American Journal of Applied Sciences 6 (5): 1021-1027, 2009
ISSN 1546-9239

Integrating Gene Expression Programming and Geographic Information Systems for Solving a Multi Site Land Use Allocation Problem

Khalid A. Eldrandaly
Department of Information Systems,
Faculty of Computers and Informatics, Zagazig University, Egypt

Abstract: Problem statement: Land use planning may be defined as the process of allocating different activities or uses to specific units of area within a region. Multi sites Land Use Allocation Problems (MLUA) refer to the problem of allocating more than one land use type in an area. MLUA problem is one of the truly NP Complete (combinatorial optimization) problems. **Approach:** To cope with this type of problems, intelligent techniques such as genetic algorithms and simulated annealing, have been used. In this study a new approach for solving MLUA problems was proposed by integrating Gene Expression Programming (GEP) and GIS. The feasibility of the proposed approach in solving MLUA problems was checked using a fictive case study. **Results:** The results indicated clearly that the proposed approach gives good and satisfactory results. **Conclusion/Recommendation:** Integrating GIS and GEP is a promising and efficient approach for solving MLUA problems. This research focused on minimizing the development costs and maximizing the compactness of the allocated land use. The optimization model can be extended in the future to maximize also the spatial contiguity of the allocated land use.

Key words: Multi site land use allocation, GIS, gene expression programming, SDSS

INTRODUCTION

Land use planning may be defined as the process of allocating different activities or uses such as agriculture, manufacturing industries, recreational activities or conservation to specific units of area within a region[14]. Land use planning is a special allocation problem, where the planner, by manipulating the proportions and locations of land uses, seeks to satisfy one or more goals. Land use planning is a potentially challenging search and optimization task, as the planner must frequently take into account complex non-linear interactions between parcels of land allocated to particular land uses[11]. In these circumstances, land use allocation must try to reconcile multiple conflicting interests as rationally and transparently as possible[4], which, among other things, involves evaluating land units not only with regard to their suitability for competing uses but also with regard to such factors as contiguity among units assigned to the same use and the compactness of the single-use land masses so created[3,5,12,13]. Multi sites Land Use Allocation Problems (MLUA) refer to the problem of allocating more than one land use type in an area. MLUA problems can be solved with optimization modeling, which uses the concept of dividing an area into cells, defining the potential land use types and searching the optimal distribution for these land uses across all cells subject to a set of criteria and constraints[1]. Depending on the size of the region and on the spatial resolution required, an enormous increase in the number of decision variables can easily result[14]. MLUA problem is one of the truly NP Complete (combinatorial optimization) problems. The computational burden on computer programs for land-use allocation, which makes exact optimization methods such as integer programming infeasible when there are more than 2000 or 3000 land units to be allocated[3], is increased by simultaneous consideration of multiple possible uses. It is, therefore, necessary to turn to heuristic algorithms capable of achieving near-best solutions in a reasonable time[13]. Intelligent techniques such as genetic algorithms and simulated annealing, have been used; see for example[1,2,6,11,13-15]. Geographic Information Systems (GIS), which are computer-based information system that enable capture, modeling, storage, retrieval, sharing, manipulation, analysis and presentation of geographically referenced data[16], can provide the input for optimization algorithms and can be used to present the results generated by these algorithms[14]. This study demonstrates how Gene Expression Programming (GEP), a recently developed AI approach, can be

Corresponding Author: Khalid A. Eldrandaly, Department of Information Systems,Faculty of Computers and Informatics, Zagazig University, Egypt

integrated with GIS for solving a modified version of the non-linear integer program land use allocation model developed by Aerts and Herwijnen[2]. A prototype GEP-based Spatial Decision Support System is developed for solving a fictive case study.

Gene expression programming: Gene expression programming, an artificial problem solver inspired in natural genotype/phenotype system, was invented by Ferreira in 1999[7] and incorporates both the simple, linear chromosomes of fixed length similar to the ones used in genetic algorithms and the ramified structures of different sizes and shapes similar to the parse trees of genetic programming. Thus, the phenotype of GEP consists of the same kind of ramified structure used in genetic programming, but the ramified structures created by GEP (expression trees) are the expression of a totally autonomous genome[9].

There are two main players in gene expression programming: the chromosomes and the Expression Trees (ETs) or programs. The expression of the genetic information's encoded in the chromosome. As in nature, the process of information decoding is called translation and this translation implies a code and a set of rules. The genetic code of gene expression programming is very simple: a one-to-one relationship between the symbols of the chromosome and the nodes they represent in the trees. The rules determine the spatial organization of nodes in the expression trees and the type of interaction between sub-ETs. Therefore, there are two languages in GEP; the language of the genes and the language of expression trees and, thanks to the simple rules that determine the structure of ETs and their interactions, it is possible to immediately infer the expression tree given the sequence of a gene and vice versa. This unequivocal bilingual notation is called Karva language. Figure 1 shows an example of expression trees and Karva language[8].

The fundamental steps of gene expression programming are schematically shown in Fig. 2. The process begins with the random generation of the chromosomes of a certain number of individuals (the initial population). Then these chromosomes are expressed and the fitness of each individual is evaluated against a set of fitness cases (also called selection environment). The individuals are then selected according to their fitness (their performance in that particular environment) to reproduce with modification, leaving progeny with new traits. These new individuals are, in their turn, subjected to the same developmental process: Expression of the genomes, confrontation of the selection environment, selection and reproduction with modification. The process is repeated for a certain number of generations or until a good solution has been found[9].

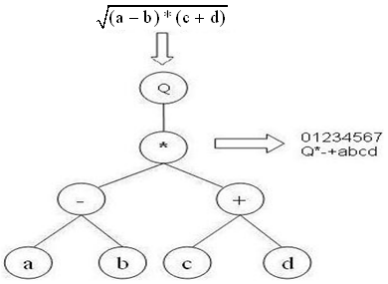


Fig. 1: An example of expression trees and Karva language[8]

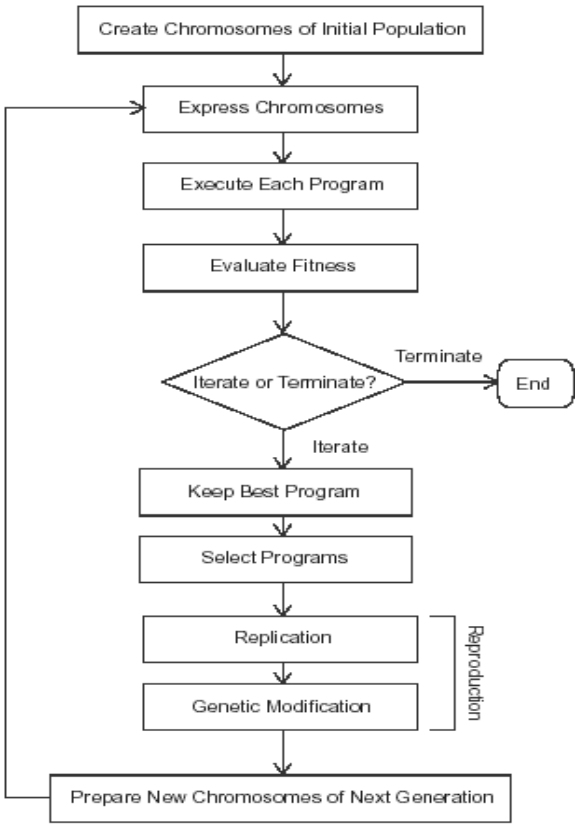


Fig. 2: The flowchart of GEP[9]

The main difference between GA, GP and GEP resides in the nature of the individuals: GA individuals are symbolic strings of fixed length (chromosomes) whereas GP individuals are trees of different sizes and shapes. GEP individuals are also (expression) trees of different sizes and shapes, encoded as strings of fixed length (chromosomes) using Karva notation. Thus, GEP retains the benefits of GAs and GP, while it overcomes some of their limitations: GAs chromosomes are easy to manipulate genetically, but they lose in functional complexity, whereas GP trees exhibit functional

complexity, but are computational expensive. Moreover, in GEP there is no such thing as an invalid expression (by contrast to GP) and the structural organization of GEP chromosomes allows the unconstrained modification of the genome. GEP genetic operators always produce valid expression. Thus the basis for the novelty of GEP resides on the revolutionary structure of GEP genes[8].

MATERIALS AND METHODS

A prototype spatial decision support system is developed by integrating GIS and GEP for solving the MLUA as shown in Fig. 3. The system is designed using ESRI's MapObjects®-Java Edition which is a powerful collection of client- and server side components that developers can use to build custom, cross-platform Geographic Information System (GIS) applications.

Preparing spatial data from GIS: GIS provides the detailed spatial data for the optimization process. The development costs and the compactness values of the allocated land uses are used to evaluate the fitness of each candidate solution. GIS is used in calculating the development cost for each land use. This cost varies with locations depending on specific physical attributes of the area, such as soil type, elevation and slope.

Encoding candidate solutions: An important step for implementing GEP is to design chromosomes according to the problem domain. In this study, the multi site land use allocation problem (MLUA) is to find out the optimal {x, y} coordinates for k different land uses within the spatial dimensions of N x M cells. Each chromosome is represented by a two-dimensional array representing the grid or the map of the area under study.

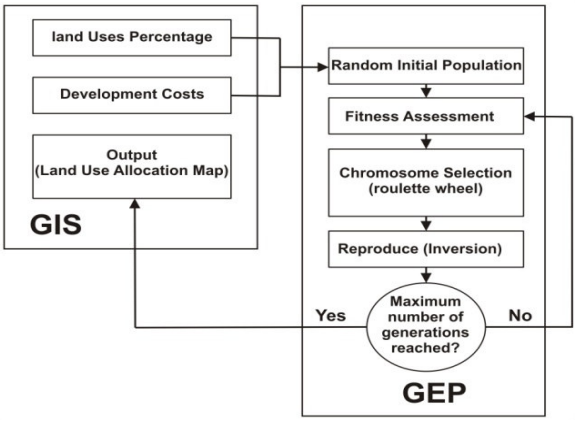


Fig. 3: Integration of GEP and GIS for MLUA

Each entry in this two dimensional array is filled with the ID of the land use supposed to occupy the corresponding location in the actual map. Figure 4 shows a chromosome with three land uses distributed randomly (their percentages are 20, 40 and 40%).

Creating initial population for the candidate solutions: An initial population is created using a random procedure. Each individual is a candidate solution. A trial and error-based algorithm is used to fill the chromosomes of the initial population with the available land uses randomly while meeting the specified percentage of each land use. The population size should be determined for creating the breeding pool. However, there is no agreement on the size of the population for optimization procedures[10]. If the size is set too small, there will not be enough individuals to find the best solution. If the size is too large, longer time is required for solving the problem.

Defining fitness functions: The evolutionary process is mainly dependent on fitness functions. The fitness functions should be used to assess the performance of each solution or individual (chromosome). It is obvious that fitness functions are crucial to the determination of the final results. There is no unique way that defines the fitness functions which is related to a problem domain[10]. In this research, the objective functions of the multi site land used allocation problem (MLUA) is minimizing the development costs and maximizing the spatial compactness of the allocated land uses. The development costs are simply the sum of the costs of all the cells of the grid. These costs vary with location because they are depending on specific physical attributes of the area, such as soil type, elevation and slope. The development costs are inversely proportional to the fitness of the chromosome. Spatial compactness merely encourages cells of equal land use to be allocated next to one another, but this may result in divided patches[3]. Compactness is a value that represents the degree at which the cells of each land use are patched together. Compactness is directly proportional to the fitness of the chromosome.

2	1	2	0	1
2	0	2	2	1
0	1	0	2	1
2	2	1	1	2
1	1	2	0	1

Fig. 4: An example of the used chromosome

The calculations of the fitness functions are based on a modified version of the non-linear integer program land use allocation model developed by Aerts and Herwijnen[2].

According to Aerts and Herwijnen[2] the optimization model can be written as follows:
Minimize:

$$w_1 \cdot \sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{M} c_{ijk} x_{ijk} - w_2 \cdot \sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{M} b_{ijk} x_{ijk}$$

Subject to:

$$\sum_{k=1}^{K} x_{ijk} = 1 \;\; \forall i = 1,...,N, j = 1,...,M \;\; x_{ijk} \in \{0,1\}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{M} x_{ijk} = T_k \;\; \forall k = 1,...,K$$

$$b_{ijk} = x_{i-1jk} + x_{i+1jk} + x_{ij-1k} + xi_{j+1k}$$

$$\forall k = 1,\ldots,K, i = 1,\ldots,N, j = 1,\ldots,M$$

$$x_{ijk} = 0$$

$$\forall k = 1,...,K, i \in \{0, N+1\}, j \in \{0, M+1\}$$

Where:

- N and M = The number of rows and columns of the grid; k is the different land uses
- X_{ijk} = A decision variable assigning land use k to cell (i, j)
- T_k = The fixed total number of cells to be allocated with land use k
- C_{ijk} = The development costs which are involved with allocating land use type k at cell (i, j)
- b_{ijk} = The number of cells neighboring cell (i, j), that have the same land use k; w_1 is the weight of the cost objective and, w_2 is the weight of the compactness objective

The above model cannot be solved using GEP because of the following two reasons:

- The model can result in negative values of the fitness function and the chromosome's fitness value has to be greater than or equal to Zero
- The model doesn't take into consideration the difference between the magnitudes of the development costs and the compactness. In most cases values of the development costs are in order of thousands or even millions. On the other hand values of the compactness are usually in order of hundreds. Thus, the contribution of compactness to the final fitness value will be much smaller than that of the development costs. In this case it will be difficult to determine the typical values for w_1 and w_2 that will equate the contribution of the two parameters, the development costs and compactness

To solve the above two problems, the costs can be calculated on a reversed scale and can then be normalized. The final modified model can be written as follows:
Maximize:

$$w_1 \cdot \frac{C_{max} - \sum_{K=1}^{K}\sum_{i=1}^{N}\sum_{J=1}^{M} C_{ijk} C_{ijk}}{C_{max} - C_{min}} - w_2 \cdot \frac{\sum_{K=1}^{K}\sum_{i=1}^{N}\sum_{J=1}^{M} b_{ijk} x_{ijk} - comp_{min}}{comp_{max} - comp_{min}}$$

Where:

- C_{max} = The maximum possible cost of a chromosome
- C_{min} = The minimum possible cost of a chromosome
- $Comp_{max}$ = The maximum possible compactness of a chromosome and
- $Comp_{min}$ = The minimum possible compactness of a chromosome

The values of C_{min}, C_{max}, $comp_{min}$ and $comp_{max}$ can be calculated approximately taking into consideration that the compactness and the development costs values of all the chromosomes in one iteration are bounded in the ranges ($comp_{min}$, $comp_{max}$) and (C_{min}, C_{max}) respectively. This condition ensures the validity of the fitness calculation of each chromosome. C_{min} and C_{max} can be calculated using sorting techniques while $comp_{min}$, $comp_{max}$ can be calculated by generating a large number of random chromosomes at the beginning of each iteration and then the smallest and the highest compactness are taken as $comp_{min}$ and $comp_{max}$ respectively. Since these four values (C_{min}, C_{max}, $comp_{min}$ and $comp_{max}$) are calculated approximately as mentioned before, the proposed GEP algorithm must be adaptive. That is, if a new chromosome has a development cost or compactness value that lies outside the calculated boundaries then the relevant variable must be updated to reflect this new finding and the fitness value of the whole population is recalculated.

Roulette wheel selection: In GEP, individuals are selected according to their fitness by roulette-wheel sampling. Each individual receives a slice of the roulette-wheel proportional to its fitness. Then the roulette is spun as many times as there are individuals in the population so that the population size is maintained from generation to generation[8].

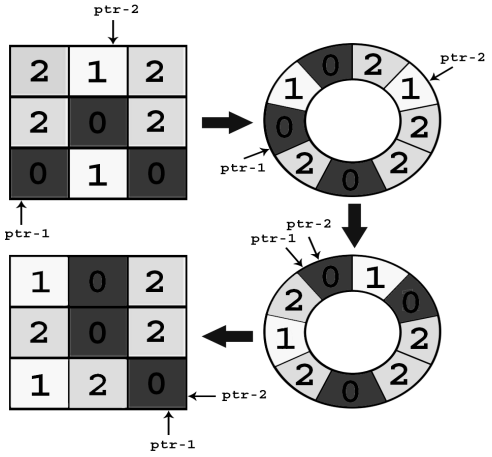


Fig. 5: Inversion process using pointers notation

Inversion operator: Inversion is the most powerful of the combinatorial-specific genetic operators, causing populations to evolve with great efficiency even if used as the only source of genetic modification. The inversion operator randomly selects the chromosome, the multigene family (MGF) to be modified, the inversion points in the MGF, then inverts the sequence between theses points. Each chromosome can only be modified once by this operator[8]. Inversion is the only used combinatorial-specific genetic operator in the proposed GEP algorithm and it is implemented using pointer notation. A pointer is just a marker that can move back and forth through the cells of a chromosome. Two pointers are placed in two random cells of the chromosome (the two dimensional matrix). The first pointer moves forward and the second moves backward until they meet. During their movement they exchange the values of the cells visited by both of them at each step. The pointer notion enables both the user and the programmer to view the two dimensional structure of the map as one dimensional cyclic structure. The major advantage of using this kind of pointers is that they give all the cells of the chromosome equal probabilities of being affected by the inversion operator, even those cells that are found at the beginning and at the end of the map, as shown in Fig. 5. Inversion is not directly applied to all the selected chromosomes but it is applied to a selected number of chromosomes according to a predefined inversion probability percentage.

Elitism: Elitism is a mechanism which ensures that the Chromosomes of the most highly fit member(s) of the population are passed on to the next generation without being altered by genetic operators (inversion).

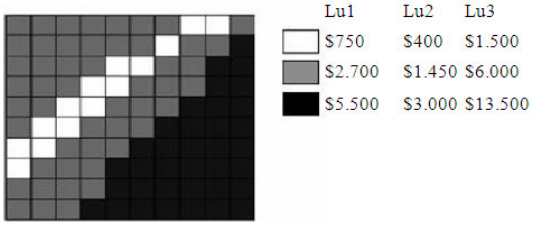


Fig. 6: Map of development costs per land use type[2]

Using elitism ensures that the minimum fitness of the population can never reduce from one generation to the next. Elitism has proved to have excellent effect in reaching good solutions faster without high risk of being trapped in local optima.

Fixed number of iterations: Except generating the initial population, all the steps are repeated for a predefined number of iterations (generations). The elite member of the final population will be our final solution.

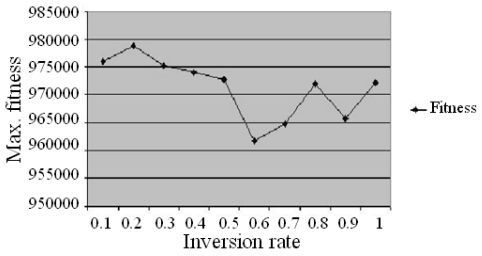
Application of the proposed approach: In order to demonstrate how our proposed approach can be used in solving MLUA problems, the fictive case study developed by Aerts and Herwijnen[2] is used. Consider a study area measuring 10×10 cells (N = M = 10) with 3 land use types Lu1, Lu2 and Lu3 (K = 3). The required spatial coverage of the three land use types is taken as 57% for Lu1, 29% for Lu2 and 14% for Lu3. The used study area with its fictitious development costs, which in a real case might be derived from physical attributes, is shown in Fig. 6.

Setting the GEP parameters: One of the major steps in preparing to use the proposed procedure is the setting of GEP parameters such as population size, generation number and inversion rate.

RESULTS AND DISSUCION

Several experiments were carried out to determine the proper values of GEP parameters for solving the fictive case study. The results of these experiments can be summarized as follows:

- Inversion rates between 10 and 30% produce good results as shown in Fig. 7
- The population size between 200 and 300 individuals will yield good results as shown in Fig. 8. A larger size of population may be required when the problem is extremely complicated
- Generation numbers between 1300 and 1500 produce good results, although better results can be obtained using higher generation numbers as shown in Fig. 9



7: Max. Fitness vs. inversion rate

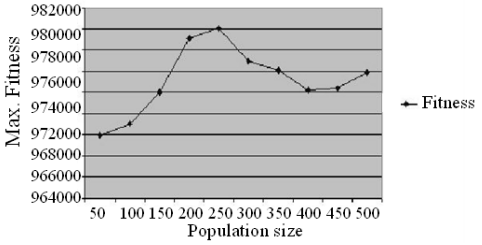


Fig. 8: Max Fitness vs. population size

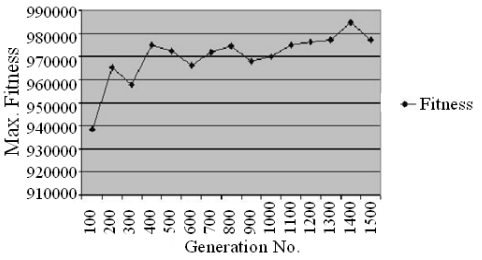


Fig. 9: Max. Fitness vs. generation number

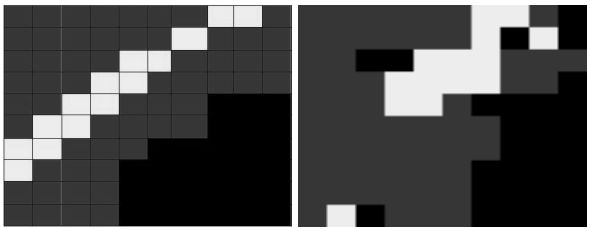


Fig. 10: Comparison of LINGO and GEP solutions at w1 = w2 (a) LINGO Solution (development cost = $ 281574) (b) GEP Solution (development cost = $ 293400)

To check the validity of the proposed GEP algorithm, the optimal solution of the case study was first obtained using LINGO 8.0 (http://www.lindo.com). Exact optimization solver such as LINGO can easily solve the modest size problems up to 10×10 cells.

Fig. 11: GEP solution (w1 = 0, w2 = 1, Generation No. = 1400, Population size = 250, Inversion rate = 0.2, Development cost = $321900, Compactness = 288)

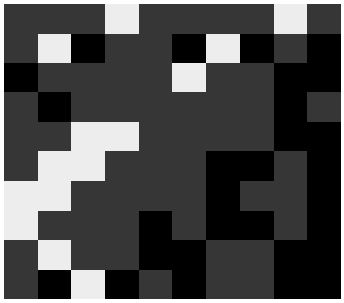


Fig. 12: GEP solution (w1 = 1, w 0 = 1, Generation No. = 1400, Population size = 250, Inversion rate = 0.2, Development cost = $289850, Compactness = 172)

Then the LINGO results have been compared with the results obtained using the proposed GEP algorithm as shown in Fig. 10. This comparison indicated clearly that the proposed approach gives good and satisfactory results. Heuristic approaches such as GEP are robust, fast and capable of solving large combinatorial problems such as MLUA, but they do not guarantee the optimal solution. Comparison of LINGO and GEP solutions at w1 = w2. Figure 11 and 12 show the solution of the case study using different weights of the development costs (w1) and compactness (w2).

CONCLUSION

MLUA problem is one of the truly NP Complete problems. To cope with this type of problems, intelligent techniques such as genetic algorithms and simulated annealing have been used. In this research, we proposed a new approach for solving MLUA problem by integrating GIS with GEP. Land use allocation model proposed by Aerts and Herwijnen[2] was modified and solved using the proposed approach.

The feasibility of the proposed approach in solving MLUA problems was checked using a fictive case study. The results obtained indicate that integrating GIS and GEP is a promising and efficient approach for solving MLUA problems. This research focused on minimizing development costs and maximizing the compactness of the allocated land use. The optimization model can be extended in the future to maximize also the spatial contiguity of the allocated land use. This work is intended as a first step toward developing an ArcGIS extension for Land Use Allocation (ArcGIS LU Analyst). The proposed extension or tool box would greatly enhance the decision making capabilities of the ArcGIS.

ACKNOWLEDGMENT

I would like to thank my colleague Haitham Gamal, Computer Science department, College of Computers, Zagazig University, for his help and support during the accomplishment of this paper.

REFERENCES

1. Aerts, J., M. Herwijnen, R. Janssen and T.J. Stewart, 2005. Evaluating spatial design techniques for solving land-use allocation problems. J. Environ. Plan. Manage., 48: 121-142. DOI**:** 10.1080/0964056042000308184
2. Aerts, J. and G. Heuvelink, 2002. Using simulating annealing for resource allocation. Int. J. Geograph. Inform. Sci., 16: 571-587. DOI: 10.1080/13658810210138751
3. Aerts, J., E. Eisinger, G. Heuvelink and T. Stewart, 2003. Using linear integer programming for multi-site land use allocation. Geograph. Anal., 35: 148-169. DOI: 10.1353/geo.2003.0001
4. Carsjens, G.J. and W. Van der Knaap, 2002. Strategic land-use allocation: Dealing with spatial relationships and fragmentation of agriculture. Landscape Urban Plan., 58: 171-179. DOI: 10.1016/S0169-2046(01)00219-5
5. Cromley, R.G. and D.M. Hanink, 2003. Scale-independent land-use allocation modeling in raster GIS. Cartograph. Geograph. Inform. Sci., 30: 343-350. DOI: 10.1559/152304003322606247
6. Duh, J.D. and D.G. Brown, 2007. Knowledge-informed Pareto simulated annealing for multi-objective spatial allocation. Comput. Environ. Urban Syst., 31: 253-281. DOI: 10.1016/j.compenvurbsys.2006.08.002
7. Ferreira, C., 2001. Gene expression programming: A new adaptive algorithm for solving problems. Complex Syst., 13: 87-129. http://www.gene-expression-programming.com/webpapers/GEP.pdf
8. Ferreira, C., 2006. Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence. 2nd Edn., Springer, pp: 3-116.
9. Ferreira, C., 2004. Gene Expression Programming and the Evolution of Computer Programs. In: Recent Developments in Biologically Inspired Computing, Castro, L.N. and F.J. Zuben (Eds.). pp: 82-103. http://www.gene-expression-programming.com/webpapers/ferreira-bic2004.pdf
10. Li, X. and A. Yeh, 2005. Integration of genetic algorithms and GIS for optimal location search. Int. J. Geograph. Inform. Sci., 19: 581-601. DOI: 10.1080/13658810500032388
11. Matthews, K.B., S. Craw, S. Elder, A.R. Sibbald, I. MacKenzie, 2000. Applying genetic algorithms to multi-objective land use planning. Proceedings of the Conference on Genetic and Evolutionary Computation, Las Vegas. http://www.macaulay.ac.uk/LADSS/papers/moga-preprint.pdf
12. Nalle, D.J., J.L. Arthur and J. Sessions, 2002. Designing compact and contiguous reserve networks with a hybrid heuristic algorithm. Forest Sci., 48: 59-68. http://www.ingentaconnect.com/content/saf/fs/2002/00000048/00000001/art00006
13. Ines, S.R., B.M. Marcos, C.M. Rafael and M.B. David, 2008. Algorithm based on simulated annealing for land-use allocation, Comput. Geosci., 34: 259-268. DOI: 10.1016/j.cageo.2007.03.014
14. Stewart, T.J., J. Ron and H. Marjan van, 2004. A genetic algorithm approach to multiobjective land use planning. Comput. Operat. Res., 31: 2293-2313. DOI: 10.1016/S0305-0548(03)00188-6
15. Sharma, S.K. and B.G. Lees, 2004. A comparison of simulated annealing and GIS based MOLA for solving the problem of multi-objective land use assessment and allocation. Proceedings of the 17th International Conference on Multiple Criteria Decision Analysis, Whistler, Canada. http://www.bus.sfu.ca/events/mcdm/MCDMProgram/Abstract1/AA58%20CF%20Sharma%20Land%20Assessment.pdf
16. Worboys, M. and M. Duckham, 2004. GIS: A Computing Perspective. 2nd Edn., CRC Press, USA., pp: 2.