IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 July 2023, accepted 19 July 2023, date of publication 24 July 2023, date of current version 1 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3298553

RESEARCH ARTICLE

Adaptive Fuzzy Inference Decision Strategy for Service Robots Tidying Up Objects

YU-TING SU, TZUU-HSENG S. LI, (Member, IEEE), EN-HAUH LIU, HSU-MING TSAO, YU-HSIUNG WU, AND CHENG-YEH YANG

aiRobots Laboratory, Department of Electrical Engineering, National Cheng Kung University, Tainan City 70101, Taiwan

Corresponding author: Tzuu-Hseng S. Li (thsli@mail.ncku.edu.tw)

This work was supported by the National Science and Council, Taiwan, under Grant MOST 111-2221-E-006-190-MY2.

ABSTRACT Tidying up objects poses a complex challenge for service robots, particularly when it involves scheduling operations that require real-time action for achieving optimal results. This paper presents a decision system aimed at selecting priority locations to address the problem of robots searching for miscellaneous items in indoor environments. Initially, three datasets with varying complexities are generated using real environmental data, and the environmental information is integrated into feature matrices. Subsequently, the rule base of the fuzzy inference system is trained and optimized using an equilibrium optimizer, and its performance is compared with other commonly used algorithms. The feature matrices and the optimized rule base are then incorporated into the fuzzy inference system, leveraging the traveling salesman problem to determine the optimal sequence for visiting locations. The accuracy and efficiency of the proposed method are validated through physical implementation tests using analog data. Finally, the system is applied and evaluated in real-world scenarios to validate its effectiveness. The video of experiment is available at https://youtu.be/rjMDgopM-9M.

INDEX TERMS Autonomous mobile robot, equilibrium optimizer, fuzzy inference system, tidying up object problem, travelling salesman problem.

I. INTRODUCTION

Tidying up objects is a challenging task for robots as it requires the integration of multiple technologies such as environment recognition, object identification, SLAM, kinematics, power distribution, and visual identification [1]. The task demands the efficiency of robots' overall performance. Recently, several studies have conducted research on robotic tidy-up technology. For instance, the Human Support Robot (HSR) has been proposed as a solution to household problems. Literature [2] have demonstrated how the HSR can accomplish tasks in various environments, explained robot mechanisms, and described the results of using HSR during the competition of RoboCup@Home, highlighting the growing need for robots in social life. Reference [3] used quantifiable hierarchical methods to analyze man-defined actions and make simple and effective decisions for easy to complex sorting tasks. Similarly, another literature [4] proposed a model general software structure for Automatic Mobile Manipulator Robots. These robots can smoothly navigate narrow indoor spaces, use complex human voice commands to interact with humans, independently detect objects, and accomplish various home tasks. In [5], the use of HSR applications, through visual recognition, object manipulation, and motion planning technology, was described to achieve home sorting tasks.

In addition to the application of HSR in sorting sundries, scholars also considered the geometric shape and color information of objects together to split and align various objects in the point cloud. The principal components analysis (PCA) scheme [6], [7] was adopted to estimate the object frame and its placement status so as to find the objects that might need to be sorted by strategy. Literature [8] put forward a robot system of automatic sorting of tableware for tidying up tableware in cafeterias which have large amounts of tableware. The focus of [8] is to classify and collect the

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaojie Su.

tableware placed on the tray detected by an RGB-D camera. The effectiveness of the prototype robot and the auxiliary rotating machine was verified by experiments during which two kinds of rotating machines grasped tableware, discarded leftovers, and collected multiple tableware, while multiple robots sorted overlapped tableware.

Numerous studies and applications have been carried out on home organizing. However, if the number of objects to be sorted is too high, robots may need to repeatedly retrieve miscellaneous items and carry them to their designated location, resulting in significant electricity consumption and increased time. To address this issue, this study utilizes both self-made and commercial robots equipped with baskets as mobile carriers.

The fuzzy inference method, proposed by Professor Zadeh, transforms human experience and cognition into fuzzy judgment rules. This is done by quantifying membership degrees with mathematical functions, which ultimately deduce the size of the control quantity. Fuzzy set theory has become increasingly popular in intelligent systems due to its simplicity and resemblance to human reasoning. Recent research has shown that technology integration proposed by many scholars performs well. The rule base is defined according to the relationship between the specification set and language label, and a fuzzy knowledge map is generated. Machine learning experiments using this approach show that it effectively reduces operation time and improves accuracy [9]. In [10], membership function of fuzzy inference was used as a feature and integrated with the analytic hierarchy process to adopt the Takagi-Sugeno-Kang (TSK) inference system to infer communication between vehicles. This was used to summarize the application of the final forwarding decision, and experiment results showed good scalability. Another paper [11] applied fuzzy inference systems to robots using the Evolving Fuzzy Logic System for Socially Assistive Robots (EFS4SAR). This approach combined the traditional rule-based fuzzy logic system with an evolutionary algorithm for socially assistive robots. This method was modeled through the evolutionary process of nature and proven to be creative. Experimental results showed that fuzzy rules evolving over time were personalized according to individual preferences and treatment needs of patients and were of higher originality. Finally, the Adaptive Neuro-Fuzzy Inference System (ANFIS) integrated with Global Positioning System (GPS) was used to develop an automated guided vehicle (AGV) collision-free system. This was tested in a noisy environment through simulation and showed good performance in terms of both time and path [12], [13], [14].

Recent literature suggests that the integration of fuzzy inference is highly compatible and provides reasonable solutions to various problems. As such, this study takes the fuzzy inference method as the theoretical basis and integrates it with metaheuristic algorithms to explore how to prioritize the location of robots when tidying up objects.

Many scholars have proposed the development of metaheuristic algorithms, with popular ones including Particle Swarm Optimization (PSO) [15], Genetic Algorithm (GA) [16], Artificial Bee Colony Algorithm (ABC) [17], Grey Wolf Optimizer (GWO) [18], Sine Cosine Algorithm (SCA) [19], Whale Optimization Algorithm (WOA) [20], Harris Hawks' Optimization (HHO) [21], Equilibrium Optimizer (EO) [22], and Prairie Dog Optimization Algorithm (PDOA) [23]. These algorithms are widely used in various fields. In recent years, scholars have proposed several metaheuristic algorithms to solve scheduling problems. For instance, a hybrid particle swarm optimization (HPSO) method was used to solve the problem of idleness in the flexible assembly system (FAS), achieving the minimum completion time [24]. An MVO-GA method was proposed to improve the computing efficiency of the cloud and accurately assign tasks based on different characteristics [25]. The Discrete Artificial Bee Colony (DABC) algorithm was used to minimize manufacturing time for the Distributed Heterogeneous No Wait Flow-shop Scheduling Problem (DHNWFSP) [26]. Similarly, GWO was used to effectively deal with job shop scheduling problems [27]. A multi-objective method based on the Modified Sine-Cosine Algorithm (MSCA) was proposed to optimize completion time and energy for task scheduling problems of MPS [28]. An Improved Multi-Objective Whale Optimization Algorithm (IMOWOA) was proposed to solve the MOHFSP-DRP and obtain the optimal solution set based on Pareto [29]. The integration of two metaheuristic algorithms also showed promising results, as seen in the improved Harris Hawks Optimization (HHO) algorithm based on Simulated Annealing (SA) for scheduling jobs in the cloud environment [30]. Scholars have also proposed integrating the Equilibrium Optimizer (EO) and White Shark Optimizer (WSO) to find better scheduling for the Power Schedule Problem (PSP) and improve overall optimization [31].

The literature reviewed above shows that the integration of metaheuristic algorithms with technologies from different fields has significantly improved job scheduling methods proposed by scholars. Furthermore, numerous experiments conducted in various fields have demonstrated the effectiveness of metaheuristic algorithms in solving job scheduling problems. Therefore, this study aims to conduct research on optimization integration using metaheuristic algorithms and fuzzy inference systems. Simulation tests will be conducted in different dimensions and in a real environment. Specifically, the study will focus on the task of two robots tidying up objects and determining the best position for each object to be tidied up.

The Traveling Salesman Problem (TSP) involves finding the shortest route for a traveling salesman who must visit a set of cities without repetition, starting from one city and returning to the starting city. It is a graph heuristic combinatorial optimization problem. Heuristic algorithms used to solve TSP can be broadly classified into construction heuristics algorithms and improvement heuristics algorithms. Construction heuristics algorithms start from an empty solution and add solution components repeatedly until the complete solution is constructed. Examples of construction heuristics algorithms include the Greedy Algorithm [32] and Dynamic Programming [33]. On the other hand, improvement heuristics algorithms take a solution generated by a construction heuristics algorithm as the starting point and spend more time repeatedly refining it to improve its quality. Examples of improvement heuristics algorithms include the GA [34], ACO [35], and PSO [36], [37]. Some scholars have also applied various optimization methods in machine learning to solve TSP and have made progress [38], [39], [40]. Based on the literature mentioned above, the proposed approach in this research aims to address the problem of location priority in TSP by employing fuzzy inference systems combined with metaheuristic algorithms such as the Greedy Algorithm, Dynamic Programming, and Genetic Algorithm. The theoretical analysis and experimental results demonstrate that this method is highly efficient in optimizing TSP.

Other sections in this paper are described as follows: Section II provides a problem description, including the robots used, the software structure, and the environment. In Section III, we describe the method proposed in this paper, explain the overall system structure, and present the feasibility of experiments. Section IV presents the experimental results, which include comparisons between the proposed method and other methods in different dimensions. Finally, Section V summarizes the findings and implications of the research, shedding light on potential directions for future studies.

II. PROBLEM DESCRIPTION

This section describes how to set up an environment in a space and considers how to effectively tidy up objects using a robot after placing them randomly in this environment. The problem description is mainly divided into three types, namely, environment description, robot description, and description of tidying up.

A. ENVIRONMENT DESCRIPTION

The reason for defining the environmental parameters at the beginning of the research was to verify if the method employed in this investigation could effectively collect the path data of objects through simulation. Once the simulation results confirmed the feasibility of the method, the data would be further tested in an actual environment. For the experimental setup, two shelves and a table were designated as the areas where objects were randomly positioned. The extent of the environmental field is illustrated in Figure 1.

As depicted in Fig. 1, the green dots indicate the areas where objects can be randomly placed, which include a table (1.5 m $*$ 0.8 m) and two shelves (0.9 m $*$ 0.5 m). The environmental locations were measured in the real field. The orange marks correspond to six positions that are considered ideal for the robots to tidy up the objects. These positions are denoted as Pos. 1, Pos. 2, Pos. 3, Pos. 4, Pos. 5, and Pos. 6. The blue marks indicate the initial and final points of the robots, which are also the starting and ending points of the tidying up process. These points are denoted as Pos. 7, Pos. 8, Pos. 9, and Pos. 10.

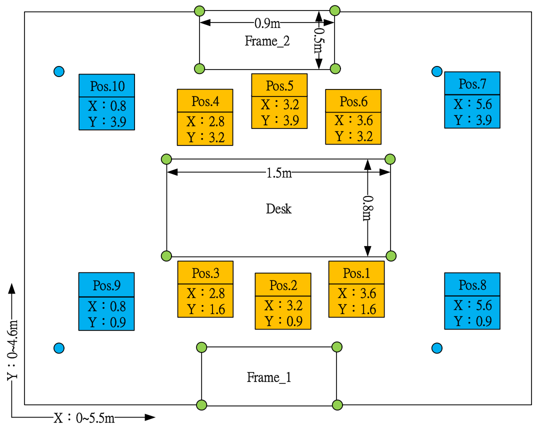

FIGURE 1. Description of the field for tidying up.

TABLE 1. List of objects to be tidied up.

		
(a) crosshead screwdriver	(b) slotted screwdriver	(c) box cutter
		
(d) object cardboard box	(e) part box 1	(f) part box 2

B. ROBOT APPLICATION PLATFORM DESCRIPTION

The first application platform used is a self-made Automatic Mobile Manipulator Robot (AMMR) shown in Fig. 1(a). This type of AMMR has AGV mobility, allowing it to move freely to any point in space. The robot is equipped with a mechanical arm to grasp objects, and there is enough space above it to place objects to be tidied up, making it suitable for tidying up tasks.

For the structural design of AMMR, the upper end was designed with a 6+1 axis series mechanic arm, a depth camera (the "eye" in hand), an end suction cup, and embedded systems including NANO and NUC, as well as a computer. Among the embedded systems, NANO was mainly used for image recognition, NUC was used to control the moving position of the chassis of AMMR, and the computer was used to calculate the overall arm kinematics of AMMR.

The lower end of the robot was implemented by a wheeled mobile chassis mechanism, which enables the robot to successfully complete the instructions according to the moving path. Since the robot in this experiment was designed to optimize its path when moving, it was capable of automatically moving to the location selected by the identification system [41]. The mobile chassis of the robot is equipped with the simultaneous localization and mapping (SLAM) system. The chassis sensing system is based on a set of RGB-D cameras and dual 2D optical radars. After comprehensively analyzing the sensed information, this system uses ORB characteristic values to obtain the current location of the robot and model the field. It also extracts key frames to improve accuracy and reduce the processing flow of multiple images. The SLAM technology uses closed-loop detection to correct errors in map modeling. The SLAM design algorithm used in the chassis integrates data from the depth camera and optical radars simultaneously and dynamically adjusts the weight according to the real environment to improve the overall positioning accuracy of the robot.


(a) AMMR. (a) OMRA.

FIGURE 2. Robots application platform.

The second application platform used in this research is the Omnidirectional Mobile Robot Arm (OMRA) as shown in Fig. 2(b). It was equipped with a TM5-700m six-axis cooperative robot arm on the top of the platform, which had a TOYO's parallel claw at the end. An Intel® RealSense™ L515 depth-of-field camera was installed onto the arm to detect and recognize an irregular tool in an unknown state, allowing the robot to flexibly grasp and release the tool through the point cloud and image information. The chassis had omnidirectional mobility using Mecanum wheels, allowing the robot to move laterally and obliquely. Dual LiDARs were equipped on the left front and right rear of the robot, which provided the robot with information around itself with no blind spots when moving. The SLAM Algorithm was utilized to construct the environment map, position and navigate the robot, which enabled it to plan the route in the scene and reach the designated location. Compared with the traditional differential wheel, the designed robot had more diverse behaviors and ranges of movement.

C. DESCRIPTION OF TIDYING UP OBJECTS

The objective is to efficiently tidy up objects that are randomly placed in a workplace. The objects that need to be tidied up are mainly tools that are frequently used by workers, such as screwdrivers, utility knives, screws, nuts, and boxes. The end-effector selected by the robot for grasping and absorbing objects had a clamping force of no more than 2 kg. The objects that can be clamped are listed in Table 1. During the process of tidying up, the depth camera set up in the environment initially recognizes the randomly placed objects and then marks the position of each identified object.

The robots were used to tidy up randomly placed objects in a workplace. The objects consisted mainly of tools frequently used by workers, such as screwdrivers, utility knives, screws, nuts, and boxes, and the maximum clamping force of the robot's end-effector was 2 kg. The depth camera set up in the environment first identified the objects randomly placed in the environment and marked the position of each identified object. The robots then selected the most appropriate position to move and tidy up the objects based on the method described in Section III. The AMMR robot was used to absorb and tidy up boxes, while the OMRA robot was used to grasp and clamp objects. Once objects were collected at one location, the robots would identify if there were any other objects left in the environment. If the task of tidying up was accomplished, the robots would return to their original starting positions. The process of tidying up is shown in Fig. 3.

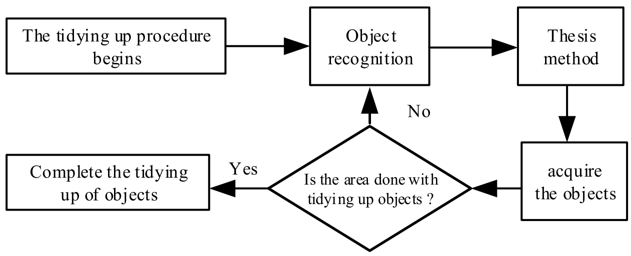

FIGURE 3. Process of tidying up.

III. ADAPTIVE FUZZY INFERENCE DECISION STRATEGY METHOD

This section discusses the research methodology, including the research design, proposed methods, simulation environment, and physical environment strategy evaluation. The research design aims to integrate the Fuzzy Inference System (FIS) and Equilibrium Optimizer (EO) algorithm into the Adaptive Fuzzy Inference Decision Strategy (AFIDS). The resulting AFIDS combines TSP with GA to determine the optimal location for the robot to identify and tidy up objects. The research methodology encompasses a set of processes and settings related to tidying up that must be completed to achieve the research objectives. Based on the research purpose, problems, and objectives, the integration is designed to facilitate the research.

A. METHODOLOGY DESIGN

This section discusses the location scheduling of robots in a messy environment. At first, literature on the advantages of FIS and the characteristics of metaheuristic algorithms were reviewed to determine their applicability to scheduling or adjustment problems related to adaptive functions. The most important attributes include the number of objects to be tidied up, obstacle avoidance, path length, and speed of

TABLE 2. The relationship between the fuzzy set input parameters and the membership degree of the tidy up location for objects.

Feature_1	Feature_2	Relation	Fuzzy Set	Relation
F_{ij}^{dist}	F_{ij}^{obs}	[0 1]	Level_1	[0 1/9]
			Level_2	[1/9 2/9 3/9]
			Level_3	[2/9 3/9 4/9]
			Level_4	[3/9 4/9 5/9]
			Level_5	[4/9 5/9 6/9]
			Level_6	[6/9 7/9 8/9]
			Level_7	[8/9 9/9]

TABLE 3. Incorporating the priority of location selection into the output parameters and membership functions relationship.

Output	Relation	Fuzzy Set	Relation
The degree of superiority or inferiority of the tidy up locations for objects	[0 1]	Best (Lv.1)	[0 1/9]
		Better (Lv.2)	[1/9 2/9 3/9]
		Good (Lv.3)	[2/9 3/9 4/9]
		Normal (Lv.4)	[3/9 4/9 5/9]
		Bad (Lv.5)	[4/9 5/9 6/9]
		Worse (Lv.6)	[6/9 7/9 8/9]
		Worst (Lv.7)	[8/9 9/9]

tidying up. The efficiency of tidy up mainly depends on the evaluation criteria for the best location according to the task schedule to generate a high tidy up rate. Therefore, this paper proposes a genetic scheduling method that integrates a hybrid metaheuristic algorithm and fuzzy inference decision strategy.

B. FUZZY INFERENCE SYSTEM

1) FUZZY SET

The process of selecting the optimal location requires critical feature factors, which can only be accurately determined through computation. To address this issue, scholars in literature [42], [43], [44] proposed using fuzzy sets to select features that can improve the identification results. By employing fuzzy inference methods to recognize datasets, satisfactory results can be achieved.

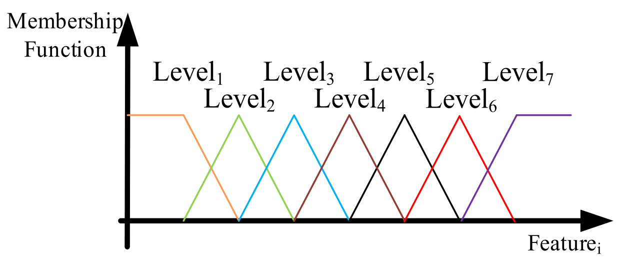

FIGURE 4. Fuzzy sets.

The method of selecting the optimal location using fuzzy theory mainly consists of environmental characteristics that constitute the entire set, as shown in Fig. 4. Then, membership functions are used to quantify the fuzzy sets. For the application of fuzzy theory, we use the fuzzy set with multiple feature values to analyze and judge the priority level of each location to obtain the most objects, as shown in Table 2.

In Table 3, the context of determining the priority sequence for organizing objects in the environment, we select two features for evaluation. The F_{ij}^{dist} between objects scattered in different areas and the current position information of the robot, as well as the number of obstacles information F_{ij}^{obs} between each location i in the environment and the robot's position j.

Initially, we prioritized the feature of distance between locations, aiming to minimize the overall time required for organizing the environment. However, since the robot's grasping motion is fixed, the time required for each object retrieval remains constant. Therefore, our tests did not show a significant reduction in the overall time required for environmental organization.

As a result, we considered the feature of obstacle quantity between locations. We assumed these obstacles could be other robots or randomly generated obstacles. The presence of obstacles would lead to waiting time for the robot to have them removed or to navigate around them. Our main focus was to minimize the waiting time during the object retrieval process. Therefore, by minimizing the number of obstacles, we aimed to minimize the time spent waiting for their removal or navigating around them.

Combining the information, we can conclude that the feature of relative distance and the feature of obstacle quantity between locations have a significant predictive capability for determining the priority order of robot's location organization in the environment. However, it should be noted that the fixed grasping motion limits the effectiveness of solely considering the distance feature. By incorporating the obstacle quantity feature, we can achieve a more efficient organization process.

To simplify the feature information complexity, the membership degree of the two feature sets is normalized to [1, 0]. F_{ij}^{dist} has seven levels, ranging from Level_1 to Level_7, where a higher level indicates a greater distance and an unfavorable relative position decision. Similarly, F_{ij}^{obs} is also divided into seven levels, ranging from Level_1 to Level_7, where a higher level represents more obstacles in the interval, resulting in a more complex position decision. Table 2 shows the selection order of location priorities, with the output membership degree ranging from 0 to 1. The fuzzy subset definitions for the output are as follows: Best, Better, Good, Normal, Bad, Worse, and Worst.

2) MEMBERSHIP FUNCTION

The membership function is characterized by quantifying the properties of a fuzzy set through mathematical formulas after the fuzzy set has been quantified. The membership function describes the continuous properties of an infinite fuzzy set, typically using triangular, trapezoidal, Gaussian, Bell-shaped, S-shaped, and Z-shaped membership functions. In this article, real-time performance is achieved, so the triangular membership function, which provides faster results,

TABLE 4. Initial fuzzy rule base for a decision system of tidy up locations for objects.

Loc_{lv} (F_{ij}^{obs} \ F_{ij}^{dist})	$Level_1$	$Level_2$	$Level_3$	$Level_4$	$Level_5$	$Level_6$	$Level_7$
$Level_1$	Lv.1	Lv.1	Lv.1	Lv.2	Lv.3	Lv.3	Lv.4
$Level_2$	Lv.1	Lv.2	Lv.2	Lv.2	Lv.3	Lv.4	Lv.5
$Level_3$	Lv.1	Lv.2	Lv.3	Lv.3	Lv.4	Lv.5	Lv.6
$Level_4$	Lv.1	Lv.2	Lv.3	Lv.4	Lv.5	Lv.6	Lv.7
$Level_5$	Lv.2	Lv.3	Lv.4	Lv.5	Lv.5	Lv.6	Lv.7
$Level_6$	Lv.3	Lv.4	Lv.5	Lv.6	Lv.6	Lv.6	Lv.7
$Level_7$	Lv.4	Lv.5	Lv.5	Lv.6	Lv.7	Lv.7	Lv.7

is used as follows,

$$\mu(\mathrm{x}) = \begin{cases} 0, & if \quad x < \alpha \\ \dfrac{x-\alpha}{\beta-\alpha}, & if \quad x \in [\alpha, \beta] \\ \dfrac{\gamma - x}{\gamma-\beta}, & if \quad x \in [\beta, \gamma] \\ 0, & if \quad x > \gamma \end{cases} \tag{1}$$

where variables satisfy the inequality $0 < \alpha < \beta < \gamma$.

3) DECISION MAKING LOGIC

By converting environmental characteristics from a dataset into a fuzzy set, designing membership distributions, and performing mathematical operations using membership functions, the current characteristic value can be obtained. After obtaining the current characteristic value, the membership degrees are generated by mapping it to the graph of its corresponding membership function. This process is known as fuzzification. Finally, to obtain the output result, a fuzzy inference process is necessary to determine the relationship between each characteristic.

The Fuzzy Inference Engine is composed of many logical rules and is based on the Takagi-Sugeno fuzzy model [45] The rule base can systematically generate the required IF-THEN rules from input feature information to the required priority output results. Additionally, it can adjust the front-end structure and parameter values and the back-end structure and parameter values based on the data. Therefore, this model is often used in control systems and other related inference fields. The inference formula is shown as,

$$\begin{cases} If\ F_{ij}^{dist} is A_k\ AND\ F_{ij}^{obs}\ is\ B_k \\ THEN\ Output_{result} = f(F_{ij}^{dist}, F_{ij}^{obs}) \end{cases} \tag{2}$$

where F_{ij}^{dist} and F_{ij}^{obs} represent the environmental characteristics of the input feature, A_k and B_k represent the degree of membership of each characteristic, and $Output_{result}$ represents the output, which is the position evaluation level. The initial rule base for this design, based on the characteristics of the sliding mode [46], [47], [48], [49], [50], is shown in Table 4.

4) DEFUZZIFICATION PROCESS

The final step of fuzzy inference is defuzzification, which converts the fuzzy output into a crisp value. There are various methods for defuzzification, but in this case, we will use the centroid method for its fast computation. The centroid method calculates the center point of the output by computing the weighted average of the membership function values as bellow,

$$Output_k^{result}(x) = \frac{\sum_{k=1}^{m} Output_k^{result}\, \mu(Output_k^{result})}{\sum_{k=1}^{m} \mu(Output_k^{result})} \tag{3}$$

where, μ_k is the membership degree of the *k-th* rule's output set, $Output_k^{result}$ is the *k-th* rule's output value, and m is the total number of rules.

C. EQUILIBRIUM OPTIMIZER

The Equilibrium Optimizer (EO) [22] is a metaheuristic algorithm that is based on physical laws, specifically using volume-controlled dynamic mass balance technology. This algorithm is inspired by the concept of balance in physics and aims to find the optimal solution in the search space while maintaining a balanced system state. The optimization process of EO can be divided into two stages: exploration and exploitation.

During the exploration stage, EO uses a random walk strategy to explore the search space. Each individual conducts a random walk and calculates its fitness value. These fitness values are then utilized to update the concentration of elements in the control capacity matrix to achieve dynamic mass balance with volume control. In the exploitation stage, EO employs a particle swarm optimization algorithm to optimize the objective function. The algorithm uses the optimal position of each individual and the global optimal position to update the position and velocity of everyone for a better search of the optimal solution.

Overall, EO exhibits the following characteristics: it is based on physical laws, uses a metaheuristic algorithm, employs dynamic mass balance and volume control, and combines both exploration and exploitation stages. The mass balance allows for the establishment of a dynamic environment by accounting for the physical processes of mass entering, leaving, and generating within the controlled volume. Generally, a first-order differential equation is used to describe this process,

$$V\frac{dC}{dt} = G + QC_{eq} - QC \tag{4}$$

where the parameter C represents the concentration inside the control volume (V), G is the rate of mass generation inside the control volume, Q denotes the volumetric flow rate entering and leaving the control volume, C_{eq} gives the concentration at an equilibrium state, and $V\frac{dC}{dt}$ is the rate of change of mass in the control volume. G is used to improve the convergence during the equilibrium development phase and is selected using GP.

When $V\frac{dC}{dt}$ reaches zero, the system reaches a stable equilibrium state. The differential equation can be transformed to the following equation,

$$C = \frac{G}{\lambda V}(1-F) + (C_0 - C_{eq})F + C_{eq} \tag{5}$$

where F is the convergence factor of the algorithm, λ is the flow rate, and C_0 represents the initial concentration in the control volume. The steps of the Equilibrium Optimizer are organized as following steps:

–First, the particle's concentration is initialized as

$$C_{initial} = C_{min} + rand_q\,(C_{max} - C_{min})\; q = 1, 2, \cdots, n \tag{6}$$

where C_{max} and C_{min} denote the upper and lower boundary, respectively, and $rand_q$ represents a random number between 0 and 1 for individual q.

–Second, construct an equilibrium state pool to enhance global search ability and avoid being trapped in local optimal solutions. Consisting of the four current best candidate solutions, the equilibrium state pool is designed as,

$$C_{eq,pool} = \{C_{eq1}, C_{eq2}, C_{eq3}, C_{eq4}, C_{eq,avg}\} \tag{7}$$

–Third, calculate the convergence factor F as bellow,

$$F = a_1 sign(r-0.5)\left[e^{-\lambda t} - 1\right] \tag{8}$$

$$t = \left(1 - \frac{Iter}{Max\_Iter}\right)^{\left(a_2 \frac{Iter}{Max\_Iter}\right)} \tag{9}$$

where F is the convergent factor obtained after the overall operation, a_1 represents the weight of the exploration search, and t presents the time of the iteration function as shown in (9), and where $Iter$ represents the current iteration number, Max_Iter represents the overall maximum iteration number. a_2 represents the weight of the exploitation search.

–Fourth, calculate the mass generation rate G to enhance the overall local search capability of the algorithm,

$$G = G_0 F \tag{10}$$

$$G_0 = GP(C_{eq} - \lambda C)C \tag{11}$$

$$GP = \begin{cases} 0.5rand_1 & rand_2 \geq GP \\ 0 & rand_2 \leq GP \end{cases} \tag{12}$$

where GP represents the mass generation rate control parameter vector, while $rand_1$ and $rand_2$ are both random variables between 0 and 1.

–Fifth, update the best solution by iteratively applying (5) for a certain number of iterations and optimizing the best solution until the end of the iteration.

The equilibrium algorithm utilizes fitness functions to achieve the required convergent solution for the problem. In the case of prioritizing the selection of robot storage locations, a fitness function is defined to find the optimal solution of the fuzzy rule base, which can be applied in various hierarchical environments. The pseudo-code for the overall equilibrium is listed in Algorithm 1.

Algorithm 1 The Pseudo-Code of EO Optimizer

Input: ($Particles_{max}$, $Iter_{max}$, $Dims.$, Low_{bound}, Up_{bound}, $Fit.$)
Output: ($Fit._{best}$, $Particle_{best}$)
Intitialize: a_1=2, a_2=1, GP=0.5
while $Iter <= Iter_{max}$
for $q = 1$: $Particles_n$
    $particles < Low_{bound}$
    $particles > Up_{bound}$
    Randomly generate n particles by (6).
    Caculate the fitness function of each particle=$Ceq(q)_{fit}$
    The four particles with the best local solutions identified through all particles comparison= $Ceq1_{particle}$, $Ceq2_{particle}$, $Ceq3_{particle}$, $Ceq4_{particle}$
    The four fitness value with the best local solutions identified through all fitness value comparison = $Ceq1_{fit}$, $Ceq2_{fit}$, $Ceq3_{fit}$, $Ceq4_{fit}$
end for
Save best particle to totalbest solution.
Caculate t by (9).
for i=1:$Particles_{max}$
    Caculate C by (5).
    Contruct the equilibrium pool by (7).
    Caculate F by (8).
end for
    $Iter$= $Iter$+1
end while

D. TRAVELLING SALESMAN PROBLEM (TSP)- GENETIC ALGORITHM (GA)

TSP is a well-known combinatorial optimization problem that involves finding the shortest circular route through a

given set of cities for a salesman to visit. Due to its complexity, traditional solving methods are often impractical, and researchers have developed various heuristic algorithms to address this problem. One such algorithm is GA, which simulates evolutionary mechanisms such as natural selection, mating, and genetic mutation to optimize the combination of individual genes and ultimately find the optimal solution. To apply GA to the TSP, the following steps are generally taken:

1) Initialize a population, where every chromosome (individual) represents a possible route.
2) Evaluate the individual fitness, which is calculated as the total length of the route.
3) Select individuals with higher fitness as parents and produce the next generation through crossover and genetic mutation. In the TSP problem, crossover can be achieved by selecting a crossover point in two routes, splitting and exchanging them, to create two new individuals. Genetic mutation can be achieved by randomly swapping the positions of two cities in a route.
4) Repeat steps 2) and 3) until a stopping condition is met, such as reaching a certain fitness level or exceeding a certain runtime.

TABLE 5. Data distribution for three dimensions of environments.

The level of spatial environment	Low	Medium	High
Area	1~3	4~6	7~9
Number of positions	6~18	24~36	42~54
Maximum number of objects per position	10	10	10
Number of obstacles between positions	0~7	0~14	0~21
Unit for generating the dataset	30	30	30

5) Select the individual with the highest fitness as the optimal solution, which represents the shortest route.

While genetic algorithms are effective at solving complex optimization problems like the TSP, it's important to note that they cannot guarantee finding the global optimal solution. However, they can usually find a solution that is very close to optimal. This is especially relevant for the TSP problem, which is classified as an NP-hard problem and cannot be solved in polynomial time. Therefore, heuristic algorithms like genetic algorithms are often the most practical and effective solutions for solving the TSP problem.

E. ADAPTIVE FUZZY INFERENCE DECISION STRATEGY

FIS is well-suited for solving multi-factor problems, as it can use system architecture to find reasonable decisions. On the other hand, the EO algorithm has shown excellent performance in terms of fast convergence and finding the best solution, as demonstrated by verification and comparison with other algorithms. The rule base in FIS is usually designed based on system or expert experience and can be adjusted based on one-time or system feedback. In contrast, the EO algorithm can optimize the data in the database, adjust and develop features through algorithmic exploration, and ultimately obtain the global optimal solution. Therefore, this paper proposes using the EO algorithm to adjust the initial rule base and improve the overall intelligence of the positioning system.

1) DATA GENERATION

First, we generate a database using a simulation system, where the robot's current location, the feature F_{ij}^{dist} of each tidy-up object, and the feature F_{ij}^{obs} of obstacles to avoid during the tidy-up route are randomly generated based on the map presented in Fig.1. Next, we use the EO algorithm to train this data, setting three different dimensions of data based on Table 5. In Table 5, the number of positions represents the number of locations in the spatial environment. Regarding the number of objects, we focus on objects that the robot can pick up, as described in Section II, with a maximum of 10 objects that can be placed at each position as the benchmark. This parameter is fixed because the time required for the gripping process depends on the movement of the mobile robotic arm. Finally, the number of obstacles represents the number of obstacles encountered by the robot during the movement process from one position to the next, defined based on different dimensions of the dataset.

2) TRAINING THE MODEL TO OPTIMIZE THE FUZZY RULE BASE

After generating the data, we use F_{ij}^{dist} and F_{ij}^{obs} to create feature matrices, namely distance matrix ($Dist_{mat}$) and obstacle matrix (Obs_{mat}). The $Dist_{mat}$ is composed of F_{ij}^{dist}, which is represented by the following equation,

$$F_{ij}^{dist} = Distance_{time} + Grasp_{time} \tag{13}$$

where the $Distance_{time}$ consists of the distance between the robot's current position and the next position as well as the time required for grasping an object at each position ($Grasp_{time}$). These two time-based measurements are integrated into $Dist_{mat}$ given in Eqn. (15) to calculate the total time.

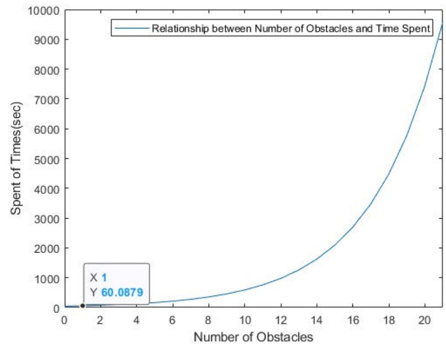

FIGURE 5. The relationship curve between the number of obstacles and the time spent by the robot for obstacle avoidance.

As for the Obs_{mat} is composed of F_{ij}^{obs}, we assume that each obstacle incurs a unit of time. We define the distance for the robot to navigate around obstacles as exhibiting exponential growth. The robot's movement speed remains constant. Consequently, as the number of obstacles increases, the robot will require more time, and the feasible distance will exhibit exponential growth. The relationship between obstacle avoidance time and the number of obstacles is shown as follows,

$$F_{ij}^{obs} = Obstacle_{time} = \alpha \times e^{\beta \times obs_{num}} \tag{14}$$

where $\alpha = 46.425$, $\beta = 0.2536$, and obs_{num} represents the number of obstacles. The relationship curve between the number of obstacles and the time spent by the robot for obstacle avoidance is shown in Fig.5. These two matrices are then integrated to form the following cost matrix,

$$\begin{aligned} Cost_{mat} &= Dist_{mat} + Obs_{mat} \\ &= \begin{bmatrix} F_{11}^{dist} & \cdots & F_{1j}^{dist} \\ \vdots & \ddots & \vdots \\ F_{i1}^{dist} & \cdots & F_{ij}^{dist} \end{bmatrix} + \begin{bmatrix} F_{11}^{obs} & \cdots & F_{1j}^{obs} \\ \vdots & \ddots & \vdots \\ F_{i1}^{obs} & \cdots & F_{ij}^{obs} \end{bmatrix} \end{aligned} \tag{15}$$

The TSP-GA algorithm is applied to find the optimal cost variables ($Cost_{var}$) and the priority order from the starting position to each position for each generated dataset, based on $Cost_{mat}$. After simulating the generated data and inputting them into the fitness function, we adopt the balancing algorithm to obtain an optimized solution, which is then used to adjust the fuzzy rule base. Finally, we obtain the optimized fuzzy rule base.

In this study, a customized fitness function was utilized as the standard for training the rule base in (16). The objective was to minimize the fitness value, indicating a better rule base for practical implementation.

$$Fitness(n) = \sum_{n=0}^{max} Time(n) - ReferenceTime \qquad (16)$$

where the variable "*Time*" represents the time required for the robot to travel the distance information between its current position and the target object, as well as the time required to avoid obstacles. The "*Reference Time*" is obtained globally through dynamic programming. We used the equilibrium algorithm to optimize the fuzzy rule base initially by using this fitness function. The "*max*" represents the maximum number of objects or items that can be accommodated in the tidy up space or environment during a single computation or operation. The overall design process of AFIDS is illustrated in Fig. 6.

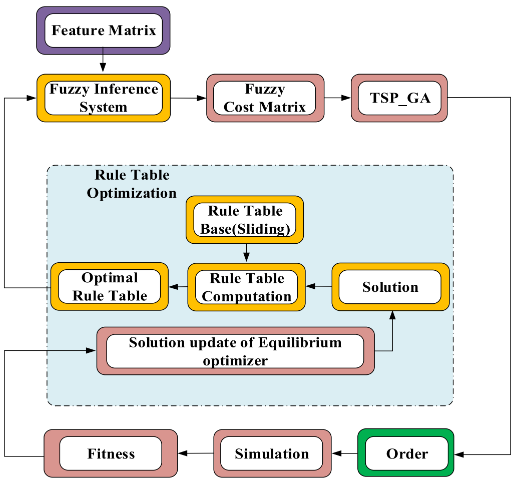

FIGURE 6. AFIDS flow chart.

IV. EXPERIMENTAL RESULTS

In this section, we conduct tests on the proposed method and analyze the experimental results. The experimental data was generated using Matlab 2023a software and run on a computer equipped with an Intel(R) Core(TM) i7-8700K 3.70 GHz CPU and 32GB RAM. The research process involves transforming the actual environment into simulated environment information and deriving three levels of information. In order to present different situations with varying conditions in the experiment, we randomly generated datasets based on the environment information of each dimension to verify the method proposed in this paper. We also analyzed the degree of discrimination of features, trained the rule base, and established the priority order of tidy up positions.

TABLE 6. Execution results of different feature matrices in low-level environment space.

Low-Level	$Dist_{mat}$	$Cost_{mat}$	$F\text{-}Cost_{mat}$
BSET (sec)	682	370	370
WORST (sec)	3866	2640	2696
AVG (sec)	2142.133	1465.733	1457.233

TABLE 7. Execution results of different feature matrices in medium-level environmental space.

Medium-Level	$Dist_{mat}$	$Cost_{mat}$	$F\text{-}Cost_{mat}$
BSET (sec)	2174	2174	2174
WORST (sec)	8520	4834	4738
AVG (sec)	5415.133	3483	3443.133

TABLE 8. Execution results of different feature matrices in high-level environmental space.

High-Level	$Dist_{mat}$	$Cost_{mat}$	$F\text{-}Cost_{mat}$
BSET (sec)	5102	4546	4532
WORST (sec)	16406	6764	6704
AVG (sec)	9771.867	5652.067	5605.4

TABLE 9. Statistical analysis of the correlation between input features and output result.

Dataset		p-value-	Pearson
$Dist_{mat}$ -$F\text{-}Cost_{mat}$	Low-Level	5.478e-07	0.646
	Medium-Level	1.336e-18	0.513
	High-Level	5.390e-73	0.639
Obs_{mat} - $F\text{-}Cost_{mat}$	Low-Level	6.180e-17	0.882
	Medium-Level	2.045e-80	0.871
	High-Level	1.026e-132	0.787

A. CORRELATION ANALYSIS BETWEEN INPUT FEATURE MATRIX AND OUTPUT RESULTS

To analyze the correlation between the input features and the output results, we conducted an experiment using statistical measures such as the p-value and Pearson correlation coefficient. A p-value smaller than 0.05 indicates a statistically significant difference in organization time among different obstacle quantities [51], [52], [53]. The Pearson correlation coefficient measures the linear correlation between two variables, ranging from −1 to 1, where 1 represents a perfect positive correlation, −1 represents a perfect negative correlation, and 0 indicates no linear correlation.

After performing p-value and Pearson correlation coefficient tests on the input environmental features at three levels and the corresponding output results, as shown in Table 9, we can draw the following conclusions. The low-level input feature, the distance matrix, exhibits a higher correlation with the output quality matrix (Pearson coefficient of 0.6456), and it has a very small p-value (5.4778e-07), indicating a significant linear relationship between these two features. The low-level input feature, the obstacle matrix, demonstrates an even higher correlation with the output quality matrix

TABLE 10. Experimental results of optimizing fuzzy rule base with various algorithms.

Algorithm	PSO	GA	ABC	GWO	SCA	WOA	HHO	EO	PDO	FCSO
Best	-61664	-61648	-61664	-61664	-61660	-61660	-61660	-61664	-61656	-61664
Average	-61587.7	-61386.5	-61644.4	-61631.1	-61582.8	-61543.1	-61386.8	-61650.1	-61550.7	-61569.5
Worst	-61228	-61056	-61584	-61532	-61504	-61256	-60888	-61540	-61360	-61292
STD	86.495	171.768	22.143	41.299	40.167	89.441	188.3537	26.71398	64.03735	83.05533
Average Time	1913.423	1906.42	3755.653	1841.196	1853.473	1840.863	4318.408	1848.605	1913.83	1902.016
Rank of Fitness	4	10	2	3	5	8	9	1	7	6
Rank of Time	7	6	9	2	4	1	10	3	8	5
Total Rank	6	9	6	2	3.5	3.5	10	1	8	6

TABLE 11. Low-level site dimension expanded tidying up time.

F-Costmat	*Expansion of site dimensions=1*					*Expansion of site dimensions=2*					*Expansion of site dimensions=3*				
	GM	DP	GA	ACO	PSO	GM	DP	GA	ACO	PSO	GM	DP	GA	ACO	PSO
1st Run	370	370	208	264	312	1948	1896	1704	1792	1776	2366	2290	2290	2304	2264
2nd Run	696	696	414	504	342	1790	1454	1264	1264	1718	2346	2170	2064	2064	3516
3rd Run	1128	904	554	764	1022	1434	1334	940	1020	1274	2360	2172	2172	2124	2790
4th Run	1116	800	342	676	342	1484	1468	1328	1364	1364	2796	2696	2696	2684	2720
5th Run	1048	1048	612	652	698	1456	1396	1254	1366	1212	2288	2204	2204	2052	3066
6th Run	786	698	342	414	516	1934	1882	1622	1702	1726	3018	2062	2062	2994	2870
7th Run	1250	858	458	770	458	1168	1164	1066	1070	1500	2174	2138	2034	2096	3214
8th Run	912	912	616	626	614	1854	1322	1190	1288	2290	2086	1898	1810	1820	1940
9th Run	926	758	396	568	566	1310	1310	1232	1232	1504	2720	1972	1972	2346	4054
10th Run	1018	1018	612	726	802	1438	1422	1386	1320	1454	2242	2174	2086	2124	2878
Average	**925**	**806.2**	**455.4**	**596.4**	**567.2**	**1581.6**	**1464.8**	**1298.6**	**1341.8**	**1581.8**	**2439.6**	**2177.6**	**2139**	**2260.8**	**2931.2**
Rank	5	4	1	2	3	4	3	1	2	5	4	2	1	3	5

(Pearson coefficient of 0.8816), and it has an exceedingly small p-value (6.1797e-17), suggesting a more significant linear relationship between these two features. The medium-level and high-level input matrices also exhibit higher correlations with the output quality matrix, and they have small p-values, indicating significant linear relationships in predicting the output quality. This implies that the distance matrix and obstacle matrix, as input features, have a higher predictive capability for determining the quality of the robot's organization time, with the obstacle matrix showing stronger predictive power. Additionally, the medium-level and high-level features also demonstrate higher correlations and significance, indicating their greater contribution in predicting the results.

Overall, these findings suggest that the input features, specifically the distance matrix and obstacle matrix, have a significant predictive capability for determining the quality of the robot's organization time. Furthermore, the medium-level and high-level features also exhibit high correlations and significance, indicating their substantial contribution to predicting the results.

B. SELECTION AND COMPARISON OF FEATURE PROPERTIES

The selection of advantageous features is crucial for the robot's decision-making in determining the tidying up position. We aim to demonstrate whether feature selection can significantly reduce the overall time required for tidying up objects using fuzzy inference. We use three different feature matrices: $Dist_{mat}$, $Cost_{mat}$ (which includes distance and obstacle information), and the fuzzy cost matrix obtained from fuzzy inference (F- $Cost_{mat}$). We input these matrices into GA to obtain the best path for the robot to tidy up the objects. As shown in Tables 6-8, using the fuzzy inference matrix can effectively improve the quality of the tidying up path, especially in cases where the data dimensionality is high, the area of the messy environment is large, and there are many obstacles.

C. RULE TABLE COMPARISON WITH OTHER METAHEURISTIC ALGORITHMS

The $F\text{-}Cost_{mat}$ has shown good performance in validating the tidying up objects time, but the rule base in fuzzy inference plays a critical role as the knowledge core of the overall recognition system. Therefore, we integrate the fuzzy rule base using the equilibrium optimizer algorithm as a benchmark for optimizing data, which enables more efficient tidying up of objects.

In situations where environmental information has a high degree of complexity, finding a global solution can be challenging and prone to getting stuck in local solutions. In this experiment, we used the equilibrium optimizer algorithm to test high-dimensional environmental data, and the results were recorded in Table 10. This table lists the best, average, worst, and standard deviation of the results obtained by each algorithm after running 30 times. Smaller values of

TABLE 12. Medium -level site dimension expanded tidying up time.

$F\text{-}Cost_{mat}$	Expansion of site dimensions=4					Expansion of site dimensions=5					Expansion of site dimensions=6				
	GM	DP	GA	ACO	PSO	GM	DP	GA	ACO	PSO	GM	DP	GA	ACO	PSO
1st Run	3274	3034	2984	3024	3800	3864	3408	3408	3826	3876	4398	3926	3926	5656	4706
2nd Run	3174	2818	2818	2894	5102	3728	3528	3528	3648	4246	3896	3800	3800	5900	8314
3rd Run	3318	3054	3054	3180	5278	3324	3276	3276	3344	5900	4890	4738	4802	4894	6834
4th Run	3172	2928	2928	3022	4340	3698	3602	3602	3798	4840	4412	4008	4008	4282	6512
5th Run	3740	2728	2728	2840	2856	4724	3608	3608	3998	5288	4076	3432	3432	3882	6054
6th Run	3232	3164	3024	3152	3448	4552	4420	4420	4836	5474	4172	3972	3884	5880	8324
7th Run	2230	2174	2174	2230	2732	3606	3206	3206	4492	5974	4554	4254	4254	4684	5388
8th Run	3200	2956	2956	2996	4316	2692	2640	2640	2878	3202	4618	4158	4162	4506	4638
9th Run	3162	2942	2942	2956	3060	3600	3176	3176	3318	3726	4404	3960	3960	4442	4702
10th Run	2630	2402	2402	2422	3822	3532	3532	3532	3746	3856	5004	4684	4748	5016	5244
Average	**3113.2**	**2820**	**2801**	**2871.6**	**3875.4**	**3732**	**3439.6**	**3439.6**	**3788.4**	**4638.2**	**4442.4**	**4093.2**	**4097.6**	**4914.2**	**6071.6**
Rank	4	2	1	3	5	3	1.5	1.5	4	5	3	1	2	4	5

TABLE 13. High-level site dimension expanded tidying up time.

$F\text{-}Cost_{mat}$	Expansion of site dimensions=7					Expansion of site dimensions=8					Expansion of site dimensions=9				
	GM	DP	GA	ACO	PSO	GM	DP	GA	ACO	PSO	GM	DP	GA	ACO	PSO
1st Run	4986	4758	4854	5110	5408	6170	5646	5646	5960	10064	6688	6068	6100	6484	8680
2nd Run	5452	4996	5144	5700	8946	6010	5554	5504	6038	9012	6570	6206	6270	7794	9140
3rd Run	4654	4638	4638	5198	9828	6040	5416	5464	6468	12078	6892	6400	6444	7128	13528
4th Run	5614	5102	5114	5622	8514	6258	5794	5886	6282	7906	7348	6688	6616	8254	9572
5th Run	4854	4546	4558	5650	7330	6968	5800	5796	6658	8136	6626	6422	6422	6588	13908
6th Run	4732	4568	4568	4892	6110	5492	5392	5392	5746	6402	9774	6068	6078	6646	13552
7th Run	5220	5088	5088	8646	7806	6758	6078	6138	7046	11786	6494	6068	5834	6370	10396
8th Run	5604	5412	5412	6838	9734	5192	4972	4976	5324	6446	6744	6148	6216	6702	8152
9th Run	5400	4876	4892	5486	5662	6008	5888	5888	7582	8578	7326	6510	6426	7586	10442
10th Run	5402	5370	5430	5540	6010	6302	5962	5962	6334	7038	5722	6422	5550	6168	8612
Average	**5191.8**	**4935.4**	**4969.8**	**5868.2**	**7534.8**	**6119.8**	**5650.2**	**5665.2**	**6343.8**	**8744.6**	**7018.4**	**6300**	**6195.6**	**6972**	**10598**
Rank	3	1	2	4	5	3	1	2	4	5	4	2	1	3	5

STD indicate higher stability of the algorithm's performance in solving the problem. The experimental results show that the EO algorithm is effective in solving the efficiency problem related to tidying up objects.

D. COMPARISON RESULTS OF AFID INTEGRATED WITH TSP METHODS

Finally, in this study, we integrated five TSP methods using a F-Costmat and an optimized fuzzy rule base. For the TSP methods, we assessed the greedy (GM), dynamic programming (DP), genetic algorithm (GA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) on low, medium, and high-level environmental data problems. The experimental results are listed in Tables 11-13.

The tables, 11 to 13, present the time results for the robot's object tidying task in three different environmental levels. Table 11 demonstrates that the GA algorithm achieves the shortest tidying time in the TSP problem. Based on the experimental results in Table 12, it is evident that DP shows the potential to outperform GA in certain cases. Furthermore, Table 13 highlights that DP can efficiently tidy objects in a brief time, especially in higher-level environmental settings.

TABLE 14. Computational time for tidying up optimization results with different levels of site expansion.

Level	Expansion of site dimensions	GM	DP	GA	ACO	PSO
Low	1	0.000254	0.000448	0.035	0.550	0.601
	2	0.000175	0.004344	0.037	1.427	0.564
	3	0.000113	0.066649	1.717	2.448	0.761
Medium	4	0.000219	0.939458	1.781	4.477	1.167
	5	0.000313	12.09854	1.802	6.6778	1.955
	6	0.000218	145.3862	1.886	9.211	2.181
High	7	0.000208	1594.747	1.904	11.809	3.150
	8	0.000913	12973.62	1.963	15.296	4.015
	9	0.000137	167577.1	2.096	19.686	5.913

In summary, the findings from Tables 11 to 13 lead to the following conclusions. The GA algorithm performs well in terms of data testing results, while GM and ACO exhibit relatively moderate performance without any notable outcomes. Conversely, PSO demonstrates relatively poorer performance in the TSP test for this dataset.

The effectiveness of decision-making depends not only on finding the optimal solution but also on the computational time required for the process. Nowadays, decision-making must be feasible in real-time. The computation time for optimizing the tidying up of objects is presented in Table 14 for different field dimensions. It can be observed that the GM algorithm achieves fast computation time when dealing with low-level environmental information. However, as the environmental information becomes more complex, dynamic programming requires a significant amount of computation time, which is less ideal for decision-making. On the other hand, the GA algorithm exhibits reliable performance in terms of computation time across all three levels of environmental information. Therefore, based on the findings of this study on object tidying tasks, considering its efficient results and computational speed, the GA is a favorable choice. The corresponding video of experiment is available at https://youtu.be/rjMDgopM-9M.

V. CONCLUSION AND FUTURE WORK

This paper focuses on the conversion of real data into simulated data and the compilation of data from low-level to high-level environmental information for research and analysis purposes, utilizing large databases. The analytical findings can be summarized into three key points.

Firstly, the effectiveness of feature data selection was validated by employing the fuzzy inference method to obtain the most relevant input feature information.

Secondly, by integrating the rule base in fuzzy inference with the balancing algorithm, improved fitness values were achieved. This integration involved replacing the original rule base and combining it with the TSP method.

Lastly, we successfully integrated the Fuzzy Cost Matrix with GA to optimize the selection and order of robot tidying up positions for objects. Throughout our research, we conducted experiments on feature selection, the integration of the EO algorithm, and the utilization of GA for finding the optimal selection and order of tidying up positions for the robot. These results have undergone rigorous testing in real-world environments. During the testing process, we have achieved successful completion of the task for tidying up objects using two robots.

For future work, researchers can explore dimensionality reduction techniques on the developed AFIDS feature matrix method to investigate its potential for reducing computation time in high-level environments. In addressing the path scheduling problem, machine learning methods can be employed to adjust the parameters of GA mating and mutation functions. The objective is to further enhance the accuracy and efficiency of the AFIDS developed in this research.

REFERENCES

[1] N. Abdo, C. Stachniss, L. Spinello, and W. Burgard, "Robot, organize my shelves! Tidying up objects by predicting user preferences," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1557–1564.

[2] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase, "Development of human support robot as the research platform of a domestic mobile manipulator," *Robomech J.*, vol. 6, no. 1, p. 4, Dec. 2019.

[3] Z. Yan, N. Crombez, J. Buisson, Y. Ruichck, T. Krajnik, and L. Sun, "A quantifiable stratification strategy for tidy-up in service robotics," in *Proc. IEEE Int. Conf. Adv. Robot. Social Impacts (ARSO)*, Jul. 2021, pp. 182–187.

[4] F. Wang, J. R. G. Olvera, and G. Cheng, "Optimal order pick-and-place of objects in cluttered scene by a mobile manipulator," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6402–6409, Oct. 2021.

[5] T. Matsushima, Y. Noguchi, J. Arima, T. Aoki, Y. Okita, Y. Ikeda, K. Ishimoto, S. Taniguchi, Y. Yamashita, S. Seto, S. Shane Gu, Y. Iwasawa, and Y. Matsuo, "World robot challenge 2020—Partner robot: A data-driven approach for room tidying with mobile manipulator," 2022, *arXiv:2207.10106*.

[6] J. Piao, H. Jo, and J.-B. Song, "Robotic tidy-up tasks using point cloud-based pose estimation," in *Proc. 20th Int. Conf. Control, Autom. Syst. (ICCAS)*, Busan, South Korea, Oct. 2020, pp. 316–319.

[7] J. Piao, H. Jo, and J.-B. Song, "Pose estimation and image matching for tidy-up task using a robot arm," *J. Korea Robot. Soc.*, vol. 16, no. 4, pp. 299–305, Dec. 2021.

[8] D. Zhu, H. Seki, T. Tsuji, and T. Hiramitsu, "Tableware tidying-up robot system for self-service restaurant–detection and manipulation of leftover food and tableware-," *Sensors*, vol. 22, no. 18, p. 7006, Sep. 2022.

[9] L. T. Hong Lan, T. M. Tuan, T. T. Ngan, L. H. Son, N. L. Giang, V. T. Nhu Ngoc, and P. V. Hai, "A new complex fuzzy inference system with fuzzy knowledge graph and extensions in decision making," *IEEE Access*, vol. 8, pp. 164899–164921, 2020.

[10] A. Hawbani, E. Torbosh, X. Wang, P. Sincak, L. Zhao, and A. Al-Dubai, "Fuzzy-based distributed protocol for vehicle-to-vehicle communication," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 3, pp. 612–626, Mar. 2021.

[11] D. Dell'Anna and A. Jamshidnejad, "Evolving fuzzy logic systems for creative personalized socially assistive robots," *Eng. Appl. Artif. Intell.*, vol. 114, Sep. 2022, Art. no. 105064.

[12] M. S. Gharajeh and H. B. Jond, "Hybrid global positioning system-adaptive neuro-fuzzy inference system based autonomous mobile robot navigation," *Robot. Auto. Syst.*, vol. 134, Dec. 2020, Art. no. 103669.

[13] M. Samadi Gharajeh and H. B. Jond, "An intelligent approach for autonomous mobile robots path planning based on adaptive neuro-fuzzy inference system," *Ain Shams Eng. J.*, vol. 13, no. 1, Jan. 2022, Art. no. 101491.

[14] M. H. Haider, Z. Wang, A. A. Khan, H. Ali, H. Zheng, S. Usman, R. Kumar, M. U. M. Bhutta, and P. Zhi, "Robust mobile robot navigation in cluttered environments based on hybrid adaptive neuro-fuzzy inference and sensor fusion," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9060–9070, Nov. 2022.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN Conf.*, 1995, pp. 1942–1948.

[16] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–72, Jul. 1992.

[17] B. Akay and D. Karaboga, "Artificial bee colony algorithm for large-scale problems and engineering design optimization," *J. Intell. Manuf.*, vol. 23, no. 4, pp. 1001–1014, Aug. 2012.

[18] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[19] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.

[20] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[21] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.

[22] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowl.-Based Syst.*, vol. 191, Mar. 2020, Art. no. 105190, doi: 10.1016/j.knosys.2019.105190.

[23] A. E. Ezugwu, J. O. Agushaka, L. Abualigah, S. Mirjalili, and A. H. Gandomi, "Prairie dog optimization algorithm," *Neural Comput. Appl.*, vol. 34, no. 22, pp. 20017–20065, Nov. 2022.

[24] X. Li, K. Xing, and Q. Lu, "Hybrid particle swarm optimization algorithm for scheduling flexible assembly systems with blocking and deadlock constraints," *Eng. Appl. Artif. Intell.*, vol. 105, Oct. 2021, Art. no. 104411.

[25] L. Abualigah and M. Alkhrabsheh, "Amended hybrid multi-verse optimizer with genetic algorithm for solving task scheduling problem in cloud computing," *J. Supercomput.*, vol. 78, no. 1, pp. 740–765, Jan. 2022.
[26] H. Li, X. Li, and L. Gao, "A discrete artificial bee colony algorithm for the distributed heterogeneous no-wait flowshop scheduling problem," *Appl. Soft Comput.*, vol. 100, Mar. 2021, Art. no. 106946.
[27] T. Jiang and C. Zhang, "Application of grey wolf optimization for solving combinatorial problems: Job shop and flexible job shop scheduling cases," *IEEE Access*, vol. 6, pp. 26231–26240, 2018.
[28] M. Abdel-Basset, R. Mohamed, M. Abouhawwash, R. K. Chakrabortty, and M. J. Ryan, "EA-MSCA: An effective energy-aware multi-objective modified sine-cosine algorithm for real-time task scheduling in multiprocessor systems: Methods and analysis," *Exp. Syst. Appl.*, vol. 173, Jul. 2021, Art. no. 114699.
[29] W. Yankai, W. Shilong, L. Dong, S. Chunfeng, and Y. Bo, "An improved multi-objective whale optimization algorithm for the hybrid flow shop scheduling problem considering device dynamic reconfiguration processes," *Exp. Syst. Appl.*, vol. 174, Jul. 2021, Art. no. 114793.
[30] I. Attiya, M. Abd Elaziz, and S. Xiong, "Job scheduling in cloud computing using a modified Harris hawks optimization and simulated annealing algorithm," *Comput. Intell. Neurosci.*, vol. 2020, Mar. 2020, Art. no. 3504642.
[31] S. N. Makhadmeh, M. A. Al-Betar, K. Assaleh, and S. Kassaymeh, "A hybrid white shark equilibrium optimizer for power scheduling problem based IoT," *IEEE Access*, vol. 10, pp. 132212–132231, 2022.
[32] Z. Wang, H. Duan, and X. Zhang, "An improved greedy genetic algorithm for solving travelling salesman problem," in *Proc. 5th Int. Conf. Natural Comput.*, 2009, pp. 374–378.
[33] Y. Luo, W. Ding, and B. Zhang, "Optimization of task scheduling and dynamic service strategy for multi-UAV-enabled mobile-edge computing system," *IEEE Trans. Cognit. Commun. Netw.*, vol. 7, no. 3, pp. 970–984, Sep. 2021.
[34] P. Zhang, J. Wang, Z. Tian, S. Sun, J. Li, and J. Yang, "A genetic algorithm with jumping gene and heuristic operators for traveling salesman problem," *Appl. Soft Comput.*, vol. 127, Sep. 2022, Art. no. 109339.
[35] Y. Liu and B. Cao, "A novel ant colony optimization algorithm with levy flight," *IEEE Access*, vol. 8, pp. 67205–67213, 2020.
[36] B. A. S. Emambocus, M. B. Jasser, M. Hamzah, A. Mustapha, and A. Amphawan, "An enhanced swap sequence-based particle swarm optimization algorithm to solve TSP," *IEEE Access*, vol. 9, pp. 164820–164836, 2021.
[37] S. K. Hadia, A. H. Joshi, C. K. Patel, and Y. P. Kosta, "Solving city routing issue with particle swarm optimization," *Int. J. Comput. Appl.*, vol. 47, no. 15, pp. 30–38, Jun. 2012.
[38] A. Ometov, A. Mezina, and J. Nurmi, "On applicability of imagery-based CNN to computational offloading location selection," *IEEE Access*, vol. 11, pp. 2433–2444, 2023.
[39] R. Zhang, C. Zhang, Z. Cao, W. Song, P. S. Tan, J. Zhang, B. Wen, and J. Dauwels, "Learning to solve multiple-TSP with time window and rejections via deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 1325–1336, Jan. 2023.
[40] D. Scott, S. G. Manyam, D. W. Casbeer, and M. Kumar, "A Lagrangian algorithm for multiple depot traveling salesman problem with revisit period constraints," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 1, pp. 690–702, Jan. 2023.
[41] C.-Y. Yang and T.-H. S. Li, "Keyframe-based simultaneous localization and mapping using LiDAR and camera integration and its application to autonomous mobile robot," in *Proc. Conf. ARIS*, Taipei, Taiwan, Aug. 2022, pp. 1–10.
[42] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision With the OpenCV Library*. Sebastopol, CA, USA: O'Reilly Media, 2008.
[43] M.-K. Hu, "Visual pattern recognition by moment invariants," *IEEE Trans. Inf. Theory*, vol. IT-8, no. 2, pp. 179–187, Feb. 1962.
[44] Z. Huang and J. Leng, "Analysis of Hu's moment invariants on image scaling and rotation," in *Proc. 2nd Int. Conf. Comput. Eng. Technol.*, 2010, pp. 476–480.
[45] D. Kukolj, "Design of adaptive Takagi–Sugeno–Kang fuzzy models," *Appl. Soft Comput.*, vol. 2, no. 2, pp. 89–103, Dec. 2002.
[46] C.-S. Chiu, Y.-L. Ouyang, and C.-Y. Ku, "Terminal sliding mode control for maximum power point tracking of photovoltaic power generation systems," *Sol. Energy*, vol. 86, no. 10, pp. 2986–2995, Oct. 2012.
[47] F. Piltan and N. B. Sulaiman, "Review of sliding mode control of robotic manipulator," *World Appl. Sci. J.*, vol. 18, no. 12, pp. 1855–1869, 2012.
[48] F. T. Soares, N. L. Cappelli, A. P. Garcia, and C. K. Umezu, "Fuzzy control applied to an electrical power generation system mounted on tractors for driving of agricultural implements," *Engenharia Agrícola*, vol. 36, no. 5, pp. 846–857, Oct. 2016.
[49] C.-D. Liang, M.-F. Ge, Z.-W. Liu, X.-S. Zhan, and J. H. Park, "Predefined-time stabilization of T–S fuzzy systems: A novel integral sliding mode-based approach," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 10, pp. 4423–4433, Oct. 2022.
[50] J. Boo and D. Chwa, "Fuzzy integral sliding mode observer-based formation control of mobile robots with kinematic disturbance and unknown leader and follower velocities," *IEEE Access*, vol. 10, pp. 76926–76938, 2022.
[51] Z. Yang and H. Ren, "Feature extraction and simulation of EEG signals during exercise-induced fatigue," *IEEE Access*, vol. 7, pp. 46389–46398, 2019.
[52] Y. Liu, Y. Mu, K. Chen, Y. Li, and J. Guo, "Daily activity feature selection in smart homes based on Pearson correlation coefficient," *Neural Process. Lett.*, vol. 51, no. 2, pp. 1771–1787, Jan. 2020.
[53] M. Murtaza, M. Sharif, M. Yasmin, M. Fayyaz, S. Kadry, and M. Y. Lee, "Clothes retrieval using M-AlexNet with mish function and feature selection using joint Shannon's entropy Pearson's correlation coefficient," *IEEE Access*, vol. 10, pp. 115469–115490, 2022.

YU-TING SU received the B.S. degree from the Department of Electrical Engineering, National United University, Miaoli, Taiwan, in 2010, and the M.S. degree from the Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung, Taiwan, in 2012. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan. His research interests include metaheuristic algorithms, artificial intelligence control systems, robotic manipulator, and home service robot.

TZUU-HSENG S. LI (Member, IEEE) received the B.S. degree from the Tatung Institute of Technology, Taipei, Taiwan, in 1981, and the M.S. and Ph.D. degrees in electrical engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, in 1985 and 1989, respectively.

From 1996 to 2009, he was a Researcher with the Engineering and Technology Promotion Center, National Science Council, Tainan. From 1999 to 2002, he was the Director of Electrical Laboratories, NCKU. From 2009 to 2012, he was the Dean of the College of Electrical Engineering and Computer Science, National United University, Miaoli City, Taiwan. From 2009 to 2018, he was the Vice President of the Federation of International Robot-Soccer Association. Since 1985, he has been with the Department of Electrical Engineering, NCKU, where he is currently a Distinguished Professor. He has been the Director of the Center for Intelligent Robotics and Automation, NCKU, since 2014. His research interests include artificial and/or biological intelligence and applications, fuzzy system and control, home service robots, humanoid robots, mechatronics, 4WIS4WID vehicles, and singular perturbation methodology. He was elevated to CACS Fellow and RST Fellow, in 2008 and 2018, respectively. He was a recipient of the Outstanding Automatic Control Award from the Chinese Automatic Control Society (CACS), Taiwan, in 2006; the Outstanding Research Award from the Ministry of Science and Technology, in 2016; and the Outstanding Robotics Engineering Award from the Robotics Society of Taiwan (RST), in 2017. He was a Technical Editor of the IEEE/ASME TRANSACTIONS ON MECHATRONICS and an Associate Editor of the *Asia Journal of Control*. He is an Editor-in-Chief of *The International Journal of Robotics Research* and an Associate Editor of the *International Journal of Electrical Engineering*, the *International Journal of Fuzzy Systems*, and the IEEE TRANSACTIONS ON CYBERNETICS. He was elected as the President of the CACS, from 2008 to 2011, and the RST, from 2012 to 2015.

EN-HAUH LIU received the B.S. and M.S. degrees from the Department of Electrical Engineering, National Cheng Kung University (NCKU), Tainan, Taiwan, in 2020 and 2022, respectively. He is currently a Software Engineer with Novatek Microelectronics Corporation, Taiwan. His research interests include fuzzy systems, metaheuristic algorithms, humanoid robots, service robots, robot motion control, manipulator motion planning, and bin packing optimization.

HSU-MING TSAO received the B.S. degree from the Department of Mechanical Engineering, National Cheng-Kung University (NCKU), Tainan, Taiwan, in 2021. He is currently pursuing the M.S. degree with the Department of Electrical Engineering. His research interests include service robot, robotic manipulator, motion control, and mechanical drawing.

YU-HSIUNG WU received the B.S. degree from the Department of Electrical Engineering, National University of Kaohsiung (NUK), Kaohsiung, Taiwan, in 2021. He is currently pursuing the M.S. degree in program on AI robotics with National Cheng Kung University (NCKU). His research interests include mobile robot, robotic manipulator, artificial intelligence, and the Internet of Things.

CHENG-YEH YANG received the B.S. degree from the Department of Electrical Engineering, National Cheng Kung University (NCKU), Tainan, Taiwan, in 2020. He is currently a Firmware Engineer with MediaTek Inc., Taiwan. His research interests include service robot, computer vision, visual-LiDAR SLAM, and system modeling.