http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (23): 7616-7620, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.7616.7620

Test-sheet Composition Using Cellular Genetic Algorithm with an Improved Evolutionary Rule

Ankun Huang, Dongmei Li and Kuan Lu
School of Information Science and Technology, Beijing Forestry University, 100083, Beijing, China

Abstract: For an intelligent test-sheet composition system, compared with the traditional test-sheet composition using genetic algorithm, cellular genetic algorithm can significantly improve the convergence velocity and further improve the convergence in the process. However, in the process of mutating, both the diversity index of cellular population and the possibility of escaping from local-best will be decreased. Therefore, this study proposes a cellular genetic algorithm with an improved evolutionary rule applied in the process of test-sheet composition. Experimental results show that the convergence velocity is improved and the speed of decreasing diversity index of cellular population is delayed.

Key words: Test-sheet composition, multi-objective strategy, cellular genetic algorithm, evolution rule

INDRODUCTION

The Genetic Algorithm (GA), a calculation model simulating natural selection and biological evolution process, has strong robustness and optimization function. It is first enunciated by U.S. professor J.Holland when he studied nature and artificial adaptive system (Holland, 1975). Alone with the development of genetic algorithm, it has a wide application in every walk of life, especially in function optimization and artificial intelligence.

As a multi-objective constrained optimization problem, intelligent test-sheet composition systems generally fall into the following categories (Xiao *et al*., 2012): maximum weight algorithm, analysis stochastic selection law, recollection trial method and GA. In which process, the GA is the most commonly used algorithm. However, GA has shortcomings of difficult convergence and slow optimization, which is hardly find the global optimal solution (Lin *et al*., 2012). Cellular Genetic Algorithm (CGA), a branch of the genetic algorithm, is proposed to solve this population search optimization problem which has applied in various fields such as meteorology etc.

CGA used in test-sheet composition which is a multiple constrained conditions optimization problem will effectively improve the convergence speed and the convergence (Canyurt and Hajela, 2010; Lu *et al*., 2013; Nebro *et al*., 2009; Kamkar *et al*., 2010). Prerequisite of CGA evolution is good population diversity. Due to mutation in the process of generating, population diversity takes on the tendency of descension until the steady state which is not conducive to finding a globally optimal solution and decrease the possibility of escaping local-best.

Therefore, in this paper, we propose a cellular genetic algorithm with an improved evolution rule (CEGA) simulating life complicated activities like alive, dead or competition better. For testing CEGA performance, it is applied in the process of test-sheet composition. After considerable number of tests, our experimental result show that CEGA is better than the traditional genetic algorithm both in optimization speed and global convergence.

CELLULAR AUTOMATA AND MATHEMATICAL MODELS OF INTELLGIENT TEST-SHEET CONPOSITION PROBLEM

Cellular automata principle: Cellular automata, a kinetics model discrete both in time, state and space (Kari, 2005), assembling the multidisciplinary edge domain of mathematics, physics and computer science, has a broad application prospects. Four parts, cellular, neighbors, cellular space evolution rules which are the key in algorithm, compose the basic cellular automata. By the simple evolution rules to imitate the action of life, the cellular population can generate complex behavior which can simulate the natural evolution better. It can be expressed as:

$$A = (L_r, S, N, f) \tag{1}$$

Corresponding Author: Dongmei Li, School of Information Science and Technology, Beijing Forestry University, Beijing, 100083, China

Where:

A	:	Entire set of cellular
L_r	:	Cellular space,
Subscript r	:	Dimension of space, r can take any positive integer. In this paper, r = 2
S	:	The finite set of cellular stats. Each cellular is in a certain state in every moment. S may be {1, 0}: 1 represents a "live" state of the cellular; 0 represents a "dead" state of the cellular
N	:	A set of all cellular within an area
f	:	The evolution rule, i.e., is a state transition function, denoted by:

$$f: s_i^{t+1} = f(s_i^t, s_n^t) \tag{2}$$

The mostly used evolution rule is the typical rule of "the game of life" developed by the famous mathematician Conway. In the rule of "the game of life" the cellular's state can only be "1" or "0", i.e., "life" or "death":

$$\text{If } s^t = 1, \text{ then } s^{t+1} = \begin{cases} 1 & s = 2, 3 \\ 0 & s \neq 2, 3 \end{cases} \tag{3}$$

$$\text{If } s^t = 0, \text{ then } s^{t+1} = \begin{cases} 1 & s = 3 \\ 0 & s \neq 3 \end{cases} \tag{4}$$

where:

t	:	The state of cellular at the t moments
s^{t+1}	:	The state of cellular when t = t+1
s	:	The number of neighbor cellular alive in the t moment

Mathematical model of intelligent test-sheet composition problem: The test quality offered by a test-sheet composition system depends on not only the quality of test items but also the satisfied test sheets to meet the various requirements of assessment parameters, such as the difficulty degree, the expected testing time etc. Thus, in this paper, a test-sheet consists of m questions. Each question consists of n attributes, containing the question number, exposure, time, chapter and scores, etc. Each question can be represented by an n-dimensional vector l:

$$l = (a_1, a_2, a_3, \ldots, a_n) \tag{5}$$

where, a_1 represents the title number of the question, a_2 represents difficulty, a_4 represents time, a_5 represents exposure, a_6 represents scores; therefore the structure of one test-sheet can be described like this:

$$M = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{24} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & a_{34} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & a_{m4} & \cdots & a_{mn} \end{pmatrix} \tag{6}$$

The first column of the matrix M generally represents the serial number of the question; a_{ij} (i = 1, 2, 3... m; j = 1, 2, 3..., n) represents the j-th attribute of the i-th question.

CELLULAR GENETIC ALGORITHMIC WITH AN IMROVED EVOLUTIONARY RULE

Improved evolutionary rule is the core of CEGA in this paper. Based on the same operation for each generation of sample groups, it replaces "the life of game" in CGA applied in the intelligent test-sheet composition process.

Determination of cellular space and coding scheme: A two-dimensional space where every cell distributes is used as cellular space. The neighbors of cellular adopt the model of 8-Moore according to the method proposed by Kari (2005) and Kirley *et al.* (1999); Meanwhile, the cellular located the boundary of the space uses the up and down, left and right connection which is good for improving the possibility of the population diversity by forming a torus topology structure. And the improved evolutionary rule adopts the evolution proposed by Lu *et al.* (2010):

$$\text{If } s^t = 1, \text{ then } s^{t+1} = \begin{cases} 1 & s = 1, 2, 3, 4 \\ 0 & s \neq 1, 2, 3, 4 \end{cases} \tag{7}$$

$$\text{If } s^t = 0, \text{ then } s^{t+1} = \begin{cases} 1 & s = 4, 5, 6, 7 \\ 0 & s \neq 4, 5, 6, 7 \end{cases} \tag{8}$$

Generally the binary encoding or decimal encoding is adopted as the encoding manner of the genetic algorithm. However, in the case of abounding sample groups, binary encoding will cause the long chromosome which will lead to a long period of time waste and a massive space cost. There is no doubt that the decimal encoding is the best choice.

Parameter settings: For comparing the difference between CA, CGA and CEGA in the test-sheet composition process, the same parameters are adopted in the genetic process. In order to keep the population diversity, the size of sample is identified as 625. In this study, the method of selecting the descent proposed by Liu and Liu (2010) is that the likelihood to be selected in directly proportional to the values of probability. And the crossover rate of 0.6 and mutation rate of 0.1 are adopted from the Lu and Liu (2005)' research.

Determination of the fitness function: Each question contains six attributes in test paper: question number, difficulty, chapter, time, exposure and score. the vector g is expressed as (a_1, a_2, a_3, a_4, a_5, a_6). It is assumed that a test paper contains n questions, thus producing a n*6 matrix Q just like Eq. 6:

$$Q = (a_{i1}, a_{i2}, a_{i3}, a_{i4}, a_{i5}, a_{i6}); \quad (i = 1, 2, 3,..., n) \tag{9}$$

c_i, x_i, e_i (i = 1, 2, 3,..., m, m is the number of chapter), respectively represent the attributable scores of each chapter according to users' requirements, the attributable scores of each chapter in actual papers and chapters' errors that users can tolerate.

Determining the function f1:

$$f_1 = \sum_{i=1}^{n} a_{i2}a_{i6} \tag{10}$$

Determining the function f2:

$$f_2 = \sum_{i=1}^{n} a_{i4}a_{i6} \tag{11}$$

Determining the function f3:

$$f_3 = \sum_{i=1}^{n} a_{i5}a_{i6} \tag{12}$$

Determining the function d1:

$$d_1 = f_1/100\text{-dif} \tag{13}$$

The variability dif in Eq. 13 is a difficulty level value required by the user and its value interval is 0 to 100;

Determining the function d2:

$$d_2 = f_2/100\text{-time} \tag{14}$$

The variability time in Eq. 14 represents the total time required by the user and its value interval is 0 to 100;

Determining the function d3:

$$d_3 = f_3/100\text{-p} \tag{15}$$

The variability p in Eq. 15 represents the exposure required by the user and its value interval is 0 to 100;

Determining the function d4, its value interval is 0 to 100:

$$d_4 = \sum_{i=1}^{m} h_i \tag{16}$$

Determining the function h_i:

$$h_i = \text{Max}(|\, c_i\text{-}x_i \,|\text{-}e_i, 0) \tag{17}$$

Determining the target function:

$$H = (d_1+d_2+d_3+d_4)/4 \tag{18}$$

Generally the bigger the individuals' fitness value is the better. Therefore the objective function is transformed:

$$H = 100\text{-}H \tag{19}$$

In order to maintain the diversity of the population and promote competition between similar individuals, exponential scaling method is adopted and converted into the fitness function:

$$F\text{: } F = \exp(\beta H) \tag{20}$$

The idea of using the method of simulated annealing which is proposed by Song and Liu (2008), wherein the â is commonly 0.05. The greater the F value is, the more about the strength of the copy tends to be those with maximum fitness individuals.

Steps of cellular genetic algorithm:

- **S1:** Initialization and random generation of initial population M. The amount of samples encoded in decimal encoding of population A totals to T. Select rate is Pc; crossover rate is Pm; mutation rate is Pv; The largest genetic algebra is Max
- **S2:** Define a two-dimensional array GA[I][J] as the vector of cellular space, wherein I*J = M; All individuals are randomly distributed in this space
- **S3:** Calculate the fitness value of each cellular by the function F and randomly determining the initial state of the cellular: "1" or "0"
- **S4:** Join the evolution which is expressed by Eq. 7 and 8 if use CEGA or join the evolution which is expressed by Eq. 3 and 4 if use CGA. Skip the S4 if use CA
- **S5:** Using function F to calculate the fitness value of each cellular in current generation after generating and select out samples with a certain selectivity rate Pc and add them to the set S
- **S6:** Select out individuals orderly from the set S and cross over with each other according to the crossover rate and the new one should be calculated out its fitness values by the function F. If the fitness

Table 1: The genetic algorithm's parameters

Parameters	Pv	Pm	Maximum of evolution generation (MAX)
Value	0.1	0.6	2000

Table 2: The constrained conditions

Constrained conditions	Total score	The No. of every kind ques.	The score of each chapter	Permissible error of each chapter	Difficulty degree	Sample size
Value	100	{10,5,5, 3,3}	{25,15,10, 10,20,20}	5	{55,65,75}	625

where: The property (difficulty value, time, etc.) values of every question are generated randomly

Table 3: The result of average maximum fitness value and average convergence time

Algorithm	Average maximum fitness value	Average convergence time (msec)
GA	73.5	1631
CGA	83.7326	898
CEGA	87.41	787

Table 4: Comparing difficulty of the three algorithms

Algorithm	55	65	75
GA	60.1971	61.8442	70.5975
CGA	58.3067	67.1029	72.8711
CEGA	56.0901	64.5801	74.7414

value is greater than its corresponding fitness value of the female parent, the genetic information of the female parent should be replaced by the new individual's genetic information, otherwise, do not replace

S7: Mutate with Pv and replace the original individual by adding to the cellular space

S8: If it reaches the MAX of evolution, show the individual of max fitness value in current cellular space; otherwise, go S4

COMPARISON AND ANALYSIS OF EXPERIMENTAL RESULTS

To evaluate the performance of the CEGA, two experiments have been conducted to compare performance and quality between GA, CGA, CEGA based on the test-sheet composition system. Firstly, a steady size of item banks including 560 multiple-choice questions, 560 true or false questions, 530 fill in the blank questions, 420 short answer questions and 460 comprehensive questions is used to evaluate the efficiency and fitness scores of the test-sheet composition only using one of GA, CGA or CEGA. Secondly, different difficulty degree required are measured to describe the distance required value and experimental value. We make 50 times optimization operation in each algorithm respectively.

The genetic algorithm's parameters are listed in Table 1. The constrained conditions are listed in Table 2.

The item banks having 2530 questions are used to evaluate the algorithms' efficiency and effectiveness. The effectiveness is measured by the fitness score of the function F. The result of effectiveness is shown in column 2 of Table 3, where the CEGA has more steady and generally higher fitness scores than GA and CGA. And the result of efficiency is shown in column 3 of Table 3, where the response time of the CEGA is lower than others.

In order to analyze the quality of the two sets of papers, we compare the difficulty which the user requires with the actual average difficulty. Keeping other test paper parameters unchanged, we give a different difficulty coefficient gradient for each method. From Table 4, it can be seen that the results of test sheets out of the CEGA and CGA are all with the permissible error range and the error range in CEGA is smaller than CGA.

CONCLUSIONS

An improved evolutionary rule is applied in the CGA to replace "the game of life", then, the CEGA is applied in the process of test-sheet composition in this paper. Compared with CGA, CEGA improves the possibility of escaping from local-best and the speed of convergence and keep the speed of population diversity of decreasing slower. The next step could add new parameters and change the cellular automata models to compose better test-sheet.

ACKNOWLEDGMENT

This study is supported by Beijing Forestry University Resources Sharing Course (Data Structure), Beijing Higher Education Reform (Computer Application Professional Core Courses Reform and Teaching Resources Construction Aiming at Improving Programming Design and Software Development Ability) and the National College Students' Training Programs of Innovation and Entrepreneurship (No.201310022051).

REFERENCES

Nebro, A.J., J.J. Durillo, F. Luna, B. Dorronsoro and E. Alba, 2009. A cellular genetic algorithm for multiobjective optimization. Int. J. Intell. Syst., 7: 726-746.

Canyurt, O.E. and P. Hajela, 2010. Cellular genetic algorithm technique for the multicriterion design optimization. Struct. Multidiscip. Optim., 40: 201-214.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. 8th Edn., University of Michigan Press, Ann Arbor, IM., USA., ISBN: 9780472084609, Pages: 185.

Kamkar, I., M.T. Poostchi and R.A.T. Mohammad, 2010. A cellular genetic algorithm for solving the vehicle routing problem with time windows. Adv. Intell. Soft Comput., 75: 263-270.

Kari, J., 2005. Theory of cellular automata: A survey. Theor. Comput. Sci., 334: 3-33.

Kirley, M., X.D. Li and G.D. Green, 1999. Investigation of a cellular algorithm that mimics landscape ecology. Proceedings of the 2nd Asia-Pacific Conference on Simulated Evolution and Learning, Canberra, Australia, November 24-27, 1998, Springer, Berlin, Heidelberg, pp: 90-97.

Lin, H.Y., J.M. Su and S.S. Tseng, 2012. An adaptive test sheet generation mechanism using genetic algorithm. Math. PProblems Eng., Vol. 2012. 10.1155/2012/820190

Liu, Y.F. and S.Y. Liu, 2010. Algorithm based on genetic algorithm for sudoku puzzles. Comput. Sci., 37: 225-226.

Lu, K., D.M. Li, J. Yu and J.X. Wang, 2013. An intelligent test paper construction method based on cellular genetic algorithm. Comput. Eng. Appl., 49: 57-60.

Lu, Y.H. and H. Liu, 2005. Auto-generating examination papers based on integer coding and adaptive genetic algorithm. Comput. Eng., 23: 232-233.

Lu, Y.M., M. Li and L. Li, 2010. The cellular genetic algorithm with evolutionary rule. Chin. J. Electron., 38: 1603-1607.

Song, W. and Q. Liu, 2008. Business process mining based on simulated annealing. Chin. J. Electron., 36: 135-139.

Xiao, G.X., W.C. Zhao, W. Zhu and J.H. Zheng, 2012. Iterant problems replacement method based on genetic algorithm with test paper intelligent generation. Comput. Eng., 38: 150-152.