Mobile Robot Path Planning Based on Genetic Algorithm and Immune Genetic Algorithm

Sendren Sheng-Dong Xu* and Ya-Po Wu

***Abstract*—Based on the Genetic Algorithm (GA) and Immune Genetic Algorithm (IGA), this letter discusses the mobile robot path planning with different velocity constraints. If the cost function for the optimal robotic path planning is defined as the distance travelled by a robot, the optimal solution thus means the shortest path. However, in some cases, a mobile robot may move in different velocities due to the different terrain conditions. Under such cases, the shortest path will not necessarily represent the shortest time. GA and IGA are utilized to solve the optimal robotic path planning issue considering obstacle avoidance and velocity constraints. Four different terrain conditions are applied in this study, and four different moving velocities are assumed, respectively. Simulation results indicate that both GA and IGA will work effectively to get the optimal path for mobile robotic navigation. There is little difference between both of their CPU execution time. In some cases, IGA will get the same results with those by GA, however, in some cases, IGA can get the better results than those by GA.**

Index Terms*—Genetic Algorithm (GA), Immune Genetic Algorithm (IGA), mobile robot path planning, obstacle avoidance, velocity constraints.

I. INTRODUCTION

THE evolutionary algorithms, mimicking the biological mechanism to get an optimal design in specific constraints, have been widely applied to the control, path planning, and navigation in robot research, e.g., see [1]-[17]. Genetic Algorithm (GA) was first proposed by Holland 1975 [1] based on the genetic scheme and bio-evolution. Being a global search algorithm, it is capable of producing better solutions in complex situations through chromosome representation, reproduction, crossover, and mutation [1], [18].

The basic theory of the immune system was first proposed by Jerne in 1973 [19]. Richter set up a mathematical model based on the theory proposed by Jerne [20]. Dasgupta proposed the immune system theory and discussed its model applicable to various research areas in 1997 [21]. Dote developed an algorithm to model the optimization problem of the immune system in 1988 [22]. The design concept of Immune Genetic Algorithm (IGA) simulates the immune system. According to the characteristics of antibodies to antigens, the antibodies react to antigens. Through calculation, the most suitable antibodies can be found. The best antibodies in the evolution of each generation can be selected. The antibodies and antigens are regarded as spatial solutions and fitness functions, respectively, and the similarity of antibody populations is used to increase the diversity of antibody populations, thereby reducing the possibility of falling into the regional optimal solution. Therefore, in the process of solving, IGA can quickly converge to get an optimized solution and then improve the efficiency of the overall algorithm.

In the research of the mobile path planning, if we divide the real map into many grids, and give the starting point as well as the end point, then many different paths can be selected to complete the moving mission. In general, the cost function for the optimal robotic path planning will be defined as the distance traveled by a robot, and the optimal solution thus means the shortest path, i.e., the shortest time. However, in some cases, a robot may move in different velocities due to the different terrain conditions. In this study, we consider the mobile robotic path planning issue based on this viewpoint. Under different terrain conditions, a robot will be assumed to move with different velocity constraints, and the shortest path will not necessarily represent the shortest time. Therefore, GA and IGA are considered in this study to complete the optimal robot navigation.

The rest of the letter is organized as follows. Section II discusses the design of GA. Section III states the design of IGA. Section IV gives the simulation and discussion. Finally, Section V draws the conclusion.

II. DESIGN OF GENETIC ALGORITHM

The Genetic Algorithm (GA) is operated by the following steps:

Step 1: According to the number of the parent population, randomly initialize the path number. Here, the initialization is regarded as the first-generation parent population.

Step 2: According to the fitness function, design the fitness value of each gene in the parent population.

Step 3: Use reproduction to decide the gene groups of the mating pool.

Step 4: Select genes from the mating pool to do crossover and replace the originally selected genes.

Step 5: Select genes from the group after mating to operate mutation action, and replace the originally selected genes.

Step 6: Jump back to Step 2 and recalculate the fitness value of the new generation of gene groups until the iteration is completed. The best gene of each generation will be output.

A path from the starting point to the end point is defined as a gene. The initialization must generate the same number of paths as the genes in the parent population. The most important step is to ensure that the path of each gene must be continuous and unobstructed during the initialization process in order to complete a path without error.

This paper was first submitted on December 21, 2020.

This work was supportted by Ministry of Science and Technology (MOST), under the Grant MOST 109-2221-E-011-074.

S. S.-D. Xu and Y.-P. Wu are with the Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei, Taiwan.

*corresponding author's e-mail: sdxu@mail.ntust.edu.tw.

In the genetic algorithm, by designing and calculating the fitness function, we can judge if a gene is good or worse. The good gene will get the chance to be selected, and the bad one will be eliminated. We design the fitness function as:

$$f_k = \frac{1}{\sqrt{n_k} \cdot t_k}, \tag{1}$$

in which f_k represents fitness value of the k^{th} gene; n means the total number of walking grids; t is total moving time in the k^{th} gene path.

III. Design of Immune Genetic Algorithm

The Immune Genetic Algorithm (IGA) is operated by the following steps:

Step 1: According to the number of the parent population, randomly initialize the path number. Here, the initialization is regarded as the first-generation parent population.

Step 2: Calculate the fitness value of each gene in the parent population according to the designed fitness function.

Step 3: Calculate: (i) the similarity of fitness values between genes and genes; (ii) the expected reproductive rate; (iii) the probability for the genes to be selected and copied to the mating pool.

Step 4: Select genes to the mating pool according to the selection probability.

Step 5: Select genes from the mating pool to do crossover and replace the originally selected genes.

Step 6: Select genes from the group after mating to operate mutation action, and replace the originally selected genes.

Step 7: Because the good genes of each generation are retained, inferior genes with poor fitness values will be deleted in the process. Therefore, it is necessary to judge whether the number of gene groups is the same as that of the designed parent population. If they are different, a new gene path must be generated to complement the number of gene groups.

Step 8: Jump back to Step 2 and recalculate the fitness value of the new generation of gene groups until the iteration is completed. The best gene of each generation will be output.

The calculation of similarity is derived from the theory proposed by Jerne [19]. A variety of antibodies are produced by lymphocytes to fight against foreign antigens. Although each antibody has a specific antigen to fight against, there are still a lot of similarities between antibodies. For this part, IGA defines how similar the information contained in each gene of the parent population will be.

Assume that there are n genes in the gene group of each generation, and the length of each gene is expressed as $m \times 1$, that is, each gene is a vector of m elements. Two genes are represented as $\mathbf{u} = \{u_1, u_2, \cdots, u_m\}$ and $\mathbf{v} = \{v_1, v_2, \cdots, v_m\}$, respectively, and the fitness values of the two genes are $f_{\mathbf{u}}$ and $f_{\mathbf{v}}$, respectively. Let ε represent the threshold for the similarity of genes, and $\varepsilon > 0$. Then, the similarity of two genes $f_{\mathbf{u}}$ and $f_{\mathbf{v}}$ can be set as the following $Q_s(\mathbf{u}, \mathbf{v})$ [23]:

$$1 - \varepsilon \leq Q_s(\mathbf{u}, \mathbf{v}) = \frac{f_{\mathbf{u}}}{f_{\mathbf{v}}} \leq 1 + \varepsilon. \tag{2}$$

Next, we discuss the expected reproductive rate. If there are n genes in the gene group, the probability of the k^{th} gene being selected for reproduction in the whole group can be calculated by the following equation:

$$e_k = \frac{f_k}{(C_k)^{\beta}}, \tag{3}$$

in which e_k means the expected reproductive rate; f_k represents fitness value of the k^{th} gene; C_k is the number of gene similarities calculated for the k^{th} gene satisfying Eq. (1); β is one of the important setting parameters influencing the fitness value of genes and the number of similarity in the expected reproductive rate.

Concerning the selection probability, it can be calculated by the following equation:

$$P_{sk} = \frac{e_k \cdot n}{\sum_{i=1}^{n} e_i}, \tag{4}$$

in which P_{sk} represents the probability of the k^{th} gene being selected for mating and mutation; e_k is the expected reproductive rate calculated in Eq. (4); $\sum_{i=1}^{n} e_i$ represents the sum of the expected reproductive rates of all genes in this generation; n represents the number of parent population. Here, the probability of each gene being copied to the mating pool is redefined. Finally, just like the roulette type selection in the genetic algorithm, the greater the fitness value of the recalculated gene, the more the path meets the requirement. Then, it has more chance to be selected for reproduction, and entering the mating pool to reproduce the next generation.

IV. Simulation and Discussion

In this study, we design two kinds of grid-based maps by the size of Case (1) 30×30; Case (2) 50×50, and discuss the following two terrain cases, respectively: Case (a) Without terrain variation; Case (b) Four different terrain conditions and four different moving velocities, respectively. For the size of 30×30 grid-map, we have designed the experiments with three sets of different starting points and end points as Case (i) {(11, 25), (30, 07)}; Case (ii) {(30, 07), (02, 28)}; Case (iii) {(02, 28), (17, 14)}. For the size of 50×50 grid-map, we have designed the experiments with five sets of different starting points and end points as Case (I) {(46, 48), (07, 26)}; Case (II) {(04, 02), (28, 44)}; Case (III) {(22, 03), (29, 19)}; Case (IV) {(29, 19), (42, 26)}; Case (V) {(42, 07), (22, 03)}. Moreover, the Open Vehicle Routing Problem (OVRP) is also discussed with: Case (A) The grid-based map with the size of 30×30, starting points and end points as {(01, 03), (17, 14)}, and four intermediate nodes; Case (B) The grid-based map with the size of 50×50, starting and end points as {(26, 28), (15, 36)}, and nine intermediate nodes. Both GA and IGA are utilized to do the path planning. We use 100, 300, and 500 iterations. The number of the genes in the parent population is 10. The number of genes in the mating pool is 10. The mutation probability is 80%, and the crossover probability is 10%. In IGA, ε is 95% and β is 2. Some of the simulation results are shown in Figs 1-14.

All the simulation results are in compliance with the restrictions on the mobile robot path movement, such as the obstacle avoidance effect of turning and moving on different terrain conditions.

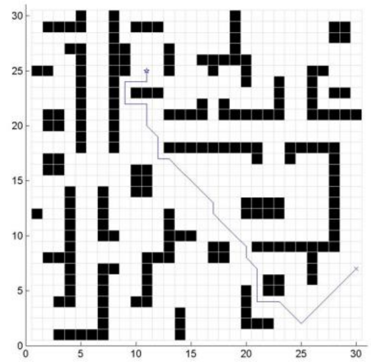

Fig. 1. Path planning by the parent population (the 0th generation) of GA in Case (1)-(a)-(i) 30×30 grid-map, without terrain variation, starting point and end point {(11, 25), (30, 07)}.

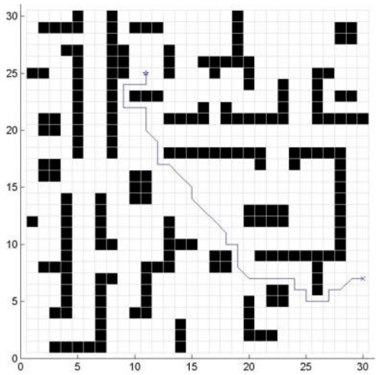

Fig. 2. Path planning by the 1th generation of GA in Case (1)-(a)-(i) 30×30 grid-map, without terrain variation, starting point and end point {(11, 25), (30, 07)}.

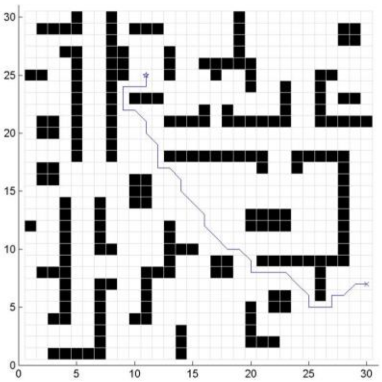

Fig. 3. Optimal path planning by GA in Case (1)-(a)-(i) 30×30 grid-map, without terrain variation, starting point and end point {(11, 25), (30, 07)}.

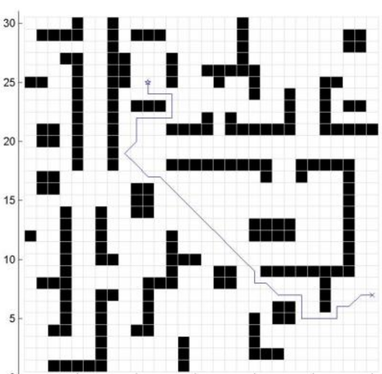

Fig. 4. Path planning by the parent population (the 0th generation) of IGA in Case (1)-(a)-(i) 30×30 grid-map, without terrain variation, starting point and end point {(11, 25), (30, 07)}.

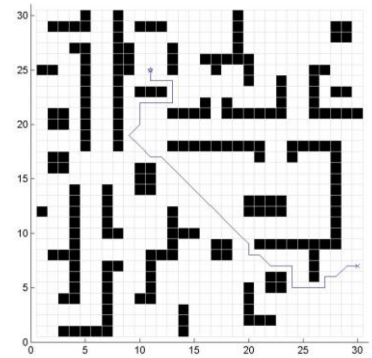

Fig. 5. Path planning by the 1th generation of IGA in Case (1)-(a)-(i) 30×30 grid-map, without terrain variation, starting point and end point {(11, 25), (30, 07)}.

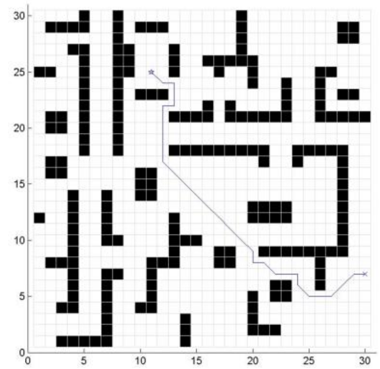

Fig. 6. Optimal path planning by IGA in Case (1)-(a)-(i) 30×30 grid-map, without terrain variation, starting point and end point {(11, 25), (30, 07)}.

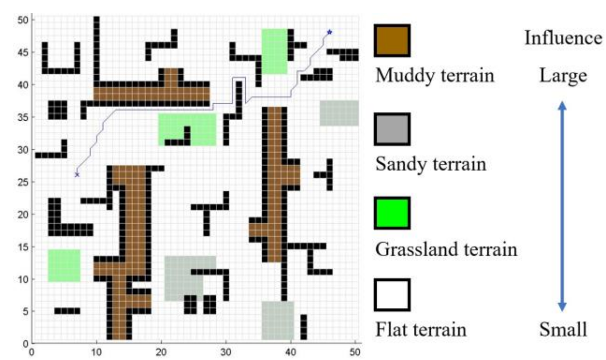

Fig. 7. Path planning by the parent population (the 0th generation) of GA in Case (2)-(b)-(I) 50×50 grid-map, four different terrain conditions, starting point and end point {(46, 48), (07, 26)}.

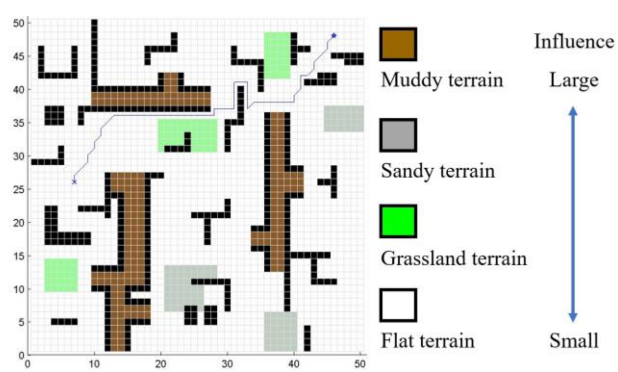

Fig. 8. Path planning by the 1th generation of GA in Case (2)-(b)-(I) 50×50 grid-map, four different terrain conditions, starting point and end point {(46, 48), (07, 26)}.

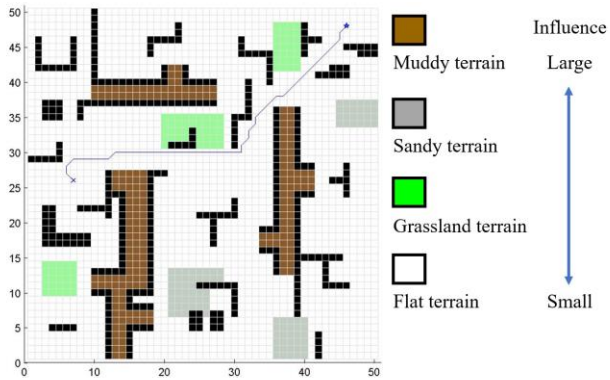

Fig. 9. Optimal path planning by GA in Case (2)-(b)-(I) 50×50 grid-map, four different terrain conditions, starting point and end point {(46, 48), (07, 26)}.

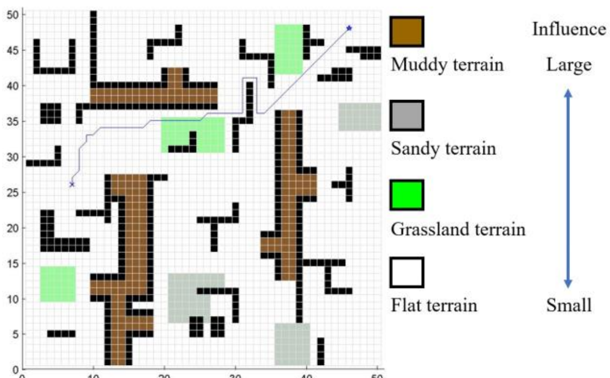

Fig. 10. Path planning by the parent population (the 0^{th} generation) of IGA in Case (2)-(b)-(I) 50×50 grid-map, four different terrain conditions, starting point and end point {(46, 48), (07, 26)}.

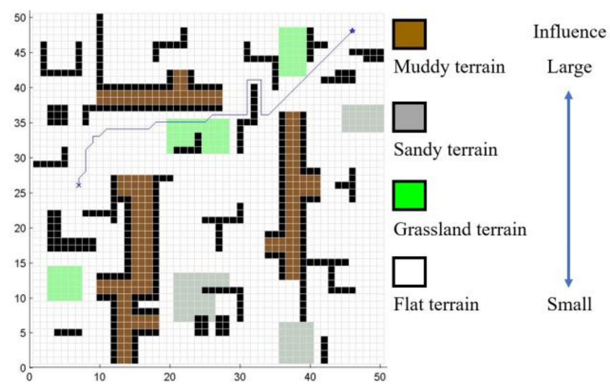

Fig. 11. Path planning by the 1^{th} generation of IGA in Case (2)-(b)-(I) 50×50 grid-map, four different terrain conditions, starting point and end point {(46, 48), (07, 26)}.

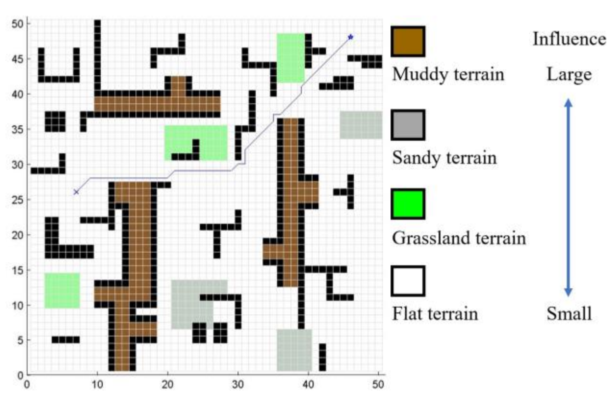

Fig. 12. Optimal path planning by IGA in Case (2)-(b)-(I) 50×50 grid-map, four different terrain conditions, starting point and end point {(46, 48), (07, 26)}.

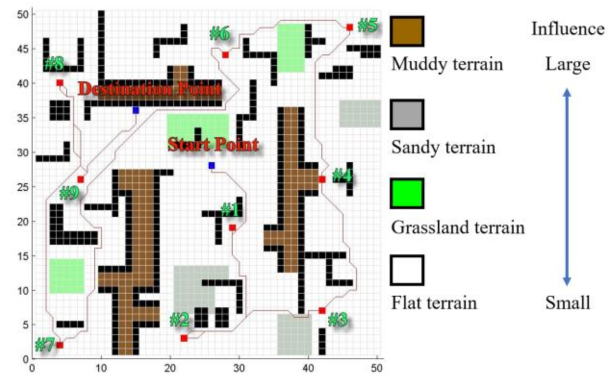

Fig. 13. Optimal path planning by GA in Case (2)-(b)-(B) 50×50 grid-map, four different terrain conditions, OVRP with s starting point and end point as {(26, 28), (15, 36)} and nine intermediate nodes.

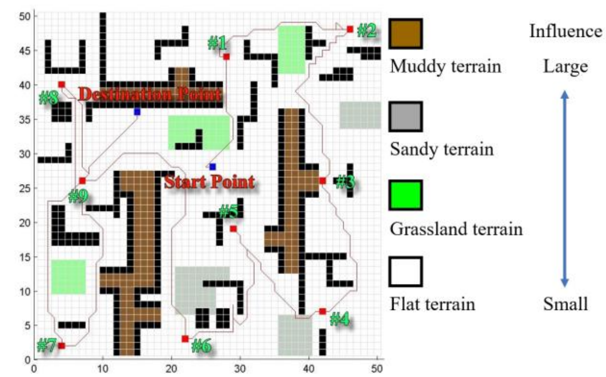

Fig. 14. Optimal path planning by IGA in Case (2)-(b)-(B) 50×50 grid-map, four different terrain conditions, OVRP with starting point and end point as {(26, 28), (15, 36)} and nine intermediate nodes.

The original shortest path planning is also changed after calculation by algorithms due to terrain changes. By using GA and IGA, it is not easy to fall into the local solutions. It still can be seen from the simulation results that even if the number of iterations has been completed, there are still some fragments of imperfect paths. However, it is very close to the optimal solution for the entire problem. IGA is better than GA in fitness performance in various path planning cases. Under the cases discussed in this research for mobile robot path planning, IGA in some cases performs better than GA even though there is little difference between their CPU execution time.

V. Conclusion

In this study, we consider the mobile robotic path planning issue based on the fact that a robot may move under different terrain conditions. A robot will be assumed to move with different velocity constraints and the shortest path will not necessarily represent the shortest time. Since both Genetic Algorithm (GA) and Immune Genetic Algorithm (IGA) own the characteristic of not easy falling into the local solution, we can utilize them to solve the optimal robotic path planning issue considering different terrain cases. Four different terrain conditions are applied in this study, and thus four different moving velocities are assumed, respectively. Moreover, the Open Vehicle Routing Problem (OVRP) is also discussed. Simulation results show that both GA and IGA will work effectively to get the optimal path for mobile robotic navigation. There is little difference between both of their CPU execution time. In some cases, IGA will get the same results with those by GA, however, in some cases, IGA can get the better results than those by GA.

REFERENCES

[1] John H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[2] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 31, no. 1, pp. 718-724, Feb. 2004.

[3] X. Yang, M. Moallem, and R. V. Patel, "A layered goal-oriented fuzzy motion planning strategy for mobile robot navigation," *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 35, no. 6, pp. 1214-1224, Dec. 2005.

[4] G. Chesi and Y. S. Hung, "Global path planning for constrained and optimal visual servoing," *IEEE Trans. Robotics,* vol. 23, no. 5, pp. 1050-1060, Oct. 2007.

[5] G. Antonelli, S. Chiaverini, and G. Fusco, "A fuzzy-logic-based approach for mobile robot path tracking," *Fuzzy Systems, IEEE Trans. Fuzzy Systems,* vol. 15, no. 2, pp. 211-221, April 2007.

[6] C. Cai, C. Yang, Q. Zhu, and Y. Liang, "A fuzzy-based collision avoidance approach for multi-robot systems," *IEEE International Conference on Robotics and Biomimetics*, pp.1012-1017, Sanya, Dec. 15-18, 2007.

[7] H. Qu, S. X. Yang, A. R. Willms, and Z. Yi, "Real-time robot path planning based on a modified pulse-coupled neural network model," *IEEE Trans. Neural Networks,* vol. 20, no. 11, pp. 1724-1739, Nov. 2009.

[8] Z. Yao and K. Guta, "Distributed roadmaps for robot navigation in sensor networks," *IEEE Trans. Robotics,* vol. 27, no. 5, pp. 997-1004. Oct. 2011.

[9] Y. Jia, X. Song, and S. S.-D. Xu, "Modeling and motion analysis of four-Mecanum wheel omni-directional mobile platform," in *Proc. CACS International Automatic Control Conference*, Sun Moon Lake, Nantou, Taiwan during Dec. 2-4, 2013.

[10] H.-C. Huang, S. S.-D. Xu, and H.-S. Hsu, "Hybrid Taguchi DNA swarm intelligence for optimal inverse kinematics redundancy resolution of six-DOF humanoid robot arms," *Mathematical Problems in Engineering*, vol. 2014, Article ID 358269, pp. 1-9, 2014.

[11] H.-C. Huang, S. S.-D. Xu, and C.-H. Chiang, "Optimal fuzzy controller design using an evolutionary strategy-based particle swarm optimization for redundant wheeled robots," *International Journal of Fuzzy Systems*, vol. 17, no. 3, pp. 390-398, 2015.

[12] S. S.-D. Xu, H.-C. Huang, T.-C. Chiu, and S.-K. Lin, "Development of a biologically inspired RBFNN and its application to self-tuning motion control of autonomous vehicles," *iRobotics*, vol. 1, no. 2, pp. 28-34, 2018.

[13] A. Koubaa, H. Bennaceur, I. Chaari, and S. Trigui, A. Ammar, M.-F. Sriti, M. Alajlan, O. Cheikhrouhou, Y. Javed, "Introduction to mobile robot path planning," in book: *Robot Path Planning and Cooperation: Foundations, Algorithms and Experimentations*, pp.3-12, Springer International Publishing, April 2018.

[14] B. K. Patle, B. L. Ganesh, A. Pandey, D. R. K. Parhi, A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Defence Technology*, vol. 5, no. 4, pp. 582-606, 2019.

[15] S. S.-D. Xu, H.-C. Huang, T.-C. Chiu, S.-K. Lin, "Biologically-inspired learning and adaptation of self-evolving control for networked mobile robots," *Applied Sciences,* vol. 9, no. 5, pp. 1-18, March 12, 2019.

[16] S. S.-D. Xu, H.-C. Huang, Y.-C. Kung, and S.-K. Lin "Collision-free fuzzy formation control of swarm robotic cyber-physical systems using a robust orthogonal firefly algorithm," *IEEE Access*, vol. 9, pp. 9205-9214, 2019.

[17] S. S.-D. Xu, H.-C. Huang, Y.-C. Kung, and S.-K. Lin, "A networked multirobot CPS with artificial immune fuzzy optimization for distributed formation control of embedded mobile robots," *IEEE Trans. Industrial Informatics*, vol. 11, no. 1, pp. 414-422, 2020.

[18] K.-Y. Wu, S. S.-D. Xu, and T.-C. Wu, "Optimal scheduling for retrieval jobs in double-deep AS/RS by evolutionary algorithms," *Abstract and Applied Analysis*, vol. 2013, pp.1-17, 2013.

[19] N. K. Jerne, "The immune system," *Science American*, vol. 229/1, pp. 52-60, 1973.

[20] P. H. Richter, "A network theory of the immune system," *European Journal of Immunology*, vol. 5, no. 5, pp. 350-354, May 1975.

[21] D. Dasgupta, "Artificial neural networks and artificial immune systems: Similarities and differences," *International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, Orlando, Florida, USA, Oct. 12-15, 1997, vol. 1, pp. 873-878.

[22] Y. Dote, "Soft computing (immune networks) in artificial intelligence," *International Conference on Systems, Man, and Cybernetics,* vol. 2, San Diego, California, USA, Oct. 14, 1998, pp. 1382-1387.

[23] X. Chen, G.-Z. Tan, and B. Jiang, "Real-time optimal path planning for mobile robots based on immune genetic algorithm," *Journal of Central South University*, vol. 39, no. 3, pp. 577-583, June 2008.

Sendren Sheng-Dong Xu received the Ph.D. degree from the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 2009. He is currently a Professor with the Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei, Taiwan. His current research interests include intelligent control systems, signal processing, image processing, and embedded systems.

Ya-Po Wu received the M.S. degree from the Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei, Taiwan, in 2013. He is currently an engineer in the Realtek Semiconductor Corporation, Hsinchu, Taiwan. His research interests include the algorithm application and software coding.