Fuzzy Logic optimized controller for a commercial Greenhouse

Sivasharmini Ganeshamoorthy
Faculty of Engineering Environment and Computing
Coventry University
Coventry, England
ganeshamos@uni.coventry.ac.uk

***Abstract*— Agriculture is currently in the phase of transformation to cope with the climate change problems. Therefore, people have started to focus on autonomous greenhouse especially to control the climate. The aim of this is to generate Fuzzy Logic Controller (FLC) to control the climate and irrigation parameters such as temperature, humidity, soil moisture, light and PH. In this, membership functions parameters and fuzzy rules are optimized through Genetic Algorithm (GA)**

Keywords*— *Agriculture, Genetic Algorithm, Greenhouse, Fuzzy logic controller, Irrigation, PH,

I. INTRODUCTION

The world is currently in the stage of transformation due to increasing population and new technologies. Most importantly, people have started to focus on healthy dietary habits by consuming fresh products. This not only helps them to maintain their fitness but also helps to manage their busy working schedules. However, adverse weather condition, lack of suitable lands for farming and political ignorance are acting as a barrier for open field cultivation. Therefore, Agriculture which plays a strategic role in country's economy has started to seek for new opportunities to improve its system. So taking all these factors into account farmers have started to shift from the traditional open cultivation to 'Greenhouse' to overcome these problems. Greenhouse can be operated manually and by smart technologies. Nevertheless, smart greenhouse are preferred to avoid manual errors, maintain the high quality, overcome scarcity of resources and save energy, money and time. This smart greenhouse has been built using artificial intelligence techniques. The aim of this is to generate a favourable microclimate for the plant through controlling the sunlight, humidity, soil moisture, soil PH and temperature that is appropriate for growth of the plants. Heaters, shade screens, irrigation system and evaporative cooling are some of the advanced methods used in greenhouse. Moreover, number of control strategies (eg: Fuzzy Logic Controller (FLC), Neural Network and Adaptive Predictive Control) have been used for the optimization of greenhouse microclimate.

After, gaining the background knowledge from the existing research papers this study aims to generate an autonomous greenhouse. In this, it is controlled with a Fuzzy Logic controller to maintain microclimate and irrigation. Furthermore, this study has proposed the genetic algorithm technique to optimize the greenhouse fuzzy logic controller. The purpose of Fuzzy logic controller is to control ambient temperature (temperature_error), relative humidity, light intensity, soil moisture and soil PH level. These parameters can be controlled by the actuators such as heating, cooling, irrigation, lighting and PH control systems.

II. LITERATURE REVIEW

Creation of autonomous greenhouse can be built through many artificial intelligence techniques. Fuzzy logic controller among them is adopted to deal with non-linearity of mutual effect parameters such as temperature and humidity (Salim, 2017). Moreover, it has helped to maintain reliable and high accurate results for actuators with less energy consumption. The study conducted by Ali *et al*,.(2016) and Ali *et al*,.(2018) have used fuzzy logic controller to control inside temperature and relative humidity through actuators such as heating, ventilating, humidifying and dehumidifying systems. Another study (Revathi and Sivakumaran, 2016) have only focused on temperature through adopting basic fuzzy logic controller and controlled by heating system. In this, temperature is optimized through Particular Swarm Optimization (PSO) technique. On the contrary, Syam *et al*., (2015) have used watering to control the green house temperature and humidity. In this study, the temperature and humidity are controlled by FLC through setting up the watering time duration for the irrigation systems.

In contrast, from the above studies Alpay and Erdem (2019) have considered four parameters: temperature, relative humidity, soil moisture and light intensity in their studies. Therefore, in this study fuzzy logic controller has been used through implementing five control units for heating, cooling, irrigation, shading and lighting. In this, heating and cooling actuators controlled temperature and relative humidity. Irrigation actuator controlled soil moisture and relative humidity whereas shading controlled light intensity and temperature. Moreover, lighting controlled light intensity. Chauhan and Gupta (2019) have conducted a similar study however, have considered an additional parameter which is soil PH level. To measure this special parameter authors have used PH sensor. The PH level in this investigation was compensated through turning on and off the PH motor pump. Differently, Gultom *et al*., (2017) have focused on humidity and PH level. Scholars in this have used water sprinkle system to control the humidity whereas neutralizing automatic system to compensate the values of PH.

III. GREENHOUSE DESIGN

The actual greenhouse has not been manually constructed however, in order to construct the greenhouse, it needs to be designed in an adaptable format. For this FLC system, there are five key inputs such as temperature (temperature error), relative humidity, light intensity, soil moisture and soil PH level. In this, PH is measured through soil PH sensor, humidity and temperature are measure with DHT11 sensor, soil moisture is measured through soil moisture sensor and light intensity is measured through LDR. During this , all the measurements resulted from the sensors are transferred to the actuators such as heaters, cooling fans, lighting (lamps) and water pump ,PH up motor pump and PH down motor pump. Number of sensors differs according to the size of the greenhouse. Therefore, actuators respond in accordance to the sensor(s) of particular area within the greenhouse. For an example when the temperature increases in Area1, the cooling fan in Area 1 will turn on automatically whereas there will not be any chances noticed in other areas.

IV. Fuzzy controller

Fuzzy Logic Controller (FLC) was generate with the aid of MATLAB Fuzzy Logic Toolbox. This Fuzzy Controller is shown in Figure 1. The main benefit of fuzzy control is creating rules using human expert knowledge. Initially, the choices of linguistic input and out variables need to be defined to commence the design. Following that, crisp inputs need to be converted to fuzzy sets. Then, the rules for the input and out variables need to be set. Next, the fuzzy inferences is accomplished with the aid of fuzzy rules. Finally, appropriate defuzzification needs to be performed to gain the crisp control value which will then inserted to the actuators. Each steps are briefly discussed below:

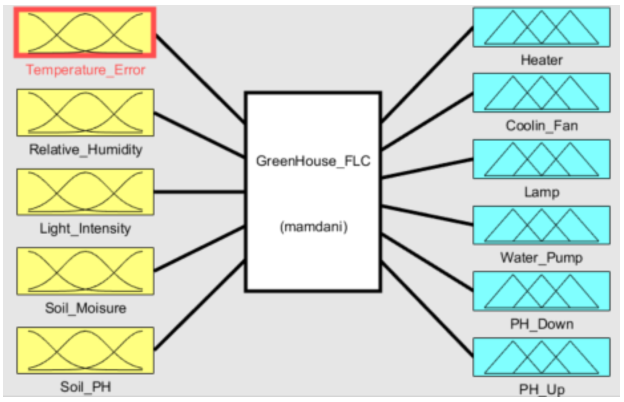


Figure 1: Greenhouse Fuzzy logic controller (Generated using MATLAB Fuzzy Toolbox)

A. *Fuzzy Logic Controller -Inputs*

FLC in this study has total of five inputs such as temperature_error, relative humidity, light intensity, soil moisture and soil PH.

1. Temperature_Error

In the greenhouse, the appropriate temperature to grow plants is 24-32 °C. This is called as temperature set point (Tset). However, the inside temperature (Tinside) in greenhouse can be affected by the external factors within the range of -10- to 40 °C, where these values are measured from the sensors. Therefore, to control the temperature, temperature_error will be calculated using the formula given below:

Temp_error = Tsetpoint−Tinside (1)

The universe of discourse for the Temperature_error is between -10 to 10. This input has 5 linguistic values which are High Negative(HN),Low Negative(LN),Zero(Z),Low positive (LP) and High Positive(HP).As shown in Figure 2, high negative and high positive have trapezoid membership functions(MFs) whereas low negative, zero, low positive have triangle as MFs. These membership functions reflect the temperature_error occurred inside the greenhouse. For an example if the inside temperature (Tinside) increases than the Tsetpoint, then the temperature error will be negative. Vice Versa if the inside temperature (Tinside) decreases than the Tsetpoint then the error will be positive.

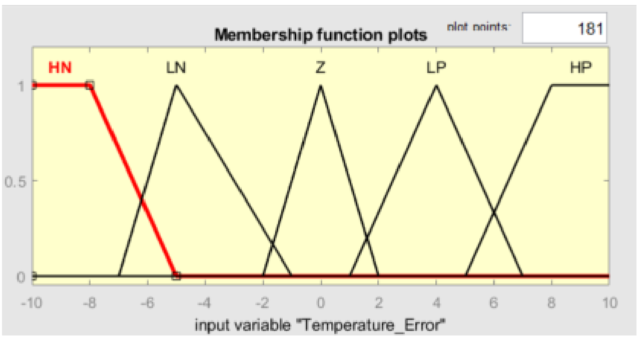


Figure 2: Membership function of Temperature_Error

2. Relative Humidity

Relative humidity is a percentage of water vapour in the air that could hold in a particular temperature. In general this is measured in percentage and the universal discourse range is between is 0 to 100.

This input has 5 linguistic values such as Very Low(VL), Low(L), Medium(M), High(H) and Very High(VH).As shown in Figure 3, very high and very low have trapezoid membership functions(MFs) whereas high, medium, low have triangle as MFs.

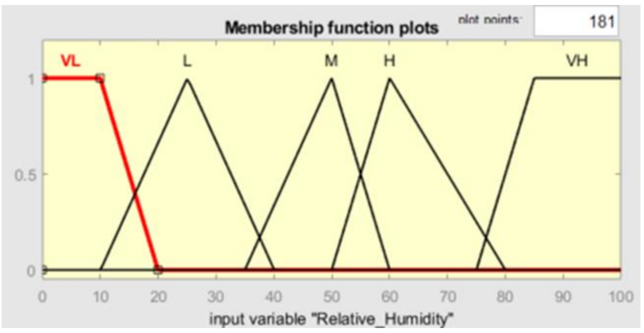


Figure 3: Membership function of Relative_Humidity

3. Soil Moisture

Water exist in the gaps between the soils particles can be defined as soil moisture. This is measured in percentage and the universal discourse range is 0 to 100.

This input has 5 linguistic values such as Very Low (VL), Low (L), Medium (M), High (H) and Very High (VH). As shown in Figure 4, very high and very low have trapezoid membership functions (MFs) whereas high, medium, low have triangle as MFs.

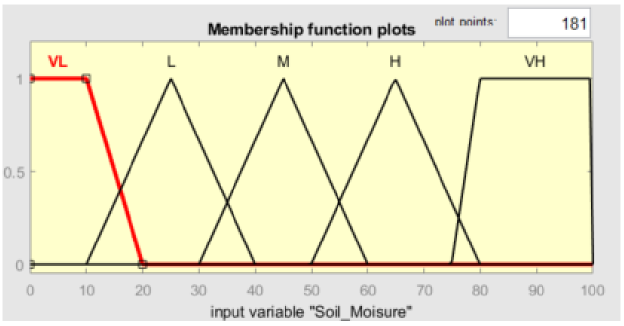


Figure 4: Membership function of Soil_Moisture

4. Light Intensity

The amount of light formed by a lamp is called light intensity. This is measured in lux which is a unit of illumination. The universal discourse range is 0 to 20000 (in FLC it has scaled as 0-2)

This input has 5 linguistic values such as Very Low (VL), Low (L), Medium (M), High (H), and Very High (VH). As shown in Figure 5, very high and very low have trapezoid membership functions (MFs) whereas high, medium, low have triangle as MFs.

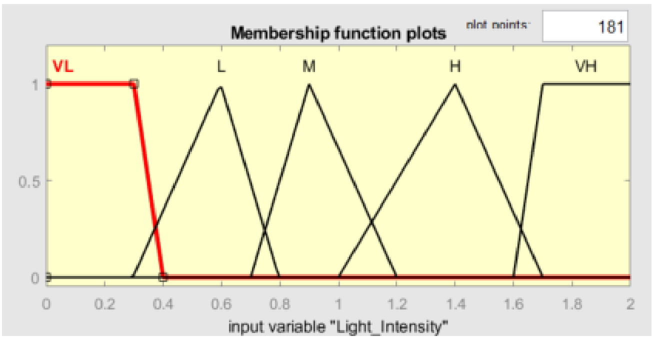


Figure 5: Membership function of Light_Intensity

5. Soil PH

The degree of acidity or alkalinity of the soil is defined as soil PH. The range of PH is between 0 and 14. When the PH value is less than 7, the liquid is acidic whereas when the PH is greater than 7, the liquid is alkaline. The universal discourse range is 0 to 14. This input has 5 linguistic values Such as Strong Acidity (SA), Weak Acidity (WA), Neutral, Weak Alkalinity and Strong Alkalinity. As shown in Figure 6, strong acidity and strong alkalinity have trapezoid membership functions (MFs) whereas weak acidity, neutral, weak alkalinity have triangle as MFs.

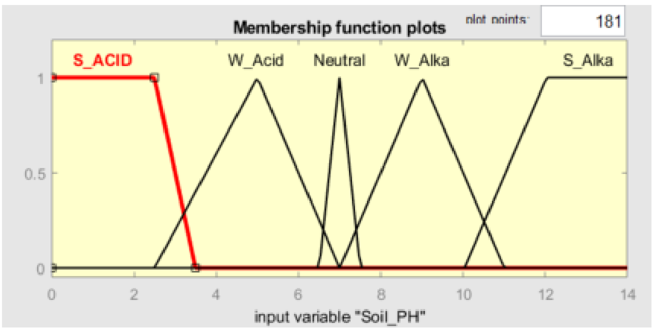


Figure 6: Membership function of Soil_PH

B. Fuzzy Logic Controller- Outputs

The FLC in this study contains six actuators such as cooling fan, heaters, lighting (lamp), water pump, PH down motor pump and PH up motor pump.

1. Cooling Fan

This is measured in μm. The universe of discourse for the cooling fan is 0-25. This output has 5 linguistic values such as Very Low (VL), Low (L), Medium (M), High (H), and Very High (VH). As shown in Figure 7, very high and very low have trapezoid membership functions (MFs) whereas high, medium, low have triangle as MFs.

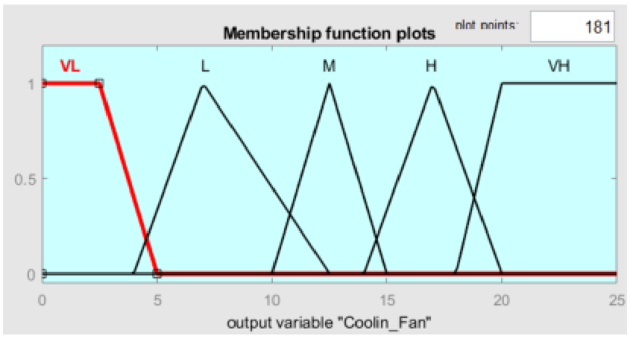


Figure 7: Membership function of Cooling_Fan

2. Heater

This is measured in kW. The universe of discourse for the heater is 0-10. This output has 5 linguistic values such as Very Low (VL), Low (L), Medium (M), High (H) and Very High (VH). As shown in Figure 8, very high and very low have trapezoid membership functions (MFs) whereas high, medium, low have triangle as MFs.

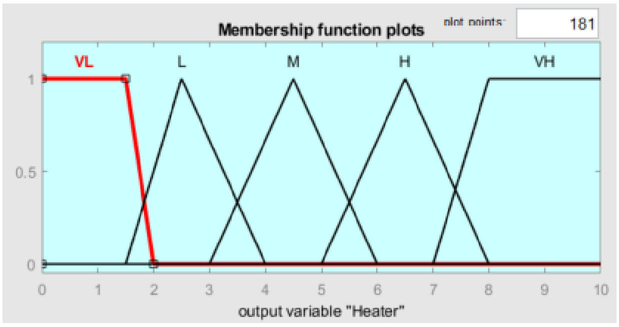


Figure 8: Membership function of Heater

3. Lighting (lamp)

This is measured in lux. The universe of discourse for the lighting is 0-20000. This output has 5 linguistic values such as Very Low (VL), Low (L), Medium (M), High (H) and Very High (VH). As shown in Figure 9, very high and very low have trapezoid membership functions (MFs) whereas high, medium, low have triangle as MFs.

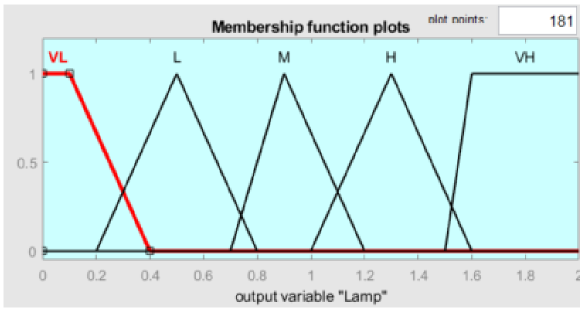


Figure 9: Membership function of lighting (lamp)

4. Water pump

This is measured in time duration (seconds). The universe of discourse for the water pump is 0-300. This output has 5 linguistic values such as Very Low (VL), Low (L), Medium (M), High (H), and Very High (VH). As shown in Figure 10, very high and very low have trapezoid membership functions (MFs) whereas high, medium, low have triangle as MFs.

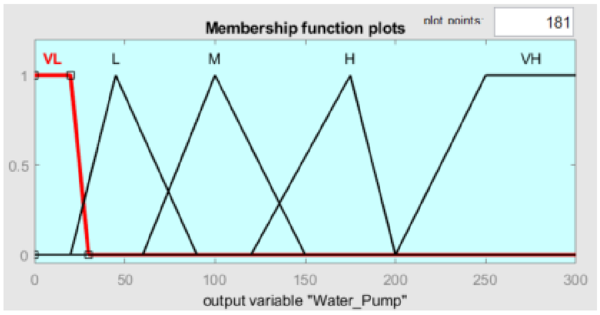


Figure 10: Membership function of Water Pump

5. PH up motor pump

This is measured in time duration (seconds). The universe of discourse for the water pump is 0-150. This output has 3 linguistic values which are Off, Medium (M) and Low (L). This is used to maintain the soil PH level 6.5-7.5. Therefore, if the PH value is less than 7 (acid), PH up motor pump will be turned on. The time duration of this motor pump depends on the strength of the acidity. As shown in Figure 11, off and high have trapezoid membership functions (MFs) whereas medium has a triangle.

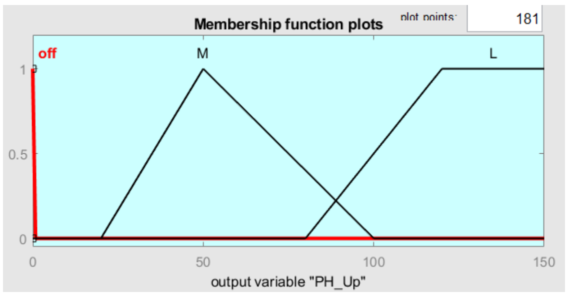


Figure 11: Membership function of PH_Up motor pump

6. PH down motor pump

This is measured in time duration (seconds). The universe of discourse for the water pump is 0-150. This output has 3 linguistic values which are Off, Medium (M) and Low (L). This is used to maintain the soil PH level 6.5-7.5. Therefore, if the PH value is greater than 7 (alkaline), PH down motor pump will be turned on. The time duration of this motor pump depends on the strength of the alkaline. As shown in Figure 12, off and high have trapezoid membership functions (MFs) whereas Medium has a triangle.

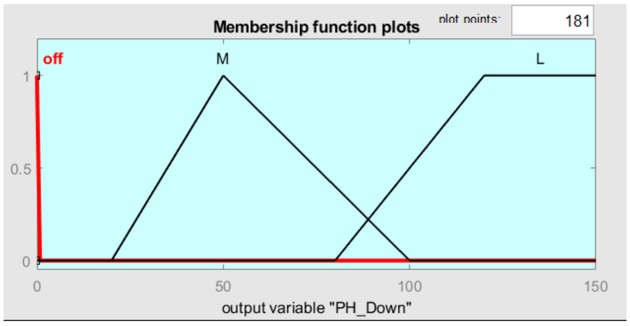


Figure 12: Membership function of PH_Down motor pump

C. Fuzzy Logic Inference

Fuzzy inference is a procedure that converts inputs into output with the aid of theory of fuzzy sets. Mamdani Fuzzy inference method is the most adopted fuzzy inference technique in comparison to the other methods such as Sugeno (TSK) fuzzy model. Mamdani fuzzy inference model comprises four important steps. The first step is the process of fuzzying the crisp input value. Secondly, rule inference and thirdly it is about aggregating the rule outputs. Final step is converting fuzzy values to crisp value through defuzzification which will be performed using centroid (Centre of Gravity). The rule evaluation method AND and OR are setup for the purpose of using minimum and maximum membership values correspondingly. Moreover, rule implication and aggregation methods are setup to maximum and minimum correspondingly. The following section explains these four steps in more detail.

1. Fuzzification
2. Fuzzy Rules
3. Rule Evaluation, Inference and Implication
4. Defuzzification

1. Fuzzification

Fuzzication is a process of identifying the degree of membership function for the crisp input. This is explained through an example. As shown in the Figure 13 when the relative humidity is 52 percentage, it falls into two membership functions: medium and high (figure x). For the crisp value 52, the related fuzzy terms are 0.2 (High) and 0.75 (Medium).

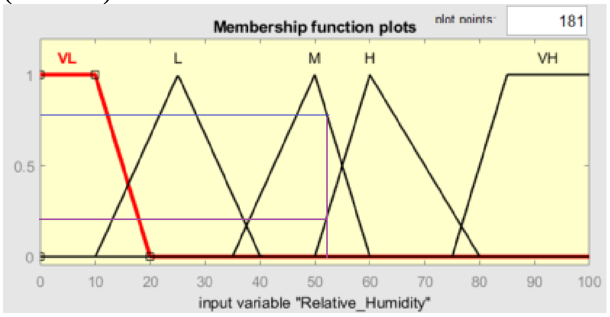


Figure 13: Example of Fuzzification

2. Fuzzy Rules

The purpose of fuzzy rules is to convert the fuzzy inputs to fuzzy outputs with the aid of rule inference, implication and aggregation. These rules are used to link the inputs with AND and OR statements which have an impact on one or more outputs. However, this study has only adopted the rules linked to the AND statements. In this study, the FLC contains 65 fuzzy rules in total. These rules are categorized by the behaviour they control. For an example, there are rules connected to inputs relative humidity and soil moisture where the output is water pump duration. The rules for this inputs and outputs are shown in the Table 1 below. For an example, one of the rule is if relative humidity is very low and soil moisture is very low then the water pump duration is very high. The entire list of fuzzy rules is presented in appendix.

RH SM	VL	L	M	H	VH
VL	VH	H	H	M	VL
L	VH	H	M	M	L
M	H	H	M	M	M
H	H	M	M	L	VL
VH	M	M	L	L	VL

Table 1: Rules related to Relative_Humidity and Soil_Moisture

3. Rule Evaluation, Inference and Implication

Rule evaluation is the procedure of achieving outputs fuzzy set for a given set of input values through using the fuzzy rules. For an example, when the input parameters are relative humidity and soil moisture through the controller of water time duration, the applicable rules are listed in the Table 1. When the input values for relative humidity is 52%, the related fuzzy terms are 0.2 (High) and 0.75 (Medium) (See Figure 14 (a)). On the other hand, when the soil moisture is 78% the related fuzzy terms are 0.1 (High) and 0.45 (Very High) (See Figure 14(b)). The AND operator has been used to build the list of rules with the purpose of setting the minimum values. In other words, finding the lowest value, among the linguistic values that satisfies the rules (See Table 2)

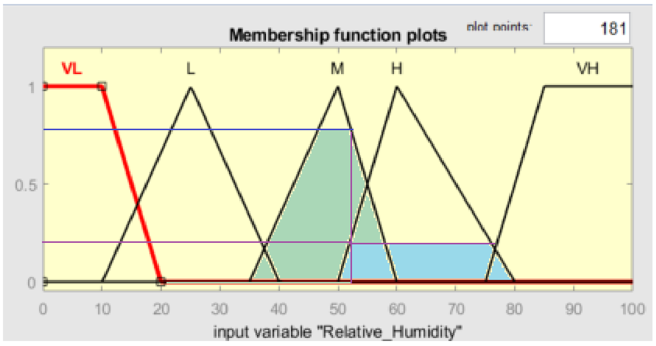


Figure 14(a): MF activation of Relative Humidity

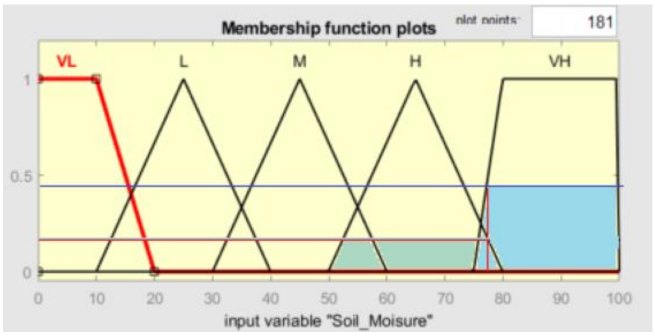


Figure 4(b): MF activation Soil_Moisture

	M=0.75	H=0.2
H=0.1	Min(0.75,0.1) = 0.1(M)	Min(0.2,0.1) = 0.1(L)
VH=0.45	Min(0.75,0.45) = 0.45(L)	Min(0.2,0.7) = 0.2(L)

Table 2: Rule Evaluation

The aggregation is achieved through taking the maximum values of derived results where the Water Pump Time equal to 0.1(M), 0.45(L). Following this, these identified values will be mapped in the output variable which is the water pump. As shown in the Figure 15, aggregation is preformed through the combination of membership functions [0.1(M), 0.45(L)]

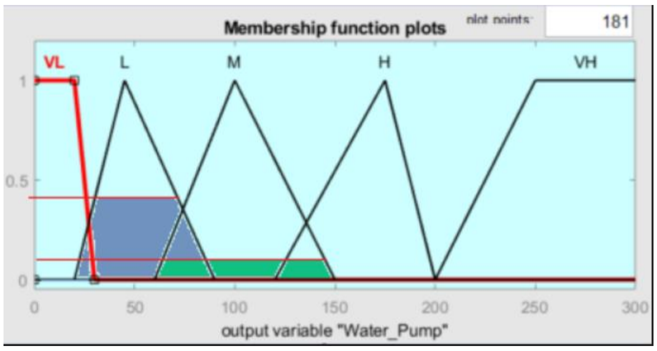


Figure 15: Aggregation of output membership

4. Defuzzification

Defuzzification is the procedure of transforming the fuzzy values into crisp values which is then passed to actuators. In this study as stated before centre of gravity (COG) defuzzification method was adopted. In this study, aggregation which is continuous is estimated through calculating COG over the sample of evenly distributed discrete points (2).

$$COG = \Sigma\ (wi)\ qi{=}1\ \Sigma\ (wi)\ qi{=}1 \quad (2)$$

Although, the above discussed methods are used to perform fuzzy inference, it can be also achieved through MATLAB Fuzzy Toolbox. The Figure 16 below displays the inference and defuzzification achieved using the same values as discussed in the above example. From the MATLAB Fuzzy Toolbox the obtained COG value is 65.5.

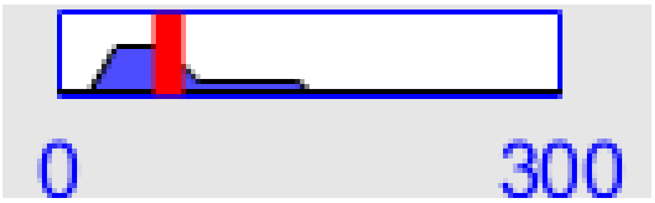


Figure 16: Rule inference results from MATLAB Fuzzy Toolbox

V. OUTPUT BEHAVIOUR ANALYSIS

Surface plots are generated in order to analyse whether the FLC obtains the expected behaviour. These plots represents the defuzzified values response of the controllers (actuators) through inserting one or two input parameters. This is to see the behaviours of the actuators (output) when one or more input parameters are changed.

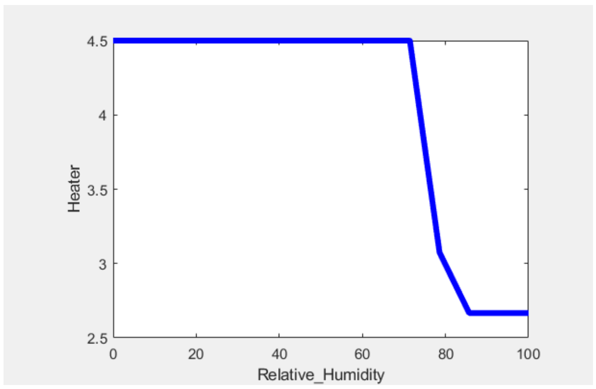


Figure 17: Surface Plot of Relative _Humidity Vs Cooling_Fan

As shown in the Figure 17, when the relative humidity increases within the greenhouse, the heating system slowly starts to response to the input parameter. In other words, heating system will remain high until the relative humidity reaches its desired point. After that, heating system will start to gradually decrease.

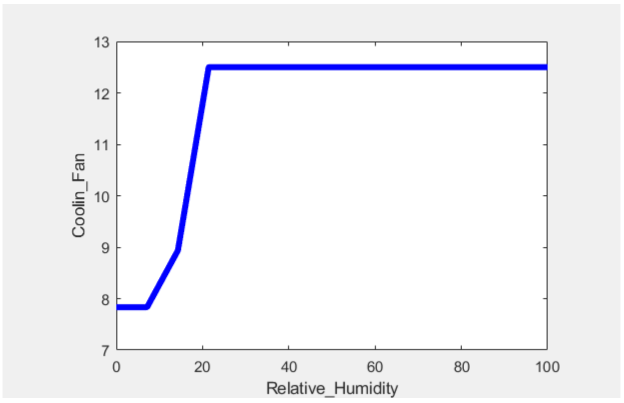


Figure 18: Surface Plot of Relative_Humidity Vs Cooling_Fan

As shown in the Figure 18, when the relative humidity increases within the greenhouse, the cooling system will also increase. Once relative humidity reaches its desired point the cooling system will remain high to maintain high relative humidity.

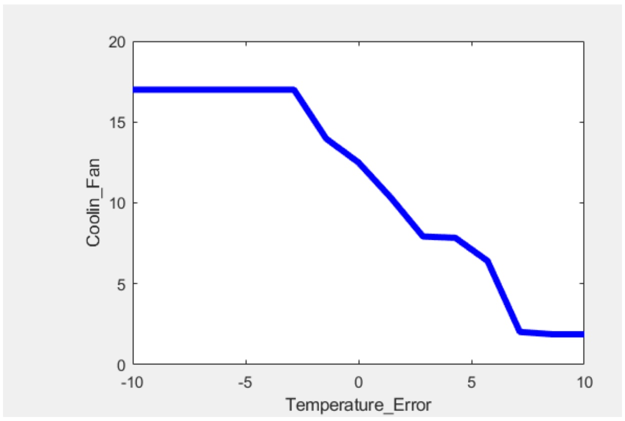


Figure 19: Surface Plot of Temperature_Error Vs Cooling_Fan

As shown in Figure 19, the value for the temperature_error is negative, inside_temperature is greater the set_temperature. Vice versa if the temperature_error is positive inside temperature is lower than the set temperature. If temperature_error changes from negative to positive it means temperature inside is changing from maximum to minimum value. Therefore cooling system will change from higher value to low value.

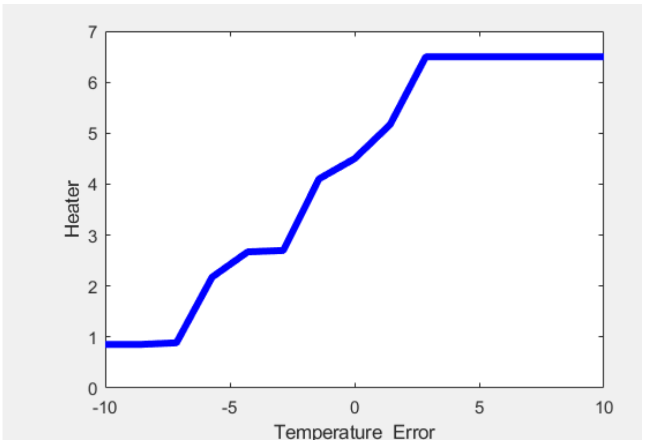


Figure 20: Surface Plot of Temperature_Error Vs Heater

As shown in Figure 20, the value for the temperature_error is negative, inside_temperature is greater the set_temperature. Vice versa if the temperature_error is positive, inside_ temperature is lower than the set_temperature. If temperature_error changes from negative to positive it means temperature inside is changing from maximum to minimum value. Therefore heating system will act in a similar pattern as temperature_error change, where heating will increase when temperature_error increases.

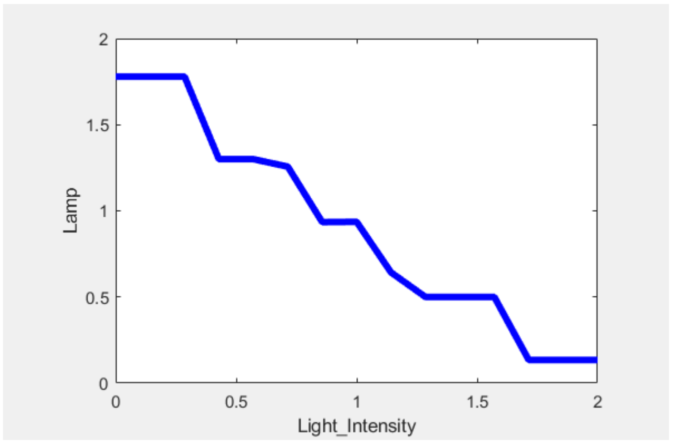


Figure 21: Surface Plot of Light_Intensity Vs Lighting (Lamp)

As shown in the Figure 21, when the light intensity increases the lighting system (lamp) will decrease. For an example, during winter if the light intensity is low lightning system will reach its maximum point to control the greenhouse desired lighting.

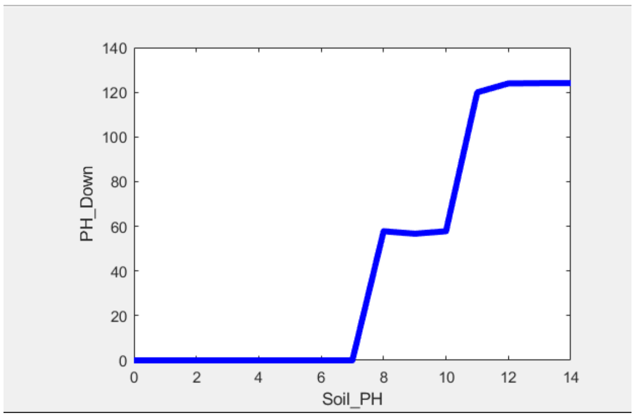


Figure 22: Surface Plot of Soil_PH Vs PH_Down motor pump

If the soil PH exceeds the neutral (PH=7), it is referred as alkaline. To maintain the desired PH level within the greenhouse PH down water pump should be turned on. According to the strength of alkaline PH down motor pump time level will be increased. This is clearly illustrated in the Figure 22, where PH down water pump time duration gradually increases when PH value is between 7 to14.

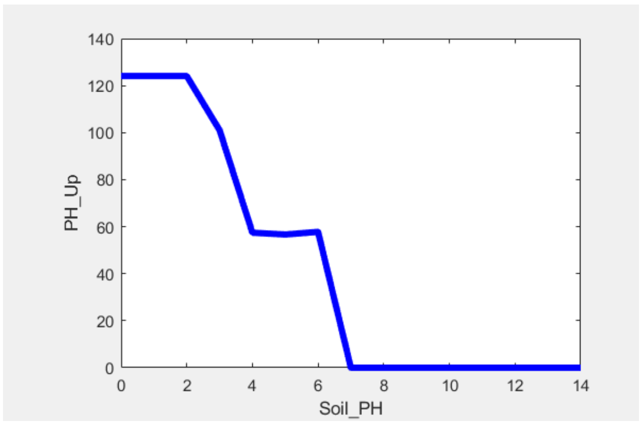


Figure 23: Surface Plot of Soil_PH Vs PH_Up motor pump

In contrast , if the soil PH is less than neutral (PH=7), it is referred as acid. To maintain the desired PH level within the greenhouse, PH up water pump should be turned on. According to the strength of acid, PH up motor pump time level will be increased. As illustrated in the Figure 23, when the PH value decreases from 7 to 0 (strengthening the acidity) the time duration of the PH up motor pump will increase.

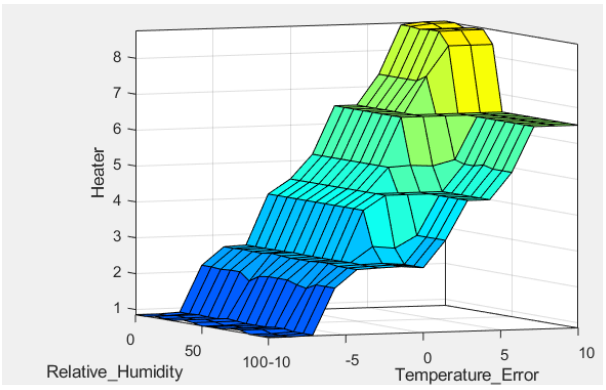


Figure 24: Surface Plot of Temperature_Error Vs Relative_Humidity Vs Heater

As illustrated in the Figure 24, when the temperature_error is negative (inside temperature is maximum) and relative humidity is very low, heating system will be very low. On the other hand, when the temperature_error is positive (inside temperature is minimum) and relative humidity is high, heating system will be high. Therefore, when the temperature_error increases from negative to positive and relative humidity changes from maximum to minimum, it is noticed that heating system gradually increases from minimum to maximum values.

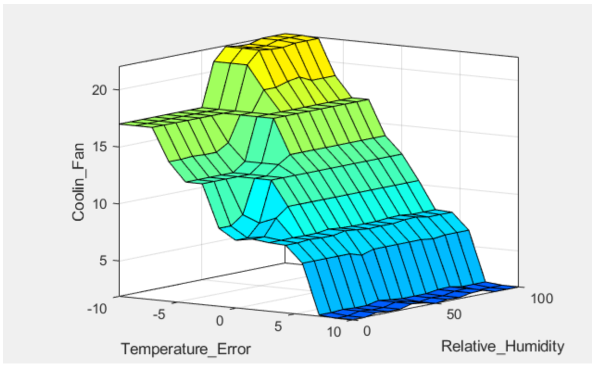


Figure 25: Surface Plot of Temperature_Error Vs Relative_Humidity Vs Cooling_Fan

As illustrated in the Figure 25, when the temperature_error is negative (inside temperature is maximum) and relative humidity is very low, cooling system will be high. On the other hand, when the temperature_error is positive (inside temperature is minimum) and relative humidity is high, cooling system will be very low. Therefore, when the temperature_error increases from negative to positive and relative humidity changes from maximum to minimum, it is noticed that cooling system gradually decreases from maximum to minimum values.

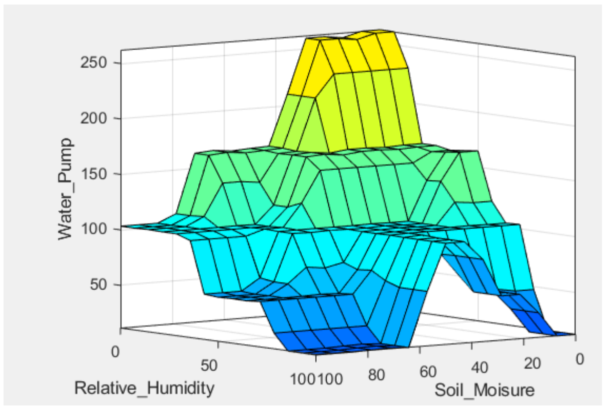


Figure 26: Surface Plot of Soil_Moisture Vs Relative_Humidity Vs Water_Pump

As illustrated in Figure 26 , when the relative humidity and soil moisture are very low, water pump duration will be very high. Vice Versa when the relative humidity and soil moisture are very high, water pump will be very low. Similarly, when the soil moisture is very low and relative humidity is very high, water pump will be very low. In contrast, when the soil moisture is very high and relative humidity is very low, water pump will be medium. When both relative humidity and soil moisture are medium, water pump will be medium as well. Therefore, when the soil moisture increases water pump duration decreases. Likewise, when the relative humidity increases water pump duration decreases.

VI. FLC OPTIMIZATION

Input and output values of MFs and Fuzzy rules are generated with the human expert knowledge. Nevertheless, this will not produce the optimal results. Solution for a problem can be optimized through genetic algorithm. This generic algorithm contains four key steps. Firstly, the problem solution needs to be encoded as chromosome and population size should be defined. Secondly, to evaluate the chromosome performance fitness function needs to be defined. Thirdly, chromosome needs to be selected with the aid of roulette wheel selection to randomly create initial population. Following this, the offspring chromosome are generated through using genetic operators which are crossover and mutation. Finally, new generation of chromosomes are created. This process will be repeated until the stopping criteria is met or desired number of generations are achieved. The sections below explains the procedure of optimizing MF parameters, fuzzy rules through GA algorithm.

A. MF's Parameter optimization- Mamdani model

The initial step to apply the GA is to create chromosome through MFs parameters of inputs and outputs of the FLC. In general, chromosome contains list of binary but in the scenario MFs parameters are defined as string of values. The reason for using real numbers is because when the chromosome is represented in binary the length of the chromosome is long. For an example binary representation of 300 is 0000000100111100.

The Mamdani model is adopted in this study for the creation of FLC. Therefore, length of chromosome can be calculated by summing up all MFs values. Hence, 175 is obtained as the total length of chromosome. The explanation of this calculation is explained below:

In this study, the input parameters are temperature _error, relative humidity, soil moisture, light and soil PH level. Each input includes two trapezoid MFs with four parameters (2x4=8) and three triangles with three parameters (3x3=9).Therefore, the 5 input variables have (17 *5)=85 parameters.

The output variables for this study are heating, cooling fan, lighting (lamp) and water pump. Each output includes two trapezoid MFs with four parameters (2x4=8) and three triangles with three parameters (3x3=9) whereas PH up motor pump and PH down motor pump include two trapezoid (2x4 = 8) and one triangle (1x3=3).Therefore, the length of the output parameters are (17x4)+ (11x2) =90. This derives to the total of 175 parameters which is a combination of input and output parameters. Therefore, each chromosome length will be equal to 175.

The population size for this study is 1000 and fitness function is defined as Mean Square Error (MSE). To calculate the MSE function a dataset has been given. The initial population is randomly created through roulette wheel selection approach. The huge space in roulette wheel is captured by best chromosomes (lowest MSE). Hence, the chromosome wheel space ratio is calculated through the ratio of 1/MSE.

As a next step crossover operator has been applied to the selected chromosomes. There are different types of crossover however, this study has adopted two-point split approach. The split points are chosen randomly and crossover probability is defined as 0.7 as usual. Then, to create the offspring chromosomes mutation operation is used with the probability of 0.001. This approach helps to achieve the local optimum than the global optimum. In this, mutation can be derived by adding or subtracting a small value. At last, these step will be repeated until the stopping criteria is met or desired number of generations are achieved.

B. MF's Parameter optimization- Sugeno Model

The recommended solution for this study is Mandani Model. Nevertheless, the following section discusses how GA is optimized differently while adopting Sugeno Model for the MFs.

In MATLAB Fuzzy logic Toolbox Sugeno Model can only be used for output variables. Therefore, when Sugeno Model is applied to membership functions of output variables, the results will be presented as a single number. This is called 'Singletons'. In this study, if heater is considered it will only have 'turn on' and 'turn off' values rather than having a continuous range of values. Hence, there will be not be any optimization for the output. So the chromosome length will be determined by the total number of the membership of the parameters of the input variables which is 85. This calculation is already explained in the section above. The remaining GA steps will be same as discussed before.

C. Fuzzy Rule Optimization

As mentioned in section above fuzzy rules are generated randomly with the aid of human experts. Hence, there is a need to optimize the results which are not accurate. For this purpose GA are incorporated to optimize the number of rules.

Firstly, it important to identify all potential combination of rules. In this study, for the chosen FLC the potential number of rule combination is 17,578,125. This is created through considering all number of MFs for each input and output and multiply them together.

There are nine variables (temperature_error, relative_ humidity, soil_moisture, light, soil_PH, heating, cooling, lighting(lamp) and water pump) with five parameters in MFs and two variables (PH down motor pump and PH up motor pump) with three parameters in MFs.

Consequently, combination of rules are calculated through
59 x 32 = 17,578,125

Although, there are 17,578,125 rules in total only 3125 rules can be practically implemented and this is achieved from multiplication of inputs (5). Following this, the results for each of these rules can be calculated using multiplying the MFs of outputs which 5625 is (54 x 32).

As an initial step, chromosome length needs to be determined. This is calculated through multiplying the number of number of rules which are correspondence of consequence (3125) and number of outputs of the FLC (6). So the value for the chromosome size is 3125 X6= 18750.

The second step is defining the population size. In this, the population size is 1000 and to evaluate the chromosome the fitness function is defined as MSE. To choose the best chromosome roulette wheel selection is defined. Consequently, to create offspring chromosome genetic operator crossover is performed in the selected chromosomes. In this step, two-point split approach is adopted to reduce the mistake of crossing over different MFs. For an example, cross over in two chromosomes can only happen when it have similar MFs values. In this, as usual cross over probability is 0.7.

The next, genetic operator mutation is obtained through increasing or decreasing by 1 for the chosen genes. Again, in this probability, it is 0.0001. At last, these step will be repeated until the stopping criteria is met or desired number of generations are achieved.

D. Fuzzy Optimum rules and performance

Optimizing the number of rules through GA has produced precise results however, the drawback of this approach is having large number of rules (3125). To overcome this problem, the optimized fuzzy model can be generated with less number of rules and less errors using GA. For this, similar steps will be followed including the usage of fitness function. The fitness function is defined through proposing weights to minimise both number of rules and errors.

$$\boxed{f(S) = w_P MSE + \ w_N \frac{N_s}{N_{ALL}}}$$

Where,
w_P – Weight to prioritize MSE
w_N – Weight to prioritize number of rules
N_s – Number of fuzzy rules in a solution
N_{ALL} - Number of all possible fuzzy rules in a solution

Moreover, in this fitness function, when the aim is to reduce the number of rules, this can be achieved through increasing the value of W_N. On the other hand, when the aim is to reduce the errors, this can be attained through increasing the W_P.

REFERENCES

[1] Chauhan A.K and Gupta.P.K,‘‘Intelligent Greenhouse Environment Monitoring and Automatic Controlling System Using IOT’’, International Journal of Recent Technology and Engineering (IJRTE), vol-7, Issue-6C, April 2019.

[2] J. H. Gultom, M. Harsono, T. D. Khameswara and H. Santoso, "Smart IoT Water Sprinkle and Monitoring System for chili plant," 2017 International Conference on Electrical Engineering and Computer Science (ICECOS), Palembang, 2017, pp. 212-216.

[3] M.S.Salim, ‘‘Design and simulation of intelligent greenhouse climate controller’’, J. of Eng. and Applied Sci. vol-12,Issue-3, 2017, pp. 690-695.

[4] Ö.Alpay and E. Erdem, ‘‘The Control of Greenhouses Based on Fuzzy Logic Using Wireless Sensor Networks’’, International Journal of Computational Intelligence Systems, vol-12, Issue-1, 2019, pp.190-203.

[5] R. Ben Ali, E. Aridhi, M. Abbes and A. Mami, ‘‘Fuzzy logic controller of temperature and humidity inside an agricultural greenhouse’’, 2016 7th International Renewable Energy Congress (IREC), Hammamet, 2016, pp. 1-6.

[6] R.B.Ali, S.Bouadila and A.Mami, ‘‘Development of a Fuzzy Logic Controller applied to an agricultural greenhouse experimentally validated’’, Applied Thermal Engineering,vol- 141, 2018, pp. 798-810.

[7] R.Syam,W.H.Piarah and B.Jaelani, ‘‘Controlling Smart Green House Using Fuzzy Logic Method’’, International Journal on Smart Material and Mechatronics, IJSMM, vol- 2, Issue-2, 2015, pp.116–120.

[8] S. Revathi and N. Sivakumaran, ‘‘Fuzzy Based Temperature Control of Greenhouse’’,vol-49, Issue-1, 2016, pp. 549-554.

APPENDICES

Appendix 1: Membership Functions

Input

Temperature_Error [10,10] ^0C

Membership function	Parameters
HN	(10,10,-8,-5)
LN	(-7,-5,-1)
Z	(-2,0,2)
LP	(1,4,7)
HP	(5,8,10,10)

Relative Humidity [0,100] %

Membership function	Parameters
VL	(0,0,10,20)
L	(10,25,40)
M	(35,50,60)
H	(50,60,80)
VH	(75,85,100,100)

Lighting Intensity [0,2] Lux

Membership function	Parameters
VL	(0,0,0.3,0.4)
L	(0.3,0.6,0.8)
M	(0.7,0.9,1.2)
H	(1,1.4,1.7)
VH	(1.6,1.7,2,2)

Soil Moisture [0-100] %

Membership function	Parameters
VL	(0,0,10,20)
L	(10,25,40)
M	(30,45,60)
H	(50,65,80)
VH	(75,80,100,100)

PH [0-14]

Membership function	Parameters
S_Acid	(0,0,2.5,3.5)
W_Acid	(2.5,5,7)
Neutral	(6.5,7,7.5)
W_Alka	(7,9,11)
S_Alka	(710,12,14,14)

Output

Heater [0-10]

Membership function	Parameters
VL	(0,0,1.5,2)
L	(1.5,2.5,4)
M	(3,4.5,6)
H	(5,6.5,8)
VH	(7,8,10,10)

Cooling _Fan [0-25] micron

Membership function	Parameters
VL	(0,0,2.5,5)
L	(4,7,12.5)
M	(10,12.5,15)
H	(14,17,20)
VH	(18,20,25,25)

Lighting (lamp) Power [0-2] x 10^4

Membership function	Parameters
VL	(0,0,0.1,0.4)
L	(0.2,0.5,0.8)
M	(0.7,0.9,1.2)
H	(1,1.3,1.6)
VH	(1.5,1.6,2.2)

Water__Pump_Time duration [0-300] sec

Membership function	Parameters
VL	(0,0,20,30)
L	(20,45,90)
M	(60,100,150)
H	(120,175,200)
VH	(200,250,300,300)

PH_Down_Time duration [0-150] sec

Membership function	Parameters
Off	(0,0,0,0)
Medium	(20,50,100)
High	(80,120,150,150)

PH_Up_Time duration [0-150] sec

Membership function	Parameters
Off	(0,0,0,0)
Medium	(20,50,100)
High	(80,120,150,150)

APPENDIX 2: FUZZY RULES

1. IF (TEMPERATURE_ERROR IS HP) AND (RELATIVE_HUMIDITY IS VL) THEN (HEATER IS VH)(COOLIN_FAN IS VL)
2. IF (TEMPERATURE_ERROR IS HP) AND (RELATIVE_HUMIDITY IS L) THEN (HEATER IS VH)(COOLIN_FAN IS VL)
3. IF (TEMPERATURE_ERROR IS HP) AND (RELATIVE_HUMIDITY IS M) THEN (HEATER IS H)(COOLIN_FAN IS VL)
4. IF (TEMPERATURE_ERROR IS HP) AND (RELATIVE_HUMIDITY IS H) THEN (HEATER IS H)(COOLIN_FAN IS VL)
5. IF (TEMPERATURE_ERROR IS HP) AND (RELATIVE_HUMIDITY IS VH) THEN (HEATER IS H)(COOLIN_FAN IS VL)
6. IF (TEMPERATURE_ERROR IS LP) AND (RELATIVE_HUMIDITY IS VL) THEN (HEATER IS H)(COOLIN_FAN IS L)
7. IF (TEMPERATURE_ERROR IS LP) AND (RELATIVE_HUMIDITY IS L) THEN (HEATER IS H)(COOLIN_FAN IS L)
8. IF (TEMPERATURE_ERROR IS LP) AND (RELATIVE_HUMIDITY IS M) THEN (HEATER IS H)(COOLIN_FAN IS L)
9. IF (TEMPERATURE_ERROR IS LP) AND (RELATIVE_HUMIDITY IS H) THEN (HEATER IS M)(COOLIN_FAN IS L)
10. IF (TEMPERATURE_ERROR IS LP) AND (RELATIVE_HUMIDITY IS VH) THEN (HEATER IS M)(COOLIN_FAN IS L)
11. IF (TEMPERATURE_ERROR IS Z) AND (RELATIVE_HUMIDITY IS VL) THEN (HEATER IS M)(COOLIN_FAN IS L)
12. IF (TEMPERATURE_ERROR IS Z) AND (RELATIVE_HUMIDITY IS L) THEN (HEATER IS M)(COOLIN_FAN IS M)
13. IF (TEMPERATURE_ERROR IS Z) AND (RELATIVE_HUMIDITY IS M) THEN (HEATER IS M)(COOLIN_FAN IS M)
14. IF (TEMPERATURE_ERROR IS Z) AND (RELATIVE_HUMIDITY IS H) THEN (HEATER IS M)(COOLIN_FAN IS M)
15. IF (TEMPERATURE_ERROR IS Z) AND (RELATIVE_HUMIDITY IS VH) THEN (HEATER IS L)(COOLIN_FAN IS M)
16. IF (TEMPERATURE_ERROR IS LN) AND (RELATIVE_HUMIDITY IS VL) THEN (HEATER IS L)(COOLIN_FAN IS M)
17. IF (TEMPERATURE_ERROR IS LN) AND (RELATIVE_HUMIDITY IS L) THEN (HEATER IS L)(COOLIN_FAN IS M)
18. IF (TEMPERATURE_ERROR IS LN) AND (RELATIVE_HUMIDITY IS M) THEN (HEATER IS L)(COOLIN_FAN IS H)
19. IF (TEMPERATURE_ERROR IS LN) AND (RELATIVE_HUMIDITY IS H) THEN (HEATER IS L)(COOLIN_FAN IS H)
20. IF (TEMPERATURE_ERROR IS LN) AND (RELATIVE_HUMIDITY IS VH) THEN (HEATER IS L)(COOLIN_FAN IS H)
21. IF (TEMPERATURE_ERROR IS HN) AND (RELATIVE_HUMIDITY IS VL) THEN (HEATER IS VL)(COOLIN_FAN IS H)
22. IF (TEMPERATURE_ERROR IS HN) AND (RELATIVE_HUMIDITY IS L) THEN (HEATER IS VL)(COOLIN_FAN IS H)
23. IF (TEMPERATURE_ERROR IS HN) AND (RELATIVE_HUMIDITY IS M) THEN (HEATER IS VL)(COOLIN_FAN IS H)
24. IF (TEMPERATURE_ERROR IS HN) AND (RELATIVE_HUMIDITY IS H) THEN (HEATER IS VL)(COOLIN_FAN IS VH)
25. IF (TEMPERATURE_ERROR IS HN) AND (RELATIVE_HUMIDITY IS VH) THEN (HEATER IS VL)(COOLIN_FAN IS VH)
26. IF (LIGHT_INTENSITY IS VL) THEN (LAMP IS VH)
27. IF (LIGHT_INTENSITY IS L) THEN (LAMP IS H)
28. IF (LIGHT_INTENSITY IS M) THEN (LAMP IS M)
29. IF (LIGHT_INTENSITY IS H) THEN (LAMP IS L)
30. IF (LIGHT_INTENSITY IS VH) THEN (LAMP IS VL)
31. IF (RELATIVE_HUMIDITY IS VL) AND (SOIL_MOISURE IS VL) THEN (WATER_PUMP IS VH)
32. IF (RELATIVE_HUMIDITY IS VL) AND (SOIL_MOISURE IS L) THEN (WATER_PUMP IS VH)
33. IF (RELATIVE_HUMIDITY IS VL) AND (SOIL_MOISURE IS M) THEN (WATER_PUMP IS H)
34. IF (RELATIVE_HUMIDITY IS VL) AND (SOIL_MOISURE IS H) THEN (WATER_PUMP IS H)
35. IF (RELATIVE_HUMIDITY IS VL) AND (SOIL_MOISURE IS VH) THEN (WATER_PUMP IS M)
36. IF (RELATIVE_HUMIDITY IS L) AND (SOIL_MOISURE IS VL) THEN (WATER_PUMP IS H)
37. IF (RELATIVE_HUMIDITY IS L) AND (SOIL_MOISURE IS L) THEN (WATER_PUMP IS H)
38. IF (RELATIVE_HUMIDITY IS L) AND (SOIL_MOISURE IS M) THEN (WATER_PUMP IS H)
39. IF (RELATIVE_HUMIDITY IS L) AND (SOIL_MOISURE IS H) THEN (WATER_PUMP IS M)
40. IF (RELATIVE_HUMIDITY IS L) AND (SOIL_MOISURE IS VH) THEN (WATER_PUMP IS M)
41. IF (RELATIVE_HUMIDITY IS M) AND (SOIL_MOISURE IS VL) THEN (WATER_PUMP IS H)
42. IF (RELATIVE_HUMIDITY IS M) AND (SOIL_MOISURE IS L) THEN (WATER_PUMP IS M)
43. IF (RELATIVE_HUMIDITY IS M) AND (SOIL_MOISURE IS M) THEN (WATER_PUMP IS M)
44. IF (RELATIVE_HUMIDITY IS M) AND (SOIL_MOISURE IS H) THEN (WATER_PUMP IS M)
45. IF (RELATIVE_HUMIDITY IS M) AND (SOIL_MOISURE IS VH) THEN (WATER_PUMP IS L)
46. IF (RELATIVE_HUMIDITY IS H) AND (SOIL_MOISURE IS VL) THEN (WATER_PUMP IS M)
47. IF (RELATIVE_HUMIDITY IS H) AND (SOIL_MOISURE IS L) THEN (WATER_PUMP IS M)
48. IF (RELATIVE_HUMIDITY IS H) AND (SOIL_MOISURE IS M) THEN (WATER_PUMP IS M)
49. IF (RELATIVE_HUMIDITY IS H) AND (SOIL_MOISURE IS H) THEN (WATER_PUMP IS L)
50. IF (RELATIVE_HUMIDITY IS H) AND (SOIL_MOISURE IS VH) THEN (WATER_PUMP IS L)
51. IF (RELATIVE_HUMIDITY IS VH) AND (SOIL_MOISURE IS VL) THEN (WATER_PUMP IS VL)
52. IF (RELATIVE_HUMIDITY IS VH) AND (SOIL_MOISURE IS L) THEN (WATER_PUMP IS L)
53. IF (RELATIVE_HUMIDITY IS VH) AND (SOIL_MOISURE IS M) THEN (WATER_PUMP IS M)
54. IF (RELATIVE_HUMIDITY IS VH) AND (SOIL_MOISURE IS H) THEN (WATER_PUMP IS VL)
55. IF (RELATIVE_HUMIDITY IS VH) AND (SOIL_MOISURE IS VH) THEN (WATER_PUMP IS VL)
56. IF (SOIL_PH IS S_ACID) THEN (PH_DOWN IS OFF)(PH_UP IS L)
57. IF (SOIL_PH IS W_ACID) THEN (PH_DOWN IS OFF)(PH_UP IS M)
58. IF (SOIL_PH IS W_ALKA) THEN (WATER_PUMP IS M)(PH_DOWN IS M)(PH_UP IS OFF)
59. IF (SOIL_PH IS S_ALKA) THEN (WATER_PUMP IS M)(PH_DOWN IS L)(PH_UP IS OFF)
60. IF (SOIL_PH IS NEUTRAL) THEN (WATER_PUMP IS M)(PH_DOWN IS OFF)(PH_UP IS OFF)

Appendix 3: MATLAB TOOLBOX FILE

Code can be found in the following link: https://github.com/moorthysiva/GreenHouseFLC

Comparing CEC ’05 Functions Using Optimization Techniques

Sivasharmini Ganeshamoorthy
Faculty of Engineering Environment and Computing
Coventry University
Coventry, England
ganeshamos@uni.coventry.ac.uk

I. INTRODUCTION

Number of optimization algorithms have been introduced in past years to resolve real world optimization disputes. Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Evolutionary Programming (EP) and Simulated Annealing (SA) are some of the well-known approaches for optimization algorithm. Each of these algorithms are designed with a unique characteristic to solve unique problem. Therefore, outstanding results can be obtained when suitable algorithm is adopted for the specific problem. Moreover, when the algorithm is applied for the unique problem that it is designed for it may perform well as designed but when the same algorithm is applied for a different problem it may not produce desired results. Therefore, it is important to evaluate the algorithms in a more precise manner by stipulating common termination criterion, problem sizes, and through performing scalability. This is focused on CEC’2005 with 25 benchmark functions (Suganthan *et al*., 2005).

In this study, Function 12 - Schwefel’s Problem 2.13 and Function 21 - Rotated Hybrid Composition Function are selected among these 25 benchmarks functions. To evaluate these functions optimization algorithms such as Genetic Algorithm (GA) and Particle Swarm Optimization Algorithm (PSO) are adopted. In this, to get the better performance, 15 iterations are applied to each function. This can avoid minimum values that occurs coincidently.

Following sections in this study discuss on the evaluation of these functions comparing optimizations techniques such as GA, PSO. The aim of this is to identify the best optimization technique for this problem.

II. SCHWEFEL’S PROBLEM 2.13

Schwefel’s Problem 2.13 is formulated in below equation and visualized in Figure 1.

$$F_{12}(\mathbf{x})=\sum_{i=1}^{D}(\mathbf{A}_i-\mathbf{B}_i(\mathbf{x}))^2+f\_bias_{12},\ \mathbf{x}=[x_1,x_2,\dots,x_D]$$

$$\mathbf{A}_i=\sum_{j=1}^{D}(a_{ij}\sin\alpha_j+b_{ij}\cos\alpha_j),\ \mathbf{B}_i(x)=\sum_{j=1}^{D}(a_{ij}\sin x_j+b_{ij}\cos x_j),\ \text{for } i=1,\dots,D$$

D: dimensions

$\mathbf{A}$, $\mathbf{B}$ are two $D*D$ matrix, a_{ij}, b_{ij} are integer random numbers in the range [-100,100],

$\boldsymbol{\alpha}=[\alpha_1,\alpha_2,\dots,\alpha_D], \alpha_j$ are random numbers in the range $[-\pi,\pi]$.

Properties:

- Multi-modal
- Shifted
- Non-separable
- Scalable
- $\mathbf{x}\in[-\pi,\pi]^D$, Global optimum $\mathbf{x}^*=\boldsymbol{\alpha}$, $F_{12}(\mathbf{x}^*)=f\_bias_{12}=-460$

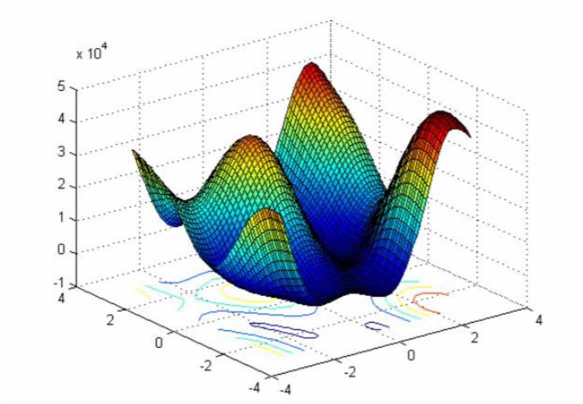


Figure 1: Schwefel’s Problem 2.13 3D map for 2-D function

III. ROTATED HYBRID COMPOSITION FUNCTION

Rotated Hybrid Composition Function is formulated in below equation and visualized in Figure 2.

$f_{1-2}(\mathbf{x})$: Rotated Expanded Scaffer’s F6 Function

$$F(x,y)=0.5+\frac{(\sin^2(\sqrt{x^2+y^2})-0.5)}{(1+0.001(x^2+y^2))^2}$$

$$f_i(\mathbf{x})=F(x_1,x_2)+F(x_2,x_3)+\dots+F(x_{D-1},x_D)+F(x_D,x_1)$$

$f_{3-4}(\mathbf{x})$: Rastrigin’s Function

$$f_i(\mathbf{x})=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i)+10)$$

$f_{5-6}(\mathbf{x})$: F8F2 Function

$$F8(\mathbf{x})=\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$$

$$F2(\mathbf{x})=\sum_{i=1}^{D-1}(100(x_i^2-x_{i+1})^2+(x_i-1)^2)$$

$$f_i(\mathbf{x})=F8(F2(x_1,x_2))+F8(F2(x_2,x_3))+\dots+F8(F2(x_{D-1},x_D))+F8(F2(x_D,x_1))$$

$f_{7-8}(\mathbf{x})$: Weierstrass Function

$$f_i(\mathbf{x})=\sum_{i=1}^{D}(\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k(x_i+0.5))])-D\sum_{k=0}^{k\max}[a^k\cos(2\pi b^k\cdot 0.5)],$$

a=0.5, b=3, k_{max}=20

$f_{9-10}(\mathbf{x})$: Griewank’s Function

$$f_i(\mathbf{x})=\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$$

$\boldsymbol{\sigma}=[1,1,1,1,1,2,2,2,2,2]$,

$\boldsymbol{\lambda}=[5*5/100; 5/100; 5*1; 1; 5*1; 1; 5*10; 10; 5*5/200; 5/200]$;

$\mathbf{M}_i$ are all orthogonal matrix

Properties:

- Multi-modal
- Rotated
- Non-Separable
- Scalable
- A huge number of local optima
- Different function’s properties are mixed together
- $\mathbf{x}\in[-5,5]^D$, Global optimum $\mathbf{x}^*=\mathbf{o}_1$, $F_{21}(\mathbf{x}^*)=f\_bias_{21}=360$

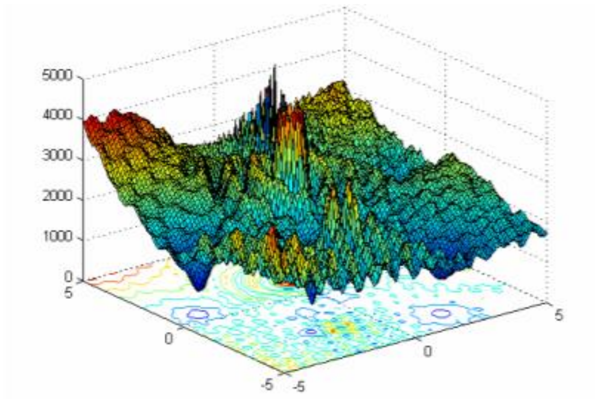


Figure 227: Schwefel's Problem 2.13 3D map for 2-D function

II. Optimization Techniques

A. Genetic Algorithm

As discussed before GA has been adopted to resolve optimization disputes. The basic concept of this has been influenced by natural selection and genetics. Initially, the solution of a problem in GA is encoded as a string of bits called chromosome and evaluated with fitness function. Among this best chromosomes are selected for the initial population of chromosomes. Best chromosomes are identified by the results of fitness function. Genetic operators which are crossover and mutation are adopted to create the offspring chromosomes. This offspring chromosomes will then be added to the new populations. This will be repeated until it meets the stopping criteria.

B. Particle Swarm Optimization Algorithm(PSO)

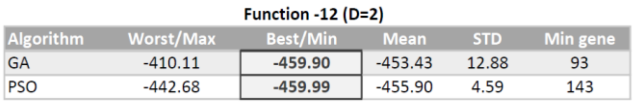
PSO is an alternative method to resolve optimization disputes. This is generated through number of agents (particles) and allocating velocities to each agent. The performance of each agent at a place and during movement is determined through using fitness function. In this, the movement of the agent depends on previous best results. The stopping criteria will be met when predefined number of iterations are reached and when the performance is static for number of iterations.

The achieved results clearly state that algorithms are unique for each problem hence it will not show best results in all situations. Therefore, the best results can be obtained through testing number of algorithms. Although, two algorithms were chosen the gained results were approximately closer to global minima. In conclusion, optimization can solve many problems however, it is hard to guarantee that an algorithm will solve a specific type of a problem.

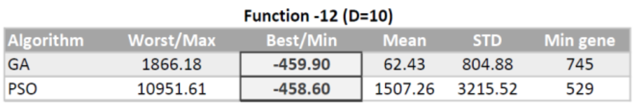
III. Results and Discussion

This study has evaluated the optimization of Schwefel's Problem 2.13 (Function 12) and Rotated Hybrid Composition Function (Function 21) through using Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). In each technique both the functions were ran 15 times and results were recorded. The two dimension (2D) and ten dimension (10D) of the functions are used to evaluate the optimization. The results contain the best performance, worse performance, mean, standard deviation and minimum number of generations. These algorithms have been run in MATLAB and those codes are included in Appendix 6 and 7.

2D and 10D evaluation results for Schwefel's Problem are shown in Table 1. According to the results all the scenarios have reached the global minima. Lowest standard deviation has reached by PSO with 2D. GA with 10D is the slowest scenario among them which took 746 iteration to achieve the global minimum. While comparing the scenarios, PSO with 10D has given the worst performance and bit far away from global minima. On the other hand, GA with 2D has given the best results with minimum iteration 93. The entire results which includes the performance of each run are shown in Appendices 1 and 3.


Function -12 (D=2)

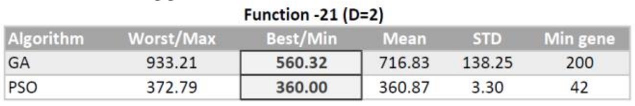
Algorithm	Worst/Max	Best/Min	Mean	STD	Min gene
GA	-410.11	-459.90	-453.43	12.88	93
PSO	-442.68	-459.99	-455.90	4.59	143


Function -12 (D=10)

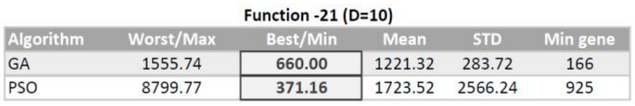
Algorithm	Worst/Max	Best/Min	Mean	STD	Min gene
GA	1866.18	-459.90	62.43	804.88	745
PSO	10951.61	-458.60	1507.26	3215.52	529

Table 1: Schwefel's Problem 2.13 Results Summary

2D and 10D evaluation results for Rotated Hybrid Composition Function are shown in Table 2. According to the results not all the scenarios reached the global minima. However, PSO algorithm has reached closely to global minima than GA algorithm. Lowest standard deviation is reached by PSO with 10D. PSO with 10D is the slowest scenario among them which took 925 iteration to achieve the global minima. While comparing the scenarios, GA with 2D has given worst performance and deviating from global minimum. On the other hand, PSO with 10D has given best results with minimum iteration 42.
The entire results which includes the performance of each run are shown in Appendices 2 and 4.


Function -21 (D=2)

Algorithm	Worst/Max	Best/Min	Mean	STD	Min gene
GA	933.21	560.32	716.83	138.25	200
PSO	372.79	360.00	360.87	3.30	42


Function -21 (D=10)

Algorithm	Worst/Max	Best/Min	Mean	STD	Min gene
GA	1555.74	660.00	1221.32	283.72	166
PSO	8799.77	371.16	1723.52	2566.24	925

Table 2: Rotated Hybrid Composition Function Results Summary

IV. Conclusion

The results achieved clearly state that algorithms are unique for each problem hence, it will not show the best results in all situations. However, the best results can be obtained through testing number of algorithms. In the two algorithms chosen for this study, the results gained to optimize the functions were approximately closer to global minima. In conclusion, optimization can solve many problems however, it is hard to guarantee that an algorithm will solve a specific type of a problem.

References

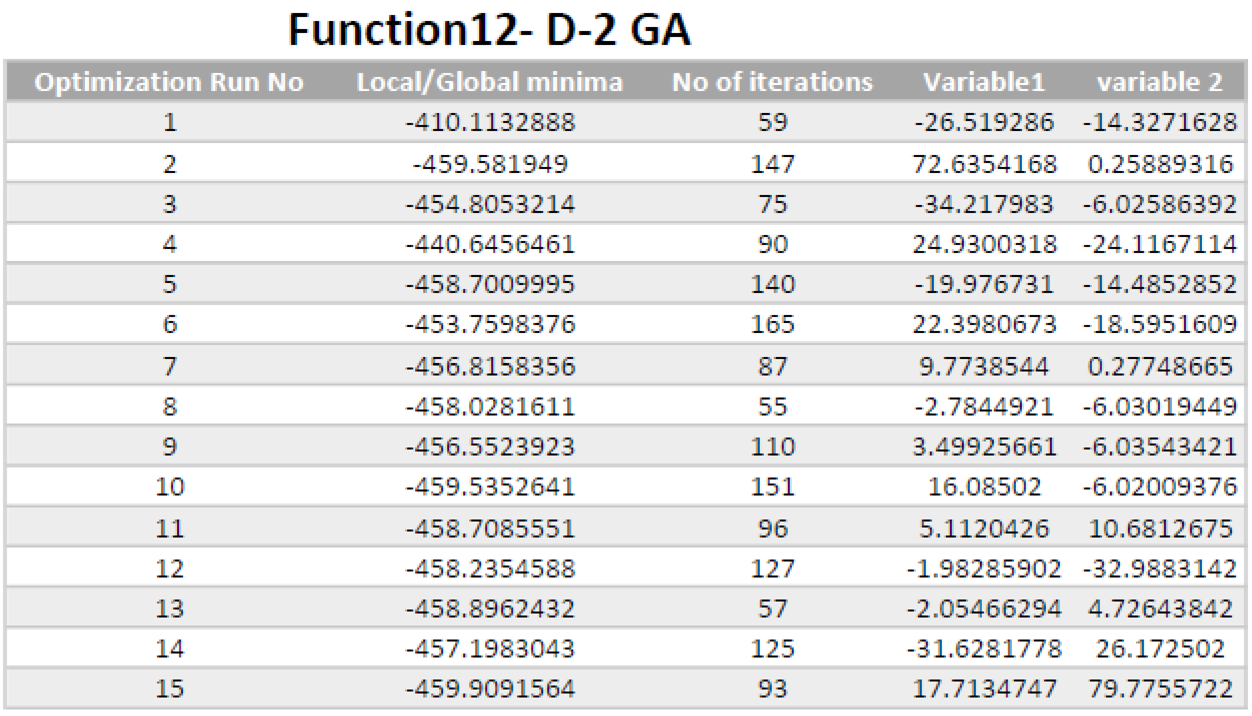
[1] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger and S. Tiwari, Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization KanGAL Report , #2005005,2005

[2] MathWorks. (2020a) *What Is Particle Swarm Optimization?* [online] Available from: <https://se.mathworks.com/help/ga ds/what-is-particle-swarm-optimization.html>].

[3] MathWorks. (2020b) *What Is the Genetic Algorithm?* [online] Available from: <https://se.mathworks.com/help/ga ds/what-is-the-genetic-algorithm.html> [25 March 2020].

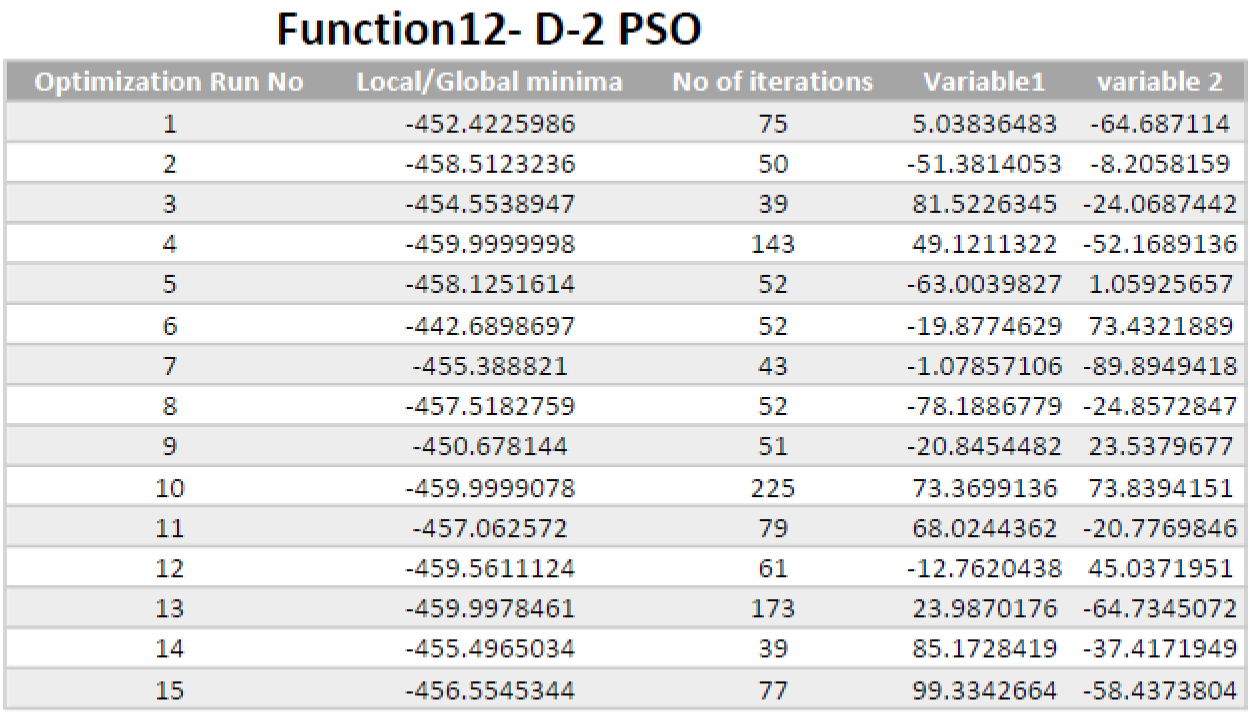
APPENDICES

Appendix 1: Two Dimensional Results for Schwefel's Problem



Function12- D-2 GA

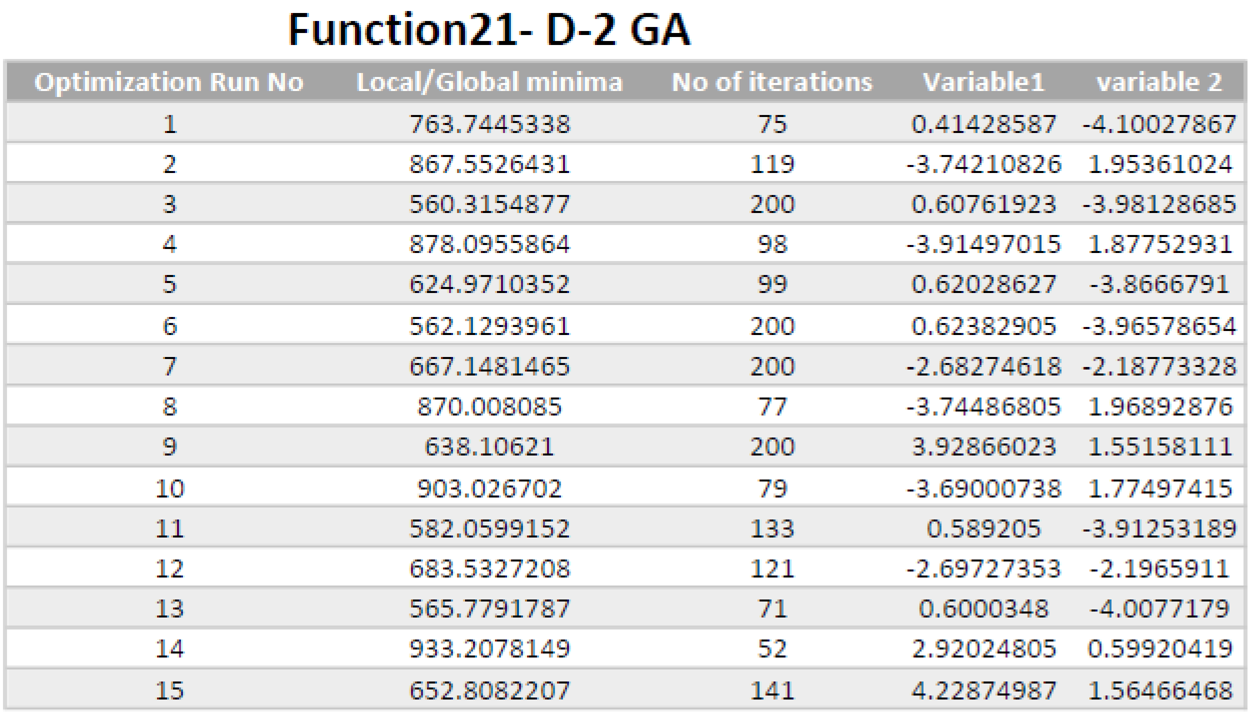
Optimization Run No	Local/Global minima	No of iterations	Variable1	variable 2
1	-410.1132888	59	-26.519286	-14.3271628
2	-459.581949	147	72.6354168	0.25889316
3	-454.8053214	75	-34.217983	-6.02586392
4	-440.6456461	90	24.9300318	-24.1167114
5	-458.7009995	140	-19.976731	-14.4852852
6	-453.7598376	165	22.3980673	-18.5951609
7	-456.8158356	87	9.7738544	0.27748665
8	-458.0281611	55	-2.7844921	-6.03019449
9	-456.5523923	110	3.49925661	-6.03543421
10	-459.5352641	151	16.08502	-6.02009376
11	-458.7085551	96	5.1120426	10.6812675
12	-458.2354588	127	-1.98285902	-32.9883142
13	-458.8962432	57	-2.05466294	4.72643842
14	-457.1983043	125	-31.6281778	26.172502
15	-459.9091564	93	17.7134747	79.7755722



Function12- D-2 PSO

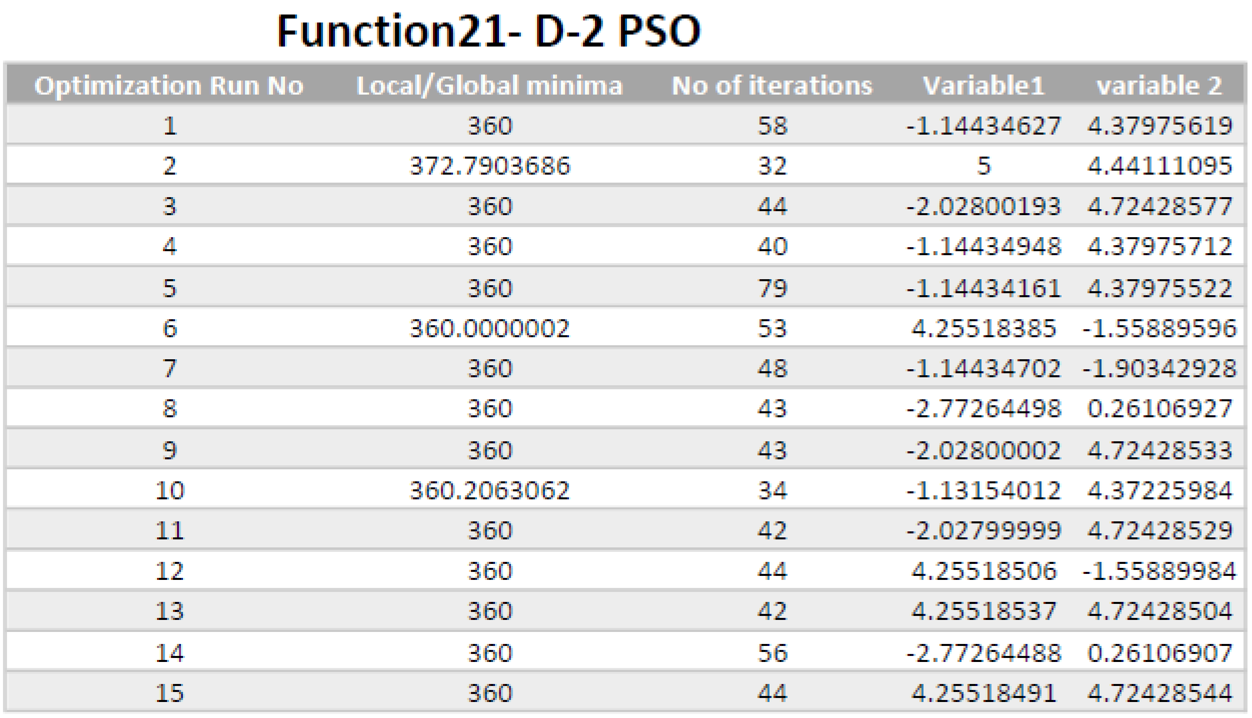
Optimization Run No	Local/Global minima	No of iterations	Variable1	variable 2
1	-452.4225986	75	5.03836483	-64.687114
2	-458.5123236	50	-51.3814053	-8.2058159
3	-454.5538947	39	81.5226345	-24.0687442
4	-459.9999998	143	49.1211322	-52.1689136
5	-458.1251614	52	-63.0039827	1.05925657
6	-442.6898697	52	-19.8774629	73.4321889
7	-455.388821	43	-1.07857106	-89.8949418
8	-457.5182759	52	-78.1886779	-24.8572847
9	-450.678144	51	-20.8454482	23.5379677
10	-459.9999078	225	73.3699136	73.8394151
11	-457.062572	79	68.0244362	-20.7769846
12	-459.5611124	61	-12.7620438	45.0371951
13	-459.9978461	173	23.9870176	-64.7345072
14	-455.4965034	39	85.1728419	-37.4171949
15	-456.5545344	77	99.3342664	-58.4373804

Appendix 2: Two Dimensional Results for Rotated Hybrid Composition Function



Function21- D-2 GA

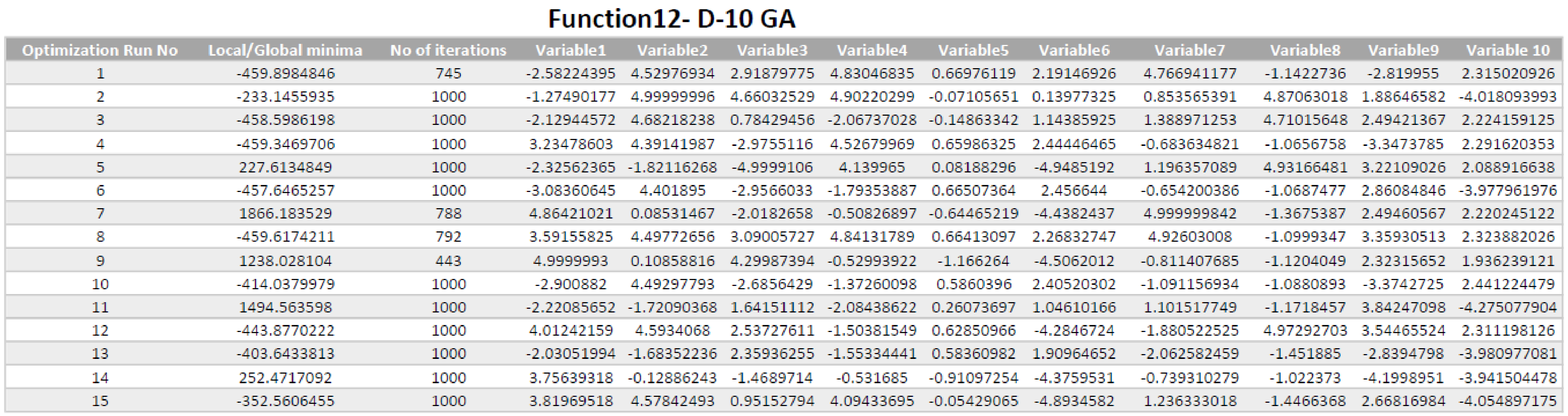
Optimization Run No	Local/Global minima	No of iterations	Variable1	variable 2
1	763.7445338	75	0.41428587	-4.10027867
2	867.5526431	119	-3.74210826	1.95361024
3	560.3154877	200	0.60761923	-3.98128685
4	878.0955864	98	-3.91497015	1.87752931
5	624.9710352	99	0.62028627	-3.8666791
6	562.1293961	200	0.62382905	-3.96578654
7	667.1481465	200	-2.68274618	-2.18773328
8	870.008085	77	-3.74486805	1.96892876
9	638.10621	200	3.92866023	1.55158111
10	903.026702	79	-3.69000738	1.77497415
11	582.0599152	133	0.589205	-3.91253189
12	683.5327208	121	-2.69727353	-2.1965911
13	565.7791787	71	0.6000348	-4.0077179
14	933.2078149	52	2.92024805	0.59920419
15	652.8082207	141	4.22874987	1.56466468



Function21- D-2 PSO

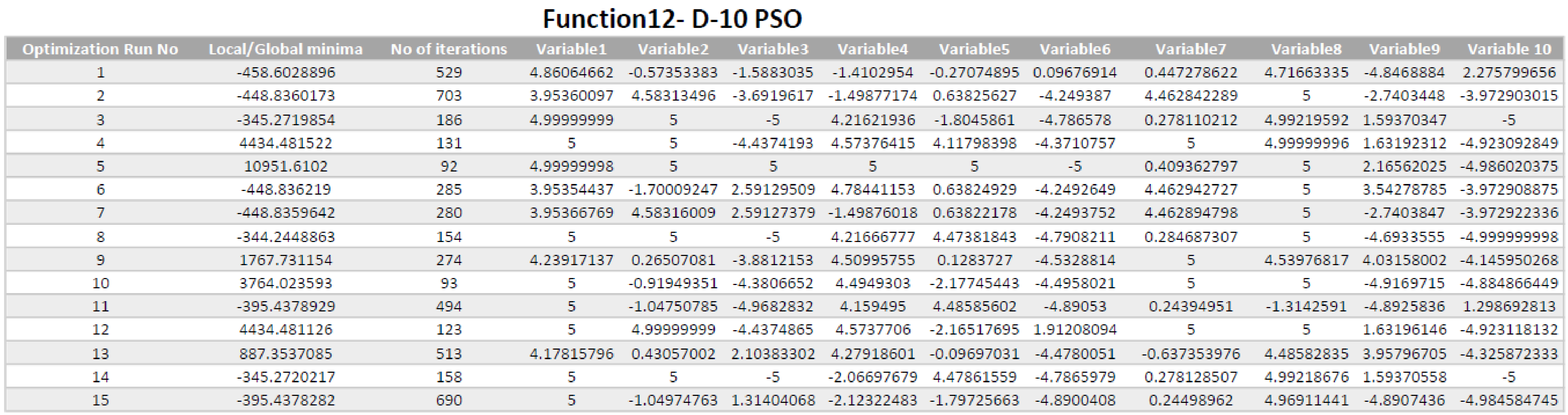
Optimization Run No	Local/Global minima	No of iterations	Variable1	variable 2
1	360	58	-1.14434627	4.37975619
2	372.7903686	32	5	4.44111095
3	360	44	-2.02800193	4.72428577
4	360	40	-1.14434948	4.37975712
5	360	79	-1.14434161	4.37975522
6	360.0000002	53	4.25518385	-1.55889596
7	360	48	-1.14434702	-1.90342928
8	360	43	-2.77264498	0.26106927
9	360	43	-2.02800002	4.72428533
10	360.2063062	34	-1.13154012	4.37225984
11	360	42	-2.02799999	4.72428529
12	360	44	4.25518506	-1.55889984
13	360	42	4.25518537	4.72428504
14	360	56	-2.77264488	0.26106907
15	360	44	4.25518491	4.72428544

Appendix 3: Ten Dimensional Results for Schwefel's Problem



Function12- D-10 GA

Optimization Run No	Local/Global minima	No of iterations	Variable1	Variable2	Variable3	Variable4	Variable5	Variable6	Variable7	Variable8	Variable9	Variable 10
1	-459.8984846	745	-2.58224395	4.52976934	2.91879775	4.83046835	0.66976119	2.19146926	4.766941177	-1.1422736	-2.819955	2.315020926
2	-233.1455935	1000	-1.27490177	4.99999996	4.66032529	4.90220299	-0.07105651	0.13977325	0.853565391	4.87063018	1.88646582	-4.018093993
3	-458.5986198	1000	-2.12944572	4.68218238	0.78429456	-2.06737028	-0.14863342	1.14385925	1.388971253	4.71015648	2.49421367	2.224159125
4	-459.3469706	1000	3.23478603	4.39141987	-2.9755116	4.52679969	0.65986325	2.44446465	-0.683634821	-1.0656758	-3.3473785	2.291620353
5	227.6134849	1000	-2.32562365	-1.82116268	-4.9999106	4.139965	0.08188296	-4.9485192	1.196357089	4.93166481	3.22109026	2.088916638
6	-457.6465257	1000	-3.08360645	4.401895	-2.9566033	-1.79353887	0.66507364	2.456644	-0.654200386	-1.0687477	2.86084846	-3.977961976
7	1866.183529	788	4.86421021	0.08531467	-2.0182658	-0.50826897	-0.64465219	-4.4382437	4.999999842	-1.3675387	2.49460567	2.220245122
8	-459.6174211	792	3.59155825	4.49772656	3.09005727	4.84131789	0.66413097	2.26832747	4.92603008	-1.0999347	3.35930513	2.323882026
9	1238.028104	443	4.9999993	0.10858816	4.29987394	-0.52993922	-1.166264	-4.5062012	-0.811407685	-1.1204049	2.32315652	1.936239121
10	-414.0379979	1000	-2.900882	4.49297793	-2.6856429	-1.37260098	0.5860396	2.40520302	-1.091156934	-1.0880893	-3.3742725	2.441224479
11	1494.563598	1000	-2.22085652	-1.72090368	1.64151112	-2.08438622	0.26073697	1.04610166	1.101517749	-1.1718457	3.84247098	-4.275077904
12	-443.8770222	1000	4.01242159	4.5934068	2.53727611	-1.50381549	0.62850966	-4.2846724	-1.880522525	4.97292703	3.54465524	2.311198126
13	-403.6433813	1000	-2.03051994	-1.68352236	2.35936255	-1.55334441	0.58360982	1.90964652	-2.062582459	-1.451885	-2.8394798	-3.980977081
14	252.4717092	1000	3.75639318	-0.12886243	-1.4689714	-0.531685	-0.91097254	-4.3759531	-0.739310279	-1.022373	-4.1998951	-3.941504478
15	-352.5606455	1000	3.81969518	4.57842493	0.95152794	4.09433695	-0.05429065	-4.8934582	1.236333018	-1.4466368	2.66816984	-4.054897175



Function12- D-10 PSO

Optimization Run No	Local/Global minima	No of iterations	Variable1	Variable2	Variable3	Variable4	Variable5	Variable6	Variable7	Variable8	Variable9	Variable 10
1	-458.6028896	529	4.86064662	-0.57353383	-1.5883035	-1.4102954	-0.27074895	0.09676914	0.447278622	4.71663335	-4.8468884	2.275799656
2	-448.8360173	703	3.95360097	4.58313496	-3.6919617	-1.49877174	0.63825627	-4.249387	4.462842289	5	-2.7403448	-3.972903015
3	-345.2719854	186	4.99999999	5	-5	4.21621936	-1.8045861	-4.786578	0.278110212	4.99219592	1.59370347	-5
4	4434.481522	131	5	5	-4.4374193	4.57376415	4.11798398	-4.3710757	5	4.99999996	1.63192312	-4.923092849
5	10951.6102	92	4.99999998	5	5	5	5	-5	0.409362797	5	2.16562025	-4.986020375
6	-448.836219	285	3.95354437	-1.70009247	2.59129509	4.78441153	0.63824929	-4.2492649	4.462942727	5	3.54278785	-3.972908875
7	-448.8359642	280	3.95366769	4.58316009	2.59127379	-1.49876018	0.63822178	-4.2493752	4.462894798	5	-2.7403847	-3.972922336
8	-344.2448863	154	5	5	-5	4.21666777	4.47381843	-4.7908211	0.284687307	5	-4.6933555	-4.999999998
9	1767.731154	274	4.23917137	0.26507081	-3.8812153	4.50995755	0.1283727	-4.5328814	5	4.53976817	4.03158002	-4.145950268
10	3764.023593	93	5	-0.91949351	-4.3806652	4.4949303	-2.17745443	-4.4958021	5	5	-4.9169715	-4.884866449
11	-395.4378929	494	5	-1.04750785	-4.9682832	4.159495	4.48585602	-4.89053	0.24394951	-1.3142591	-4.8925836	1.298692813
12	4434.481126	123	5	4.99999999	-4.4374865	4.5737706	-2.16517695	1.91208094	5	5	1.63196146	-4.923118132
13	887.3537085	513	4.17815796	0.43057002	2.10383302	4.27918601	-0.09697031	-4.4780051	-0.637353976	4.48582835	3.95796705	-4.325872333
14	-345.2720217	158	5	5	-5	-2.06697679	4.47861559	-4.7865979	0.278128507	4.99218676	1.59370558	-5
15	-395.4378282	690	5	-1.04974763	1.31404068	-2.12322483	-1.79725663	-4.8900408	0.24498962	4.96911441	-4.8907436	-4.984584745

Appendix 4: Ten Dimensional Results for Rotated Hybrid Composition Function

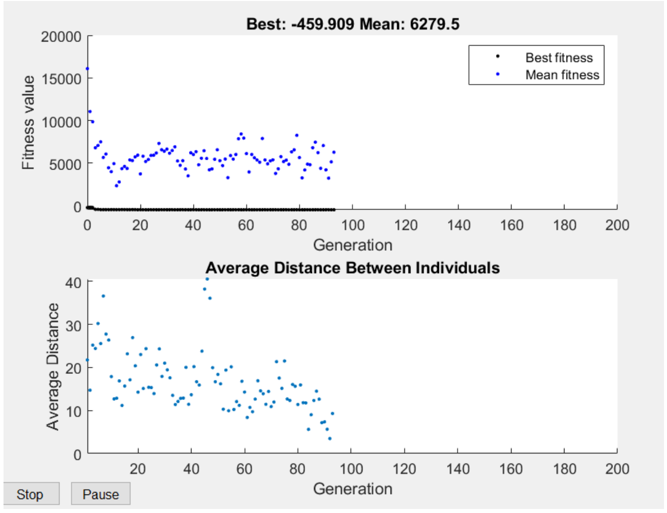
Function21- D-10 GA

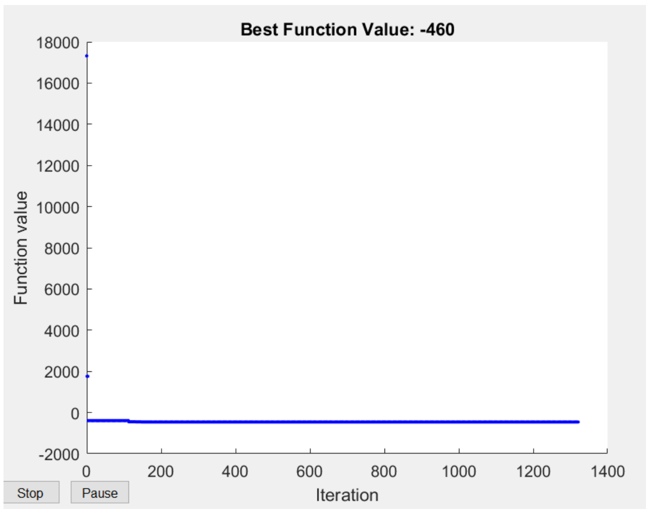
Optimization Run No	Local/Global minima	No of iterations	Variable1	Variable2	Variable3	Variable4	Variable5	Variable6	Variable7	Variable8	Variable9	Variable 10
1	1260.002236	183	-1.27228291	-1.29052751	2.58735827	2.24246115	-3.42427476	1.55451433	0.630167256	-3.9855752	3.99827515	-2.190758293
2	1291.554109	155	-2.81947967	-1.75639803	-4.2605675	3.06071564	-2.13837155	2.69930873	-1.660807101	-2.6305383	-0.1017853	3.346392112
3	1326.858253	174	-3.24188621	-2.19077521	-3.5103146	2.26394895	-2.13512881	1.80530922	-1.636467409	-1.6598961	0.51464461	3.215960266
4	1555.738999	178	-4.61014407	-1.04976706	-1.6418749	3.21902843	-2.91368082	-0.3143959	0.563033191	-4.6088127	4.1350247	2.986399021
5	1525.814013	141	-3.51420301	-1.22211356	-4.5568225	2.69088314	0.39994014	1.78960737	-1.598963201	-2.0536503	-1.9218412	3.978078808
6	1482.327776	175	-4.16391592	-1.48960764	-3.0929373	2.53237719	-0.53272814	1.41387912	-0.255456322	-2.6767948	0.15981484	2.673239127
7	1434.887966	171	-3.86618867	-0.91684819	-3.3962216	2.10935224	1.7087074	1.16730008	0.001426631	-0.4452919	-0.7677175	3.554075531
8	1223.950061	158	-2.53369886	-1.96770043	-3.7428268	1.83748297	2.32866894	1.9421009	-0.996507202	-0.3687012	-1.2942147	3.460866212
9	660.0023601	166	-2.66394384	-2.17970919	-3.1791748	2.02006413	2.43222496	1.50177534	-0.7390978	-0.2911232	-1.4310961	2.744655213
10	860.0001807	123	-3.79209935	1.90728699	1.75006085	2.93087023	-1.26127972	-0.9486988	3.345822478	-3.3168099	2.95675065	3.34013728
11	1510.411028	145	-3.0952792	-2.15464596	-3.3652851	1.91184516	0.30884907	1.36244126	-0.127699234	-0.4256227	-1.288102	3.592378146
12	860.0004915	154	-3.79212008	1.90746299	1.74982062	2.93042465	-1.26194895	-0.9486492	3.34592974	-3.3179326	2.95688677	3.340346866
13	1249.838366	149	-3.45037432	-1.08771284	-4.6297378	2.36148688	-3.1178264	2.10113912	-1.601743135	-2.2553998	-0.5404466	2.708881246
14	1218.486256	180	-3.4630317	-1.58663063	-3.992789	2.47207123	-2.52352732	2.02990066	-1.725293828	-2.0858664	0.06871699	3.253650576
15	860.0003784	142	-3.79227945	1.90699164	1.75014267	2.93098583	-1.26086688	-0.9489288	3.346071041	-3.3176168	2.95719475	3.34068744

Function21- D-10 PSO

Optimization Run No	Local/Global minima	No of iterations	Variable1	Variable2	Variable3	Variable4	Variable5	Variable6	Variable7	Variable8	Variable9	Variable 10
1	474.7280012	171	5	5	-5	4.21620201	4.47865085	-4.7865668	0.278096366	4.99216143	-4.6894442	-5
2	474.7279776	125	5	5	-5	4.21621431	4.4786143	-4.7865887	0.278122388	4.9921819	-4.6894708	-4.999999998
3	371.1640221	925	3.95369508	4.58314559	-3.6919595	4.78446496	0.63823136	-4.2493937	4.462808778	4.99999996	-2.7403437	-3.972880292
4	474.7279763	135	5	5	-5	4.21621311	4.47861211	-4.7865959	0.278130775	4.99219292	-4.6894834	-5
5	471.2726366	341	5	5	-5	4.21751114	-1.80424615	-4.7665818	0.250668386	4.97576086	1.59178003	1.267589038
6	474.7280501	155	5	5	-5	4.21621604	4.47857744	-4.7866106	0.278156428	4.99223548	1.59365026	-4.999999959
7	1080.262797	179	4.04973296	4.4676038	-5	4.15606342	0.09700467	-5	-5	4.91208833	3.22027562	-4.189102482
8	417.0680895	170	-1.19699759	5	-4.9631799	4.18706722	-1.80858281	-4.8128974	0.303179847	4.99999997	-4.6746497	-5
9	5254.481064	130	5	5	-4.4374927	4.57379681	4.11799265	-4.3710915	4.999999998	5	-4.6512063	-4.923112236
10	474.7279772	163	5	5	-5	4.21621084	4.47861101	1.49658947	0.278133431	4.99219414	1.59371178	-5
11	424.5630408	545	5	-1.04669474	1.31526366	4.15937918	4.48572354	-4.8906223	0.243542436	4.96898025	-4.8932436	-4.984365909
12	931.3903141	129	4.27722582	4.58956205	2.70144134	5	0.64294277	1.91292667	-1.92669204	5	-2.6928614	-4.012230069
13	5254.481084	98	5	4.99999999	-4.4375259	-1.70940039	4.11799449	1.91207562	5	5	1.63194513	-4.923134238
14	474.7279906	149	5	5	-5	-2.06697224	4.47862932	-4.7865728	0.278099518	4.9921615	-4.6894525	-5
15	8799.769193	132	4.73580947	5	-5	-1.6275726	4.45833432	-4.3864911	5	-1.5772361	-4.5310148	-4.951715444

Appendix 5: Visualizations Two Dimensional Schwefel's Problem





Genetic Algorithm **Particle Swarm Optimization**

Appendix 6: Code for Genetic Algorithm Evaluation

```
clc, clear, close all
rng default
global initial_flag

% Function -  and Dimention =2

%% GA with 15 iterations
initial_flag = 0;
options = optimoptions('ga','PlotFcn',{@gaplotbestf,@gaplotdistance});
for i=1:15
initial_flag = 0;

[ga_func_x,ga_func_val,ga_func_exit_flag,ga_func_output] =
ga(@(x)benchmark_func(x,12),2,options)
%[ga_func_x,ga_func_val,ga_func_exit_flag,ga_func_output] =
ga(@(x)benchmark_func(x,12),10,options)
%[ga_func_x,ga_func_val,ga_func_exit_flag,ga_func_output] =
ga(@(x)benchmark_func(x,21),2,options)
%[ga_func_x,ga_func_val,ga_func_exit_flag,ga_func_output] =
ga(@(x)benchmark_func(x,21),10,options)
ga_func_x_matrix(i,:) = ga_func_x
ga_func_val_matrix(i) = ga_func_val
ga_func_exit_flag_matrix (i) = ga_func_exit_flag
ga_func_output_matrix (i) = ga_func_output
ga_func_gene_matrix(i,:) = ga_func_output.generations
% save visualizations to file
fig = sprintf('GA_Figure_Function12_D2(%d).fig', i) ;
savefig(fig)
end

%% GA 15 iteration measures
ga_func_val_max = max(ga_func_val_matrix)
ga_func_val_min = min(ga_func_val_matrix)
ga_func_val_mean = mean(ga_func_val_matrix)
ga_func_val_std = std(ga_func_val_matrix);
% find no of which generation iteration we find the minimum
min_gen = find(ga_func_val_matrix == min(ga_func_val_matrix(:)));
no_of_gene_min = ga_func_gene_matrix(min_gen)
```

Appendix 7: Code for Particle Swarm Optimization Algorithm Evaluation

```
%% PSO with 15 Iterations and Dimention =2
options = optimoptions('particleswarm','PlotFcn',{@pswplotbestf});
initial_flag = 0;
upper_limit = [5,100];
lower_limit = [-5,-100];
for i=1:15
[swarm_func_x,swarm_func_val,swarm_func_exit_flag,swarm_func_output] =
particleswarm(@(x)benchmark_func(x,12),2,lower_limit,upper_limit,options)
%[swarm_func_x,swarm_func_val,swarm_func_exit_flag,swarm_func_output] =
particleswarm(@(x)benchmark_func(x,12),10,lower_limit,upper_limit,options)
%[swarm_func_x,swarm_func_val,swarm_func_exit_flag,swarm_func_output] =
particleswarm(@(x)benchmark_func(x,21),2,lower_limit,upper_limit,options)
%[swarm_func_x,swarm_func_val,swarm_func_exit_flag,swarm_func_output] =
particleswarm(@(x)benchmark_func(x,21),10,lower_limit,upper_limit,options)
swarm_func_x_matrix(i,:) = swarm_func_x
swarm_func_val_matrix(i) = swarm_func_val
swarm_func_exit_flag_matrix(i) = swarm_func_exit_flag
swarm_func_output_matrix(i) = swarm_func_output
swarm_func_gene_matrix(i,:) = swarm_func_output.iterations
% save visualizations to file
fig = sprintf('PSO_Figure_Function12_D2.fig', i) ;
savefig(fig)
end
%% Particle Swarm calculations
swarm_func_val_max = max(swarm_func_val_matrix)
swarm_func_val_min = min(swarm_func_val_matrix)
swarm_func_val_mean = mean(swarm_func_val_matrix)
swarm_func_val_std = std(swarm_func_val_matrix)
% find no of which generation iteration we find the minimum
min_gen = find(swarm_func_val_matrix == min(swarm_func_val_matrix(:)));
no_of_gene_min = swarm_func_gene_matrix(min_gen)
```