مجلة العلوم البحثة والتطبيقية
Journal of Pure & Applied Sciences
www.Suj.sebhau.edu.ly ISSN 2521-9200
Received 25/11/2017 Revised 02/02/2018 Published online 30/06/2018

Automat cellule technique for Timetable problem

Ali. Alarabi[a] , *Khadija. Binyahmed[b]
[a]Electronic and electrical engineering, Faculty of engineering and Technology, Sebha University, Libya
[b] Mathematics, Sciences, Sebha University, Libya
*Corresponding author: kha.Binyahmed@sebhau.edu.ly

Abstract In this paper we try engage to one of the famous educational problem. A timetable is a combinatorial optimization problem where there are allot of suggested solutions proposed for weekly table and authors for semester. This type of problem is a dynamic. The purpose from these solutions to escape from the clashes that exist frequently. here we going to present one technique with which we try to solve this problem. Our technique is an automat cellule. With this technique we will use a matrix to distribute the courses. Our matrix is a two dimension table. Probabilistic method to escape from clashes. This technique work with some roles which controls the cellule in the network.

Keywords: automat cellule, matrix.

الجدول الدراسي باستخدام الخلايا الالية

علي عرابي[1] و *خديجة بن يحمد[2]
[1] الهندسة الالكترونية والكهربائية – كلية العلوم الهندسة والتقنية– جامعة سبها، ليبيا
[2] قسم رياضيات – كلية العلوم – جامعة سبها، ليبيا
*للمراسلة: kha.Binyahmed@sebhau.edu.ly

الملخص نحاول في هذه الورقة إيجاد حل لمشكلة الجدول الدراسي وذلك باستخدام الخلايا الاليه حيت نستعين أيضا. باستخدام مصفوفة ذات بعدين حيت نقوم في الخطوة الأولي بتوزيع المواد بشكل عشوائي داخل خلايا المصفوفة تم نطبق الخوارزمية علي المصفوفة ، نكرر الخطوة عدد من المرات ونراقب توزيع المواد داخل المصفوفة.

الكلمات المفتاحية: الجدول الدراسي، مصفوفه.

1. Introduction:

The timetable problem is not a new problem but its exist in any education centre. Its combinatorial optimization problem and it has confirmed to be an NP complete problem[2]. where the administrator always trying to find a good table which not contain clashes between the different courses. Usually, some institutions tries to modify the previous tables to be used for the next year[1]. Our idea is to choose four departments. From every one of them take common courses between these departments. From mathematic we have MA 201, MA202, MA301, MA401, MA403 and from Physics we have Ph302, Ph605, Ph402. From Statistics we have St201, St202, St401, St402. From computer science we have Cs201, cs301, cs401,cs403, cs303. Here we try to apply one technique which is called automat cellules. We hope that this technique can organize the courses as possible. We also we need to calculate the chance of clashes where we use this formula for that

1.1 Timetable

Timetable problem is studded due to wide range of application. Often this problem is solved manually and by using computers tools[3].
Many approaches has been used which based on local search or constraints.
Allot of models have been described to arrive to an ideal solution. One of this solution is graph theoretical models[4]. Some of this approaches given satisfy results.

1.2 Problem type

This type of problem is not static where the courses always in the same time and in the same place. The situation in the university in unstable where the number of the students always increases and not all of them will passes the courses from the first time. This situation can create the clashes of the courses. Also the time for every one of the staff is not same. So we can say that this type of problem classified as an static problem.

2. A matrix for data

To distribute the courses on the table with clashes, we need a matrix which present all the courses. To distribute the data on the matrix first we present every course by a number. So MA101 can be present by number 1, MA201 by No 2, MA301 by No 3, MA401 by No 4, MA402 by No 5. Statics courses can be presented as ST101 by No 6, ST201 by No 7, ST202 by No 8, ST302 by No 9. Physics courses as PH302 by No 10, Ph402 by No 11, Ph605 by No 12. Computer courses as Cs201 by No 13, Cs202 by No 14, Cs301 by No 15, Cs401 by No 16, Cs403 by No 17, Cs402 by No 18, Cs501 by No 19, Cs502 by No 20. For the test we will add another 5 courses to see the result from using this technique.

2.1 Statistic method

As the problem is not a static, the solution can include a dynamic solution. So to calculate the chance that the clashes not happens we use an statistic method for that. This method is:

CNn=N!/n!(N-n) where N is the number of cells and n is the number of courses[6]

3. An automat cellule

Cellular Automata (CA) were introduced in the late 1940's by John von Neumann (von Neumann, 1966; Toffoli, 1987) and Stanislaw Ulam. From the more practical point of view it was more less in the late 1960's when John Horton Conway developed the *Game of Life* (Gardner, 1970; Dewdney, 1989; Dewdney, 1990). CA's are *discrete dynamical systems* and are often described as a counterpart to *partial differential equations*, which have the capability to describe *continuous* dynamical systems. The meaning of *discrete* is, that space, time and properties of the automaton can have only a finite, countable number of states. The basic idea is not to try to describe a complex system from "above" - to describe it using difficult equations, but simulating this system by interaction of cells following easy rules.

Hence the essential properties of a CA are

- a *regular n-dimensional lattice* (n is in most cases of one or two dimensions), where each *cell* of this lattice has a discrete state,
- a *dynamical behaviour*, described by so called *rules*. These rules describe the state of a cell for the next time step, depending on the states of the cells in the neighbourhood of the cell.

3.1 Cell-Space, Neighbours and Time

Let us define the **cell-space** as

$$L = \left\{ (i,j) \middle| i,j \in \mathbb{N}, 0 \le i < n, 0 \le j < m \right\},$$

Where *i, j* are the number of column/row of the lattice with the maximum extent of *n* columns and *m* rows. Let

$$N_{i,j} = \left\{ (k,l) \in L \middle| |k-i| \le 1 \, and \, |l-j| \le 1 \right\}$$

be the definition of the ***Moore* neighbourhood**. (Other neighbourhood definitions are similar. E.g. for the Extended Moore neighbourhood you have to replace the <= *1* with <= *2*).

Consider (as it is easier to understand) a *one dimensional* cellular automaton with *two possible states* for each cell, in mathematical terms $Z = \{0,1\}$, and *totalistic rules*, meaning, that the next state of each cell depends only on the sum of the states of the adjacent cells. So the state of cell z_i for the **next time step** *(t+1)*, one could define the totalistic rule as

$$z_i(t+1) = \begin{cases} 1, if \left(z_{i-1}(t) + z_i(t) + z_{i+1}(t) \right) = \varsigma \\ \ldots \\ 0, otherwise \end{cases}$$

Meaning that the state of the *core-cell* z_i becomes *1* if the sum of the neighbourhood cells including the core-cell is ς, *0* otherwise. To write this formula for the two dimensional automaton is not very different from this formulation and will be done in the examples section describing the Game of Life.

4. The application

In this work we going to apply the matrix of two dimensions to distribute the courses as in the following steps:-

1- first of all the courses will be distributed in an random way on the matrix. where the elements or the courses stored in cells as matrix[i][j].

2- We apply the rules of the automate on our matrix.

3- We repeat step No 2 many time to see the distribution of the courses. Instead of the binary numbers we going to use symbols of the courses direct.

4- = After the above steps we will obtain an matrix with all the courses.

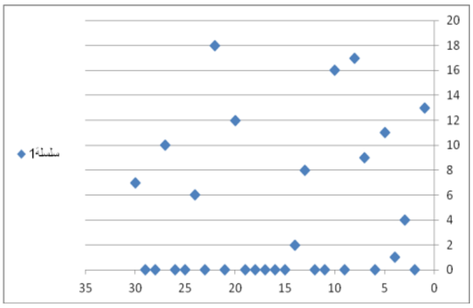
5- = we repeat the work 25 different courses.

4.1 The result

In this work we have use processor Intel core i5. We have applied this technique for two groups of 18 and 25 courses.

Table1 distribution of 18 courses

13	0	14	1	11
0	9	17	0	16
0	0	8	2	0
0	0	0	0	12
0	18	0	6	0
0	10	0	0	7

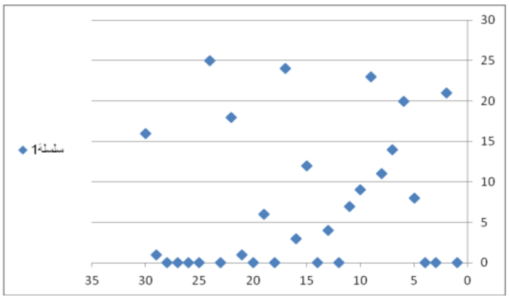


1-Plot present the distribution of 18 courses started from binary matrix

The chance of clashes for 18 courses is
C n= 30/18= .

Table2 distribution of 25 courses

0	21	0	0	8
20	14	11	23	9
7	0	4	0	12
3	24	0	6	0
1	18	0	25	0
0	0	0	1	16



2-Plot present the distribution of 25 courses started from binary matrix.

The chance of clashes for 25 courses is
Cn=30/25=   .

We have got groups of courses, where every group contains the courses numbers without clash exist. But this can see with just 18 courses. We note that this algorithm can classify the courses given in a little time. In plot No2 we observe the technique with 25 courses. Where we found that this technique to distribute group of 25 courses easily.

5. Perspective

The major advantage of this approach is that it can be used to classify a big number of courses in a little time. In the moment of using a big number of courses will appear that table of general courses will not contain more space. This is relative of using automat cellule technique. We have observed in table2 that one course its appear two times.

As we have a positive action from the application of the simple rules of this technique, in the approach future we will use another types of these technique which may give best result.

6. The references

[1]- Solving university timetabling problems using advanced genetic algorithms, Spyros Kazarlis, Vassilios Petridis, Pawlina Fragkou, Aristotle university of Thessaloniki, Technological Educational institute of Serres, Greece.

[2]- Solving university course timetabling problems using construction swarm optimization with local search, Ruey_Mawchen and hstofangshih, department of computer science and information engineering, national chinyi university of technology, Taichung, Taiwan, ssw 1999-4893, 2013.

[3]- Interactively solving school timetabling problems using extensions of constraint programming, Hadrien combazard, Fabien Demazean, Narendra Jussien and Philippe David, Ecole des Mines de Nantes, LINACNRS, Nantes cedex3, France.

[4]- An introduction to timetabling, D. De WERRA, swiss federal institute of technology, Department of mathematic, ch-115, Lausanne, Switzerland.

[5]- Agraph-based hyper heuristic for educational timetable problems, Edmund K Burke, Barry Mccollum, Amnon Meisels, Sanja Petrovic, Rong qu, Autmated scheduling ooptimization & plamming grup, university of Nottingham NG8 1bb, uk, school f computer science, queen's university Belfast, Belfast BT7 1NN, uk, Dpartement of computer science, Ben Gurion university, Beer-sheva 84105, Israel, Eurpean journal of operational research, 176: 177-12, 2007.

[6]- Maristella Agosti, Fabio Crestani and Massimo Mellucci [1997], on the use of information retrieval techniques for the automatic construction of hypertext, Information Processing & Management, Vol. 33, No. 2, pp. 133-144, 1997.

[7]- Albert0 Aimar a, James Casey b, Nikos Drakos ‘, Ian Hannell a, Arash Khodabandeh a, Paolo Palazzi a, Bertrand Rousseau a, Mario Ruggier, WebLinker [1995], a tool for managing WWW cross-references, Computer Networks and ISDN Systems 28 (1995) 99-105

[8]- Tao-min Cheng, Chung-wei Feng [2003], An effective simulation mechanism for construction operation, Automation in construction, Volume 12, Issue 3, May 2003, pages 227-244.

[9]- Stephen Van Dulken [1999], Free Patent databases on the internet: a critical view, World Patent Information, Volume 21, Issue 4, December 1999, Pages 253-257.

[10]- Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran [2008], Fundamentals of Computer Algorithms. Universities Press(2nd Edition) India 2008.