RUBIK'S CUBE AS A BENCHMARK FOR STRATEGIES OF SOLUTION SEARCH IN DISCRETE SYSTEMS PRESENTING STATE EXPLOSION PROBLEM. MODEL WITH ORDINARY AND COLORED PN

Emilio Jiménez-Macías [a], Francisco Javier Leiva-Lázaro[b], Juan-Ignacio Latorre-Biel [c], Mercedes Pérez de la Parte[d]

[a,b] University of La Rioja. Industrial Engineering Technical School.
Department of Electrical Engineering. Logroño, Spain
[c] Public University of Navarre. Department of Mechanical Engineering, Energetics and Materials.
Campus of Tudela, Spain
[d] University of La Rioja. Industrial Engineering Technical School.
Department of Mechanical Engineering. Logroño, Spain

[a] emilio.jimenez@unirioja.es, [b] francisco-javier.leiva@unirioja.es,
[c] juanignacio.latorre@unavarra.es, [d] mercedes.perez@unirioja.es

ABSTRACT

This paper presents an analysis of Rubik's Cube and its methods of resolution, used to expose, in a simple and easily understandable to students way, the state explosion problem faced by discrete systems and the possibilities of dealing with the problem based on analysis, sihmulation or a combination of both. The goal is not to advance knowledge of the cube, which is used simply as a benchmark, but to show an analogy of how in discrete production systems is given that: a) you may not have a solution to evolve the system until the desired state (the desired output), b) or sometimes a solution is available, although not optimal, c) and the combination of analytical techniques and simulation often improves the solution, but still not be optimal d) and it may even known how to get the optimal solution, but it is impossible to put into practice due to the computational (or time) cost. Additionally, by modeling the system with a PN, all the developed analysis on the system is valid on the model, allowing thus advance knowledge of the PN model. The lines to develop various PN models of Rubik's cube with PN formalisms are also exposed.

Keywords: workstation design, work measurement, ergonomics, decision support system

1. INTRODUCCIÓN

One of the biggest problems faced by those who have to deal with discrete systems is the well-known state explosion problem (Ajay, 2012, Sturtevant et al., 2009). This problem occurs in many production and logistics systems whose behavior can be represented by a discrete model, and makes it very difficult to plan production to achieve the expected result, ie to bring the system to a specific state (eg, produce 20 cars of a certain model and color, 30 other ...). The use of analysis techniques allow dealing with these systems and obtained solutions to the problem, that is, allow knowing how to bring the system to the desired state (eg, how to produce exactly the desired cars, in number for any type and model, for that day). But often it is not possible, or at least it is not known, how to analytically treat these systems efficiently, and in those cases we are forced to use simulation.

But both in one case (analytical treatment) and the other one (through simulation) it is difficult to find an optimal solution of the system, ie, how to reach the desired state of the most favorable manner according to an established criteria (eg, how to produce the desired cars for the day with the fewest hours of work, or the lowest possible economic cost, or the lowest environmental impact) (Jiménez et al., 2012). Moreover, the problem of knowing the optimal solution could be solved by simulation, if infinite computational possibilities were available. Ie it is known how to find optimal solutions, but you can not get them because the computational requirements would be prohibitive with the available resources (temporary, economic, or technological). Often those cases, in which a solution is known, although it is not the optimal one, are found. Furthermore, the combination of simulation and analytical techniques can lead to improved solutions, often without being the optimal (or sometimes even being) at least it is better than the previous ones (Jimenez et al., 2006, 2009).

All of those behaviors, inherent to discrete systems, which suffer the state explosion problem, and therefore inherent to the production processes that can be modeled by discrete formalisms, can be seen in a very intuitive and graphical way, through a benchmark: the well-known Rubik cube. Thus, this system can be used as a basis for exposing students of engineering or mathematics such conduct by an easy to understand analogy.

This paper presents the analogy of solving Rubik cube with the problem to finding the solution of a production system, and discusses the types of solutions that can be found using different analytical and simulation techniques, as well as combinations between them, for different types of initial states which may exist.

2. BENCHMARK: CUBE RUBIK

2.1. The system and its states

In the original Rubik's Cube (3×3×3) we find eight vertices, with 3 possible orientations each, and twelve edges, with two possible orientations each (Rubik, 2013, 2013B), considering (without loss of generality) that the central pieces are fixed. Therefore there are 8! ways of place the vertices, with 3^8 posibilities for the orientations, and 12! ways to plave the edges, with 2^{12} orientations. That is, the system in total presents the number of $8!\cdot 3^8\cdot 12!\cdot 2^{12} = 519.024.039.293.878.272.000$ possibilities, or possible states of the system. But those are the states that can be obtained by decomposing the system and composing it again changing the position and orientation of the pieces (provided the central pieces are fixed). But they are not the possible states of the system that can be obtained from the initial satate by the movements of the cube (which is what we want to know). In fact, from the initial state we can obtain only 1/12 of those states, since, seven of the vertices can be oriented independently, but the eighth orientation depends on the previous seven (1/3), and also one of the orientation of the edge pieces depends on the orientation of the other eleven (1/2), and given 10 positions of the edges the other 2 edge pieces can be placed in a fixed distribution of the two rest of places (1/2). Considering the above, the real number of possible permutations is of our system is (1); ie standard 3x3x3 cube provides a quantity exceeding 43 trillion possible permutations.

$$\frac{8!\cdot 3^8\cdot 12!\cdot 2^{12}}{12} = 43.252.003.274.489.856.000 \qquad (1)$$

2.2. Methods of resolution

The resolution algorithms used in this work are the following ones (Demaine et al., 2011; Korf, 1997):

2.2.1. Thistlethwaite method

This method was created by Morwen Thistlethwaite, a mathematician at the University of Tennessee, in 1981 (Jaapsch, 2013). The method is based on studying a problem as a group of subproblems, restricting each position in groups of positions that can be solved using a series of predetermined algorithms. Each of the sub-problems consist on fixe a certain position and see the movements that can be made free. Subgroups created are:

G0 = L, R, F, B, U, D
G1 = L, R, F, B, U2, D2
G2 = L, R, F2, B2, U2, D2
G3 = L2, R2, F2, B2, U2, D2
G4 = {I} → Cube solved

From the tables created for items in each group, he found a sequence of moves that led to another smaller group. A randomly chosen cube belongs to G0. From the G0 group element, subgroups belonging to G1will be obtained, and so on until the solved cube belonging to the group G4.

Originally this resolution algorithm solved the Rubik's Cube in 52 moves, but successive changes in the subgroups created permit solving it in fewer moves.

2.2.2. Kociemba algorithm.

Thistlethwaite algorithm was improved by Herbert Kociemba 1992, reducing the number of groups to only two (Kociemba, 2013).

G0 = L, R, F, B, U, D
G1 = L, R, F2, B2, U2, D2
G2 = {I}.

As in the Thistlethwaite method, the Kociemba method examines the elements between the groups G0 and G1, to find the optimal solution between groups. Using this method the cube can be solved with a maximum of 20 moves.

2.2.3. Layer by layer method.

This resolution system is the most used by initiates in the Cube, allowing cube solving very simple and sequential, needing no intuitive resolution system or memorizing complex solvers (Rubikaz, 2013). The disadvantage of this method of solution is its high number of moves needed to solve the cube as it generally exceeds 100 moves in most initial states.The resolution system consists of 7 stages:

S1. - Form a cross on the top face. This step creates a cross on the top, so that the colors match also in adjoining layers.

S2. - Place each of the vertices of the top face. At this stage of the placement of the upper vertices forming a T in each of the faces. Each vertex must be positioned so that each of its three colors match the colors of three adjacent faces.

S3. - Complete the central face. It is using the above process, but the central parts of the layer. These parts have only two colors, so that the resolving system is simpler than the previous one.

S4. - Form a cross on the underside, keeping the rest of unchanged faces. First, form a bar on the underside, and then continue with the cross.

S5. - Set the colors of the sides of the cross on the underside. At this stage the colors are placed at each end of the cross corresponding to each of the sides holding the cross on the underside. To do so first turn the top layer until having at least two side colors in their correct position.

S6. - Place each of the vertices of the lower face without orientation. In this step, we need to put the last 4 vertices in place, but no matter their orientation.

S7. - Align the corners of the underside. This step will guide each of the vertices that have been previously placed. Once the cube vertices are oriented, the cube is solved.

2.2.4. Optimal algorithm.

The optimal algorithm development was part of the research group of Herbert Kociemba, which managed to show that any cube can be solved in fewer than 20 moves (Kociemba, 2013).

In order to apply such a claim they had to make a thorough study of the various types of symmetries in the cube, which would reduce the number of possible states considerably.

They developed the symmetric states in the cube, since according to the arrangement of the colors, despite being in a different order, two states may be symmetrical, so that the resolution algorithm is the same for both.

The known symmetries to develop the optimal algorithm amounts to 164.604.041.664. For instance, the states that requie 20 movements are reduced from 1.091.994 to 32.625, because of the use of the simetries.

There are 48 kinds of possible symmetry elements (Table 1). All the states have at least a type of symmetry, except the initial state; however, other states present various types of symmetries. The combination of the 48 types of symmetries provide with different groups of symmetries to use.

Table 1: Types of symmetry

Type of simetry	Elements
1/2 rotation around an edge	6 elements
Reflection through a plane	6 elements
1/2 rotation around a face	3 elements
Reflection through a plane	3 elements
1/4 rotation around a face	2 x 3 elements
1/4 rotation + reflection through the center	2 x 3 elements
1/3 rotación alrededor de una arista	2 x 4 elements
Reflection through the center	1 elements
1/3 rotation + reflection through the center	2 x 4 elements
Identity (no movement)	1 elements

3. EXPERIMENTS ON THE SYSTEM

3.1. Methodology

In the study of this project, the analysis of 200 different states has been developed. Four tables have been filled with 50 different cases each, varying the number of random movements of the satates, from 1 to 50. Each state has been solved by the different methods that have been discussed in previous points.

3.2. Results

Table 2 shows the results obtained from a series of resolution with all methods with states achieved with from 1 to 50 random movements from the initial state. There is a column with the random movements that drive to the state from the initial state (so that the table is completely reproducible). The other columns presents the results of the simulation of each of the states for each resolution method, including the optimal algorithm, in which it has taken considerable time simulation for the 200 cases studied. The optimal algorithm simulation was performed using iterative deepening search by analyzing an average of 14.550.000.000 nodes at depth 18, and an average of 11.300 simulation seconds for each state (resulying in 2,91·1.012 nodes with a time of 628 simulation hours in total). The time for solving a state with the simplest methods is (depending on the state) just a few seconds.

Only 1 table is shown, for space questions, of the 4 with similar experiments developed to validate the methodology, for obvious space issues, but in the next section the figures with their results are shown.

4. ANALYSIS AND INTERPRETATION OF RESULTS

Table 2 allows reproducing the results, and presents the exact values of the simulations, but the most appropriate way to understand them is by the results showns in thw figures.

Figure 1 shows the data of the resolution for each of the methods. A first reflection is the great difference between advanced methods and method face to face. Obviously, having advanced algorithms gives a tremendous advantage in terms of results, but those advanced algoritms also present other disadvantages, specifically the time resolution, and the difficulty of implementation (face to face is so simple that a person can learn it in a few hours).

Figure 1 does not show the resolution of the face to face method starting from each of the colors, in order to simplify the drawing, and those 6 resolutions are shown in Figure 2. What Figure 1 additionally presents is, for every test, maximum value of 6 the 6 sides, the minimum value, and the one of the 6 obtained by the algorithm by default, which always select the same color (called Automatic method in the Figures).

It can be seen that choosing the right color with this method can reduce the movement cost very considerably (difference between red and green lines). In this case, even without advanced methods available, if you only have face-to-face method, the simulation of the 6 cases with little effort (6 times the effort of only one simulation) allows in average improving in a high percentage the solution.

Keep in mind that improved rarely solutions is proportional to stress (this is not an exception). For example, Figure 3 shows the advanced cases: advanced, the 2-phase, and the optimum. It also comes in red an upper bound to the minimum value of resolution, because that value will always be less than 20 (maximum demonstrated value movements required from any state), and less than or equal to the movements necessary to get the state from the initial state (as with the reverse sequence resolves sure).

Table 2: Sequence of 50 resolution tests with all methods, random states of 1 to 50 movements

n	Random movements	C1	C2	C3	C4	C5	C6	Aut.	Adv.	2Ph	Opt.
1	D	1	1	1	1	1	1	1	1	1	1
2	R2,B’	2	99	2	2	2	155	51	2	2	2
3	U2,R’,L	127	140	79	82	67	80	3	3	3	3
4	L2,B2,R,F’	123	106	148	116	107	105	146	4	4	4
5	U2,B’,L2,D’,R2	125	136	132	116	94	134	107	5	5	5
6	L2,B2,F’,R’,F’,B2	135	77	129	86	69	116	108	6	6	6
7	U,D,F2,D',R,F,L2	131	124	92	101	139	120	127	12	7	7
8	R',B2,L2,B2,R2,B',F2,U2	115	104	104	108	101	114	112	8	8	7
9	D2,L,R2,U2,D2,R,B',F2,L	154	87	104	124	110	161	111	18	9	9
10	B,D',B',L2,R',U2,L2,R,D,R'	87	117	98	112	100	105	137	22	10	10
11	L',F',L2,D2,F2,L,R',F,U2,B,L2	112	127	97	114	90	145	71	24	11	11
12	U,D2,F2,U2,L2,B,R,U,B,D',L2,U'	104	102	119	155	136	100	120	29	12	12
13	U2,L',R2,B2,U,L,R2,U',D,R2,B',F,L2	112	146	127	118	120	162	130	32	13	13
14	B',U2,B2,D2,L2,U',F',L,B,U',B2,U',D2,L	125	142	129	110	160	100	106	28	14	14
15	U,B',R2,B,L',F',L',R',B',F2,L2,R',F',D,L2	146	162	120	147	102	123	115	32	15	15
16	U2,L,R,U2,L2,B2,D,F2,L2,F2,U2,L',D,B2,L',R2	107	153	135	113	134	116	115	31	14	14
17	L2,R,U2,D,L,R',B2,U',L2,R2,F,L,D,B2,R2,B,D2	136	129	111	123	112	123	147	30	18	17
18	D,F2,R2,B,L2,R',U',L,R,B',D2,L',R,F,R',F,D',L2	106	89	148	115	116	92	96	30	19	18
19	L2,R,B',F2,L',R2,B2,R2,F2,L',R',B2,F2,U',L,R2,F,D,L	139	120	123	120	116	129	126	31	16	16
20	R2,B2,F2,U,R',B',L2,B',F2,L',R',B',F',L,B,F,R2,L2,U2,L2	119	110	129	147	133	145	114	31	17	17
21	L2,R',F',L',U2,D2,R2,B',U',D,R',B2,L2,F',R',F2,L2,R2,D2,R,L2	126	111	144	121	132	113	112	31	17	17
22	L,B2,L2,F2,D,L,R',B,U,F2,U',R,U2,L',B2,L2,R,U,B2,U,R',L2	118	131	138	147	124	147	109	35	17	17
23	D,R',F2,L,U,F2,L,R2,D,F,U2,F2,L2,F',L',R',U2,D,B',L,R2,F2,L'	110	122	114	139	116	119	121	33	19	18
24	L2,F2,U',L2,R',F,L2,R,B2,F',L2,B2,L,R',F',R2,B2,R2,B',F2,L',R',D2,R'	118	110	137	111	111	115	112	29	18	18
25	B2,U,D',R,D2,L',B,F2,D,L',B2,F,L2,R',B,L2,R2,B2,D,L',R2,F,L',U',R'	115	146	135	139	105	120	145	33	18	18
26	L2,U',F',R2,F,D,F',U',B2,D2,R2,D,B2,L,D',B',U2,L2,R2,D2,L2,F2,U,R,F',R2	134	126	163	120	153	122	105	34	19	18
27	L',D2,F,U',D2,B,L2,R2,F,L',F',D2,L,B',F',R,B2,U',B',D',L',R2,F2,L2,R',F2,R	130	144	115	109	133	154	118	32	19	18
28	R,F',L2,U2,L2,F2,R2,B2,D2,R',D,L2,U2,B2,L,F2,R,F,R2,D2,L,B2,U2,L2,U2,R',U,R	98	95	112	115	122	145	142	29	18	17
29	D,B,U,D2,R,U',L2,B,F2,U',D2,L,R',F,L,R2,F2,R2,D2,L2,U',L,D2,L2,R',B',R,B2,L2	134	107	113	115	132	154	146	29	17	17
30	L,R',B',F,U2,B2,F2,L,B2,D2,F',U2,R2,B,R2,F2,L2,D2,L',B,F2,U',D',L,D,L,R',F',D,R'	127	148	128	121	112	121	114	31	18	17
31	B',D',L2,B,L',U2,D',F,U',D',L2,D,F2,R2,U,R2,B2,D,R,U,R',B,R2,U',B2,L2,R2,D,L,U',L2	149	137	125	115	101	132	132	28	19	18
32	L,F',U2,R',U,L2,F',D',L',B2,R',B2,F2,L2,B',R',U2,B2,R',U2,L,U2,D,B',F2,R',D',R,U2,F2,R,U'	130	131	137	147	110	97	130	37	19	18
33	F2,L',B,R2,F,L',R',U',D',R2,F,U',D2,L,R',F2,L2,R',B2,F2,R',F2,L2,R2,B',U,L,R,B2,D,B2,R2,L2	112	135	137	113	136	158	112	24	17	17
34	L,U2,B',U',L2,R',B2,F,L',R2,B',F2,U2,D',L,B,L,B2,F',L,F,U2,D2,L2,D,L,R,F',L',F2,D2,R2,D2,F2	115	130	135	134	121	101	135	32	18	18
35	D',L2,F2,D',L2,R,U,F,L',R2,B2,F2,R2,B2,L2,R,D2,L2,R2,B2,U',L2,U,F2,L',B2,R,B2,F2,R2,B2,R,B',D2,L2	122	113	70	130	94	122	70	33	18	18
36	R,D2,R',U,B',L',F2,L2,R,B2,F2,R2,U',L2,R2,F,D',B2,L,R',D2,R',U,L2,B2,U',D2,B2,U2,R,F2,U,L,R',U2,L2	123	102	132	111	117	86	111	27	18	18
37	L,U2,D',R2,U2,R2,B,U2,B,L',U,L2,U',B2,L2,D',L',R2,U',L',B2,L,U2,L',U2,B2,R2,F,L2,U2,R2,D',R2,F2,L',F,L	102	101	91	134	97	98	91	32	19	18
38	R,B2,L',R,U2,D',L2,F,L,U',D2,B2,R,B',F2,R',D2,L2,R,B2,L2,R',B2,L2,U,B2,F2,U,D,R',F2,R2,B2,D2,R',B2,F',L'	149	87	106	140	130	121	140	32	19	18
39	U',B,L2,D',L2,D,F2,L,R,F',R2,D',L,U,R2,U,L',R,B',F',D2,L',R2,D2,L2,R2,F,R2,B',U2,L,B2,U',F',L,D,L,R',F2	112	124	109	120	113	147	124	30	19	18
40	B,U,D2,F2,U2,F,L2,B',U2,F2,R2,D2,F',U2,F',U2,D2,F,R2,U,L2,U2,R,U',F,R,D',L,U',D2,F2,L2,U',B2,L2,R2,U2,F2,U',L'	145	105	97	150	122	112	97	32	19	17
41	L,B',F2,U',B',D2,L,R,B',F,U,B2,L2,U,R',U',D,L,U',F2,L,R2,B2,U2,B',F,L2,R,F',U',L,B2,D,F2,L2,U,L,R2,F',R2,L	101	125	126	150	116	141	101	31	19	18
42	R,D,L',R,D,L2,U2,R',D2,F2,R,B2,D',R2,U2,F2,L',R',U',B,U2,D2,F,L',D',R,U2,L',R',D',L2,F',L2,U',D',L',R2,U,L',R',U',F2	109	129	108	134	124	129	134	24	19	16
43	D2,R2,U2,B2,F,R',U2,L',R,F',D,B2,F2,R',B,R',U2,B2,U,R2,U',L,U2,D',L,D,L,R,D,L',B2,R,U,D2,F,L',U,R2,F',L,R2,D2,F2	86	145	132	145	114	125	86	30	19	17
44	R',B2,D',B2,R2,D',R',U,L2,R',U,L,D2,F2,L2,R',B2,R',F,L,R,F2,R,B,F2,U,D,L',F,R2,B',R,B',F2,L',D,R,D',L2,R2,U2,L2,F,R2	143	130	131	133	141	90	130	30	19	18
45	R2,U',D,R',U',D2,L,F',L2,R,U',D2,L,F,L,B2,L',R2,U2,D',L',D',R',D',B',U',L2,U',L,D2,B,D2,B2,L2,B',L2,B2,U2,L2,R2,F2,D,F',L',R2	146	150	138	130	122	133	150	26	18	18
46	U',D',R',U',L',R,F',L2,B',D',F2,R',B2,U',B',D2,F,L,D2,L',U2,D',R2,F2,L',U2,L',R2,U2,D2,L2,R',F,L2,F2,D,R',B2,L2,F,L2,F,L2,B2,L2,R'	138	108	112	124	117	126	124	31	18	18
47	D',L2,B2,F2,D2,B,L2,R2,F2,D2,B2,R2,F2,R',B,L2,B2,F2,U,R',D,L',R,U2,L,R2,F2,D2,F',R',D',R',U',L2,R,F2,L,R,F2,L,U,F',L2,F',L2,R',U'	122	141	130	128	122	107	128	33	17	17
48	L2,R,B2,U2,L2,D2,L2,D,L2,D2,R2,F',L2,U2,B,U',D2,L2,U',R,U,D',R2,F',L2,F,L2,U2,L2,D,L',F2,L',R',F2,U2,L2,R',B,R2,F2,R,D',L2,F2,U,D,R2	108	117	125	107	136	104	104	25	19	18
49	R2,D2,B,F2,L2,D2,R',U,D2,L2,F,U,B',L',F2,D2,L,B,R',B2,F',L2,B2,R2,B,L2,R',U2,L2,D2,L,U,B2,F',D2,R2,U2,B',L2,B2,L2,R2,B2,F2,L,F,R',L2,U2	133	116	121	122	121	122	122	29	19	18
50	F2,D2,L,B2,L2,R2,F2,U2,L',U',B',R2,U2,L,B',U,D,L',U,R',B',R2,B',R',B2,F',U',L',R2,D2,B,R',B2,L,R,D,L2,R2,D2,L2,R2,U',D2,L,F2,L',F2,L2,R',F'	140	144	121	101	139	129	101	33	19	18

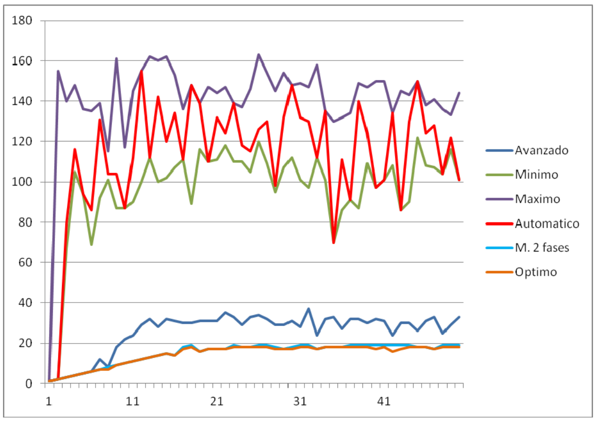

Figure 1: Resolution by all methods

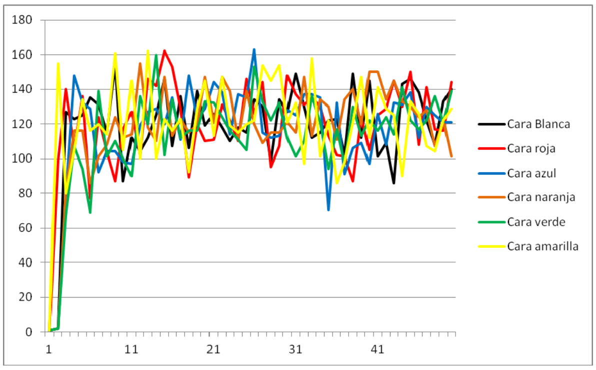

Figure 2: Resolution face to face method, implemented by the 6 possible faces

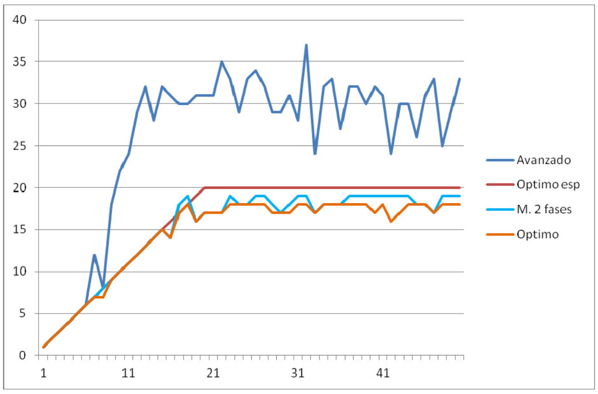

Figure 3: Resolution by the 3 advanced methods: Advanced, 2-Phases, and Optimal

We see the very advanced method substantially improves the result compared to face-to-face method, even in the best face possible. Besides its computational cost is only slightly higher (takes a few seconds on a personal computer environment), but its implementation is more difficult.

Keep in mind that solutions improvement rarely is proportional to effort (this is not an exception). For example, Figure 3 shows the advanced cases: advanced, the 2-phase, and the optimum. It also presents in red an upper bound to the minimum value of resolution,

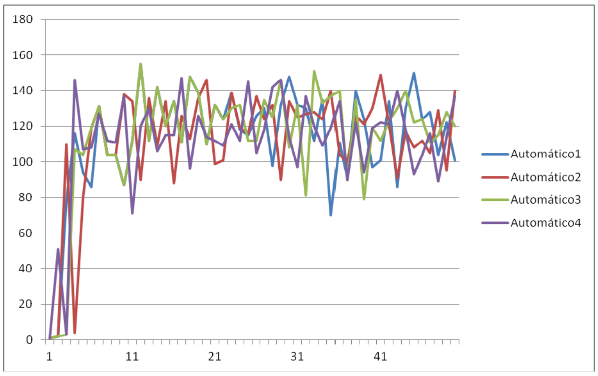

Figure 4a: Resolution by the method face to face in the 4 different sets of tests

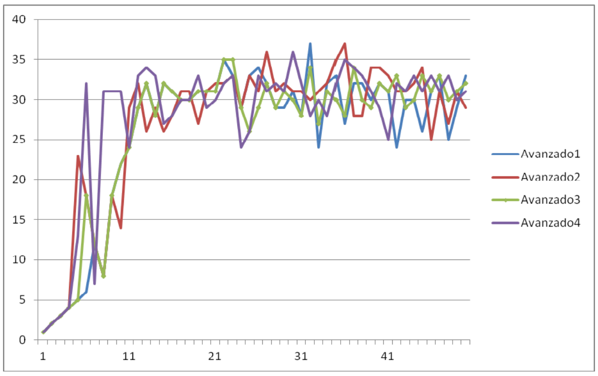

Figure 4b: Resolution by the advanced method in the 4 different sets of tests

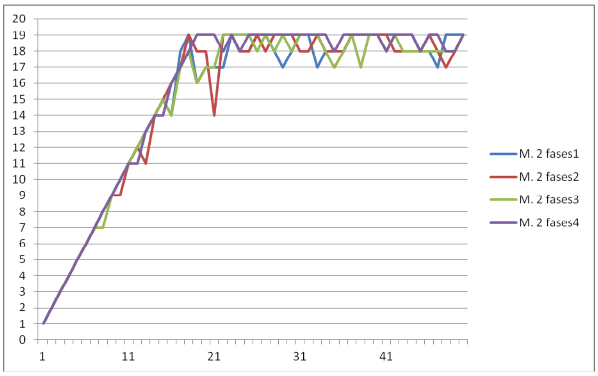

Figure 4c: Resolution method 2 phases in the 4 different sets of tests

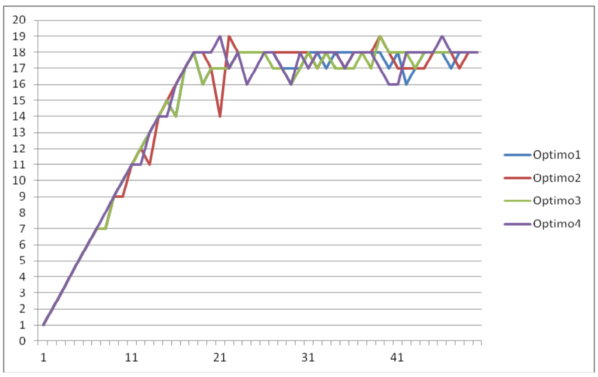

Figure 4d: Optimal Resolution method in the 4 different sets of tests

because that value will always be less than 20 (demonstrated maximum movements required from any state), and less than or equal to the movements necessary to get the state from the initial state (as the reverse sequence is a solution). The very advanced method substantially improves the result compared to face-to-face method, even in the best face possible. Besides, its computational cost is only slightly higher (takes a few seconds on a personal computer environment), but its implementation is more difficult.

The 2-phase method still further improves (less than half, in average) but computational cost is also much higher, and also the difficulty of implementation. And the optimal method improves just a little earlier, but still requires much more time to resolution (now in the order of hours).

Figures 4a to 4c show in a single graph the methods used for each of the four tests, to check that those results constitute a trend and not casuality).

5. PN APPROACH

After analyzing the system, the next step is to study the feasibility of building a model with PN. In addition, depending on the formalism and the approaches, different models could be developed (Silva 1993). The fundamental requirement is that any model represents exactly what we need to know about the system, and in this case, this is the states. Thus, all developed models will have the advantage that we know its behavious since we know a lot of information about the system (such as the number of reachable states or the ways to obtain any state). Furthermore, the analysis of models may provide information of the system; for example, any repetitive component determined from the incidence matrix of the model provides information on how to solve the system since each of the intermediate nodes (Latorre et al., 2009, 2103, 2013B).

In this section we will present different approaches to model the Rubik's Cube by PN. Such models are not developed in the paper, because each of them requires more space than an article to study it in detail, and only the Figures would also require more than one paper. So, the aim is simply to present the simplest possibilities, and the relationships between them; with this information any PN expert can build the models without problems.

5.1. Approach of labels with Colored PN

It is conceptually the simplest approach. The color lavels of the faces are considered as tokens, the places where can be each label are the PN places, and as transitions each of the 12 possible moves (spin in both directions of each of the 6 sides). This PN have these characteristics:

Tokens: 6 colors, and 8 tokens each color
Places: 48 positions
Transitions: 12 moves

In this model, the constraints derived from relative positions of the colors (such that no two of the same color on both sides of an edge) derive from the internal structure of the model and the initial marking. If anyone analises this PN model without knowing that it represents a Rubik cube, it would not be easy initially appreciate that there is a close relationship between labels for the various blocks that make up the cube.

Also noteworthy is that this model would represent the same system (would be equivalent) if were to be used 48 pieces each color, as the only possible solution in the cube presents always each label in the same position.

5.2. Approach of Physical blocks with Coloured PN

Marks are considered as tokens, and as PN places the places where the parts can be. Therefore there exist 20 places and 20 tokens. But only the corner pieces can be in the corners, and the pieces of the sides can only be in the side. Therefore, the model will have two unconnected sub-networks: une with 8 places/tokens and the other one with 12 places/tokens. But additionally must have other 20 subnetworks to indicate the orientations of each of the parts (2 possible orientations of the side pieces, and the 3 possible positions of the corners pieces). With all this, the system will have (all with 12 transitions, corresponding to the movements):

- Subnet 1
  Tokens: 8 tokens, of 1 color each
  Places: 8 positions
- Subnet 2
  Tokens: 12 tokens, of 1 color each
  Places: 12 positions
- Subnets 3 (8 subnets):
  Tokens: 1
  Places 3 (corresponding to the possible orientations)
- Subnets 4 (12 subnets):
  Tokens: 1
  Places: 2 (corresponding to possible orientations)

5.3. Approaches with ordinary PN

Any of the previous models can be made in colorless PN (generalized PN, which also are ordinary since no weights on the arcs exist in these models). It is well known that the ability of modeling the colored PN is exactly the same as ordinary PN, although obviously condensation capacity is much higher.

The way to convert the two previous models is as simple as unfolding them, ie divide each PN in as many networks as colors. For example, colored PN made in 5.1 would be now 6 ordinary PN, each with 48 places, 12 transitions and 6 tokens, or even 48 PN with 48 places, 12 transitions and one token. The PN indicated in section 5.2 could be done by 8 PN with 8 places, 12 PN with 12 places, 8 PN with 3 places, and 12 PN with 2 places, all of them with 12 transitions and one token.

6. CONCLUSIONS

The analysis developed from the tests shows an analogy between the well known system Rubik cube and real discrete production systems suffering from the state explosion problem:

- The optimum is almost always impossible to achieve, due to computational costs for its calculation and for lack of methods and algorithms to achieve it (apart from "brute force").
- With equal computational resources, research (analysis, ie, more efficient algorithms) dramatically reduces the results
- With equal computational resources and algorithms, the use of simulation (different options using the same algorithm) can lead to improvements in many cases.
- The Rubik cube benchmark is analogous to discrete production systems, because of the combinatorial explosion or state explosion inherent to discrete systems, and can be used for learning (teaching) or for deeper understanding (research).

That analogy can be enriched by modeling the system using a formalism eminently useful for modeling discrete systems with state explosion, such as Petri nets. Thus, following construction of the models (in one of the many different possibilities offered by the paradigm of Petri nets as the great family of formalisms that is), a lot of properties of these models are known (as much as are known of the system: the Rubic cube).

ACKNOWLEDGMENTS

This paper has been partially supported by e grant of the University of La Rioja and Banco Santander.

REFERENCES

Ajay, K., 2012. Search Techniques To Contain Combinatorial Explosion in Artificial Intelligence, *International Journal of Engineering Research & Technology* (IJERT), Sep 2012.

Demaine, E.D., Demaine, M.L., Eisenstat, S., Lubiw, A., Winslow, A., 2011. Algorithms for Solving Rubik's Cubes, *Cornell University*, Jun 2011.

Jaapsch http://www.jaapsch.net/puzzles/thistle.htm

Jimenez, E., Perez, M., Latorre, J.I., 2006. Industrial applications of Petri nets: system modelling and simulation. *Proceedings of International Mediterranean Modelling Multiconference* 2006, pp. 159-164

Jimenez, E., Perez, M., Latorre, J.I., 2009. Modelling and simulation with discrete and continuous PN: semantics and delays. *Proceedings of 21st European Modeling and Simulation Symposium, Vol II*, pp. 14-19

Jimenez, E., Tejeda, A., Perez, M., Blanco, J., Martinez, E., 2012. Applicability of lean production with VSM to the Rioja wine sector. *International Journal of Production Research*, 50 (7), 1890–1904

Kociemba, 2013. http://kociemba.org/cube.htm

Korf, R.E., 1997. Finding optimal solutions to Rubik's Cube using pattern databases, *Proc. Nat. Conf. on Artificial Intelligence (AAAI-97)*, Providence, Rhode Island, Jul 1997, pages 700–705.

Latorre, J.I., Jimenez, E., Blanco, J., Sáenz-Díez, J.C., 2013. Integrated methodology for efficient decision support in the Rioja wine production sector. *International Journal of Food Engineering*, (In press).

Latorre, J.I., Jimenez, E., Perez, M., 2009. Decision taking on the production strategy of a manufacturing facility. An integrated methodology. *Proceedings of 21st European Modeling and Simulation Symposium, Vol II*, pp. 1-7.

Latorre, J.I., Jimenez, E., Perez, M., 2013. Simulation-based Optimisation of Discrete Event Systems by Distributed Computation. *Simulation-Transactions of the Society for Modeling and Simulation International*, (In press).

Latorre, J.I., Jimenez, E., Perez, M., 2013. The optimization problem based on alternatives aggregation Petri nets as models for industrial discrete event systems. *Simulation-Transactions of the Society for Modeling and Simulation International*, 89 (3), 346–361.

Rubik,2013. http://en.wikipedia.org/wiki/Rubik's_Cube

Rubik, 2013b. http://www.rubiks.com/

Rubikaz, 2013. http://www.rubikaz.com/resolucion.php

Silva, M., 1993. Introducing Petri nets. In *Practice of Petri Nets in Manufacturing*, Di Cesare, F., (editor), pp. 1-62. Ed. Chapman&Hall.

Sturtevant, N., Felner, A., Barrer, M., Schaeffer, J., Burch, N., 2009. Memory-Based Heuristics for Explicit State Spaces, *IJCAI'09 Proceedings of the 21st international jont conference on Artifical intelligence*, Pages 609-614.