

Decision support for irrigation project planning using a genetic algorithm

Sheng-Feng Kuo^a, Gary P. Merkle^b, Chen-Wuing Liu^{c,*}

^a*Civil Engineering Department, National Ilan Institute of Technology, Ilan 260, Taiwan*

^b*Biological & Irrigation Engineering Department, Utah State University, Logan, UT 84341, USA*

^c*Agricultural Engineering Department, National Taiwan University, Taipei 107, Taiwan*

Accepted 10 January 2000

Abstract

This work presents a model based on on-farm irrigation scheduling and the simple genetic algorithm optimization (GA) method for decision support in irrigation project planning. The proposed model is applied to an irrigation project located in Delta, Utah of 394.6 ha in area, for optimizing economic profits, simulating the water demand, crop yields, and estimating the related crop area percentages with specified water supply and planted area constraints. The user-interface model generates daily weather data based on long-term monthly average and standard deviation data. The generated daily weather data are then applied to simulate the daily crop water demand and relative crop yield for seven crops within two command areas. Information on relative crop yield and water demand allows the genetic algorithm to optimize the objective function for maximizing the projected benefits. Optimal planning for the 394.6 ha irrigation project can be summarized as follows: (1) projected profit equals US\$ 114,000, (2) projected water demand equals $3.03 \times 10^6 \text{ M}^3$, (3) area percentages of crops within UCA#2 command area are 70.1, 19, and 10.9% for alfalfa, barley, and corn, respectively, and (4) area percentages of crops within UCA#4 command area are 41.5, 38.9, 14.4, and 5.2% for alfalfa, barley, corn, and wheat, respectively. Simulation results also demonstrate that the most appropriate parameters of GA for this study are as follows: (1) number of generations equals 800, (2) population sizes equal 50, (3) probability of crossover equals 0.6, and (4) probability of mutation equals 0.02. © 2000 Published by Elsevier Science B.V.

Keywords: Irrigation management; Genetic algorithm; Crop yield; Decision support; Optimization; Simulation

* Corresponding author. Tel.: +886-2362-5462; fax: +886-2363-9557.
E-mail address: cwlin@cc.ee.ntu.edu.tw (C.-W. Liu)

1. Introduction

Irrigation planners need to analyze complex climate-soil-plant relationships and apply mathematical optimization techniques to determine the optimally beneficial crop patterns and water allocations. A computer-based model to simulate the climate-soil-plant systems with a novel mathematical optimization technique could facilitate irrigation planners in reaching sound decisions prior to each crop season.

Many existing models (Hill et al., 1982; Keller, 1987; Smith, 1991; Prajamwong, 1994) have been developed to simulate on-farm irrigation water demands based on climate-soil-plant systems. The traditional optimizing irrigation planning model attempts to obtain the optimum values to satisfy the objective function and constraints. Traditional optimization models in irrigation planning have received extensive interest (Lakshminarayana and Rajagopalan, 1977; Maji and Heady, 1978; Matanga and Marino, 1979; Jesus et al., 1987; Paudyal and Gupta, 1990; Raman et al., 1992). Matanga and Marino (1979) developed an area-allocation model to maximize the gross benefit from yields of crops under consideration subject to total water supply. Later, Paudyal and Gupta (1990) applied a multilevel optimization technique to resolve the complex problem of irrigation management in a large heterogeneous basin. Jesus et al. (1987) developed a linear optimization model for planning the management of Irrigation District No. 38 in the State of Sonora, Mexico. Raman et al. (1992) developed a linear programming model (LP) to generate optimal cropping patterns from previous drought experiences. Maji and Heady (1978) developed an optimal cropping pattern and a reservoir management policy for the Mayurakshi irrigation district in India. Lakshminarayana and Rajagopalan (1977) applied a linear programming model to examine the problem of optimal cropping pattern and water release from canals and tube wells in the Bari Doab basin in India. Based on the literature review, it can be stated that traditional optimization methods have limitations in finding global optimization results and are difficult to apply to a complex irrigation-planning problem since they search from point to point for the optimization. On the other hand, the genetic algorithm method searches the entire population instead of moving from one point to the next and can, therefore, overcome the limitations of the traditional methods.

Although John Holland and his colleagues pioneered the concept of a ‘genetic algorithm’ at the University of Michigan in the early 1970s (Davis, 1991), it has recently been applied to ‘real-world’ problems. A genetic algorithm is a search procedure that uses random choice as a tool to direct a highly exploitative search through the numerical coding of a given parameter space (Goldberg, 1989). Goldberg (1989) stated that genetic algorithms (GAs) differ from conventional optimization and search procedures in the following ways: (1) GAs work with a coding of the parameter set, not the parameters themselves, (2) GAs search from a population of points, not a single point, (3) GAs use objective function information, not derivatives or other auxiliary knowledge, and (4) GAs use probabilistic transition rules, not deterministic rules.

Genetic algorithms, although not extensively applied to irrigation engineering applications, have been applied to several optimization problems (Wang, 1991; Fahmy et al., 1994; McKinny and Lin, 1994; Wentzel et al., 1994). Wentzel et al. (1994) used GAs to optimize a pipe-network pumping system at New Mexico State University. The

investigation applied a pipe-network simulation model (KYPIPE) to represent the hydraulic properties of the systems and also a genetic algorithm package (GENESIS) to direct the search for optimum and near-optimum solutions. Fahmy et al. (1994) used GAs for economic optimization of river management. According to their results, GAs more efficiently generate near-optimum solutions for large and complex water resource problems than dynamic programming techniques. Wang (1991) proposed a genetic algorithm for function optimization and applied to calibration of a conceptual rainfall-runoff model. McKinny and Lin (1994) incorporated GAs in a groundwater simulation model to solve three water management problems: (1) maximum pumping from an aquifer, (2) minimum cost in water supply development, and (3) minimum cost in aquifer remediation. Analysis results indicate that GAs can be effectively used to obtain globally (or, at least near globally) optimal solutions to these groundwater management problems. This study also demonstrates that constraints can be incorporated into the formulation and do not require derivatives with respect to decision variables as in nonlinear programming.

This study focuses mainly on developing an irrigation and planning model to simulate an on-farm irrigation system, and optimize the allocation of the irrigated area to alternative crops for maximum net benefit by using of a customized genetic algorithm.

2. On-farm irrigation scheduling

When estimating irrigation water requirements, the on-farm daily soil water balance may be calculated using Eq. (1):

$$\text{SMD}_j = \text{SMD}_{j-1} - \text{IRR}_j - \text{PE}_j + \text{ET}_j + \text{EWS}_j \quad (1)$$

where SMD_j and SMD_{j-1} denote the soil moisture depletion values at the j th and $j-1$ th day; IRR_j represents the depth of irrigation water at the j th day; PE_j denotes the effective rainfall at the j th day; ET_j is the evapotranspiration rate at the j th day; and EWS_j is the evaporation rate of wet soil surface after irrigation and or rainfall on the j th day.

Based on the generated daily weather data, Hargreaves equation (Hargreaves et al., 1985) was used to calculate the reference crop coefficient:

$$\text{ET}_0 = 0.0023R_a(T + 17.8)\sqrt{T_{\max} - T_{\min}} \quad (2)$$

where ET_0 denotes the (grass) reference crop coefficient; R_a represents the extraterrestrial radiation (equivalent mm/day); T is the mean daily air temperature ($^{\circ}\text{C}$); T_{\max} denotes the maximum daily air temperature ($^{\circ}\text{C}$); and T_{\min} represents the minimum daily air temperature ($^{\circ}\text{C}$).

The basal crop coefficient (K_{cb}) represents the effects of the crop canopy on evapotranspiration and varies with time of year. The rate of change of the basal crop coefficient with time can be approximated as a linear increase (or decrease), as expressed in the Eq. (3) (Prajamwong, 1994):

$$K_{\text{cb}}^j = K_{\text{cb}}^{j-1} + (j_k - j_{k-1}) \times \frac{K_{\text{cb}}^{k-1} - K_{\text{cb}}^{k-1}}{j_k - j_{k-1}} \quad (3)$$

and,

$$j_{k-1} \leq j \leq j_k$$

where K_{cb}^j denotes the basal crop coefficient for Day j ; K_{cb}^{k-1} represents the basal crop coefficient at the current stage; j_{k-1} is the first day of current crop stage; j_k denotes the first day of the next crop growing stage; k represents the stage of development of the crop; and j is the day of year.

The daily reference crop evapotranspiration (ET_0) is used to calculate the potential (ET_c) and actual crop evapotranspiration (ET_{ca}), as given in Eqs. (4) and (5), respectively.

$$ET_c = (K_{cb} + K_s)ET_0 \quad (4)$$

where ET_c denotes the potential crop evapotranspiration (mm/day); K_{cb} represents the basal crop coefficient; and ET_0 is the (grass) reference crop evapotranspiration (mm/day).

$$ET_{ca} = (K_{cb}K_aK_s)ET_0 \quad (5)$$

and

$$K_a = \frac{\ln(100(\theta_j - \theta_{wp}/\theta_{fc} - \theta_{wp}) + 1)}{\ln(101)} \quad (6)$$

$$K_s = (1 - K_c) \left\{ 1 - \left(\frac{t_w}{t_d} \right)^{1/2} \right\} F_w \quad (7)$$

where ET_{ca} denotes the ‘actual’ crop evapotranspiration; K_a represents the soil moisture stress coefficient; K_s is the coefficient for evaporation rate from a wet soil surface after irrigation and or rainfall; θ_j denotes the soil moisture by volume at the j th day; θ_{fc} and θ_{wp} represent soil moisture by volume at field capacity and wilting point; t_w is the time in days since wetting due to irrigation and or rainfall; and t_d denotes the time in days required for the soil surface to dry after an irrigation and or rainfall event.

For on-demand irrigation scheduling, irrigation should be performed when the soil moisture depletion (SMD) initially exceeds the allowable depletion (AD). The required amount, or application depth, in a given irrigation (IRR), and allowable depletion (AD), can be mathematically described by Eqs. (8) and (9), respectively:

$$IRR_j = \frac{SMD_j}{(E_c E_a)} \quad (8)$$

$$AD_j = (\theta_{fc} - \theta_{wp}) RZ_j MAD_{stage} \quad (9)$$

where IRR_j denotes the irrigation requirement on the j th day; SMD_j represents the soil moisture depletion on the j th day; E_c is the conveyance coefficient; E_a denotes the water application efficiency; RZ_j represents the crop’s root depth on the j th day; and MAD_{stage} is the maximum allowable soil water depletion on each stage.

For each crop type, the cumulative water requirement in a growing season is the sum of the irrigation application depths at each time during the growing season. The cumulative

water requirement for each command area is the sum of seasonal crop irrigation water requirements within the command area. Finally, the cumulative irrigation water requirement for the project is the sum of the water requirements of each command area within the project.

The amount of infiltration and runoff are calculated based on the irrigation water or effective rainfall multiplied by the percentage of deep percolation and runoff due to irrigation and rainfall. The model user enters percentage values of infiltration and runoff. The cumulative amount of infiltration is used to calculate the crop yield reduction due to waterlogging.

Two factors influence the relative crop yield: (1) water stress due to insufficient water for crop evapotranspiration; and (2) waterlogging due to infiltration, produced by over-irrigation and or precipitation. Although the percentage of relative crop yield starts at 100% at the beginning of a growing season, the value can be reduced to less than 100% if there is any water stress or waterlogging during the growing season.

The relative yield reduction due to water stress is calculated at the end of each growth stage based on the ratio of cumulative potential crop evapotranspiration ($ET_{c,stage}$), and 'actual' crop evapotranspiration ($ET_{ca,stage}$) in each stage. The relationships can be described by Eqs. (10)–(12) (Neale, 1994):

$$Y_{am,stage} = 1 - K_{y,stage} \left(1 - \frac{ET_{ca,stage}}{ET_{c,stage}} \right) \quad (10)$$

$$ET_{ca,stage} = \sum_{j=STGBEG}^{j=STGEND} ET_{ca} \quad (11)$$

$$ET_{c,stage} = \sum_{j=STGBEG}^{j=STGEND} ET_c \quad (12)$$

where $Y_{am,stage}$ denotes the relative yield reduction due to water stress at each stage; $K_{y,stage}$ represents the crop coefficient at the same stage; $ET_{ca,stage}$ is the 'actual' crop evapotranspiration at the end of the stage; $ET_{c,stage}$ denotes the potential crop evapotranspiration at the end of the stage; $j=STGBEG$ and $j=STGEND$ represent the Julian days at the beginning and end of the stage; and ET_{ca} and ET_c are daily crop potential and actual evapotranspiration, respectively.

The minimum value of $Y_{am,stage}$ at each growth stage was chosen to be representative of the relative yield reduction due to water stress over the entire season ($Y_{am,season}$) as given by Eq. (13):

$$Y_{am,season} = \text{Min}(Y_{am,1}; Y_{am,2}; \dots; Y_{am,k}) \quad (13)$$

where k denotes the number of growth stages.

The cumulative infiltration within the root zone will reduce soil aeration due to waterlogging and influence the crop yield. Based on the only consideration of total infiltration during the crop growth period, the relative yield reduction due to waterlogging is calculated at the end of the season based on the ratio of cumulative total infiltration, INF_{season} , and the maximum net depletable depth, d_n , in the root zone. These relationships

can be represented by Eqs. (14) and (15) (Prajamwong, 1994):

$$Y_{a,season} = 1 - a \left(\frac{INF_{season}}{d_n} \right) \quad (14)$$

$$d_n = MAD \times AM \times R_z \quad (15)$$

where $Y_{a,season}$ denotes the relative yield reduction due to infiltration over the entire season; a is the empirical coefficient; MAD represents the maximum allowable depletion (fraction); AM is the available soil moisture (mm/m); and R_z denotes the maximum root depth (m).

The product of relative yield reduction due to water stress over the entire season ($Y_{am,season}$) and relative yield reduction due to waterlogging over the entire season ($Y_{a,season}$) is the final value of relative crop yield at the end of the growing season.

3. Implementation of genetic algorithm

Genetic algorithms have recently been applied to functional optimization problems. Functional optimization problems can be described as ‘real-world’ problems with an objective of obtaining the minimum or maximum global values within specified constraints. For decision support in irrigation project planning, our ‘real-world’ problem attempts to obtain the optimal crop area-allocated values to maximize the benefit of an irrigation project, given various constraints (e.g. maximum and minimum planted areas by crop type, and maximum volume of water supply). The genetic algorithm module has been implemented with the on-farm irrigation scheduling module to maximize the project benefit. The computational procedure of the genetic algorithm modulus can be globally divided into the following steps: (1) receive input data, (2) user interface to enter genetic algorithms, (3) define chromosome to represent problem, (4) decode chromosome into real number, (5) constraints control, (6) fitness value from object function, and (7) three simple genetic algorithm operators: reproduction, crossover, and mutation. Fig. 1 describes the procedures of genetic algorithm. The following sections provide descriptive detail about each of the steps.

3.1. Data requirements

Four sets of data are necessary to perform the GA: (1) number of generations, (2) population sizes, (3) probability of crossover, and (4) probability of mutation. From a mathematical perspective, the number of generations can be viewed as the number of ‘iterations’ to effectively control the termination criterion in the model. With the number of chromosomes in one population, GA can parallel search a group of points, not a single point as the conventional optimization method. Therefore, population sizes determine the number of chromosomes that are to be considered simultaneously in each generation. The probabilities of crossover and mutation are used in the simple genetic algorithm to effectively control the operators of crossover and mutation and they must range from 0 to 1. Goldberg (1989) suggested that satisfactory GA performance requires the choice of high-crossover and low-mutation probabilities, as well as a moderate population size.

Input four data: number of generations, population sizes, probabilities of crossover and mutation;

Define the length of chromosome to represent the problem;

Randomize initial population;

for i=1 to number of generations

{

for j=1 to population sizes

{

decode chromosome into a real number to represent crop planted area;

calculate fitness value from objective function;

constraints control;

update the maximum benefit within this generation;

}

Simple genetic algorithm: reproduction, crossover, mutation;

}

Fig. 1. The implementation of GA to plan irrigation water management.

3.2. Representative design chromosome

A chromosome's length consists of a fixed number of binary digits. Also, the position and random number values influence the chromosome's decoded value, which is related to representation adequacy of the actual problem. To design a chromosome's length to represent an irrigation project, the cumulative number of crops within each command area are first calculated. Each crop is then assigned seven binary digits to represent its area, which can range from 1 to 100%, in all of the percentage points, of the cumulative area in each command area (seven binary digits give a value of $0-2^7-1$, or 0–127 in decimal). Finally, the length of a chromosome equals the cumulative number of crop types multiplied by seven.

For example, two command areas have been considered in the Delta, Utah irrigation project for testing the model. The first command area, UCA#2, includes three crop types and the second command area, UCA#4, includes four crop types. Therefore, seven crop types are within these two command areas of the Delta, Utah irrigation project, and the length of a chromosome should be $7 \times 7 = 49$. While considering a chromosome string of 49 binary digits, the seven crops in the two Delta, Utah command areas can be depicted in coded form as shown in Fig. 2.

3.3. Decoding a chromosome into a real number

The chromosome can be decoded into a decimal number to represent the crop area within each command area. The conventional decoding method is used in this study.

Binary	0010010	1010111	0001111	1100000	0010100	1011001	1100110
Crop	1	2	3	4	5	6	7
Command Area	<----- UCA #2 ----->			<----- UCA #4 ----->			

Fig. 2. A sample chromosome coding scheme to represent seven crops in the Delta project.

Consider a problem with k decision variables $x_i, i=1, 2, \dots, k$, defined on the intervals $x_i \in [a_i, b_i]$. Each decision variable can be decoded as a binary substring of length m_i . The decoded decimal, x_i , can be obtained from Eq. (16) (McKinny and Lin, 1994):

$$X_i = a_i + \frac{b_i - a_i}{2^{m_i-1}} \sum_{j=0}^{m_i} b_j 2^j \tag{16}$$

The following case study from the Delta, Utah project contains seven crop types in the two command areas. Therefore, this problem has seven decision variables (x_i), and k equals seven. Without considering inherent crop area constraints, the percentage area of each crop type can range from 1 to 100% of the total command area. Therefore, the interval for each decision variable can be represented $x_i \in [1, 100]$, and a_i equals 1 and b_i equals 100. In conclusion, Eq. (16) can decode the binary digits into an actual number in the range from 1 to 100. The next step is to transfer this decimal number into crop area percentage, $Area_{j,\%}$, and area, $Area_{j,ha}$, within each command area. A simple averaging technique was used, as given by Eqs. (17) and (18):

$$Area_{j,\%} = \frac{\text{Decimal Value}_j}{\sum_{j=1}^{NC} \text{Decimal Value}_j} \times 100 \tag{17}$$

$$Area_{j,ha} = \frac{Area_{j,\%}}{100} \times Area_{uca} \tag{18}$$

where j is the crop index; NC denotes the number of crops within each command area and $Area_{uca}$ represents the area of each command area.

3.4. Simple genetic algorithm

Goldberg (1989) contended that a simple genetic algorithm (SGA) includes three operators: (1) reproduction, (2) crossover and (3) mutation.

Reproduction is a process in which individual strings are copied according to their objective function values. Copying strings according to their fitness values implies that strings with a higher value have a higher probability of contributing one or more offspring in the next generation (Goldberg, 1989). The portion of each chromosome in the mating pool (P_s) ranges from 0 to 1 and it can be represented as Eq. (19):

$$P_s[i] = \frac{f[i]}{\sum_{i=1}^M f[i]} \tag{19}$$

where M denotes the number of chromosomes within one population (e.g. 50); and $f[i]$ represents the fitness value of chromosome i .

In this study, $f[i]$, representing the benefit to the irrigated project of chromosome i , and $P_s[i]$ are calculated at the end of the population sizes loop to represent the portion of each benefit within the mating pool. $P_s[i]$ is chosen and the related chromosome $f[i]$ is updated for the next generation if its value is higher than the random number value. Therefore, the chromosome with a higher $P_s[i]$ has a higher likelihood of being chosen one or more times by the next generation. From the reproduction operator, the number of chromosomes to be chosen (e.g. 50) for the next generation always equal the population sizes (e.g. 50).

Once the mating pool has been filled, the crossover operator is performed with two randomly selected chromosomes to mate and develop the novel chromosomes of the next generation. The crossover operator includes three steps: (1) randomly selected chromosome to mate, (2) randomly selected crossover break site to cross over the mated chromosomes, and (3) to perform crossover if probability of crossover (P_c) is greater than the random number.

To demonstrate the idea of operators in the of simple genetic algorithm, a simple calculation was applied :

$$f(x) = -0.25x^2 + 8x + 2$$

where x is an integer value on the range from 0 to 31.

Table 1 shows the simple genetic algorithm example. The four selected chromosomes and the randomly generated initial population is shown in the first column of Table 1. The initial population can be decoded into binary strings as in the second column of Table 1. The fitness values of each string, calculated based on the objective function, along with the sum fitness, average fitness, and maximum fitness, are shown in the third column. The portion for each string ($P_s[i]$), calculated by individual fitness divided by the sum fitness, is shown in the fourth column. The expected number of each string to appear in the mating pool is $P_s[i]$ times chromosome size (i.e. 4) and these values are shown in the fifth column. Column six shows the actual number of each string selected in the mating pool; the mating pool is shown in column seven. Columns eight and nine show the partners for each chromosome and the crossover sites for each set of parents randomly. The second population of strings is shown in column 10. The decoded x values and fitness values for these strings are shown in columns 11 and 12, respectively. By comparing the fitness values of initial (i.e. column 3) and second generation (i.e. column 12) in Table 1, it is obvious that the sum, average, and maximum string fitness have improved from 154.75 to 199, 38.69 to 50.5, and 65 to 66, respectively. The simple genetic algorithm works from the simple example.

3.5. Objective function and fitness value

In this study, the objective function includes the income from crop harvest, cost of irrigation water, and crop production cost. The objective is to maximize the irrigation project benefit from the seven crops growing in the two command areas. Within the

Table 1
A simple genetic algorithm method

(1) Initial population	(2) x value	(3) $f(x)$	(4) $P_s[i]$ $f_i/\Sigma f$	(5) Expect number	(6) Actul number	(7) Mating pool	(8) Mate random	(9) Break point	(10) Second population	(11) x value	(12) $f(x)$
10010	18	65	0.42	1.7	2	10010	3	2	10100	20	62
00100	4	30	0.19	0.76	1	10010	4	3	10000	16	66
11000	24	50	0.33	1.3	1	00100	1	2	00010	2	17
11111	31	9.75	0.06	0.23	0	11000	2	3	11010	22	57
Sum		154.75	1.00	3.99	4						199
Average		38.69	0.25	1.00	1						50.5
Maximum		65	0.42	1.70	2						66

calculation loop of chromosome size, the objective function returns a fitness value to the model and then updates the fitness value and related crop-allocated area if this value is higher than previously. At the end of the chromosome loop, the subsequent fitness value is the highest benefit within the loop. Also, the maximum fitness value is selected from the generation number loop. Therefore, the fitness value and related crop area are the optimum results at the end of the calculations. The objective function is mathematically expressed in Eq. (20).

Maximize:

$$\sum_{i=1}^N \sum_{j=1}^{NC} (P_{i,j} Y_{i,j} - SD_{i,j} - FER_{i,j} - LB_{i,j} - OC_{i,j}) A_{i,j} - WP \sum_{i=1}^N \sum_{j=1}^{NC} Q_{i,j} \quad (20)$$

where i, j is the command area and crop index; N is the number of command area within irrigated project; NC is the number of crops within each command area; $P_{i,j}$ is unit price of the j th crop in the i th command area (\$/ha); $Y_{i,j}$ is yields per hectare of j th crop in the i th command area (tonne/ha); $SD_{i,j}$ is seed cost per hectare of j th crop in the i th command area (\$/ha); $FER_{i,j}$ is fertilizer cost of the j th crop in the i th command area (\$/ha); $LB_{i,j}$ is labor cost of j th crop in the i th command area (\$/ha); $OC_{i,j}$ is operation cost of j th crop in the i th command area (\$/ha); $A_{i,j}$ is planted area of j th crop in the i th command area (ha); WP is unit price of irrigation water (\$/m³); $Q_{i,j}$ is cumulative water requirement of j th crop in the i th command area (m³).

The objective function is subject to the following constraints:

1. To consider social factors and to prevent one high-value crop from dominating the search for maximum benefit, maximum and minimum area percentages must be considered for the crops:

$$\text{MinArea}_{i,j} \leq \text{AreaPer}_{i,j} \leq \text{MaxArea}_{i,j}; \text{ for some } i, j$$

2. where $\text{MinArea}_{i,j}$, and $\text{MaxArea}_{i,j}$ (%) are the minimum and maximum percentage area values of crop j in command area i , respectively.
3. The cumulative water demand of crop j in command area i should be less than the available water supply for each command area:

$$\sum_{j=1}^{NC} \text{QDem}_{i,j} \leq \text{QSup}_i; \text{ for all } i$$

where $\text{QDem}_{i,j}$ denotes the irrigation water requirement for crop j in command area i ; and QSup_i represents the available water supply for command area i .

3.6. Model development

This work also develops a model based on on-farm irrigation scheduling and the simple genetic algorithm to support irrigation planning and management. The model consists mainly of six basic modulus: (1) main module directs the running of the model with pull-down menus ability, (2) data module to enter the required data by friendly user-interface,

(3) weather generation module to generate the daily weather data, (4) on-farm irrigation scheduling module to simulate the daily water requirement and relative crop yield, (5) genetic algorithm module to optimize the project maximum benefit, and (6) results module to present results by tables, graphs and printouts.

Six basic data types are required for the model: (1) project site and operation data, (2) command area data, (3) seasonal water supply data, (4) monthly weather data, (5) soil properties data, and (6) crop phenology and economic data. Herein, the weather generation module is adopted from CADSM (Prajamwong, 1994) to generate daily reference crop evapotranspiration and rainfall data based on the monthly mean and standard deviations data. The on-farm irrigation scheduling module receives the basic project data and generated daily weather data to simulate the on-farm water balance. The daily simulation procedure includes three programming loops: (1) number of command

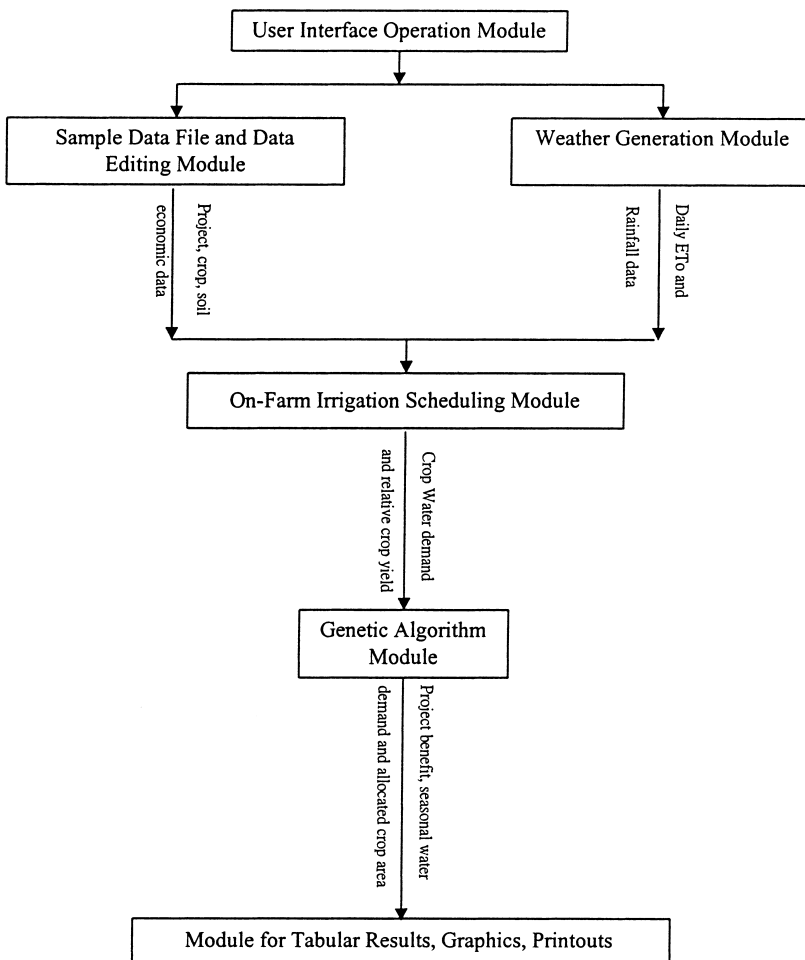


Fig. 3. The framework and logic employed in the irrigation decision support model.

areas within the simulated irrigation project, (2) number of crops within each command area, and (3) number of days from planting to harvest for each crop type. The daily simulation begins from the first command area in the project, the first crop within the command area, and the first Julian day for each crop. The procedure continues until all crops in each command area and all command areas in the irrigation project are processed. The output from this module includes relative crop yield and crop irrigation water requirements. Both outputs are the required inputs for the following genetic algorithm optimization module. Fig. 3 presents the framework and logic employed in the irrigation decision support model.

4. Application and results

4.1. Site description

The Wilson Canal System, close to the city of Delta in central Utah, was used in this study and is part of the many diversions in the Sevier River Basin operated by the Abraham Irrigation Company as an on-demand irrigation system with a good communications network. The Wilson Canal is 11,480 m in length with a source of water from the Gunnison Bend Reservoir. Figs. 4 and 5 depict the location of the Sevier River Basin and the Abraham Irrigation System in Utah, respectively.

The climate in the Delta area is essentially a cold desert type, which is arid with cold winters and warm summers. The UCA#2 and UCA#4 command areas were selected within the Wilson Canal System for evaluating the model. The UCA#2 command area has a 2896 m water course, 83.3 ha planted area, and three crop types: alfalfa, barley, and corn. On the other hand, the UCA#4 command area has a 12,350 m water course, and 311.3 ha planted area. In addition, four crop types are planted: alfalfa, barley, corn, and wheat.

Based on the 1993 recorded weather data from Delta, Utah, Fig. 6 summarizes the daily simulation results from the on-farm irrigation scheduling module to demonstrate the relationship between soil moisture, irrigation depth and rainfall for the alfalfa and barley crops in the UCA#2 command area. Tables 2 and 3 list the seasonal outputs for the UCA#2 and UCA#4 command area, respectively.

4.2. Application of the genetic algorithm

As mentioned in the previous section, four parameters are required to perform the GA method. As GA is based on random searching and is independent of random starting point, a series of runs with the same parameters must be performed to find the most appropriate parameters for the applied problems. In this study, three rules were followed to find the appropriate parameters as follows:

1. Based on the Dejong (1975) recommendation for a satisfactory genetic algorithm performance, a high crossover probability, a low mutation probability, and a moderate population size (e.g. 30–100) were used.

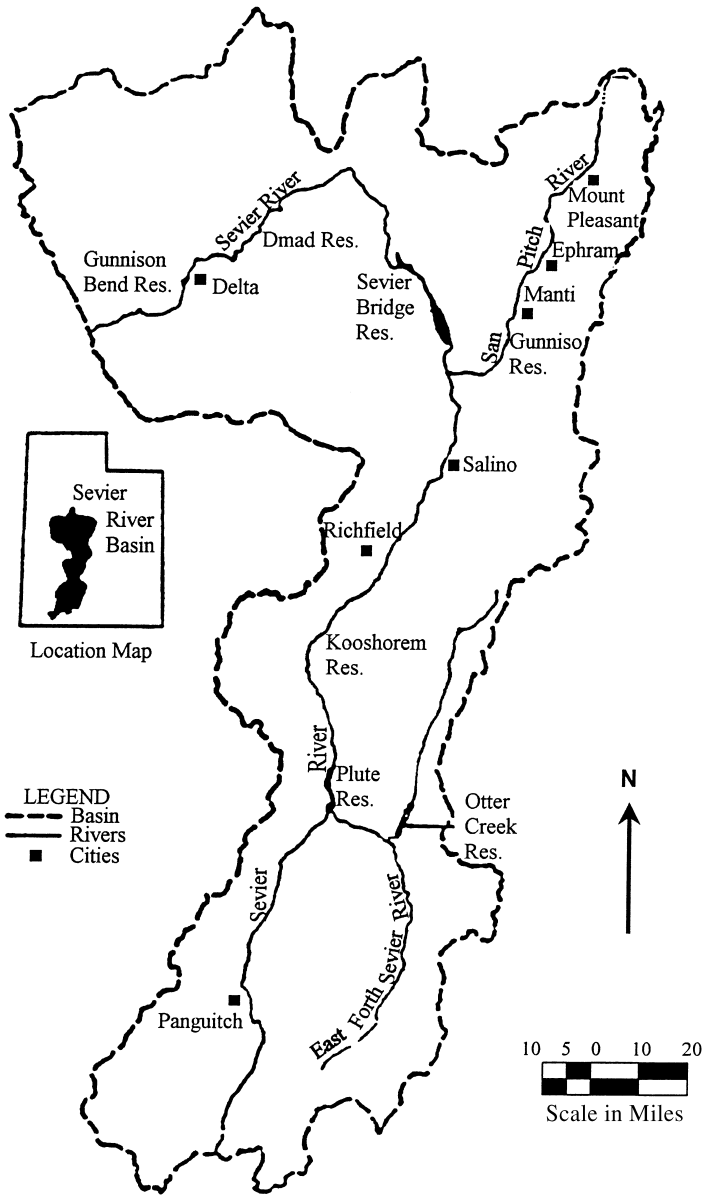


Fig. 4. Sevier River Basin, Utah (Tzou, 1989).

2. A series of runs with the same parameters is performed because the result for each run is essentially different and independent of the random starting point. If the standard deviation from all of the runs is high, the parameters may be inappropriate for obtaining the optimal or near-optimal solutions. On the other hand, parameters with a low standard deviation from all the runs have a better chance of obtaining the optimal

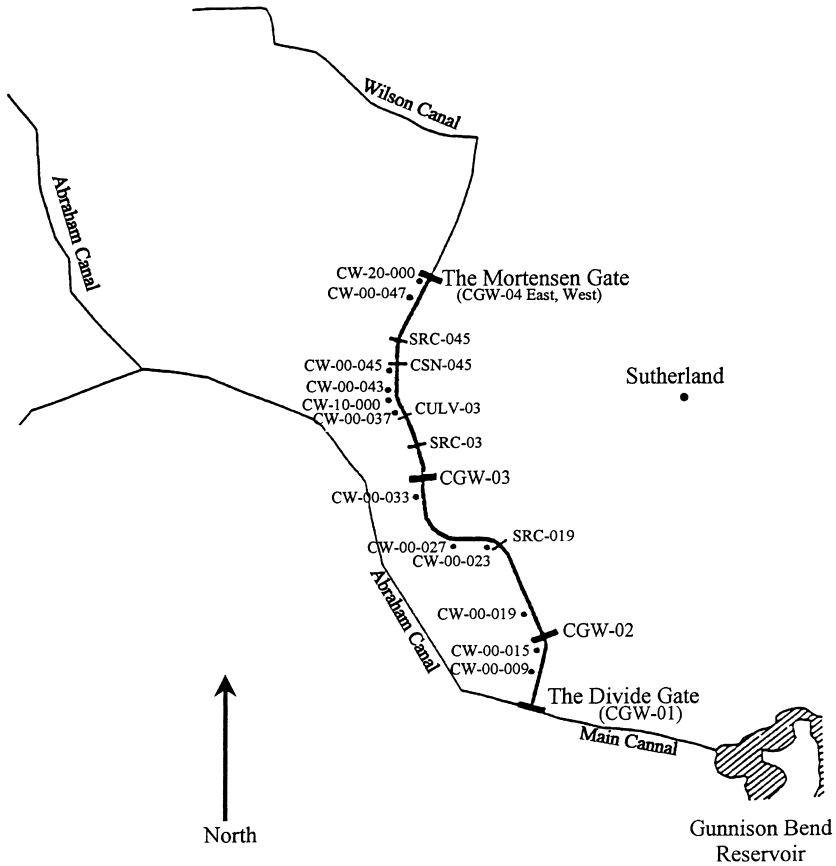


Fig. 5. The Abraham Irrigation System (Tzou, 1989).

Table 2

Seasonal outputs for the UCA#2 command area from the on-farm irrigation scheduling submodel

	Alfalfa	Barley	Corn
Potential ET (mm)	1038.03	555.58	514.84
Actual ET (mm)	907	505.71	460.94
Evaporation from wet soil surface (mm)	2.08	21.35	13.37
Number of irrigations	6	4	3
Total irrigation depth (mm)	1067.92	441.71	471.82
Deep percolation (mm)	70.14	29.39	37.19
Surface runoff (mm)	28.49	11.94	15.11
Yield reduction due to water stress (%)	11.43	3.59	14.54
Yield reduction due to waterlogging (%)	2.6	1.09	1.09
Relative crop yield (%)	86.27	95.36	84.52

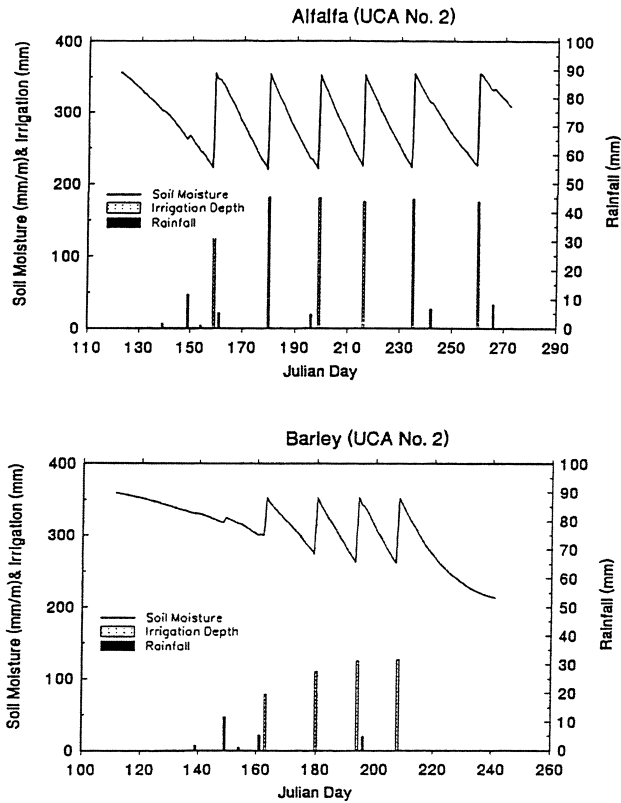


Fig. 6. The relationship between soil moisture, irrigation depth and rainfall for alfalfa and barley crops in the UCA#2 command area.

Table 3

Seasonal outputs for the UCA#4 command area from the on-farm irrigation scheduling submodel

	Alfalfa	Barley	Corn	Wheat
Potential ET (mm)	1039.33	572.08	523.35	611.18
Actual ET (mm)	906.15	528.68	469.61	558.05
Evaporation from wet soil surface (mm)	3.38	37.85	21.88	34.4
Number of irrigations	7	6	4	6
Total irrigation depth (mm)	1039.5	531.4	490.9	539.37
Deep percolation (mm)	68.32	35.13	38.42	35.64
Surface runoff (mm)	27.75	14.27	15.61	14.48
Yield reduction due to water stress (%)	11.73	3.49	14.16	5.39
Yield reduction due to waterlogging (%)	3.04	1.56	1.35	1.58
Relative crop yield (%)	85.59	95.0	84.68	93.11

or near-optimal results for the applied problems. Several sets of parameters are required to apply the model and the optimum set has a higher average and a lower standard deviation of the project's benefit from a series of runs.

The concept of simple genetic algorithm (SGA) describes a situation in which those strings with lower fitness values are thrown away and only strings with higher fitness values are kept from generation to generation. Therefore, as expected, the average fitness values from all strings has the tendency to increase, and the standard deviation decreases from the initial to final generation. Also, the chromosomes will look similar at the end of the generation because the fitness values from all strings will not differ greatly at this time.

5. Results and discussion

To test the effectiveness of the SGA work, four parameters are assigned as follows: (1) number of generations equal 100; (2) population sizes equal 50; (3) probability of crossover (P_c) equals 0.6; and (4) probability of mutation (P_m) equals 0.02. Fig. 7 displays a sample graph from the GA method to represent the Delta project's benefit during the searching process. Fig. 8 illustrates the relationship between the average, standard deviation and maximum values of benefit with the number of generations after running the model. The SGA performs quite well because the average of the fitness values has a tendency to increase from US\$ 88,292 to US\$ 139,113 as the number of generation increases. On the other hand, the standard deviation of fitness decreases from US\$ 17,127 to US\$ 3291. The maximum benefit is updated from generation to generation if the benefit has been improved. Notably, the maximum and average fitness values have nearly the same tendency to increase as a function of generation numbers.

Tables 4 and 5 present a sample of binary digit of 50 strings with their fitness values at begin and end of generation, respectively. The first column is the number of the strings

Table 4
Simple genetic algorithm run at the beginning of the generation

No.	String	Fitness (US\$ 1000)
Generation 0		
1	000101001010010110101101111100011110101111000000	79.236
2	0001011010110110000011001110101101010101110111010	74.611
3	01011000011010101000010100010010101111110001111	69.945
4	0011001001111000001100101111011110111001111101001	70.094
5	0010110001110100010010110100001001010101110011001	85.023
45	101101110110010100010111001100110010100000000011	130.484
46	0100110111010111111011001101101100001100011100000	74.232
47	0111010100111101011111000001111010001101101001110	77.295
48	0110011101111010000011110110011011011000010000101	100.946
49	1100001111100010100111111101010010011011000101001	96.992
50	1011110110101110100100101110111010111001111010000	68.877
Average		88.292
S.D.		17.127

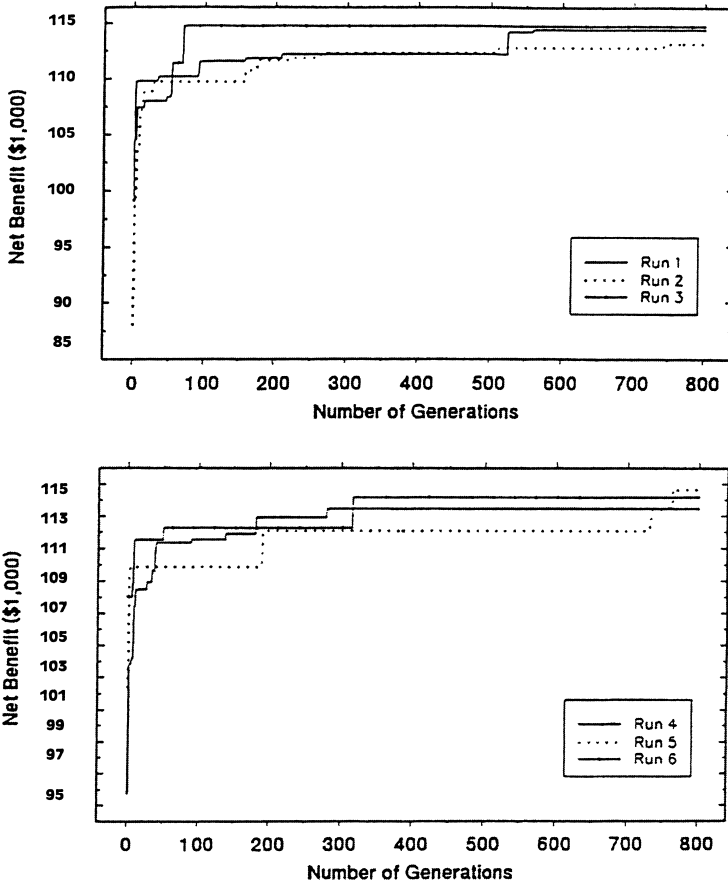


Fig. 7. Sample graphs from the genetic algorithm method with a population sizes of 50, P_c of 0.6, and P_m of 0.02.

within one population, the second column represents the binary digits for each string, and the final column is the fitness values for each strings. As presented and expected, each string includes 49 binary digits to represent seven crops within two command areas. The order of binary digits in each string appears to differ markedly in Table 4 because the fitness values have wide ranges at the beginning of the generation. On the other hand, Table 5 reveals that the order of binary digits in each string closely resemble each other because the fitness values from all the strings differ little at the end of the generation.

Based on many tests, the most appropriate parameters for this study are as follows: (1) number of generations equal 800; (2) population sizes equal 50; (3) probability of crossover equals 0.6; and (4) probability of mutation equals 0.02. Table 6 summarizes the final results from 10 runs for these parameters. Each run appear to be successful because the benefit, water demand and related crop-allocated areas are rather/quite/extremely close. As presented, the maximum benefit is up to US\$ 114,734 in run 3, with a standard

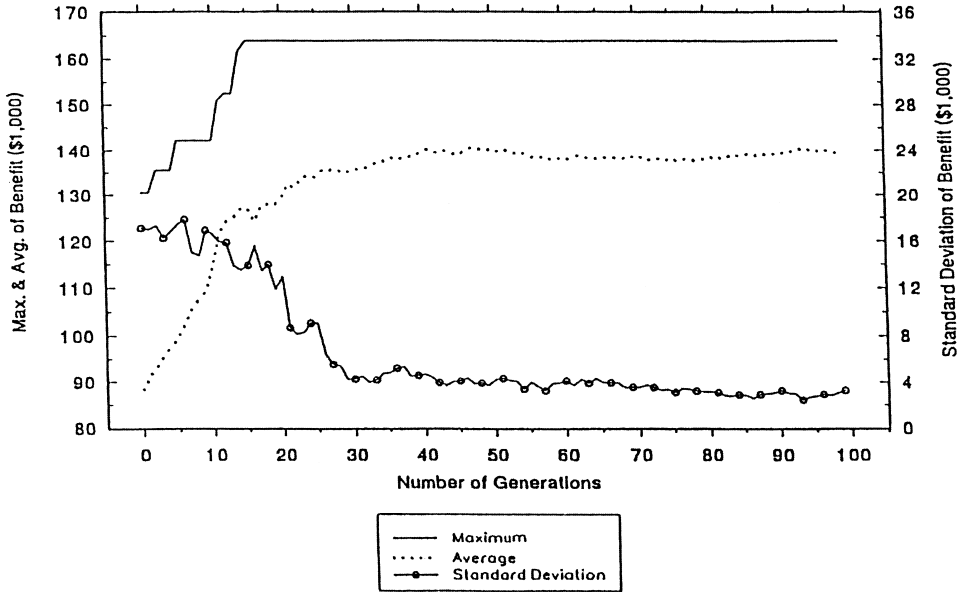


Fig. 8. The relationship between maximum, average and standard deviation of fitness with number of generations for SGA.

deviation of benefit as low as US\$ 646 from the 10 runs Therefore, the parameters are confident of obtaining the near global optimal values for this irrigation project planning problem. The average values in Table 6 can be considered as the optimal planning for Delta, Utah as shown in Table 7.

Two other optimization method, the simulated annealing (SA; Walker, 1992) and iterative improvement method, are used in this study to compare the applicability of the

Table 5
Simple genetic algorithm run at the end of the generation

No.	String	Fitness (US\$ 1000)
Generation 100		
1	1100001011111001000101110011000011000101010001001	141.668
2	1100000011111001000101110011000011000101010001001	141.584
3	1100000011101001000101110011001011000101010001001	135.004
4	1100000011101101000101110011000011000101010001001	141.723
5	1100000011111001000101110011000011000101010001001	141.584
45	1100000011111001000101110011001011000101010001001	134.794
46	1100001011101101000101110011000011000101010001001	141.807
47	1100000011111001000101110011001011000101010001001	134.794
48	1100000011111001000101110011001011000101010001001	134.794
49	1100000011111001000101110011000011000101010001001	141.584
50	1100000011101101000101110011001011000101010001001	134.933
Average		139.113
S.D.		3.291

Table 7

The optimal planning from GA method for the Delta irrigation project

	Net benefit (US\$ 1000)	Water demand (1000 m ³)	Alfalfa (%)	Barley (%)	Corn (%)	Wheat (%)
Project	113.826	3029.594	–	–	–	–
UCA#2	–	736.362	70.105	19.044	10.85	–
UCA#4	–	2293.232	41.467	38.926	14.42	5.187

GA method for planning the Delta, Utah irrigation project. Although the SA method represents the second global optimization method, the iterative improvement method is recognized as the conventional optimization method to apply in this study. Table 8 summarizes the results from the GA, SA and iterative improvement methods for the Delta irrigation project.

Table 8 presents some valuable information as discussed in the following.

1. Column 2 indicates that the maximum benefits of the genetic algorithm and simulated annealing methods are US\$ 113,826 and US\$ 111,494, respectively. The maximum benefits from the genetic algorithm and simulated annealing methods are quite close. Therefore, both methods can find the near global optimum benefit. On the other hand, the maximum benefit from the iterative improvement method is only US\$ 105,444. Clearly, the conventional optimization method (iterative improvement) can only occasionally find the local optimum values. However, the global optimization methods can consistently obtain the near optimal values.
2. Considering the crop area percentages in columns 5–8, the three optimization methods have the same tendency to search for the portions of crops within the UCA#2 and

Table 8

Comparison of the three optimization methods^a

(1) Methods	(2) Items	(3) Max. benefit (US\$ 1000)	(4) Water demand (1000 m ³)	(5) Alfalfa (%)	(6) Barley (%)	(7) Corn (%)	(8) Wheat (%)
Genetic algorithm (S.D.=0.646)	Project	113.826	3029.594	–	–	–	–
	UCA#2	–	736.362	70.105	19.044	10.85	–
	UCA#4	–	2293.232	41.467	38.926	14.42	5.187
Simulated Annealing (S.D.=1.581)	Project	111.494	3002.337	–	–	–	–
	UCA#2	–	723.566	67.533	19.133	13.336	–
	UCA#4	–	2278.771	41.216	30.757	22.758	5.27
Iterative Improvement (S.D.=2.911)	Project	105.444	2922.927	–	–	–	–
	UCA#2	–	665.456	56.056	23.583	20.361	–
	UCA#4	–	2257.471	39.693	27.996	21.619	10.693

^a Note: S.D. means standard deviation.

UCA#4 command areas. Restated, the order of crop portions in decreasing order is alfalfa>barley>corn in the UCA#2 command area, and alfalfa>barley>corn>wheat in the UCA#4 command area. Therefore, all three optimization methods have the ability to look at Delta, Utah for maximum benefit. However, the GA and SA methods can obtain more global optimal values than the iterative improvement method. Table 8 also reveals that the crop area percentages from the genetic algorithm and simulated annealing methods for the UCA#2 and UCA#4 command areas closely resemble each other in terms of maximum benefit. On the other hand, the crop area percentages from the iterative improvement method for UCA#2 have different crop areas than the GA and SA methods.

3. Consider the standard deviation in column 1 for the three methods. The standard deviations are selected from the most appropriate data set for the GA and SA methods and are calculated from maximum benefit for many runs (i.e. 10 runs for the GA and SA methods, and 50 runs for the iterative improvement method). As shown, the GA method has the lowest standard deviation, 0.646, followed by 1.581 for the SA method. The iterative improvement has the highest standard deviation of 2.911. Therefore, the order of benefit for the three methods is as follows: GA method>SA method>iterative improvement method.
4. In considering the water demand for Delta, Utah in column 4, the order of water requirements is as follows: Iterative Improvement method (2,922,927 m³)<SA method (3,002,337 m³)<GA method (3,029,594 m³). Based on previous discussions of maximum benefit, crop area percentages, standard deviation, and water demand for the three optimization methods, the GA and SA methods are clearly preferred over the iterative improvement to help irrigation managers or governmental agencies to reach irrigation project planning decisions.

6. Conclusions

This work develops a model based on the on-farm irrigation scheduling and the simple genetic algorithm optimization method to provide guidelines on irrigation planning and management. The model consists mainly of six basic modulus: (1) main module directs the running of the model with pull-down menus ability, (2) data module to enter the require data by friendly user-interface, (3) weather generation module to generate the daily weather data, (4) on-farm irrigation scheduling module to simulate the daily water requirement and relative crop yield, (5) genetic algorithm module to optimize the project maximum benefit, and (6) results module to present results by tables, graphs and printouts.

In addition, the model is applied to Delta, Utah for optimizing the maximum crop production benefits and identifying the crop area-allocated percentages with the application of the genetic algorithm method. The simple genetic algorithm performs quite well in this study because the average of fitness increases and the standard deviation of fitness decreases from initial to final generation. The most appropriate parameters of GA for this study are as follows: (1) number of generations equal 800, (2) population sizes equal 50, (3) probability of crossover equals 0.6, and (4) probability of mutation

equals 0.02. The optimal planning for Delta, Utah can be summarized as follows: (1) project benefit equals US\$ 113,826, (2) project water demand equals $3.029 \times 10^6 \text{ M}^3$, (3) crop area percentages within UCA#2 are 70.1, 19 and 10.9% for alfalfa, barley, and corn, respectively, and (4) area percentages of crops within UCA#4 command area are 41.5, 38.9, 14.4, and 5.2% for alfalfa, barley, corn and wheat, respectively.

Acknowledgements

This study also demonstrates that the genetic algorithm can be efficiently applied to plan the Delta, Utah project. In doing so, the genetic algorithm can hopefully be applied to solve more complicated irrigation water management problems.

References

- Davis, L., 1991. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.
- Dejong, K.A., 1975. An analysis of the behavior of a class of genetic adaptive systems. Ph.D. Dissertation, University of Michigan, MI.
- Fahmy, H., King, P., Wentzel, M., Seton, J., 1994. Economic optimization of river management using genetic algorithms. In: *Proceedings the 1994 International Summer Meeting International Conference on Genetic Algorithms*, Kansas City, MI.
- Goldberg, D.E., 1989. Genetic algorithms. In: *Search, Optimization and Machine Learning*. Addison-Wesley, New York.
- Hargreaves, G.L., Hargreaves, G.H., Riley, J.P., 1985. Agricultural Benefits for the Senegal River Basin. *J. Irrig. Drain. Eng. ASCE* 111 (3), 265–275.
- Hill, R.W., Keller, A.A., Boman, B., 1982. Crop yield models adapted to irrigation scheduling programs. Appendix F: CRPSM user manual and sample input and output. Utah Agricultural Experiment Station, USU, Research Report 100.
- Jesus, C.M., Marino, M.A., Holzapfel, E.A., 1987. Planning model of irrigation district. *J. of Irrig. Drain. Eng. ASCE* 113 (4), 549–563.
- Keller, A.A., 1987. The USU unit command area model. Water Management Synthesis II Project, WMS Report No. 71, USAID, Logan, UT.
- Lakshminarayana, V., Rajagopalan, S.P., 1977. Optimal cropping pattern for basin India. *J. of Irrig. Drain. Division ASCE* 103 (IR1) 53–71.
- Maji, C.C., Heady, E.O., 1978. Interemporal allocation of irrigation water in the Mayurakshi Project (India): an application of chance-constrained linear programming. *Wat. Resour. Res.* 14 (2), 190–196.
- Matanga, G.B., Marino, M.A., 1979. Irrigation planning 1. Cropping pattern. *Wat. Resour. Res.* 15 (3), 672–678.
- McKinny, D.C., Lin, M.D., 1994. Genetic algorithm solution of groundwater management models. *Wat. Resour. Res.* 30 (6), 1897–1906.
- Neale, C.M.U., 1994. Water management. Course Notes, Utah State University, Logan, UT.
- Paudyal, G.N., Gupta, A.D., 1990. Irrigation planning by multilevel optimization. *J. of Irrig. Drain. Eng. ASCE* 116 (2), 273–291.
- Prajamwong, S., 1994. Command area decision support system for irrigation projects. Ph.D. Dissertation, Utah State University, Logan, UT.
- Raman, H., Mohan, S., Rangacharya, N.C.V., 1992. Decision support for crop planning during droughts. *J. of Irrig. Drain. Eng. ASCE* 118 (2), 229–241.
- Smith, M., 1991. CROPWAT: Manual and guidelines. FAO of UN, Rome, Italy.
- Tzou, C.T., 1989. Verification and calibration of the USU hydraulic model. M.S. Thesis, Utah State University, Logan, UT.

- Walker, J.C., 1992. Simulated Annealing applied to the peanut growth model for optimization of irrigation scheduling. Ph.D. Dissertation, North Carolina State University, Raleigh, NC.
- Wang, Q.J., 1991. The genetic algorithm and its application to calibrating conceptual rainfall-runoff models. *Wat. Resour. Res.* 27 (9), 2467–2471.
- Wentzel, M.W., King, J.P., Finch, P.R., Blair, A.W., 1994. Pipe network pumping strategy optimization by genetic algorithm. In: Proceedings of the 1994 International Summer Meeting International Conference on Genetic Algorithms, Kansas City, MI.