Optimization model for irrigation water management in the Dry Zone of Sri Lanka

A Dissertation Submitted for the Degree of Master of Computer Science

S.K Walpita

University of Colombo School of Computing

2021

UCSC

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis. This thesis has also not been submitted for any degree in any university previously.

Student Name: Sachith Kalinda Walpita

Registration Number: 2016MCS112

Index Number: 16441122

29/11/2021

Signature of the Student & Date

This is to certify that this thesis is based on the work of Mr. SK Walpita under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by,

Supervisor Name: Ms M.W.A.C.R Wijesinghe

30/11/2021

Signature of the Supervisor & Date

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to all those who provided me the possibility to complete this thesis. I offer my special gratitude to my supervisor Ms M.W.A.C.R Wijesinghe, who given me valuable guidance and direction to make this task success.

I would like sincerely to thank all the lectures and the post-graduate department who invested the full effort in achieving the goal.

Last but not the least, I would like to thank you my family and my friends for supporting and encouraging me throughout this thesis and the MSc program.

ABSTRACT

Sri Lanka has a rich irrigation system since ancient times, and the history of irrigation dates back over two millennia and consists of more than thirty thousand lakes. This irrigation system is mainly spread in the Dry Zone in Sri Lanka and mostly using for rice cultivation. In 2019, paddy rice yield for Sri Lanka was 47,954 hg per ha. Paddy rice yield of Sri Lanka increased from 22,485 hg per ha in 1970 to 47,954 hg per ha in 2019 growing at an average annual rate of 1.89%. Irrigation demand for the cultivation is increasing with rising food production. Hence, efficient water management is a key requirement to satisfy the increasing irrigation demand.

The current irrigation plan has failed to manage the limited water resources to satisfy the demand. Recent reports on water management in the Dry Zone indicates the water distribution in not meeting the demands of agriculture in terms of reliability, adequacy & timeliness. Reduced wastage of water is a key factor to improving the efficiency of irrigation water management. The main causes of irrigation water wastage are releasing water in excess of the actual irrigation demand and overestimating irrigation demand. The main reason for over estimating is poor consideration of environment factors effects to the crop water requirement. It is possible to reduce the water issue by calculating the most accurate crop water requirement at a particular stage and making an adjustment to the plan while considering environmental factors. The complexity of the irrigation system is the main reason for inaccurate estimation in irrigation issue planning. In computer science, artificial intelligence is the best solution to resolve complex problems. This study used Genetic Algorithms, which fall under Artificial Intelligent to build an optimization model for irrigation water management. This study developed an optimization model for the Rajanganaya Irrigation System and used field data in the 2015 Yala and 2016 Maha seasons to evaluate the final results of the optimization model. Rajanganaya Irrigation System is located in Anuradhapura district of North Central Province which covers an approximate 2500 Ha area with 39 Km canal network.

In this study, represented the irrigation system in an appropriate way to use the Genetic Algorithm to recommend an optimum irrigation distribution plan. The model was run for various values of population, generations, cross-over, and mutation probabilities. It is found that the appropriate parameters for population size, number of generations, crossover rate and mutation-rate. Furthermore, the cross-over phase in the Genetic Algorithm operation was modified to get the most accurate result and to apply system limitations such as minimum and maximum irrigation supply.

Key words: Genetic Algorithm, Optimization model, Irrigation system, Irrigation demand

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The Mahaweli Development Project gives a tremendous enhancement to the national economy of Sri Lanka with numerous objectives. In 1961 initiated Mahaweli Development program to make irrigation facilities in dry zone cultivation as the major objective. The project area contains nine main Mahaweli zones and covers 39 percent of the whole island and 55 percent of the Dry Zone. Therefore, efficient water management in the Mahaweli zones is very important to the food production and enhancement to the economy of Sri Lanka.

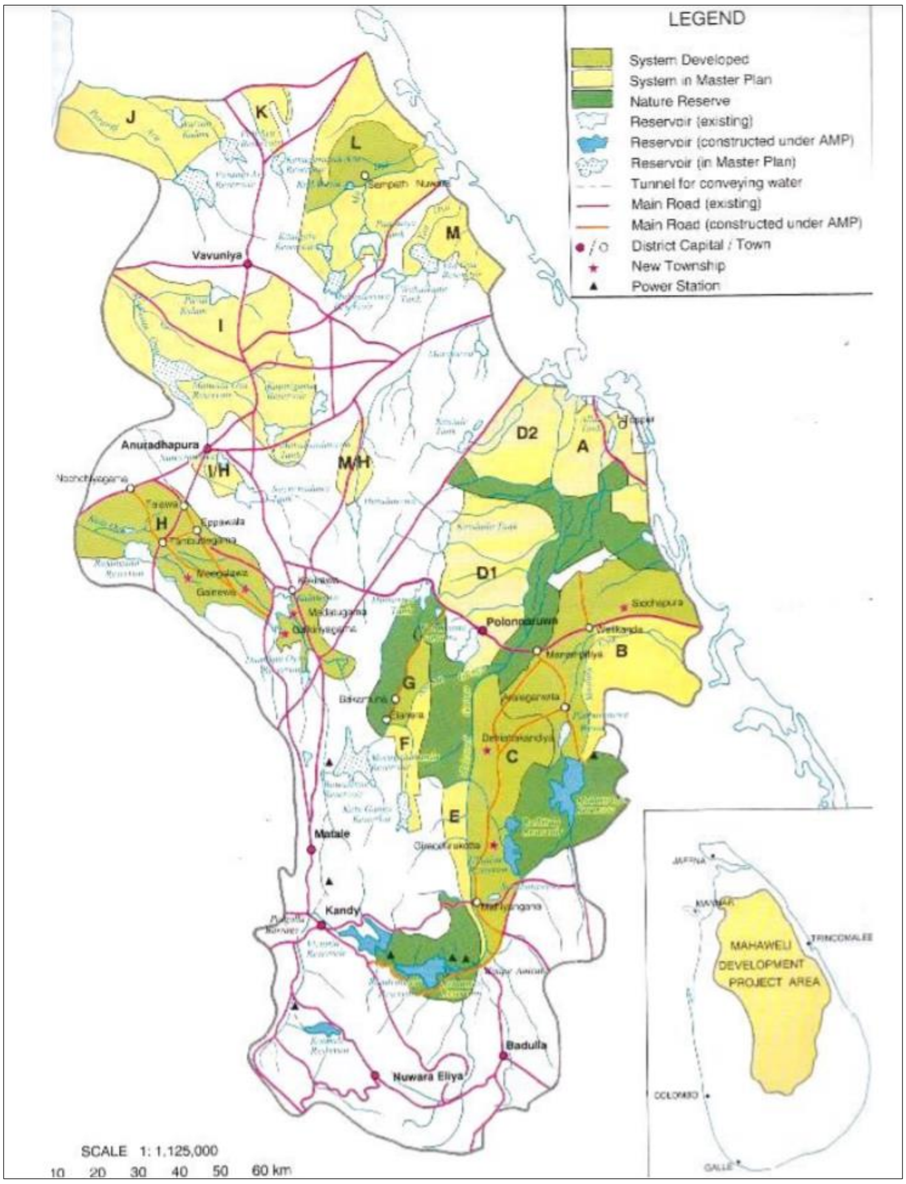


Figure 1. 1 Location of Mahaweli System

On one hand, as a result of the growing population, demand for food and water in both developed and developing countries increases. On the other hand, climate change in the world impacts on water resources and it directly affects water availability and water quality. In dry zone of Sri Lanka approximately 85 percent of irrigation water utilizing for paddy cultivation in two main rainy seasons namely "Maha" and "Yala". The management of collected water in irrigation reservoirs is important to facilitate irrigation water for both Yala and Maha seasons. Different studies indicate the present water distribution mechanism is not meeting the demand for water for cultivation by farmers in terms of timeliness and reliability. This study focused to develop an optimization model to recommend an optimal irrigation scheduling plan for maximum utilization of irrigation water through minimize wastage. An optimization approach based on Genetic Algorithm (GA) to minimize the gap between water demand and supply to minimize wastage of water. Although certain modifications have been applied to the GA to represent the irrigation model and get optimal results from the optimization model.

1.1 Motivation

In Sri Lanka, the agriculture sector is given less priority for ICT related solutions, but other sectors are emerging with ICT. Artificial intelligence is a new technology in the agricultural field that will open a new era. Irrigation management is a very complex issue in Sri Lanka. Therefore, using AI will allow us to make complex decisions through computer learning. As there are many search algorithms used in artificial intelligence, the genetic algorithm is one such algorithm which is used in solution generation in AI. Even though GA used in other fields mostly but not in agriculture. This study shows how GA can be used for solution generation in agriculture. In this study, the solution has been implemented based on the Mahawali H zone, and if the solution is feasible in practical usage, it will be able to apply to the entire irrigation system for effective management of the irrigation water.

1.2 Statement of the problem

Water distribution in irrigation systems should be both equitable and efficient. In current irrigation practice, we are unable to make an equitable and efficient distribution plan due to the complexity of the irrigation system. Equity in water distribution describes that the farms located within a particular irrigation scheme receive the appropriate amount of water per unit of land (Wickramaarachchi, 2002). Wastage of irrigation water should be minimized in an efficient distribution plan and the difference between irrigation demand and supply indicates the amount of wastage. In the current distribution plan, overestimating more than actual irrigation demand is the key factor affects to increase the gap between actual demand and supply of irrigation water. Actual irrigation demand is based on the crop water requirement in a particular area and other environmental factors. In the current distribution plan, crop water requirements are taken into account, but no scientific mechanism is used to account for other environmental factors influencing irrigation demand. Making adjustments to the irrigation water distribution plan based on environmental factors will help to further reduce irrigation water supply and wastage. Therefore, it requires an effective water distribution plan to satisfy the irrigation demand while minimizing the gap between actual demand and supply.

1.3 Research Aims and Objectives

1.3.1 Aim

The aim of this study is to develop an accurate and equitable irrigation distribution plan to achieve maximum efficiency in irrigation water distribution.

1.3.2 Objectives

The major objective in the study is to develop an optimization model based on Genetic Algorithm (GA) to maximize efficiency in irrigation water distribution by minimizing the difference between irrigation demand and supply.

The specific objectives were as follows:

1. To represent irrigation system in according to the genetic algorithm approach.
2. To add modifications to phases in GA to improve accuracy and performance.
3. To evaluate the irrigation plan recommended by Optimization Model with the actual water use in irrigation system.
4. To evaluate GA performance with input parameters

1.4 Scope

The scope of this study is to develop an optimization model for effective irrigation water management in the Mahaweli H zone. The Rajanganaya irrigation system is the main irrigation system in the Mahaweli H zone and is located in the North Central province and the North Western province of Sri Lanka. The system consists of two major canal systems, namely the Left Bank (LB) main canal and the Right Bank (RB) main canal. These two main systems provide irrigation facilities to a 7200 Ha command area and the Left Bank main canal covers 2559 Ha in seven tracts.

For this study, the Left Bank Irrigation Canal System was chosen to develop the Optimization Model based on Genetic Algorithm (GA). The LB canal system area is divided into 7 main blocks and managed by a Block Manager assisted by a Block Irrigation Engineer (BIE).

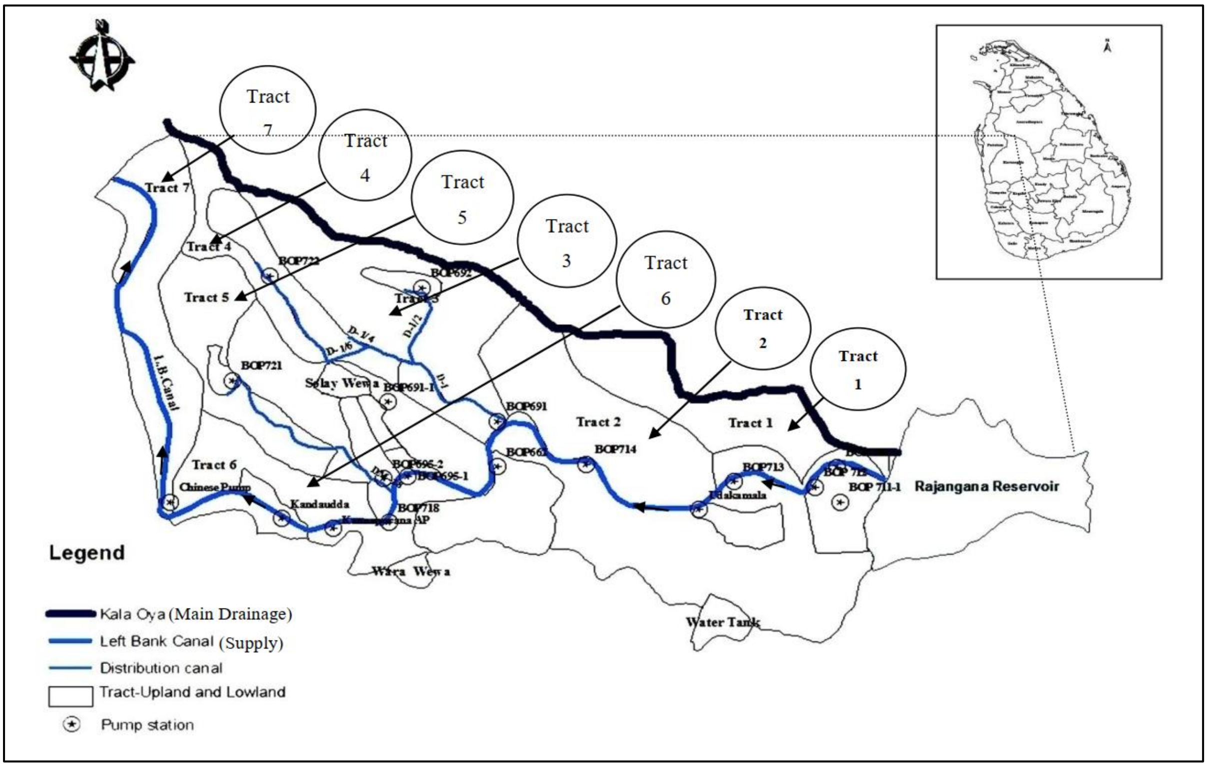


Figure 1.2 Left Bank (L.B.) Canal System of Rajangana Irrigation Scheme

1.5 Structure of the Thesis

Chapter one: Provides a brief introduction to the project background, motivation, problem domain, project objectives with the aim.

Chapter two: Addresses literature review, results of the similar research work with comparing their results and providing an insight into this project.

Chapter three: Defined the methodology of the proposed system.

Chapter four: Brief the aspects such as evaluation methods, designed experiments with the results obtained.

Chapter five: Outlines the conclusion with the limitation of current finding and work for the future.

CHAPTER 2
LITERATURE REVIEW

Sri Lankan farmers are growing paddy which is reaching 3,876,000 MT per annum and it fulfils 90% of national demand which was only 40% in 1950. Sri Lanka has 0.7% approximate rate of population increment which has been increasing the rice consumption about 1.4 MT per year and its effect for increasing rice demand 1.1% per year (H. Chemjong and N.T.S. Wijesekera, 2017). In a case study included the rice yield 4.2 MT/Ha and 4.0 MT/Ha in Yala and Maha season respectively and has indicated that average rice yield 4.5 MT/Ha. But here is potentiality of yield is 7-12 MT/Ha (Ministry of Irrigation, 2010).Water management plays an important role in food production and support to economic growth of country. Paddy cultivation is based on irrigation water and duty in Sri Lanka are 1300 mm and 1750 mm in Maha and Yala respectively (Ministry of Irrigation, 2010).In 2002 T.N Wickramarachchi studied water management in the dry zone of Sri Lanka. In this work, author had noted current irrigation practice in Sri Lanka. Irrigation Department Guidelines using to prepare water scheduling plan. In each season, water schedule is being prepared and there is discussion with farmers for the consensus. Especially, Minor and Medium Systems are being managed by Irrigation Department of Sri Lanka with participation of farmer leaders

Generally, water issue practices depend on the crop types, actual rain, starting time of cropping, time to reach maturity phase and efficiency of channel therefore this can be varied from plan. Hence, it is very important to compare both plan and actual practices of water issue to manage the water efficiently (Wickramaarachchi, 2002) .

A research under the topic of Evaluation of Irrigation Water Issue Practice for Better Water Management at Rajangana Reservoir Sri Lanka, (H. Chemjong and N.T.S. Wijesekera, 2017) carried out and data between 2008-2013 was used in this research to create a theoretical water requirement by recommended a guideline by using 75% of probable rainfall values and it is named as “Recommended Irrigation Plan” .This plan modified in consideration of actual rainfall during seasonal agriculture operations. This modification was named as “Anticipated water use” because this modification represents the actual water issues. This research considering the actual rainfall making adjustments to reduction of water issue by 35% and 8% in Maha and Yala seasons. In this research two methods were used to evaluate the total water demand. One method uses planned water quantities in the beginning of each season and other method use anticipated water use during season with considering the rainfall and evaporation.

Evaporation of Kalawewa-ID and DL1 Agro ecological region was used to calculate the recommended irrigation plan. Rainfall of Rajangana and actual pan evaporation of Maha iiluapallama was used to calculate the Anticipated water use (AWU). The value is compared in monthly and seasonally. Following algorithm was used for calculation.

$$\frac{RIP - ANWU * 100}{ANWU}$$

RIP -Recommended Irrigation Plan

ANWU- Anticipated water use

Research has carried out the study of minor tanks in Trincomalee and this study has recommended to improve the efficiency of water resource uses. In research conducted in Sri Lanka has mentioned the need of good policy framework and commitment of research area in water sector.

A research paper mentioned that reduction of the ponded depth can save 23% water with reduction of only 6% paddy production. It shows that the water volume-based taxes has better results than conventional area based taxes (H. Chemjong and N.T.S. Wijesekera, 2017).

2.1 Machine learning algorithm for irrigation scheduling

Research related to the irrigation demand forecasting shows possibility of develop a model based on machine learning algorithms (K Ullah, M Hafeez, 2011). In order to improve water productivity, water management practices, data driven models in data mining and various hydrological methods have become vital. In order to develop a water demand, forecast model it is mandatory to understand the irrigation system past behavior, behavior of future hydrological attributes and current land trends.

By having a reliable and accurate irrigation water demand forecasting model it will provide information to water users and managers. According to the finding’s decision trees, fuzzy logic, genetic algorithms can be used to perform prediction, clustering, and classification can define as data mining analytical tools. In the study they have used prediction and classification as the data mining tools. As a data classification tool decision tree learn hidden relationships between the classifier and classifying attributes by using the training dataset. Then apply that learned knowledge to the testing dataset.

2.2 Classification Methods

2.2.1 Decision Tree (DT)

In a study carried out by three researchers, they suggested that relationship between class attributes and non-class attributes of database can be extracted using the rules that generated in decision tree (Khan, 2011). These rules are shown in Figure 2.1 that represent unique path from root to each leaf node in the tree.

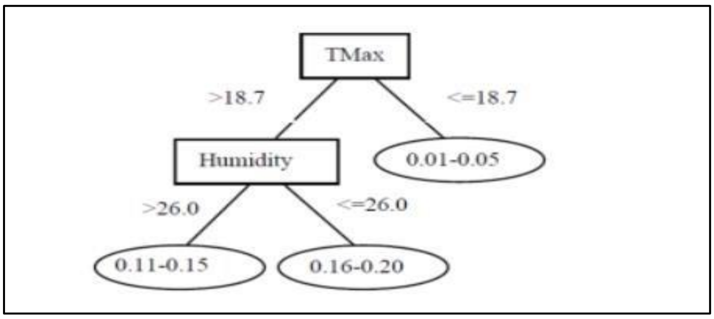


Figure 2.1: An Example of Decision tree generated from our dataset

In this figure 2.1 each node represents an attribute and the class value represented by leaf for the records in decision tree. The decision tree is built using C4.5 algorithm which takes a divide and conquer approach in order to build a decision tree using the training dataset by using the principle of information gain.

2.2.2 Systematically Developed Forest of Multiple Trees (SysFor)

More logical rules and patterns can be developed using a number of trees called forest (Khan, 2011). It was the same as the C4.5 algorithm but there were some modifications such as it got the desired number of trees from the user.

By using the gain ration approach, it then selects attributes that are good for classifying the class values. In order to select which attribute to select as the "good attribute" they determine goodness threshold value. By using the above identified good attributes, use them as root attributes for the tree and many trees. But if the selected good attributes are less than defined number of trees in SysFor then find furthermore in the next level (second level) in the built tree. Because of this SysFor able to develop huge number of trees by using the dataset. In Figure 2a, 2b show example tree that generated in SysFor. By using these trees able to predict class value of future by voting approach. In this approach find leaves where future records falls in and finally leaf that having highest accuracy chose in order to predict class value of the future records.

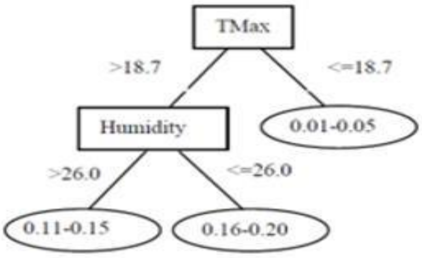


Figure 2.2 a: Tree generated in SysFor based on first good attribute

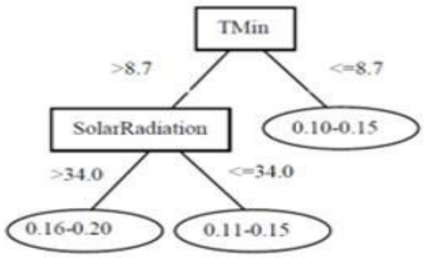


Figure 2.2b: Tree generated in SysFor based on second good attribute

2.2.3 Evapotranspiration (ETc) based Prediction

AP Verma mentioned on an article evapotranspiration can be defined as a process that combination of transpiration of water through the plant tissues and evaporation of water from the soil surface and plant surface (Verma, 2018).

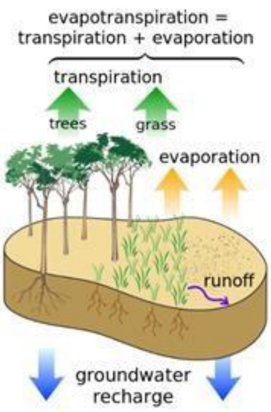


Figure 2.3: Evapotranspiration combined process (Verma, 2018)

In this process overall ET for crop or landscape is the same as the seasonal water requirement in the entire growing season. So, this process defines how much irrigation farmers need in order to maintain the water stress level for the plant.

In the Penman approach ET is calculated based on factors such as local weather condition and crop system which estimates are needed. Knowing about local weather is important because water factors define drying power of air. By using the following factors can measure ET losses:

- Humidity
- Wind
- Temperature
- Solar radiation.

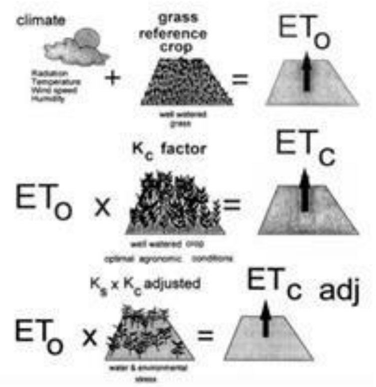


Figure 2.4: Penman Approach (Verma, 2018)

To calculate the Crop Evapotranspiration use reference evapotranspiration (ETo) and crop coefficient Kc. As shown in the above image empirical formula to calculate ETc is ETc= Kc xETo. This method was developed in order to calculate ETc that helps in making irrigation management decisions. (Verma, 2018)

2.3. Genetic Algorithm for irrigation scheduling

Optimization for water delivery schedule in open channel irrigation system can be done by using genetic algorithm (J. B. Nixon, G. C. Dandy and A. R. Simpson, 2001). By identifying the constraints and significant objectives in this system they developed a GA framework to represent them. In this research they identified objectives such as maximizing the number of orders that delivered with a time in requested time and minimize channel flow rate variations. In finding of this research irrigation preferences are to be ±24h rather than ±12h accounted [2. Avoiding exceedance in channel capacity is one of the constraints they have identified. They used this approach in an idealized system in channel spur in five irrigators. In findings of research, they showed that GA techniques are good for irrigation order schedules.

GA can define as a search procedure that based on natural selection and it is a population genetics mechanism. In this reach they define in GA technique it has roots in adaptation and biological process of 'survival of the fittest'. Under certain assumptions it has proven that GA find a global optimum and finite time to find. These results however worth in many GA applications that introduce domain-specific heuristics in violate required assumptions. Many engineering problems have been solved by the finding in GA technique.

2.3.1 Irrigation delivery schedule Genetic Algorithm representation

According to the findings each irrigation places an order to requested in starting time with a specific time duration and with a flow rate. GA was used in order to each order to schedule for a plan day as a string of numbers. In this string integer; value represents number of hours scheduled to be shifted and string represent the order number. Positive order shift represents 'holding off' an order and negative order shift represent 'bringing forward' an order.

Order number (#)	1	2	3	4	5
Order shift (hours)	+2	0	−3	+24	+6

Figure 2.5: A typical GA string for a scheduling with five orders

Figure 2.5 shows a solution of string represents of five irrigation orders that needs to be scheduled. In this research they found some constraints in scheduling such as the number of orders to be scheduled determined by the start time of orders hence by GA strings. And also scheduled start times always remain in planning period of the GA process.

Research carried by Kampanad Bhaktikul has been developed an optimization model based on Genetic Algorithms (GAs) for real time allocation of irrigation water supplies (Bhaktikul Kampanad, 2001).Main objective of that project was to develop a model to minimize the gap between irrigation demand and supply. There were concerns to apply modifications to Genetic Algorithm to implement complex irrigation system.

2.4 Conclusion

Through this chapter discussed the different kinds of researchers that conducted about the irrigation water management which shows the need of an appropriate distribution plan to avoid the waste of water and satisfy the demand of water for the cultivation. Therefore, most of the researchers have been used machine learning as the best solution to develop a simulation model for irrigation water distribution plan according to the demand of water for agriculture based on the collected rainfall data and stoked water capacity within the irrigation system. Machine-learning methods can use to develop simulation model and that model use to enhance the efficiency of these irrigation control approaches due to their capacity to learn from past experiences. Decision trees, fuzzy logic, genetic algorithms can be used to perform prediction, clustering, and classification of simulation model. These algorithms can make better guesses after it has already read some of the data and a trend has been noticed that the algorithm can extrapolate on. In comparing other Machine learning algorithms with the Genetic Algorithm, GA just only require one objective function for calculating of the fitness of an individual while other algorithms require more information to perform a search. In the solution development process, GA can obtain multiple solutions from different generations while other algorithms can generate only one solution. Because of the genetic operators like crossover and mutation GA is more likely to produce global optimal solution.

So, GA process can be defined as a best solution to develop a simulation model to effective irrigation water distribution plan according to the demand of water and effective rainfall in agriculture area.

CHAPTER 3

METHODOLOGY

3.1 Genetic Algorithm (GA)

Genetic Algorithms (GAs) were first introduced by John Holland in 1971.In 1975 Holland published the book titled "Adaptation in natural and artificial systems", which for many years was a standard reference. Since then, GAs have been applied to a wide range of problems (Bhaktikul Kampanad, 2001).Genetic Algorithm is a search procedure based on the natural selection and natural genetics. Genetic Algorithm that answers the problem in this study and make an optimal solution. Characteristics in GA combine the concept of the survival of fittest with genetic operators extracted from nature to develop a robust search mechanism.

Based on coding, fitness evolution and genetic operation can be solved nonlinear optimization problem without any constraints using genetic algorithm. Using this algorithm able to generate initial individuals to high quality population individuals. This each individual represent solution for problem to be solved. Using a fitness function, able to measure the quality of each representation by adaptation to a certain environment. In GA it searches space consists with strings that represent solution to the problem and known as chromosomes. Population can know as chromosomes which have their associated fitness. Following figure (*Figure 3.1*) show a step diagram of GA process.

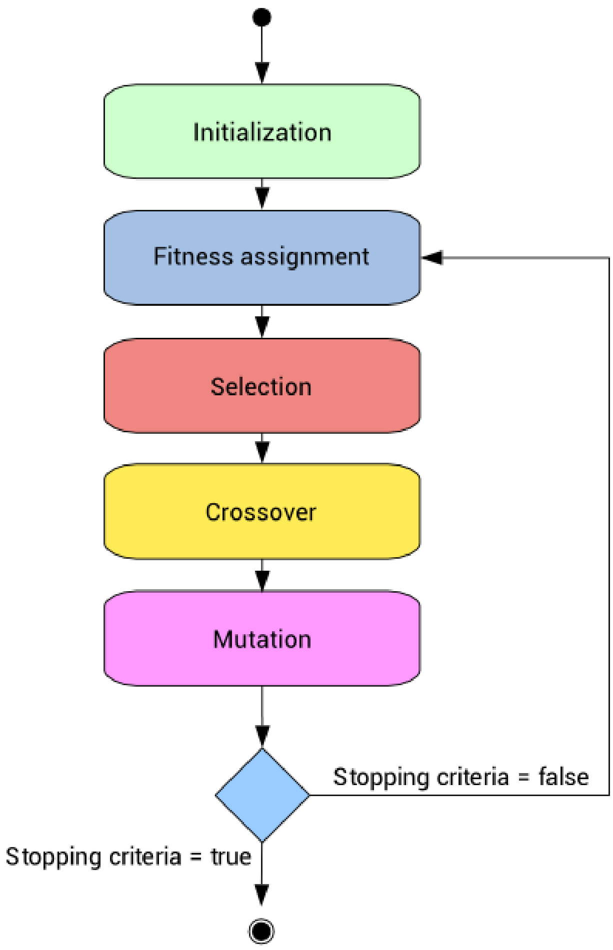


Figure 3.1: Genetic Algorithm process

GA evolves a population of initial individuals to a population of high-quality individuals, where each individual represents a solution of the problem to be solved. The quality of each rule is measured by a fitness function as the quantitative representation of each rule's adaptation to a certain environment. The procedure starts from an initial population of randomly generated individuals.

Procedure: {

1. [Start] Generate random population of n chromosomes (suitable solutions for the problem).
2. [Fitness] Evaluate the fitness f(x) of each chromosome x in the population.
3. [New population] Create a new population by repeating following steps until the new population is complete

a. [Selection] Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected).

b. [Crossover] With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.

c. [Mutation] With a mutation probability mutate new offspring at each locus (position in chromosome).

d. [Accepting] Place new offspring in a new population.

4. [Replace] Use new generated population for a further run of algorithm.

5. [Test] If the end condition is satisfied, stop, and return the best solution in current population.

6. [Loop] Go to step 2.}

3.2 Initialization

Initialization is the first step of the Genetic Algorithm and it is a key factor to the success of the solution. The problem must be defined in an appropriate representation in the initialization process to apply GA operations in next steps. The following steps include the representation of the problem.

1. Identify decision variables
2. Identify system restrictions
3. Representation of irrigation system
4. Chromosome and population representation

3.2.1 Identify decision variables

The development of the optimization model began with the identification of factors affecting the irrigation demand, and these factors are considered as decision variables in this study.

1. Crop Water Requirement (CWR)
2. Effective Rainfall (R_E)
3. Evapotranspiration (ET)
4. Canal Efficiency (CE)

a) Crop Water Requirement (CWR)

Crop Water Requirements (CWR) are defined as the depth of water [mm] needed to meet the water consumed through evapotranspiration (ETc) by a disease-free crop, growing in large fields under non-restricting soil conditions including soil water and fertility, and achieving full production potential under the given growing environment. (L.S.Pereira , I.Alves, 2013)

Precent irrigation practice in Sri Lanka CWR calculates for a day and measured by mm per hectare(mm/ha). CWR is highly dependent on the type of crops and growth stage of crop. Paddy is the main crop type in the study area and 30% irrigation water utilizing for the land preparation process. At the beginning of each season, the Irrigation Department and farmers decide the crop types and total area of cultivation in each block. Then the Block Irrigation Engineer (BIE) calculates the daily CWR for a particular block (CWR_B).

Daily crop water requirement of block (CWR_B);

$$CWR_B = \sum_{i=1}^{n} C_i A_i$$

C_i = CWR of crop type i

A_i= Total land area of crop type i

Total CWR of LB canal system (CWR_{LB});

$$CWR_{LB} = \sum_{i=1}^{7} CWR_{Bi}$$

b) Effective Rainfall (R_E)

Definition of effective rainfall in agriculture is the contributing part of the actual rainfall for cultivation. Effective rainfall depends on the characteristics of soil, crop type and the rainfall duration. The following formula was suggested by the

Irrigation Department in their Irrigation Guideline to calculate the effective rainfall for paddy cultivation in the dry zone of Sri Lanka.

$$R_E = 0.67\ R_F$$

R_E = Effective rainfall in mm

R_F = Actual rainfall in mm

c) Evapotranspiration (ET)

Evapotranspiration can be defined as a process that combination of transpiration of water through the plant tissues and evaporation of water from the soil surface and plant surface. Modified Penman method recommended by Department of Irrigation to calculate ET values.

$$\text{Evapotranspiration (ET)} = \text{Evaporation} + \text{Transpiration}$$

d) Canal Efficiency (CE)

Condition and the hydraulic technology of canals in the irrigation system cause to lose significant quantities of water. In gravity flow hydraulic technology, overall canal conveyance efficiency of 70% as recommended by Department of Irrigation (Wickramaarachchi, 2002)

$$CE = 70\ \%$$

3.2.2 Identify system restrictions

Irrigation water supply for each block (R_B) should be greater than or equal to the irrigation demand (d_i) of that Block.

$$d_i \leq R_B$$

The total releases from the reservoir (R_T) cannot exceed the canal capacity (CC)

$R_T \leq CC$

Capacity of canal masher in $m^3\ s^{-1}$

3.2.3 Representation of Irrigation system

The irrigation system should be represented in an appropriate way to build an optimization model using Genetic Algorithms. Each tract in the LB canal system contains field canals (FC) and distributary canals (DC) and provides irrigation water for a particular command area. This particular command area was considered as a node (Gene) to represent the irrigation network as following.

Tract No	Node no (Gene)	Name of Canal	Command Area (Ha)	Canal length (km)
Tract 1	**1**	FC1	6.47	0.21
	2	FC2	23.96	0.20
	3	D1	124.64	1.10
	4	D2	140.02	2.40
Tract 2	**5**	FC1	32.54	0.20
	6	D1	98.74	1.73
	7	D5	52.61	0.65
	8	D2	153.78	0.52
Tract 3	**9**	D1	575.46	2.31
	10	D2	124.64	2.65
Tract 4	**11**	Sole Wewa	275.19	3.91
Tract 5	**12**	FC1	21.04	0.22
	13	D1	503.39	7.60
Tract 6	**14**	FC1	22.66	0.44
	15	FC3	14.57	0.27
	16	FC5	28.33	0.46
	17	D5	47.75	0.74
	18	FC18	40.47	4.93
	19	FC26	20.23	0.28
Tract 7	**20**	FC1	2.21	0.02
	21	FC2	24.00	0.80
	22	FC9	15.11	0.73
	23	FC14	33.00	0.58
	24	FC19	27.30	0.83
	25	FC28	7.91	0.37
	26	FC29	7.11	0.56
	27	FC30	5.51	0.27
	28	FC31	3.81	0.01
	29	FC33	5.51	0.26
	30	FC34	7.11	0.26
	31	FC35	6.31	0.45

	32	FC36	7.91	0.17
	33	FC38	34.90	0.77
	34	FC42	16.71	0.55
	35	FC48	14.31	0.81
	36	FC49	7.11	0.39
	37	FC54	37.10	0.49
Total of Left Bank Canal:			**2,559.44**	**39.12**

Table 3.1: Representation of Irrigation system

3.2.4 Chromosome and population representation

Representation of chromosome affects to the performance and accuracy of GA solution. A problem specific representation gives more efficiency to the algorithm. Chromosome can represent in three ways such as binary, gray and real value representation. The binary representation can be improved by introducing more bits to the chromosome string, but this has the effect of slowing down execution time, and also making GA operations more difficult. Real value representation simplifies the GA operation and leads to faster execution than other representation. Therefore, real values were used to represent the chromosome. Irrigation supply to a node (x_i) selected as the gene value of a chromosome and each chromosome contains 37 genes. Determine the optimum population size for the problem affects to the accuracy of GA. Increasing the population size increases the number of crossover operations and also increases the accuracy of the GA. In this study population size can be parameterized in GA operations and 200 was selected as the initial value of parameter. To initialize the value of gene (x_i) used a ratio as a parameter (R_d) and based on history data found this ratio does not exceed 1.5.

$$x_i = R_d\ (d_i) \qquad R_d \leq 1.5$$

x_i= irrigation supply to node(gene) i
R_d= irrigation supply ratio
d_i= irrigation demand of node(gene) i

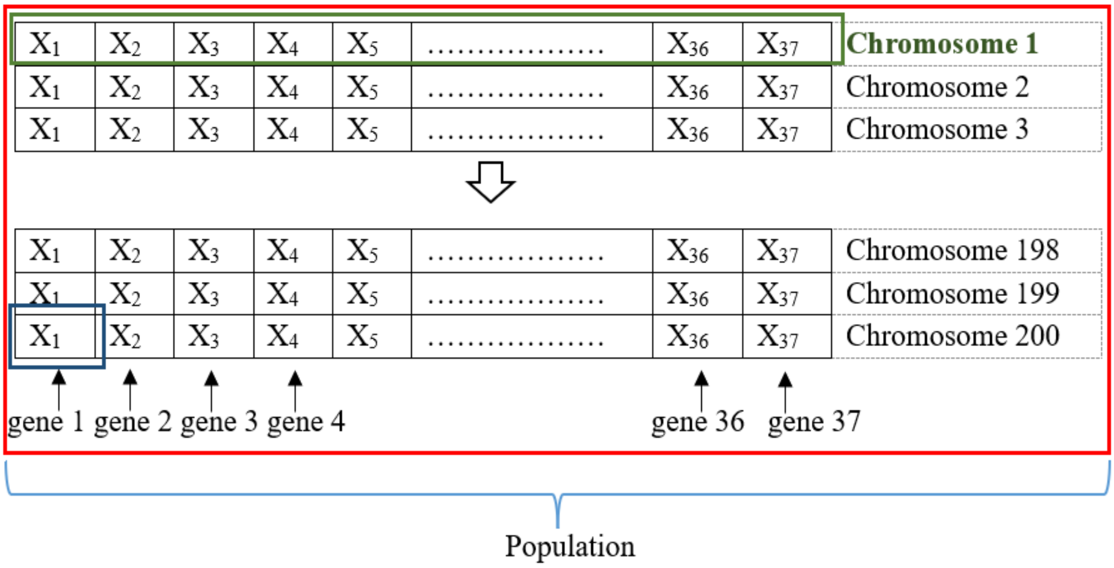


Figure 3.2 – Chromosome and population representation

3.3 Fitness assignment

Continuing the analogy of natural selection in biological evolution, the fitness function determines how fit an individual organisms adapt to generate next population. It is the only step in the algorithm that determines how the chromosomes will change over time, and can mean the difference between finding the optimal. The objective function is used to calculate fitness score, and its definition is critical to a successful GA solution.

The formulation of an objective function is a critical step in the Genetic Algorithm. This study formulated the objective function to minimize water wastage as following.

Estimating the daily irrigation demand of node i (d_i)

$$d_i = \frac{[\,\mathrm{CWR}_i - (\,\mathrm{RE}_i - \mathrm{ET}_i)A_i\,]}{CE}$$

CWR_i	= Crop water requirement of node i	CE	= Canal efficiency
RE_i	= effective rainfall of node i	A_i	= Area of node i
ET_i	= Evapotranspiration of node i		

To minimize the wastage of water, should minimize the gap between the actual irrigation demand and the irrigation water supply. The following objective function (Z) is expressed as the most appropriate formula to minimize the wastage of irrigation water.

$$Minimise\ \ Z = \sum_{i=0}^{n} \frac{(d_i - x_i)^2}{d_i}$$

d_i= irrigation demand of node i x_i= irrigation supply to node i
n= number of nodes

Fitness score of Chromosome (F_c);

$$F_c = \frac{1}{1 + Z}$$

3.4 Selection

The idea of the selection phase of GA is to select the fittest individuals and let them pass their better genes to the next generation. Two pairs of individuals are selected based on their fitness scores. Individuals with high fitness have greater probability to be selected for reproduction of new generation. In GA using three methods for selection such as Roulette Wheel Selection, Stochastic Universal Sampling (SUS) and Tournament Selection. In this study used Roulette Wheel Selection method to select individuals(parents) with high fitness.
For example defined 6 chromosomes including 6 genes as below,

Irrigation Demand	**8**	**10**	**26**	**17**	**15**	**16**
Chromosome 1	10	12	32	23	16	20
Chromosome 2	12	16	24	25	18	21
Chromosome 3	14	18	30	21	20	19
Chromosome 4	8	10	28	20	22	22
Chromosome 5	10	18	38	18	14	18
Chromosome 6	12	15	30	33	16	24

Table 3.2: Initial chromosomes

1st step : Compute the fitness score (Fi) value for each chromosome

Z (1) $= (10\text{-}8)^2/10 + (12\text{-}10)^2/12 + (32\text{-}26)^2/32 + (23\text{-}17)^2/23 + (16\text{-}15)^2/16 + (20\text{-}16)^2/20$

$=4.248$

F (1) $= 1/\ (1+Z\ (1))$

= 0.19

F (2) = 0.111

F (3) = 0.098

F (4) = 0.183

F (5) = 0.109

F (6) = 0.066

Total = 0.757

2nd step : The probability for each chromosome is formulated by:

P[i] = **F**[i] / **Total**

P (1) = 0.19 / 0.757 = 0.25

P (2) = 0.146

P (3) = 0.13

P (4) = 0.241

P (5) = 0.144

P (6) = 0.087

3rd step : Compute the cumulative probability values (Ci):

C (1) = 0.25

C (2) = 0.146 + 0.25 = 0.396

C (3) = 0.13 + 0.396 = 0.526

C (4) = 0.241 + 0.526 = 0.767

C (5) = 0.144 + 0.767 = 0.911

C (6) = 0.087 + 0.911 = 1

4th step : To process roulette-wheel process generate random number R in the range 0-1 as follows

R (1) = 0.201

R (2) = 0.284

R (3) = 0.099

R (4) = 0.822

R (5) = 0.398

R (6) = 0.501

5th step : If random number **R** (k ;k=1,2,3,4,5,6) is greater than **C**(i) and smaller than **C**(i+1) then selects

Chromosome(i) as a chromosome in the new population for next generation:

```
for (int k = 0; k < 6; i++) {
    for (int i = 0; i< 6; i++) {
        if (R [k] <= C[i]) {
            PopulationChromosomes.Add(Chromosomes[i]);
            break;
        }
    }
}
```

Chromosome 1	10	12	32	23	16	20
Chromosome 2	12	16	24	25	18	21
Chromosome 1	10	12	32	23	16	20
Chromosome 5	10	18	38	18	14	18
Chromosome 3	14	18	30	21	20	19
Chromosome 3	14	18	30	21	20	19

Table 3.3: Selected chromosomes for the new population

Chromosomes in the population thus became:

Chromosome 1	10	12	32	23	16	20
Chromosome 2	12	16	24	25	18	21
Chromosome 3	10	12	32	23	16	20
Chromosome 4	10	18	38	18	14	18
Chromosome 5	14	18	30	21	20	19
Chromosome 6	14	18	30	21	20	19

Table 3.3: Chromosomes in the new population

3.5 Crossover

Crossover is the most important phase in a genetic algorithm used to combine the genetic information of two parents to generate new offspring. Different types of crossover operators used in real coded GA such as Single point crossover, Linear crossover, Blend crossover and simulated binary crossover. In this study used Blend crossover operator with a single random point. In GA the number of mate chromosomes is controlled using crossover rate. Example stated in section 3.2 is continuing as bellow;

6th step : Parent chromosome which will mate is randomly selected based on crossover rate. Generate random numbers R (k) as follow

R (1) = 0.191

R (2) = 0.259

R (3) = 0.760

R (4) = 0.006

R (5) = 0.159

R (6) = 0.340

7th step : If **R(k) < crossover rate** selects Chromosome (k) as a parent.

In this study crossover rate used as a GA parameter.

If crossover rate= 25% (0.25)

C (1) , C(4) and C(5) will be selected for crossover

Chromosome 1	10	12	32	23	16	20
Chromosome 4	10	18	38	18	14	18
Chromosome 5	14	18	30	21	20	19

Table 3.5: Eligible chromosomes for crossover

8th step : If eligible chromosomes count is an odd number, add the first eligible chromosome to end of the eligible list

Chromosome 1	10	12	32	23	16	20
Chromosome 4	10	18	38	18	14	18
Chromosome 5	14	18	30	21	20	19
Chromosome 1	**10**	**12**	**32**	**23**	**16**	**20**

Table 3.6: Modified eligible chromosomes list for crossover

9th step : Select parent pairs for crossover

Chromosome 1 x Chromosome 4

Chromosome 5 x Chromosome 1

10th step : The next process is choosing the position of the crossover point. To choose the position generating random numbers (number of parent pairs) between 1 to length of chromosome -1

R (1) = 3

R (2) = 4

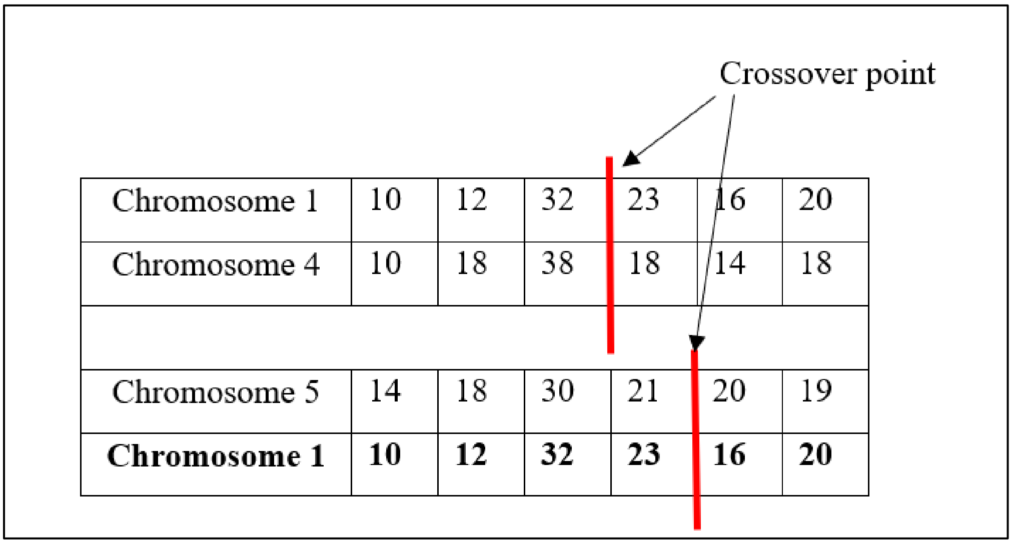


Figure 3.3 – Crossover points

11th step : After selection of a crossover point, next process is determining the gene value of child. According to previous studies average crossover mechanism in real coded genetic algorithm has many advantages over uniform crossover. (Anantkumar J. Umbarkar , P. D. Sheth, 2015).Blend crossover is an extension of average crossover to improve the accuracy of GA operation. Blend crossover as opposed to averaging, chooses a child value at random between parents.

Given the two parents x_1 and x_2 where $x_1 < x_2$, the blend crossover randomly selects a child in the range R [$x_1-\alpha(x_2-x_1)$, $x_2+\alpha(x_2-x_1)$]. To apply system restriction mentioned on section 3.2.2 modified the Blend crossover as bellow and chosen α as 0.5

Min value of range (R_{min}),

If R_{min} < Irrigation demand of node i

R_{min} = Irrigation demand of node i

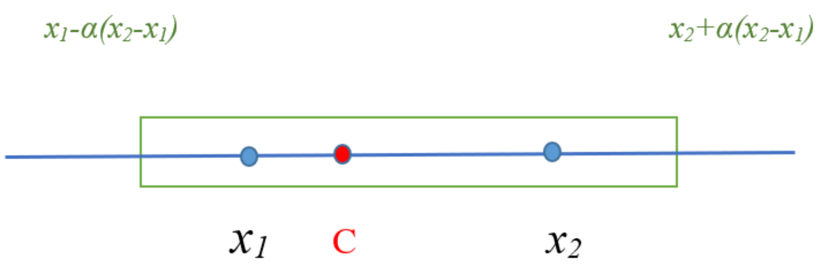


Figure 3.4 – Select child value in Blend crossover

R [1] = 10-0.5(10-10),10+0.5(10-10) = [10,10]

If R_{min} equal to R_{max} inherits parents value to child

C [1] =10

R [2] = 12-0.5(18-12),18+0.5(18-12) = [9,21]

C [2] = 15

R [3] = 12-0.5(32-38),38+0.5(32-38) = [29,41]

C [3] = 35

R [4] = 18-0.5(23-18),23+0.5(23-18) = [15.5,25.5]

C [4] = 20

R [5] = 14-0.5(16-14),16+0.5(16-14) = [13,17]

C [5] = 15

R [6] = 18-0.5(20-18),20+0.5(20-18) = [17,21]

C [6] = 18

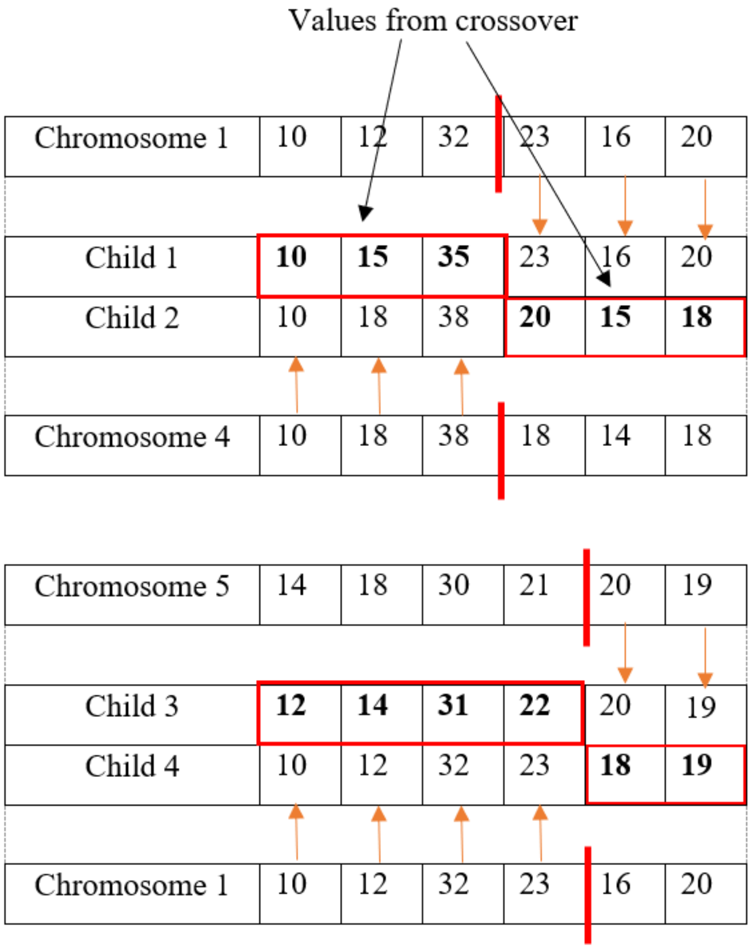


Figure 3.5 –Creation of new child in crossover operation

The new offspring are added to the population and then population thus became:

Chromosome 1	10	15	35	23	16	20
Chromosome 2	12	16	24	25	18	21
Chromosome 3	10	12	32	23	16	20
Chromosome 4	10	18	38	20	15	18
Chromosome 5	12	14	31	22	20	19
Chromosome 6	10	12	32	23	18	19

Table 3.7: New chromosomes after crossover

3.6 Mutation

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of GA chromosomes to the next. Mutation preventing the population of chromosomes from becoming too similar to each other. Mutation probability M_r involves to control number mutation occurs during evolution.

Number of genes (G_n) = Population size x Chromosome size

Number of mutations = G_n x M_r

12th step : In this study mutation rate is a parameter in GA evolution. According to the example in previous section;

Number of mutations = 6 x 6 x 5% =1.8 ≈ 2

To select mutation points generates two (number of mutations) random numbers between 1 to G_n (number of genes)

R [1] =11 R [2] =21

Chromosome 1	10	15	35	23	16	20
Chromosome 2	12	16	24	25	18	21
Chromosome 3	10	12	32	23	16	20
Chromosome 4	10	18	38	20	15	18
Chromosome 5	12	14	31	22	20	19
Chromosome 6	10	12	32	23	18	19

Figure 3.6 –Mutation points

13th step : To decide the mutation value generates random values between minimum irrigation supply and maximum irrigation supply of node i

Min irrigation supply of node 3 = 24

Maximum irrigation supply of node 3 = 38

R [24,38] = 28

Min irrigation supply of node 5 = 15

Maximum irrigation supply of node 5 = 20

R [15,20] = 16

14th step : Replace Gene values by mutation values

Chromosome 1	10	15	35	23	16	20
Chromosome 2	12	16	24	25	16	21
Chromosome 3	10	12	32	23	16	20
Chromosome 4	10	18	28	20	15	18
Chromosome 5	12	14	31	22	20	19
Chromosome 6	10	12	32	23	18	19

Figure 3.7 –Replace gene values from mutation

Finishing mutation process completing the first iteration of GA operation. After first iteration can evaluates fitness score of each chromosome.

Chromosome	Fitness score before 1st Iteration	Fitness score after 1st Iteration
Chromosome 1	0.190	0.128
Chromosome 2	0.111	0.132
Chromosome 3	0.098	0.189
Chromosome 4	0.183	0.184
Chromosome 5	0.109	0.140
Chromosome 6	0.066	0.185
Total	**0.757**	**0.958**

Table 3.8: Fitness score after first iteration

The above comparison shows the total fitness value has been increased in the new population. This GA process such as evaluation, selection, crossover and mutation will be repeated until meet the termination condition.

3.7 Termination

The termination condition of a Genetic Algorithm is important in determining when a GA process will terminate. In GA using following conditions to terminate the GA run;

- If no improvement in the population
- If an absolute number of generations has reached a predefined value
- If fitness score value has reached a predefined value

In this study used predefined number of generations to terminate the GA process. After the termination, it selects the best chromosome from the last generation as the final outcome

Irrigation Demand	**8**	**10**	**26**	**17**	**15**	**16**
Irrigation supply (output)	**9**	**11**	**25**	**18**	**16**	**18**

Figure 3.4 –Final outcome

3.8 Implementation

Develop an application to implement the optimization model is an objective of this study. This application will be used to insert inputs to the GA and evaluate the final result. Following technical stack used to implement the application;

Processor	Intel(R) Core (TM) i5-8250U CPU @ 1.60GHz
RAM	8GB
Operating system	Windows 10
Library	.Net core 3.1
Language	C#
Database	MS SQL Server 2016

Table 3.9 –Technical stack

3.8.1 Application Inputs

This optimization model uses two types of inputs

1. Inputs related to the irrigation operation
    a. Daily Crop Water Requirement (CWR)
    b. Rainfall (RE)
    c. Evapotranspiration (ET)
    d. Canal Efficiency (CE)
    e. Maximum expected wastage
    f. Date
2. Parameters related to the Genetic Algorithm
    a. Crossover rate
    b. Mutation rate
    c. Number of chromosomes within a generation
    d. Number of generations

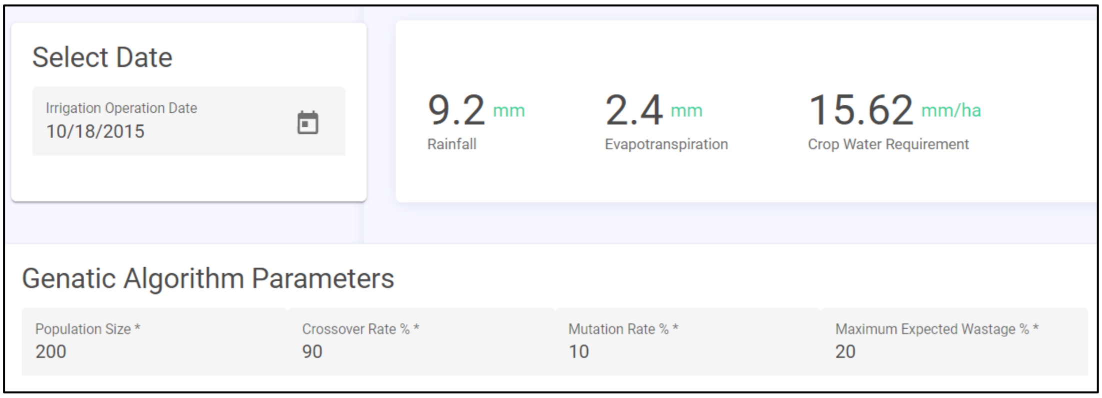


Figure 3.8 –Sample input

The present study used field data of the Rajangana Irrigation System and its left bank canal in 2015 Maha and 2016 Yala seasons. Department of Irrigation (ID) is the Government organization responsible for the Rajangana reservoir system. Following data collected from ID Rajangana, ID Anuradhapura and ID Colombo for above two seasons.

Data	Station	Data source
Detail of area	Rajangana	ID Rajangana
Cultivation area data	Rajangana	ID Rajangana
Blocking Out Plans	Rajangana Irrigation Scheme	ID-Anuradhapura
Issue Trees	Rajangana Irrigation Scheme	ID Rajangana
Command area data	Rajangana	ID Rajangana
Daily Crop Water Requirement	Rajangana	ID Rajangana
Pan Evaporation	Maha-Illupallama	ID Rajangana & ID Colombo
Daily rainfall data	Rajangana	ID Rajangana

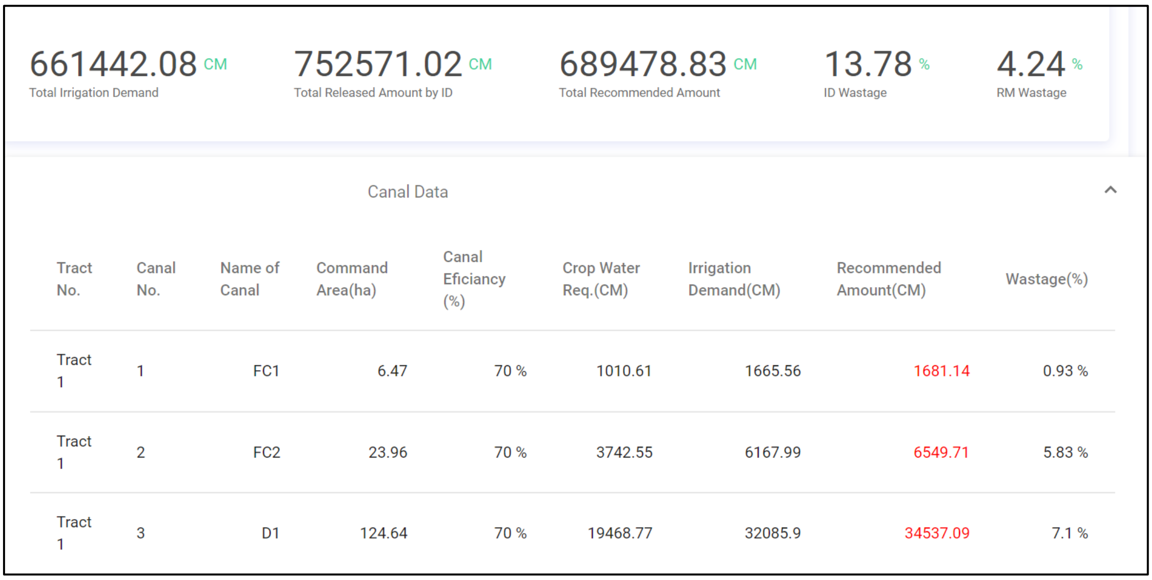
Table 3.10 –Data sources

Above data should preprocess and dimensionality reduce to use as decision variables in optimization model and to evaluate the recommended Irrigation plan against the current irrigation plan. Dimensionality reduction is transforming the existing data to new dimensions without losing the key information. Sometimes large-scale problems bring several dimensions to become difficult to visualize. Because that those kinds of dimensions need to be drop in order to visualize. In this selected dataset also need to conduct the dimensionality reduction because that every data no needs in the decision-making process.

3.8.1 Application Outcomes

The application returns two types of outcomes based on the above inputs.

1. Outcomes related to daily irrigation operation
   - I. Total irrigation demand
   - II. Actual water released to the Irrigation system by ID
   - III. Recommended irrigation supply to each irrigation canal and total amount
   - IV. Wastage in actual water released
   - V. Wastage in optimization model
2. Outcomes to evaluate GA performance
   - I. Fitness value of each generation
   - II. Execution time to get final result



661442.08 CM Total Irrigation Demand

752571.02 CM Total Released Amount by ID

689478.83 CM Total Recommended Amount

13.78 % ID Wastage

4.24 % RM Wastage

Canal Data

Tract No.	Canal No.	Name of Canal	Command Area(ha)	Canal Eficiancy (%)	Crop Water Req.(CM)	Irrigation Demand(CM)	Recommended Amount(CM)	Wastage(%)
Tract 1	1	FC1	6.47	70 %	1010.61	1665.56	1681.14	0.93 %
Tract 1	2	FC2	23.96	70 %	3742.55	6167.99	6549.71	5.83 %
Tract 1	3	D1	124.64	70 %	19468.77	32085.9	34537.09	7.1 %

Figure 3.10 –Output in daily Irrigation operation

Date	2015-10-18						
Tract No.	Canal No.	Name of Canal	Command Area(ha)	Crop Water Req. (CM)	Irrigation Demand (CM)	**Recommended Amount (CM)**	**Wastage (%)**
Tract 1	1	FC1	6.47	1010.61	1665.56	**1681.14**	**0.93%**
Tract 1	2	FC2	23.96	3742.55	6167.99	**6549.71**	**5.83%**
Tract 1	3	D1	124.64	19468.77	32085.9	**34537.09**	**7.10%**
Tract 1	4	D2	140.02	21871.12	36045.14	**38567.76**	**6.54%**
Tract 2	5	FC1	32.54	5082.75	8376.73	**9400.29**	**10.89%**
Tract 2	6	D1	98.74	15423.19	25418.5	**28332.15**	**10.28%**
Tract 2	7	D5	52.61	8217.68	13543.31	**13885.2**	**2.46%**
Tract 2	8	D2	153.78	24020.44	39587.37	**41460.16**	**4.52%**
Tract 3	9	D1	575.46	89886.85	148139.8	**149855.4**	**1.14%**
Tract 3	10	D2	124.64	19468.77	32085.9	**33898.09**	**5.35%**
Tract 4	11	Sole Wewa	275.19	42984.68	70841.77	**71816.67**	**1.36%**
Tract 5	12	FC1	21.04	3286.45	5416.3	**5621.09**	**3.64%**
Tract 5	13	D1	503.39	78629.52	129587	**134625.1**	**3.74%**
Tract 6	14	FC1	22.66	3539.49	5833.33	**6125.94**	**4.78%**
Tract 6	15	FC3	14.57	2275.83	3750.73	**3982.48**	**5.82%**
Tract 6	16	FC5	28.33	4425.15	7292.96	**7606.02**	**4.12%**
Tract 6	17	D5	47.75	7458.55	12292.21	**12514.61**	**1.78%**
Tract 6	18	FC18	40.47	6321.41	10418.13	**10489.2**	**0.68%**
Tract 6	19	FC26	20.23	3159.93	5207.79	**5755**	**9.51%**
Tract 7	20	FC1	2.21	345.2	568.91	**619.98**	**8.24%**
Tract 7	21	FC2	24	3748.8	6178.29	**6779.98**	**8.87%**
Tract 7	22	FC9	15.11	2360.18	3889.74	**4218.42**	**7.79%**
Tract 7	23	FC14	33	5154.6	8495.14	**8896.26**	**4.51%**
Tract 7	24	FC19	27.3	4264.26	7027.8	**7619.82**	**7.77%**
Tract 7	25	FC28	7.91	1235.54	2036.26	**2107.09**	**3.36%**
Tract 7	26	FC29	7.11	1110.58	1830.31	**1903.21**	**3.83%**
Tract 7	27	FC30	5.51	860.66	1418.43	**1563.35**	**9.27%**
Tract 7	28	FC31	3.81	595.12	980.8	**1054.52**	**6.99%**
Tract 7	29	FC33	5.51	860.66	1418.43	**1473.82**	**3.76%**
Tract 7	30	FC34	7.11	1110.58	1830.31	**1893.15**	**3.32%**
Tract 7	31	FC35	6.31	985.62	1624.37	**1724.89**	**5.83%**
Tract 7	32	FC36	7.91	1235.54	2036.26	**2127.62**	**4.29%**
Tract 7	33	FC38	34.9	5451.38	8984.26	**9968.85**	**9.88%**
Tract 7	34	FC42	16.71	2610.1	4301.63	**4512.31**	**4.67%**
Tract 7	35	FC48	14.31	2235.22	3683.8	**3797.09**	**2.98%**
Tract 7	36	FC49	7.11	1110.58	1830.31	**1964.63**	**6.84%**
Tract 7	37	FC54	37.1	5795.02	9550.6	**10550.68**	**9.48%**
TOTAL				**401343.38**	**661442.1**	**689478.8**	**4.24%**

Table 3.11 –Sample output for a day

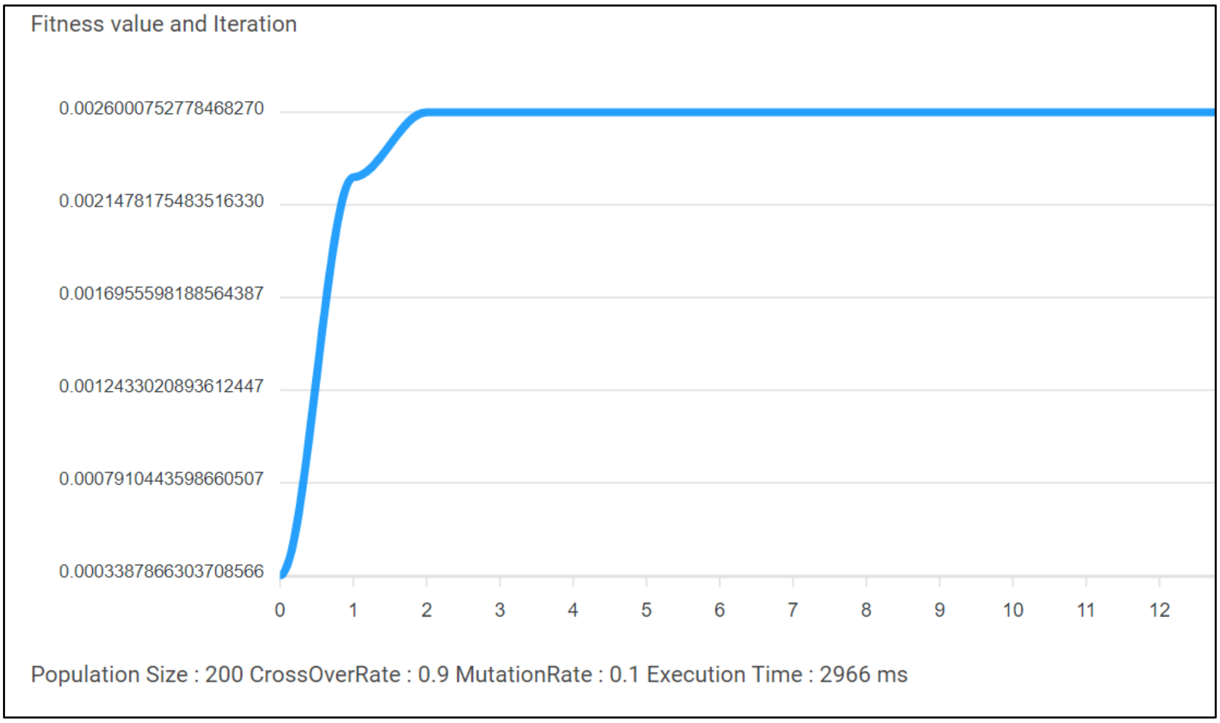


Figure 3.11 –Execution time and Fitness score of each generation

CHAPTER 4

EVALUATION AND RESULTS

This chapter precents the final results of the Optimization Model and analyzed results using appropriated statistical methods.

4.1 Equitability of recommended irrigation distribution plan.

Making an equitable irrigation distribution plan is a part of aim in this study. To make irrigation distribution plans equitable, they should satisfy the irrigation demand of each command area with recommended irrigation supply. Table 4.1 shows recommended irrigation supply to each command area by Optimization Model for a particular date.

Date	01/01/2016		
Command area	Irrigation Demand (CM)	Recommended Irrigation supply (CM)	Wastage (%)
1	605.41	609.38	0.655754
2	2241.97	2243.64	0.074488
3	11662.74	11743.38	0.691433
4	13101.87	13122.31	0.156008
5	3044.81	3057.24	0.408236
6	9239.24	9242.29	0.033011
7	4922.79	4996.34	1.494071
8	14389.41	14427.26	0.263041
9	53846.61	53977.48	0.243042
10	11662.74	11740.73	0.668711
11	25749.92	25842.35	0.358953
12	1968.74	2001.86	1.682294
13	47102.92	47125.75	0.048468
14	2120.33	2153.68	1.572868
15	1363.34	1363.8	0.033741
16	2650.88	2674.53	0.892157
17	4468.04	4494.72	0.59713
18	3786.84	3791.8	0.13098
19	1892.95	1923.46	1.61177
20	206.79	208.08	0.623821
21	2245.71	2251.86	0.273855
22	1413.86	1414.62	0.053754
23	3087.86	3096.17	0.269118
24	2554.5	2590.21	1.397925

25	740.15	743.66	0.474228
26	665.29	672.14	1.029626
27	515.58	516.49	0.1765
28	356.51	370.7	3.980253
29	515.58	529.53	2.705691
30	665.29	668.81	0.529093
31	590.44	594.97	0.767224
32	740.15	750.61	1.413227
33	3265.64	3269.94	0.131674
34	1563.58	1586.99	1.497205
35	1339.01	1340.65	0.122479
36	665.29	666.71	0.213441
37	3471.5	3535.9	1.855106

Table 4.1: Recommended irrigation supply in each command area

The result in table 4.1 shows that the recommended irrigation supply has been satisfying the irrigation demand of each command area with minimum wastage.

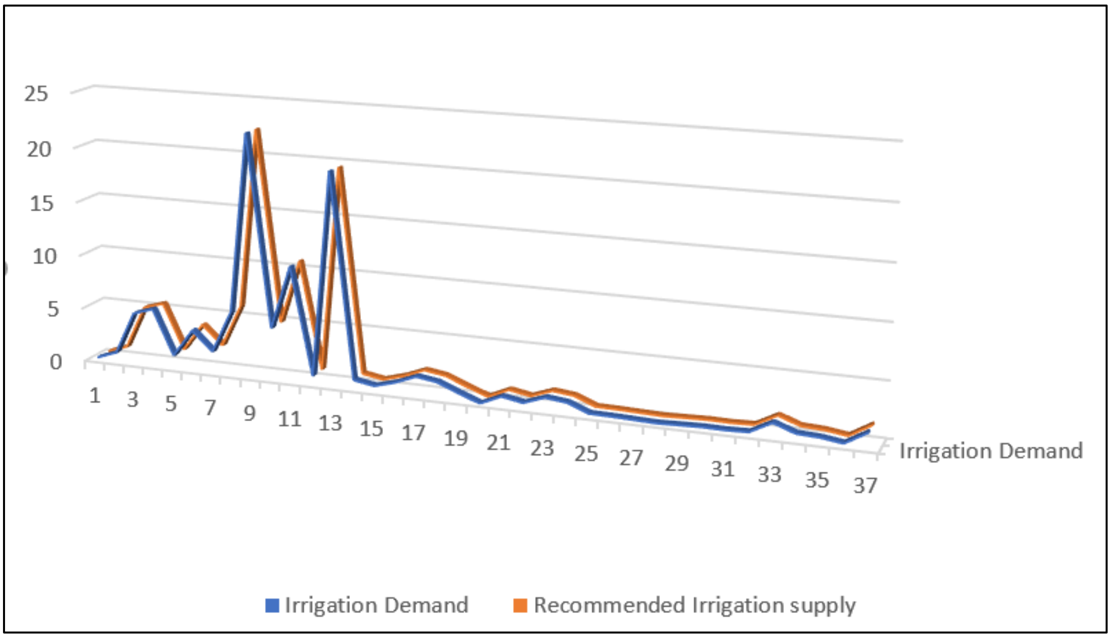


Figure 4.1 Percentage of irrigation demand and recommended supply in 2015-2016

The graph in the figure 4.1 illustrates the percentage of irrigation demand and the percentage of recommended supply of each command area over period in 2015- Yala and 2016-Maha seasons. It illustrates Optimization Model has recommended irrigation supply to parallel the irrigation demand of each command area.

4.2 Consideration of environmental factors in the Optimization Model

Making adjustments to the irrigation water distribution plan based on environmental factors affects further reducing irrigation water supply. The following Table 4.2 compares the recommended irrigation supply with different environmental factors for the same Crop Water Requirement (CWR).

(ET) Evapotranspiration

Date	CWR (mm/ha)	Rainfall (mm)	ET (mm)	Irrigation demand (CM)	Recommended supply (CM)
2015-10-11	15.62	0	2.4	661442.14	662628.85
2015-10-12	15.62	11	2.4	390918.93	393189.7
2015-10-13	15.62	21.5	2.4	132692.21	133356.64
2015-10-14	15.62	11.3	2.4	383540.98	385362.51
2015-10-15	15.62	0	2.4	383540.98	385362.51
2015-10-16	15.62	5	2.4	538477	541515.73
2015-10-17	15.62	0	2.4	661442.14	667240.15
2015-10-18	15.62	9.2	2.4	435186.32	437063.12
2015-10-19	15.62	0	2.4	661442.14	664290.04
2015-10-20	15.62	5.4	2.4	528639.8	530887.16

Table 4.2: Comparison of recommended irrigation supply in the same CWR

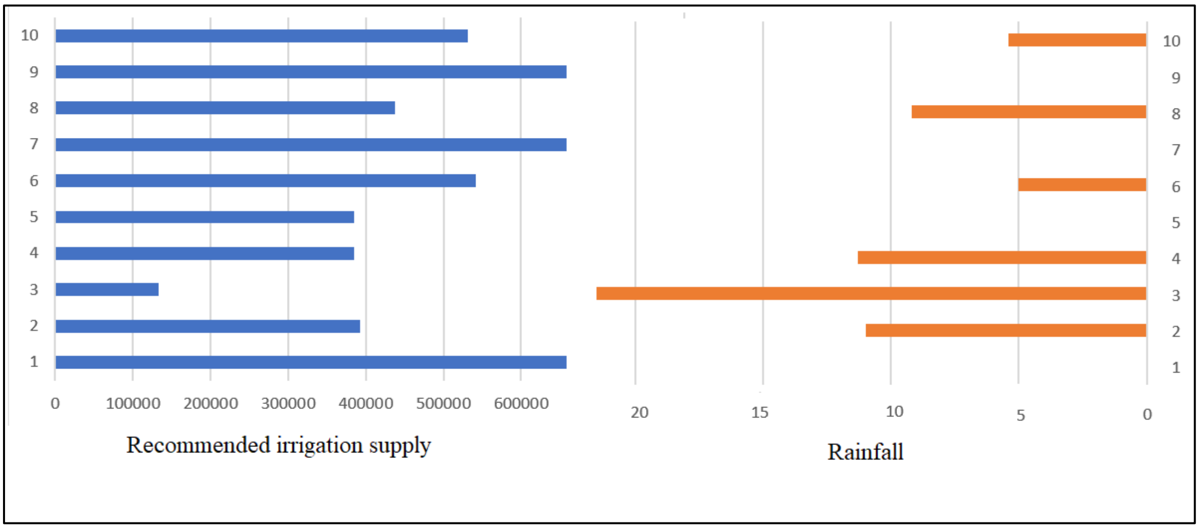


Figure 4.2: Recommended irrigation supply and rainfall (2015/10/11 to 2015/10/20)

In the above figure 4.2 illustrates that the recommended irrigation supply has changed inverse to the rainfall. 2015-10-13 has the maximum rainfall (21.5mm) in the selected date range. According to the above results, it shows minimum irrigation supply (133356.64 CM) has been recommended for the date 2015-10-13.

4.3 The difference between irrigation demand and recommended irrigation supply

The major objective of the study was to develop an optimization model to achieve maximum efficiency in irrigation water distribution through minimize the difference between the irrigation demand and supply. To illustrate the gap between demand and recommended supply, the percentage difference was computed using the following equation.

$$Difference = \frac{[RIS - IWD\,]}{IWD}*100$$

RIS - Recommended irrigation supply

IWD- Irrigation water demand

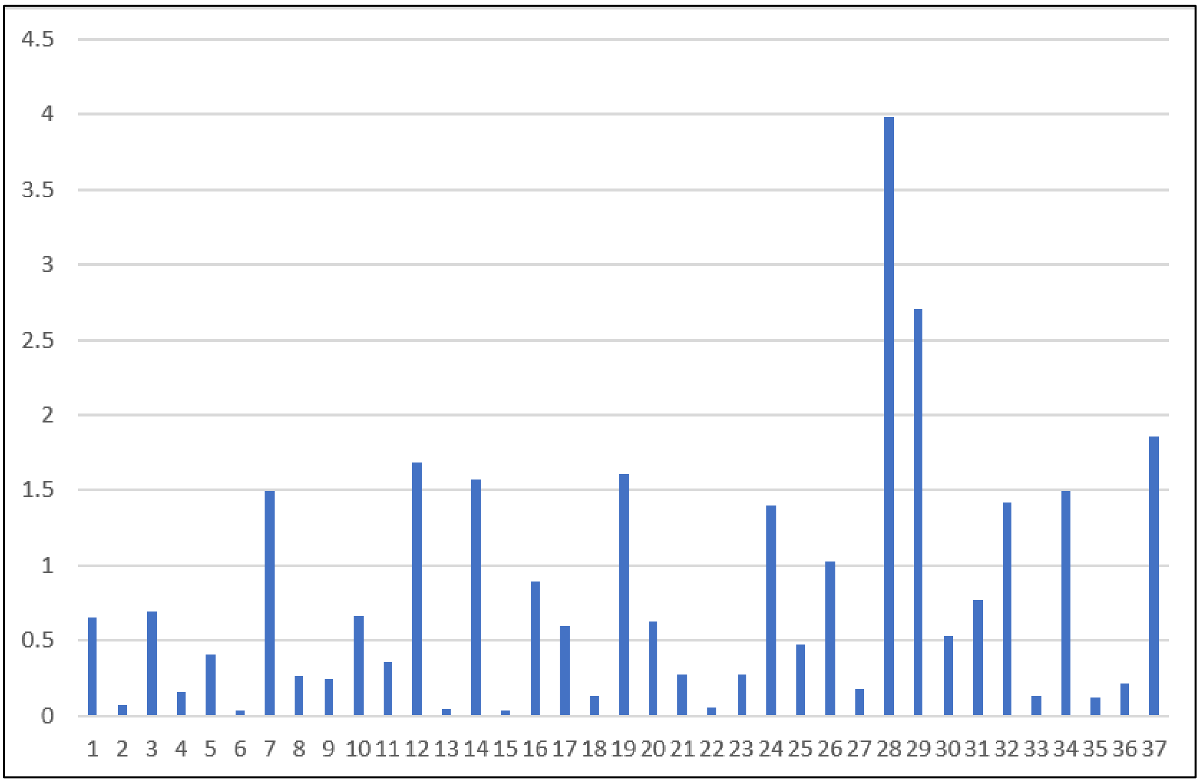


Figure 4.3: Difference between irrigation demand and recommended supply in command areas (2015- 2016Maha and 2016-Yala season)

Figure 4.3 illustrates the difference between irrigation demand and recommended irrigation supply in each command area in the Rajangana irrigation scheme. It shows 28^{th} command area has the maximum difference (3.9%) and the minimum wastage (0.03%) in 15^{th} command area.

Cubic Meter Million (CMM)

Month	Irrigation Demand (CMM)	Recommended supply (CMM)	Difference (%)
October	20.5	20.67	0.83%
November	6.5	6.55	0.77%
December	7.34	7.42	1.09%
January	7.45	7.52	0.94%
February	0.24	0.244	1.67%
March	1.17	1.18	0.85%
May	22.44	22.62	0.80%
June	8.26	8.32	0.73%
July	9.27	9.36	0.97%
August	9.45	9.56	1.16%
September	1.87	1.89	1.07%

Table 4.3: Difference between irrigation demand and recommended supply in (2015-2016Maha and 2016-Yala season)

Table 4.3 illustrates the difference between irrigation demand and supply for each month in the range of 0.73% to 1.67%. It shows the gap between the demand and the recommended irrigation supply is at a minimum value.

4.4 Comparison of the actual irrigation supply and the recommended irrigation supply

In this section compares the recommended irrigation plan with the actual amount of irrigation water issued by the Irrigation Department. In this comparison used data in 2015-2016 Maha and 2016-Yala seasons in Rajangana irrigation scheme.

Cubic Meter Million (CMM)

Month	Irrigation Demand (CMM)	ID Supply (CMM)	Recommended supply (CMM)
October	20.50	23.76	20.67
November	6.50	7.62	6.55
December	7.34	8.38	7.42
January	7.45	8.80	7.52
February	0.24	0.28	0.24
March	1.17	1.43	1.18

Table 4.4: Irrigation water issued by ID in 2015- 2016Maha season and recommended irrigation supply

Month	Irrigation Demand (CMM)	ID Supply (CMM)	Recommended supply (CMM)
May	22.44	26.56	22.62
June	8.26	10.38	8.32
July	9.27	11.18	9.36
August	9.45	11.68	9.56
September	1.87	2.13	1.89

Table 4.5: Irrigation water issued by ID in 2016-Yala season and recommended irrigation supply

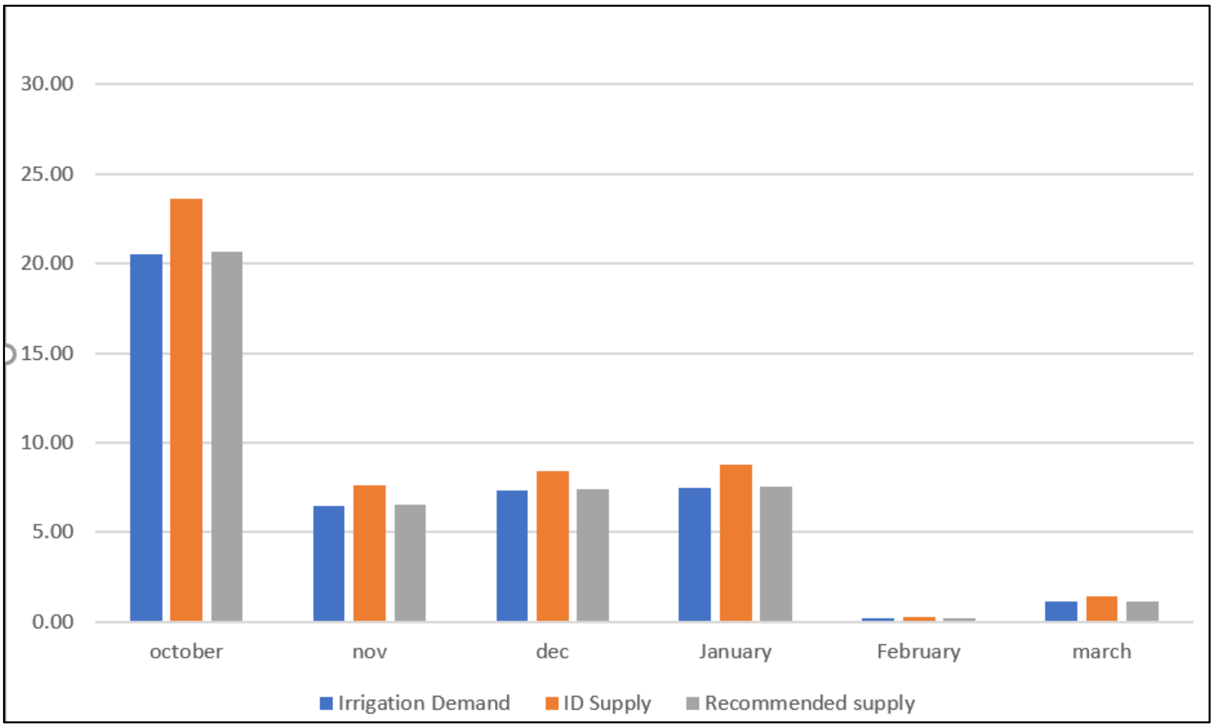


Figure 4.4: Irrigation water issued by ID in the 2015-2016 Maha season and recommended irrigation supply

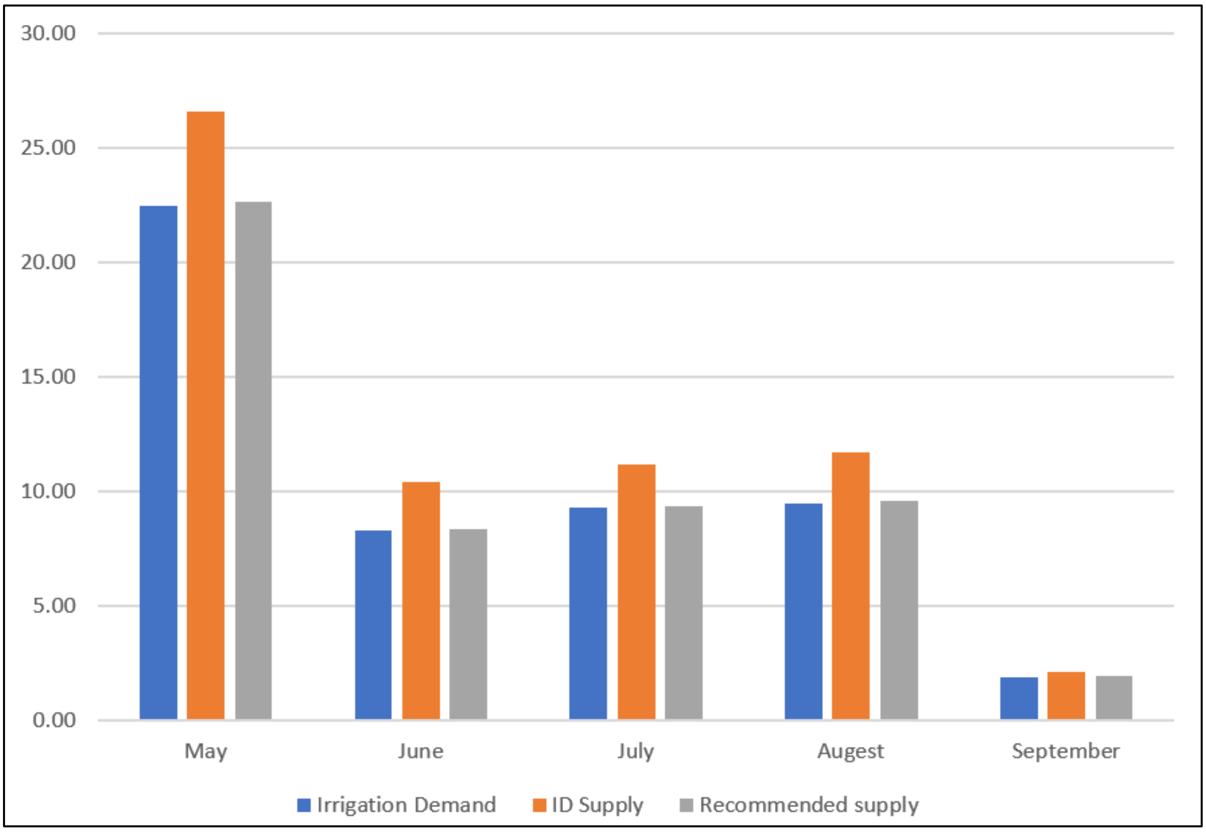


Figure 4.5: Irrigation water issued by ID in the 2016 Yala season and recommended irrigation supply

Above tables (Table 4.4, Table 4.5) and figures (Figure 4.4, Figure 4.5) illustrate that the recommended amount of irrigation supply in each month is always less than to the issued amount of the Irrigation Department.

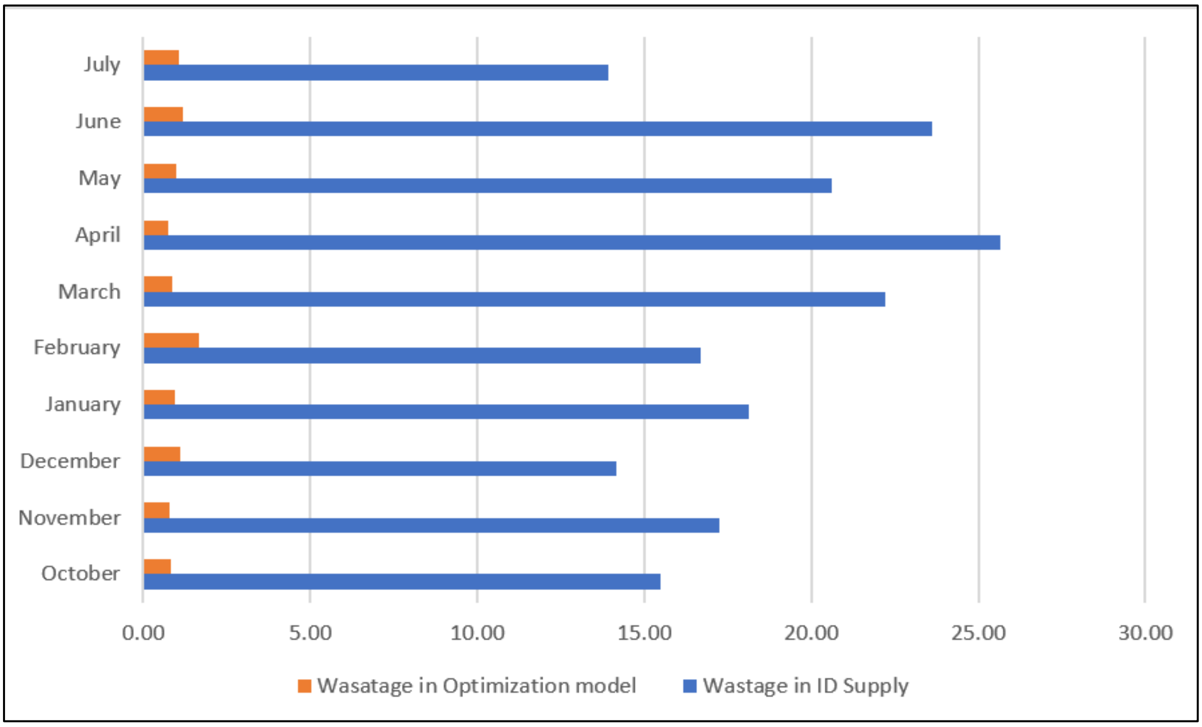


Figure 4.6: Irrigation water wastage in actual irrigation issue and wastage in Optimization model (2015- 2016Maha and 2016-Yala season)

Figure 4.6 illustrates the irrigation water wastage of the ID supply vs. Optimization Model. Maximum wastage of 25.67% has been occurred in the ID supply in April but wastage of recommended supply is at a minimum value of 0.73%. The maximum wastage of the recommended irrigation plan is 1.67% and the wastage of every month is less than to the minimum wastage (14.17%) of ID supply. The above results illustrate that the overall wastage of the Optimization model is at a minimum value compared to the actual wastage of ID distribution.

4.5 Effects of Genetic Algorithm parameters on the final result

GA parameters such as crossover rate and mutation rate affect to the final result of the recommendation model. Selection of the most appropriate values has caused the optimization model to succeed.

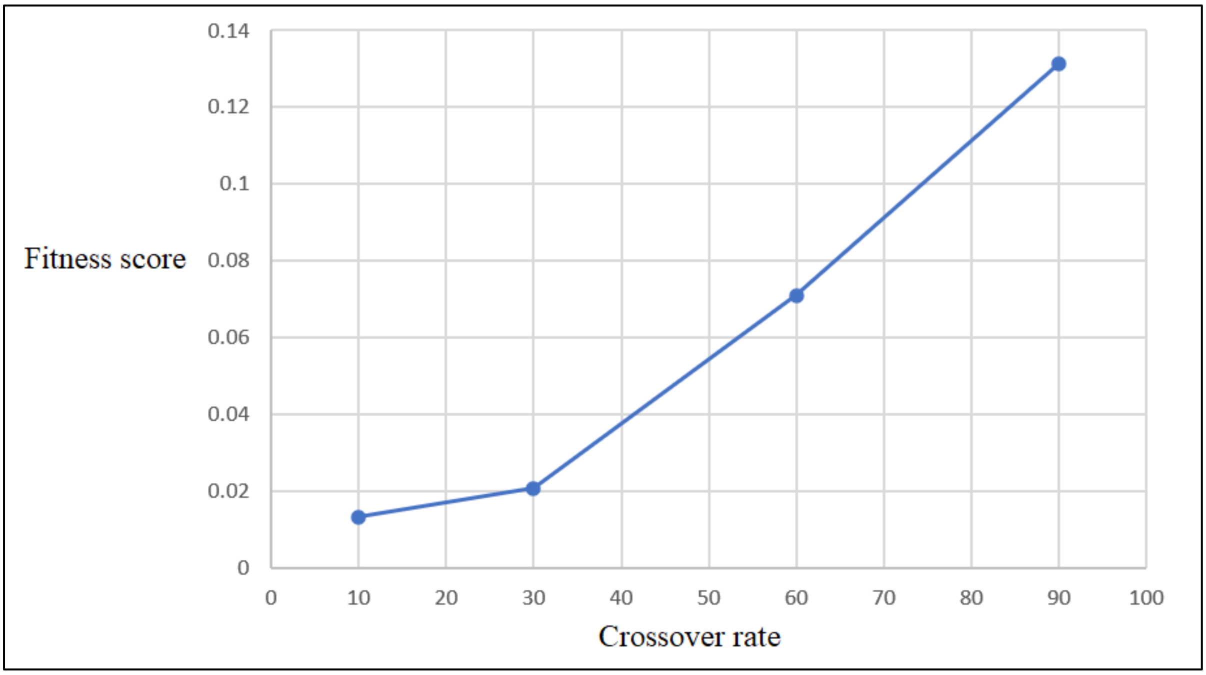


Figure 4.7 Fitness score increasing vs Crossover rate

Figure 4.7 illustrates the fitness score increasing with the crossover rate. Increasing the number of chromosomes involved in the GA operation has affected to increase the fitness score with crossover rate. Therefore, in this study used 90% as the crossover rate.

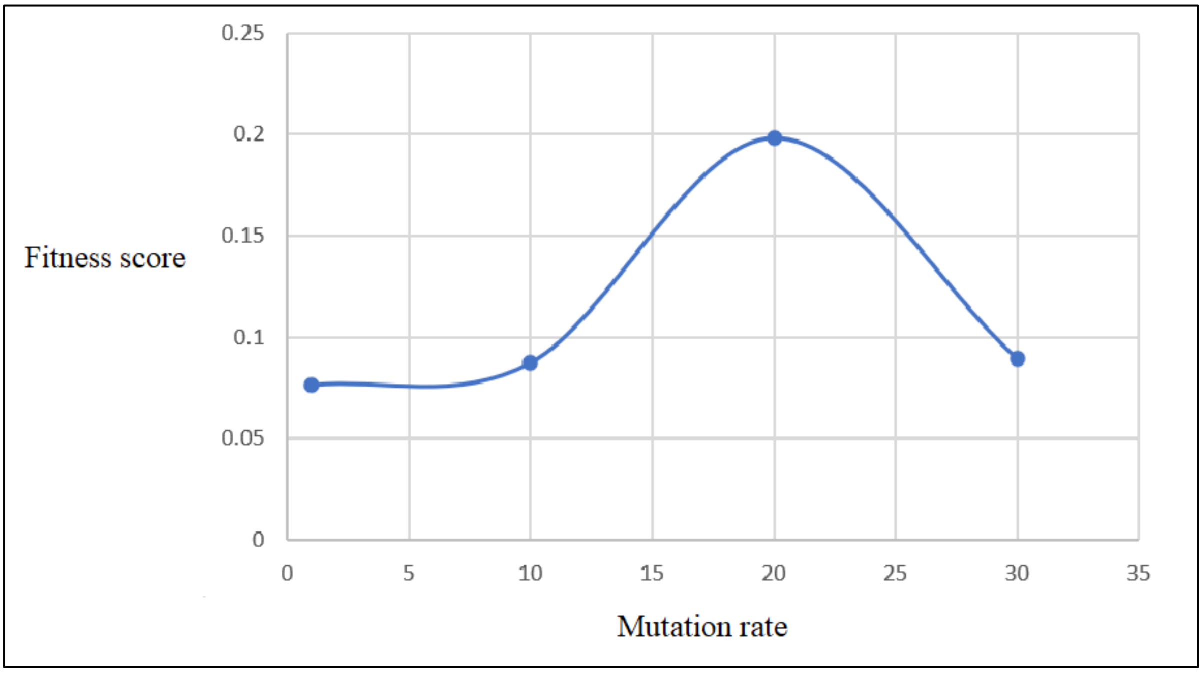


Figure 4.8 Fitness score vs Mutation rate

Figure 4.8 illustrates fitness score increasing to a maximum value and then decreasing with the mutation rate. Therefore, in this study used a mutation rate within the range of 15% to 25% as an optimal value.

CHAPTER 5

CONCLUSION AND FUTURE WORK

The Rajangana irrigation system wastes a significant amount of irrigation water each month, ranging from 13.9% to 25.6 %. This wastage of the current irrigation system revealed the need for an appropriate water distribution plan. Comparison of actual water issues at the Left Bank cannel disclosed a significant over issues of water much more than actual irrigation demand in Yala and Maha seasons. Therefore, minimizing the gap between irrigation demand and supply is required to create an equitable and efficient irrigation management system.

In the present study, a Genetic Algorithm based model has been developed to give an optimum irrigation distribution plan. The conversion of the irrigation system into the computational model of GA has facilitated the success of the GA operations. The final result of the Optimization Model has been affected by the crossover rate and mutation rate. Appropriate GA parameters identified from this study are: Number of generations = 200, Population size = 200, Crossover probability = 0.9 and Mutation Probability = 0.1.

The optimization model has recommended irrigation supply to parallel the irrigation demand of each command area with a minimum difference between the demand and supply. The maximum monthly wastage of the recommended irrigation plan is 1.67% and the minimum is 0.76%. The consideration of environmental factors affecting to the irrigation demand has helped to reduce the irrigation supply even further and minimizing the wastage. The comparison of actual water distribution of the Irrigation Department over the Recommendation Model showed that the recommended irrigation plan is more accurate and efficient.

Finally, we can conclude that the optimization model can recommend an optimum irrigation distribution plan to achieve maximum efficiency in irrigation water distribution in an equitable and accurate way.

5.1 Future works

In order to improve the efficiency of the results in the Optimization Model, domain areas can be further expanded into the entire system of the Mahawali zone. This Optimization Model considered the environmental factors such as rainfall and evapotranspiration, in order to improve the accuracy of the result following factors can be considered;

- Nature of Soil,
- Effect of climatic factors such as wind and temperature
- Effect of nature of irrigation network

In development of this solution assumed that the water amount is in satisfied level, but based on the environment factors such as drought can lead water amount to a low level. As a further enhancement of this project in order to minimize the damage to crops in a scenario like this, can consider the above factors in water distribution.

REFERENCES

Ministry of Irrigation, 2010. *Water resources Management Sri Lanka,* s.l.: Ministry of Irrigation.

Anantkumar J. Umbarkar , P. D. Sheth, 2015. CROSSOVER OPERATORS IN GENETIC ALGORITHMS: A REVIEW. *ICTACT Journal on Soft Computing.*

Bhaktikul Kampanad, 2001. *The Development of a Genetic Algorithm for Real Time Water Allocation and Water Scheduling in Complex Irrigation Systems,* Edinburgh: School of Civil and Environmental Engineering The Unive For example, we defined 6 chromosomes, including 6 genes, as below rsity of Edinburgh.

H. Chemjong and N.T.S. Wijesekera, 2017. *Evaluation of Irrigation Water Issue Practice for Better Water Management at Rajangana Reservoir, Sri Lanka.* s.l., UNESCO Madanjeet Singh Centre for South Asia Water Management.

J. B. Nixon, G. C. Dandy and A. R. Simpson, 2001. A genetic algorithm for optimizing off-farm irrigation. *Journal of Hydroinformatics |,* Volume 3, p. 11.

K Ullah, M Hafeez, 2011. *Irrigation Demand forecasting using remote sensing and meteorological data in semi-arid regions.* s.l.:International Association of Hydrological Sciences.

Khan, M. I. Z. &. H. M., 2011. Irrigation Water Requirement Prediction through Various Data Mining Techniques Applied on a Carefully Pre-processed Dataset. *Journal of Research and Practice in Information Technology*.

Krupakar H., J. A. &. D. G., 2016. *A Review of Intelligent Practices for Irrigation Prediction.* s.l., s.n.

L.S.Pereira , I.Alves, 2013. Crop Water Requirements. *Reference Module in Earth Systems and Environmental Sciences.*

Mahmood Khan,A. B. M. Saiful Islam,Mohsin Hafeez, 2011. Irrigation Water Requirement Prediction through Various Data Mining Techniques Applied on a Carefully Pre-processed Dataset. *Journal of Research and Practice in Information Technology,* 43(22), p. 17.

M, H. J. &. a. K., 2000. *Data Mining: Concepts and Techniques.* Simon Fraser University: Morgan Kaufmann.

N.T.S., C. H. &. W., 2017. *Evaluation of Irrigation Water Issue Practice for Better.* s.l., s.n.

N., T. S. &. W., 2018. *A Case Study on the Retention Tanks in the Walawe Ganga River Basin, Sri Lanka.* Moratuwa, Sri lanka, s.n.

Nixon J. B., D. G. C. &. S. A. R., 2001. A genetic algorithm for optimizing off-farm irrigation scheduling. *Journal of Hydroinformatics,* 3(1), pp. 11-22.

S., I. K. A. U. & J., M. D., 1995. Impact of management interventions on the performance of five irrigation schemes in Sri Lanka. *International Irrigation Management Institute (IIMI),* Volume xiv, p. 76.

Ullah, K. &. H. M., 2011. Irrigation demand forecasting using remote sensing and meteorological data in semi-arid regions. *IAHS-AISH Publication,* pp. 157-162.

Verma, A., 2018. *Medium.* [Online]
Available at: Evapotranspiration: Understanding and Predicting Plant Water Requirement
[Accessed 10 01 2021].

Verma, P., 2018. *Evapotranspiration: Understanding and Predicting Plant Water Requirement.* [Online]
Available at: https://medium.com/fasalapp/evapotranspiration-understanding-and-predicting-plant-water-requirement-6c1165f054c5
[Accessed 16 09 2020].

Wickramaarachchi, T., 2002. A styudy of irrigation water use for paddy cultivation in the Dry Zone of Sri Lanka. Volume 1, pp. 3-8.

APPENDICES

Appendix A – Source code for implement Genetic Algorithm

This appendix contains the source code used to implement Genetic Algorithm in the optimization Model.

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Runtime.InteropServices;
using System.Threading.Tasks;
using IrrigationSimulator.Data;
using IrrigationSimulator.Models;
using System.Linq;
using Microsoft.EntityFrameworkCore;

namespace IrrigationSimulator.Managers
{
    public class GenaticAlgorithm
    {

        //Environmental and Aggriculture related parameters
        double CropWaterRequirement;//mmha
        double rainfall; //mm
        double evapotranspiration; //mm
        //Crop Water Requirement
        double[] CWRArray;
        double[] Area;
        double[] canalEfficiancy;
        double maxdemndratio;
        int noOfNodes;

        //Genetic Algorithm related parameters
        int populationSize ;
        double cutoffFitness;
        double maxFitness ;
        double crossOverRate;
        double mutationRate ;

        List<Node> nodes = new List<Node>();
        Chromosome bestChromosome = new Chromosome();
        Population population = new Population();
        List<Population> populations = new List<Population>();

        public GenaticAlgorithm() {

            nodes = new List<Node>();
            bestChromosome = new Chromosome();
            population = new Population();
            populations = new List<Population>();

            rainfall =0; //mm
            evapotranspiration=0; //mm
                          //Crop Water Requirement
            CWRArray=new double[37];
```

```
    Area=new double[37];
    canalEfficiancy=new double[37];
    maxdemndratio=0;
    noOfNodes=0;

    //Genetic Algorithm related parameters
    populationSize=0;
    cutoffFitness=0;
    maxFitness=0;
    crossOverRate=0;
    mutationRate=0;
}

public GeneticAlgorithmResponse Process(GeneticAlgorithmRequest geneticAlgorithmRequest) {

    Intialization(geneticAlgorithmRequest);
    CreateIrrigationDemand();

    if (nodes[0].irrigationDemand > 0)
    {

        CreateFirstPopulation();
        var _result = Evaluation();

        return _result;
    }
    else {

        Chromosome chromosome = new Chromosome
        {
            chromosomeID =  1,
            populationNo = 1,
            chromosomeNo = 1,
            Genes = nodes,
            Fitness =1

        };


        GeneticAlgorithmResponse geneticAlgorithmResponse = new GeneticAlgorithmResponse
        {

            bestChromosome = chromosome,

            TotalDemad = 0,
            TotalSupply = 0,

        };

        return geneticAlgorithmResponse;
    }

}

public void Intialization(GeneticAlgorithmRequest geneticAlgorithmRequest)
{
    CropWaterRequirement = geneticAlgorithmRequest.CropWaterRequirement.Value;
    CWRArray = geneticAlgorithmRequest.CWRArray;
    Area = geneticAlgorithmRequest.Area;
    rainfall = geneticAlgorithmRequest.Rainfall;
```

```
    evapotranspiration = geneticAlgorithmRequest.Evapotranspiration;
    canalEfficiancy = geneticAlgorithmRequest.CanalEfficiancy;
    maxdemndratio = geneticAlgorithmRequest.Maxdemndratio;
    noOfNodes = geneticAlgorithmRequest.NoOfNodes;

    populationSize = geneticAlgorithmRequest.PopulationSize;
    cutoffFitness = geneticAlgorithmRequest.CutoffFitness;
    crossOverRate = geneticAlgorithmRequest.CrossOverRate;
    mutationRate = geneticAlgorithmRequest.MutationRate;
}

public void CreateIrrigationDemand() {

    //RE = 0.67 (RF)
    double effectiveRainFall = 0.67 * (rainfall);

    if (effectiveRainFall <= 0) {
        effectiveRainFall = 0;
    }

    for (int i = 0; i < noOfNodes; i++) {

        //Calculate amount of water supply by rain  in cubic meter
        double _cwr = CropWaterRequirement * Area[i] * 10;

        double _naturalwaterSupply = (((effectiveRainFall - evapotranspiration) / 1000) * Area[i] * 10000);

        double irrigationDemand = 0;

        if (_naturalwaterSupply >= _cwr)
        {

            irrigationDemand = 0;
        }
        else {

            irrigationDemand = Math.Round((((_cwr - (_naturalwaterSupply)) / canalEfficiancy[i]) * 100), 2);
        }

        Node node = new Node
        {
            nodeNo = i + 1,
            commandArea = Area[i],
            CWR = _cwr,
            irrigationDemand = irrigationDemand

        };

        nodes.Add(node);

    }


    double[] DemandArray = nodes.Select(r => r.irrigationDemand).ToArray();


}

public void CreateFirstPopulation() {
```

```
double minDemand;
double maxDemand;


// STEP 01 - create 1st Population

List<Chromosome> chromosomes = new List<Chromosome>();

// create  populationSize chromosomes
for (int i = 0; i < populationSize; i++)
{

    // create noOfNodes genes
    List<Node> genes = new List<Node>();

    for (int j = 0; j < noOfNodes; j++)
    {
        // create random number between minDemand and maxDemand
        Random random = new Random();

        minDemand = nodes[j].irrigationDemand;
        maxDemand = nodes[j].irrigationDemand * maxdemndratio;


        Node node = new Node
        {
            nodeNo = j + 1,
            commandArea = nodes[j].commandArea,
            CWR = nodes[j].CWR,
            irrigationDemand = nodes[j].irrigationDemand,
            //Get random number between mindemand and maxdemand as initial irrigation supply of node
            irrigationSupply = Math.Round((random.NextDouble() * (maxDemand - minDemand) +
minDemand), 2)


        };

        if (node.irrigationDemand==0) {
            node.irrigationSupply = 0;
        }


        genes.Add(node);
    }


    Chromosome chromosome = new Chromosome
    {
        chromosomeID = i + 1,
        populationNo = 1,
        chromosomeNo = i + 1,
        Genes = genes,
        Fitness = 0

    };

    chromosomes.Add(chromosome);

}
```

```
            population.populationNo = 1;
            population.Chromosomes = chromosomes;

        }

        public GeneticAlgorithmResponse Evaluation() {

            List<double> _fitnessValues = new List<double>();

            while (population.populationNo <= populationSize)
            {
                Population pop = new Population();
                pop.populationNo = population.populationNo;
                pop.Chromosomes = population.Chromosomes;
                pop.Fitness = population.Fitness;
                populations.Add(pop);

                //STEP 02 -Evaluation

                for (int i = 0; i < population.Chromosomes.Count(); i++)
                {
                    double Z = 0;

                    //compute the objective function value for each chromosome
                    for (int j = 0; j < population.Chromosomes[i].Genes.Count(); j++)
                    {

                        double Zi = (Math.Pow((population.Chromosomes[i].Genes[j].irrigationDemand -
population.Chromosomes[i].Genes[j].irrigationSupply), 2)) /
population.Chromosomes[i].Genes[j].irrigationDemand;

                        Z += Zi;
                    }


                    double _denominator = 1 + Z;

                    population.Chromosomes[i].Fitness = 1 / _denominator;

                }

                //Get maximum fitness
                maxFitness = population.Chromosomes.Max(r => r.Fitness);

                _fitnessValues.Add(maxFitness);
                //Get total fitness
                population.Fitness = population.Chromosomes.Sum(r => r.Fitness);


                if (maxFitness < cutoffFitness)
                {
                    //STEP 03  - Calculate Probability

                    foreach (var Chromosome in population.Chromosomes)
                    {

                        Chromosome.Probability = Chromosome.Fitness / population.Fitness;
                        Chromosome.isParent = false;
                    }
```

```
//STEP 04 - Calculate cumalative probability

for (int i = 0; i < populationSize; i++)
{

    if (i == 0)
    {
        population.Chromosomes[0].CumulativeProbability = population.Chromosomes[0].Probability;
    }
    else
    {
        population.Chromosomes[i].CumulativeProbability = population.Chromosomes[i -
1].CumulativeProbability + population.Chromosomes[i].Probability;
    }


}

//STEP 05 - generate populationSize random number R in the range 0-1

double[] RandomNumbers = new double[populationSize];

double Min = 0;
double Max = 1;

Random randNum = new Random();

for (int i = 0; i < RandomNumbers.Length; i++)
{
    RandomNumbers[i] = randNum.NextDouble() * (Max - Min) + Min;
}



//STEP 06- Create  mating pool

MatingPool matingPool = new MatingPool();
matingPool.poolNo = 1;


List<Chromosome> matingPoolchromosomes = new List<Chromosome>();

// Select chromosome for create mating pool

for (int i = 0; i < RandomNumbers.Length; i++)
{

    for (int j = 0; j < population.Chromosomes.Count; j++)
    {

        if (RandomNumbers[i] <= population.Chromosomes[j].CumulativeProbabity)
        {

            matingPoolchromosomes.Add(population.Chromosomes[j]);
            break;
        }

    }
```

```
}

matingPool.Chromosomes = matingPoolchromosomes;


// Apply Cross over rate

//generate a random number R (0-1)as the number of population.

double[] crossRateRandomNumbers = new double[populationSize];

Random crorNum = new Random();

for (int i = 0; i < crossRateRandomNumbers.Length; i++)
{
    crossRateRandomNumbers[i] = randNum.NextDouble() * (1 - 0) + 0;
    matingPool.Chromosomes[i].chromosomeNo = i ;
}



for (int i=0;i< crossRateRandomNumbers.Count(); i++) {

    if (crossRateRandomNumbers[i]< crossOverRate) {

        matingPool.Chromosomes[i].isParent = true;
    }

}


//Eligeble list for CrossOver

List<Chromosome> eligbleList = new List<Chromosome>();

eligbleList= matingPoolchromosomes.Where(r => r.isParent == true).ToList();


// STEP 06 -Crossover  (Average Concept ) - One point Random - Croossover point 18,19

int[] crossoverRandomPoints = new int[eligbleList.Count];

Random crossRandNum = new Random();

for (int i = 0; i < eligbleList.Count; i++)
{
    crossoverRandomPoints[i] = crossRandNum.Next(0, 36);
    //crossoverRandomPoints[i] = 18;
}



// Crate chromosomes for genaration 2

List<Chromosome> pop2_chromosomes = new List<Chromosome>();


if (eligbleList.Count%2==1) {
    eligbleList.Add(eligbleList.First());
}
```

```
for (int i = 0; i < eligbleList.Count/2; i++)
{


    int _crossPoint = crossoverRandomPoints[i];

    // get two parents

    int parent1Index = i*2;
    int parent2Index = parent1Index+1;

    if (parent2Index >= eligbleList.Count) {

        parent2Index = 0;
    }

    Chromosome Parent1 = new Chromosome();
    Chromosome Parent2 = new Chromosome();

    Parent1 = eligbleList[parent1Index];
    Parent2 = eligbleList[parent2Index];

    // Child 1 (crossover 0 to _crossPoint )
    Chromosome child1 = new Chromosome();

    child1 = Parent1;

    for (int j = 0; j < _crossPoint; j++)
    {

    //Given the two parents x1 and x2 where x1< x2, the blend crossover randomly selects a child in
the range[x1 - α(x2 - x1), x2 + α(x2 - x1)]

    double p1 = Parent1.Genes[j].irrigationSupply;
    double p2 = Parent2.Genes[j].irrigationSupply;

    double X1 = Math.Min(p1,p2);
    double X2 = Math.Max(p1, p2);

    double rndMin = X1 - 0.5*(X2 - X1);
    double rndMax = X2 + 0.5*(X2 - X1);

        if (rndMin < Parent1.Genes[j].irrigationDemand)
        {
            rndMin = Parent1.Genes[j].irrigationDemand;
        }

        Random crossovRandomNum = new Random();

    double crossOverValue = crossovRandomNum.NextDouble() * (rndMax - rndMin) + rndMin;

        if (crossOverValue> X1) {
            crossOverValue = X1;
        }

        child1.Genes[j].irrigationSupply = crossOverValue;


    }
```

```
int _relaceIndex = Parent1.chromosomeNo;

matingPool.Chromosomes[_relaceIndex] = child1;


// Child 2 (crossover _crossPoint-noOfNodes )
Chromosome child2 = new Chromosome();

child2 = Parent2;

for (int j = _crossPoint; j < noOfNodes; j++)
{

    //Given the two parents x1 and x2 where x1< x2, the blend crossover randomly selects a child
in the range[x1 - α(x2 - x1), x2 + α(x2 - x1)]

    double p1 = Parent1.Genes[j].irrigationSupply;
    double p2 = Parent2.Genes[j].irrigationSupply;

    double X1 = Math.Min(p1, p2);
    double X2 = Math.Max(p1, p2);

    double rndMin = X1 - 0.5 * (X2 - X1);
    double rndMax = X2 + 0.5 * (X2 - X1);

    if (rndMin < Parent1.Genes[j].irrigationDemand)
    {
        rndMin = Parent1.Genes[j].irrigationDemand;
    }

    Random crossovRandomNum = new Random();

    double crossOverValue = crossovRandomNum.NextDouble() * (rndMax - rndMin) + rndMin;

    if (crossOverValue > X1)
    {
        crossOverValue = X1;
    }

    child2.Genes[j].irrigationSupply = crossOverValue;


}

_relaceIndex = Parent2.chromosomeNo;

matingPool.Chromosomes[_relaceIndex] = child2;

}

pop2_chromosomes = matingPool.Chromosomes;

//STEP 07 - Mutation

//Random mutation generates a solution randomly within the entire parameter range
//https://engineering.purdue.edu/~sudhoff/ee630/Lecture04.pdf


int numberOfmutations = Convert.ToInt32(populationSize * noOfNodes * mutationRate);
```

```
// Generate Random numbers between 0 - numberOfmutations

int[] mutRandomPoints = new int[numberOfmutations];

Random mutRandNum = new Random();

for (int i = 0; i < mutRandomPoints.Length; i++)
{
   mutRandomPoints[i] = mutRandNum.Next(0, numberOfmutations);
}

// replace gene

for (int i = 0; i < mutRandomPoints.Count(); i++)
{

   // select chromosome for mutation
   for (int j = 1; j <= populationSize; j++)
   {

      if (mutRandomPoints[i] < noOfNodes * j)
      {


         int modulus = mutRandomPoints[i] % noOfNodes;
         int _geneIndex;

         if (modulus > 0)
         {
            _geneIndex = modulus - 1;
         }
         else
         {
            _geneIndex = 0;
         }

         // find min and max irrigationSupply supply of node i (gene) out of all chromosomes ;

         var minX = pop2_chromosomes.Min(r => r.Genes[_geneIndex].irrigationSupply);
         var maxX = pop2_chromosomes.Max(r => r.Genes[_geneIndex].irrigationSupply);

         // generate random number between minX and maxX

         var Xnew = randNum.NextDouble() * (maxX - minX) + minX;

         if (Xnew> pop2_chromosomes[j - 1].Genes[_geneIndex].irrigationSupply) {

            Xnew = pop2_chromosomes[j - 1].Genes[_geneIndex].irrigationSupply;
         }

         pop2_chromosomes[j - 1].Genes[_geneIndex].irrigationSupply = Xnew;

         break;
      }
   }


}
```

```
                population.populationNo += 1;
                population.Chromosomes = pop2_chromosomes;
                population.Chromosomes.ToList().ForEach(c => c.populationNo = population.populationNo);
            }
            else
            {
                break;
            }
        }

        bestChromosome = population.Chromosomes.OrderByDescending(r => r.Fitness).FirstOrDefault();

        var list=_fitnessValues;

        double _totalDemad = bestChromosome.Genes.Sum(r => r.irrigationDemand);


        double _totalSupply = bestChromosome.Genes.Sum(r => r.irrigationSupply);

        double _differnt = _totalSupply -_totalDemad;

        double _precentage = 0;
        if (_totalDemad>0) {
             _precentage = _differnt / _totalDemad * 100;
        }



        GeneticAlgorithmResponse geneticAlgorithmResponse = new GeneticAlgorithmResponse
        {

            bestChromosome = bestChromosome,
            FitnessValues = _fitnessValues,
            TotalDemad = _totalDemad,
            TotalSupply = _totalSupply
        };


        return geneticAlgorithmResponse;
    }

}

public class Population {

    public int populationNo { get; set; }
    public List<Chromosome> Chromosomes { get; set; }

    public double Fitness { get; set; }
}


public class MatingPool
{
    public int poolNo { get; set; }
    public List<Chromosome> Chromosomes { get; set; }
}

}
```