

PATH PLANNING ALGORITHM BASED ON TEACHING-LEARNING-BASED-OPTIMIZATION FOR AN AUTONOMOUS VEHICLE

Ahmed D. Sabiha^{1,*}, Mohamed A. Kamel², Ehab Said¹, Wessam M. Hussein³

¹Department of Mechatronics Engineering, Military Technical College, Cairo, Egypt

²Department of Mechanical Engineering, Military Technical College, Cairo, Egypt

³Department of Mechatronics Engineering, Egyptian Academy for Engineering and Advanced Technology (EAEAT), Cairo, Egypt

*E-mail of corresponding author: ahmeddesokyeg@gmail.com

Resume

This study provides a teaching-learning-based optimization (TLBO) path planning method for an autonomous vehicle in a cluttered environment, which takes into account path smoothness and the possibility of collision with nearby obstacles. The path planning problem is tackled as a multi-objective optimization in order to plan an efficient path that allows the vehicle to travel autonomously in crowded settings. The TLBO algorithm is used to find the ideal path, with the goals of finding the shortest path to the target site and maximizing path smoothness, while avoiding obstacles and taking into account the vehicle's dynamic and algebraic properties.

Article info

Received 1 October 2021

Accepted 18 January 2022

Online 22 February 2022

Keywords:

autonomous vehicle
path planning
optimization
teaching-learning-based
optimization
TLBO

Available online: <https://doi.org/10.26552/com.C.2022.2.C33-C42>

ISSN 1335-4205 (print version)

ISSN 2585-7878 (online version)

1 Introduction

The ability to reach a target and avoid obstacles are critical elements of autonomous vehicle navigation. As a result, the vehicle interacts with its surroundings and perceives the world in which it works. Path planning is essential for the vehicle to be totally autonomous and dependable.

Its goal is to plan a series of suitable paths via multiple points and segments that allow the vehicle to complete its tasks and arrive at the desired destination while avoiding collisions with surrounding obstacles, as well as achieve other objectives such as minimizing the travelled distance, mission time, consumed energy and/or any other objectives based on the mission type. For path planning, there are two approaches: (i) off-line or global path planning and (ii) online or local path planning [1]. Typically, global path planning generates a high-level, low-resolution path based on knowledge of the environmental map, as well as current and historical perceptive environmental information. This method is effective in creating a path that is optimized. However, the response to dynamic or unanticipated impediments is insufficient. The local path planning approach, on the other hand, does not require a priori knowledge

of the local environment. Based on the obtained on-board sensory data, local path planning generates a minimum-level high-resolution path. However, when the environment is congested or the goal is a significant distance away, however, this method is ineffective. As a result, it is preferable to mix the two approaches in order to maximize their benefits while minimizing some of their drawbacks [2].

Path planning navigational approaches can be categorized into classical and bio-inspired approaches [3]. Before the 2000s, most traditional tactics were employed, but the bio-inspired methods became the most popular [4]. The main idea behind traditional approaches is to either find a realistic solution or confirm that there is not one. Cell decomposition [5-6], roadmap [7], sampling-based algorithms [8] and artificial potential field (APF) [9-10] are the most common traditional techniques. These methods are usually not mutually incompatible and to optimize the path planning process, a hybrid algorithm combining the two traditional techniques is used [11]. Although traditional techniques are straightforward they have a number of limitations, including long calculation times, inability to execute in real time, and regional minima trapping. As a result, bio-inspired techniques



to circumvent these shortcomings have been proposed [3, 12].

Due to their ability to handle environmental uncertainties and find the near-optimal path, while considering the vehicle's algebraic and dynamic constraints, as well as any other constraints corresponding to the vehicle or its assigned mission, computational intelligent approaches have recently become the most dominant in the field of mobile robot navigation. Genetic algorithm (GA), fuzzy logic (FL), neural network (NN), particle swarm optimization (PSO), firefly algorithm (FFA) and most recently, teaching-learning-based optimization are the main bio-inspired methodologies utilized for unmanned ground vehicles (UGVs) path planning (TLBO). Bhaskar et al. [13] propose a GA-based path planner for a mobile robot to design a collision-free path. The proposed method uses GA with a modified mutation operation as a solution strategy to guarantee a random selection of grids from the available zones without any repetition. The suggested model aims to reduce computing complexity by shrinking the search space. Application of the PSO aids to reduction of calculations and maintenance of more steady convergence characteristics. In [14] an improved PSO algorithm is proposed to plan an optimized smooth path for mobile robots to tackle the local trapping and premature convergence issues. This algorithm is combined with the continuous high-degree Bezier curve to smooth the path. In [15], an optimal method of path planning based on FFA with self-adaptive population size is proposed. The evaluation of the degree of collision is established at the costs of avoiding the collision. Two nonlinear functions are presented to determine the population size based on the degree of population collision. Individuals are then added or removed from the firefly population. In terms of solution stability, running time and convergence speed, the proposed algorithm has the better performance.

It should be mentioned that all the bio-inspired techniques have some advantages and disadvantages. As a result, various hybrid algorithms combining two strategies have been developed to increase the path planner's overall performance [16-19]. It is also worth noting that all the evolutionary and swarm intelligence-based algorithms are probabilistic and require the same set of regulating parameters, such as population size and generation count. Furthermore, different algorithms necessitate their own sets of control settings. The GA, for instance, employs mutation and crossover rates. The PSO, too, makes advantage of inertia weight, as well as social and cognitive characteristics. The proper adjustment of algorithm-specific parameters is a critical aspect affecting optimization algorithm performance. The incorrect tuning for specific parameters of the algorithm either increases processing effort or produces the local best result. Considering this fact, the TLBO is proposed as it does not need any algorithm specific parameters.

The TLBO technique is considered a metaheuristic algorithm proposed by Rao et al. in 2011 [20]. It is inspired by the process of teaching and learning, via a simplified mathematical model of knowledge improvements gained by students in a class [21]. The basic idea of the TLBO is that a group of learners constitutes the population, while the teacher is considered the most learned person, i.e., the learner with the best solution. The TLBO is divided into two phases, the teaching phase and the learning phase. The first phase means learning from the teacher, while the second one means learning through the interaction between learners.

For the path planning of robotic manipulators, the TLBO has been frequently proposed [22-23]. However, just a few studies have used the TLBO to plan UGV paths. For the mobile robot path planning, the nonlinear inertia weighted teaching-learning-based optimization (NIWTLBO) technique is proposed in [24]. Coordinate system transformation creates a new map representation of the path between the start and goal points. To obtain a globally optimal path, the NIWTLBO method is used to optimize the path's objective function. The proposed method's effectiveness is demonstrated by simulation results. However, the findings of the experiments are not included. To acquire the parameters of the adaptive neural fuzzy inference system (ANFIS), the TLBO is proposed in [25]. Comparisons of the obtained results utilizing the recommended approach to findings from other intelligent algorithms such as PSO, invasive weed optimization (IWO) and biogeography-based optimization (BBO), are used to validate the results. The high quality of the simulation results confirms that the TLBO-based ANFIS is an effective alternative strategy for tackling the differential drive wheeled mobile robot navigation problem.

In [26], a conformal geometric algebra and TLBO are proposed for path planning of mobile robot avoiding all the possible obstacles in the navigation task. The proposed method is conformal spheres to represent the mobile robot and the obstacles. Then, a conformal translator is proposed to translate the mobile robot sphere to a new sphere position. The TLBO algorithm optimizes this translator by minimizing the distance between the new position and the objective position while maximizing the distance between the new position and any potential obstacles. In [27], the TLBO technique is proposed for path planning to guide a three-wheeled robot. The objective function has been proposed to optimize the path planning and reaching destination point. The obtained simulation results show that the TLBO technique finds the near-optimal path. In [28], a hybrid strategy combining TLBO and the shuffled frog leaping algorithm (SFLA) is presented for a mobile robot to enhance exploitation efficiency and overcome SFLA's sluggish convergence rate. The proposed approach is named as shuffled teaching-learning-based optimization (STLBO) algorithm. In comparison to the standard TLBO and PSO, simulation findings demonstrate that

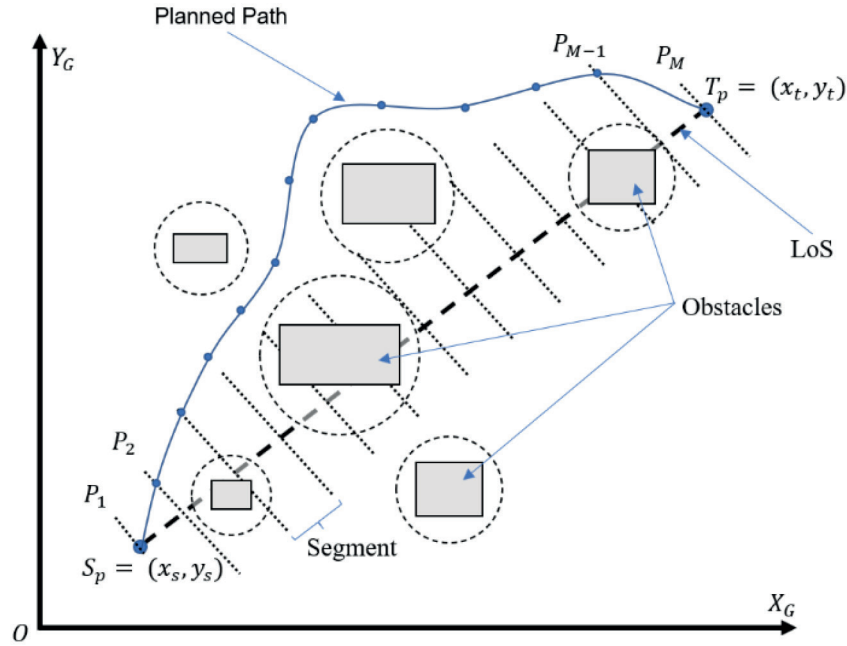


Figure 1 Path planning modeling in 2D space

the STLBO produces better results.

From a review of the literature, the following results are summarized for the problem of unmanned vehicles' path planning:

- Traditional methods are straightforward. However, they have significant drawbacks, such as long computation times, difficulty with online implementation and local minima trapping;
- Bio-inspired approaches are able to find the near-optimal path. However, their main demerit is the selection and tuning of the algorithm-specific parameters; and
- The TLBO is widely applied in the path planning of robotic manipulators [22-23]. However, very few works focused on applying TLBO for UGV path planning [26-28].

Compared to the related works in the literature, the main contributions of this paper can be summarized as follows:

- Developing the near-optimal path planning algorithm based on the TLBO that obtains the desired collision-free path that the vehicle must follow to reach its target; and
- Modelling and simulation of the proposed algorithm to validate its efficiency.

2 Path planning problem

Let $q \in C$ be the vector of generalized coordinates for the vehicle, $q(t) = [x, y, \theta]^T$ and C is the vehicle's configuration space. The vehicle moves in an unknown cluttered environment starting from S_p with coordinates (x_s, y_s) , while the goal is to reach the target location T_p with coordinates (x_t, y_t) , as shown in Figure 1. During motion, let q be the current posture of the vehicle's

center of gravity with coordinates (x, y) and orientation angle θ .

In the vehicle configuration space C , the start and the target points are connected with a straight line-of-sight (LoS), regardless the obstacles. Considered as the shortest path, LoS is divided into $(M - 1)$ segments and M points starting from P_1 to P_M , where every two successive points define a segment which is a part of the path linking S_p to T_p . Consequently, to avoid the potential collision with the surrounding obstacles and considering all other constraints, a new path consists of $(M - 1)$ segments and M points (including source and target locations) is planned. Based on the aforementioned explanation, the objective of the path planning algorithm is to find the optimum coordinates of $(M - 2)$ points (as S_p to T_p are already known) that minimizes the path length, maximizes the path smoothness and avoids the potential collision with the surrounding obstacles. For this purpose, TLBO is proposed.

As mentioned before, the vector of vehicle position is defined as $q(t) = [x, y, \theta]^T$. As the path is divided into $(M - 1)$ segments, at the end of any segment, the vehicle position can be obtained as:

$$q_i = [x_i, y_i, \theta_i], \forall i \in 2, 3, \dots, (M - 1), \quad (1)$$

In this work, points P_1 and P_M are considered as the starting and target points S_p and T_p , respectively. Consequently, the optimal path planning problem can be stated as: finding the optimum coordinates (x_i, y_i) and orientation angle $\theta_i, i \in 2, 3, \dots, (M - 1)$ in order to minimize the path length and maximize the path smoothness such that:

$$\min_u J(u), J(u) = w_1 J_1(u) + w_2 J_2(u), \quad (2)$$

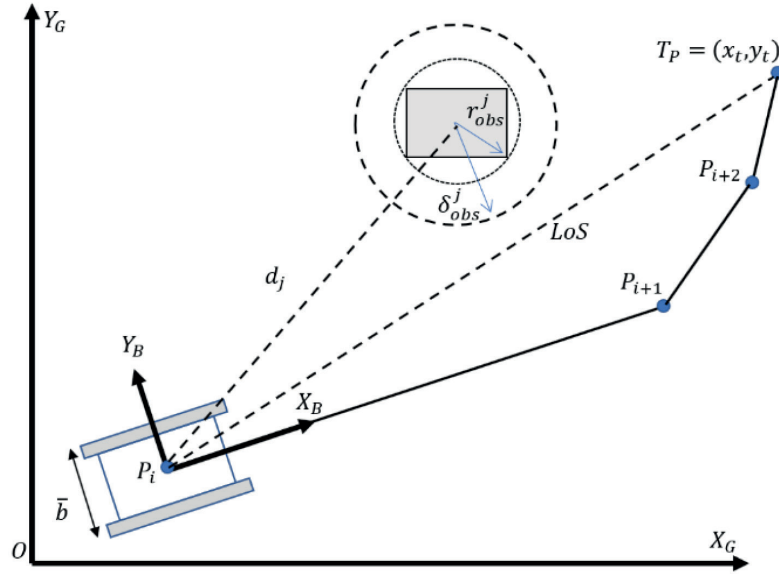


Figure 2 The vehicle-obstacle interaction

where $J(u)$ is the total objective function, $J_1(u)$ is the objective function that minimizes the path length, $J_2(u)$ is the objective function that maximizes the path smoothness, w_1 and w_2 are weighting factors and u is the vector of m design variables, such that:

$$u = [x_i, y_i], \forall i \in \{2, 3, \dots, (M-1)\}, u \in R^{2 \times (M-2)}. \quad (3)$$

2.1 Objective function calculation

The values of J_1 and J_2 can be calculated as follows:

- Minimizing the path length via J_1 :

The total path length is determined by:

$$J_1(u) = S = \min \sum_{i=1}^M S(P_i, P_{i+1}) = \sum_{i=1}^M (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2, \quad (4)$$

where $S(P_i, P_{i+1})$ presents the distance between two successive points P_i and P_{i+1} .

- Maximizing the path smoothness via J_2 .

The smoothness of a path is a very important attribute in the vehicle path planning, since the vehicle should not significantly change its direction suddenly. The path smoothness leads to reduction of the energy consumption and time waste. So, the smoothness is considered as a second objective. Therefore, when maximizing the path smoothness, the total of the vehicle's turning angle is minimized. As a result, the objective function of the vehicle's turning angle is defined as follows:

$$J_2(u) = \Theta = \min \sum_{i=1}^{M-1} |\theta_i|, \quad (5)$$

where θ_i is the angle between two vectors $\overrightarrow{P_{i-1}P_i}$ and $\overrightarrow{P_iP_{i+1}}$

$$\theta_i = \text{atan2}\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right) - \text{atan2}\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right) \quad (6)$$

$$\forall i \in \{2, 3, \dots, M\}.$$

2.2 Optimization constraints:

The following constraints apply to the optimization problem:

- The boundary constraints:

in which the coordinates of P_i are always within the vehicle's configuration space C , i.e.

$$\{x_i, y_i\} \in C, \forall i \in \{2, 3, \dots, (M-1)\}. \quad (7)$$

- The collision avoidance constraint:

During the vehicle motion, the distance between the vehicle and the j^{th} obstacle, $j = \{1, 2, \dots, N_{obs}\}$, where N_{obs} is the number of detected obstacles, can be defined as:

$$d_j = \sqrt{(x_i - x_{obs_j})^2 + (y_i - y_{obs_j})^2}, \quad (8)$$

$$\forall i \in \{2, 3, \dots, (M-1)\}, \forall j \in \{1, 2, \dots, N_{obs}\},$$

where obs_j denotes the center of gravity of the j^{th} obstacle. As shown in Figure 2, each obstacle j is surrounded by a circle with radius r_{obs}^j . To avoid the collision between the vehicle and the j^{th} obstacle, d_j must be greater than the minimum safety distance δ_{obs}^j presented as:

$$\delta_{obs}^j = r_{obs}^j + d_{safe}, \quad (9)$$

where:

$$d_{safe} = \zeta \bar{b}, \quad (10)$$

where ζ is a weighting factor and \bar{b} is the vehicle width.

The weighting factor ζ is a critical value to ensure the collision avoidance of the vehicle. Therefore, $0.5 < \zeta < 1$. The selection of ζ value is significantly critical. While increasing ζ will ensure collision avoidance, it will decrease the possibility of finding the feasible solution of the optimization problem, results in a high computational burden. In this work, ζ is selected to be $\frac{2}{3}$.

Consequently, the collision avoidance constraint can be described as:

$$d_j \geq \delta_{obs}^j, \forall j = \{1, 2, \dots, N_{obs}\}. \quad (11)$$

As a result, the path planning multi-objective optimization problem can be mathematically represented as follows:

$$\min_u J(u), \quad J(u)w_1S + w_2\Theta, \quad (12)$$

subjected to:

$$\{x_i, y_i\} \in C, \forall i \in \{2, 3, \dots, (M-1)\}, \quad (13)$$

$$d_j \geq \delta_{obs}^j, \forall j = \{1, 2, \dots, N_{obs}\}. \quad (14)$$

As the TLBO is a meta-heuristic optimization technique, the optimization problem should be solved as an unconstrained problem. Therefore, the collision avoidance constraint should be included in the objective function. This can be obtained by adding penalties to the objective function for violating the constraints [29]. For a constrained optimization problem with c constraints, the objective function can be represented as an unconstrained problem as follows:

$$J' = J + a_k \sum_{j=1}^c G_j[g_j(X)], \forall j = \{1, 2, \dots, c\}, \quad (15)$$

where G_j is the penalty function of the constraint g_j and a_k is a positive constant known as the penalty parameter. Equation (15) indicates that while minimizing the objective function, a positive penalty is added whenever a constraint is violated [29]. There are many common penalty functions. In this paper, the considered penalty is being proportional to the amount of violation. The values of the constants a_k can be adapted to change the contribution of the penalty terms relative to the magnitude of the whole objective function. As a result, Equation (15) can be re-written as:

$$J(u) = w_1S + w_2\Theta + \sigma \sum_{j=1}^{N_{obs}} R_{obs}^j, \quad \forall j = \{1, 2, \dots, N_{obs}\}, \quad (16)$$

where σ is the penalty coefficient and R_{obs}^j can be calculated as follows:

$$R_{obs}^j = \begin{cases} 0 & \text{if } d_j \geq \delta_{obs}^j \\ \frac{1}{d_j} - \frac{1}{\delta_{obs}^j} & \text{if } d_j < \delta_{obs}^j \end{cases}, \forall j = \{1, 2, \dots, N_{obs}\}. \quad (17)$$

This means that if d_j is greater than δ_{obs}^j , i.e., the distance between the vehicle and the j^{th} obstacle is safe. Therefore, the collision avoidance constraint is not violated and no penalty is added to the objective function. However, if d_j is less than δ_{obs}^j , the vehicle will collide with the j^{th} obstacle. As a result, a penalty is added to the objective function in order to avoid the potential collision.

3 TLBO algorithm

After defining the optimization problem in Section 2, the TLBO algorithm is applied to obtain the optimum path that minimizes the total objective function presented in Equation (15).

Teaching-learning is an important part of the process in which each individual seeks to learn knowledge from others in order to improve themselves. Rao et al. [20, 30] and Rao and Patel [31] proposed the TLBO algorithm, which simulates the traditional teaching-learning phenomenon of a classroom. The algorithm mimics two essential types of learning: (i) from the teacher (known as the teacher phase) and (ii) through the interaction between all the learners (known as the learner phase). Because the TLBO algorithm is considered as a population-based optimization method, a group or set of learners is considered as a population and the design variables are the various subjects offered to the learners. The mean result of each learner is analogous to the objective function of the optimization problem. The best value of the objective function (the best learner) is considered as a teacher, while in each iteration, the TLBO is trying to improve the solution. The execution of TLBO algorithm is given as follows:

Step 1. Construction of the vector of design variables \mathbf{X} (the vector of learners' subjects):

For each learner k , the vector of D_v subjects (the number of design variables) is:

$$\mathbf{X} = [x_c, y_c], \forall c \in \{2, 3, \dots, (M-1)\}, \quad (18)$$

As a result, the length of the vector of design variables is: $D_v = 2 \times (M-2)$

Step 2. Initialization:

- Select the number of learners N_p .
- Select the maximum number of iterations i_{max} .
- Generate a random population of learners as follows:

$$X_{(k,j)}^0 = X_j^{\min} + r_j \times (X_j^{\max} - X_j^{\min}), \quad (19)$$

$$k = [1, 2, \dots, N_p], j = [1, 2, \dots, D_v],$$

where, r_j is a uniformly distributed random number within the range $\in [0, 1]$, N_p denotes the number of learners in a class and X_j^{\max} X_j^{\min} represent the upper and lower bounds of each design variable j , respectively.

- Set the iteration counter i to 1

Step 3.5 Update the values of the design variables:

Based on $DifMean_{i,j}$, the existing solution is updated according to the following expression:

$$X_{(i,k,j)}^{new} = X_{(i,k,j)} + DifMean_{i,j}. \quad (23)$$

Step 3.6 Update the solution:

Recalculating the objective function based on the updated values $X_{(i,j,k)}^{new}$. If $X_{(i,j,k)}^{new}$ yields a better solution than $X_{(i,j,k)}$, then $X_{(i,j,k)}^{new}$ replaces $X_{(i,j,k)}$. Otherwise, $X_{(i,j,k)}$ is preserved. In other words:

$$X_{(i,k,j)} = \begin{cases} X_{(i,k,j)} & \text{if } X_{(i,k,j)}^{new} > J(X_{(i,k,j)}) \\ X_{(i,k,j)}^{new} & \text{if } X_{(i,k,j)}^{new} < J(X_{(i,k,j)}) \end{cases} \quad (24)$$

Step 4. Learner phase:

In this phase, each learner interacts randomly with other learners to enhance their knowledge.

Step 4.1 Interaction:

Consider the two learners P and Q , the interaction between them to improve their knowledge can be obtained as follows:

$$X_{(i,P,j)}^{new} = \begin{cases} X_{(i,P,j)}, & \text{if } J_p = J_q \\ X_{(i,P,j)} + r_j \times (X_{(i,P,j)} - X_{(i,Q,j)}), & \text{if } J_p < J_q, \\ X_{(i,P,j)} + r_j \times (X_{(i,Q,j)} - X_{(i,P,j)}), & \text{if } J_p > J_q \end{cases} \quad (25)$$

Step 4.2 Select the best values for each learner after interaction:

Re-calculate the objective function based on the updated values obtained from Equation (25). Similar to Equation (24), if the new solution corresponding to $X_{(i,j,k)}^{new}$ is better than the current one $X_{(i,j,k)}$, then $X_{(i,j,k)}$ is replaced by $X_{(i,j,k)}^{new}$. Otherwise, $X_{(i,j,k)}$ is

preserved. The accepted individuals of the population are passed and moved on to the next iteration.

Step 5. Termination criterion:

The proposed algorithm checks for convergence of the iterative process after each iteration. If the termination criterion is achieved, terminate the algorithm and output the best solution. Otherwise, setting $i = i + 1$ and, proceed to Step 3.

In this work, the termination criterion is selected as if the objective function does not change within 30 consecutive iterations, the optimization algorithm is terminated. The flowchart of the TLBO procedure is shown in Figure 3.

4 Simulation results analysis

The proposed path planning algorithm is successfully implemented in simulation. The goal of simulation is to show that the proposed path planning algorithm is able to find the near-optimal global offline obstacle-free path and to check the stability and performance of the proposed TLBO algorithm.

In the simulation, it is assumed that all the obstacles are already detected and that the vehicle's workspace is totally known. The vehicle's start point is $S_p = (200, 0)$ and the target point is $T_p = (700, 300)$. In addition, there are four detected obstacles in the workspace with known locations, while the safety distance for each obstacle is $d_{safe} = 30$ cm.

As can be seen from Figure 4, the proposed algorithm succeeded to generate a planned path with avoiding the

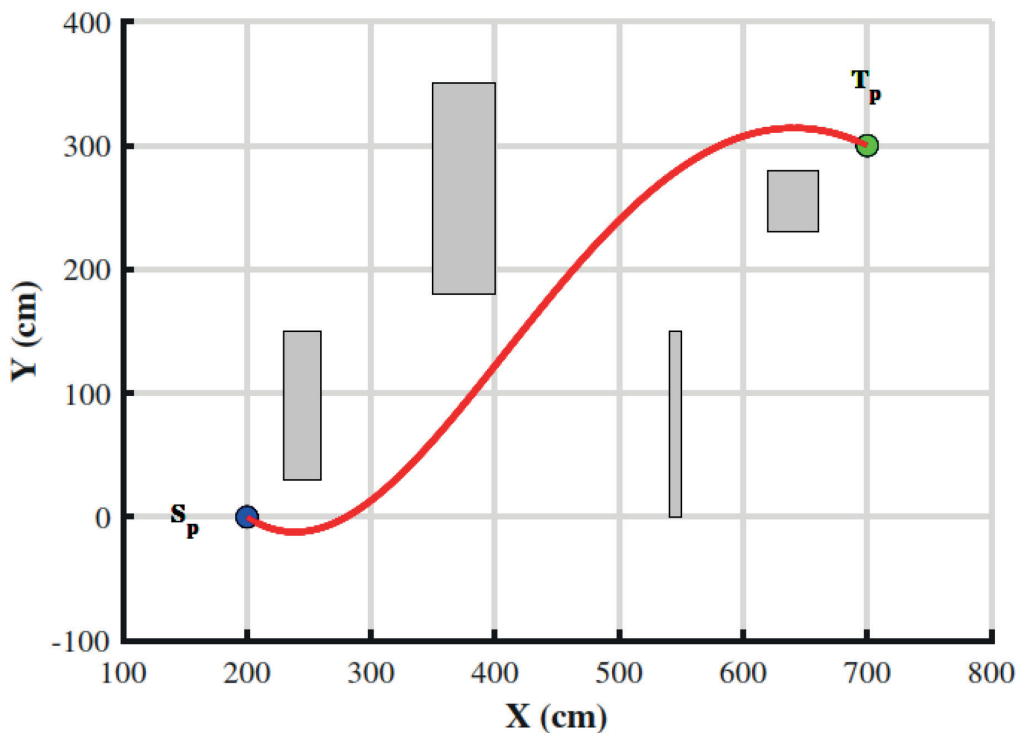


Figure 4 Path planning trajectory in XY plane during the simulation

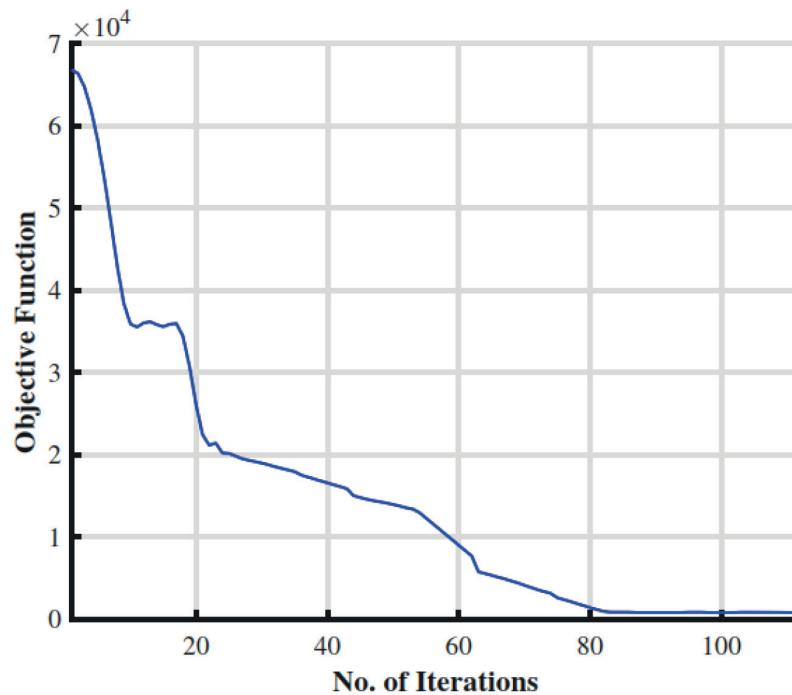


Figure 5 The objective function values history in terms of the number of iterations

Table 1 TLBO performance analysis

Average number of iterations	Average execution time (sec.)	Objective function			
		Worst	Best	Mean	Standard deviation
114	76	892	871	873	3.62

Table 2 TLBO parameters

Parameter	Value
Weighting factor (ζ)	2/3
Number of design variables (D_v)	2
Number of population (N_p)	100
Weighting value (w_1)	0.65
Weighting value (w_2)	0.35

collision with surrounding obstacles. In addition, Figure 5, shows that the objective function converges to the minimum value within an average of 84 iterations.

Since the TLBO is a meta-heuristic optimization technique and based on a random population in the first iteration, it is not guaranteed to find the global optimum solution. Therefore, it is of great importance to check its performance and stability. This can be obtained by executing the proposed algorithm many times to ensure that the obtained solution is near the optimal one. As can be seen from Table 1, the proposed algorithm is executed 100 times. It is clear that the objective function converges to the minimum value within an average of 114 iterations. In addition, from values of the objective function's standard deviation, it is clear that most of the values are close to the average. All the values of the TLBO parameters are given in Table 3.

5 Conclusions

This paper proposed a TLBO algorithm to provide a path planning method for an autonomous UGV in a cluttered environment. The suggested approach was successful in obtaining the near-optimal pathways, while avoiding nearby obstacles and it can manage system kinematic and algebraic restrictions. The TLBO algorithm is used to find the near-optimal path between the start and destination locations, taking into account path length minimization, path smoothness optimization and avoiding potential collisions with nearby obstacles. Many remarkable features are achieved by the proposed algorithm: 1) it is simple and can be modelled and simulated; and 2) using TLBO there is no need for adjustment of specific parameters such as GA, FFA, PSO and other meta-heuristic optimization techniques. Finally,

the simulations evaluate the proposed path planning algorithm in a cluttered environment, the simulations validate the effectiveness of the proposed path planning algorithm.

References

- [1] RAJA, P., PUGAZHENTHI, S. Optimal path planning of mobile robots: a review. *International Journal of Physical Sciences* [online]. 2012, **7**(9), p. 1314-1320. eISSN 1992-1950. Available from: <https://doi.org/10.5897/IJPS11.1745>
- [2] ZHANG, H., BUTZKE, J., LIKHACHEV, M. Combining global and local planning with guarantees on completeness. In: International Conference on Robotics and Automation ICRA: proceedings. 2012. p. 4500-4506.
- [3] PATLE, B., BABU L, G., PANDEY, A., PARHI, D. R. K., JAGADEESH, A. A review: on path planning strategies for navigation of mobile robot. *Defence Technology* [online]. 2019, **15**(4), p. 582-606. ISSN 2214-9147. Available from: <https://doi.org/10.1016/j.dt.2019.04.011>
- [4] INJARAPU, A. S., GAWRE, S. K. A survey of autonomous mobile robot path planning approaches. In: International Conference on Recent Innovations in Signal Processing and Embedded Systems RISE: proceedings [online]. IEEE. 2017. p. 624-628. Available from: <https://doi.org/10.1109/RISE.2017.8378228>
- [5] SCHWARTZ, J. T., SHARIR, M. On the “piano movers” problem I. the case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Communications on Pure and Applied Mathematics* [online]. 1983, **36**(3), p. 345-398. eISSN 1097-0312. Available from: <https://doi.org/10.1002/cpa.3160360305>
- [6] WEIGL, M., SIEMIATKOWSKA, B., SIKORSKI, K. A., BORKOWSKI, A. Grid-based mapping for autonomous mobile robot. *Robotics and Autonomous Systems* [online]. 1993, **11**(1), p. 13-21. ISSN 0921-8890. Available from: [https://doi.org/10.1016/0921-8890\(93\)90004-V](https://doi.org/10.1016/0921-8890(93)90004-V)
- [7] CHOSET, H., BURDICK, J. Sensor-based exploration: the hierarchical generalized voronoi graph. *The International Journal of Robotics Research* [online]. 2000, **19**(2), p. 96-125. ISSN 0278-3649, eISSN 1741-3176. Available from: <https://doi.org/10.1177/02783640022066770>
- [8] CHOSET, H., LYNCH, K., HUTCHINSON, S., KANTOR, G., BURGARD, W., KAVRAKI, L., THRUN, S. Principles of robot motion: theory, algorithms and implementation. Reviews. *The Knowledge Engineering Review* [online]. 2007, **22**(2), p. 209-211. ISSN 0269-8889, eISSN 1469-8005. Available from: <https://doi.org/10.1017/S0269888907218016>
- [9] KHATIB, O. Real-time obstacle avoidance for manipulators and mobile robots. In: *Autonomous Robot Vehicles* [online]. COX, I. J., WILFONG, G. T. (eds.). New York: Springer, 1986. p. 396-404. ISBN 978-0-387-97240-4, eISBN 978-1-4613-8997-2. Available from: <https://doi.org/10.1007/978-1-4613-8997-2>
- [10] SABIHA, A., KAMEL, M., SAID, E., HUSSEIN, W. Trajectory generation and tracking control of an autonomous vehicle based on artificial potential field and optimized backstepping controller. In: 12th International Conference on Electrical Engineering ICEENG: proceedings [online]. IEEE. 2020. p. 423-428. Available from: <https://doi.org/10.1109/ICEENG45378.2020.9171708>
- [11] MASEHIAN, E., AMIN-NASERI, M. A voronoi diagram-visibility graph-potential field compound algorithm for robot path planning. *Journal of Robotic Systems* [online]. 2004, **21**(6), p. 275-300. eISSN 1556-4967. Available from: <https://doi.org/10.1002/rob.20014>
- [12] CAI, K., WANG, C., CHENG, J., DE SILVA, C. W., MENG, M. Q.-H. Mobile robot path planning in dynamic environments: a survey [online]. arXiv preprint arXiv:2006.14195, 2020. Available from: <https://doi.org/10.15878/j.cnki.instrumentation.2019.02.010>
- [13] BHASKAR, B. S., RAUNIYAR, A., NATH, R., MUHURI, P. K. Zone-based path planning of a mobile robot using genetic algorithm. In: *Industry 4.0 and advanced manufacturing*. CHAKRABARTI, A., ARORA, M. (eds.). Singapore: Springer, 2021. ISBN 9789811556890, p. 263-275.
- [14] SONG, B., WANG, Z., ZOU, L. An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree bezier curve. *Applied Soft Computing* [online]. 2021, **100**, p. 1-11. ISSN 1568-4946.
- [15] LI, F., FAN, X., HOU, Z. A firefly algorithm with self-adaptive population size for global path planning of mobile robot. *IEEE Access* [online]. 2020, **8**, p. 168951-168964. eISSN 2169-3536. Available from: <https://doi.org/10.1109/ACCESS.2020.3023999>
- [16] XIN, D., HUA-HUA, C., WEI-KANG, G. Neural network and genetic algorithm based global path planning in a static environment. *Journal of Zhejiang University-Science A* [online]. 2005, **6**(6), p. 549-554. ISSN 1009-3095. eISSN 1862-1775. Available from: <https://doi.org/10.1007/BF02841763>
- [17] KHELCHANDRA, T., HUANG, J., DEBNATH, S. Path planning of mobile robot with neuro-genetic-fuzzy technique in static environment. *International Journal of Hybrid Intelligent Systems* [online]. 2014, **11**(2), p. 71-80. ISSN 1448-5869, eISSN 1875-8819. Available from: <https://doi.org/10.3233/HIS-130184>

- [18] CASTILLO, O., NEYOY, H., SORIA, J., MELIN, P., VALDEZ, F. A new approach for dynamic fuzzy logic parameter tuning in ant colony optimization and its application in fuzzy control of a mobile robot. *Applied Soft Computing* [online]. 2015, **28**, p. 150-159. ISSN 1568-4946. Available from: <https://doi.org/10.1016/j.asoc.2014.12.002>
- [19] WANG, X., SHI, Y., DING, D., GU, X. Double global optimum genetic algorithm-particle swarm optimization-based welding robot path planning. *Engineering Optimization* [online]. 2016, **48**(2), p. 299-316. ISSN 0305-215X, eISSN 1029-0273. Available from: <https://doi.org/10.1080/0305215X.2015.1005084>
- [20] RAO, R. V., SAVSANI, V. J., VAKHARIA, D. Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design* [online]. 2011, **43**(3), p. 303-315. ISSN 0010-4485. Available from: <https://doi.org/10.1016/j.cad.2010.12.015>
- [21] RAO, R. V. Teaching-learning-based optimization algorithm. In: *Teaching learning based optimization algorithm: and its engineering applications* [online]. RAO, R.V. (ed.). Cham: Springer, 2016. p. 9-39. ISBN 978-3-319-22731-3, eISBN 978-3-319-22732-0. Available from: <https://doi.org/10.1007/978-3-319-22732-0>
- [22] SAVSANI, P., JHALA, R. L., SAVSANI, V. J. Comparative study of different metaheuristics for the trajectory planning of a robotic arm. *IEEE Systems Journal*. 2014, **10**(2), p. 697-708. ISSN 1932-8184, eISSN 1937-9234.
- [23] RAO, R. V., WAGHMARE, G. Design optimization of robot grippers using teaching-learning-based optimization algorithm. *Advanced Robotics* [online]. 2015, **29**(6), p. 431-447. ISSN 0169-1864, eISSN 1568-5535. Available from: <https://doi.org/10.1080/01691864.2014.986524>
- [24] WU, Z., FU, W., XUE, R., WANG, W. A novel global path planning method for mobile robots based on teaching-learning-based optimization. *Information* [online]. 2016, **7**(39), p. 1-11. eISSN 2078-2489. Available from: <https://doi.org/10.3390/info7030039>
- [25] AOUF, A., BOUSSAID, L., SAKLY, A. TLBO-based adaptive neurofuzzy controller for mobile robot navigation in a strange environment. *Computational Intelligence and Neuroscience* [online]. 2018, **2018**, 3145436. ISSN 1687-5265, eISSN 1687-5273. Available from: <https://doi.org/10.1155/2018/3145436>
- [26] HERNANDEZ-BARRAGAN, J. Mobile robot path planning based on conformal geometric algebra and teaching-learning based optimization. *IFAC-PapersOnLine* [online]. 2018, **51**(13), p. 338-343. ISSN 2405-8963. Available from: <https://doi.org/10.1016/j.ifacol.2018.07.301>
- [27] KASHYAP, A. K., PANDEY, A. Optimized path planning for three-wheeled autonomous robot using teaching-learning-based optimization technique. In: *Advances in materials and manufacturing engineering*. LI, L., PRATIHAR, D. K., CHAKRABARTY, S., MISHRA, P. C. (eds.). Singapore: Springer, 2020. p. 49-57. ISBN 978-9811513060.
- [28] SINGH, G., SHARMA, N., SHARMA, H. Shuffled teaching learning-based algorithm for solving robot path planning problem. *International Journal of Metaheuristics*. 2020, **7**(3), p. 265-283. ISSN 1755-2176, eISSN 1755-2184.
- [29] MOHAMAD, S. A., KAMEL, M. A. Optimization of cylinder liner macro-scale surface texturing in marine diesel engines based on teaching-learning-based optimization algorithm. *Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology* [online]. 2021, **235**(2), p. 329-342. ISSN 1350-6501, eISSN 2041-305X. Available from: <https://doi.org/10.1177/1350650120911563>
- [30] RAO, R. V., SAVSANI, V. J., VAKHARIA, D. Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems. *Information Sciences* [online]. 2012, **183**(1), p. 1-15. ISSN 0020-0255. Available from: <https://doi.org/10.1016/j.ins.2011.08.006>
- [31] RAO, R., PATEL, V. An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. *International Journal of Industrial Engineering Computations* [online]. 2012, **3**(4), p. 535-560. ISSN 1923-2934, eISSN 1923-2926. Available from: <https://doi.org/10.5267/j.ijiec.2012.03.007>