A project report
submitted
to

PROF. N.R. PATEL

AND

PROF. T.P. RAMA RAO

VIKRAM SARABHAI LIBRARY
INDIAN INSTITUTE OF MANAGEMENT
VANDRAPUR, AHMEDABAD-380025

in

fulfillment of requirements
of the
Project course
by

P. RAJENDRA KUMAR
PIYUSH MITTAL

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD
DESIGN OF AN IRRIGATION SYSTEM USING COMPUTER GRAPHICS ON THE H.P. 1080 / 2700 SYSTEM

A Project Report

Submitted to

Prof. N. R. Patil

and

Prof. T.P. Rama Rao

In fulfilment of the requirements of the "Project Course"

by

RAJENDRA KUMAR, P.

&

PIYUSH MITTAL

INDIAN INSTITUTE OF MANAGEMENT

AHMEDABAD
ACKNOWLEDGEMENTS

On the successful completion of our project, we would like to record our gratitude for Prof. NR Patel and TP Rama Rao for rendering valuable guidance at many crucial stages in the project.

Thanks are also due to Mr. Ravi Chowdhury, Computer Centre, IIMA for allowing us free access to the HP-1000/2700 System.

We also thank Mr. Devarajan, Mr. Shyama and not the least Ms. Kamini for their valuable tips in the debugging and implementation stages of the project.

- Authors
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Title</th>
<th>Page no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Abstract</td>
<td>1</td>
</tr>
<tr>
<td>2. Scope &amp; Methodology</td>
<td>4</td>
</tr>
<tr>
<td>3. Input</td>
<td>7</td>
</tr>
<tr>
<td>4. Process</td>
<td>11</td>
</tr>
<tr>
<td>5. Output</td>
<td>13</td>
</tr>
<tr>
<td>6. Scope for Future Applications</td>
<td>14</td>
</tr>
<tr>
<td>7. A User's Guide</td>
<td></td>
</tr>
<tr>
<td>8. Appendix I</td>
<td></td>
</tr>
<tr>
<td>9. Appendix II(a)</td>
<td></td>
</tr>
<tr>
<td>10. Appendix II (b)</td>
<td></td>
</tr>
<tr>
<td>11. Appendix II (c)</td>
<td></td>
</tr>
</tbody>
</table>
ABSTRACT

This project is a small part of the prestigious Narmada River Project being undertaken by the Gujarat state.

The state has to give water up to a group of fields called a 'Chak'.

The entire state is geographically divided into villages and each village is subdivided into chaks. Each Chak has a maximum area of 5-20 hectares and is made of several villages.

Our project arrived at the following 6-

(1) Given a grouping and proposed plan of giving water to Chaks, estimate the cost.

(2) If regrouping of fields were allowed across Chaks, new feed points for Chaks would come about. Now we tried to find a lower cost Scheme for supplying water.
SCOPE AND METHODOLOGY

Scope

Our project was initially started to achieve these tasks:

i) enable the managers to specify a particular path for feeding the water from the initial feed point to the feed points of each chake and to compute the costs of taking such a path.

ii) find out the optimal path i.e., find out those paths whose total length is minimum as far as length and hence cost is concerned.

iii) enable the manager to specify his choices of the field groupings into chakes and thereby provide him with the necessary flexibility to regroup chakes and thereby try for a least cost grouping of fields into chakes as well as the networth required to feed water to each of the chakes at the least cost.

The first of these i.e., the evaluative model is fully implemented in our system. The second could not be implemented because of practical difficulties which will be explained in Chapter 4. As for the third i.e., a major part i.e., the regrouping, the
the simulation of the physical regrouping is implemented while
the recalculation of the optimal path length has not been done
due to the same difficulties as earlier.

Methodology

Our methodology consisted of having discussions in detail
with Prof. N.R.Patel and Prof. T.P.Rama Rao both of whom have
experience with the actual system and hence were able to explain
the functioning of the system in detail. This greatly helped
us in the system analysis stage. However, certain features of
the real system could not be implemented directly and hence
reasonable assumptions were made at the system design stage. Another
important feature of the system design stage were our efforts to
familiarise ourselves with the escape sequences (hardware
functions) which were to prove very helpful in the later stages
of system implementation.

System implementation consisted of developing software
to

1) populate the required databases,
2) achieve the tasks specified.

The software was written using FORTRAN-77 language.
One feature of our package is the fact that for graphic displays, extensive usage is made of both Escape Sequences and the Advanced Graphics Package (AGP) from the User Programs Library of the HP-1000/2700 system. More of this will be dealt with in detail in Chapter 4.

Our system follows the following framework:

![Diagram showing inputs, process, and output]

A description of each of these follows.
3. **INPUT**

The map of one of the districts in Gujarat KAPURAI was chosen to build our model. In this one specific village was chosen. This village consisted of 5 Chaks and the first four of these had 7 fields each while the last of the Chaks had 4 fields. The graphical plot of this village is given in Exhibit (1).

The map was put on the graphics tablet (the digitizer) of the HP 2700 colour monitor. Using a program called "KBJ" each point that was required to define boundaries of each of the 32 fields was fed into the computer. KBJJ resulted in a file called KNEW and this file had the following record structure.

<table>
<thead>
<tr>
<th>NO. OF POINTS</th>
<th>INDEX</th>
<th>THE POINTS THEMSELVES</th>
</tr>
</thead>
</table>

Thus this file had 32 records with variable lengths. Now this in the same form had no meaning for our system and hence the reorganization of the database KNEW into a more usable input database database called 'FIELDS' was achieved using the program NARM.

'Fields' had the following record structure:
Incidentally K8NO produced normalized coordinates which had to be denormalized while populating the FIELDS database. Also the program PROG 1 populated another database called "CHAKS", this database held Chak level data within the following record structure:

<table>
<thead>
<tr>
<th>Chak</th>
<th>Field</th>
<th>No. of Points</th>
<th>The Points</th>
<th>Area</th>
<th>Height</th>
</tr>
</thead>
</table>

Inputs were also given from time to time by the user himself, this package being highly user interactive. Another database created normally as a result of which the general applicability of the system suffers was called "POINTS". It stored data in the following record structure:

<table>
<thead>
<tr>
<th>Chak</th>
<th>No. of Fields</th>
<th>Area</th>
<th>Field with Maximum Height</th>
</tr>
</thead>
</table>

Thus there were inputs from 4 different areas:
1) From "FIELDS"
2) From "CHARS"
3) From "POINTS"
4) From USER'S KEYBOARD.

The first two were created entirely by the KNEW file (created using the graphics tablet interface). The third was created manually.

In fact the major feature of the 'input' part of the system was the usage made of data inputted from the tablet and incorporating that in the system implementation very effectively.
4. PROCESS

Basically we had started out with three tasks to be implemented.

i) Evaluate cost of the given path.

ii) Evaluate cost on the optimal path.

iii) Regroup fields in Chaks and reevaluate optimal paths and thence suggest heuristics for optimization.

Subroutines (actually parts of main program, PROG 2, itself) were incorporated to perform each task.

But before that a brief introduction to the processing part of PROG 1 follows.

An array was opened to store 'Chaks' data. The record structures defined earlier dictated the choice of the array dimensions. Then the three databases were created.

The 'FIELDS' database was created using two do loops the outer for chak level processing and the inner for field level processing. These gave chak numbers and field numbers. The number of points and the points themselves were written using the field KNEW got as a result of KBHO. The Area was
computed for each field by splitting up the polygon into a set of triangles and then evaluating the area. The height was written using the uniform random generator between 100 & 200. This step had to be resorted to because we had not fed in contour data and determining the heights was a major problem. Thus we generated the random numbers as assumed and that the highest point in the chak would be at the left hand top most corner of the field having this height. We are bothered about highest point in each chak because the government's job is to feed in the water from the village level to the chak level. The left top most corner was chosen as the feed point for the entire village.

The other two databases were populated as follows:

1) POINTS → Manually.
2) CHAKS → Chak # and No. of Fields in the CHAK ... were given by the keyboard while CHAK area and field with maximum height were determined as in PROG 1.
As described earlier PROG2 is a well-documented, modular software package and each module is explained below:

1) Length and Cost calculation: The points are plotted on the screen at appropriate locations after plotting the village map on screen. The system asks the manager to "give the next point". It also gives the manager another chance to revise his choice (the "Are you sure" prompt). If he doesn't it asks for the next point. This is repeated till the manager terminates the dialogue by pressing '0'. Once he presses zero the length of the path chosen by him and the cost (Length x Cost) are displayed.

The second option was not implemented because as of now there are no algorithms to evaluate the optimal path in case of such a problem. However, heuristics were suggested to us but since the efficacy of such procedures was in doubt we preferred not trying to try and making errors.

Another approach that is on the cards is to treat the problem as a STINER tree and then proceed to solve it. However, it has not been attempted as part of this project.

As for the third part except the recalculation of optimal lengths all other parts have been done.
In other words, we have now implemented evaluative models but not optimisation models. One interesting observation made by Prof. N. R. Patel about this problem is that using the state-of-art technology we can solve for 6 ordinary points only while we have 72 ordinary points i.e., an increase in complexity of 266 times as compared to the existing state-of-art.

Another significant feature is the fact that we have used the AGP library extensively as well as escape sequences.
5. OUTPUT

The output of the DSS designed by us gives different kinds of outputs based on the main menu options chosen.

If the evaluative model is chosen, the computer asks the manager for the path whose length he evaluates. The outputs are in terms of -

i) the cost of the canal chosen

ii) the length of the chosen canal.

iii) A highlight of the path chosen by the manager.

At the end, the system is returned to the main menu.

In case option 3 is chosen then the system asks for field \texttt{Field}, Chak \texttt{Chak to and Chak from} and calculates from the area whether the field can be moved or not. If it can't be moved, the message is conveyed to the manager and the control is returned to the main menu. If it can be moved, the system asks him if he indeed does want to move it. If he does want it the system does the following -

i) Changes colour of the field

ii) Updates the Chak \texttt{Chak} file after updating the array.
iii) Checks if maximum heights are changed now from Chak to Chak.

iv) If it has changed, it highlights the new point and transfers control to the first option evaluation stage.

And of course as in any DSS the Quit option has been incorporated. The choice of this option leads to the end of PROG 2 and control is transferred to PROG 3.

**Warning**: Run PROG 2 on Terminal #3 (HP 2700).
6. **FUTURE APPLICATIONS**

The package we have developed is certainly general and can solve much larger problems than has been taken up. This is one area in which the package will be of a very great help.

We plan to tackle the problem of optimization ourselves using the STEINER algorithm. This should give us a very good advantage in marketing the system to potential users. As of now, the evaluation model itself on repeated application can lead us to an optimal path but we can certainly implement the same using more efficient algorithms, and it seems more elegant too.

The present system can be further developed as a knowledge based system to ultimately come up with itself.

1) **Optimal grouping of fields into Chaka,**

2) **Optimal path specifications:** The general nature of the package can with minimal changes be used to tackle larger problems with many more fields. Moreover with further clarifications regarding the constraints of grouping could lead us to quicker more elegant problem solving.