Crop Scheduler Report

Kshitij Prasad, Rohan Gupta, Shoaib Khan

Mentor: Dr. P Krishna Reddy

Contents

1. Introduction

Agriculture in India has a significant history. Today, India ranks second in farm output. Agriculture and allied sectors like forestry and fisheries accounted for 16.6% of the GDP in 2009, utilizing about 50% of the total workforce in the country. Even though there are ample opportunities in the agriculture sector, farmers don't have any system which can be recommend to them in crop selection. Farmer tends to grow crops which are commonly gown in their locality. They are not aware of all the possible crops which can be grown there. There are many parameters that directly or indirectly affects crop production which are not taken into consideration for crops selection by the farmer. These parameters are water availability, climate parameters like humidity, temperature, sunshine, wind velocity, dew, frost, soil parameters like soil type, fertility and its physiochemical properties, pH of the soil, and some other parameters like labour availability, investment, risk of crop failure, fertilizers, machinery etc. Even after this the market (demand and supply) also plays an important role in crop selection. It is not necessary that the crop which has maximum profit can be easily sold in the market. Moreover, farmers have got limited resources and they do not know how to utilize them to generate maximum profit. Generally they cultivate same crop or select crop which was the most profitable last year but these crops may not properly utilize available resources at the farmers command. So, there is a need for a system which will generate possible best plan to cultivate the crop which may fetch them the maximum profit through the integrated best use of the above production factors.

The ideal method for a farmer to solve this problem is to take into consideration all these factors:

- Crop Details: The details of the attributes mentioned above that affect crop production.
- Weather Forecast: The future prediction of the weather like temperature, rainfall, humidity, solar radiation etc.
- Market Forecast: The future prediction of demand and supply of crops.
- Farmer Resource: Details of the farmer resources like no. of plots, area, district and soil type of each plot, water availability, labour availability, investment, machinery etc.

It is difficult for a farmer to get all these data and even if he gets them it is impossible for him to get with the best crop plan by taking into consideration of all the above factors.

Hence an IT solution is required to simulate the problem. It will get crop details from Agriculture Department, weather forecast from Weather Department, farmer's resource base from the farmer. Right now there is no department in India which predicts extensively the demand and supply of agricultural production of different crops hence this parameter is skipped. A model will be built which will take input from the above dataset and predict the best favourable crop plan.

Above problem is not easy as it seems, it is a quite a difficult problem from the perspective of IT. Difficulty of the problem is in its diverse nature of parameters which affects the crop production. Firstly, there are many parameter that affects crop production and still some of the parameters are yet to be identified. Secondly, among the identified parameters how each parameter affects crop production? For example, it is known that the labour availability affects crop production. If the labour availability is less for a crop than the labour required, then the chances of crop failure will increase. But there is no research work which gives a function for risk of crop failure due to labour availability like linear, exponential etc. The result of the system depends on weather forecast. Many times the weather forecast is not accurate which will lead to wrong results.

In this paper, the proposed model will calculate the risk of a crop for each parameter such as water, temperature etc. and then superimpose the risk of each parameter to get total risk for a crop. Crop plan which are having lower risk will be shown at the top of proposed solution. Briefly, approach is divided into three sub parts.

- Identification of important parameters which affect the crop production such as water, labour, temperature, how each parameter affect the crop and the domain of each parameter.
- Associating risk with each crop. This risk will come into play when temperature, water and labour availability doesn't match with the ideal conditions for each crop. As each parameter is independent of each other, their risk can be superimposed to calculate the final risk of the crop.
- Generation of crop plan in the decreasing order of risk.

The rest of the paper is organised as follows. Section 2 gives Background of the problem. Section 3 presents related work done in the domain of this problem. Section 4 presents the problem statement and issues in it. Section 5 shows the detailed work done by us. Section 6 throws light on the prototype built on the basis of approach discussed under section 5. Section 7 shows the limitation of our approach. Section 8 throws light on the future work that can be performed to improve the quality of results. Section 9 concludes the paper and Section 10 contains appendix.

2. Background

India has different agro climatic regions. Different agro-climate region may have different soil-types. According to different seasons and weather conditions, the total span of cultivation varies. There is also variation of sunlight, temperature, humidity and wind speed. All these weather conditions play a major role in deciding what crop to grow. Water availability varies from same source in different seasons. Water availability for crop cultivation depends on rainfall. Labour availability also varies according to seasons. Water availability tends to decrease as the cultivation season progresses. Initially, due to rainfall, water available in wells is high but the water level decreases as the monsoon is over. Fertilizers, manures, fungicides, weedicides, farm energy and power available to that particular region also affect the agriculture significantly.

A farmer can have fragmented farm holding. These farms can vary in soil type and having different water resources. Now several crops can be cultivated in a farm. Each crop has risk associated with them. Each crop requires a particular weather type and different crop nutrition needs. Each crop has several growth stages. Each growth stage of the crop needs certain water and labour and farm power machines. Each crop growth stage has different periods. So the requirement of crop growth resources varies from time to time as the crop grows in the fields. The cultivation depends on the right schedule of the farming operations. If a farmer starts practicing agriculture after 10 or 15 days from the correct scheduled time, he may suffer a great loss.

Farmers have got limited crop production resources and they do not know how to utilize them to generate maximum profit. Also, the farmer has not the clear idea about which crop will generate the maximum profit. Generally they cultivate same crop or select crop which was the most profitable last year but these crops may not proper utilize available resource. For example a farmer may have 10 acres of land but the resources he is having allow him to cultivate only 3 acres of land. So, there is a need for a system which will generate possible best plan to cultivate the crop which may fetch them the maximum profit.

3. Related Work

1. A Framework to Identify Temperature-Based Suitable Crop Cultivation Period Using Sliding Window[7]:

This paper discuss the change in productivity if the temperature of the region crosses either maximum or minimum thresholds of crop's convenient temperature values. Given the temperature values for a given year and region, this paper discuss, how temperature variation influences the crop productivity and proposed a framework to identify suitable crop cultivation period. The introduced notion called ‘penalty matrix’ which contains penalty values which indicates the negative effect on the crop productivity for the given temperature deviation. A penalty value is a function of crop cultivation stage and magnitude of the temperature deviation from the crop’s convenient temperature values. This helps us understand how time period suitable for the crop is changing over the years with respect to temperature. Cultivation of a crop during time period suitable for that crop with respect to temperature results in higher yield. It also helps in assessing the feasibility of introducing new crops.

Crop cultivation is done in consecutive stages and each stage consists of certain number of weeks. For example, there are three growth stages for rice crop namely vegetative, reproductive and ripening. Each stage is further divided into sub-stages. The vegetative stage refers to a period from germination to the initiation of panicle; the reproductive stage, refers to a period from panicle primordial initiation heading; and the ripening stage is a period from heading to maturity

Yield capacity of the crop is primarily determined during one or more stages of the crop growth. For example, for rice crop, reproductive stage is subdivided into pre-heading and post heading periods. Yield capacity of rice crop is determined during pre-heading period and ultimate yield which is based on amount of photosynthates.

Every stage of the crop cultivation requires different critical and optimum temperatures for favourable growth of the crop. Each cultivation stage requires several weeks to complete. For a given crop, every crop cultivation stage has following temperature variables which indicate various thresholds.

- **Critical maximum temperature (TC_max)**: TC_max is the highest threshold level for the given stage. If (T_max > TC_max), it leads to significant negative influence on the crop growth.

- **Critical minimum temperature (TC_min):** TC_min is the lowest threshold level for the given stage. If (T_min < TC_min), it leads to significant negative influence on the crop growth.

- **Optimum maximum temperature (TC_optmax):** If (T_avg > TC_optmax), the yield of the crop will also get negatively influenced. But the negative influence is less than the negative influence if (T_max > TC_max).

- **Optimum minimum temperature (TC_optmin):** If (T_avg < TC_optmin) the yield of the crop will also get negatively influenced. But the negative influence is less than the negative influence if (T_min < TC_min).

- **Optimum temperature (TC_opt):** TC_opt is the suitable temperature value for crop growth. It is to be noted that TC_optmin ≤ TC_opt ≤ TC_optmax.

About Penalty Matrix (PM): The penalty matrix contains penalty values which are decided based on the negative influence of temperature deviation on crop productivity. For a given crop C, the penalty matrix (PM) is a 3 dimensional matrix. Each element is denoted by PM [i][j][k] where 'I' denotes the crop stage , 'j' indicates the category of temperature deviation and 'k' denotes the difference of regional temperature and crop's convenient temperature values in degrees celsius. The Penalty value is in the range of 0 to 1.

Our temperature module is designed on the same principle. Calculated risk associated with each crop cultivation stage is based on difference of regional temperature and crop optimal temperature values. For every cultivation stage, it is necessary to capture the influence of the deviation of regional temperature values from crop's convenient temperature values, on crop growth. The result is then input to superposition module that calculate overall risk.

2. An Equation for Modelling the Temperature Response of Plants using only the Cardinal Temperatures [5]: This paper is regarding how the crop growth is affected by the temperature in between maximum and minimum temperature in which crop can survive. It presents some of the models such as linear model, bilinear model and beta distribution models. beta distribution model is being used in our project.

- **Linear Model:** Within a limited range of temperature it has been found that the rate of development of crop is a linear function of the temperature. In this range, the time required to develop to a certain stage is related to the temperatures above a specified base or minimum temperature.
- **Bilinear Model:** The linear model approach fails to account for the fact that temperatures greater than T_{opt} delay growth or development. Hence, a bilinear approach in which two different linear equations are used to describe the responses to sub-optimum and supra-optimum temperatures.

$$\begin{cases} r = a_1 + b_1 T & T < T_{opt} \\ r = a_2 + b_2 T & T < T_{opt} \end{cases}$$

where r: daily rate of growth at any temperature

- **Beta distribution Model:** Activity increases slowly with temperature at values just above a base at which activity is zero, it then increases linearly with temperatures in an intermediate range, and finally increases slowly as temperatures approach an optimum. At temperatures above the optimum, activity is reduced as temperatures increase, and eventually ceases when a maximum temperature is reached. This can be best depicted by the following equation

$$Activity\ Ratio = \frac{r}{R_{max}} = \left(\frac{T_{max} - T}{T_{max} - T_{opt}}\right)\left(\frac{T}{T_{opt}}\right)^{\frac{T_{opt}}{T_{max} - T_{opt}}}$$

Where r: daily rate of growth at any temperature

Beta distribution model is used to calculate the risk of crop failure associated due to temperature.

3. Decision-making in Agriculture: A farm-level Modelling Approach [6]:
The general objective of this research paper is to identify and construct a type of farm-level model that will have the ability to quantify the likely impact of change in markets. The specific objective is to construct a model of a representative farm which consists of accounting identities. The model is of a deterministic type. There is linking of farm-level model to a sector level model developed by Meyer as well as outputs from several other institutions in terms of macro-economic variable.

The impact of the general system theory on the philosophy and metho9dology of modelling and simulation will be reviewed in terms of the two types of models, namely deterministic and stochastic models as well as the two different approaches towards modelling, namely normative and positive.

To understand the basic principle of general system, the relationship between objects and factor influencing the object's environment has been identified. This system can be based on analytical procedures. The application of analytical procedures was based on the argument that by breaking down an object into its different components, and them studying each of the components in isolation, one would be able to understand the original object better. However, in order to successfully apply analytical procedures, the following two assumptions have to apply: firstly, interaction between the different components of the object has to be absent or extremely weak and secondly the relationships between the different components of the object have to be linear.

The different methods of modelling that emanated from the two different approaches as well as the advantages and disadvantages of each of these methods has helped us in understanding the difference in each model and helped us to define our model and evaluate our model with well-defined existing models.

4. Problem statement and Issues

Problem Statement

Design a system which will generate the optimal (most-profitable and less risky) crop plan in given several plots of land having different soil types, different types of crops with several varieties with different water requirement, seasonal labour requirement, non-uniform labour availability, year-wise non-uniform water availability, unpredictable weather etc.

Issues

1. There are many parameters which affect crop production like soil types, temperature, water availability, labour availability, rainfall, pesticide and fertilizers, humidity, evaporation rate etc. and market parameters like profit, investment which govern demand and supply of a crop.

These parameters are the identified ones and still many parameters are left. As you can see, the problem is very complex because of the fact that each parameter will affect crop production differently which implies a different algorithm for each parameter. Coming up with an algorithm for a parameter is not an easy task. One needs to have a sound understanding of the way it affects the crop and ability to capture the effect in quantified manner.

2. There are some market parameters like profit, investment which govern demand and supply of a crop. Since there is no research on effects of crop production due to market parameters, they are taken as input parameters.

3. Understanding how each parameter affects crop production. After deciding the parameters responsible for crop production, the effects on crop production due to variation in each of parameter were studied. To decide how shortage of each parameter will affect the crop production or more formally how each parameter shortage will contribute to the crop failure is a challenging task. For example, if the water availability is more than the water required then there will not be any shortage of water and hence no risk in crop production. Now suppose the water required is more, then how will one calculate the risk given particular amount of water shortage?

4. There are many parameters which affects crop production. To decide whose shortage will affects more and whose shortage affects less is a challenging task.

5. Deciding the model for the system which encapsulates the complexity of the problem statement. Since there are many parameters, a single algorithm for all these parameters would results in a problem like updating the whole algorithm even if understanding of one parameter changes. Hence, the system was broken into different modules, process each model individually and then combine them to get the final output.

6. No research work in this domain has been published. Therefore various meetings with mentor were needed to help in designing the system.

7. The result of algorithms have to be tested. According to the results obtained, algorithms have to be modified to get the desired result.
8. Due to lack of the ideal results, the result has to be verified from our understanding and resources available on internet.

Since this problem has never been attempted before and we had no agriculture background, it was a very challenging problem for us.

4. Proposed Approach

In this section, we will give you a brief idea about the way we attacked the problem. Then, we will talk about the framework and algorithms used for solving the problem.

Basic Idea

Since there are many parameters which affect the crop production. To simplify the problem, some of the parameters have been pruned and parameter which affects most are taken into

consideration. Each parameter was closely studied to capture i) How it affects the crop production, II) How much it affects the crop production in respect to other parameters.
We noticed that all the selected parameters are independent of each other, then why not use this opportunity to make the problem easy and solvable. At this stage, we divideed the complex problem into various module. Each module captures the effect of each parameter. Final result is achieved after the superimposition of the results of each module. Above idea is explained in detail in the Framework section.

Framework

In this section, we will talk about the framework used to solve this problem. Before divinging into the details of framework, let us talk about the parameters pruning and input data format.

Selected Parameters

There are various parameters that are responsible for crop production. It would be very difficult to solve the problem, if all the parameters are taken at once. Therefore we have taken parameters which affect the most. We decided these parameters based on the meeting with our mentor. Selected parameters are as follows:

- **Water:** The supply of freshwater is an absolute essential for all forms of agriculture, although the amount of water required varies greatly between different agricultural types and climatic regions. Shortage of water leads to drought with obvious agricultural and societal impacts. Hence it is very important to have a right assessment of water needed by crop. There are various types of sources of water for agriculture. Rainfall, wells and canals constitute the main ones.
- **Climate**: Temperature, rainfall, humidity and solar radiation, wind velocity are major factors in deciding which crop to cultivate and which are not. Climate varies as region varies. For every crop there is a feasible range of temperature in which crop can survive and there is optimal temperature in which it will have maximum yield.
- **Soil Type:** A number of soils are found in India and it is a deciding factor in choosing the crop. Various types of soils are found in India. Extremely high, semidry, dry, slight dry, slight moist, moist, wet, extremely wet. A crop which is grown in one type of soil is not necessary to give the same yield in another soil. The different types of soil are alluvial soil, black soil, red/yellow soil, laterite soil, mountain soil and desert soil. They are different in various physical, chronicle properties, fertility status and thereby production of soil.
- **Labour:** Labour availability also plays an important role in deciding which crop to choose. Sugar cane requires a large availability of labour than wheat. If there is a shortage of labour then the delay in harvesting may lead to crop failure.
- **Capital Requirement per acre:** For different crops capital requirement per acre is different. There are some farmers which have less amount of capital to invest in farming. So based on the capital farmer has, the choice of crop varies.
- **Risk:** It is the probability of crop failure. The chances of crop failure will vary depending on the crop. For example, chances of crop failure is high for cotton.
- **Profit:** It is the amount of profit that can be generated from the crop. The amount of profit varies for different crop. For example, sugarcane is highly profitable than grain crops like Maize, Wheat and Raise.

Input Data Format

Below is the detailed format in which the system user (Farmer) will provide the input. **Crop Name:** Name of the crop like Wheat, Maize etc.

- **Profit:** The profit for a crop means the expected profit for the crop that a farmer can get from the crop per acre. It will be in the range of 1 to 10 where 1 indicating the lowest value and 10 the highest.
- **Risk:** This parameter means the risk of failure of crop. It will be in range of 1 to 10 where 1 indicating the lowest value and 10 the highest.
- **Investment:** The investment of a crop mean the overall investment required for the crop per acre. The unit of this parameter is rupees/acre.
- **Start Early:** This parameter denotes the earliest start time possible for growing the crop in terms of (Week, Month) like (2, 4) indicates 2^{nd} week of April.
- **Start Late:** This parameter denotes the late start time possible for growing the crop in terms of (Week, Month) like (2, 4) indicates 2^{nd} week of April.
- **Total Duration:** The total duration is defined as the no of weeks required from sowing to harvesting.
- **Soil Type:** The various types of soil suitable for crop growth.
- **Phenophase No.:** This denotes the various stages of crop life cycle which are Vegetative Activity, Flower Budding, Flowering (Bloom) and Fruit development and majority.
- **Water:** This parameters denotes the water required per day in each Phenophase. The unit will be hours/day.
- **Labour:** This input is number of labour required in each Phenophase. It will be in the range of 1 to 10 where 1 indicating the lowest value and 10 the highest.
- **Duration:** The duration is defined as the no of weeks required for each crop phases.
- **TMin:** The minimum temperature suitable for the crop in each Phenophase expressed in °C.
- **TMax:** The maximum temperature suitable for the crop in each Phenophase expressed in °C.
- **WeekId:** The week number of the year.
- **Rainfall:** The expected rainfall during the week. It is expressed in terms of hours.
- **DistrictTMin:** The minimum expected temperature for that week expressed in °C.
- **DistrictTMax:** The maximum expected temperature for that week expressed in °C.

This input data format is decided after having multiple meeting with the mentor. At first, we didn't' take phenophase details into consideration. But after some meetings, we realized that we are skipping a lot of details and our system will output wrong answer if phenophase is not taken into consideration.Now, you have got basic idea about the selected parameters and input data format. Let's understand the framework used to solve this complex problem. Framework has been divided into three phases so that one can understand easily.

Phase I: Crop Plan Generation

This phase generates all possible crop plan such that no crops duration are overlapping. Crop plan is based on start week and duration of each crop. Using brute force, time complexity of generating crop plan will be exponential but using dynamic programing, the complexity is reduced from exponential to polynomial time complexity. The generated crop plan are input to various module.

Input: CropName, Start week and duration
Output: Each crop plan is of the form List<c1, c2, …, cn> where ci is a crop

Phase II: Module based Evaluation

This phase is dedicated to devise algorithm for each module. Input of each module is generated crop plan and output is the risk for that crop plan. These individual module risks are input to superposition phase.
Input: Crop plan in form of List<c1, c2, …,cn>
Output: Module risk from 1-10(very low to very high)

Since each parameter affects crop production differently, therefore different algorithm has been used for each module. These are described as follows:

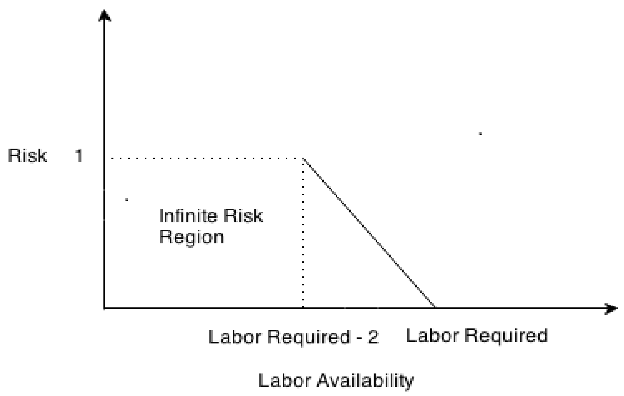
- **Temperature Module**

Beta distribution is being used to calculate risk based on temperature. Risk increases slowly with temperature at values just above the base. After that, risk increases linearly with temperatures in an intermediate range, and eventually becomes stagnant after maximum temperature.

$$Risk = 1 - \frac{r}{R_{max}} = 1 - \left(\frac{T_{max} - T}{T_{max} - T_{opt}}\right)\left(\frac{T}{T_{opt}}\right)^{\frac{T_{opt}}{T_{max} - T_{opt}}}$$

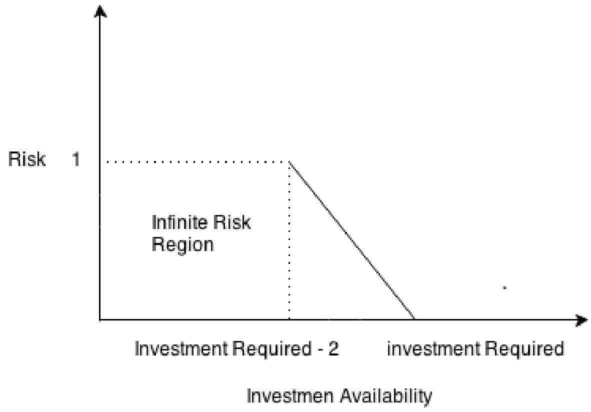
- **Labour Module**

Labour module is used to calculate risk associated with the availability of labour. If labour available is more than required labour requirement, the risk will be 0. It will increase linearly, if the available labour is slightly less than labour required. After a threshold value (as mentioned in the diagram), the risk will be infinite i.e. the crop plan will be rejected.



- **Investment Module**

Investment module is used to calculate risk associated with the capital requirement. If investment available is more than required capital requirement, the risk will be 0. It will increase linearly, if the available capital is slightly less than capital required. After a threshold value (as mentioned in the diagram), the risk will be infinite i.e. the crop plan will be rejected.



- **Water Module**

Water module is used to calculate risk associated with water availability for crop cultivation. If water available is more than the required water, then risk will be 0. The risk will increase exponentially as water required is more than water available.

$$Risk = \log_e \frac{Water\ Provided}{Water\ Required}$$

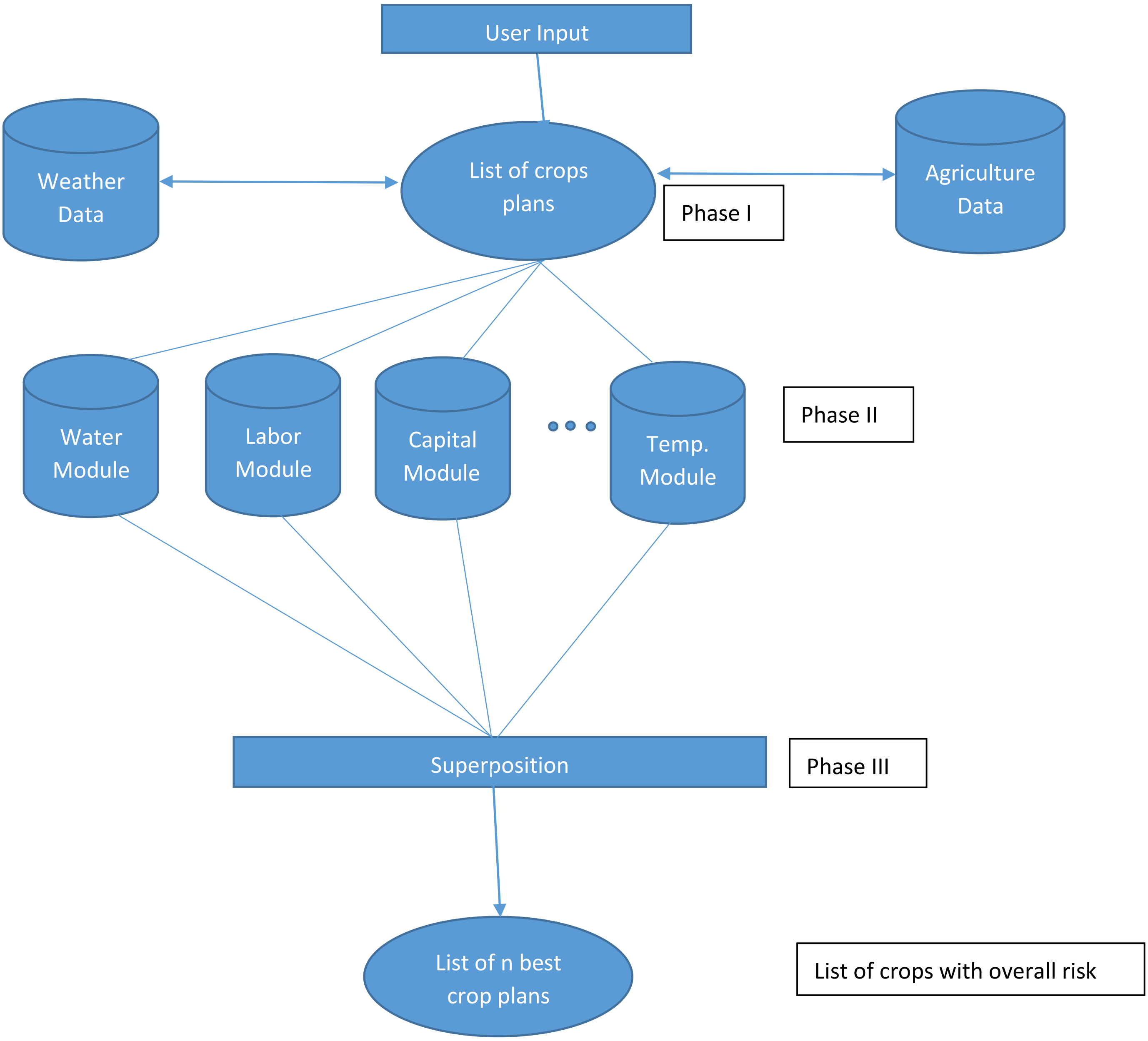
- **Soil Module**

Soil module is used to check if the land has the matching soil type as that of each of the crop of the crop plan then only the crop plan will be selected.

Phase III: Superposition

Superposition principle is applied on each parameter to calculate how each parameter individually effect crop cultivation and then combine the effect of each parameter. Overall risk is calculated from individual module's risk and then it is sorted according to overall risk and top n best crop will be output
Input: Crop Plan and risk from each module
Output: Sorted top n crop plan based on overall risk

Block Diagram representing all the phases



5. Algorithms Used

- **Crop Plan Generation Algorithm:** Generating list of crop plans based on dynamic programming

  Input: Data from agriculture department, farmer input

  Output: List of n best crop plan

$$\boxed{\begin{gathered} dp[i][j] = min(dp[j][k] + crop[l]) \ where \\ startWeek \leq j < i, 0 \leq k < no.of \ crop \ plan \ required \\ and \ 0 \leq l < no.of \ crops \end{gathered}}$$

Here dp[i][j] array denotes the top j^{th} crop plan ending at i^{th} week.

- **Temperature Algorithm:** Calculating risk due to temperature

  Input: Data from agriculture department, farmer input, crop Plan, weather data

  Output: Risk due to temperature

```
count = 0
sum = 0
for each crop plan
        for each crop in crop plan
                for each phenophase in crop
                        for each week in phenophase
                                count = count + 1
                                sum = sum + 1 − ((Tmax−T)/(Tmax−Topt)) (T/Topt)^(Topt/(Tmax−Topt))
Temperature Risk = sum / count * 10          //to make the value at the scale of 10
```

$$sum = sum + 1 - \left(\frac{T_{max}-T}{T_{max}-T_{opt}}\right)\left(\frac{T}{T_{opt}}\right)^{\frac{T_{opt}}{T_{max}-T_{opt}}}$$

- **Soil Algorithm:** Deciding whether to reject or accept crop plan

  Input: Data from agriculture department, farmer input, crop plan, weather data

  Output: No output (deciding whether to accept or reject crop plan)

```
for each crop plan
        for each crop in crop plan
                for each phenophase in crop
                        if (crop soil doesn't matches the given soil)
                                reject crop plan
```

- **Labour Algorithm:** Deciding risk due to labour availability

  Input: Data from agriculture department, farmer input, crop plan, weather data

Output: Labour risk

```
count = 0, sum = 0
for each crop plan
        for each crop in crop plan
                for each phenophase in crop
                        for each week in phenophase
                                count = count + 1
                                if (labour required – labour present < 0)
                                else if (labour required – labour present <= 2)
                                        sum = sum + (labour required – labour present)
                                else
                                        reject the crop plan
Labour Risk = sum / count
```

- **Investment Algorithm:** Calculating risk due to investment

  Input: Data from agriculture department, farmer input, crop plan, weather data

  Output: Investment risk

```
count = 0
sum = 0
for each crop plan
        for each crop in crop plan
                for each phenophase in crop
                        for each week in phenophase
                                count = count + 1
                                if (investment required – labour present < 0)
                                else if (investment required – investment present <= 2)
                                        sum = sum + (investment required – investment
                                        present)
                                else
                                        reject the crop plan
Investment Risk = sum / count
```

- **Water Algorithm:** Calculating risk due to water

Input: Data from agriculture department, farmer input, crop plan, weather data

Output: Water risk

```
count = 0
sum = 0
for each crop plan
        for each crop in crop plan
                for each phenophase in crop
                        for each week in phenophase
                                count = count + 1
                                if (investment required – labour present < 0)
                                else if (investment required – investment present <= 1)
                                        sum = sum + log_e (Water Provided / Water Required)
                                else
                                        reject the crop plan
Water Risk = sum / count
```

- **Final Risk Algorithm (Superposition Algorithm):** Calculating the overall risk for the crop plan

  Input: Risk from all the module, Weather department data

  Output: Overall risk

Overall Risk = (Base Risk + ((Temperature Risk + Water Risk + Investment Risk + Labour Risk) / 4)) /2

6. Prototype Description

A prototype was developed on the basis of framework and algorithm discussed above. Prototype development requires these things:

- Database Design
- Diverse data to populate the database
- Implementation of the above algorithms and observation of results.
  All these steps will be discussed in details

Database Design

Database design for famers input

District	No of plots	Investment (0-10)

District	Plot Number	Area (acre)	Soil Type

District	Month (1-12)	Water (hours)	Labour (0-10)

Database design for agricultural department data.

CropName	Profit (0-10)	Risk (0-10)	Investment (0-10)	StartEarly (Week, Month)	StartLate (Week, Month)	Total Duration (Week)

CropName	SoilType

CropName	PhenophaseNum	WaterRequirement (hours/day)	Labour (0-10)	Duration (weeks)	TMin (°C)	TMax (°C)

Database design for weather department data

District	WeekId	DistrictTMin (°C)	DistrictTMax (°C)	Rainfall (hours/week)

Data set collection

Three types of dataset has been generated. These are

- Data from agriculture department for each crop. Details include Crop Name, Duration, Profit/.acre, Base Risk i.e. inherent risk of the crop, Investment/acre, Start early, Start late. All these parameters are explained in detail in the input data format under section 5.3. This dataset is shown in Table1.

- Data from agriculture department, this time it includes phenophase requirements of each crop. Details include crop Name, soil type, phenophase identifier, duration of each phenophase, total water requirement in hours per day for each phenophase, number of labour per per day for each phenophase, optimal temperature for each phenophase of the crop. It is shown in table2.

- Data collected from weather department. This dataset include the weather detail of Rangareddy district for ever weak. It is shown in table 3.

Table 1

Crop name	Profit (1-10)	Risk (1-10)	Investment (1-10)	Start early (Week)	Start late (Week)	Duration (Weeks)
Cotton	4	8	7	23	35	26
Turmeric	8	4	7	23	35	39
Sugarcane	10	3	8	23	35	51
Redgram	7	5	4	23	28	23
Rice	5	7	6	23	35	21
Maize	6	7	6	23	35	22
castor	7	5	3	20	30	21
Ground nut	4	6	5	23	30	17
Chilli	5	6	8	23	44	22
Palak	6	8	7	1	52	8
Amaranthus	6	8	7	1	52	8
Black gram	6	4	5	23	52	12
Chickpea	3	6	4	23	52	13
Sunflower	4	3	3	1	20	13
Green gram	3	3	4	12	25	12

Table.2

Crop name	Phenophase	Total Water requirement (Hour/day)	Labour (number /day (1-10)	Min Temp	Max temp	Duration (Weeks)
Cotton	Emergence	3	9	22	38	1
	Vegetative phase	1	5	22	38	3
	Square formation	4	2	22	38	4
	Flowering	3	2	22	38	3
	Boll formation	3	2	22	38	2
	Boll development	3	2	22	38	5
	harvest	0	10	22	38	8
Turmeric	Germination	1	8	22	38	1
	Rhizome development	4	8	22	38	37
	Harvest	0	9	22	38	1
Sugarcane	Germination	3	7	28	30	5
	Tillering	3	7	22	30	17
	Grand growth	3	7	22	30	21
	Ripening	0	4	22	30	8

Redgram	Germination	2	7	22	30	1
	Vegetative phase	1	2	22	30	8
	Flowering	3	2	22	30	5
	Pod maturity	3	3	22	30	7
	Harvest	0	5	22	30	2
Rice	Nursery	2	3	19	38	4
	Tillering	4	8	19	38	4
	Panicle initiation	3	2	19	38	4
	Flowering	3	2	19	38	4
	Grain maturity	1	2	19	38	4
	Harvest	0	2	19	38	1
Maize	Germination	2	7	22	35	1
	Vegetative phase	2	2	22	35	7
	Flowering	2	2	22	35	4
	Grain formation	2	2	22	35	4
	Grain maturity	2	2	22	35	5
	Harvest	0	6	22	35	1
castor	Germination	1	7	22	35	1
	Vegetative phase	2	5	22	35	4
	Flowering	3	4	22	35	8
	Capsule formation	2	3	22	35	4
	Capsule maturity	2	6	22	35	4
Ground nut	Germination	1	6	22	35	1
	Vegetative phase	2	7	22	35	4
	Flowering	3	3	22	35	3
	Peg formation	3	3	22	35	4
	Pod development	3	3	22	35	4
	Harvest	0	4	22	35	1
Chilli	Nursery	1	6	22	35	4

	Vegetative phase	2	5	22	35	4
	Flowering	3	3	22	35	2
	Fruit formation	3	3	22	35	4
	Fruit maturity	3	3	22	35	4
	Harvest	0	9	22	35	4
Palak	Germination	1	2	22	35	1
	Vegetative phase	3	2	22	35	3
	Harvest	0	7	22	35	4
				22	35	
Amaranthus	Germination	2	2	22	35	1
	Vegetative phase	4	2	22	35	3
	Harvest	0	7	22	35	4
Black gram	Germination	1	4	22	30	1
	Vegetative phase	2	4	22	30	3
	Flowering	3	3	22	30	3
	Pod maturity	3	3	22	30	4
	Harvest	0	5	22	30	1
Chickpea	Germination	1	7	22	30	1
	Vegetative phase	1	4	22	30	4
	Flowering	2	3	22	30	3
	Pod maturity	2	3	22	30	4
	Harvest	0	5	22	30	1
Sunflower	Germination	1	6	20	30	1
	Vegetative phase	2	3	20	30	3
	Flowering	3	3	20	30	4
	Grain maturity	4	3	20	30	4
	Harvest	0	7	20	30	1
Green gram	Germination	1	5	20	30	1
	Vegetative phase	3	3	20	30	3
	Flowering	3	2	20	30	3
	Pod maturity	2	3	20	30	4
	Harvest	0	5	20	30	1

Table .3
Weather (Ranga Reddy district)

Week Id	Avg T Max	Avg Tmin	Cum. RF (mm)
1	27.8	12.6	0.4
2	28.7	13.5	0.6
3	29.6	13.5	3.3
4	30.4	14.0	0.2
5	30.7	14.4	0.8
6	31.3	15.5	0.5
7	32.3	16.3	2.9
8	33.2	16.6	1.0
9	34.1	16.7	1.1
10	34.9	18.4	3.9
11	35.3	19.3	8.1
12	36.8	20.4	5.8
13	36.8	20.8	4.5
14	37.3	21.8	4.3
15	37.7	22.7	5.0
16	38.1	23.4	9.8
17	38.5	24.0	5.3
18	39.2	25.0	3.7
19	39.2	25.4	8.6
20	39.4	25.7	10.3
21	39.2	25.9	1.2
22	38.8	26.2	6.2
23	36.9	25.3	19.7
24	34.5	24.2	35.6
25	33.0	23.8	24.4
26	32.6	23.7	19.7
27	32.2	23.5	29.6
28	31.4	23.3	42.9
29	30.9	23.0	42.1
30	30.3	22.8	38.7
31	29.9	22.7	38.8
32	29.9	22.6	45.1
33	30.0	22.6	32.6
34	30.0	22.6	36.2
35	30.0	22.4	37.5
36	30.3	22.5	29.4
37	30.7	22.3	59.2
38	30.7	22.4	38.9
39	31.4	22.1	25.0
40	31.0	21.5	34.5

41	31.2	20.9	23.7
42	30.9	19.7	21.3
43	30.6	18.7	11.6
44	29.7	18.2	13.6
45	29.6	17.4	8.8
46	29.5	18.1	17.8
47	29.7	17.3	3.7
48	29.4	16.0	10.2
49	29.3	14.9	0.7
50	29.0	13.7	0.6
51	28.7	12.9	2.3
52	28.6	13.1	1.0

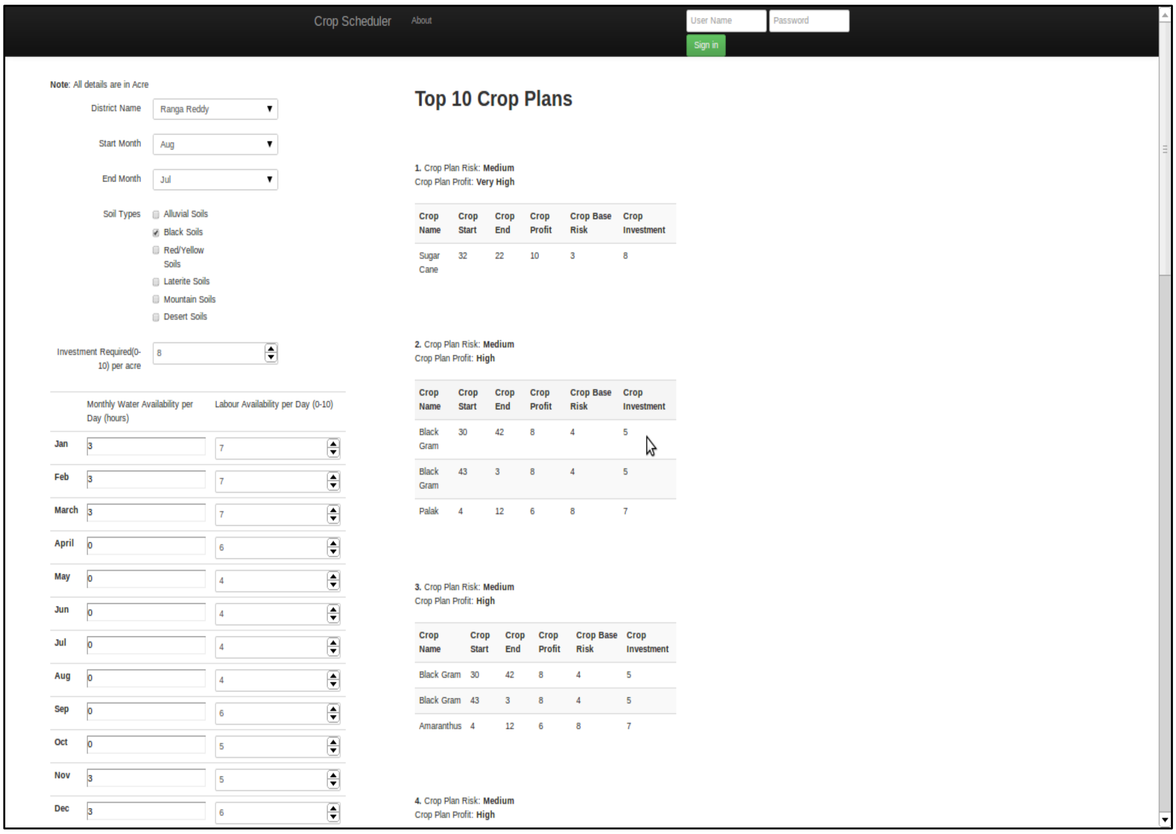
Sample Results

The system has been tested and checked with diverse inputs. The implemented system include labour, capital, water, temperature modules for calculating the risk of each crop plan. All these modules were tested against diverse input cases. Some of these tests and their results will be shown below.

All these results are generated from snapshot of the implemented system. On the left side of each figure is the input given by the farmer and on the right side is the result generated by the system.

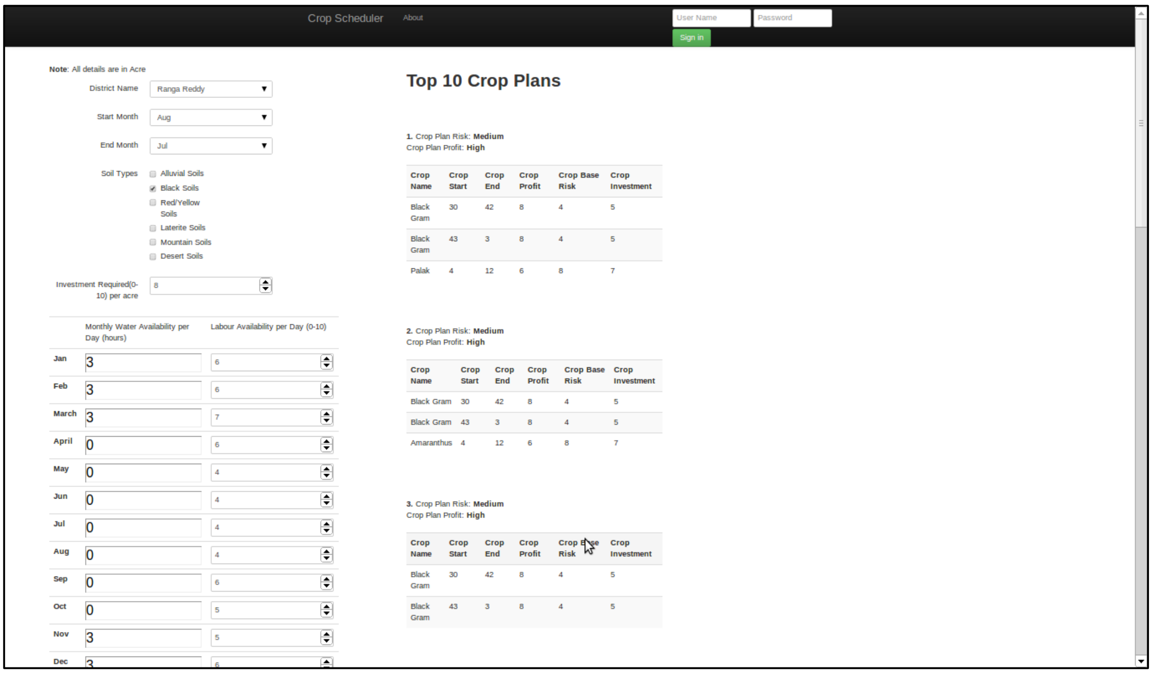
1. This test case shows the crop plan from August to July provided investment of 8(Very High), Rangareddy District, Black Soil. You might have noticed that water requirement from Jan to March is 3 and from Sep to Dec is 3 but there is no water requirement from May to Aug. Above statement can be justified by rainfall during May to Aug which is sufficient for crops like sugarcane, Black Gram and Amaranths.

   Sugarcane is on the top of list because of being most profitable and less risky.

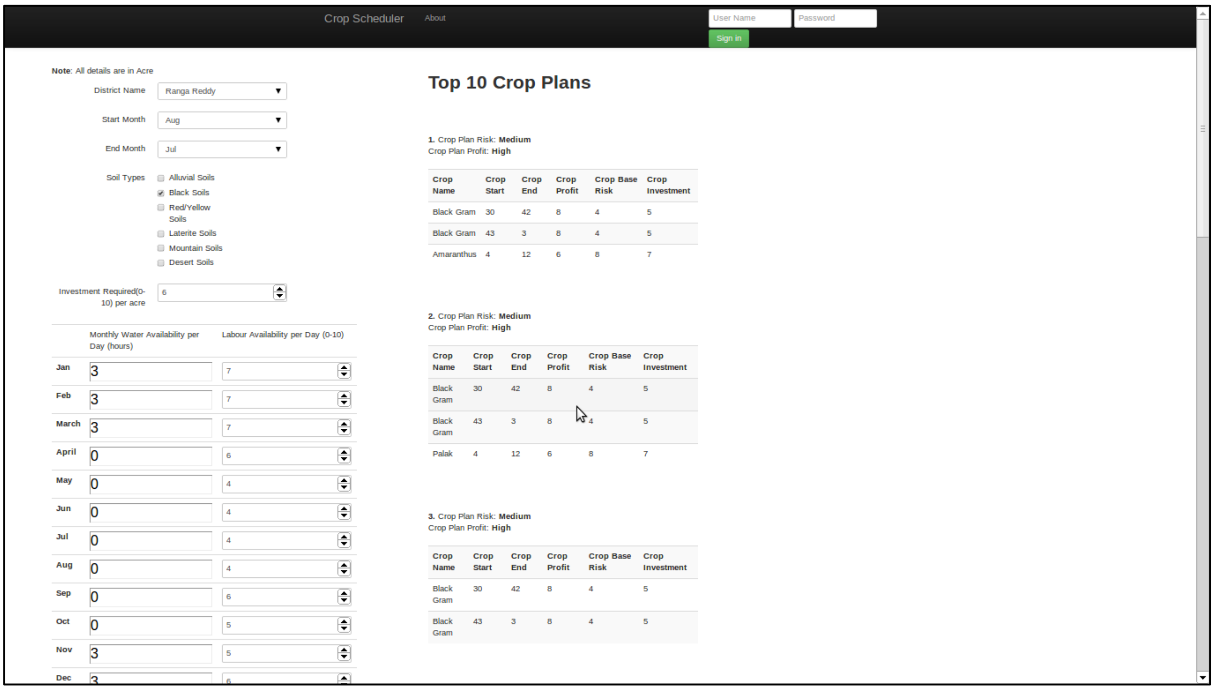


*Left side of the above picture contains the input given by the farmer and right side contains the result generated by the system.

2. This test case shows the crop plan from August to July. In this case, all the parameters are kept same as 1st except labour availability. Labour availability has been decreased for some months which resulted in the downfall of sugarcane preference in the top 10 list.



3. This test case shows the downfall of sugarcane because of reduction in the investment. Sugarcane requires high investment of 8 but farmer has provided only 6 which imply high risk on sugarcane.



7. Limitations and discussion

Limitations

- Our model generates the crop based on the weather details of the previous year data. But since the weather is not static every time hence our output may vary than the actual output, if there is large difference between the current weather and the previous year weather data.

- There are still many opportunities left which can be used to produce better results. For example: more water consuming crops can be grown on the plots which are closer to tube well or canal.

- Since we haven't considered all the parameters affecting crop production, our system many not produce optimal results for some cases. This step was necessary to reduce the problem complexity and make it solvable.

Discussion

The initial phase of project was dedicated to understand the problem statement and realizing the complexity associated with agriculture production. The result of first phase led to conclusion that problem has too many parameter which affect the production and they all can't be modelled as it is very difficult to quantify and develop their relationship with final result. Therefore, number of parameters has been reduced which reduced the complexity. Also, these parameters are assumed to be independent of each other. The benefits of reduced independent parameters are reduction in complexity and well defined parameters which are easy to quantify and with known effect on production. The input parameters are divided into groups like farmer input, agricultural department input and weather department input. Once the parameter are decided, the set of parameters that affect crop production are selected and studied how each parameter affects crop production and the domain of each parameter. Now with clear understanding of how each parameters affects crop production, the effect of the parameter can be captured into an algorithm. Capturing the effect of all the parameters into one algorithm is difficult and if in future there is better understanding of how a parameter affects crop production then it will be difficult to update the algorithm. So, a different approach is needed to solve the problem. Based on assumption that most of the parameters are independent of each other i.e. how any parameter affects crop production is independent how any other parameter affects crop production. The complex problem is divided into multiple different sub problems. Here superposition comes into play. Since each of the parameters that affects crop production are independent, therefore separate module are created for each of the parameter. The objective of system is reduced to generate a list of n crop plans with the associated profit and risk. It will help the farmers to get an idea of different crop plans that he can follow based on the profit and risk. In each module, an algorithm is used to evaluate the risk associated with the parameter. After getting the risks from all the parameters, risks are combined to get the overall risk associated with the crop plan. From this set, best n crop plans are selected.

8. Future Work

- Since each module is implemented independent of other, their code can be easily modified in latter stage if one comes up with the better understanding of a particular module. This way one need not to change the whole code. This is the benefit of using the superposition model.
- The system is based on basic parameters for generating crop plan. More parameters can be added later by writing a separate module for each of the new parameter and updating the superposition algorithm to take these parameters into effect.

- An approach to the problem based on machine learning may help in getting better results.
- Future weather details can be taken into account to generate better crop plan

9. Conclusions

Superposition provides an efficient and convenient method to model the system. For the scope of this project, this complex problem has been reduced by dividing system to smaller module. Hence, the complexity of system is broken into independent module which can be tackled easily. The results are good and in line with expected outcome. There is further scope of improvement in terms of improving algorithm and including more parameters and defining relationship between parameters which will be more holistic approach.

10. References

1. YASH P. ABROL, Division of Plant Physiology, Indian Agricultural Research Institute, New Delhi, India, KEITH T. INGRAM, University of Georgia, College of Agricultural and Environmental Sciences, Georgia Agricultural Experiment Station, Griffin, Georgia, USA, Effects of higher day and night temperatures on growth and yields of some crop plants, http://www.fao.org/docrep/w5183e/w5183e08.htm
2. Natural Resources Management and Environment Department, Irrigation Water Management: Irrigation water needs, http://www.fao.org/docrep/s2022e/s2022e07.htm

3. Department of Agriculture and Cooperation, Ministry of Agriculture, Government of India , http://eands.dacnet.nic.in/Cost_of_Cultivation.htm
4. Directorate of Marketing & Inspection (DMI), Ministry of Agriculture, Government of India, http://agmarknet.nic.in/
5. WEIKAI YAN and L. A. HUNT, Department of Plant Agriculture, Crop Science Division, University of Guelph, Guelph, Ontario, Canada, 9 July, 1999, An Equation for Modelling the Temperature Response of Plants using only the Cardinal Temperatures
6. P G Strauss, Department of Agriculture Economics, Extension and Rural Development, Faculty of Natural and Agricultural Sciences, University of Pretoria, May 2005, Decision-making in Agriculture: A Farm-level Modelling Approach
7. S. Bhargav, P. Krishna Reddy and B.Bhaskar Reddy, Center for Data Engineering, International Institute of Information Technology (IIIT-H), Hyderabad, India, 2010, A framework to identify temperature-based suitable crop cultivation period using sliding window approach
8. A decision support system for farm planning using AgriSupport II B. Recio, F. Rubio, J. and A. Criado
9. B. Recio, F. Rubio, J.A. Criado, Departamento de Matema´tica Aplicada a la Ingenierı´a Agrono´mica, Escuela Te´cnica Superior de Ingenieros Agro´nomos, Universidad Polite´cnica de Madrid, Avenida de la Complutense s/n, Madrid, Spain, 3 July 2002, A decision support system for farm planning using AgriSupport II

11. Appendixes

Language and framework used

We used JavaScript as a language and AngularJs as a framework. We also used other libraries such as twitter bootstrap to make it more visually appealing. Since the AngularJs is being used, increase in readability and decrease in code size was obvious.

Code for different algorithms

Since the complete code is about 2k lines. I am only going to share the code for above discussed algorithms.

1) Code for crop plan generation

```
for (var i = startWeek; i !== endWeek; i = (i + 1) % 52)
        {
            var sortArr = new Array();
            var count = 0;
            for (var j = startWeek; j !== i; j = (j + 1) % 52)
            {
                for (var k = 0; k < $scope.cropPlanCount; k++)
                {
                    for (var l = 0; l < $scope.result.length; l++)
                    {
                        for (var beg = parseInt($scope.result[l].startWeek) - 1;
                                beg !== parseInt($scope.result[l].startEndWeek) - 1;
                                beg = (beg + 1) % 52)
                        {
                            var end = (beg + parseInt($scope.result[l].duration)) % 52;

                            if ($scope.cropPlanDP[j][k].plan.length === 0)
                            {
                                if ((startWeek < i && beg < end &&
                                        beg >= startWeek &&
                                        end === i) ||
                                        (startWeek > i && (
                                                (beg < end && (end === i || beg >= startWeek))
                                                || (beg > end && beg >= startWeek && end === i)
                                                )))
                                {
                                    if (count < $scope.cropPlanCount)
                                    {
                                        var plan = [{"start": beg, "end": end, "index": l}];
                                        var risk = calculateRisk($scope, plan);
                                        for (var m = 0; m < count; m++)
                                        {
                                            if (arraysIdentical(sortArr[m].plan,
                                                    [{"start": beg, "end": end, "index": l}]))
                                                break;
                                            else if (m === count - 1)
                                            {
                                                count++;
                                                sortArr.push({"risk": risk,
                                                    "plan": plan});
                                                break;
                                            }
                                        }
                                        if (count === 0)
                                        {
                                            count++;
                                            sortArr.push({"risk": risk,
                                                "plan": plan});
                                        }
                                        sortArr.sort(compareFunc);
                                    }
                                    else if (risk < sortArr[count - 1].risk)
                                    {
                                        for (var m = 0; m < count; m++)
                                        {
                                            if (arraysIdentical(sortArr[m].plan, plan))
                                                break;
                                            else if (m === count - 1) {
                                                sortArr.pop();
                                                sortArr.push({"risk": risk,
                                                    "plan": plan});
                                            }
                                        }
                                        sortArr.sort(compareFunc);
                                    }
                                }
                            }
```

```
                        else
                        {
                            if ((j < i && beg < end &&
                                    beg > j && end === i) ||
                                    (j > i && (
                                            (beg < end && (end === i || beg > j))
                                            || (beg > end && beg > j && end === i)
                                            )))
                            {
                                //var risk = $scope.cropPlanDP[j][k].risk +
parseInt($scope.result[l].risk);
                                var plan = new Array();

                                for (var tmp = 0; tmp < $scope.cropPlanDP[j][k].plan.length;
tmp++)
                                    plan.push($.extend(true, {},
$scope.cropPlanDP[j][k].plan[tmp]));

                                plan.push({"start": beg, "end": end, "index": l});
                                var risk = calculateRisk($scope, plan);
                                if (count < $scope.cropPlanCount)
                                {
                                    for (var m = 0; m < count; m++)
                                    {
                                        if (arraysIdentical(sortArr[m].plan, plan))
                                            break;
                                        else if (m === count - 1)
                                        {
                                            sortArr.push({"risk": risk, "plan": plan});
                                            count++;
                                            break;
                                        }
                                    }
                                    if (count === 0)
                                    {
                                        sortArr.push({"risk": risk, "plan": plan});
                                        count++;
                                    }
                                    sortArr.sort(compareFunc);
                                }
                                else if (risk < sortArr[count - 1].risk)
                                {

                                    for (var m = 0; m < count; m++)
                                    {
                                        if (arraysIdentical(sortArr[m].plan, plan))
                                            break;
                                        else if (m === count - 1)
                                        {
                                            sortArr.pop();
                                            sortArr.push({"risk": risk, "plan": plan});
                                        }
                                    }
                                    sortArr.sort(compareFunc);
                                }
                            }
                        }
                    }
                }
            }
        }

        for (var j = 0; j < count; j++)
            $scope.cropPlanDP[i][j] = sortArr[j];
    }
```

2) Code for soil module

```
//soil check
    for (var i = 0; i < plan.length; i++)
    {
        var index = plan[i].index;
        var flag = 0;
        for (var j = 0; j < $scope.result[index].cropSoils.length; j++)
        {
            for (var k = 0; k < $scope.soilTypes.length; k++)
            {
                if (($scope.result[index].cropSoils[j] === $scope.soilTypes[k][0]) &&
$scope.soilTypes[k][1])
                {
                    flag = 1;
                    break;
                }
            }
            if (flag)
                break;
        }
        if (flag !== 1)
            return 1000;
    }
```

Here, if soil types does not match for any crop in the cropPlan, then that plan is rejected by returning 1000.

3) Code for Temperature module

```
var tempRisk = 0;
    var count = 0;
    for (var i = 0; i < plan.length; i++)
    {
        var index = plan[i].index;
        var beg = plan[i].start;
        for (var j = 0; j < $scope.result[index].phenophase.length; j++)
        {
            for (var k = beg;
                    k !== (beg + parseInt($scope.result[index].phenophase[j].duration)) % 52;
                    k = (k + 1) % 52)
            {
                count++;
                var Tmin = parseFloat($scope.result[index].phenophase[j].Tmin);
                var Tmax = parseFloat($scope.result[index].phenophase[j].Tmax);
                var Topt = (Tmin + Tmax) / 2.0;
                var T = (parseFloat($scope.districtDetails[$scope.district].Tmin[k])
                        + parseFloat($scope.districtDetails[$scope.district].Tmax[k])) / 2.0;
                var risk = 0;
                risk = 1.0 - ((Tmax - T) / (Tmax - Topt)) * Math.pow(T / Topt, Topt / (Tmax - Topt));
                if (risk < 0)
                    risk *= -1;
                if (risk > 1)
                    risk = 1;
                tempRisk = tempRisk + risk * 10;
            }
            beg = (beg + parseInt($scope.result[index].phenophase[j].duration)) % 52;
        }
        var marao = false;
    }
    tempRisk = tempRisk / count;
```

Here tempRisk variable stores the risk generated due to temperature module.

4) Code for water module

```
//Water Risk
 var waterRisk = 0;
 $scope.weeklyWater = [];
 var weekIndex = 0;
 for (var i = 0; i < $scope.months.length; i++)
 {
     var value = $scope.months[i];
     var monthlyWater = parseFloat($scope.waterAvailability[value]);
     var limit = $scope.hashMapMonthToWeek[$scope.tempMonths[$scope.hashMap[value] + 1]];
     for (; weekIndex < limit; weekIndex++)
         $scope.weeklyWater[weekIndex] = monthlyWater;
 }
 ;
 for (var i = 0; i < plan.length; i++)
 {
     var index = plan[i].index;
     var beg = plan[i].start;
     var cropWaterRisk = 0;
     for (var j = 0; j < $scope.result[index].phenophase.length; j++)
     {
         var phenoWater = 0;
         var count = 0;
         for (var k = beg;
                 k !== (beg + parseInt($scope.result[index].phenophase[j].duration)) % 52;
                 k = (k + 1) % 52)
         {
             var waterGiven = $scope.weeklyWater[k];
             var rainfall = parseFloat($scope.districtDetails[$scope.district].rain[k]);
             phenoWater += waterGiven + rainfall / 10;
             count++;
         }
         var waterRequired = parseFloat($scope.result[index].phenophase[j].water);
         phenoWater /= count;
         if (phenoWater < waterRequired)
         {
             cropWaterRisk += 1 - 1 / (waterRequired / phenoWater);
         }


         beg = (beg + parseInt($scope.result[index].phenophase[j].duration)) % 52;
     }
     waterRisk += cropWaterRisk / $scope.result[index].phenophase.length;
     var marao = false;
 }
 waterRisk /= plan.length;
```

Here, waterRisk variable denotes the risk due to waterModule.

4) Code for Investment module

```
//Investment Risk
    var investRisk = 0;
    for (var i = 0; i < plan.length; i++)
        investRisk += parseFloat($scope.result[plan[i].index].invest);
    investRisk /= plan.length;
    if (investRisk <= $scope.investmentAvailability)
        investRisk = 0;
    else if (investRisk - $scope.investmentAvailability > 2)
        return 1000;
    else
        investRisk = (investRisk - $scope.investmentAvailability) * 5;
```

Here, investRisk variable denotes the risk due to investment module.

5) Code for Labour module

```
//Labour Risk
    var labourRisk = 0;
    $scope.weeklyLabour = [];
    var weekIndex = 0;
    for (var i = 0; i < $scope.months.length; i++)
    {
        var value = $scope.months[i];
        var monthlyWater = parseFloat($scope.labourAvailability[value]);
        var limit = $scope.hashMapMonthToWeek[$scope.tempMonths[$scope.hashMap[value] + 1]];
        for (; weekIndex < limit; weekIndex++)
            $scope.weeklyLabour[weekIndex] = monthlyWater;
    }
    ;

    for (var i = 0; i < plan.length; i++)
    {
        var index = plan[i].index;
        var beg = plan[i].start;
        var cropLabourRisk = 0;
        for (var j = 0; j < $scope.result[index].phenophase.length; j++)
        {
            var phenoLabour = 0;
            var count = 0;
            for (var k = beg;
                    k !== (beg + parseInt($scope.result[index].phenophase[j].duration)) % 52;
                    k = (k + 1) % 52)
            {
                phenoLabour += $scope.weeklyLabour[k];
                count++;
            }
            var laborRequired = parseFloat($scope.result[index].phenophase[j].labor);
            phenoLabour /= count;

            if (laborRequired - phenoLabour > 2)
                return 1000;
            else if (laborRequired - phenoLabour > 0)
                cropLabourRisk += (laborRequired - phenoLabour) * 5;

            beg = (beg + parseInt($scope.result[index].phenophase[j].duration)) % 52;
        }
        labourRisk += cropLabourRisk / $scope.result[index].phenophase.length;
        var marao = false;
    }
    labourRisk /= plan.length;
```

Here, labourRisk variable denotes the risk due to labour module.