SOLVING LAND USE MODEL WITH MATRIX REPRESENTATION USING GENETIC ALGORITHM

D. Selvi[1] and G. Velammal[2]

[1]*Department of Mathematics, Jayaraj Annapackiam College for Women (Autonomous), Periyakulam – 625 601. Theni District. Tamil Nadu. India.*

[2]*Department of Mathematics, Sri Meenakshi Govt. College for Women (Autonomous), Madurai District. Tamil Nadu. India.*

ABSTRACT

Land use model may be defined as the process of allocating different activities or uses to meet the given objective. This is a complex process, as in land use planning decisions must be made not only on what to do (selection of activities) but also on where to do it, adding a whole extra class of decision variables to the problem. Due to increasing population, and human activities on land to meet various demands, land uses are being continuously changed without a clear and logical planning with any attention to their long term environmental impacts. Genetic algorithm is an evolutionary approach for solving space layout and optimization problems. One of the most difficult problems in architectural design is space layout problem. Planning process is the main parameter for minimizing inefficiencies and unnecessary cost and maximizing the assert value and Space layout problem commonly occurs in warehouse, hotel, building floors, containers, shelves etc. In the present paper, we propose genetic algorithm using matrix representation for finding good quality solutions for land use allocation problem.

Keywords: *Genetic algorithm, Land use planning, Matrix representation.*

1. INTRODUCTION

Land use planning is the process of allocating different activities or uses (such as residential, manufacturing industries, recreational activities agriculture, industries, forest etc) to meet the given objective. This is a complex process, as in land use planning decisions must be made not only on what to do (selection of activities) but also on where to do it, adding a whole extra class of decision variables to the problem. In the past many of these problems have been handled using linear programming approaches.

However, the recent trends, such as increased involvement of stakeholders, increased complexity on decision making, spatial integrity, and use of Geographical

Information Systems make it necessary that other approaches should be tried. Land use problems have been tackled by Fuzzy logic method, Simulation models [8], the combination of multi-criteria evaluation techniques and mathematical programming [6], goal programming model[5]. Heuristic approach, such as simulated annealing [2] was also found applicable to this problem. Matthews et al. [3], [4] explored the potential of applying GA to spatially integrated land-use management problem and also developed a GA-based spatial decision support system (DSS) that allows land managers to explore their land use options and potential impacts of land use changes.

Genetic Algorithm is a powerful and broadly applicable stochastic search and optimization technique. It works with a population of *"individuals"*, each representing a possible solution to a given problem. Each individual is assigned a *"fitness score"* according to how good the solution is to the problem. The highly-fit individuals are given opportunities to *"reproduce"*, by *"cross breeding"* with other individuals in the population. This produces new individuals as *"offspring"*, which share some features taken from each *"parent"*. The least fit members of the population are less likely to get selected for reproduction, and so *"die out"*. A whole new population of possible solutions is thus produced by selecting the best individuals from the current "generation", and mating them to produce a new set of individuals. This new generation contains a higher proportion of the characteristics possessed by the good members of the previous generation. In this way, over many generations, good characteristics are spread throughout the population.

In the present paper, we propose novel genetic algorithm using matrix representation for finding good quality solutions for land use allocation problem.

2. MATHEMATICAL MODEL FOR LAND ALLOCATION

Stewart [7] formulates the land use allocation problem as follows: To express the land use map in terms of $R \times C \times K$ binary variables X_{rck}, such that $x_{rck} = 1$ *if* $u_{rc} = k$. *and* $x_{rck} = 0$ otherwise. Let u_{rc} in rows be the land use allocated to the cell in row r and column c of the grid. Let us suppose that possible land uses are labeled from 1 to K. A land use map is an allocation of a land to every grid cell in the region, and the aim is to identify the land use map which best achieves the decision maker's objectives.

With this formulation, it is recognized that the selection of a land use map is an integer programming problem involving $R \times C \times K$ binary variables. If the problem is solved explicitly in terms of the x_{rck} then by definition we would require $\sum_{k=1}^{K} x_{rck} = 1$ for each grid cell (r,c). Typically, additional land use restrictions of the form: $\lambda_k \le N_K \le \mu_K$ where $N_K = \sum_{r=1}^{R}\sum_{c=1}^{C} x_{rck}$, (i.e.) the number of cells allocated to land use k, may apply for some or all land uses.

2.1. We Model the land use problem as follows

Suppose landscape can be represented by a two-dimensional grid of cells arranged into I rows and J columns. Let $u_{ij} = k$ where $(k = 1,2,....K)$ be the use allocated to the cell (i, j). Let P_{ijk} be the cost factor for allocating k^{th} use in the plot (i, j), and V_{ijk} be the decision variable. Out of I*J*K variables, we have to select I×J variables which minimize the objective function $Z = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} P_{ijk} V_{ijk}$

Subject to the constraints

$$\sum_{i=1}^{I}\sum_{j=1}^{J} V_{ijk} = N_K \qquad \text{for } k = 1, 2, \ldots\ldots\ldots K \qquad \ldots (E_1)$$

$$\sum_{k=1}^{K} V_{ijk} = 1 \qquad \text{for } i = 1, 2, \ldots\ldots\ldots I$$

$$\text{for } j = 1, 2, \ldots\ldots\ldots J \qquad \ldots (E_2)$$

$$V_{ijk} = \begin{cases} 1, & \text{if the grid cell } (i, j) \text{ is allocated to use } k \\ 0, & \text{otherwise} \end{cases}$$

3. GENETIC ALGORITHM FOR LAND USE PLANNING

The outline of the proposed algorithm is as follows:

ALGORITHM

Step 1: Initial population of M, I × J matrix is created randomly.

Step 2: The fitness values of all the members of the current population are evaluated.

Step 3: The M members of the next generation are generated as follows:

(i) M_1 elite members of the current generation are retained. That is M members with best M_1 fitness values are selected.

(ii) The crossover percentage p is fixed. Then $M_2=p(M-M_1)$ members a generated using the crossover operation. For this purpose M_2 pairs members are selected from the current population. From each pair of pare solutions one offspring solution is created using a crossover operator.

(iii) $M_3= (1-p)\ (M-M_1)$ members are created using mutation. For this purpose members of the current population are selected. From each of these members a new solution is created by applying a mutation operator.

Hence altogether $M_1+M_2+M_3=M$ members of the next generation (or the n population) are generated.

Step 4: Step 2 and Step3 are repeated R times.

4. REPRESENTATION OF FEASIBLE SOLUTIONS

The feasible solution of the land use plan is represented by matrix with I ro and J columns, each entry k (k = 1,2,....,K) in the grid cell denotes the use (facili which is allocated in the cell (i, j).

For example

Suppose the land has 32 plots. To allocate these 32 plots to 5 different u (Residential, shopping, Hospital, School and Industry) where N_1 = 18, N_2 = 5, N_3 = N_4 = 2, N_5 = 4 we represent these plot as 4*8 matrix. Here N_k denotes the numbe allotment to the use (facility) k.

2	4	5	1	1		1	1
	1	1	2	1	1	5	1
2	1	4	1	5		1	1
1	5	2	1	1	1	2	1

Here Cells (1,1), (2,4), (3,1), (4,3) and (4,7) are allocated to the use2 (Shopping), cells (1,6), (2,1), and (3,6) are allocated to the use3 (Hospital), cells (1,2) and (3,3) are allocated to the use4 (School) cells (1,3), (2,7), (3,5) and (4,2) are allocated to the use5 (Industry) and the remaining cells are allocated to use1 (Residential).

In MATLAB Genetic algorithm tool, we must represent the initial solution as a vector. In general, feasible solution of $m \times n$ plot can be represented as a vector of length mn $(x_1\ x_2\ \ldots\ldots\ldots x_n\ x_{n+1}\ldots\ldots\ldots\ldots x_{2n}\ldots\ldots\ldots\ldots x_{mn})$.

Where

$X_{0.n+1} = x_1$ is the use allocated to $(1,1)^{th}$ cell,

$X_{0.n+2} = x_2$ is the use allocated to $(1,2)^{th}$ cell,

...

$X_{0.n+n} = x_n$ is the use allocated to $(1,n)^{th}$ cell

$X_{1.n+1} = x_{n+1}$ is the use allocated to $(2,1)^{th}$ cell

....

$X_{1.n+n} = x_{2n}$ is the use allocated to $(2,n)^{th}$ cell

.....

$X_{(i-1).n+j}$ is the use allocated to $(i,j)^{th}$ cell

....

$X_{(m-1).n+n}$ is the use allocated to $(m, n)^{th}$ cell

4.1. CREATION OF INITIAL POPULATION

The initial population is randomly generated by the following procedure

Step 1. Take a feasible vector of length mn for the given uses.

Step 2. Generate a random number r between [1, mn].

Step 3. Generate a feasible vector using back track method.

4.2. FITNESS FUNCTION

The objective function to be minimized is $\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} P_{ijk}V_{ijk}$. We also have to ensure that the constraint set of equation (E_1) and (E_2) are to be satisfied. The proposed representation satisfied all those constrain. Hence we choose the fitness function is

$$f(X) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} P_{ijk}V_{ijk}.$$

The fitness values of the members of the current population are calculated and the total fitness $\sum f(X_i) = TF$ is calculated. $P_i = \frac{f(x_i)}{TF}$ is calculated.

The following various crossover and mutation operators are proposed to restore the feasibility of the solution.

3. SELECTION

The selection used here combine the tournament selection and elite approaches. 6 members with best fitness values are selected.

4. MUTATION OPERATORS

A child undergoes mutation according to the percentage p_m of population mutated. We use the following mutation cyclically.

4.1. ROW INTERCHANGE MUTATION

For given $m \times n$ matrix, select two numbers r1, r2 such that $1 \leq r1 < r2 \leq m$. Then row r1 and row r2 are interchanged. Suppose first and third row are selected. Interchange r1 and r2, the resultant matrix is a feasible solution.

Parent A=

1	4	5	2	1	3	1	1
3	1	1	2	1	1	5	1
2	1	4	1	5	3	1	1
1	5	2	1	1	1	2	1

↓

Child C =

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3	1	[illegible]
3	1	1	2	1	1	5	1
1	4	5	2	1	3	1	1
1	5	2	1	1	1	2	1

.4.2. COLUMN INTERCHANGE MUTATION

For given $m \times n$ matrix, select two columns c1, c2 such that $1 \leq c1 < c2 \leq n$. Then olumn c1 and c2 are interchanged. Suppose second and fourth columns are selected. nterchange c1 and c2, the resultant matrix is a legal solution.

.4.3. REFLECTION MUTATION

The given $m \times n$ matrix is reflected about a line. For example: If we reflect with spect to a vertical line which is the right hand side border of the matrix we get row eflective matrix.

arent A =

1	4	5	2	1	3	1	1
3	1	1	2	1	1	5	1
2	1	4	1	5	3	1	1
1	5	2	1	1	1	2	1

⇓

hild C =

1	1	3	1	2	5	4	1
1	5	1	1	2	1	1	3
1	1	3	5	1	4	1	2
1	2	1	1	1	2	5	1

Similarly, if we reflect with respect to a horizontal line below the matrix we get lumn reflective matrix.

SWAP REPAIR CROSSOVER

This type of crossover is accomplished by selecting two parent solutions and ndomly selects a position. Suppose it is in row r and column c. Let the number at this osition in A be p. Let the number in the corresponding position in B be q .Then swap p nd q in A. To get feasibility, change q in p in the neighborhood cell.

ol.5, No.2, June 2018 ISSN: 2347-9868

For example

Suppose the cell in the second row, fourth column in A and B is selected. Then p = 2, and q = 1. In the parent A swap 2 and 1.We get infeasible offspring. To regain the feasibility, use repair operator and change 1 by 2 in the neighborhood cell.

6. IMPLEMENTATION OF THE ALGORITHM

We implemented and tested two algorithms:

(i) **GA1-Swaprepair crossover & cyclic mutation**

(ii) **GA2-No crossover & cyclic mutation**

6.1. COMPUTATIONAL RESULTS

We experimentally implemented the algorithms by varying the crossover probability. In all cases the cyclic mutation was applied (i.e. row interchange operation mutation was applied in one generation, column interchange mutation in the next generation and reflexive in the next generation and so on in a cyclic manner).

We selected test problems whose cost matrix was randomly created using normal distribution. On each test problem, each of the two genetic algorithms was run ten times upto 100 generations in 20 sec. We noted the best value in the block of ten runs. Comparison of performance of GA1 and GA2 with various crossover probabilities is given below.

Comparison of performance of GA1 with various crossover probabilities

(Best solution out of 10 runs were noted)

Problem Size	Problem No.	**GA1-** ***Swap Repair Crossover*** **& Cyclic Mutation**			
		$p_c = 0.2$	$p_c = 0.4$	$p_c = 0.6$	$p_c = 0.8$
3×8×7	01	1359.9033	1289.3938	1332.4084	1319.963
	02	4483.3496	4456.6541	4475.1455	4555.3395
4×6×5	03	625.3096	611.8072	660.1353	622.1584
	04	2042.9384	2042.9384	2083.1602	2073.1885
5×10×5	05	989	973	867	1005
	05	7634.695	7192.8273	7631.7307	7503.9369
6×4×5	07	646.532	589.7905	589.5987	678.367
	08	5172.4615	5166.7549	5204.2014	5257.3727

Problem Size	Problem No.	GA1- *Swap Repair Crossover*& Cyclic Mutation			
		$p_c = 0.2$	$p_c = 0.4$	$p_c = 0.6$	$p_c = 0.8$
4×5×4	09	3657.9264	3657.9264	3753.8912	3753.8912
	10	2152.7904	2115.0727	2117.7375	2118.4748
7×10×8	11	2860.466	2621.7173	2696.6040	2884.773
	12	3229.5523	3037.1891	3301.9328	3054.8833
AVERAGE		2904.5770	2812.9222226	2892.79555555	2902.2790

Comparison of performance of GA2 with various crossover probabilities
(Best solution out of 10 runs were noted)

Problem Size	Problem No.	GA2- No *Crossover* & Cyclic Mutation			
		$p_c = 0.2$	$p_c = 0.4$	$p_c = 0.6$	$p_c = 0.8$
3x8x7	01	1394.5257	1462.2025	1442.3672	1423.9974
	02	6216.5195	6576.3282	6103.2444	6622.3037
4x6x5	03	774.9046	869.0568	819.509	926.8693
	04	2958.1135	2467.7648	3338.3824	3092.0559
5x10x5	05	1291	1256	1238	1303
	05	13808.8581	13453.3318	14742.3517	14487.6058
6x4x5	07	836.9662	814.3808	821.6852	867.9774
	08	6385.4949	6314.1675	6982.1569	6604.5178
4x5x4	09	4291.8838	4136.0322	4697.0212	3956.6218
	10	2207.3324	2202.9281	2212.5632	2228.2513
7x10x8	11	3452.725	3738.2553	3539.8785	40622.1462
	12	3962.6589	3952.7749	3960.4194	4137.6427
Average		**3965.0819**	**3936.9352**	**4158.1316**	**7189.4158**

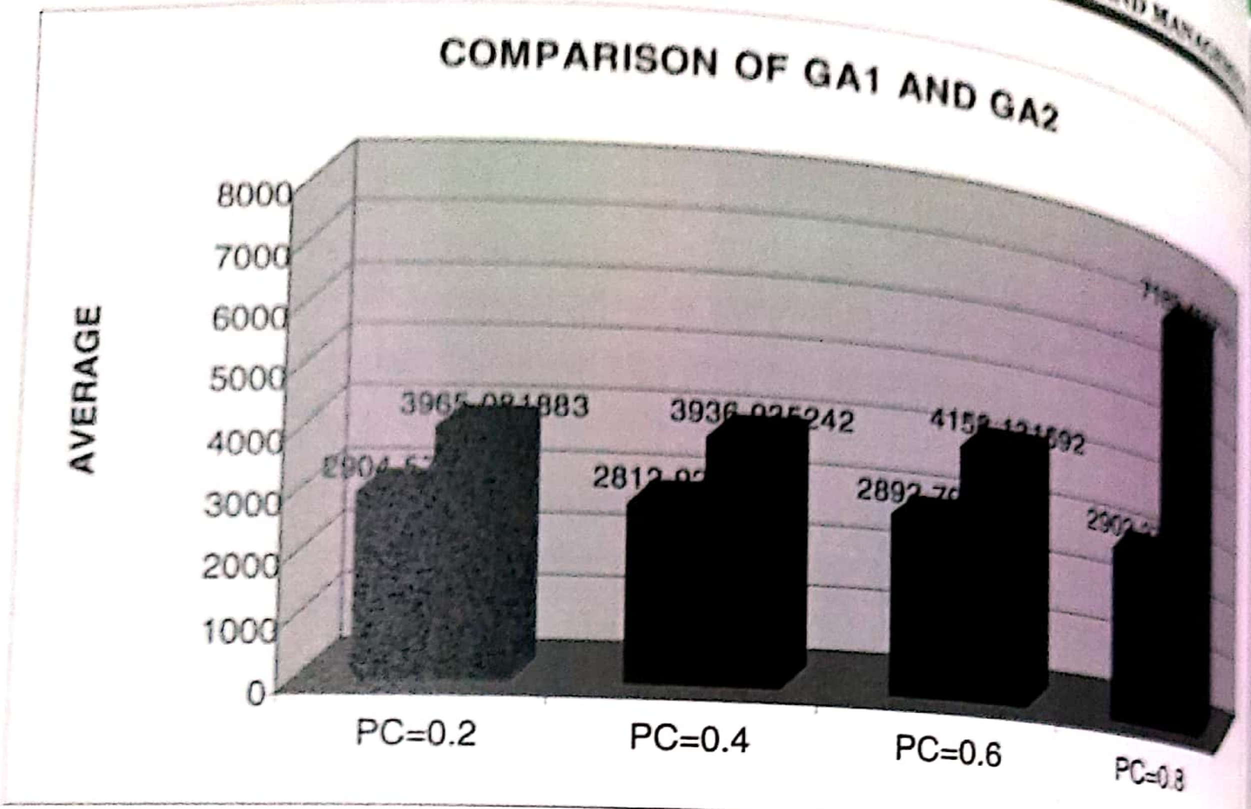

Figure 6.3

(Comparison of performance of GA1 and GA2 with various crossover probabilities)

We found that algorithm GA1 (Swap repair crossover &Cyclic mutation) produce better result than GA2 (No crossover &Cyclic mutation) for all the test problems in all crossover probability p_c=0.2, pc=0.4, pc=0.6, pc=0.8. Also we found that for the crossover probability p_c=0.4, GA1 produces better result.

. CONCLUSION

Genetic Algorithm for Land use model with matrix representation has been designed, implemented and tested. We have shown how careful selection of crossover probability and tuning of search parameters can yield superior results for the Genetic Algorithm. The approach is tested on a set of problems with different sizes. The computational results show that the algorithm produced good quality solutions in all instances tested. In the present work we attempted to solve single objective land use allocation problem. In future we may try to develop genetic algorithm for the land use allocation problem with multiple objectives and generalize the two dimensional land use problem to three dimensional allocation problem for Multi-Storey building.

REFERENCES

1. Goldberg, David E., "Genetic Algorithms in Search Optimization and Machine Learning", Pearson Education Asia Pte Ltd, 2000.
2. Lockwood C., Moore T., "Harvest scheduling with spatial constraints: a simulated annealing approach", Canadian Journal of Forest Resources 23, pp. 468-78, 1993.
3. Matthews K., "Applying genetic algorithms to multi-objective land-use planning", Ph.D thesis, Robert Gordon University [Online]. Available: http://www.mluri.sari.ac.uk/LADSS/papers/kmatth-thesis.pdf, 2001.
4. Matthews K. B., Craw S., Sibbald A. R., Mackenzie I. and Elder S., "Applying genetic algorithms to multiobjective land use planning", Proceedings of the Genetic and Evolutionary Computation Conference, pp. 613-620, 2000.
5. Oliveira F., Patias N. M. & Sanquetta C. R. "Goal programming in a planning problem", Applied Mathematics and Computation, 140, pp. 165-178, 2003.
6. Ridgley M. A. and Heil G. W., "Multicriteria planning of protected-area buffer zones: an application to Mexico's Izta-Popo national park In: Multicriteria Analysis for Land-Use Management", Dordrecht, Kluwer Academic Publishers, pp. 293-309, 1998.
7. Stewart T.J., Ron Janssen, Marjan Van Herwijnen, "A genetic algorithm approach to multiobjective land use planning", science direct, Computer and Operation Research 31, pp. 2293-2313, 2004.
8. Wu F and Webster C.J., "Simulation of land development through the integration of cellular automata and multicriteria evaluation", Environment and Planning B: Planning and Design, 25, pp. 103-126, 1998.