ACTA SCIENTIFIC AGRICULTURE (ISSN: 2581-365X)

Volume 3 Issue 10 October 2019

Review Article

Genetic Algorithm to find Most Optimum Growing Technique for Multiple-Cropping using Big-data

Vinamra Das* and Sunny Jain

VIT Vellore, Vellore, India

***Corresponding Author:** Vinamra Das, VIT Vellore, Vellore, India

Received: July 15, 2019; **Published:** September 20, 2019

DOI: 10.31080/ASAG.2019.03.0658

Abstract

In the present scenario, it is extremely important for any farmer to increase his farm productivity and using multi farming techniques is one of the most suitable way to achieve that [1]. Many new farming techniques are being introduced to which a general farmer has no access to and hence his growth rate is monotonous. Even after having the cutting-edge technology and farming techniques a general farmer has no access to any of it, that is the outreach of information technology in farming is still very low. As the number of parameters to optimize farm productivity increases, so does the permutations of number of techniques and hence expertise is needed to analyse the best farming technique for the given scenario. Given so many existing techniques which vary over even a slight change in parameters, only the experts in farming are adaptable to them and hence it is extremely important to automate the technique generation process so as to put the capability of generating the best farming output to a non -farming expert. From the farmer's point of view, smart farming should provide him with the best crop output in the most sustainable manner. Moreover multiple-cropping over single piece of land has become a necessity to meet the financial requirements and the demand -supply chain in the market. Due to multiple crops being planted on the same piece of land, it makes the technique generation process more ambiguous and time-consuming. To tackle this problem a big data analysis of farming parameters might prove to be helpful [2]. Big data analysis helps in exploiting large data sets computationally to observe hidden patterns, trends and outcome of each technique. The combination of using smart farming with the big-data for multiple-cropping can provide the most well-analysed results and the complex patterns which are not perceivable to humans in providing the most optimum use of farming resources under the given constraints. Using an analysis algorithm over the big data set might be able to provide faster and precise techniques over the complex set of quantifiable parameters and widely changing constraints.

Keywords: Smart-Farming; Multiple-Cropping; Big-Data

Introduction

In order to increase the farming output of the land under various constraints and parameters, aim is to utilise the big-data set provided. Today there are multiple number of agricultural challenges a farmer has to face including inadequate technology, improper irrigation systems, climatic factors [3]. Various parameters affect the crop productivity like soil-type, types of fertilisers used, amount and frequency of water given, dependability of multiple crops over each other when planted on a single land, ambient temperature, amount of moisture, number and type of pesticides which affects all the crops in the most optimum way, duration of sunlight exposure. Another subjective parameter is the effect of past technique used in the present scenario and the effect of present technique for the latter generation must be analysed and a way to quantify is devised. In addition to this it must be kept in mind that a well-balanced farming structure has to be maintained

which means the inter-dependence of various parameters must be well dealt with. Since, almost all the parameters are inter-related and all of them might affect the growth output in different weights, hence an algorithm is needed which maintains an optimum balance according to the importance of each parameter. For Example, the change in amount of water given might result in bigger changes in the crop output than the change in sunlight, hence in this case more priority should be given towards optimization of water content in comparison to amount of sunlight.

To implement this structure Genetic Algorithm might prove to be helpful as it is an evolutionary algorithm which foresees the effectivity of the proposed technique by extracting data from big-data set. Since the data-set is going to be very large hence, the time-complexity will also be huge to calculate the best offspring. Use of genetic algorithm here ensures minimizing time complexity, since it works on probabilistic approach to find better offspring faster than usual iterative method as herein the selection of gene with better fitness value is higher [17].

There is an estimate that more than 55% of Indians depend on farming alone [4], out of which majority are residing in rural areas with negligible or no knowledge of scientific approach of farming. For better farming techniques to be provided to every farmer with proper analysis and expertise a huge amount of manpower is needed. For example, various farmer helplines are introduced by the government but the number of experts is exponentially less than the number of farmers, resulting in delayed response. In addition to that the experts advise is going to be less informed and less precise as it is humanly impossible to retrieve and analyse big-data.

Hence the use of a big data set with the list of action of quantifiable parameters on the crop cultivation and their outputs in a tabulated way can help in analysing the crop outputs in every possible permutation quite effectively. A method of representing the whole farming technique is applied in a binary string fashion, upon which the genetic algorithm acts based on the constraints specified by the farmer in order to produce the most optimum farming technique for the given multiple crops. As a result, the most optimum technique for cultivation is generated which is completely personalized and caters to the individual constraints of each farmer in form of a binary string of the required parameters, which is delivered to the farmer in normalized language [18].

Smart farming has become the present need so as to produce the maximum crop outputs, and genetic algorithm have been used in order to predict weather [5], which ensure that the crops are safe from any climatic harsh and also determine the most favourable time to grow them by predicting the weather conditions. Big data has a huge effect on farming techniques. By providing information that can lead to analysis of data revealing such trends which are not possible manually. IoT has made information gathering much more effective and less time consuming and drones are being used to collect agricultural data efficiently [6]. Another important factor is the land allocation technique while dealing with multiple crops. The division should be done in the most optimal way such that the output of the land as a whole is highest [7]. Usage of different pesticides and their cross-cutting effects on each other also has to be optimized while dealing with crops. Further these problems become more complex as the number of crops increases, as each crop has its own pesticide demands and hence might lead to heavy cross –cutting needs. The multiple cropping problem is important as it has led to better yields and more cost-effective methods in the past, but a thorough study on the past patterns was needed and hence high expertise is needed. The above methods deal with every single aspect of farming individually and have given reasonable results but the final crop output is a cumulative of a number of factors such as soil fertility, exposure to sunlight, amount of water given, types of pesticides used, soil type, fertilizers used so on and so forth. All of these play their weighted roles in the final crop output. A method for tackling this problem of multiple parameters and quantifying them is introducing the use of genetic algorithms [8]. In the literature mentioned here, all the parameters are taken as a single equation with each parameter having its own coefficients resembling the weight of each in the final output. Then the output function is maximized using genetic algorithm. This method works well for problems with only one crop type, as the data for the function has to be fetched from a database hence an assumption is made that ample amount of data is present for each of the crops. Thinking further, if the number of crops to be cultivated increases, it is not possible to have the output data of each combination (say there are 30 crops, then the number of crop combinations itself rises to 2^30-1) which is not a feasible number to have a database for. Note these are just the number of combinations the crops can be chosen, the database further requires data points for each parameter to gain the knowledge as to how each parameter affects the growth which will in turn increase the size of database

further. Furthermore, if any of the parameters is constrained in a way (say only a single soil type is present or the amount of water is restricted), then the complexity increases even further. To find the most optimum growing technique. Hence the solution we propose is to look at the whole growing technique as a bit string and using genetic algorithm to find the technique with the highest fitness value. Advantages are that the output data for all the crops data is no longer needed as "mirroring approach" is used to determine the approximate value for fitness of offspring taking in account the fitness value of the parent generations.

The objective of this article is to make new farming techniques with reduced computational time by using the concept of genetic algorithm. Genetic algorithm is used due to their generative property and no use of computational neural systems.

Materials and Methodology

In order to implement the proposed algorithm, a big data set consisting of the list of crops, soils used in past, fertilizers used, pesticides sprayed, amount and frequency of water sprayed, time for which sunlight [9] is received throughout the day, amount of moisture, temperature, humidity, previous crop output for that particular crop is maintained. These data values are termed as input values and change over time as more techniques are introduced by the farmer, hence the database evolves over time, becoming vaster and covering a greater number of possibilities. Another set of values is maintained for each technique for that particular crop known as output values, these values determine the validity of the technique and used in fitness function calculation (a quantum for how good the technique is). The "output values" consists of the number of crops produced, number of crops planted, number of crop units sold in market, time of yield and cost required for the whole technique.

Further, the following computations are done in the given order

- Take the number of crops to be planted as N from the user.
- Check for any constraints entered by the user, For Example limited water supply, limited sunlight or only a fixed type of soil present.
- Represent the whole farming technique by using a binary string and set the initial standardised set of technique as parent generation.
- The whole binary string is divided into various subparts where each sub-part representing a particular parameter and mutation [10] are only allowed between specific bit of string.
- A fitness function is generated for each offspring (represented by a binary string) and the fittest of them is selected and mutated for second generation.
- Note that the part of the binary string which represents the parameter constrained by the user remains the same in every offspring and hence, its status is termed as locked.
- The process of generating new offspring and calculating fitness function for mutation for production of next generation is repeated until the required fitness value is obtained.
- The binary hence generated represents the most suited farming techniques and is converted into normal language for the user.

Input

For input take the number and name of crops the farmer intends to plant. The main aim here is multi-cropping so the farmer can grow more than one crop in the land. The division of the land will solely be the choice of the user (Farmer) as the process output will have no concerns with the land-distribution decided before the initiation of the process. The constraints also have to be given as input. For Example, there might be some rigid resources which cannot be changed by him such as only one type of soil might be present. In such a scenario that parameter remains constant throughout the process.

Representation of input

Once the valid input is received from the user, retrieval of data takes place from the big-data set for the corresponding crop. Herein the whole farming technique is represented as a binary string where "1" represents presence of any factor and "0" represents absence of that factors. The whole string is divided into various sub-strings, each substring quantifying that parameter in terms of 0's and 1's in their own way.

For Example:

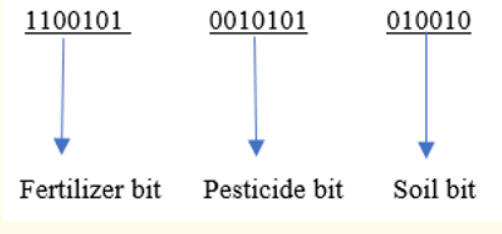


Figure 1: Sample representation of farming technique as a bit string.

Representation of each factor

Soil type

The list of soils is taken in a particular order for each crop and 0 represents the absence of that soil for the cropping technique and 1 represents presence of that soil for that technique. It can have multiple 1's in its string representing that more than one type of soil can be used.

Fertilizers user

In this case we list the standardized fertilizers used for production of the listed crops and here also 1 represents presence of that fertilizer and 0 the absence. Same goes with the types and number of pesticides used [11]. Here also multiple 1's can be used as different fertilizers can be used for different crops.

Temperature and water used

These kinds of factors deal with ranges and the position of each bit also holds significance apart from its value (0 or 1), here also 1 represents presence of that temperature range and 0 the absence. We quantify the temperature required for the given crops in temperature range as follows:

Position Bit	Temperature Bit
1st	5-10
2nd	10-15
3rd	15-20
4th	20-25
5th	25-30
6th	30-35

Table 1: Table representing bit position and the temperature range it shows.

Note

- In the temperature and the water string and the Seasonal Growth bit only one bit can be 1 as the presence of one condition surely means the absence of others.
- Each factor represents the sub-string involved for the given input. The Boolean value in these substrings may represent the range of the factor or just their presence.

Genetic Algorithm general overview

General Overview of algorithm: Genetic algorithm is a brute force hit and trial algorithm but with a probability roulette spinning method which increases the chances of acceptance of the cases which might lead towards the result. This reduces time complexity to a much larger extent [12].

Parent Generation selection: Initially the parent generation are selected as the binary bit string which represent the already existing or previously tried farming technique of each crops which the farmer intends to plant. If he wants to plant a total of 4 crops on his land (N=4) and let our big data set contains a total of 5 already tried techniques for each of the crops, then a total of 20 parent generations (represented in form of binary strings) are taken. Note that each single bit is termed as chromosome in Genetic Algorithm terminology [13].

Fitness function calculation: Fitness function is basically the quantum of how fit our offspring or already existing parent generation is. In this case fitness function represents how good or effective the farming technique has been for that crop or the combination of crops. We use a probability-based approach for selection of parents for mutation. Implying that the probability of selection for mutation is directly proportional to their fitness values.

Next generation production: The production of next generation of binary strings is generally done by dividing the string into two parts and follow a criss-cross exchange operation between the chromosomes (bits). For example: 110011 and 011110 when mutated in the criss-cross manner will produce the offspring as follows:

Mutation 1: 110 011

The first three bits are taken from the first sub-string and the last three bits are taken from the second substrings.

Mutation 2: 011 110

The first three bits are taken from the second sub-string and the last three bits are taken from the first substrings.

Evolution: On mutation, the off-springs having most high fitness function have the maximum probability of getting selected, the future generation produced generally have the higher fitness value, and if not, they are rejected by the algorithm as their probability of getting selected gets lower. This means that as we proceed

further (as the algorithm increases number of iterations) so does increases the fitness values of the offsprings, hence selection of the fitter generations takes place.

Repetition: The process of mutation, fitness function calculation and next generation production takes place until the desired fitness value of any one of the off-springs is obtained, and that gene in the generation is our desired output. Then we find the curve of best fit to find the additional number of iterations in order to find loop termination limit.

Algorithm

Initialize n fitness factors as F[1], F[2]……F[n]
/*fitness values of n genes*/

```
for (i=1; i<n;i++)
{
sum=sum+f[i]
}
for (j=1; j<n;j++)
{
P[j]=f[j]/sum; /*P[j] stores the probability of selection of F[j] gene*/
}
Initialize range [k]
/*defining the range of random() number generator for which F[k] will be
selected */
for (i=2;i<n;i++)
{
range [i] =P[i-1]to P[i]
}
for(k=1; k<number; k++)
/*k stores the number of genes to be selected for mutation */
{
r=random()
if (r=range[k])
select k
}
```

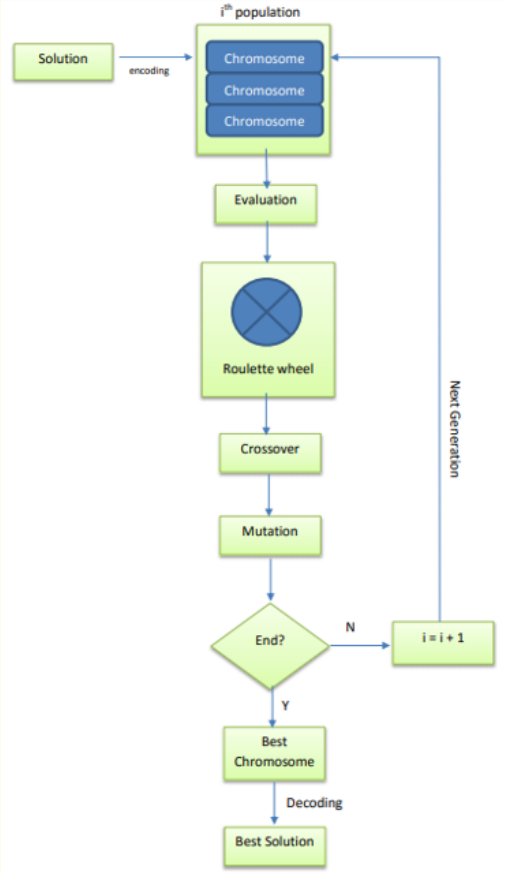


Figure 2: Flowchart of Genetic Algorithm.

Calculation of fitness function

Calculation of fitness function in genetic algorithm varies from case to case. In general, it represents how close the fitness value of produced generation is close to the desired fitness values. In certain cases, the final value of the fitness function is absolute and known to the user in advance (generally its value being 1), this case arises when we are dealing with more mathematically quantifiable problems like finding solution to a multi variable equation, determining the coefficients of curve of best fit and so on. But in many other cases majorly where a research output has to be generated, the programmer lacks the knowledge of final of apex fitness value. Same goes with this case as a definite value of fitness function is not known prior to the system; hence it lacks

the knowledge as to when to stop the mutation-fitness function generation loop. In order to limit the number of iterations the following steps are taken:

Consider three parent generation already existing, the initial fitness value the three parent generations be:

F_1=0.60 (first gene)
F_2=0.75 (second gene)
F_3=0.45 (third gene)

The highest among the three is taken as the Highest Fitness Value (HFV).

Here HFV=0.75

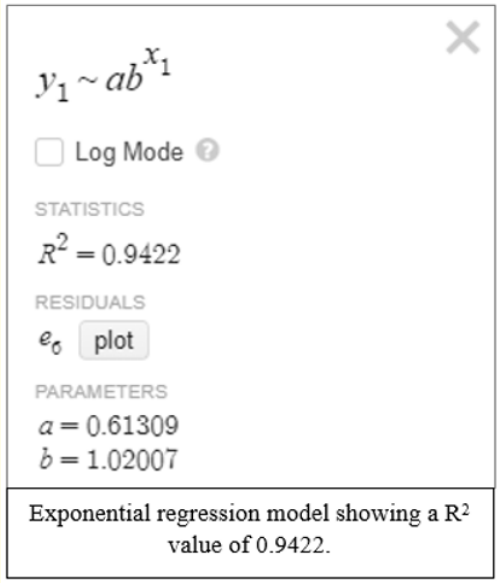
Here the loop is continued until the point that the maximum value of fitness function of each generation (max) exceeds the HFV (that is 0.75 in this case). For predicting the additional numbers of iterations from that point on, a number of increasing curve fitting models are applied for the number of generation (x-value) versus maximum fitness values for that generation (y-value).

Note: Only ever-increasing curve fitting models are applied as the maximum fitness value is expected to increase and hence over-fitting of curve is avoided. Quadratic or any higher order polynomial might give a better fit, but there is a chance that while equating it to 1 (to find the number of iterations), the roots might be negative, since quadratic functions are not always increasing.

Regression models [14] applied are

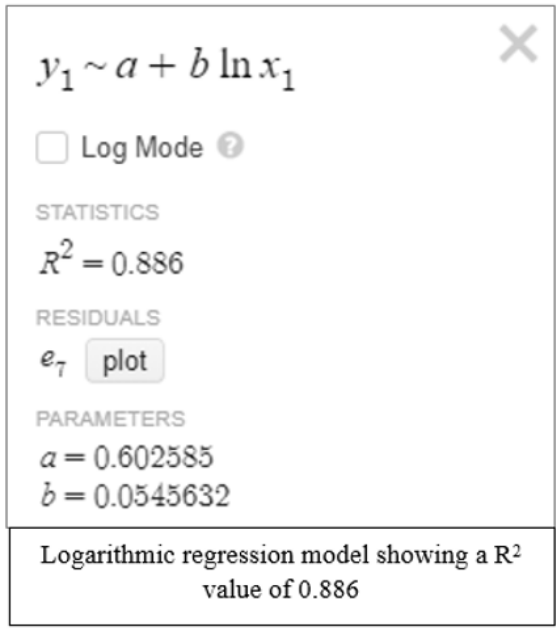
- Linear Function: (y=ax+b)
- Exponential Function: (y=ab^x)
- Logarithmic Function: (y=a+b ln(x))
- Power Function: (y=ax^b)

Whichever of the above four models shows the highest correlation value, that model will be the most suitable for deciding the number of iterations, which is computed by substituting the y value as 1. Here Y value is taken as 1 because it is the highest theoretical value any fitness function can obtain.



Exponential regression model showing a R^2 value of 0.9422.

Figure 3: Exponential model curve for the data in table 2.



Logarithmic regression model showing a R^2 value of 0.886

Figure 4: Logarithmic model curve for data in table 2.

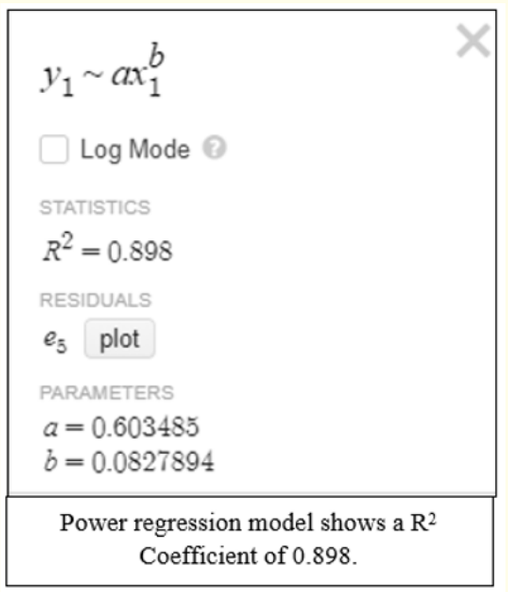


Figure 5: Power model curve for data in table 2.

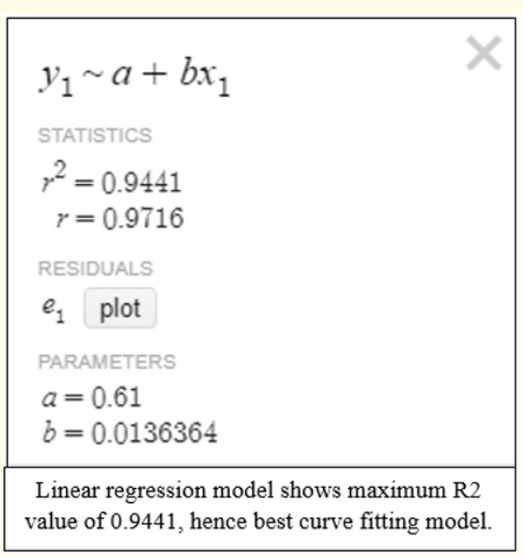


Figure 6: Linear model curve for data in table 2.

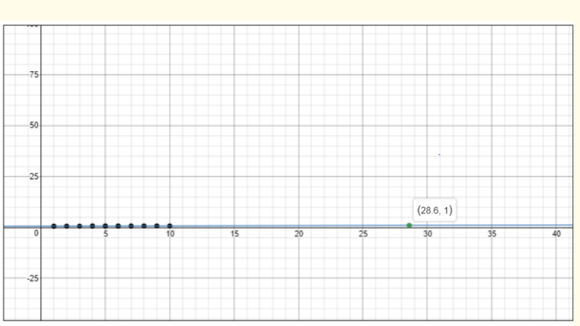


Figure 7: Representation of Linear curve fitted on the graph, with x value=1 for y value=29.

x_1	y_1
1	0.62
2	0.63
3	0.65
4	0.69
5	0.68
6	0.68
7	0.70
8	0.72
9	0.73
10	0.75

Table 2: Table showing sample fitness function values versus generation number.

(A dummy data set for number of iteration(x-axis) versus maximum fitness value for that generation (y-axis) is plotted, showing a loose trend of increasing values).

Various regression models are plotted to provide an approximate value for number of generation when fitness value reaches its theoretical maximum value of '1'.

Since the maximum coefficient of correlation is shown by the linear model here in this case (0.9422), hence its graph is plotted for number of iterations versus maximum fitness value gene generated.

It is seen that the x value becomes 1 for y=28.6, hence a total of 29 iterations more will be done once the fitness value reaches HFV (0.75 in this case).

Quantifying the value of fitness function

The binary string here represents the whole farming technique and hence the fitness function value depends upon various parameters like amount of sunlight, water content provided, fertilizers used, pesticides used and so on but these are input factors and not available in database. Hence output parameters are used to quantify the fitness value (data retrieved from big data set).

Overcoming interdependence of internal factors is a major problem faced. Since agricultural growth and farming outputs are dependent on a number of external factors and these factors don't affect the output independently but can also affect each other in various combinations (interdependence of factors on each other). In order to overcome such intermingling effects of factors over each other, only the final output of growth is taken into account while fitness value calculation. This helps, as whatever interdependence the factors might have on each other it is directly reflected in the output parameters like crop output and growth.

For example: One fertilizer might have a positive effect on that crop independently, and another type of pesticide can also have a positive effect when used independently, but when the fertilizer and pesticide combinations are used it might lead to a degrading

effect over the crop production as a whole due to chemical action of the two over each other. Hence while calculating the fitness function to tackle this problem, instead of analysing the effect of both over each other, only the crop output versus crop input ratio is taken into consideration, which in itself reflects the overall effect of the two factors over the process. If 1000 seedlings are planted and only 200 grow fully due to the two clashing factors, the fitness value corresponding to the above parameter itself becomes 0.2 (200/1000).

Components for calculation of fitness value (FV)

- **Crop output versus Crop Input (F_1)**: In order to quantify the affectivity of farming process as a whole, this factor ensures a consistent and overall healthy crop quality. If "a" seedling is planted and "b" are able to fully grow for harvesting process, then a/b is the F_1 value.
- **Sold in market versus sent to market (F_2):** This factor quantifies the crop quality grown. Only the crop output is not the quantum to declare a farming process as overall effective as it might be a possibility that a high number of seedlings do get harvested but only a few of them are able to match the threshold quality. Hence taking the crops sold in market versus sent to market gives us a number signifying the health of the harvested crops produced.
- **Standard time for yield versus time taken for actual yield (F_3):** Of all the crops taken into account for calculation, one crop must be taking the minimum amount of time to grow fully and its value is taken as 'T_1'. The current generation take a time of harvesting say T_2 (T_2>T_1), hence T_1/T_2 ratio assures that the farming technique gives effective results in a plausible amount of time. As the value of T_2 (time taken for crop generation) increases, the T_1/T_2 factor decreases, signifying that more the amount of time taken, lesser is the F_3 value.
- **Minimum cost for one crop/cost required (F_4):** From the farmer's point of view, not only getting the best crop output in the most practical amount of time is important, but so is the overall capital he has to invest for that process. More the capital required, lesser will be the F_4 value.
- **Self-learning factor (F_5):** From the point of view of the whole farming process, it also matters that how is that technique going to affect the land or other farming resources which might be used in future for some other crops. [15] Hence the self-learning factor is added which is the fitness value (FV) of the past generation mentioned in that database, hence as time passes and number of times the algorithm is implemented, this factor changes as it is in a recursive relation with the fitness value of the past generation

Note that self-learning factor is customized for each farmer as each farmer has his own set of land and other farming resources, and self-learning factor remains constant throughout the one whole technique production, and changes for every farmer as he uses the algorithm more over the span of time. It basically takes in consideration as to what has been done to the land earlier and what will be its aftermath.

Hence F_5 gets better with each phase [19].

Average=(F_1+F_2+F_3+F_4+F_5)/5 (1)

Consistency factor = 1 - (\sum_{i=1}^{n}{(Average-Fi\ ))/n}, For every Fi, less than average (2)

n is the number of factors having value less than the arithmetic mean

Fitness value = Consistency factor * Average (3)

The consistency factor ensures that the FV (fitness value) is not a sheer average of all the values as it is a possibility that a considerably high value of one factor might compensate the lower value of another, hence giving a decent average but the overall farming technique might not be up to mark.

For example: For two factors, consider F_1=0.4 and F_2=0.9. It's average is 0.65 [(0.4+0.9)/2], but the two values are not consistent hence compromising with the overall fitness quality of the technique. For this case:

Consistency factor= 1- (0.65-0.4)/1 =0.75.
Fitness value=0.75*0.65=0.4875.

For a second case take F_1=0.6 and F_2=0.7, here also average=0.65 [(0.4+0.9)/2].

But, consistency factor= 1- (0.65-0.6)/1 =0.95.
Fitness value=0.95*0.65=0.6175.

It is seen that as the data becomes more and more consistent the value of consistency factor increases and it when multiplied with average ensures a higher average value in addition with consistency of values.

Note: In order to maintain a minimum level of fitness of each of the factors (F_1, F_2, F_3, F_4, F_5), if any generation with any of the values less than a threshold value (0.2) is encountered, that whole generation is rejected no matter however high the overall average

might be, as such a low value of any factor renders the whole farming technique ineffective.

Algorithm:

Initialize F_1, F_2, F_3, F_4, F_5
/* 5 factors cumulating to fitness value */

```
Initialize count=0
average= (F1+F2+F3+F4+F5)/5;
X=Average
for (i=1; i<5; i++)
{
if (fi<average)
{
x=x-fi
count=count+1;
}
}
consistency_factor=1-X/count;
fitness_value=consistency_factor*average;
```

Mirroring approach

To find the fitness value of the upcoming generations, since there are no resources in database to calculate the F_1, F_2, F_3, F_4, F_5 values for every possible combination, hence a mirroring approach is applied for finding the FV of that gene. Let the gene G come from mutation of G_1 and G_2 with fitness values FV_1 and FV_2. In order to calculate fitness factor of G (let FV), a, mirror approach is applied by analysing the amount of resemblance it shows to its parent. More the amount of resemblance towards a fitter parent gene, more will be the FV value. For example: if the parent generation have bit strings as 100101 (fitness value FV_1) and 100111 (fitness value FV_2), and let the child offspring's string be 111011 then compare each bit which resembles the parent one, which gives a resemblance factor of 2/6 (2 bits at the same position as the parent gene) for the first one and 3/6 (3 bits at the same position as the parent gene) for the second one, hence its fitness index becomes ((FV_1 × 2/6) + (FV_2× 3/6)). The remaining bits which represent none of the parent genes (loss due to crisscross mutation), the average of FV1 and FV2 is taken. In this case, the remaining (1/6) th portion is taken as((FV_1+FV_2)/2) × (1/6).

Mutation

Once the Fitness value is calculated for every gene in the given generation, mutation of the fitter genes has to be done with higher probability. In order to ensure that the following method is undertaken:

Let the present generation genes have fitness values as FV_1 = 0.4, FV_2 = 0.6, FV_3= 0.75, FV_4 = 0.65.

From the above case the probability of selection of FV_3 should be the highest as it is the gene with highest fitness value (0.75), and probability of picking FV_1 should be the lowest (fitness value =0.4). Hence a fitness function summation (FFS) is computed and probability of selection of each gene:

Probability of selection of a gene for mutation= (fitness value)/ (fitness function summation)

Let it be termed as the P function.
Then,

- P(FV_1) =0.40/ (0.4+0.6+0.75+0.65) =0.17.
- P(FV_2) =0.60/ (0.4+0.6+0.75+0.65) =0.25.
- P(FV_3) =0.75/ (0.4+0.6+0.75+0.65) =0.31.
- P(FV_4) =0.65/ (0.4+0.6+0.75+0.65) =0.27.

In order to facilitate such a selection which is directly proportional to the above given probabilities, a random number generator is used which produces a totally random value ranging from 0 to 1.

- If the value lies in 0-0.17: FV_1 gene is selected for mutation.
- If value lies in 0.17-0.42 (0.17+0.25): FV_2 is selected for mutation.
- If value lies in 0.42-0.73 (0.42+0.31): FV_3 is selected for mutation.
- If value lies in 0.73-1.00 (0.73+0.27): FV_4 is selected for mutation.

The random number generator is executed for a limited number of times (equal to the number of parent genes needed for mutation) and the parent gene hence selected are mutated pair wise

Evolution

In order to facilitate the generation of better off-springs, the above processes of parent generation, fitness value calculation and mutation is repeated until the desired output is reached. The final string hence produced represents the best possible technique for the crop production under the given requirements.

Conclusion

In the present scenario where such a huge amount of population depends directly or indirectly onto farming. Having a big database proves to be helpful over having manual expertise. The genetic algorithm solution proposed provides a way to analyse the big data to produce the most optimum farming technique under the given constraints, with less time complexity. The automation of "farming technique generation process" has the potential to give a large number of farmers (currently lacking technical support) the ability to get the best of agricultural practices with much less hassle, like approaching for manual help. Representing presence or absence of factors as bits and mirroring approach for calculation of fitness factor covers a wide range of crop permutations. Hence using this technology in future and aiming to make it available in hands of as much farmers as possible will give them the ability to get the most optimized production techniques under the given set of constraints most optimally. Another direction for future research includes introduction of smarter and faster IoT based devices which if used in unison with our proposed method [16], will be capable to remove human intervention to a much larger extent. Using genetic algorithms reduces run time for the same purposes in generative adversarial networks from an average of 45 minutes to 10 seconds, only 0.37% of total run time produces such results.

The drawbacks include the cross cutting negative effects that might arise from fusion of multiple parameters such as a pesticide and a fertilizer might work well individually for crop production, but when used together give rise to negative effects on the crop. Keeping this in mind future scope of this work indicates towards addition of computational chemistry tools to better identify their outcomes on the crop output.

Bibliography

1. Paudel M. "Multiple Cropping for Raising Productivity and Farm Income of Small Farmers". *Journal of Nepal Agricultural Research Council* 2 (2016): 37-45.

2. Wolfert S., *et al*. "Big data in smart farming–a review". *Agricultural Systems* 153(2017): 69-80.

3. Dwivedy N. "Challenges faced by the Agriculture Sector in Developing Countries with special reference to India". *International Journal of Rural Studies* 18.2 (2011).

4. https://www.weforum.org/agenda/2017/10/more-than-55-of-indians-make-a-living-from-farming-heres-how-we-can-double-their-income/

5. Gumaste SS., *et al*. "Future weather prediction using genetic algorithm and FFT for smart farming". In Computing Communication Control and automation (ICCUBEA), 2016 International Conference on (2016): 1-6.

6. Prathibha SR., *et al*. "IOT Based Monitoring System in Smart Agriculture". In Recent Advances in Electronics and Communication Technology (ICRAECT), 2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT). (2017).

7. Ruoff E. "Optimizing Crop Land Allocation for Smallholder Farmers in Central Uganda". (2015).

8. Olakulehin OJ and Omidiora EO. "A genetic algorithm approach to maximize crop yields and sustain soil fertility". *Net Journal of Agricultural Science* 2(3) (2014): 94-103.

9. Palmer AH. "The agricultural significance of sunshine as illustrated in California. Monthly Weather Review, 48.3 (1920): 151-154.

10. Soni N and Kumar T. "Study of various mutation operators in genetic algorithms". *International Journal of Computer Science and Information Technologies* 5.3 (2014): 4519-4521.

11. Aktar W., *et al*. "Impact of pesticides use in agriculture: their benefits and hazards". *Interdisciplinary toxicology* 2.1 (2009): 1-12.

12. Hermawanto D. "Genetic algorithm for solving simple mathematical equality problem". (2013).

13. Carr J. "An introduction to genetic algorithms". Senior Project (2014).

14. Walpole Ronald E. "Probability & Statistics for Engineers & Scientists". 9th ed., Prentice Hall (2012).

15. Moula., *et al*. "Effects of Consecutive Two Years Tobacco Cultivation on Soil Fertility status at Bheramara Upazilla in Kushtia District". *Journal of Rice Research* 6.1 (2018): 190.

16. Jayaraman PP., *et al*. "Internet of things platform for smart farming: Experiences and lessons learnt". *Sensors* 16.11 (2016): 1884.

17. Goodman Erik. "Introduction to genetic algorithms". (2007). 3205-3224.

18. Ribarics Pal. "Big Data and its impact on agriculture". Ecocycles. 2 (2016): 33-34.

19. llen Lima., *et al*. "Fitness Function Design for Genetic Algorithms in Cost Evaluation Based Problems". (1996): 207-212.

Volume 3 Issue 10 October 2019