A Genetically based Evolutionary Computing Technique based on Cellular Automata

Dr. G. Sahoo and Tapas kumar***

**** Department of Computer Science & Engineering***
Birla Institute of Technology, Mesra, Ranchi, India
***** Department of Computer Science & Information Tech.***
Lingaya's Institute of Mgt & Technology, Faridbad, India

Summary
Evolutionary Computing techniques use an explicit fitness function or simulated to derive a solution to a problem from a population of individuals, over a number of generations. The general approach which allows such techniques to be used on problems in which evaluations are so costly, which cannot be expressed formally, or which are difficult to simulate, is examined [1]. Much work has been done on combining different evolutionary computing techniques particularly the genetic algorithm and neural networks. Our motivation here is to discuss how cellular automata techniques can be involved on evolutionary computation. A study of cellular automata based evolutionary computation in genetic analysis is an inherent problem. But the key problems of genetic analysis are very sensitive in the detection of fitness cells. Evolutionary algorithms use crossover operation to combine information from pairs of solutions and selection operation to retain the best solution. Here, we consider an interactive step so as to get a maximum amount of information that can be shared for the best evaluation of individual fitness cell [7].

Key words: Fitness analysis, Genetic algorithm, CABEC Model, Cellular automata.

1. Introduction

In the various fields of science and engineering evolutionary computing is usually introduced to integrate different approaches where something is unconventional or the problem to be solved is unorthodox. Evolutionary computing is a collective name for a range of problem-solving involving natural selection and genetic inheritance properties based on principles of biological evolution. These techniques are widely applied to a variety of problems, ranging quantum computing, DNA computing, membrane computing, fuzzy computing. Three types of evolutionary computing techniques are increasingly reported in the literature recently. Further, the Evolutionary Algorithms can be divided into Evolutionary Strategies (ES) and Evolutionary Programming (EP). All of these algorithms in some way are modelled after the evolutionary processes occurring in nature. Recently, researchers have begun exploring alternative computational systems based on entirely different principles. Evolutionary computing includes a number of computing algorithms which form a class of random search algorithms in which principles of natural evolution are regarded as the rules for optimization [7, 13]. The basic concepts of evolutionary computation and how these have been adopted in various ways to simulate within an artificial system and natural computing are discussed. In continuation the main components of any evolutionary algorithm namely representation and fitness analysis, mutation, recombination, and selection have also been discussed. The objective to use an evolutionary cellular automaton in genetic algorithm is to research and review the techniques encompassed within gain thorough understanding of the principles involved in evolutionary computation and their use. A new Evolutionary computing method called Cellular Automata Based Evolutionary Computing (CABEC) model has been introduced. The aim is here to discover how evolutionary computation and cellular computing processing can be combined together for better performance. This model also can be efficiently and successfully used in specific applications of image processing.

2. Cellular Automata

The computer science field has increased expectation on technologies of computations based on the theory of Cellular Automata. In recent years there are many researchers have shown interest to analysis and design techniques for complex systems**.** Cellular automata (CA) are one of the effective methods. Although Cellular Automata (CA) was proposed firstly by Von Neumann and Ulam, from theoretical point of view, in the late 1940's. John Horton Conway's "game of life" ensures the new idea of its application in the computing field [1]. CA are henceforth considered as powerful modeling approach for complex systems in which global behavior arises from the collective effect of many locally interacting simple

Manuscript received November 5, 2007
Manuscript revised November 20, 2007

components. Subsequently several tools based on CA are proposed to provide meaningful results for real world applications. Cellular automata consist of a set of cells normally arranged in a grid shape, each of which can be in a number of states. Each cell then updates itself by a rule known as state transition rule according to its own state and the states inscribed in a prescribed neighborhood. Generally, the local connecting pattern between cells distinguishes cellular automation. During an interaction, which takes place on a purely local basis, all cells are updated by the use of local transition rule synchronously by assigning numerical values to them. The values are in ascending order and depend only on the neighborhoods considered. A cellular automaton can also be viewed as consisting of a line sites where each site depends on a finite set of possible values for updating, which takes place in discrete time steps [8]. We can define cellular automata (Fig. 1) as:

CA= (Cells, Cells Space, Cell state, Neighborhoods, Rules) and satisfy the following [11].

- CA is an ideal mathematical model
- CA is generally performed in space and time.
- CA is nothing but a discrete simulation technique.
- Elements of CA are known as Cells, which consist of a finite number of states and they are arranged to form a Lattice of some dimension.
- The future state of each cell depends on the current state of that cell and the states of its neighborhood cells [8, 14].

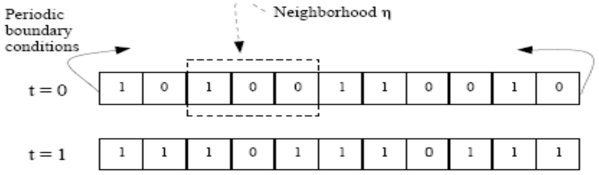


Fig 1. Block diagram of Cellular Automata

3. Genetic Algorithm

Genetic algorithms are based on evolutionary principles wherein a particular function or definition that fits the constraints of an environment survives to the next generation, and the other functions are eliminated. Genetic algorithm (GA) was developed by John Holland in the year of 1960's. GA is a method for moving from one population "chromosomes" to a new population by using a natural selection procedure. GA maintains a population of candidate solutions and this population is evolved until a competitive or near optimum solution is obtained. The idea is to efficiently find a solution to a problem in a large space of candidate solution. Standard GA is implemented where the initial position of individual is generated at random. The main advantage of a genetic algorithm is that it is able to process numerous binary strings parallel, where each binary string represents a different optimal solution to the given problem [5, 8]

A basic genetic algorithm comprises three genetic operators.

- Selection,
- Crossover
- Mutation

Genetic algorithm proceeds with an initial population of individuals which is generated either random or created as a result of some heuristic process. For every evolutionary step, known as a generation, the individuals in the current population are decoded and evaluated according to some predefined quality criterion, referred to as fitness, or fitness function. The above said procedure can be stated as [2]:

```
BEGIN
 INITIALIZE population P (f) with randomly
COMPUTE   fitness value P (f) among population
REPEAT UNTILL (TERMINATION CONDITION is satisfied)
SELECT P (f);
RECOMBINE pairs of parents;
MUTATE the resulting offspring;
EVALUATE new P (f);
 DO
END GA.
```

3.1 Fitness computation on genetic algorithm

Generally in fitness computation, individuals are selected to create a new population according to their individual correspondence fitness. Currently many selection procedures are in use, but Holland's original fitness-proportionate selection is one of the simplest techniques where individuals are selected with a probability proportional to their analogous fitness for every evolutionary step, known as a generation. Further, the individuals in the current population are decoded and elevated according to some predefined quality basis, referred to as the fitness, or fitness function. This ensures that the expected number of times an individual is chosen is relativity reciprocal to its respective performance in the population. Thus, high-fitness individuals stand a better chance of reproduction while low fitness ones are more likely to disperse [6].The roll of evaluation based function is to represent the requirements to adopt it. It forms the basis for selection and thereby quality improvement. In the problem solving approach, it represents the task to solve in the evolutionary context .Technically it is a function or procedure that assigns a quality measures to genotypes.

3.2. Existing Fitness -proportionate selection method

Genetic algorithms have enjoyed much success in a wide range of practical applications. In this application fitness computation is extremely simple since neither complex decoding nor evaluation is necessary. The method due to Holland is especially effective because he not only considered the role of mutation but also utilized genetic recombination, crossover technique. It is worth to mention here that the crossover of partial solutions greatly improves the capability of the algorithm and eventually it finds the optimum solution. As an example, we are going to enter a world of simplified genetic. Let us examine the global genetic pool of four basilosaurus belonging to this world. We will consider the "chromosomes" which encode the length of anterior members. The length of the "paw" and the length of the "fingers" are encoded by four genes: the first two encode the "paw" and the other two encode the fingers. In our representation of the genome, the circle on blue background depict the activation of a feature, the cross on green background depict its deactivation. The ideal genome (short paws and long fingers) is:

☒☒○○

The genetic pool of our population is the following one:

Subject	Genome
A	○☒☒☒
B	☒○☒☒
C	☒○☒○
D	☒☒○☒

We can see that A and B are the closest to their ancestors; they have got quite long paws and short fingers. On the contrary, D is close to the optimum, he just needs a small lengthening of his fingers. The ability to move is the main criteria of survival and reproduction. The fitness is easy to compute as we just have to give one point to each gene corresponding to the ideal. The perfect genome will then get four points. The probability of reproduction of a given subject will directly depend on this value. In our case, we get the following results:

Subject	Fitness	Reproduction Probability
A	1	1/7 = 0.143
B	1	1/7 = 0.143
C	2	2/7 = 0.286
D	3	3/7 = 0.428
Total	7	7/7=1

The reproduction pattern is the following:

Subject	Received genes	Genome	Fitness	Reproduction probability
A'	A : ○☒ D : ○☒	○☒○☒	2	2/10=0.2
B'	B : ☒○ D : ○☒	☒○○☒	2	2/10=0.2
C'	D : ☒☒ C : ☒○	☒☒☒○	3	3/10=0.3
D'	C : ☒ D : ☒○☒	☒☒○☒	3	3/10=0.3
Total			10	10/10=1

During reproduction crossovers occur at a random place center of the genome for A', B' and C', just after the first gene for D'). The link existing between the degree of adaptation and the probability of reproduction leads to a trend to the rise of the average fitness of the population. In our case, it jumps from 7 to 10. During the following cycle of reproduction, C' and D' will have a common descendant showing whole solution space can be used in the searching of best solution [12].

D’: ☒☒○+ C’: ○ = ☒☒○○

Selection is based on the survival of the fittest strategy, but the key idea is to select the better individuals of the population as in tournament selection, where the participants compete with each other to remain in the population. The most commonly used strategy to select pairs of individuals is the method of roulette-wheel selection, in which every string is assigned a slot in a simulated wheel sized in proportion to the sting's relative fitness. This ensures that highly fit sunshade a greater probability to be selected to form the next generation through crossover and mutation. After selection of the pairs of parent strings the crossover operator is applied to each of these pairs. The crossover operator involves the swapping of genetic material (bit-values) between the two parent strings. In single point crossover a bit position along the two strings is selected at random and the two parent strings exchange their genetic material as illustrated below. Parent A = al a2 a3 a4 | a5 a6,
Parent B = bl b2 b3 b4 | b5 b6
The swapping of genetic material between the two parents on either side of the selected crossover point produces the following offspring:
Offspring A' = al a2 a3 a4 | b5 b6
Offspring R' = bl b2 b3 b4 | a5 a6
The two individuals (children) resulting from each crossover operation will now be subjected to the mutation operator in the final step to forming the new generation.

The mutation operator alters one or more bit values at randomly selected locations in randomly selected strings. Mutation takes place with a certain probability, which in accordance with its biological equivalent is usually a very low probability. The mutation operator enhances the ability of the GA to find a near optimal solution to a given problem by maintaining a sufficient level of genetic variety in the population, which is needed to make sure that the entire solution space is used in the search for the best solution [10].

4. Proposed model

In this paper we have introduced a new model namely Cellular Automata Based Evolutionary Computation (CABEC) which detects cells for fitness that is to find the optimum value of the objective function. The whole process is made up of several stages and process of each stage is described of existing procedure. As the above existing fitness method is not appropriate in terms of evolutionary computing, CABEC Model has been introduced as a new model in order to meet the constraints of fitness computation or survival of fittest computation using genetic algorithm based approach through cellular automata. The block diagram of the said model is given in Fig. 2

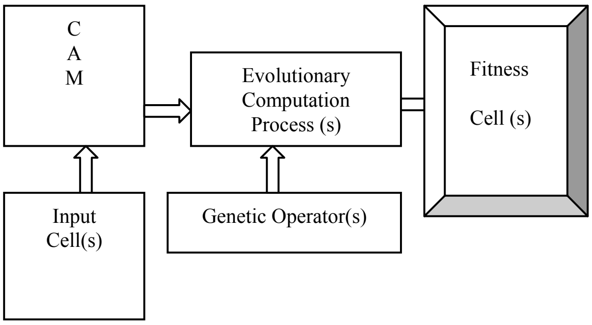
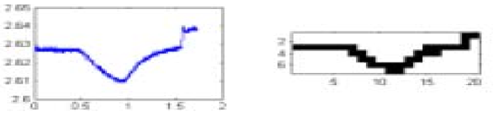


Figure 2.Cellular Automata Based Evloutinary Computing Model [CABEC Model]

Our main objective to present this model is to simulate the process of evolutionary computation and extract the fittest cells or genomes which can not be more competently detected by traditional based methods. It can only be detected easily by biological evolution procedure. The entire process is consisting of three basic stages: transform stage, recognition stage, and decision stage. Recognition and detection stages are defined corresponding to the local information and global information respectively [4].

4.1. Transform Stage

To implement the transition function of CA to recognize the weak cell, the recognition object first should be transformed into a regular lattice which can be worked on by the transition function of CA. In connection with the transform stage, the human's visual perception can be considered to decide how much fact would be sufficient for extended detection. Thus we would know how much fact should be kept after the transform stage. When we want to see the weak cells occurs, and then only need to see the visual view of main trend curve is continuous or discontinuous. Discontinuous shows only the weak edges presence. During the detection process, the region which includes all the points of the genetic curve can be split into an m*n Lattice L to find the significance of the cell that turns true if there are fittest cells existing in the consistent region of the genetic curve. When a weak or false cell will occur then only one need to see the main trend of the genetic curve, not all the data one after another. So the original genetic curve can be simplified when it is transformed into a lattice. Thus after transform stage, the trend of the curve can still be remain same (see Fig. 3) and the false cell will be detected with a minimum cost of computation. In fig. 2 the black square stands for "true" while the white one for "false" [4].



(a) (b)

Fig 3. (a) Original Genetic Curve, and (b) Genetic Curve obtained after using CABEC Model.

4.2. Recognition Stage

For convenience, a decimal number is used to represent a specific pattern of a black cell and its 8 neighboring cells. The Fig. 4 is self explanatory for the rule that translate a pattern to a decimal number representation.

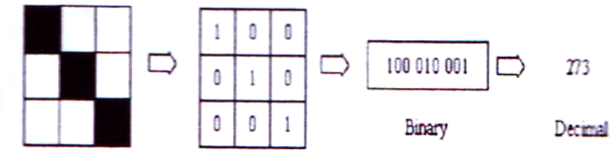


Fig. 4a. Translate a pattern to a decimal number.

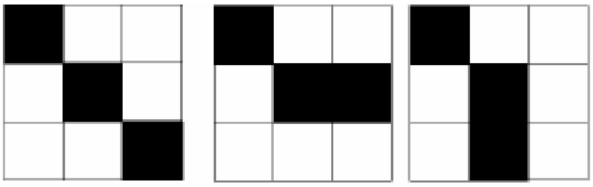


Fig. 4b. Some typical patterns corresponding to decline cells.

It can be seen after transform process that when a weak cell occurs there is a breaking in the in the resulting that corresponds to the discontinuity at a point in the genetic curve. So if a decline trend is detected from the lattice the weak cell can be detected. Some of the typical patterns which correspond to decline cells are shown in Fig. 4. It can be seen that these patterns have their own characteristics. For example, if a black cell r and its 8 neighboring cells have any of the characteristics of Fig. 4, the cell r will be thought to be a cell that makes up of the decline. The transition function of CA which completes the recognition process can be stated as

$$f = \begin{cases} True, if\ neighborhood\ is\ in\ \Phi(N) \\ False, otherwise \end{cases}$$

Where Φ(N) ={49,50,146,147,152,273,274,568}and each number in Φ(*N*) denotes a decimal number corresponding to a specific pattern. With the transition function f, the decline trend of the lattice can emerge out of the lattice of CA as shown in Fig. 5 where the left figure represents the lattice and the right figure depicts the decline pattern of the lattice. So the transition rule can be said to have the function of decline recognition [4, 5].

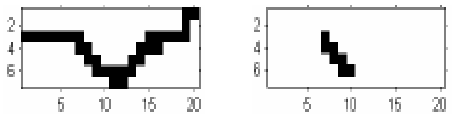


Fig. 5 Evolution of CA with the recognition transition function.

4.4. Decision Stage

The global information will be used to improve the results obtained from the initial stage by the consideration of local information. To compute the fittest cell, global information is suitable due to strong neighborhood connectivity. For convenience, the cell for which the value is true will be called the true cell. Following [4] the process required to compute the fittest or weak cells can be designed as follows.

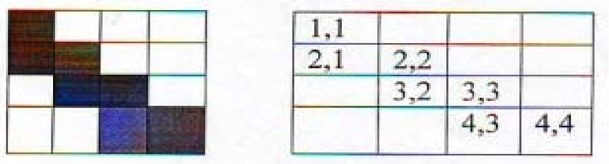
Step 1: A true cell is considered as a starting point which is given the additional value (i, j) to represent the row and the column.

Step 2: If there is a true cell to the right of the current point, it will be given the additional value (i, j+1). If a true cell is just below the current point, the additional value is (i+1, j). Similarly for just below right position the addition value will be (i+1, j+1).

Step 3: If there is no true cell in the neighborhood of the precious cell, the searching range will be expanded along the horizontal direction.

Step 4: Repeat (2) and (3) until to reach the cell in the last row or column or there is no true cell to continue the decline trend.

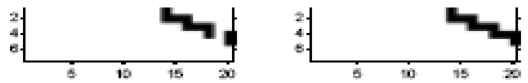
Above steps are illustrated in Fig. 6.



1,1			
2,1	2,2		
	3,2	3,3	
		4,3	4,4

Fig. 6 An example of the corrected process.

During the corrected process the last cell which has the large extra value can help in finding a declined trend. If there is no such large decline trend, the conclusion is that no dead cell occurs during the time period of the observation.If a dead cell is detected then some information on weak cells can be obtained. The weak cell starts at the point where the cell has the extra value (i,j). Also the extra value of the ending cell can approximate the magnitude and the slope of the decline trend. On the basis of the above defined corrected process the CABEC Model can give the desired output as shown in the following Fig. 7.



(a) (b)

Fig.7 (a) Before cellular approach (b) After cellular approach

4.5. Limitations of CABEC Model

It is seen that a genetic algorithm, the basic method of Evolutionary Computation, has proven to provide an efficient and powerful problem-solving strategy. The first, and most important, consideration in creating a genetic algorithm is defining a representation for the problem. Any model of this type used to specify candidate solutions must be robust; i.e., it must be able to tolerate random changes such that fatal errors do not consistently occur. The higher the fitness the better the solution is attainable. Although, these algorithms are powerful by their nature, if the fitness function is poorly chosen or defined imprecisely they may be unable to generate a proper solution to the given problem. The proposed model may not explore enough of the solution space to consistently find good solutions due to large neighborhood computation that is required for the transition function. Therefore, caution must be taken as increasing the neighborhood set and the fitness cell state that results in an exponential increase in the search space

5. Conclusion

In this paper, a new method of fittest cell detection technique based on Cellular Automata (CA) is proposed. This new simple recognition process may be used for efficient cell detection for any assigned problem that needs genetic algorithm based approach. A neighborhood detection procedure is used to establish the correct neighborhood, which gives a detailed idea for getting the proper fitness declined curve. Cellular automata approach proves to be an effective technique that could benefit immensely for reducing computation time and improving output accuracy. This model is easy to understand and supports local as well as global based information environments. Further work maybe carried out to attempt more complex processing tasks. This may require larger neighborhoods for the transition function. As the proposed model yields significant improvement in accuracy, we can conclude that it is appropriate and suitable on different multilevel computing than other existing models.

Acknowledgment

The authors would like to express their cordial thanks to anonymous referees for their valuable suggestions.

References

[1]. T. Toffoli and N. Margolus, “*Cellular automata machines”*. The MIT Press, Cambridge, Massachusetts, 1987.

[2].Michal. D. Vose, “The simple genetic algorithm”, University of Tennessee, Knoxville.

[3]. Goldberg, D.E., "Genetic and Evolutionary Algorithms Come of Age", 1994 .pp 113-1 19.

[4]. Chen Yang, Hao Ye, Guizeng Wang, “A cellular automata based method for leak detection in Pipelines”, Tsinghua, University, Beijing, 2002

[5]. Yang, C., H. Ye and G. Z. Wang, “Cellular automata modeling in Edge Recognition”, 2002.

[6] Adam Marczyk, “Genetic Algorithms and Evolutionary Computation.” 2004.

[7] John P. Cartlidge , “An analysis of evolutionary computation used in image processing techniques”,1999/2000

[8] Prof. Marek Perkowski , Hilton Tamanaha Goi , “An Original Method of Edge Detection Based on Cellular Automata”,2003

[9] Melanie Mitchell, “Computation in Cellular Automata: A Selected Review”, U.S.A., pp. 95–140., 1998

[10] Electronic Technology Directions to the Year 2000, 1995. Proceedings, “Introduction to evolutionary computing technique “, 1995 Page(s):122 - 127 1995.

[11] Huricha Ruanxiogang, “Differential equation and cellular automata model”, 2003.

[12]John Holland, “Evolution and genetic algorithm”, University of Michigan”, 1992.

[13]. L.BULL, “On model–based evolutionary computation.”, 1999, pp. 76-82.

[14] Dr. G. Sahoo and Tapas Kumar, “Theory of computation: A new approach of computation into cellular automata”, Proceeding of 2nd International Conference on Advanced Computing & Communication Technologies (ICACCT-2007).

G. Sahoo received his P.G. degree from Utkal University in the year 1980 and Ph.D degree in the area of Computational Mathematics from Indian Institute of Technology, Kharagpur in the year 1987.He is associated with Birla Institute of Technology, Mesra, Ranchi, India since 1988. He is currently working as a professor and heading the Department of Computer Science and Engineering. His research interest includes theoretical computer science, parallel and distributed computing, image processing and pattern recognition.

Mr. Tapas kumar received his B.E.(CSE) from Amravati University, Amravati, India in the year of 1998, Master of Computer Science from Guru Jambeshwar Technical University, Hissar. He is doing PhD in the area of Cellular Automata application in Image Processing from BIT Mesra, Ranchi. Currently He is working as Senior Lecturer (Selection Grade) in LIMAT, Faridabad, India. He is the life member of ISTE and International Association of Engineers (IAENG).