stopped the algorithm operation. Otherwise, it went back to step 4 and then continued to steps 5 through 10.

III. Results and Discussion

In order to verify the proposed method, two obstacle environmental maps in the literature were performed and their results were discussed. The followings describe the execution environment and their parameter settings, two used obstacle environmental maps retrieved from the literature, the path planning results on Map 20×20, and the path planning results on Map 25×25.

A. Execution environment and parameter settings

The execution environment for this research experiment was performed on a 32-bit operating system Windows 7 SP1 with 4G memory, and Intel(R) Core(TM) 2 Duo CPU E7500 @ 2.93GHz processor. In the TLILBO algorithm parameter settings, the iteration size of its implementation was set to 500. The learning population was set to 10, 20, 30, 40, and 50, respectively, to observe the behavior of TLILBO for path planning under different learning population sizes.

B. Two used obstacle environmental maps retrieved from the literature

Two maps with obstacles were retrieved from the literature of Nianyin Zeng *et al.* [22], namely the map 20×20 and the map 25×25, as shown in Fig. 2 and Fig. 3. In the map 20×20, there were 12 obstacles, the start position was (0, 0), and the end position was (19, 19). In the other map 25×25, there were 27 obstacles, the start position was (0, 0), and the end position was (24, 24). The mobile robot had to move from the start position to end position. Although we could determine the path by our eyes intuitively, the mobile robot had many paths to choose from during the moving and may not have arrived at the target. The TLILBO algorithm helped the mobile robot determine a feasible path based on the imported maps to move to the target.

C. The path planning results on Map 20×20

First, this study used TLILBO to search feasible paths based on the imported map 20×20. In this experiment, the planned paths of the mobile robot moving from the start point (0, 0) to the end position (19, 19) are discussed. The results are shown in Table I and Fig. 4. The planned paths for different learning population sizes are shown in Fig. 6 (Fig. 6 is shown at the end of the article). From Table I and Fig. 4, we could see that the learning population size influenced the path planning result. The execution time was gradually increased with the increase in learning population size, except

TABLE I
TLILBO EXECUTION RESULTS FOR THE MAP 20×20

Learning population size	Moving steps	Moving distance (unit)	Fitness	Execution time (ms)
10	61	70.527	0.014	16378
20	56	64.284	0.016	18315
30	48	**53.799**	**0.019**	22012
40	**45**	54.527	0.018	28268
50	52	60.284	0.017	25444

※Bold font shows the best result.

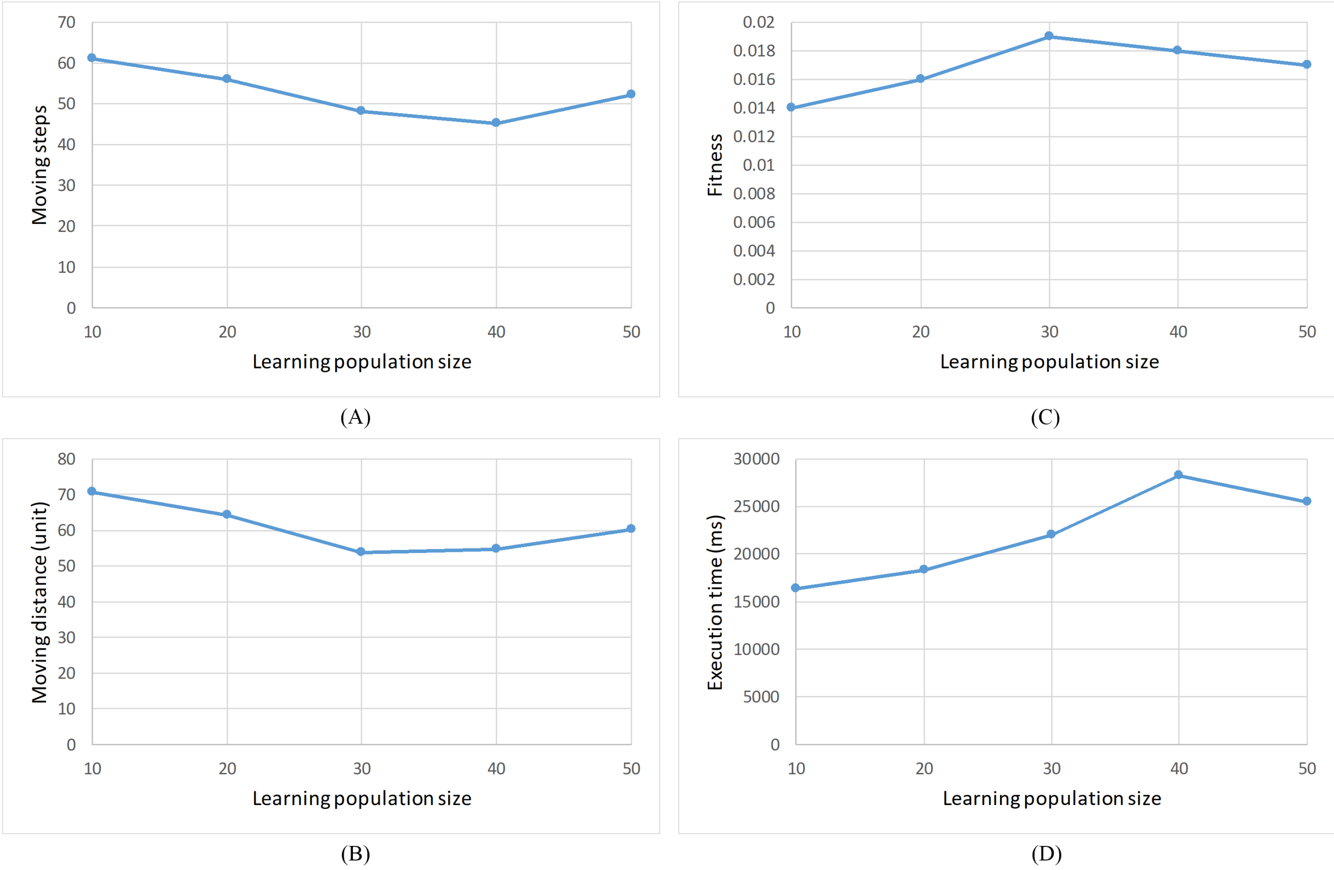

Fig. 4. TLILBO execution results for the map 20×20. (A) moving steps; (B) moving distance (unit); (C) fitness, and (D) execution time (ms).

that the learning population size was set to 50. When the learning population size was set to 30, the best fitness of 0.019 and moving distance of 53.799 were performed. However, the moving steps of 48 were not the best. When the learning population size was set to 40, the best the moving steps of 45, and the secondary moving distance of 54.527, and fitness of 0.018 were performed. We found the smallest learning population size of 10 had the worst results. From Fig. 4, we could see the planned path according to different learning population sizes.

D. The path planning results on Map 25×25

Next, this study used TLILBO to search feasible paths based on the imported map 25×25. In this experiment, the mobile robot moved from the start position (0, 0) to the end position (24, 24). The results are shown in Table II and Fig. 5. The planned paths for different learning population sizes are shown in Fig. 7 (Fig. 7 is shown at the end of the article). From Table II and Fig. 5, we could see that the execution time also gradually increased with the increase in learning population size. When the learning population size was set to 40, the best fitness of 0.015 and moving steps of 58 were performed. However, the moving distance of 336.037 was not the best. When the learning population size was set to 50, the best moving distance of 311.689, and the secondary fitness of 0.014 were performed. On the other hand, when the learning population size was set to 30, we found that it performed the secondary moving steps of 63 and the secondary fitness of 0.014, and it had better execution time than those of the learning population size when it was set to 40 and 50. We also found that a smaller learning population size had worse results, such as when the learning population size was set to 10 and 20. From Fig. 5, we could see the planned path according to different learning population sizes.

IV. Conclusion

In this study, we proposed a teaching-learning-interactive learning-based optimization (TLILBO) method to solve the problem of mobile robot path planning in a static environment. Through simulation experiments in two cases of obstacle environmental maps retrieved from the literature, it was demonstrated that the proposed TLILBO method was feasible for path planning of mobile robots in a static environment. Furthermore, the study also showed that the execution time of TLILBO in obstacle environmental maps was gradually

TABLE II
TLILBO EXECUTION RESULTS FOR THE MAP 25×25

Learning population size	Moving steps	Moving distance (unit)	Fitness	Execution time (ms)
10	68	360.304	0.013	30763
20	69	354.693	0.013	43696
30	63	357.089	0.014	53992
40	**58**	336.037	**0.015**	74958
50	65	**311.689**	0.014	113450

※Bold font shows the best result.

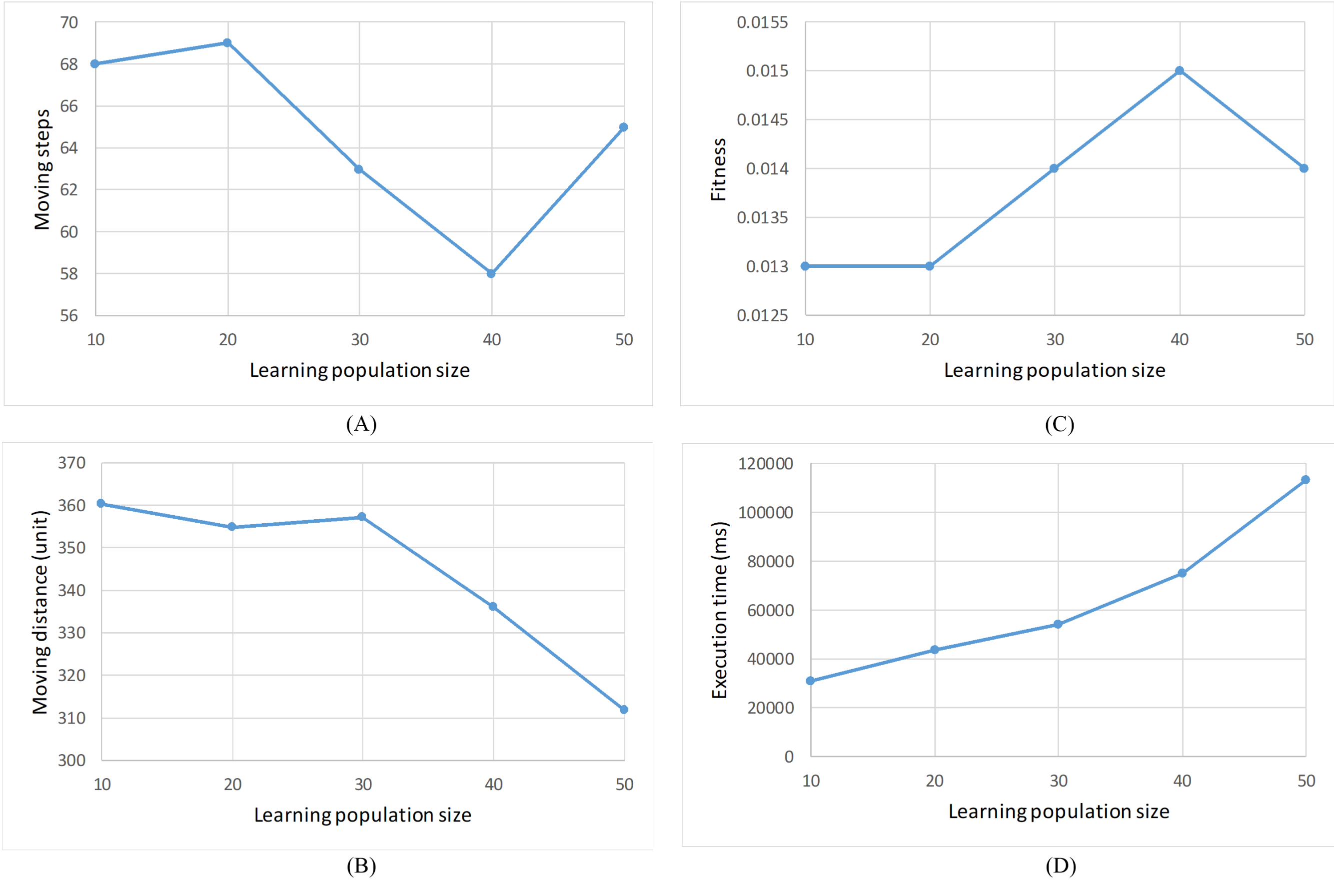

Fig. 5. TLILBO execution results for the map 25×25. (A) moving steps; (B) moving distance (unit); (C) fitness, and (D) execution time (ms).

increased with the increase in learning population size. In the map 20×20, when the learning population size was set to 30 and 40, there were better moving steps, moving distances, and fitness values. In the map 25×25, when the learning population size was set to 30, 40 and 50, there were better moving steps, moving distances, and fitness values. In the future, we will deeply discuss the issue of more leaning population sizes and compare the efficiency of TLILBO with other methods.

Acknowledgment

This work was supported in part by the Ministry of Science and Technology (MOST) in Taiwan under grant MOST107-2622-E-324-002-CC3, MOST107-2221-E-324-020, MOST107-2821-C-324-001-ES, MOST107-2218-E-005-023, and the Chaoyang University of Technology (CYUT) and Higher Education Sprout Project, Ministry of Education, Taiwan, under the project name: "The R&D and the cultivation of talent for Health-Enhancement Products."

References

[1] T. Lozano-Perez, "Spatial planning: A configuration space approach," *IEEE transactions on computers,* no. 2, pp. 108-120, 1983.
[2] F. Lingelbach, "Path planning using probabilistic cell decomposition," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, 2004, vol. 1, pp. 467-472: IEEE.
[3] P. Bhattacharya and M. L. Gavrilova, "Roadmap-based path planning-using the voronoi diagram for a clearance-based shortest path," *IEEE Robotics & Automation Magazine,* vol. 15, no. 2, 2008.
[4] J. Barraquand, B. Langlois, and J.-C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE transactions on systems, man, and cybernetics,* vol. 22, no. 2, pp. 224-241, 1992.
[5] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Transactions on Image Processing,* vol. 26, no. 9, pp. 4509-4522, 2017.
[6] D. Mishra and P. K. Kalra, "Modified Hopfield Neural Network Approach for Solving Nonlinear Algebraic Equations," *Engineering Letters,* vol. 14, no. 1, pp. 135-142, 2007.
[7] P. Olranthichachat and S. Kaitwanidvilai, "GA based Fixed Structure H∞ Loop Shaping Controller for a Buck-Boost Converter," *Engineering Letters,* vol. 16, no. 3, pp. 346-352, 2008.
[8] M. N. H. Siddique and M. O. Tokhi, "GA-based Neural Fuzzy Control of Flexible-link Manipulators," *Engineering Letters,* vol. 13, no. 3, pp. 148-157, 2006.
[9] K. Y. Chan, G. Pong, and K. Chan, "Investigation of Hybrid Particle Swarm Optimization Methods for Solving Transient-Stability Constrained Optimal Power Flow Problems," *Engineering Letters,* vol. 16, no. 1, pp. 61-67, 2008.
[10] C.-H. Yang, Y.-H. Cheng, and L.-Y. Chuang, "PCR-CTPP design based on Particle Swarm Optimization with Fuzzy Adaptive Strategy," *Engineering Letters,* vol. 20, no. 2, pp. 196-202, 2012.
[11] M. Brand, M. Masuda, N. Wehner, and X.-H. Yu, "Ant colony optimization algorithm for robot path planning," in *Computer Design and Applications (ICCDA), 2010 International Conference on*, 2010, vol. 3, pp. V3-436-V3-440: IEEE.
[12] M. Dorigo and T. Stützle, "Ant colony optimization: overview and recent advances," in *Handbook of metaheuristics*: Springer, 2019, pp. 311-351.
[13] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* vol. 34, no. 1, pp. 718-724, 2004.
[14] J. Chakraborty, A. Konar, L. C. Jain, and U. K. Chakraborty, "Cooperative multi-robot path planning using differential evolution," *Journal of Intelligent & Fuzzy Systems,* vol. 20, no. 1, 2, pp. 13-27, 2009.
[15] Y. Hu and S. X. Yang, "A knowledge based genetic algorithm for path planning of a mobile robot," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, 2004, vol. 5, pp. 4350-4355: IEEE.
[16] P. Bhattacharjee, P. Rakshit, I. Goswami, A. Konar, and A. K. Nagar, "Multi-robot path-planning using artificial bee colony optimization algorithm," in *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on*, 2011, pp. 219-224: IEEE.
[17] Y.-Q. Qin, D.-B. Sun, N. Li, and Y.-G. Cen, "Path planning for mobile robot using the particle swarm optimization with mutation

operator," in *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, 2004, vol. 4, pp. 2473-2478: IEEE.
[18] H. Keesari and R. Rao, "Optimization of job shop scheduling problems using teaching-learning-based optimization algorithm," *Opsearch,* vol. 51, no. 4, pp. 545-561, 2014.
[19] R. Rao and K. More, "Optimal design of the heat pipe using TLBO (teaching–learning-based optimization) algorithm," *Energy,* vol. 80, pp. 535-544, 2015.
[20] S. Priyambada, P. K. Mohanty, and B. K. Sahu, "Automatic voltage regulator using TLBO algorithm optimized PID controller," in *Industrial and Information Systems (ICIIS), 2014 9th International Conference on*, 2014, pp. 1-6: IEEE.
[21] Y.-H. Cheng, "Estimation of teaching-learning-based optimization primer design using regression analysis for different melting temperature calculations," *IEEE transactions on nanobioscience,* vol. 14, no. 1, pp. 3-12, 2015.
[22] R. Rao and V. Patel, "An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems," *International Journal of Industrial Engineering Computations,* vol. 3, no. 4, pp. 535-560, 2012.
[23] R. V. Rao and V. Patel, "An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems," *Scientia Iranica,* vol. 20, no. 3, pp. 710-720, 2013.
[24] S. C. Satapathy, A. Naik, and K. Parvathi, "Weighted teaching-learning-based optimization for global function optimization," *Applied Mathematics,* vol. 4, no. 03, p. 429, 2013.
[25] R. Rao and V. Patel, "A multi-objective improved teaching-learning based optimization algorithm for unconstrained and constrained optimization problems," *International Journal of Industrial Engineering Computations,* vol. 5, no. 1, pp. 1-22, 2014.
[26] S. Sultana and P. K. Roy, "Multi-objective quasi-oppositional teaching learning based optimization for optimal location of distributed generator in radial distribution systems," *International Journal of Electrical Power & Energy Systems,* vol. 63, pp. 534-545, 2014.
[27] A. Ramirez-Serrano, H. Liu, and G. C. Pettinaro, "Mobile robot localization in quasi-dynamic environments," *Industrial Robot: An International Journal,* vol. 35, no. 3, pp. 246-258, 2008.
[28] K. H. Sedighi, K. Ashenayi, T. W. Manikas, R. L. Wainwright, and H.-M. Tai, "Autonomous local path planning for a mobile robot using a genetic algorithm," in *Evolutionary Computation, 2004. CEC2004. Congress on*, 2004, vol. 2, pp. 1338-1345: IEEE.

Yu-Huei Cheng (M'12) received the M.S. degree and Ph.D. degree from the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Taiwan, in 2006 and 2010, respectively. He crosses many professional fields including biological and medical engineering, electrical and electronic engineering, and information engineering. He is currently an associate professor of Department of Information and Communication Engineering, Chaoyang University of Technology, Taichung, Taiwan. He has authored and co-authored more than 100 technical papers. His research interests include artificial intelligence, automatic control, bioinformatics, biomedical engineering, computational intelligence, embedded systems, electric and hybrid vehicles, internet of things, machine learning, mobile medical, power electronics, renewable energy and robotics. In terms of academic activities, he is currently the associate editor of IEEE Access. He is also a Senior Member (SM) of the Institute of Electrical and Electronics Engineers (IEEE) and the Universal Association of Computer and Electronics Engineers (UACEE), as well as a member of the IAENG (International Association of Engineers) and the Taiwanese Association for Artificial Intelligence (TAAI).

Pei-Ju Chao is currently an assistant professor of International School at Duy Tan University, Vietnam. Her research interests include supply chain management, organizational behavior, small and medium-sized enterprises, and e-commerce.

Che-Nan Kuo was born on December 1979 in Tainan, Taiwan. He received his B.S. degree in the Department of Computer Science from the Tunghai University, Taichung, Taiwan in 2002, and the M.S. and Ph.D. degrees from the Department of Computer Science and Information Engineering at the National Cheng Kung University, Tainan, Taiwan in 2004 and 2009. Now, he is an assistant professor in the Department of Business Administration, CTBC Financial Management College, Tainan, Taiwan. He has many excellent research papers about folded hypercubes published on famous journals, such as Theoretical Computer Science, Discrete Applied Mathematics, Information Sciences, and Computers and Mathematics with Applications. His current research interests include interconnection networks, discrete mathematics, computation theory, graph theory, and algorithm analysis.

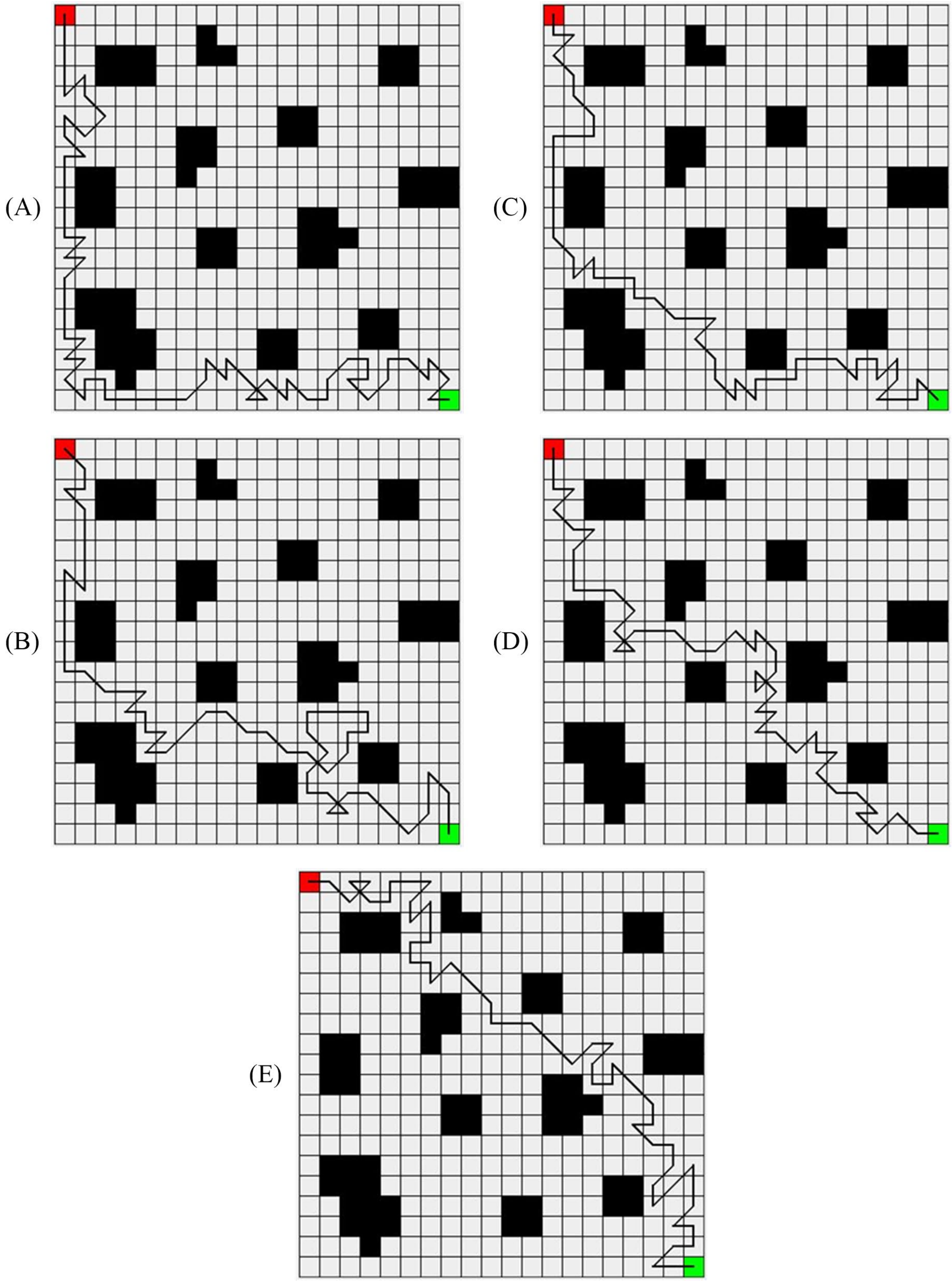

Fig. 6. The planned paths by using TLILBO based on the 20×20 obstacle environmental map with different learning population sizes. (A) learning population size set to 10; (B) learning population size set to 20; (C) learning population size set to 30; (D) learning population size set to 40, and (E) learning population size set to 50.

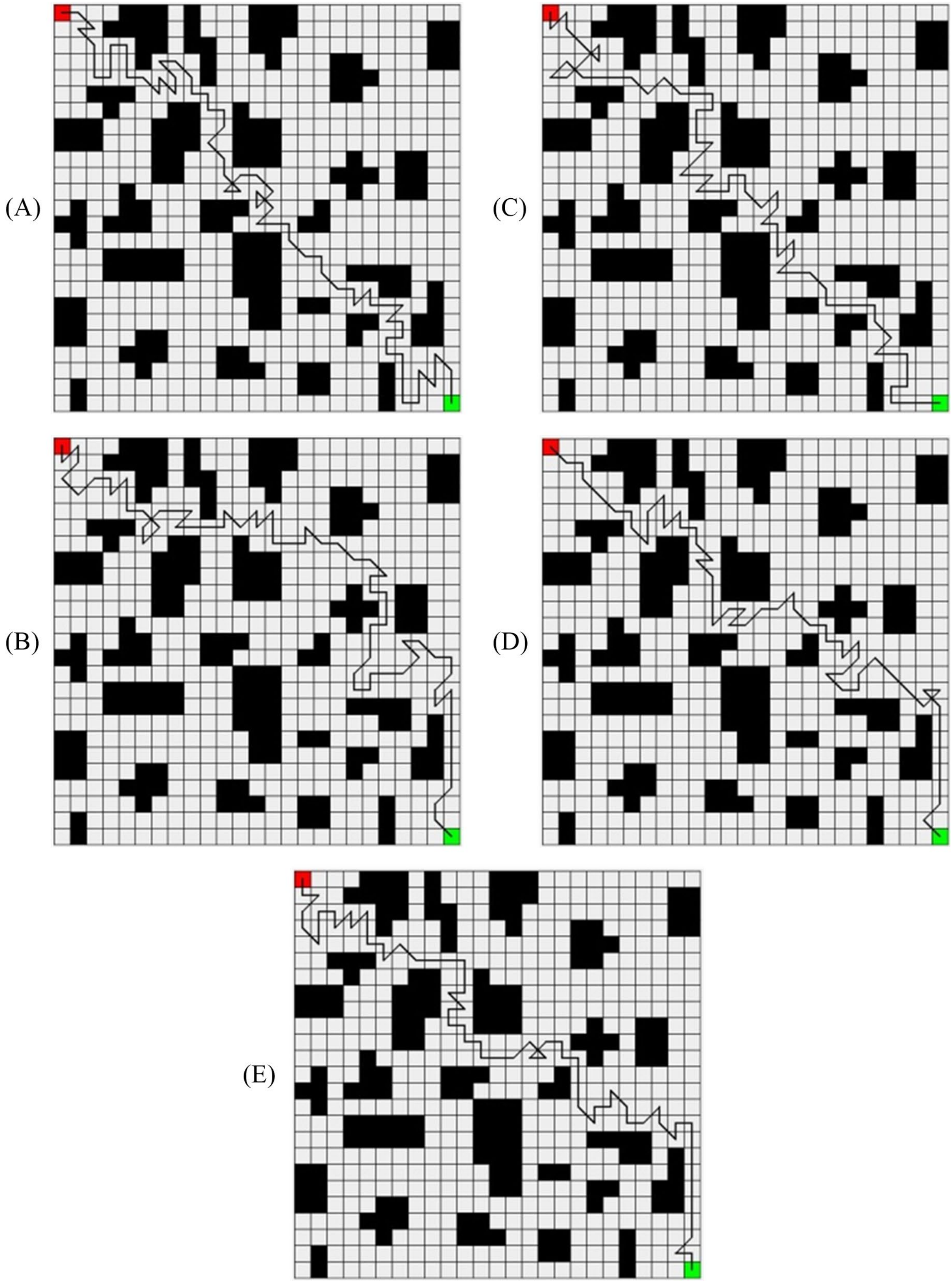

Fig. 7. The planned paths by using TLILBO based on the 25×25 obstacle environmental map with different population sizes. (A) learning population size set to 10; (B) learning population size set to 20; (C) learning population size set to 30; (D) learning population size set to 40, and (E) learning population size set to 50.