Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, Volume 21, No. 1, ISSN 1726-9679
ISBN 978-3-901509-73-5, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2010
Make Harmony Between Technology and Nature, and Your Mind will Fly Free as a Bird
*Young Researches and Scientist Paper / * Supervisor, Mentor*

CRACKING KAKURO PUZZLES BY DIFFERENTIAL EVOLUTION

BOUDNA, H[ana]

Abstract: *Kakuro is a logic puzzle, often refered to as mathematical equivalent of a crossword. There is also another one – Sudoku – which has successfully been solved by means of evolutionary algorithms. However, not many fruitful evolutionary approaches to unriddleKakuroseem to be accomplished yet. This paper uncovers this challenging problem and presents an attempt to crackKakuro by employing Differential Evolution as a solver.*
Key words: *Kakuro, Puzzle, Differential Evolution*

1. INTRODUCTION

In recent yearsKakuro puzzles have gained enormous popularity among wide group of people, representing an enigma which complexity belongs to a set of NP-problems.Sudoku puzzles may be considered to be related problem, on which many experimental papers have been published. Most authors employevolutionary algorithms to solve Sudokus. Joel Almong uses Evolutionary Computing Methodologies – Quantum Simulated Annealing, Cultural Genetic Algorithm and a hybrid between Simulated Annealing and Genetic Algorithm (Almong, 2009),Nicolau and Ryan deal with Genetic Algorithm using GramaticalEvolution(GAuGE) system (Nicolau& Ryan, 2006), in another studies the Sudoku puzzles are solved with Cultural Algorithms (Mantere&Koljonen, 2008),Parttical Swarm (Hereford &Gerlach, 2008), Genetic Algorithm (Mantere&Koljonen, 2007) or Improved Artificial Bee Algorithm (Pacurib et al., 2009).

Several researchers have been done in an attempt to find efficient methods of obtaining the solution of Kakuro puzzles.In (Simonis, 2008) authors use MILP (mixed integer linear programming) and a PseudoBoolean model mapped to a SAT solver for solving kakuro puzzles. Some attention has also been given to the use of P.R.P., and P.R.P. and C.S.E. together approaches to speed solution time (Davies et al., 2009), and a Nested Monte-Carlo Search at level 2 method (Cazenave, 2009).(Davies et al., 2008) have noted that solving Kakuro puzzles is an important and useful element for construction of codes, where run totals may form a generalised type of parity check.

The main purpose of the experiment reported here was to useDifferential Evolution(DE) approach tosolve Kakuro puzzles.

2. THE PROPOSED METHOD

2.1 TheKakuro Puzzle

A Kakuro (Cross sum)puzzle consists of a playing area of filled and empty cells similar to crosswords as shown in Figure 1. The Kakuro puzzle can consist of different size of length and width. Some black cells contain one or two numbers reading left to right and top to bottom, called "the clues". A number in the top right corner relates to an "across" clue and one in the bottom left a "down" clue.

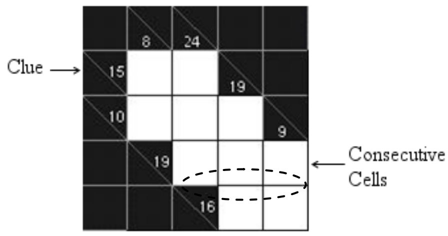

Fig. 1. Example of Kakuro puzzle

In the process of filling white empty cells ofKakuro puzzles, we need to satisfy three following rules:

1) The clues for across and down numbers are the sum of digits in that number,
2) Digits from range <1; 9> can solely be used,
3) No number may be used more than once in the same block (consecutive cells).

2.2 The Differential Evolution Approach

We use Differential Evolution (theDERand1Bin version) to solve example Kakuro puzzles from www.kakurolive.com. Before the evolutionary process starts, every white cell in all rowsis filled with a random integer value between 1 and 9with respect tovalues in across and down clues.In this step, we ensure no digitsrepeat in any consecutive cellsin rows, but there may be found repeated digitsin consecutive cellsin columns as illustrated inFigure 2.

In the optimization process, the repeated digits occureboth in rows and in columns, which is supressed by a penalty function(condition three), giving us a sum of duplicate numbers in rows and columns (Figure 3b). For example, if there are 2 repeated digits in consecutive cells (either ina row or a column) penalty value 1 is added to the cost value.If there are three repeating digits, penalty value is 3 and so on.After filling all white cells, each individual stands for a representation of one solution to given Kakuro puzzle.

Hence, conditions 1 and 3 are subject to optimization. The optimization problem is to minimize acost value. A cost value of each solution is given as sum of differences between every clue and appropriate concecutive cells(condition 1) for both rows and columns.A complete Kakuropuzzle will have a cost value of 0 and penalty value of 0.

Once we have initialized our population, general steps of the proposed DE approach are performed.

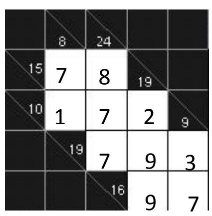

Fig. 2. Initial Solution (one generated individual)

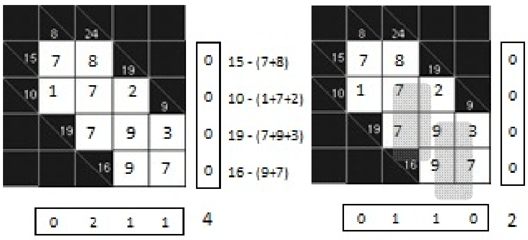

Fig. 3. Calculation of Cost Value (a) and Penalty Value (b)

In this phase, the black cells (clues)can not be changed, sothe following operations work only with white (consecutive) cells for each individual *x*:

- Randomly choose another 3 individuals*a, b*and *c*
- Mutate chosen individuals:

$$y_i = Round(c_i + F \cdot (a_i - b_i)) \qquad (1)$$

- Crossover:
  - o Generate random number from the range (0,1) for each value of white cells
  - o If the *random number<Cr*, select the value from individual x_i, otherwise select the value from individual y_i
- Check new individual for beingwithin constraints:
  - o If value in a white cell is out of bounds, generate a new onefrom interval <1; 9>
- Evaluate the newly created individual
- If $f(new\ individual) < f(x_i)$ then replace the individual in the population (retain individual with better cost value)

This cycle is evaluated untiltermination criteria are met –the maximum number of generations or the solution is is found(cost value equals to0).

3. EXPERIMENS AND RESULTS

The experiments consisted of solving six differentKakuro puzzles: two marked with *easy* difficulty, two with *medium* difficulty a two with *wicked* difficulty (each with varying size). Table 1 shows the parameters of used puzzles and parameters of DE. The parameters of differential weight *F* and crossover probability *Cr* were chosen with respect to preliminary testing. Population size *NP* and number of generations *Gen* were picked according to to number of empty cells and given clues.

Puzzle	Size	Empty	Clues	NP	F	Cr	Gen
Easy1	5x5	10	8	100	0.3	0.7	100
Easy2	7x7	19	13	200	0.3	0.7	200
Medium1	6x6	16	10	250	0.3	0.7	200
Medium2	8x8	27	18	200	0.3	0.7	500
Wicked1	8x8	35	21	300	0.3	0.7	500
Wicked2	9x9	40	22	450	0.3	0.7	500

Tab. 1.Experiments' settings

For each puzzle, the experiment was repeated 50 times, always with new randomly generated initial population. Results are summarised in Table 2.

Puzzle	Success rate	Min cfe*	Max cfe*	Mean cfe*
Easy1	60%	3,700	9,800	5,357
Easy2	74%	9,600	18,600	12,676
Medium1	76%	9,500	26,500	17,083
Medium2	50%	10,400	20,400	12,744
Wicked1	50%	49,200	81,000	63,372
Wicked2	42%	71,100	148,050	92,871

**cfe* – cost function evaluation

Tab. 2. Obtained results

As can be seen from data in Table 2, DE wasthe most successfulin solving the *Medium1* puzzle.Our experiments indicate that the difficulty level of solved puzzlesdoes not direcly relate to acquired results. In this case, the measure of difficulty is the size of puzzle. Based on the number of cost function evaluations, we can say that the *Medium1* puzzle seems to be easierthen the *Easy2* puzzle. However, the difference betwen mean value of cfe for the Medium2 and Easy2 puzzles wasminimal.

4. CONCLUSION

This paper presented the Differential Evolution approach to solve Kakuro puzzles. As we can see from the obtained results the proposed method exploiting Differential Evolution is able to solve all tested Kakuros.However, we wouldlike to point out thatnot even on the easiest Kakuro puzzle level the DE was not able to find correct solution in all 50 runs.

The future research will consist of implementing another control function for repeated digits to limit the search space.

5. REFERENCES

Almong, J. (2009). *Evolutionary Computing Methodologies for Constrained Parameter, Combinatorial Optimization: Solving the Sudoku Puzzle*, Africon, ISBN 978-1-4244-3918-8, Nairobi

Cazenave, T. (2009).Nested Monte-Carlo Search.In: *Advances in Computer Games:* 12th *International Conference*, pp. 45-54, Springer, ISBN 3642129927, 9783642129926, Spain, Pamplona

Davies, R.P.; Roach, P.A.; Perkins, S. (2009).The Use of Problem Domain Information in the Automated Solution of Kakuro Puzzles.*IAENG International Journal of Computer Science*.Vol 37, No. 2, (2010) (118-127), ISSN 1819656X

Davies, R.P.; Roach, P.A. ; Perkins, S. (2008). In: *28th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, DEC, Cambridge, ENGLAND

Hereford, J.M; Gerlach, H. (2008). Integer-valued Particle Swarm optimization applied to Sudoku puzzles, *Proceedings of Swarm Intelligence Symposium*, pp. 1-7, ISBN 978-1-4244-2704-8, MO, September 2008, St. Louis

Mantere, T.; Koljonen, J. (2007).Solving, Rating and Generating Sudoku Puzzles with GA. *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 25-28, ISBN978-1-4244-1339-3, Singapore, September 2007, CEC 2007, Singapore

Mantere, T.; Koljonen, J. (2008). Solving and AnalyzingSudokus with Cultural Algorithms, *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 4053-4060, ISBN 978-1-4244-1822-0, China,June, 2008, CEC 2008, Hong Kong

Nicolau, M.; Ryan, C. (2006). Solving Sudoku with the GAuGE System,*Proceedings ofthe 9th European Conference on Genetic Programming*, Collet, P;Tomassini, M.;Ebner, M.; Gustafson,S. (Ed.), pp. 213-224, ISBN 3-540-33143-3, Hungary, April 2006, Springer, Budapest

Pacurib, J. A.; Seno, G. M. M.; Yusiong, J. P. T. (2009). Solving Sudoku Puzzles Using Improved Artificial Bee Colony Algorithm, *Proceedings of the fourth International Conference on Innovative Computing, Information and Control*, pp. 885-888, ISBN 978-0-7695-3873-0, Taiwan, December 2009, ICICIC, Kaohsiung

Simonis, H. (2008). Kakuro as a constraint problem. *In Proceedings of the seventh Int. Works. on Constraint Modelling and Reformulation*, Australia, MODREF08, September 2008, Sydney