ISSN 1746-7233 (print)
World Journal of Modelling and Simulation
Vol. xx (2008) No. xx, pp. xxx-xxx

(Will be set by the publisher)

Estimation of Optimal Crop Plan Using Nature Inspired Metaheuristics

Millie Pant[a], Radha Thangaraj[a +], Deepti Rani[b], Ajith Abraham[c],

Dinesh Kumar Srivastava[d]

[a]*Department of Paper Technology, Indian Institute of Technology Roorkee, Saharanpur, 247 001, India*

[b]*Department of Rural Engineering, University of Evora, Evora, 7002-054, Portugal*

[c]*Center of Excellence for Quantifiable Quality of Service, Norwegian University of Science and Technology, Trondhiem, NO-7491, Norway*

[d]*Department of Hydrology, Indian Institute of Technology Roorkee, Roorkee – 247 667, India*

(Received December 2008, accepted xxx 2008, will be set by the editor)

Abstract. Irrigation management has gained significance due to growing social needs and increasing command for food grains while the available resources have remained limited and scarce. Irrigation management includes optimal allocation of water for irrigation purposes, optimal cropping pattern for a given land area and water availabilities with an objective to maximize economic returns. In the present study we consider an optimization model based on linear programming for determining optimal crop plan for command area of Pamba-Achankovil-Vaippar (PAV) link project, Kerala, India. The crop planning model considers various resource constraints (land area, seeds, manure, fertilizers etc.) availability etc. adaptive to national conditions, with the objective to maximize net irrigation benefits. For crop planning, the extent of quantity available for fertilizers, manure and seeds as inputs were unknown. Estimates for the extent of unknown minimum quantities of these resource inputs available are obtained with the help of crop planning model itself. For optimal releases made from reservoir using a multi-reservoir operation model, optimal crop plans are developed under adequate, normal and limited irrigation water defined by 50 percent, 75 percent and 90 percent water year dependable flows, respectively. The optimization model is solved using four popular Evolutionary Algorithms (EA) viz. Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE) and Evolutionary Programming (EP). EA are compared with each other in terms of average CPU time, average number of generations, standard deviation etc. the algorithms are also compared with LINGO, a popular software used for solving LPP models.

Keywords: Evolutionary Algorithms, Reservoir, Crop Plan, Constrained Optimization, Penalty Function.

[+] Corresponding author. Tel.: +91 – 9808517165; fax: +91- 132 – 2714311.
E-mail address: t.radha@ieee.org (Radha Thangaraj)

1. Introduction

Agriculture in India is one of the most prominent sectors in its economy. Agriculture and allied sectors like forestry, logging and fishing accounts nearly for 16.6% of the GDP and 8.56% of India's exports. About 43% of India's geographical area is used for agricultural activity. About 70% of India is directly dependent on agriculture, which is the main component of most state economies in India. The dependence of agriculture on monsoon has boosted the need of irrigation management.

Also, it is worth mentioning that most of the irrigation management problems can be formulated as optimization models. Some of the interesting examples include Windsor and Chow [1], who developed a multilevel optimization model for a farm irrigation system. Linear programming (LP) was used by them at second level of optimization for optimal land and water allocation. At first level dynamic programming was used to estimate the expected data for LP model. Rogers and Smith [2] employed the interaction of surface and ground water systems in irrigation management using deterministic LP model.

Lakshminarayan and Raja Gopalan [3] used an LP model to determine an optimal cropping pattern and optimal release policy from canals and tube wells for maximizing the economic returns. Matanga and Marinno [4] and Chavez-Morales et al. [5] used linear optimization models to obtain optimal cropping patterns with different objective functions. Two chance constrained LP models were formulated by Maji and Heady [6] to maximize net return from project area for the Mayurakshi project in India. Raju and Kumar [7] proposed a crop planning model with the objective of maximizing irrigation benefits for a typical irrigation system. Kuo et al. [8] used Genetic Algorithm (GA) based model for irrigation project planning for case study of Delta, Utah with the objective of maximization of net economic benefits for a cultivable command area of 394.6 ha. Raju and Kumar [9] applied Genetic Algorithms for irrigation planning in Indian context and compared the results with LP approach. Their results showed the comparable performance of GA with LP. In an application oriented research article, Mayer et al [10] discussed the optimal parameter settings of Evolutionary Algorithms for optimization of agricultural system models. In one of the recent studies Zhang et al. [11] studied the corn optimization irrigation model using Genetic Algorithms.

For the present study we analyzed the performance of four EA namely GA, EP, PSO and DE to obtain optimal crop plans under adequate, normal and limited irrigation water availability for irrigation area subject to various constraints in context of the national scenario, under the PAV link project, India. Penalty function method is used for dealing with constraints while using EA

2. Study Area: Pamba-Achankovil-Vaippar Link Project, Kerala, India Submitting

The proposed Pamba-Achankovil-Vaippar Link project has three storage reservoirs, two tunnels, necessary canal system and a few power generating units [12]. The Punnamedu reservoir (reservoir-2) is located at a higher elevation on river Pamba Kal Ar in the Pamba basin of Karala state, which serves a part/full of its downstream mandatory requirements and supplies surplus water to reservoir-1 by intra-basin export of surplus water (diversion) through tunnel-2. The Achankovil Kal Ar reservoir (reservoir-1) located on Achankovil Kal Ar River in Achankovil river basin of Kerala state, supplies water for irrigation purposes to the state of Tamilnadu, through tunnel-1 to the main canal. The water from main canal is then distributed to the command area of Vaippar basin in Tamilnadu state. The reservoir is proposed as a within-the-year storage scheme. Figure 1 shows a schematic representation of the PAV link diversion system. Besides this, reservoir-1 releases water for power generation. The Achankovil reservoir (reservoir-3), which is located on Achankovil river in the Achankovil river basin of Kerala state, besides acting as a pumped storage scheme accommodating the water drawn from the upstream reservoir-1, also serves the purpose of releasing water to downstream to meet its downstream mandatory demands. Also, if there is deficit at reservoir-1 the surplus water of reservoir-3 can be pumped back to reservoir-1. The monthly inflow at reservoir-1 for the 50 percent, 75 percent and 90 percent water year dependable flows are shown in Figure 2.

The GCA (gross command area) potential and CCA (culturable command area) potential of the project would be 145,573 and 101,555 ha, respectively. The proposal is to irrigate 91,400 ha (CCA actually considered) of area per annum with an irrigation intensity of 90 percent. The proposed cropping pattern was formulated by the Tamilnadu State Agriculture Department exclusively for this project. Crop areas given in the report were determined on the basis of the food requirements of the population likely to be benefited from project. The suggested cropping pattern consists of 8 crops namely; Paddy, Oilseed, Jowar, Vegetables (Brinjal, Ladyfinger and Beans), Pulses, Bajra, Cotton and Chillies; the proposed corresponding area allocation for each crop is 15234, 7109, 12187, 15233, 6093, 15233, 12187 and 8124 ha, respectively, and crop yields from these crops under irrigation area; 5.39, 1.51, 2.56, 3.0, 0.741, 2.56, 1.66 and 1.51 metric tones (M.T) per unit cropped area, respectively. The gross irrigation requirement for the command area as per project is 635 mcm including transmission losses. The total production from proposed irrigation, cost of produce and expenses on cultivation of various crops under irrigation conditions, i.e., total cost on seeds, fertilizers, manure, irrigation charges and labour per unit area of each crop is available from project report [13].

Optimal water released from reservoir-I is obtained through joint operation of reservoirs using DP with successive approximation (DPSA) [14]. The main canal emerging from reservoir-1, further distributes released water to users (Reach-I, II and III). In this study optimal cropping pattern is obtained for the total area lying under Reach-I, II and III.

3. Evolutionary Algorithms Used for Comparison

Evolutionary Algorithms (EAs) may be termed as general purpose algorithms for solving optimization problems. These algorithms have been successfully applied to a wide range of problems occurring in various fields [15] – [17]. Each EA is assisted with special operators that are based on some natural phenomenon. These algorithms are iterative in nature and in each iteration special operators are invoked to manipulate the population of candidate solutions in order to reach to optimal (or near optimal) solution. Although all algorithms have same modus-operendi like starting with a population of candidate solutions which are manipulated so as to be guided towards the optimum solution, each algorithm has certain unique feature associated with it which makes it different. A brief description of the three EAs used in this study is given in the following subsections. Pseudo codes of the algorithms used in the present study are given in Appendix A.

3.1. Genetic Algorithms

Genetic Algorithms (GAs) are perhaps the most commonly used EA for solving optimization problems. In fact it was the success of GAs that made the concept of EAs widely popular for solving various optimization problems. The natural phenomenon which forms the basis of GA is the concept of *survival of the fittest*. GAs were first suggested by John Holland [18]. The main operators of GA are *Selection*, *Reproduction* and *Mutation*. GAs work with a population of solutions called *chromosomes*. The fitness of each chromosome is determined by evaluating it against an objective function. The chromosomes then exchange information through *crossover* or *mutation.* In the present study we have used arithmetic crossover [19]. It is a two parent crossover operation in which two individuals selected from the population undergo reproduction to produce two new offspring. The working of Arithmetic Crossover may be defined as:

Arithmetic Crossover: if P_i and P_j are two parents the offsprings C_i and C_j are generated as

$C_i = a*P_i + (1 - a)*P_j$

$C_j = (1 - a)*P_i + a*P_j$

Where ‘a’ may be taken as a constant or variable. The Arithmetic Crossover is followed by Gaussian mutation where a Gaussian noise is added to the particles to add perturbation to the particles. More detail on the working of GAs may be obtained from [19] - [21] etc.

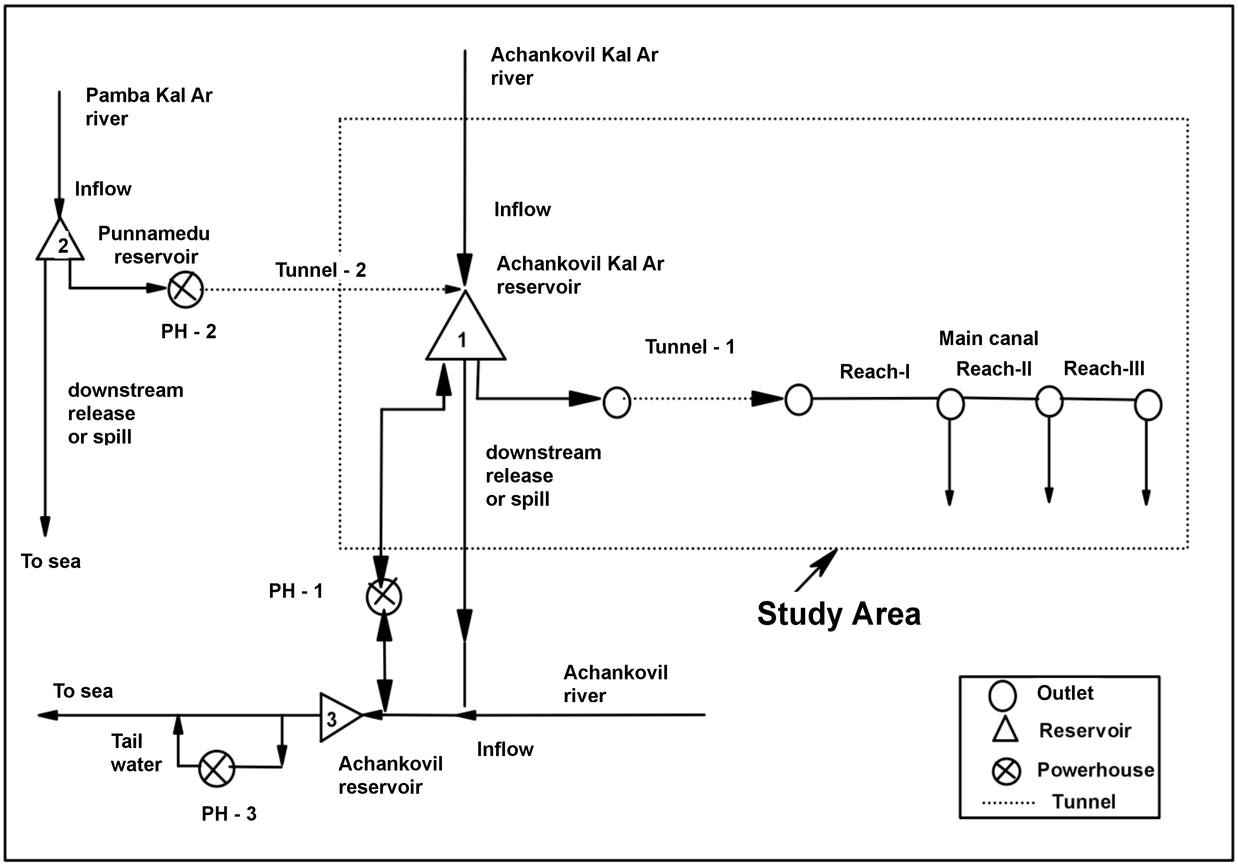

Figure1Schematic Diagram for PAV Link Diversion System

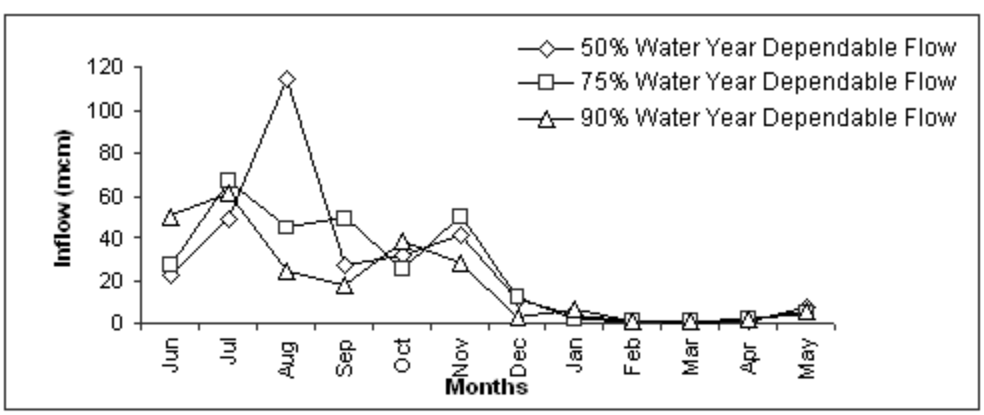

Figure 2 Monthly inflows at Reservoir-I

3.2. Particle Swarm Optimization

Particle swarm optimization (PSO) was first suggested by Kennedy and Eberhart [22]. The mechanism of PSO is inspired from the complex social behavior shown by the natural species. For a D-dimensional search space the position of the ith particle is represented as $X_i = (x_{i1},x_{i2},..x_{iD})$. Each particle maintains a memory of its previous best position $P_i = (p_{i1}, p_{i2} \ldots p_{iD})$ and a velocity $V_i = (v_{i1}, v_{i2},\ldots v_{iD})$ along each dimension . At each iteration, the P vector of the particle with best fitness in the local neighborhood, designated g, and the P vector of the current particle are combined to adjust the velocity along each dimension and a new position of the particle is determined using that velocity. The two basic equations which govern the working of PSO are that of velocity vector and position vector are given by:

$$v_{id} = \omega v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

The first part of equation (1) represents the inertia of the previous velocity, the second part is tells us about the personal thinking of the particle and the third part represents the cooperation among particles and is therefore named as the social component. Acceleration constants c_1, c_2 and inertia weight ω are predefined by the user and r_1, r_2 are the uniformly generated random numbers in the range of [0, 1].

3.3. Differential Evolution

Differential Evolution was proposed by Storn and Price [23]. It is a population based algorithm like genetic algorithms using the similar operators; crossover, mutation and selection. The main difference in constructing better solutions is that genetic algorithms rely on crossover while DE relies on mutation operator [24]. DE works as follows: First, all individuals are initialized with uniformly distributed random numbers and evaluated using the fitness function provided. Then the following will be executed until maximum number of generation has been reached or an optimum solution is found.

For a D-dimensional search space, each target vector $x_{i,g}$, a mutant vector is generated by

$$v_{i,g+1} = x_{r_1,g} + F * (x_{r_2,g} - x_{r_3,g}) \quad (3)$$

where $r_1, r_2, r_3 \in \{1,2,....,NP\}$ are randomly chosen integers, must be different from each other and also different from the running index i. F (>0) is a scaling factor which controls the amplification of the differential evolution $(x_{r_2,g} - x_{r_3,g})$. In order to increase the diversity of the perturbed parameter vectors, crossover is introduced [25]. The parent vector is mixed with the mutated vector to produce a trial vector $u_{ji,g+1}$,

$$u_{j,i,g+1} = \begin{cases} v_{j,i,g+1} & if \quad rand_j \leq Cr \vee j = k \\ x_{j,i,g} & otherwise \end{cases} \quad (4)$$

where j = 1, 2,…, D; $rand_j \in [0,1]$; CR is the crossover constant takes values in the range [0, 1] and $j_{rand} \in (1,2,.....,D)$ is the randomly chosen index.

Selection is the step to choose the vector between the target vector and the trial vector with the aim of creating an individual for the next generation. Several versions of DE are available in literature. In the present study we use the *DE/rand/1/bin*-version, which is apparently the most commonly used version**.**

3.4. Evolutionary Programming

Initially, Evolutionary Programming (EP) was introduced as an evolutionary approach to artificial intelligence [26], however, it has been successfully applied to many numerical optimization problems [27] – [29]. Optimization by EP consists of two major steps:

- Mutate all the solutions in the current population.
- Select the next generation from the mutated and the current solutions.

In the present article we use self adaptive EP (SAEP) which was introduced by Back and Schwefel [30] and Fogel [29] and was shown to be more efficient than the normal EP. In SAEP each individual is taken as a pair of real-valued vectors, (x_i , σ_i) for all i=1,…,M. The x_i's give the ith member's object variables and σ_i's the associated strategy parameters. The objective function is evaluated for each individual. Mutation in EP creates a single offspring (x_i' , σ_i'), from each parent (x_i , σ_i) for all i=1,…,M by

$$\sigma_i'(j) = \sigma_i(j) \exp(\tau\, N(0,1) + \tau'\, N_j(0,1))$$

$$x_i'(j) = x_i(j) + \sigma_i'(j)\, N_j(0,1) \quad \text{for all } j = 1,.....n. \qquad (5)$$

where N(0,1) denotes a random number distributed by Gaussian or Cauchy distribution. In the present article we used Cauchy mutation. The factors τ and τ' are commonly set to $1/\sqrt{2n}$ and $1/\sqrt{2\sqrt{n}}$ respectively.

3.5. Penalty Method for Solving Constrained Optimization Problems

The mathematical model considered in the present study is subject to various constraints and penalty function approach is used to solve the constraints. The search space in Constrained Optimization Problems (COPs) consists of two kinds of solutions: feasible and infeasible. Feasible points satisfy all the constraints, while infeasible points violate at least one of them. Therefore the final solution of an optimization problem must satisfy all constraints. In the penalty function approach, the constrained problem is transformed into an unconstrained optimization algorithm by penalizing the constraints and building a single objective function, which is minimized using an unconstrained optimization algorithm. That is,

$$F(x) = f(x) + \lambda\ p(x) \qquad (6)$$

Where

$$p(x_i, t) = \sum_{m=1}^{n_g+n_h} \lambda_m(t) p_m(x_i) \qquad (7)$$

$$p_m(x_i) = \max\{0, g_m(x_i)^\alpha\}$$
$$\text{if } \ m \in [1,......,n_g]\ (\text{inequality}) \qquad (8)$$

$$p_m(x_i) = |h_m(x_i)|^\alpha$$
$$\text{if } \ m \in [n_g + 1,......,n_g + n_h]\ (\text{equality}) \qquad (9)$$

Where α is a positive constant representing the power of the penalty. The inequality constraints are considered as g(x) and h(x) represents the equality constraints. n_g and n_h denotes the number inequality and equality constraints respectively. λ is the constraint penalty coefficient.

4. Mathematical Model

A linear programming based optimization model is used for crop planning. The model maximizes net returns from crops and yields optimal crop plan and monthly releases required from reservoir-1. Surface water, land availability, fertilizers (N, P, and K), seeds and manure requirements are considered as constraints in the model. For the purpose of modeling, the crops have been segregated as food grains, cash crops and others. Paddy, Jowar and Bajra are clubbed together as these falls under the category of food grains.

Crop Planning Model

$$\text{Max } Z = G_T - P_T \qquad (10)$$

where

$$G_T = \sum_{i=1}^{N} y_i b_i A_i \quad \text{and}$$

$$P_T = \sum_{i=1}^{N} (CS_i + CM_i + CF_i + CI_i + CLH_i) A_i$$

Subject to:

$$\sum_{i=1}^{N} (W_{i,t}) A_i \le R_t \text{ for all t} \tag{11}$$

$$\sum_{i=1}^{N} (\lambda_{i,t}) A_i \le A_T \text{ and } \sum_{i=1}^{N} A_i \le A_T \tag{12}$$

$$\sum_{i} y_i A_i \ge \sum_{i} y_T^i \text{ for i=1,3 and 6 and } y_i A_i \ge y_T^i \tag{13}$$

$$\sum_{i=1}^{N} F_{f,i} A_i \le F_{f,T} \text{ for all f} \tag{14}$$

$$\sum_{i=1}^{N} M_i A_i \le M_T \tag{15}$$

$$S_i A_i \le S_T \text{ for all i} \tag{16}$$

$$A_{i,\min} \le A_i \le A_{i,\max} \tag{17}$$

In the above model, equation (10) represents the objective function to maximize the net returns from crops and yields optimal crop plan. Equations (11) – (17) represent constraints. Surface water availability constraints which should be less than or equal to the surface water available is given by (11) and land availability constraints which should be less than or equal to the total area available are given by (12). Equation (13) represents the yield requirement constraint which should be greater than or equal to the proposed yield requirement. Fertilizers availability constraints are given by equation (14). Three types of fertilizers have been considered in the application of model, i.e., Nitrogen, Phosphorus and Potassium (N, P, K). Manure and Seed Availability Constraints are given by (15) and (16) respectively. Constraint (9) gives the Seeds Availability Constraints and finally the Bounds on Areas under Various Crops are given by (17). Index i = 1,2,…8 represents various crops for Paddy, Oilseeds, Jowar, Vegetables, Pulses, Bajra, Cotton and Chilies respectively.

A minimum crop area constraint has been specified on each crop so as to see that area occupied by crops does not fall below area under rain-fed cultivation. It has also been specified that area proposed under cotton and chilies should not be more than 18 percent and 17 percent of annual irrigation. This condition is justified because their yields have high revenues and optimally higher area allocation to these crops may cause reduction in food grain output, which is socially undesirable. It has been considered essential that total food grain production should not be less than 101, 995 M.T.

Nomenclature

Z	Annual return from irrigated agriculture
G_T	Total annual gross returns from crops
P_T	Total annual net expenses on cultivating crops
N	Total number of crops
A_i	Area under ith crop
CS_i	Expenses on seeds for ith crop per unit area

CM_i	Expenses on manure for ith crop per unit area
CF_i	Expenses on fertilizers for ith crop per unit area
CLH_i	Expenses on labor and machinery for ith crop per unit area
CI_i	Expenses on irrigation water charges for ith crop per unit area
y_i	Crop yield in weight units from ith crop per unit area
b_i	Value of crop produce from ith crop per unit yield
$W_{i,t}$	Gross irrigation requirement of ith crop during time period t in terms of depth
R_t	Irrigation water released/required from reservoir in time period t
$\lambda_{i,t}$	Use coefficient of the ith crop during time period t
A_T	Total area under irrigation per annum
y_T^i	Total yield required from ith crop
$F_{f,i}$	Quantity of fertilizer type *f* required per unit area for ith crop
M_i	Quantity of manure required per unit area for ith crop
M_T	Total available quantity of manure
S_i	Quantity of seeds required per unit area for ith crop
S_T	Total available quantity of seeds
$A_{i,\min}$	Lower limit on the area under ith crop
$A_{i,\max}$	Upper limit on the area under ith crop

5. Experimental Settings

In this section we give the data used for the mathematical model used in section 3 and the parameter settings for EA. From information available [31] estimates of average values of quantities/ha required for each crop for resource inputs, i.e., seeds, manure and fertilizers are obtained. Total requirements of these resources are obtained from these values and crop area allocation as per project report. Initially it was assumed that total quantity available for each resource is equal to total quantity required for the resource. The crop planning model is solved using LINGO package. The first trial run is made of the model assuming that the amount of each resource available is equal to the required amount, and from results it was seen that out of the total CCA, i.e., 91400 ha only 88818.64 ha is allocated to the crops, i.e., with this trial the total CCA was not allocated to various crops (please also see Table 1). Further model runs were made by varying quantity of resource availability in some percent of required amount the area allocations for these trials are given in Table 1.

Finally it was assumed that 120 percent of the total quantity initially estimated for each resource may be considered as the extent of quantity available as input, for which almost all the area proposed has been allocated (please also see Table 2).

5.1. Parameter Selection

As mentioned in earlier, in Section 3, EA are associated with certain parameters that should be fine tuned so that the algorithm gives the best performance. For the four EA taken in the present study we conducted a series of experiments for all the algorithms with varied parameters and selected the ones that gave the best results. The experimental settings are given as follows:

<u>GA settings</u>

Population size: 20

Encoding: real

Crossover: Arithmetic Crossover with crossover rate as 0.5

Mutation: Gaussian

<u>PSO settings</u>

Population size: 20

Inertia weight w: linearly decreasing

Acceleration constants c_1 and c_2: 2

<u>DE settings</u>

Population settings: 20

Crossover Constant: 0.5

Scaling Factor: 0.5

<u>EP Settings</u>

Population Size: 20

Mutation: Cauchy

In order to give a fair chance to all the algorithms we initiated the population with the same seed of random number. Maximum number of generations for all the algorithms was set as 1000. All the algorithms were executed on P-IV using DEV C++.

However, we would like to maintain that the choice of parameters is generally problem specific and may be changed depending on the number of variables, nonlinearity, number of constraints etc.

Table 1 Optimal area allocations with variable resource inputs available

Resource Inputs	Optimal Area Allocations (ha)
80%	73120.00
90%	74384.31
100%	88818.64
110%	89754.08
120%	91399.99
130%	91400.00

Table 2 Extents of resource available

Resource		Extent Availability^{+}
Fertilizers (kg)	N	5774510
	P	4911240
	K	2303256
Manure (M.T.)		1431135
Seeds (kg)	Paddy	1188252
	Oil seeds	2102154
	Jowar	93838.8
	Vegetables	178403.04
	Pulses	329049
	Bajra	255906
	Cotton	319893
	Chillies	329049

6. Numerical Results

In this section we give a comparison of numerical results obtained from the four algorithms and LINGO. A comparative performance of algorithms with each other is also given. Each EA was executed 50 times and the best value throughout the run was recorded for 50%, 75 % and 90% water year dependable flow (WYDF). From the Tables 3 (50% WYDF), 4 (75 % WYDF) and 5 (90% WYDF) we can clearly see that other than GA all the other EA gave either the performance which is at par with LINGO or is better than it. For 50% WYF, PSO gave the best result with average net benefit as 16518.5 in comparisons to 16513.1 as obtained by LINGO which is an improvement of 0.032691%. The performance of PSO is followed closely by EP, which gave a net benefit of Rs 16517.6, an improvement of 0.027244. DE gave a net benefit of Rs. 16513.1 which is same as that of LINGO. GA gave the worst performance under the given parameter settings. For 75% WYDF, DE, PSO and EP converged to the same value of 16503.3, which is an improvement of 0.328965% in comparison to the 16449.01, the net benefit obtained by LINGO. GA once again did not give very good results in comparison to other algorithms. Finally for 90% WYDF, DE, PSO and EP converged to a net benefit of Rs 15319.1 in comparison to the net benefit of Rs 15264.74 as obtained by LINGO, which is an improvement of 0.359031%. Graphical representation of objective function values for four EA and LINGO is given in Figure 3. Distributions of crops with respect to the area as obtained by DE, PSO, EP, GA and LINGO for 50%, 75% and 90% WYDF are given in Figure 4, Figure 5 and Figure 6 respectively.

In Table 6, we give the comparison of EA with each other in terms of best, worst and average fitness function values, number of generations needed to reach to the optimal solution, time taken and standard deviation. From Table 6, we can clearly see that in terms of consistency of solution, DE gave the best performance with small standard deviations in all the cases. The best and worst values of GA fluctuated the most and highest deviation was recorded for it. On comparing the time taken (in sec) we see that DE took minimum time by taking fraction of a second for reaching the optimal solution, followed by GA, whereas EP and PSO took relatively more time. Also in terms of average number of generations, DE gave the best performance for 50% and 75% WYDF, followed by PSO and EP. For 90% WYDF, PSO took 305 average numbers of generations followed by DE which took 370 average numbers of generations. The convergence graphs of average number of generations vs. objective function value for 50%, 75% and 90% WYDF with respect to DE, PSO and EP are also shown in Figures 7. In all the cases GA took more than 1000 average numbers generations to reach to the solution and hence it is not depicted in the convergence graphs.

Table 3 Results of all algorithms: 50% water year dependable flow

ITEM	DE	PSO	EP	GA	LINGO
A1	2818.71	2818.76	2858.45	5009.97	2818.639
A2	15233	15233	15232.9	14758.3	15233
A3	7109	7109	6970.71	5537.59	7109
A4	8124	8124	8124	7718.13	8124
A5	12186	12186	12186	2956.97	12186.99
A6	4433.27	4433.36	4445.71	850.346	4433.273
A7	15233	15233	15233	15614.2	15233
A8	14273.1	14283.7	14283.6	13965.9	14273.09
Z	16513.1	**16518.5**	16517.6	16000.2	16513.1

Table 4 Results of all algorithms: 75% water year dependable flow

ITEM	DE	PSO	EP	GA	LINGO
A1	0.001616	0	0.116411	1986.13	0
A2	15233	15233	15232.9	15148.9	15233
A3	5034.2	5034.29	5035.25	1232.31	5033.97
A4	8124	8124	8123.99	5460.96	8124
A5	12186	12186	12186	7155.7	12186.99
A6	5193.36	5193.54	5192.15	2948.6	5193.36
A7	16000	16000	16000	15933.6	15876.04
A8	14624.4	14624.4	14624.4	14570.1	14592.21
Z	**16503.3**	**16503.3**	**16503.3**	15731.2	16449.01

Table 5 Results of all algorithms: 90% water year dependable flow

ITEM	DE	PSO	EP	GA	LINGO
A1	0.000198	0	0	128.783	0
A2	13125.4	13125.7	13125.7	12634.3	13125.35
A3	0.000444	0	0	167.924	0
A4	8124	8124	8123.99	7088.14	8124
A5	0.000233	0	0	586.484	0
A6	927.386	927.889	927.896	683.921	927.3857
A7	16000	16000	16000	15699.2	15876.04
A8	14624.4	14624.4	14624.4	13507.6	14592.21
Z	**15319.1**	**15319.1**	**15319.1**	14455	15264.74

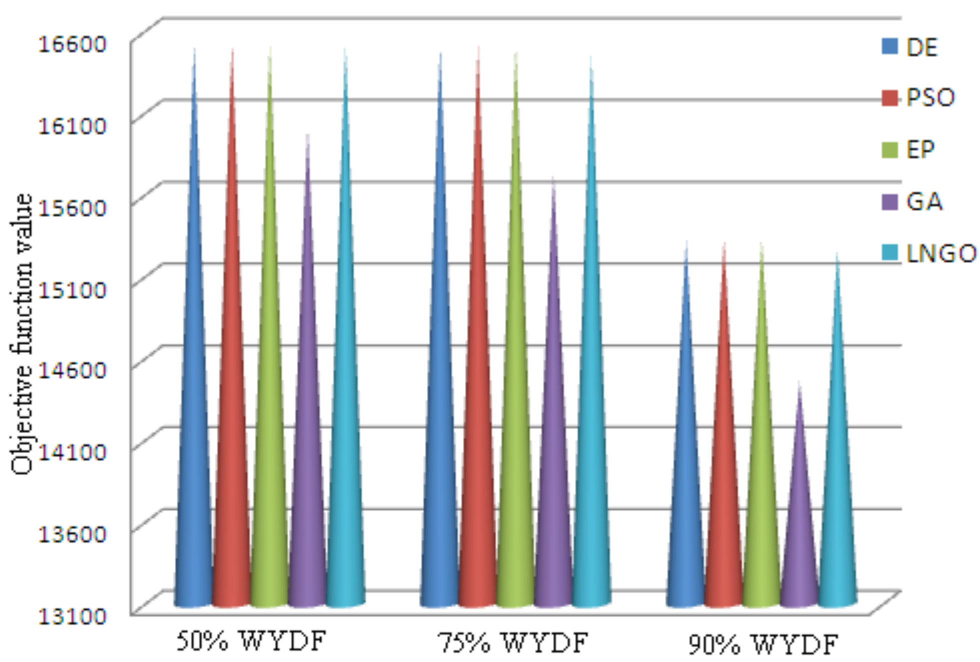

Figure 3 Comparison of objective function values for 50%, 75% and 90% WYDF as obtained by DE, PSO, EP, GA and LINGO

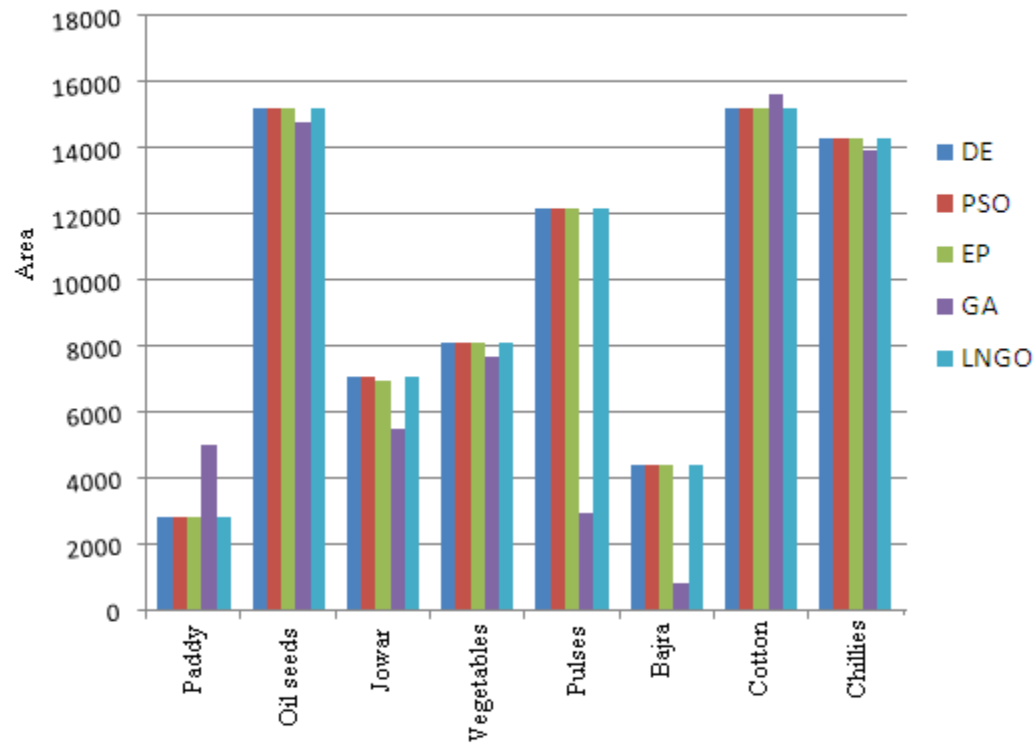

Figure 4 Distributions of crops with respect to the area as obtained by DE, PSO, EP, GA and LINGO for 50% WYDF

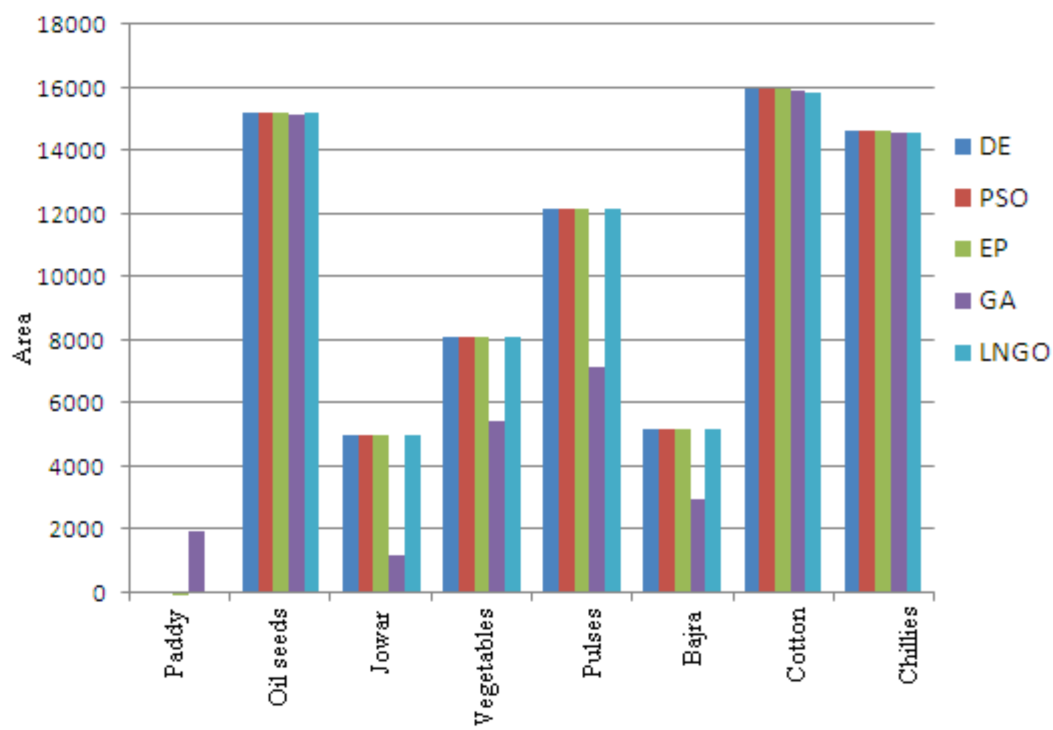

Figure 5 Distributions of crops with respect to the area as obtained by DE, PSO, EP, GA and LINGO for 75% WYDF

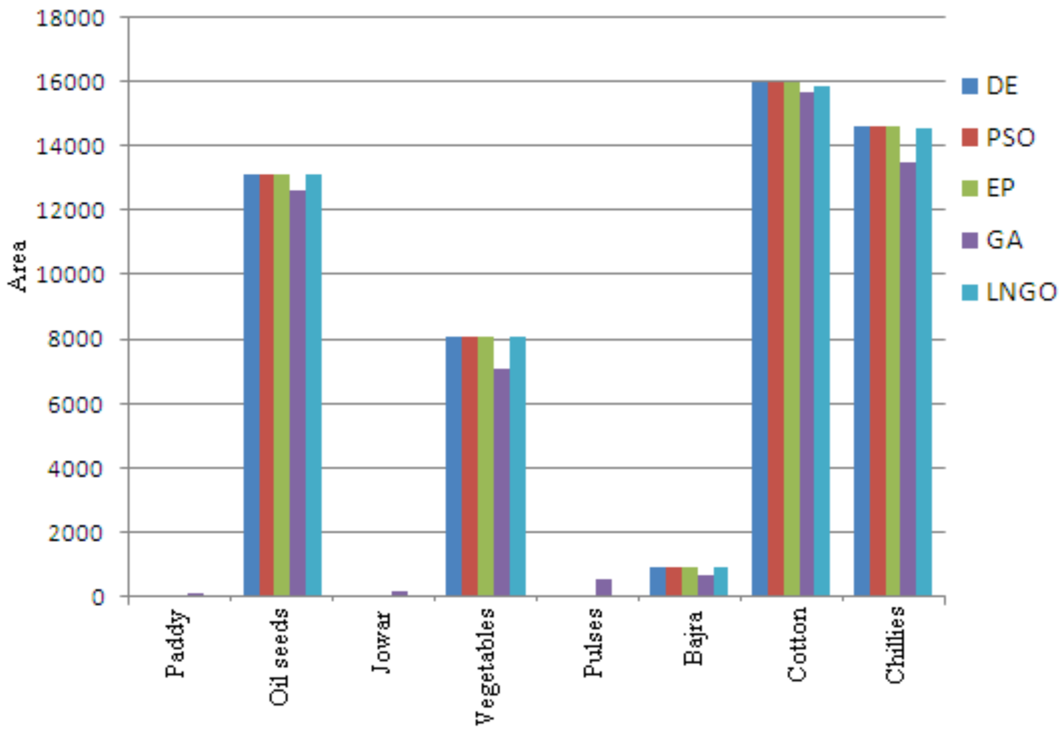

Figure 6 Distributions of crops with respect to the area as obtained by DE, PSO, EP, GA and LINGO for 90% WYDF

Table 6 Comparison Results of DE, PSO, EP and GA

50% Water Year Dependable Flow				
	DE	PSO	EP	GA
Best	16513.1	**16518.5**	16517.6	16000.2
Average	16513.1	16508.6	16507.4	15243.4
Worst	16513.1	16495.3	16478.6	14711.4
Stddev	**1.07305e-011**	12.5298	9.10771	273.106
Avg. no. of Gne.	**447**	730	874	1000^{+}
Average CPU time (sec)	**0.12**	4.48	4.78	1.2
75% Water Year Dependable Flow				
Best	**16503.3**	**16503.3**	**16503.3**	15731.2
Average	16503.3	16503.3	16486.3	15122.4
Worst	16503.3	16503.3	16378.2	14462.2
Stddev	**1.00292e-011**	5.78547e-008	23.7463	295.602
Avg. no. of Gne.	**403**	563	661	1000^{+}
Average CPU time (sec)	**0.1**	4.36	4.72	1.36
90% Water Year Dependable Flow				
Best	**15319.1**	**15319.1**	**15319.1**	**14455**
Average	15319.1	15319.1	15316	12836.9
Worst	15319.1	15319.1	15317.7	11852.1
Stddev	2.25244e-005	**8.92778e-008**	10.2506	736.197
Avg. no. of Gne.	370	**305**	585	1000^{+}
Average CPU time (sec)	**0.02**	4.32	4.58	1.1

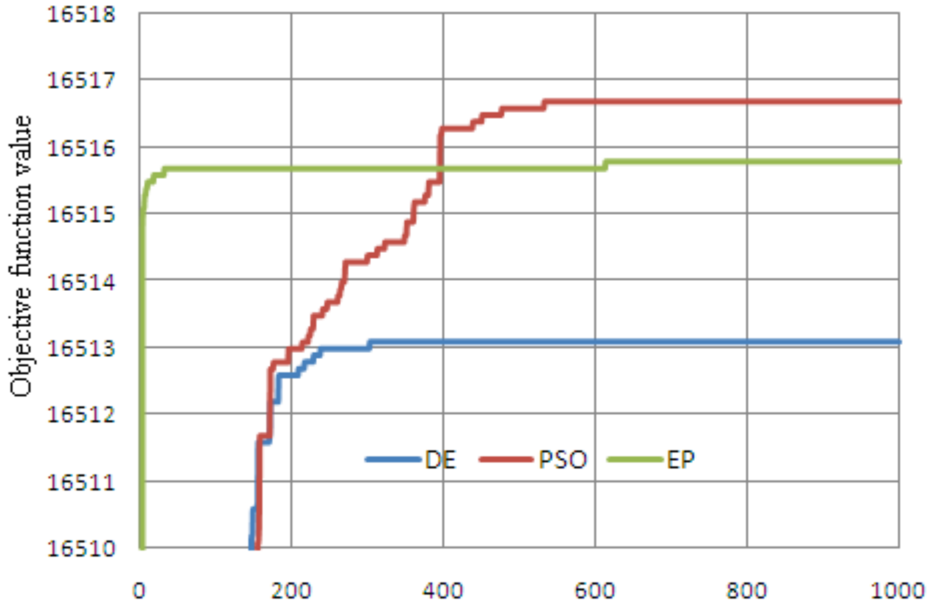

Figure 7 (a)

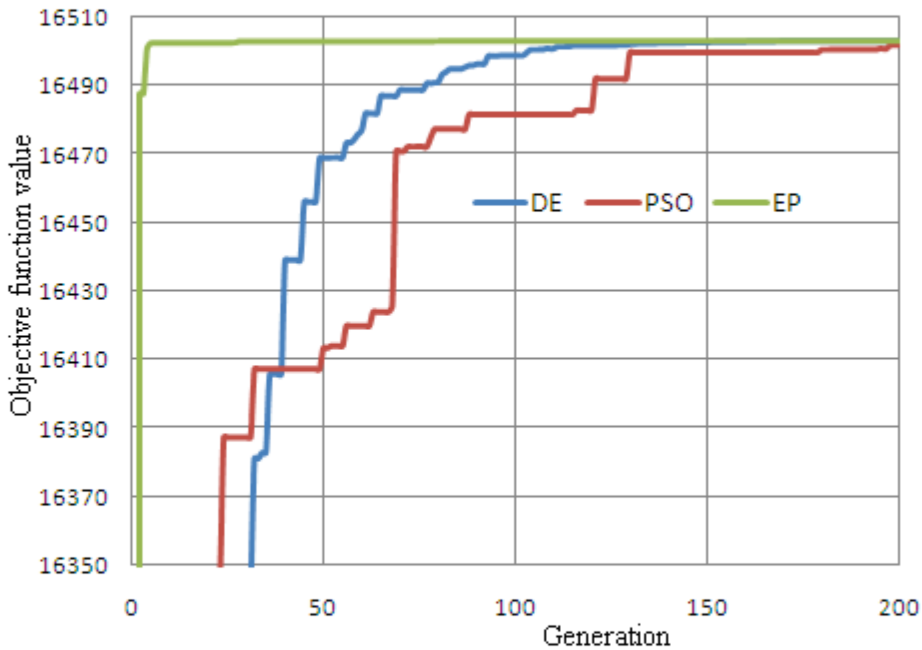

Figure 7 (b)

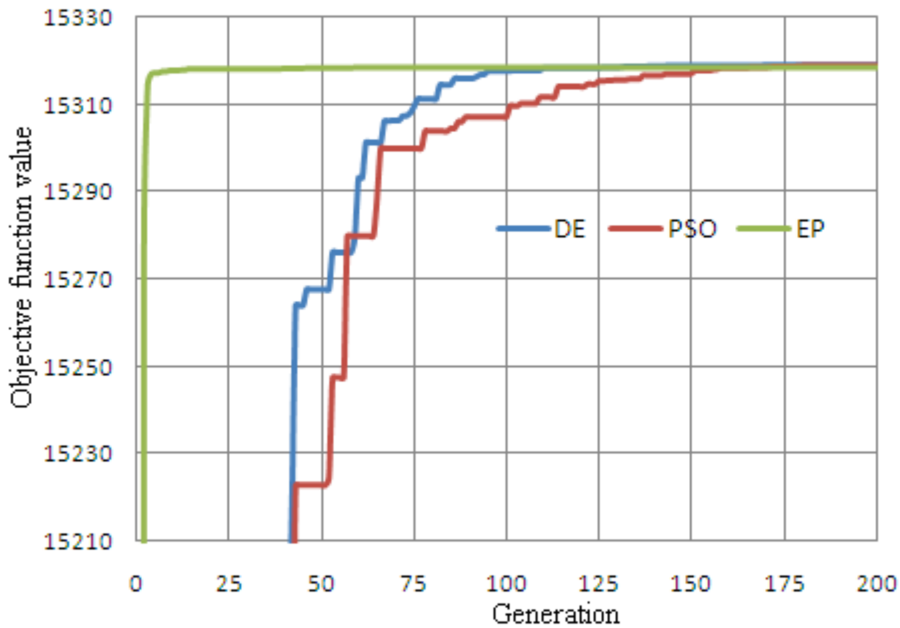

Figure 7 (c)

Figure 7 Convergence graph for objective function value vs. number of generations as obtained by DE, PSO and EP (a) 50% WYDF (b) 75% WYDF (c) 90% WYDF

7. Conclusion

The present article deals with developing an optimal crop plan model for the command area of PAV link project, Kerela, India. The mathematical model of the problem is linear in nature subject to various constraints. For optimal releases made from reservoir using a multi-reservoir operation model, optimal crop plans are developed under adequate, normal and limited irrigation water defined by 50 percent, 75 percent and 90 percent water year dependable flows, respectively. The optimization model is solved using four popular Evolutionary Algorithms; Genetic algorithm, Particle Swarm Optimization, Differential Evolution and Evolutionary Programming and also with LINGO, a software commonly used for solving LPP models. The performance of EA is compared with the performance of LINGO and also with each other. Simulation results show that PSO, DE and EP gave a better or at par performance within a satisfactory time frame in comparison to LINGO. Surprisingly GA, which has been most frequently advocated for solving such types of problems didn't give satisfactory results in comparison to LINGO and other EA. However we are making further investigations on the 'not so good performance' of GA. Among the remaining EA i.e. DE, PSO and EP none of the algorithm can be called a clear winner, but considering the consistency of performance and time duration we may say that DE gave slightly better results under the given parameter settings. Such types of studies are very beneficial for agriculture dependent country like India and can be extended further for solving more complex models.

8. References

[1] Windsor JS, Chow VT. Model of Farm Irrigation in Humid Areas. J. of Irrig. Drain. Div., ASCE, 97 (IR3), 369-385, 1971.

[2] Rogers P, Smith DV. The Integrated Use of Ground and Surface Water in Irrigation Projects Planning. American Journal of Agricultural Economics 52, 13-14, 1970.

[3] Lakshminarayan V, Rajagopalan SP. Optimal Cropping Pattern in a River Basin. Journal of Irrigation and Drainage, ASCE, 103, 53-70, 1977.

[4] Matanga G.B, Mariño MA. Irrigation Planning 1. Cropping Pattern. Water Resources Research, 15 (3), 672-678, 1979.

[5] Chávez-Morales J, Mariño MA and Holzapfel EA. Planning Model of Irrigation District. Journal of Irrigation and Drainage Engineering. ASCE, 113 (4), 549-564, 1979.

[6] Maji CC, Heady EO. Inter Temporal Allocation of Irrigation Water in the Mayurakshi Project (India): An Application of Chance Constrained Linear Programming. Water Resources Research, 14 (2), 190-196, 1978.

[7] Raju KS, Kumar DN. Irrigation Planning Using Genetic Algorithms. Water Resources Management, **18**, 163 - 176, 2004.

[8] Kuo, SF, Merkley GP and Liu CW. Decision support for irrigation project planning using a genetic algorithm. Agriculture and Water Management 45, 243–266, 2000.

[9] Raju KS, Kumar DN. Multicriteria decision making in irrigation development strategies. Journal of Agricultural Systems **62**, 117 -129, 1999.

[10] Meyer DG, Belward JA, Burrage K. Robust parameter settings of Evolutionary Algorithms for the optimization of agriculture systems models. Agriculture Systems 69 199 – 213, 2001.

[11] Bing Zhang, Shou Qi Yuan, Jian Sheng Zhang and Hong Li. Study of Corn optimization irrigation model by Genetic Algorithm. IFIP International Federation for Information Processing, 258, Computer and Computing Technologies in Ariculture Vol 1: Dialong li; (Boston Springer) 121-132, 2008.

[12] Ramakrishnan M. Systems Analysis of Multireservoirs. M.E. Dissertation, Dept. of Hydrology, Univ. of Roorkee, Roorkee, India, 1995.

[13] Feasibility Report of Pamba-Achankovil-Vaippar Link Project, NWDA, Ministry of Water Resources, New Delhi, India, 1995.

[14] Deepti Rani. Multilevel Optimization of a Water Resources System. Ph.D. Thesis, Dept. of Mathematics, Indian Inst. of Tech., Roorkee, India, 2004.

[15] Pant M, Thangaraj R, and Singh VP. Efficiency Optimization of Electric motors: A Comparative Study of Stochastic Algorithms. World Journal of Modeling and Simulation, **4**, 140 – 148, 2008.

[16] Pant M, Thangaraj R, and Abraham A. Optimization of a Kraft Pulping System: Using Particle Swarm Optimization and Differential Evolution. In Proc. of 2nd Asia Int. Conf. on Modeling and Simulation, Malaysia, IEEE Computer Society Press, USA, 637 – 641, 2008.

[17] Pant M, Thangaraj R, and Abraham A. Optimal Tuning of PI Speed Controller Using Nature Inspired Heuristics. In Proc. of 8th Int. Conf. on Intelligent System Design and Application, Italy, pp. 420 – 425, 2008.

[18] Holland JH. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. Ann Arbor, MI: University of Michigan Press.Iojpoi, 1975.

[19] Michaelwicz T. Genetic Algorithms + Data Structures = Evolutionary Programs. 2nd ed., Springer – Verlag, Berlin, 1996.

[20] Goldberg D. Genetic Algorithms in Search Optimization and Machine Learning. Addison Wesley Publishing Company, Reading, Massachutes, 1989.

[21] Deb K. An introduction to Genetic Algorithms. Sadhna 24 293 – 315, 1999.

[22] Kennedy, J. and Eberhart, R. Particle Swarm Optimization, IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, pg. IV: 1942-1948, 1995.

[23] Storn R, Price K. Differential Evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report, International Computer Science Institute, Berkley, 1995.

[24] Karaboga D, Okdem S. A simple and Global Optimization Algorithm for Engineering Problems: Differential Evolution Algorithm. Turkey Journal of Electrical Engineering 12, 53 – 60, 2004.

[25] Storn R, Price K. Differential Evolution – a simple and efficient Heuristic for global optimization over continuous spaces. Journal Global Optimization **11**, 341 – 359, 1997.

[26] Fogel DB, Owens AJ, Walsh MJ. Artificial Intelligence Through Simulated Evolution. John Wiley & Sons, New York, NY, 1966.

[27] Fogel DB. System Identification Through Simulated Evolution: A Machine Learing Approach to Modeling. Ginn Press, Needhan Heights, 1991.

[28] Fogel DB. Evolving Artificial Intelligence, PhD Thesis, University of San Diago, 1992.

[29] Fogel DB. Applying Evolutionary Programming to selected Travelling Salesman Problems. Cybernetics and Systems 24, 27 – 36, 1993.

[30] Back T, Schwefel HP. An Overview of Evolutionary algorithms for parameter Optimization. Evolutionary Computation 1, 1 – 23, 1993.

[31] Handbook on Agriculture, ICAR, Ministry of Agriculture, New Delhi, India, 1997

Appendix A

(1) Pseudo code for Genetic Algorithm

```
Begin
        Initialize the population
        For each individual calculate the fitness value.
        For i = 1 to maximum number of generations
                Do Selection, Crossover, Mutation
        End for
End.
```

(2) Pseudo code for Particle Swarm optimization

Step1: Initialization.

For each particle i in the population:

Step1.1: Initialize X[i] with Uniform distribution.

Step1.2: Initialize V[i] randomly.

Step1.3: Evaluate the objective function of X[i], and assigned the value to fitness[i].

Step1.4: Initialize P_{best}[i] with a copy of X[i].

Step1.5: Initialize Pbest_fitness[i] with a copy of fitness[i].

Step1.6: Initialize P_{gbest} with the index of the particle with the least fitness.

Step2: Repeat until stopping criterion is reached:

For each particle i:

Step 2.1: Update V[i] and X[i] according to equations (1) and (2).

Step2.2: Evaluate fitness[i].

Step2.3: If fitness[i] < Pbest_fitness[i] then P_{best}[i] =X[i], Pbest_fitness[i] =fitness[i].

Step2.4: Update P_{gbest} by the particle with current least fitness among the population.

(3) Pseudo code for Differential Evolution

```
Initialize the population
Calculate the fitness value for each particle
Do
For i = 1 to number of particles
        Do mutation, Crossover and Selection
End for.
Until stopping criteria is reached.
```

(4) Pseudo code for Evolutionary Programming

```
Begin
        Initialize the population
        For each individual calculate the fitness value.
        For i = 1 to maximum number of generations
                Do Mutation
        End for
End.
```