ISSN:0975-9646



# Comparision and Assesment of Evolutionary and Parallel Genetic Algorithms for a Cryptarithmetic Problem

**Dr. J P PATRA**
*Associate professor, Dept. of CSE*
*SSIPMT, Raipur*

**Nayan Shivhare**
*Department of Computer Science and Engineering*
*SSIPMT*
*Raipur(C.G),India*

**Shraddha Verma**
*Department of Computer Science and Engineering*
*SSIPMT*
*Raipur(C.G),India*

***Abstract*—Cryptarithmetic problem has many ways to solve using different algorithm. In this paper we proposed a solution to a problem using Parallel Genetic algorithm and Evolutionary algorithm for comparison and assessment of a cryptarithmetic problem. Comparison can be done in terms of execution times of both the algorithm in milliseconds with respect to the different numbers of variable. The result shows using chart shows that the Parallel genetic algorithm takes lesser time for execution than evolutionary algorithm.**

***Keywords*— cryptarithm, alphametics , Genetic Algorithms, Parallel Genetic Algorithms, Evolutionary**

## INTRODUCTION

Cryptarithmetic, also known as cryptarithm, alphametics, verbal arithmetic or word addition are puzzles in which a set of words is written down in the form of a long addition sum or some other mathematical problems that produces a sensible phrase and words formed by the operands[1].The object is to replace the letters of the alphabet with decimal digits to make a valid arithmetic sum.The equation is typically a basic operation of arithmetic such as addition, multiplication or division[2].The classic example,published in July 1924 issue of Strand Magazine by "Henry Dudeney" is:

```
    S E N D
+   M O R E
-----------
  M O N E Y
```

The solution to this cryptarithmetic puzzle is S=9, E=5, N=6, D=7, M=1, O=0, R=8, Y=2.

The Solution for this puzzle is shown as:

```
    9 5 6 7
+   1 0 8 5
-----------
  1 0 6 5 2
```

Each letter in this puzzle assigns different decimal digits. A good puzzle should have a unique solution and letters should make up a phrase. As, In the above example, The words made a phrase "Send More Money".

## I. LITERATURE REVIEW

### A. *Constraints Satisfaction problem*

Constraints satisfication problem are a special kind of problem where States defined by values of a fixed set of variables and Goal test defined by constraints on variable values.

#### *a) Problem definition*

A CSP consists of:

- A set of variables X = f{x1,...,xn}.
- For each variable $x_i$, a definite set $D_i$ of possible values (its domain).
- A set of constraints restricting the values that the variables can simultaneously take.

A feasible solution to a Constraints Satisfication Problem is an assignment of a value from its domain to every variable, in such a way that every constraint is satisfied. In this case, the problem is satisfiable. On the other hand, if there is no assignment of values to variables from their respective domains for which all constraints are satisfied, then the problem is unsatisfiable [3]

Algorithms for solving CSPs are aimed at simply finding a feasible solution, they can be adapted to finding an optimal solution. For instance, an objective variable can be created to represent the objective function, an initial solution is found, and then a new constraint is introduced specifying that the value of the objective variable must be better than in the initial solution. This is done repeatedly, tightening the constraint on the objective variable as each solution is found. The number of iterations, and therefore the computation time, depends on the quality of the initial solution. A common practice is to apply a heuristic method for generating an initial solution.[4]

#### *b) Evaluation of constraint satisfaction*

In this section, the evaluation of Constraints Problem as a technique for solving Constraints Satisfaction problem, and compare it with two different methods. Two general comparative papers, written from an AI standpoint, are those of Van Hentenryck (1995) and Simonis (1996). The former considers various techniques for solving combinatorial search problems, such as branch and bound, branch and cut, and local search, and compares them with CP. Simonis concludes that there are four areas where CP is most successful - scheduling, allocation, transportation and rostering. One of the practical reasons for this success is the ease with which additional problem-specific constraints can be added without any need to revise the whole program.

*c) Criteria for Evaluation*

There are various reasons why a particular technique may be chosen for a problem, including:

- Ease of implementation.
- Flexibility to handle a variety of constraints that occur in practical problems.
- Computation time.
- Solution quality.

## II. EVOLUTION OF CRYPTARITHMETIC

Cryptarithm is a genre of mathematical puzzle in which the digits are replaced by letters of the alphabet or other symbols. Cryptarithmetic is the science and art of creating and solving cryptarithms.

Cryptarithmetic puzzles are quite old and their inventor is not known. An example in The American Agriculturist of 1864 disproves the popular notion that it was invented by Sam Loyd. The name cryptarithmetic was coined by puzzlist Minos (pseudonym of Maurice Vatriquant) in the May 1931 issue of Sphinx, a Belgian magazine of recreational mathematics. In the 1955, J. A. H. Hunter introduced the word "alphabetic" to designate cryptarithms, such as Dudeney's, whose letters from meaningful words or phrases. Solving a cryptarithm by hand usually involves a mix of deductions and exhaustive tests of possibilities.

Cryptarithmetic is a class of constraint satisfaction problems which includes making mathematical relations between meaningful words using simple arithmetic operators like “+” in a way that the result is conceptually true, and assigning digits to the letters of these words and generating numbers in order to make correct arithmetic operations as well.[5]

Types of cryptarithm include the alphametic, the digimetic, and the skeletal division.

1. Alphametic - A type of cryptarithm in which a set of words is written down in the form of a long addition sum or some other mathematical problem. The object is to replace the letters of the alphabet with decimal digits to make a valid arithmetic sum.

2. Digimetic - A cryptarithm in which digits are used to represent other digits.

3. Skeletal division - A long division in which most or all of the digits are replaced by symbols to form a cryptarithm.

*b) Constraints to solve cryptarithmetic*

i) Each letter or symbol represents only one and a unique digit throughout the problem.

ii) When the digits replace letters or symbols, the resultant arithmetical operation must be correct.

iii) Total number of distinct letters should be less or equal to 10.

iv) Length of answer should not be lesser than the length of any operand.

V) Length of answer can be only one more than any of the operands.[5]

## III. EVOLUTIONARY ALGORITHM

An Evolutionary Algorithm (EA) is a common term for algorithms that utilize the adaptive behavior modeled after principles of nature.

Although the definition of Evolutionary Algorithm differs, the more common properties of EAs are that collections of potential solutions to the problem at hand are maintained.

These solutions are referred to as the population of a current generation. Each potential solution is called a chromosome.

Operations are applied to the current population to produce a new generation that will hopefully contain chromosomes that are better solutions of the problem. This process continues until some threshold value or stopping criterion is met.

The new population is produced through the operators on selected chromosomes of the current generation. Typically, the chromosomes of the current generation, to whom the operators are applied, are chosen based on their quality. In this way, it is more likely that the offspring chromosomes inherit desirable characteristics of its parents. Some heuristics or fitness functions are used to choose parent chromosomes in a generation.[6]

*a) Algorithm*

*Step 1:* Scan the input strings.
*Step 2:* Check that the input is proper.
*Step 3:* Put the letters or symbols in ARRAY[7].
*Step 4:* Apply arithmetic rules and try to reduce the solution space.
*Step 5:* If the number of distinct letter is less than 10, then fill the rest of the indices of ARRAY with don’t care symbols. This ARRAY now is our current generation.
*Step 6:* For several times, generate two random numbers m, n and swap the contents of index m and n of any one chromosome of the current generation and copy this new chromosome to the next generation.
*Step 7:* Evaluate the fitness of each chromosome of the next generation and choose the best chromosomes. Now

these best chromosomes become our current generation. Also include one random chromosome to the current generation. If there is no chromosome with error 0 in the current generation, then go to step 6. If one of the chromosomes is found with error 0, then report the solution and exit.

| **Problem** | **SEND+MORE=MONEY** | | |
|---|---|---|---|
| **Answer** | ***9567+1085=10652*** | | |
| **Time** | Min (ms) | Max(ms) | Avg(ms) |
| | 188 | 980 | 690 |

Table 3.1. Result of evolutionary algorithm

## IV. GENETIC ALGORITHM

Genetic Algorithms (GAs) are search algorithms inspired by genetics and natural selection . These algorithms are powerful search techniques that are used to solve difficult problems in many disciplines. Unfortunately, they can be very demanding in terms of computation load and memory.

Parallel Genetic Algorithms (PGAs) are parallel implementations of GAs which can provide considerable gains in terms of performance and scalability.

The most important advantage of PGAs is that in many cases they provide better performance than single population-based algorithms, even when the parallelism is simulated on conventional machines .[7]

The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached. Genetic algorithms find application in bioinformatics, computational science, engineering, economics, chemistry, manufacturing, mathematics, physics and other fields.[8]

A typical genetic algorithm requires:

- A genetic representation of the solution domain,
- A fitness function to evaluate the solution domain.

| **Var Num** | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|
| **PGA Time(ms)** | 1.112 | 8.122 | 10.455 | 15.335 | 1.112 |
| **Evol Time(ms)** | 45.32 | 60.34 | 90.77 | 100.77 | 45.32 |

Table 5.1 Comparison of PGA and Evolutionary run times

### *a) Algorithm*

*Step 1:* Extract distinct letters from input strings and put them in a list named L.

*Step 2:* Repeat the following until the desired population size is reached:

*A.For each letter in L do the following:*

2.1. Generate a random number between 0 and 9.

2.2. If the random number is equal to zero and the letter is the beginning letter of the words (which should not be zero) go to 2.4.

2.3. If the cell that is corresponding the random number is empty, put the letter into that cell, else go to step 2.1.

2.4. If the current letter is the last letter of L and the length of L is 10 then generate a random number between 1 and 9 and exchange the place of the letter from that cell to the index of zero and the current letter to that position. Keep doing it until the second letter is not a beginning letter of the words. Else go to step 2.1.

*B.* Calculate the fitness of this new individual.

*C.* Add the individual into the right place in the inner population list [9].

| **Problem** | **SEND+MORE=MONEY** | | |
|---|---|---|---|
| **Answer** | ***9567+1085=10652*** | | |
| **Time** | Min (ms) | Max(ms) | Avg(ms) |
| | 180 | 974 | 680 |

Table 5.1. Result of parallel genetic algorithm

## V. RESULT FOR COMPARISION ON GENETIC PARALLEL ALGORITHM AND EVOLUTIONARY ALGORITHM

This paper proposed an efficient evolutionary algorithm to solve decimal Cryptarithmetic problems and compares the proposed algorithm with parallel genetic algorithm to solve them. First we will review some basics of GAs and EA then the proposed algorithm will be formulated and discussed in detail.[10-21]

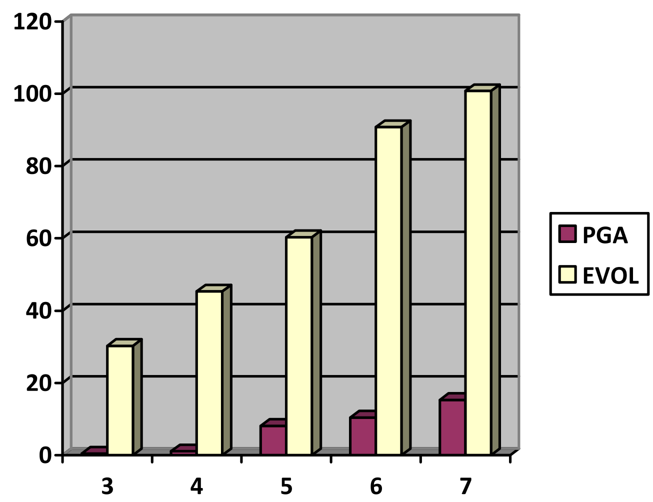


Figure 6.1 Comparison chart of time(ms)

## VI. CONCLUSION

This paper evaluates the two different algorithm to solve a cryptarithmetic problem in the terms of their execution times and shows the result using a chart to compare the execution time which indicates that the Parallel Genetic algorithm takes lesser time than evolutionary algorithm.

## REFERENCES


[1] H. E. Dudeney, in Strand Magazine vol. 68 (July 1924), pp. 97 and 214.

[2] Vinod Goel, Sketches of thought, MIT Press, 1995, pp. 87 and 88.

[3] Van Hentenryck, P., 1995. Constraint solving for combinatorial search problems: A tutorial. In: Montanari, U., Rossi, F. (Eds.), Proceedings of The First Conference of Principles & Practice of CP, Lecture Notes in Computer Science, vol. 976. Springer, Berlin, pp. 564±587.

[4] Simonis, H., 1996. A problem classi®cation scheme for unitedomain constraint solving (tutorial paper given at PACT'96). In: Second International Conference on the Practical Applications of Constraint Technology.

[5] Bonnie Averbach and Orin Chein, Problem Solving Through Recreational Mathematics, Courier Dover Publications, 2000, pp. 156.

[6] Abu Sayef Md. Ishaque, Md. Bahlul Haider, Muhammad Al Mahmud Wasid, Shah Mohammed Alaul, Md. Kamrul Hassan, Tanveer Ahsan, Md. Shamsul Alam: "An Evolutionary Algorithm to Solve Cryptarithmetic Problem", International Conference on Computational Intelligence 2004: 494-496.

[7] David Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley, Reading, MA 1989.

[8] Mariusz Nowostawski, Riccardo Poli, "Parallel Genetic Algorithm Taxonomy", KES'99.

[9] Alippi, C., Filho, J.L.R., Treleaven, P.C. (1994), "Genetic-Algorithm Programming Environments", IEEE Trans. Computer, June 1994.

[10] http://www.vincehuston.org/teaching/alphametics.html

[11] http://www.dcode.fr/cryptarithm-solver

[12] https://mysteriesexplored.wordpress.com/2011/08/24/amazing-encryption-technology-in-ancient-india-the-katapayadi-shankya/