# Evolutionary Dual Rule-based Fuzzy Path Planner for Omnidirectional Mobile Robot

Chansol Hong
School of Electrical Engineering
KAIST
Daejeon, Republic of Korea 34141
Email: cshong@rit.kaist.ac.kr

Chan Woo Park
School of Electrical Engineering
KAIST
Daejeon, Republic of Korea 34141
Email: cwpark@rit.kaist.ac.kr

Jong-Hwan Kim
School of Electrical Engineering
KAIST
Daejeon, Republic of Korea 34141
Email: johkim@rit.kaist.ac.kr

***Abstract*—Fuzzy navigator has been widely used in path planning for mobile robots because of its fast response. In this paper, evolutionary dual rule-based fuzzy path planner is proposed as a nobel path planner for omnidirectional mobile robots. The newly proposed fuzzy path planner utilizes a multiobjective evolutionary algorithm, which can take user's preference into account in path planning, for optimization of the path through scaling parameters used in fuzzification and defuzzification. The performance of evolutionary dual rule-based fuzzy path planner is evaulated through comparison of created paths and estimated Pareto-optimal fronts from dual multiobjective particle swarm optimization and dual multiobjective qauntum-inspired evolutionary algorithm. To show the validity of our path planner, path planning for omnidiretional mobile robot is simulated in MATLAB. Through the use of quantum-inspired evolutionary algorithm in Webots simulator, each unit movement of the omnidirectional mobile robot is optimized so that the non-holonomic motion of the mobile robot can be modified. The proposed evolutionary dual rule-based fuzzy path planner demonstrates the effectiveness through computer simulation for the omnidirectional mobile robot navigating in an environment with obstacles.**

## I. Introduction

Path planning with obstacle avoidance has been an important issue in mobile robots to be used in various environments. To be used in an environment with obstacles, path planning algorithms should generate a path to the target location without any collision. Different user preferences such as time consumption and minimal distance from the robot to an obstacle can be examined in evaluation of the created paths.

Some research has been conducted with variety of path planning algorithms in order to obtain an optimal path. Artificial potential field method, which assumes a goal exerting an attractive force and obstacles exerting repulsive force, has been proposed for creating a collision-free path through clear mathematical descriptions [1], [2]. Also, regarding the path planning problem as an optimization problem, genetic algorithms and neural networks based methods have been introduced in path planning and navigation problems [3]–[5]. Besides, the fuzzy logic has been introduced in these fields because of its fast response [6]–[9]. K. Park and N. Zhang introduced a dual rule fuzzy logic approach in path planning of a mobile robot [10]. The navigation with dual rule fuzzy logic utilizes two fuzzy rule tables instead of one and switches between two rules to navigate through an environment. In each step, only one rule table is selected and used in control of the velocities of the wheels. Also in their works, no optimization process has been taken place that optimality of the formed path is not guaranteed.

In this paper, evolutionary dual rule-based fuzzy path planner (EDrFPP) is proposed. As the subject mobile robot for simulation, an omnidirectional robot with three omnidirectional wheels is selected. Non-holonomic robots like two differential-wheeled mobile robot suffer from limitations in path planning. For example, the robot cannot make a sudden 90 degrees turn that it must change the direction gradually. Omnidirectional mobile robots are often assumed as holonomic on the 2-D plain environment. However, because of friction and slippery problems, omnidirectional mobile robots show non-holonomic behavior patterns that they even fail to track the path [11]. In this paper, the optimization of each unit movement of the omnidirectional mobile robot for reducing the effects of such problems in Webots environment with qunatum-inspired evolutionary algorithm (QEA) is also proposed.

In fuzzy logic theory, a measurement is represented with the degree of membership for each membership function [12]. Zavlangas *et al*. have shown that the path planning and obstacle avoidance of an omnidirectional mobile robot are possible through the use of fuzzy logic [13]. In their works, the fuzzy rule includes three inputs to control the robot: the distance between the robot and the nearest obstacle, the angle between current direction of the robot and the direction toward the nearest obstacle, and the angle between current direction of the robot and the direction toward the goal. Thus, the comprised rule base table is three dimensional, granting the proposed fuzzy logic path planner time complexity of at least $O(n^3)$ in consultaion with rule table before defuzzification.

The main contribution of our work is the development and validation of a novel fuzzy path planner EDrFPP that utilizes multiobjective evolutionary algorithm for optimization of scale parameters in fuzzification and defuzzification to take users preference into account in planning the path. Also, EDrFPP employs dual rule fuzzy logic approaches with simultaneous use of two rules and with different inputs and rules for path planning with omnidirectional mobile robot. The dimensionality of rule table is set to two that $O(n^2)$ time complexity in

978-1-5090-0626-7/16/$31.00 

TABLE I: Fuzzy rule table for navigation mode

| $\alpha_t$ / $\Delta\alpha_t$ | **NB** | **NM** | **NS** | **ZO** | **PS** | **PM** | **PB** |
|---|---|---|---|---|---|---|---|
| **NB** | | | | NB | NM | | |
| **NM** | | | | NM | | | |
| **NS** | | | | NS | ZO | | PM |
| **ZO** | NB | NM | NS | ZO | PS | PM | PB |
| **PS** | NM | | ZO | PS | | | |
| **PM** | | | | PM | | | |
| **PB** | | | PM | PB | | | |

TABLE II: Fuzzy rule table for obstacle avoidance mode

| $\alpha_t$ / $\beta_t$ | **NB** | **NM** | **NS** | **ZO** | **PS** | **PM** | **PB** |
|---|---|---|---|---|---|---|---|
| **NB** | NM | NS | ZO | ZO | | | |
| **NM** | NS | ZO | ZO | PS | | | |
| **NS** | ZO | ZO | PS | PM | | | |
| **ZO** | NB | NB | NB | PB | PB | PB | PB |
| **PS** | | | | NM | NS | ZO | ZO |
| **PM** | | | | NS | ZO | ZO | PS |
| **PB** | | | | ZO | ZO | PS | PM |

fuzzy rule output determination is guaranteed. Also, optimization of scale parameters in fuzzification and defuzzification through the use of multiobjective evolutionary algorithm is introduced to take user's preference into account in planning the path. EDrFPP is comprised of two different fuzzy rule base tables with dimensionality of two for each: navigation rule table, and obstacle avoidance rule table. EDrFPP is implemented and tested with two evolutionary algorithms for multi-objective optimization: dual multi-objective particle swarm optimization (DMOPSO), and dual multiobjective quantum-inspired evolutionary algorithm (DMQEA) [14], [15]. With DMOPSO and DMQEA, Pareto-optimal front of the path is estimated with user preferences such as time/energy consumption, safety, and smooth movement of the robot. To show the validity of the EDrFPP, an omnidirectional robot in an environment with static obstacles is simulated using MATLAB.

This paper is organized as follows. Section II introduces EDrFPP implementation. In Section III, brief introduction to multiobjective optimization algorithms used for optimization in EDrFPP and comparison between results from two optimization algorithm are provided. Section IV shows results of omnidirectional mobile robot odometry optimization and simulation results in the environment with static obstacles. Section V provides concluding remarks and future works.

## II. Evolutionary Dual Rule-Based Fuzzy Path Planner

As the term 'dual rule' suggests, EDrFPP is composed of two modes: navigation mode and obstacle avoidance mode. To determine the heading direction of the robot, the outputs of two modes are combined to a single output control signal.

### A. Navigation Mode

The navigation mode directs the robot toward the goal by taking two inputs. The first input is the angle $\alpha_t$ between current direction of the robot and the direction heading directly toward the goal. The angle is defined in the way that the goal located on the right side of current direction of the robot results in positive $\alpha_t$ and the goal located on the left side of current direction of the robot results in negative $\alpha_t$. The second input is the time difference of the first input defined as

$$\Delta\alpha_t = (\alpha_{t-1} - \alpha_t)/\Delta t. \tag{1}$$

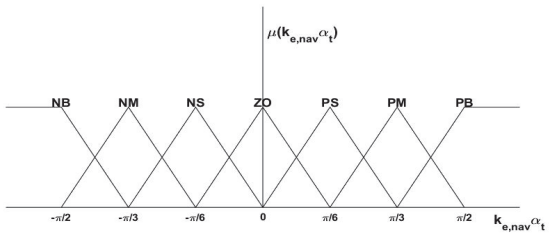
These two inputs, $\alpha_t$ and $\Delta\alpha_t$, are scaled by $k_{e,nav}$ and $k_{ce,nav}$, respectively, and are fuzzified with membership functions shown in Fig. 1. Triangular membership functions are selected so that the fuzzified value changes smoothly as the input value changes.

Then, defuzzification is taken place using fuzzy rule table shown in Table I. The rules are heuristically set in a way that the robot approaches the goal in a smooth motion. For instance, when large direction change to the left must be made in order to head toward the goal, which is on the right side of the robot, the direction change may be too large that the robot may head toward the right side of the goal if the angular velocity of direction were already given leftward to head toward the goal. Limiting the maximum possible direction change based on the angular velocity of current state would prevent the robot from overshooting the goal. The defuzzified output is scaled by $k_{u,nav}$ before updating the direction of the robot. The three scale parameters $k_{e,nav}$, $k_{ce,nav}$, and $k_{u,nav}$ are to be optimized for proper operation of EDrFPP.

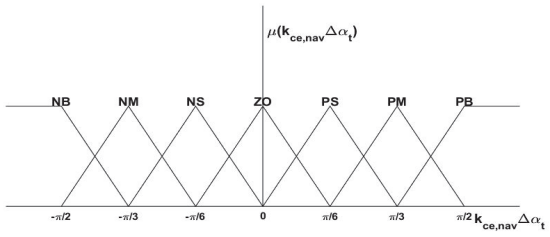
### B. Obstacle Avoidance Mode

The obstacle avoidance mode makes the robot avoid a nearest obstacle while keeping the general direction toward the goal by taking two inputs. The first input is the angle $\alpha_t$ between current direction of the robot and the direction heading directly toward the goal. The second input is the angle $\beta_t$ between the current direction of the robot and the direction of the nearest obstacle.

Similar to the navigation mode, two inputs are scaled by $k_{e,obs}$ and $k_{ce,obs}$, respectively, and fuzzified with membership functions shown in Fig. 2. Again, triangular membership

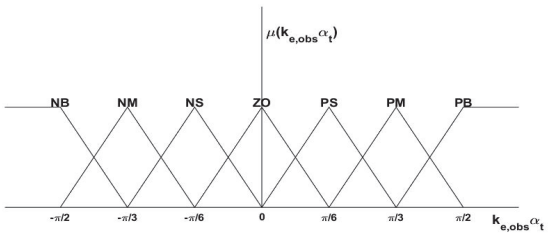



(a)

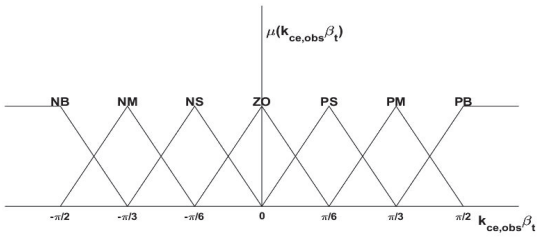



(b)

Fig. 1: Membership functions for inputs of navigation mode




(a)




(b)

Fig. 2: Membership functions for inputs of obstacle avoidance mode

functions are selected so that the fuzzified value changes smoothly as the input value changes. Note that $\alpha_t$ is multiplied by a different scale parameter from the navigation mode for granting different level of delicacy for each mode.

Then, defuzzification is taken place using fuzzy rule table shown in Table II. The rules are heuristically set in a way that the robot does not deviate significantly from the general direction toward the goal. For instance, when the nearest obstacle is directly in front of the heading direction and the goal is on the left, the robot would avoid the obstacle by turning to the left. The defuzzified output is scaled by $k_{e,obs}$, $k_{ce,obs}$, and $k_{u,obs}$, are also the targets for optimization.

### C. Output Determination

In determination of the output control signal, outputs of two modes are scaled and the robot and are combined. To scale the output, a scale parameter and the distance $d_t$ between a nearest obstacle and the robot are used. In locating the nearest obstacle, obstacles located behind the robot are not considered. The equation for the output control signal is as follows:

$$\gamma_{t+1} = \left(1 - \frac{k_v}{k_v + d_t}\right) k_{u,nav} o_{nav} + \left(\frac{k_v}{k_v + d_t}\right) k_{u,obs} o_{obs} \quad (2)$$

where $\gamma_{t+1}$ is the robot heading direction in the next time step, $o_{nav}$ is the defuzzified output of navigation mode, $o_{obs}$ is the defuzzified output of obstacle avoidance mode, and $k_v$ is the scale parameter. As the distance between the robot and the nearest obstacle becomes smaller, the second term on the right hand side of the equation would dominate over the first term so that the control would focus more on avoiding imminent obstacle rather than heading toward the goal. The scale parameter $k_v$ determines controllers responsiveness to the nearest obstacle and is a target parameter to be optimized.

With next heading direction determined, the robot proceeds to that direction by one time step and the same procedure is repeated until the robot reaches the goal or meets the failure condition such as time limit or collision. This procedure is repeated with different scale parameters obtained from a multi-objective evolutionary algorithm to generate a path that is efficient in terms of user's preference. Algorithm 1 summarizes the navigation procedure of the omnidirectional robot in proposed EDrFPP that will be repeated in order to optimize the scale parameters with two multiobjective evolutionary algorithms explained in the following section to generate preferable paths.

## III. Optimization Algorithms

EDrFPP utilizes seven scale parameters ($k_{e,nav}$, $k_{ce,nav}$, $k_{u,nav}$, $k_{e,obs}$, $k_{ce,obs}$, $k_{u,obs}$, and $k_v$), which were defined in the previous section, that should be optimized through the use of an optimization algorithm to generate preferable paths. In this section, two optimization algorithms, dual multi-objective particle swarm optimization (DMOPSO) [14] and dual multiobjective quantum-inspired evolutionary algorithm (DMQEA) [15], used in the experiment of EDrFPP are briefly introduced. Besides, comparison between results from two optimization algorithms is also presented.

**Algorithm 1** Navigation in EDrFPP

**procedure** NAVIGATION IN EDRFPP
  Initialize robot position $x_0$, $y_0$
  Calculate robot-goal distance $Dist_0$
  Calculate robot-nearest obstacle distance $d_0$
  **for** $i = 0 \rightarrow$ max iterations **do**
    $in_{1,nav} \leftarrow \text{fuzzify}(k_{e,nav}\alpha_t)$
    $in_{2,nav} \leftarrow \text{fuzzify}(k_{ce,nav}\Delta\alpha_t)$
    $o_1 = k_{u,nav}\text{defuzzify}(in_{1,nav}, in_{2,nav})$
    **if** *obstacle present* **then**
      $in_{1,obs} \leftarrow \text{fuzzify}(k_{e,obs}\alpha_t)$
      $in_{2,obs} \leftarrow \text{fuzzify}(k_{ce,obs}\Delta\beta_t)$
      $o_2 \leftarrow k_{u,obs}\text{defuzzify}(in_{1,obs}, in_{2,obs})$
      $\gamma_{t+1} = \left(1 - \frac{k_v}{k_v + d_t}\right) o_1 + \left(\frac{k_v}{k_v + d_t}\right) o_2$
    **else**
      $\gamma_{t+1} = o_1$
    **end if**
    $x_{t+1} \leftarrow x_t + v\Delta t \cos(\gamma_{t+1})$
    $y_{t+1} \leftarrow y_t + v\Delta t \sin(\gamma_{t+1})$
    **if** *collision* **then**
      **return** *failure*
    **end if**
    $Dist_{t+1} \leftarrow \sqrt{x_{t+1}^2 + y_{t+1}^2}$
    **if** $Dist_{t+1} < 0.1$ **then**
      **return** *success*
    **end if**
  **end for**
**end procedure**

### A. DMOPSO

DMOPSO is a multiobjective evolutionary algorithm that inherits the idea of particle swarm optimization (PSO) [16].

PSO employs a population composed of N particles $p_k$ where $k = 1, 2, ..., N$ with individual velocity $\boldsymbol{v_k}$ and position $\boldsymbol{x_k}$ that are initially random in M-dimensional search space. As generation passes, each particle propagates through the search space with their individual velocity, updating their positions. At each position, the objective function value is evaluated and personal best position ${}^p\boldsymbol{x}_{\boldsymbol{k}}^t$, where the fitness value is largest among all positions that the particle has been located in the past, is stored. Then, the global best position ${}^g\boldsymbol{x}_{\boldsymbol{k}}^t$, which is defined as the best position among the personal best positions of neighboring particles of $p_k$, is stored. Finally, the velocity and position of each particle are updated with following equation:

$$\begin{cases} \boldsymbol{v}_{\boldsymbol{k}}^{t+1} = c_1\boldsymbol{v}_{\boldsymbol{k}}^t + U(0,c_2)\,({}^p\boldsymbol{x}_{\boldsymbol{k}}^t - \boldsymbol{x}_{\boldsymbol{k}}^t) + U(0,c_2)\,({}^g\boldsymbol{x}_{\boldsymbol{k}}^t - \boldsymbol{x}_{\boldsymbol{k}}^t) \\ \boldsymbol{x}_{\boldsymbol{k}}^{t+1} = \boldsymbol{x}_{\boldsymbol{k}}^t + \boldsymbol{v}_{\boldsymbol{k}}^{t+1} \end{cases} \tag{3}$$

where $c_1$ and $c_2$ are constants and $U(0, c_2)$ is a uniform random number. The particles continue propagation until a termination criterion is met.

The prefix dual multi-objective of DMOPSO indicates that there exist multiple objectives for optimization and two stages of optimization procedure. In the first stage, fitness values $f_i$ for each objective $i$ are obtained for each particles current position $\boldsymbol{x}_{\boldsymbol{k}}^t$. Then, the particles dominated in terms of fitness values are culled out. With nondominated solutions from the first stage, second stage performs nondominated sorting using two objectives: global evaluation value (GEval) and crowding distance (CD). The calculation of GEval takes user's preference into account and outputs single evaluation from fitness values used in the first stage through performing Choquet fuzzy integral [17], [18]. The calculation of CD is done by estimating the largest cuboid a solution can assume in the solution space without enclosing any other solutions [19]. The detailed procedure and structure of DMOPSO are described in [14].

TABLE III: Parameters for DMOPSO and DMQEA

| Parameters | DMOPSO | DMQEA |
|---|---|---|
| No. of runs | 50 | 50 |
| The population size (n·s) | 200 | 200 |
| No. of generations | 400 | 400 |
| Subpopulation size (n) | 50 | 50 |
| No. of subpopulations (s) | 4 | 4 |
| No. of multiple observation | 5 | *N/A* |
| The rotation angle | $0.20\pi$ | *N/A* |

TABLE IV: Average nondominated solutions for two metrics

| | DMOPSO | DMQEA |
|---|---|---|
| $\mathcal{D}$ | $6.532 \times 10^8$ | $3.520 \times 10^6$ |
| $\mathcal{S}$ | 17.120 | 13.154 |

### B. DMQEA

Similar to DMOPSO, DMQEA is a multiobjective evolutionary algorithm that inherits the idea of QEA [20].

QEA employs the concepts of quantum computing such as superposition of states in a qunatum bit. In QEA, each Q-bit individual in a population is represented with multiple Q-bits, which can generate different sequences of binary bits probabilistically through multiple observations on the quantum bits that genetic diversity is realized. As generation passes, probability of each Q-bit generating a binary bit of 0 or 1 changes so that more optimal sequences of binary bits are produced.

The prefix dual multi-objective has identical meaning with DMOPSO explained above. Multiple objectives are to be optimized in the first stage and global evaluation value and crowding distance are calculated in the second stage to maintain the spread of global best individuals on the search space and proximity to Pareto-optimal front at the same time. The detailed procedure and structure of DMQEA are described in [15].

### C. Performance Metrics

To demonstrate the functionality of EDrFPP with DMOPSO and DMQEA, we set three objectives as follows:

TABLE V: Hypothesis test on two metrics

| $\mathcal{H}_0 : \bar{\mathcal{D}}_{\text{DMOPSO}} - \bar{\mathcal{D}}_{\text{DMQEA}} = 0$ | | |
|---|---|---|
| $t$-value ($p$-value) | Reject | $\mathcal{H}_1$ |
| 1.395(0.169) | NO | N/A |

| $\mathcal{H}_0 : \bar{\mathcal{S}}_{\text{DMOPSO}} - \bar{\mathcal{S}}_{\text{DMQEA}} = 0$ | | |
|---|---|---|
| $t$-value ($p$-value) | Reject | $\mathcal{H}_1$ |
| $10.013(9.631 \times 10^{-14})$ | YES | $\bar{\mathcal{S}}_{\text{DMOPSO}} - \bar{\mathcal{S}}_{\text{DMQEA}} > 0$ |

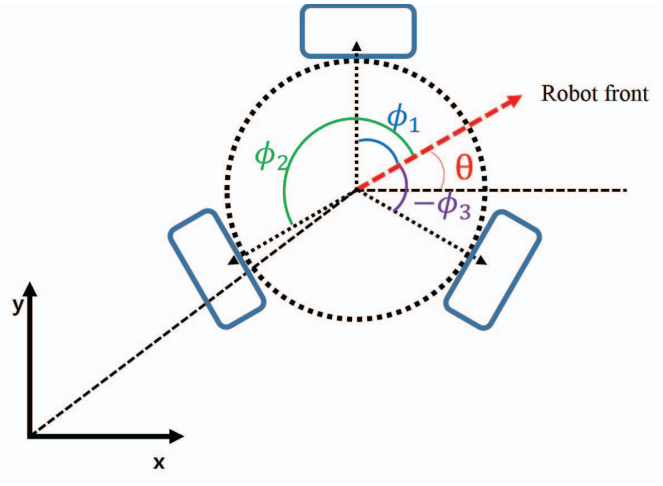



Fig. 3: Omnidirectional Mobile Robot Schematics

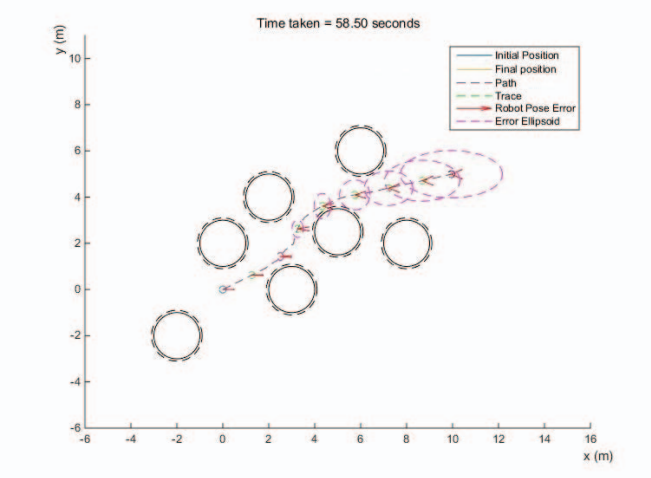
*1) Time Consumption ($f_1$):* The first objective is the time taken for the robot to reach the goal. As we set the velocity of the robot be a constant, this objective also indicates the total distance travelled for the robot to reach the goal. As robot should reach the goal fast, this objective should be minimized.

*2) Proximity to Obstacle ($f_2$):* The second objective is the minimum distance between the robot and the nearest obstacle along the path. Since there can exist some deviation from the formed path in real environment due to the robot not being ideal, keeping some distance from obstacles is preferred. As a larger value of this objective indicates that the robot has travelled along a safer path, this objective should be maximized.

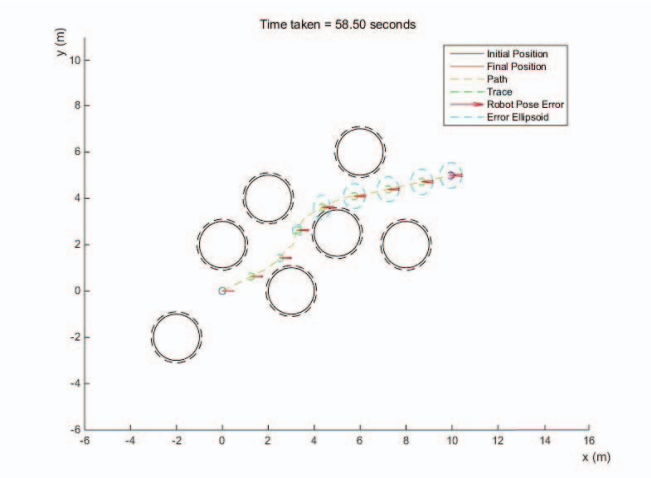
*3) Sudden direction changes ($f_3$):* The third objective is the number of turns made by robot along the path that is greater than 60 degrees. As sudden direction changes would cause the robot deviate from the formed path in real environment due to inertia and frictions with the floor, minimizing the number of sudden direction changes is preferred. Thus, this objective should be minimized.

To perform a fair comparison between DMOPSO and DMQEA, preference degrees for both multiobjective optimization algorithms be $f_1$:$f_2$:$f_3$ = 10:1:10. The normalized weights corresponding to the preference degrees were (0.4762, 0.0476, 0.4762). In Table III, note that the parameters for two optimization algorithms are set equally for fair comparison.

To compare the results from two multiobjective optimization




(a) Without optimization



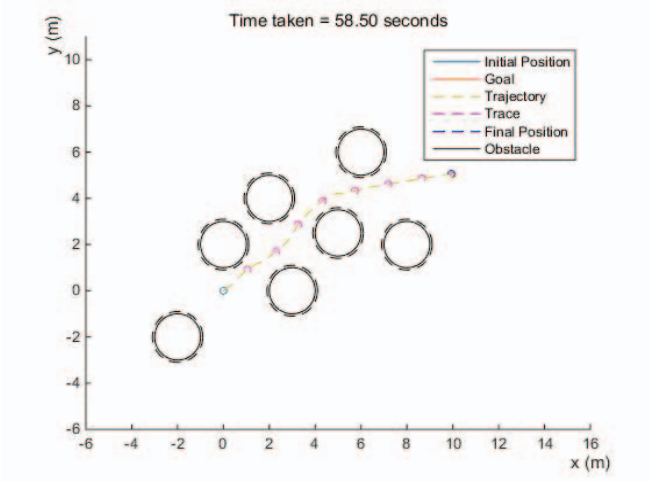

(b) With QEA optimization

Fig. 4: Odometry Error Comparison

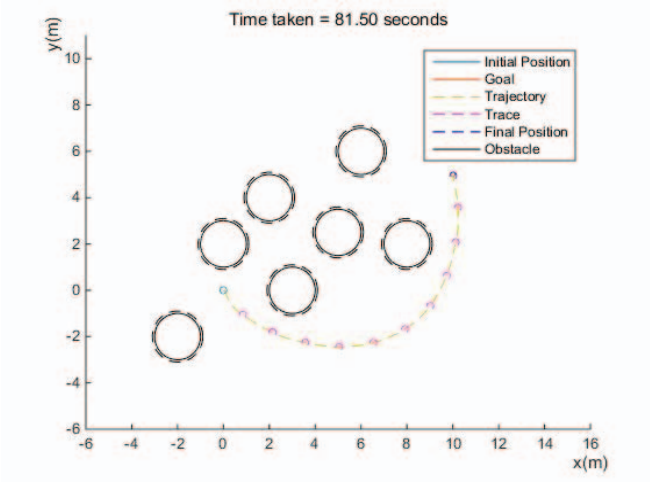
algorithms, two performance metrics are employed [21]. The diversity $\mathcal{D}$ measures the spread of nondominated solutions as equation follows [22].

$$\mathcal{D} = \frac{\sum_{k=1}^{n}(f_k^{(max)} - f_k^{(min)})}{\sqrt{\frac{1}{|N_0|}\sum_{i=1}^{|N_0|}(z_i - \bar{z})^2}} \tag{4}$$

where $N_0$ is the nondominated solution set, $z_i$ is the minimum distance between the i-th solution and the nearest solution, and $\bar{z}$ is the mean of all $z_i$. $f_k^{(max)}$ and $f_k^{(min)}$ are maximum and minimum fitness values of $k$-th objective, respectively. The size of dominated space $\mathcal{S}$ is defined as the hypervolume of the nondominated solutions at the end of optimization procedure. A large hypervolume would indicate that the region dominated by the solutions set is large that the quality of the set is high. The diversity and hypervolume of two multiobjective optimization algorithms are summarized in Table IV. The results are averaged over 50 runs.

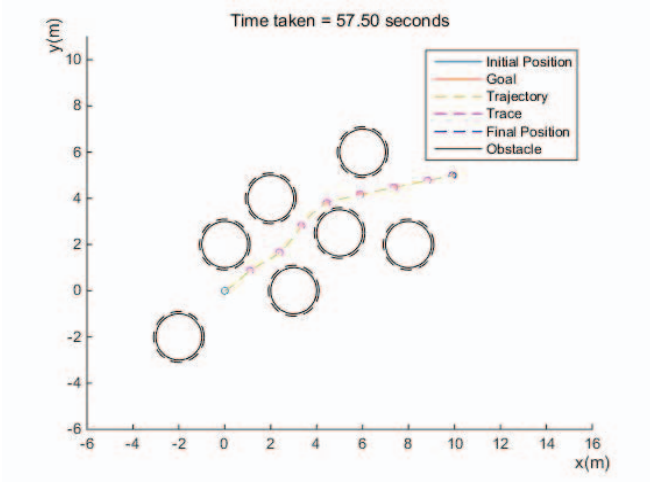



(a) Minimized time consumption

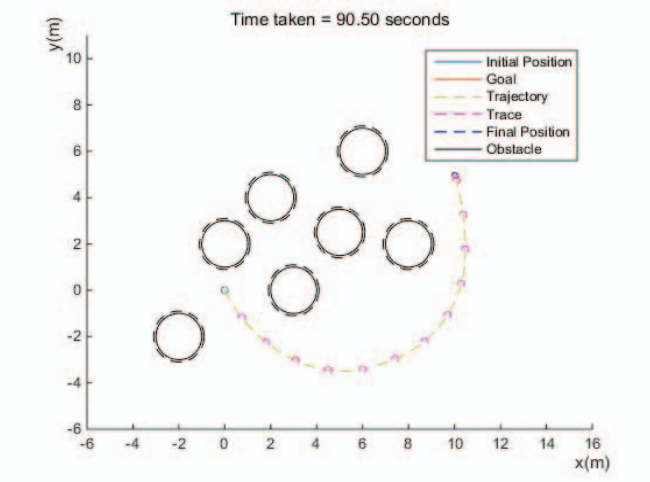



(b) Maximized minimal distance between the robot and obstacles

Fig. 5: Two samples of paths by DMOPSO




(a) Minimized time consumption




(b) Maximized minimal distance between the robot and obstacles

Fig. 6: Two samples of paths by DMQEA

Also, *t*-test was conducted in order to analyze the calculated performance metrics of two algorithms statistically. In the *t*-test, a null hypothesis, $\mathcal{H}_0$, which hypothesizes that there exists no significant difference in the result of two algorithms, is set and check if the hypothesis is supported by *p*-value obtained from the test. If *p*-value falls below the significance level, which was set 0.05 for this case, the null hypothesis is rejected and the alternative hypothesis $\mathcal{H}_1$ is supported. The two-tailed *t*-test was conducted to check whether one algorithm was superior to other algorithm in terms of diversity $\mathcal{D}$ and the size of dominated space $\mathcal{S}$.

Table V summarizes the hypothesis test results. As can be seen, the diversity measurements of DMOPSO and DMQEA do not differ significantly that null hypothesis is assumed. On the other hand, the hypervolume measurements show that the results from DMOPSO and DMQEA are significantly different from each other that the alternative hypothesis of the hypervolume of the result of DMOPSO significantly larger than that of DMQEA. However, the hypervolume metric may be misleading. Unlike well-known test functions such as DTLZ functions for performance comparison of multiobjective algorithms, navigation toward the goal while avoiding obstacles is a multiobjective problem with ideal fitness not known in advance [23]. Thus, the unknown Pareto-optimal front may not be convex that the hypervolume metric may not compare the results from two algorithms well.

## IV. Simulation

To test EDrFPP in a non-ideal environment, omnidirectional mobile robot was implemented and its unit movement was optimized using QEA in Webots environment. Besides, paths generated from EDrFPP using DMOPSO and DMQEA are visualized using MATLAB.

### A. Omnidirectional Mobile Robot

There have been some approaches to realize 2-D holonomic mobile robot. K.Nagatani *et al*. tried to improve odometry for

omnidirectional vehicle using optical flow information [11]. They argued that omnidirectional mobile robot fails to reach the goal because of the wheel slippage and tried to find parameters which affect the wheel's rotational direction with various surface conditions real experiments and optical flow information.

In this paper, we constrained the environment to single surface. Because of technological improvements in simulation tools, repeated simulations with different parameters for checking odometry errors can be done in short time with less efforts. Omnidirectional mobile robot whose schematic shown in Fig. 3 was implemented in Webots. Using Webots environment instead of using real experiments, each unit movement of the robot is optimized using QEA. In the omnidirectional mobile robot, each of three wheels are located perpendicular to the robot's center of gravity, where $i$-th wheel is located $\phi_i$ degrees away from the front of the robot. From the dynamics of the robot, translational and angular velocities of the robot are related as follows:

$$\begin{bmatrix}\varphi_1\\ \varphi_2\\ \varphi_3\end{bmatrix} = \frac{1}{r}\begin{bmatrix}-sin(\theta+\phi_1) & cos(\theta+\phi_1) & L\\ -sin(\theta+\phi_2) & cos(\theta+\phi_2) & L\\ -sin(\theta+\phi_3) & cos(\theta+\phi_3) & L\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} \quad (5)$$

or in short:

$$[\varphi_i] = \frac{1}{r}[\boldsymbol{W}][\dot{S}_i]. \quad (6)$$

The variable $\varphi_i$ means the angular velocity of $i$-th wheel which are attained from shaft encoder. $\dot{x}$, $\dot{y}$, and $\dot{\theta}$ are the differential value of 2-D domain coordinates and differential angle of the robot body to global domain. L and r are the radius of the robot body and the radius of a wheel, respectively. With direct method, odometry of the robot can be summarized as follows [24]:

$$[\dot{S}_i] = r[\boldsymbol{W}]^{-1}[\varphi_i] \quad (7)$$

$$[S_i] = r\int_0^t [\boldsymbol{W}]^{-1}[\varphi_i]dt. \quad (8)$$

Using (8), position and orientation of the robot can be obtained. However, because of the slippery problems, odometry error is accumulated as the robot proceeds with the driving data. With the robot having three wheels, three parameters are needed. As can be seen in (6) and (7), the translational velocity vector and the angular velocity vector can be converted to each other, applying parameters to $[S_i]$ would have similar effect with applying parameters to $[\varphi_i]$. As the translational matrix $[\boldsymbol{W}]$ is dependent on the direction of the front of the robot, $\theta$, the parameters should be obtained for each movement direction. In this simulation, we obtained parameters for 360 equally spaced angles from 0 to 359 degrees.

$$[S_i] + [\rho_i] = r\int_0^{unit\ time} [\boldsymbol{W}]^{-1}[\varphi_i]dt \quad (9)$$

The parameters $\rho_i$ are optimized using QEA. The fitness is calculated with Euclidean distance error and robot pose error as

$$fitness = \sqrt{error(x)^2 + error(y)^2} + |error(\theta)|. \quad (10)$$

As the error in the angle would result in larger final error than the error in the position, robot pose error in the fitness function was in degrees to have relatively larger effect than Euclidean distance error. In Fig. 4, the error ellipsoids without and with compensation parameters applied are shown, where the robot moves 11.7 m while the robot body radius is 0.11 m. As can be seen, applying the compensation parameters resulted in the reduction of error ellipsoid, particularly in robot pose error.

### B. Simulation Results

From the nondominated sets of DMOPSO and DMQEA, paths generated from two algorithms are plotted in MATLAB, shown in Figs. 5 and 6. The solutions have one objective that is best among the solutions from the set. Since the solution containing best third objective value, the number of sudden direction changes, coincided with the solution containing best first objective value, time consumption, only two paths are plotted per algorithm. As can be seen, paths formed by EDrFPP direct the robot toward the goal while avoiding obstacles. Also, both DMOPSO and DMOQEA produce wide range of paths that can be selected based on user preferences [17].

## V. Conclusion

In this paper, evolutionary dual rule-based fuzzy path planner (EDrFPP) was proposed by simultaneous use of two different rule base tables for navigation toward the goal and obstacle avoidance and optimization through multiobjective evolutionary algorithms. The contributions of outputs of two modes for updating the robots heading direction were determined based on the distance between the robot and the nearest obstacle. To take user's preference into account in path planning, multiobjective evolutionary algorithms DMOPSO and DMQEA were used to optimize seven scale parameters and the result of two algorithms were compared. The resulting paths plotted on MATLAB showed that the robot was able to reach the goal while satisfying users preference. Then, the formulated paths were simulated using Webots simulator. To reduce the error due to slippery effect and friction, compensation parameters for three omnidirectional wheels were optimized by QEA. The results verified that the error ellipsoid was reduced that more accurate path tracking was achievable.

## Acknowledgment

This work was supported by the Technology Innovation Program, 10045252, Development of robot task intelligence technology, supported by the Korea MOTIE.

## References

[1] Y.-J. Kim, J.-H. Kim, and D.-S. Kwon, "Evolutionary programming-based univector field navigation method for past mobile robots," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 31, no. 3, pp. 450–458, 2001.

[2] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on robotics and automation*, vol. 16, no. 5, pp. 615–620, 2000.

[3] J. Tu and S. X. Yang, "Genetic algorithm based path planning for a mobile robot," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 1, pp. 1221–1226, IEEE, 2003.
[4] D. Xin, C. Hua-hua, and G. Wei-kang, "Neural network and genetic algorithm based global path planning in a static environment," *Journal of Zhejiang University Science A*, vol. 6, no. 6, pp. 549–554, 2005.
[5] Y.-D. Hong, Y.-H. Kim, J.-H. Han, J.-K. Yoo, and J.-H. Kim, "Evolutionary multiobjective footstep planning for humanoid robots," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 41, no. 4, pp. 520–532, 2011.
[6] M. a. Sugeno and M. Nishida, "Fuzzy control of model car," *Fuzzy sets and systems*, vol. 16, no. 2, pp. 103–113, 1985.
[7] A. Saffiotti, "The uses of fuzzy logic in autonomous robot navigation," *Soft Computing*, vol. 1, no. 4, pp. 180–197, 1997.
[8] J.-K. Yoo and J.-H. Kim, "Fuzzy integral-based gaze control architecture incorporated with modified-univector field-based navigation for humanoid robots," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 1, pp. 125–139, 2012.
[9] J.-K. Yoo and J.-H. Kim, "Gaze control-based navigation architecture with a situation-specific preference approach for humanoid robots," *Mechatronics, IEEE/ASME Transactions on*, vol. 20, no. 5, pp. 2425–2436, 2015.
[10] K. Park and N. Zhang, "Behavior-based autonomous robot navigation on challenging terrain: A dual fuzzy logic approach," in *Foundations of Computational Intelligence, 2007. FOCI 2007. IEEE Symposium on*, pp. 239–244, IEEE, 2007.
[11] K. Nagatani, S. Tachibana, M. Sofne, and Y. Tanaka, "Improvement of odometry for omnidirectional vehicle using optical flow information," in *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 1, pp. 468–473, IEEE, 2000.
[12] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.
[13] P. G. Zavlangas, S. G. Tzafestas, and K. Althoefer, "Fuzzy obstacle avoidance and navigation for omnidirectional mobile robots," in *European Symposium on Intelligent Techniques, Aachen, Germany*, pp. 375–382, Citeseer, 2000.
[14] K.-B. Lee and J.-H. Kim, "Dmopso: Dual multi-objective particle swarm optimization," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pp. 3096–3102, IEEE, 2014.
[15] S.-J. Ryu, R. Datta, and J.-H. Kim, "Dual multiobjective quantum-inspired evolutionary algorithm for a sensor arrangement in a 2d environment," in *Robot Intelligence Technology and Applications 3*, pp. 57–66, Springer, 2015.
[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942–1948, IEEE, 1995.
[17] J.-H. Kim, J.-H. Han, Y.-H. Kim, S.-H. Choi, and E.-S. Kim, "Preference-based solution selection algorithm for evolutionary multiobjective optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 16, no. 1, pp. 20–34, 2012.
[18] K.-B. Lee and J.-H. Kim, "Multiobjective particle swarm optimization with preference-based sort and its application to path following footstep optimization for humanoid robots," *Evolutionary Computation, IEEE Transactions on*, vol. 17, no. 6, pp. 755–766, 2013.
[19] C. R. Raquel and P. C. Naval Jr, "An effective use of crowding distance in multiobjective particle swarm optimization," in *Proceedings of the 7th Annual conference on Genetic and Evolutionary Computation*, pp. 257–264, ACM, 2005.
[20] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 6, pp. 580–593, 2002.
[21] E. Zitzler, *Evolutionary algorithms for multiobjective optimization: Methods and applications*, vol. 63. Citeseer, 1999.
[22] H. Li, Q. Zhang, E. Tsang, and J. A. Ford, "Hybrid estimation of distribution algorithm for multiobjective knapsack problem," in *Evolutionary Computation in Combinatorial Optimization*, pp. 145–154, Springer, 2004.
[23] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the Congress on Evolutionary Computation (CEC-2002),(Honolulu, USA)*, pp. 825–830, Proceedings of the Congress on Evolutionary Computation (CEC-2002),(Honolulu, USA), 2002.
[24] A. Samani, A. Abdollahi, H. Ostadi, and S. Z. Rad, "Design and development of a comprehensive omni directional soccer player robot," *International Journal of Advanced Robotic Systems*, vol. 1, no. 3, pp. 191–200, 2004.