

VIETNAM NATIONAL UNIVERSITY
HO CHI MINH UNIVERSITY OF SCIENCE



PROJECT 1 – CRYPTARITHMETIC PROBLEM IN AI

ARTIFICIAL INTELLIGENCE

21CLC07

- | | | |
|----|--------------------|----------|
| 1. | Trịnh Hạnh | 21127040 |
| 2. | Nguyễn Nhật Truyền | 21127191 |
| 3. | Trần Đỗ Anh Khoa | 21127321 |
| 4. | Nguyễn Khánh Hoàng | 21127612 |

Contents

1	MEMBER INFORMATION	2
2	ASSIGNMENT	2
3	COMPLETENESS	2
4	PROGRAM SPECIFICATION	3
5	ALGORITHM SPECIFICATION	4
5.1	Exhaustive backtracking:	4
5.2	CSP backtracking	4
5.3	Genetic Algorithm	6
6	COMPARISON	6
7	EXPERIMENT RESULT	7
7.1	Level 1:	7
7.2	Level 2:	8
7.3	Level 3:	10
7.4	Level 4:	11
7.5	Level 4.5:	12
8	REFERENCES	12

1 MEMBER INFORMATION

No.	Student ID	Full name
1	21127040	Trịnh Hạnh
2	21127191	Nguyễn Nhật Truyền
3	21127321	Trần Đỗ Anh Khoa
4	21127612	Nguyễn Khánh Hoàng

2 ASSIGNMENT

No.	Task	Member
1	Find solution for the problem	All
2	Handle file I/O of the program	Nguyễn Nhật Truyền
3	Process input	Trần Đỗ Anh Khoa
4	Solve using exhaustive backtracking	All
5	Solve using CSP back-tracking	Trần Đỗ Anh Khoa Trịnh Hạnh Nguyễn Nhật Truyền
6	Solve using GA	Nguyễn Khánh Hoàng
7	Generate test case for each level	All
8	Testing for exhaustive backtracking	Nguyễn Nhật Truyền Trịnh Hạnh
9	Testing for CSP back-tracking	Nguyễn Nhật Truyền Trịnh Hạnh
10	Testing for GA	Nguyễn Nhật Truyền Trịnh Hạnh
11	Structuring folder	Trần Đỗ Anh Khoa
12	Write report	Trần Đỗ Anh Khoa Nguyễn Khánh Hoàng

3 COMPLETENESS

Level	Completeness	Notes
1	100%	Exhaustive search may run with extra time (all test cases pass under four minutes)
2	100%	Exhaustive search may run with extra time (all test cases pass under four minutes)
3	100%	Exhaustive search may run with extra time (all test cases pass under four minutes)
4	100%	Exhaustive search may run with extra time (all test cases pass under four minutes) CSP backtracking cannot run with multiple multiplication

4 PROGRAM SPECIFICATION

- **Programming language:** Python 3.11.4
- **Folder structure:**
 - The submission includes a source folder containing the source code. Inside the source folder, the structure is organized as follows:

```
source
├─CSP_1_2_3.py
├─CSP_1_2_3.txt
├─CSP_1_2_3_out.txt
├─CSP_4.py
├─CSP_4.txt
├─CSP_4_out.txt
├─Exhaustive.py
├─exhaustive.txt
├─exhaustive_out.txt
├─GA.py
├─GA.txt
├─GA_out.txt
```

- **Usage:**
 - You can add input and display output of each algorithm by modifying the relevant file text
 - Below are the steps to follow using the command prompt:
 - * Step 1: Navigate to the source folder using the command prompt.
 - * Step 2: Execute the following command with your input file located in the source folder:
python3 main.py If the above command doesn't work, you can try the alternative command:
py main.py
 - * Step 3: The program will execute by asking which algorithm to be used. You can choose 1 - Exhaustive search, 2 - CSP for the first three levels, 3 - CSP for level four, 4 - Genetic algorithm. If there are no solution found, the program will write to specific file the string "No solution found", else, the program will write the solution in values alphabet's order

5 ALGORITHM SPECIFICATION

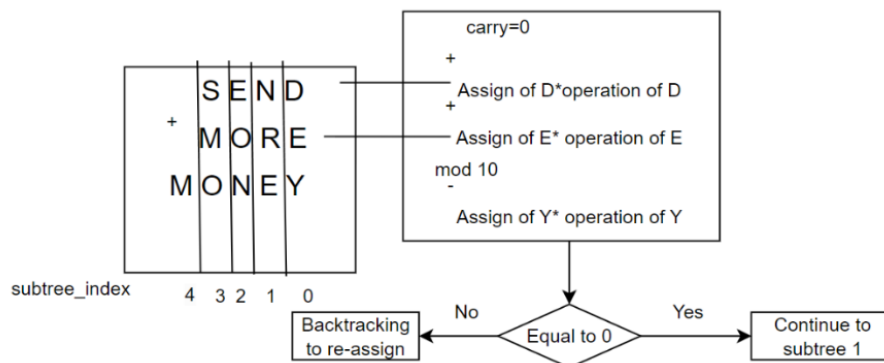
Pre-condition: In this problem, we know that the valid input must contain the number of distinct characters equal or less than 10, so we assume that when the number is greater than 10, there will be no solution found.

5.1 Exhaustive backtracking:

- The algorithm is built based on the backtracking search algorithm in theoretical course, with some simple constraint to give team a insight into this problem
- We will assign each variable with a domain. After assigning all variables, we will execute the evaluation to determine if this assignment is correct or not. If the assignment is correct, the algorithm will return the solution, else it will backtrack to the previous state and assign another domain until there is no assignment option.
- This algorithm is quite similar to brute force algorithm (try to generate all possible assignments), but we add some constraint to prune any invalid branch. Those constraint are:
 - The first character of each word must not equal to 0
 - All character must be assigned different number (we use inference forward checking to implement this)

5.2 CSP backtracking

- In this algorithm, we divide this problem into smaller problems, called subtree, and assign for each subtree. The concept is similar to do the plus operation in math.
- Assignment must be executed from right to left. To be more precise, if we want to assign the i -th subtree, we must solve the $i+1$ -th subtree to determine the value of carry. Solving step by step until assigning subtree at index 0, we can completely solve the problem. By this way, we can solve with equation including multiple plus, minus operation (level 1,2)
- In the program, we use two variables: `sub_tree` and `scope` to calculate the problem. `sub_tree` represents all the variables that contribute to the operation, and the `scope` represents the total of each variable that contributes to the operation. For example, consider the equation is `SEND+MORE=MONEY`: the first `sub_tree` will consist of three variables D,E,Y, and the scope of this `sub_tree` for D, E is 1, Y is -1 (swap side of the result in the equation). Consider equation is `E+E-E-E-E=K`, scope for E is $1+1-1-1-1=-1$, and for K is -1



- In terms of parentheses, we can solve it easily by standardizing the input equation. Loop through the equation, whenever we face the “(”, we will check if the pre-operation is minus or not. If pre-operation is minus, we will reverse all operations covered by the parentheses, and this problem becomes level 1, 2 problem.
- The constraint for first tree level, for equation SEND+MORE=MONEY would be:
 - All_diff(S,E,N,D,M,O,R,Y)
 - S !=0, M!=0
 - D+E=Y+10*X1
 - N+R+X1=E+10*X2
 - E+O+X2=N+10*X3
 - S+M+X3=O+10*X4
 - M=X4
- At level four, in this algorithm we just can implement a single multiple operation (2 operand) due to the problem with multiplying between two blocks of parentheses. The concept is similar to the first three levels, but we need to modify the data to have an appropriate solution. All we want to do is change the multiple into many plus operations.

$$\begin{array}{r}
 \begin{array}{r}
 \times \quad \text{TAKE} \\
 \quad \text{ORE} \\
 \hline
 \end{array} \\
 \begin{array}{r}
 \quad \text{TAKE} \\
 \quad \text{EEEE} \\
 + \\
 \quad \text{TAKE} \\
 \quad \text{RRRR} \\
 + \\
 \quad \text{TAKE} \\
 \quad \text{OOOO} \\
 \hline
 \text{MONEY Y}
 \end{array}
 \end{array}$$

5.3 Genetic Algorithm

- In this algorithm, we implemented it based on the template of the genetic algorithm. But there need to be some modification in order to solve this problem

– **Encoding the assignment:**

The initial part of using genetic algorithm is to represent the solution into chromosomes. In this problem, we use an char array, with the index representing the assigning number of the value. The array length must equal or less than 10, and in the case there are less than 10 unique characters, there need to add “-”, representing the empty cell (not assign this index to any value). An example of chromosome is:

```
S E N D M O R Y - -
0 1 2 3 4 5 6 7 8 9
```

– **Fitness function:**

The fitness function indicates the degree of correctness of a chromosome. In this problem, we use the min-fitness, which means the closer to 0, the better this will be. The evaluation function is: $ABS(RESULT - EQUATION)$. If the fitness of any chromosome is equal to 0, there will be the result

– **Mutation:**

Mutation operator randomly generates two index between 0 and 9 and exchanges the content of the cells in these two indices of the chromosome. Note that we need to avoid the illegal exchange, which means it should not allow an exchange that lead the assignment of first character to 0, not exchange the same index or exchange two empty cell

– **Making new generation:**

In this problem, we use asexual genetic algorithm, therefore, we do not have a crossover operator. Instead, to make new generation, we choose some best chromosome by random wheel in the current generation, add some new chromosome by mutate selected gen to the population, and take n best gen for new generation

6 COMPARISON

	Exhaustive backtracking	CSP backtracking	Genetic Algorithm
Performance	Slow, especially when need to assign “9”	Fast	Fast, but struggle in some test case
Memory	Manage one assignment	Manage one assignment	Manage mutiple assignments
Completeness	Can solve all equation,including division, two side equation(using <code>eval()</code>)	Cannot solve complex equation with multiple	Can solve all equation,including division, two side equation(using <code>eval()</code>)

7 EXPERIMENT RESULT

7.1 Level 1:

	Test case	Result		
		Exhaustive backtracking	CSP backtracking	Genetic Algorithm
1	SEND+MORE=MONEY	DEMNORSY 75160892	DEMNORSY 75160892	DEMNORSY 75160892
2	TWO-ONE=ONE	ENOTW 13246	ENOTW 17254	ENOTW 23486
3	COUPLE+COUPLE=QUARTET	ACELOPQRTU 0642591783	ACELOPQRTU 0642591783	No solution
4	YOUR+YOU=HEART	AEHORTUY 30146829	AEHORTUY 30146829	AEHORTUY 30146829
5	SIN-COS=UNITY	No solution	No solution	No solution

	Test case	Time to solve(ms)		
		Exhaustive backtracking	CSP backtracking	Genetic Algorithm
1	SEND+MORE=MONEY	54192.8783	42.8561	11.6831
2	TWO-ONE=ONE	118.006	0.9971	3.986
3	COUPLE+COUPLE=QUARTET	16385.7144	2.0043	Time limit error
4	YOUR+YOU=HEART	24535.9361	13.5058	1168580.7116
5	SIN-COS=UNITY	69189.2529	149.3766	Time limit error

7.2 Level 2:

	Test case	Result		
		Exhaustive backtracking	CSP backtracking	Genetic Algorithm
1	AA+BB+CC=ABC	ABC 198	ABC 198	ABC 198
2	SO+MANY+MORE+MEN+ SEEM+TO+SAY+THAT+ THEY+MAY+SOON+TRY +TO+STAY+AT+HOME+ SO+AS+TO+SEE+OR+HEAR+ THE+SAME+ONE+MAN+ TRY+TO+MEET+THE+ TEAM+ON+THE+MOON+ AS+HE+HAS+AT+THE+ OTHER+TEN=TESTS	AEHMNORSTY 7052618394	AEHMNORSTY 7052618394	No solution
3	ONE+ONE+ONE+ONE+ ONE+ONE-ONE=FIVE	EFINOV 013725	EFINOV 021345	EFINOV 517432
4	IN+ARCTIC+TERRAIN+AN+ ANCIENT-EERIE+ICE-TRACT+ I-ENTER+A-TRANCE =FLATIANA	ACEFILNRT 978135026	ACEFILNRT 978135026	No solution
5	AN+ACCELERATING+ INFERENTIAL+ENGINEERING +TALE+ELITE+GRANT+FEE+ ET+CETERA=INTELLIGENCE	No solution	No solution	No solution

	Test case	Result		
		Exhaustive backtracking	CSP backtracking	Genetic Algorithm
1	AA+BB+CC=ABC	3.0146	0.9965	11.0448
2	SO+MANY+MORE+MEN+ SEEM+TO+SAY+THAT+ THEY+MAY+SOON+TRY +TO+STAY+AT+HOME+ SO+AS+TO+SEE+OR+HEAR+ THE+SAME+ONE+MAN+ TRY+TO+MEET+THE+ TEAM+ON+THE+MOON+ AS+HE+HAS+AT+THE+ OTHER+TEN=TESTS	544410.969	2406.5812	Time limit error
3	ONE+ONE+ONE+ONE+ ONE+ONE-ONE=FIVE	20.9262	0.9974	1243.0845
4	IN+ARCTIC+TERRAIN+AN+ ANCIENT-EERIE+ICE-TRACT+ I-ENTER+A-TRANCE =FLATIANA	367021.8809	460.5791	Time limit error
5	AN+ACCELERATING+ INFERENTIAL+ENGINEERING +TALE+ELITE+GRANT+FEE+ ET+CETERA=INTELLIGENCE	510571.8248	2881.6489	Time limit error

7.3 Level 3:

	Test case	Result		
		Exhaustive backtracking	CSP backtracking	Genetic Algorithm
1	SEND+(MORE+MONEY)-OR+DIE=NUOYI	44714.9933	28.4469	Time limit error
2	SEND+MORE+MONEY-(OR+DIE)=NUOYI	21005.823	106.176	Time limit error
3	ONE+ONE-(ONE+ONE)+ONE=FIVE	8695.4611	3.991	Time limit error
4	ONE+ONE-(ONE+ONE-(ONE+ONE)-ONE)+ONE+ONE+ONE=FIVE	39.8726	1.1535	Time limit error
5	EARTH+AIR-(FIRE+WATER)=TURE	3424.4819	5.9847	Time limit error

	Test case	Time to solve (ms)		
		Exhaustive backtracking	CSP backtracking	Genetic Algorithm
1	SEND+(MORE+MONEY)-OR+DIE=NUOYI	DEIMNORSUY 1205639874	DEIMNORSUY 1295630874	No solution
2	SEND+MORE+MONEY-(OR+DIE)=NUOYI	DEIMNORSUY 1873490256	DEIMNORSUY 1873490256	No solution
3	ONE+ONE-(ONE+ONE)+ONE=FIVE	No solution	No solution	No solution
4	ONE+ONE-(ONE+ONE-(ONE+ONE)-ONE)+ONE+ONE+ONE=FIVE	EFINOV 013725	EFINOV 021345	No solution
5	EARTH+AIR-(FIRE+WATER)=TURE	AEFHIRTUW 135679482	AEFHIRTUW 258076194	No solution

7.4 Level 4:

	Test case	Result		
		Exhaustive backtracking	CSP backtracking	Genetic Algorithm
1	AB*BC=BDD	ABCD 1248	ABCD 1248	No solution
2	AA*BCD=DEECA	ABCDE 45620	ABCDE 45620	No solution
3	ODD*ODD=FREAKY	ADEFKORY 37842659	ADEFKORY 37842659	No solution
4	TWO*TWO=SQUARE	AEOQRSTUW 297546810	AEOQRSTUW 364217895	No solution
5	ABC*D=DBAS	No solution	No solution	No solution

	Test case	Time to solve (ms)		
		Exhaustive backtracking	CSP backtracking	Genetic Algorithm
1	AB*BC=BDD	15.9453	0.9964	10.3935
2	AA*BCD=DEECA	456.6581	0.9963	Time limit error
3	ODD*ODD=FREAKY	32514.9092	3.4942	Time limit error
4	TWO*TWO=SQUARE	67979.0307	8.6165	Time limit error
5	ABC*D=DBAS	971.5296	16.7155	Time limit error

7.5 Level 4.5:

	Test case	Result		Time to solve (ms)	
		Exhaustive backtracking	Genetic Algorithm	Exhaustive backtracking	Genetic Algorithm
1	$(A*A+B*B)*(C+D)-E*E*E-(F*G)=H$	ABCDEFGH 12493568	AGFBDECH 15867439	204.3865	29.8239
2	$(A+C*D-B)*(A*E)-(A*A-B)=G$	ABCDEG 162349	ADECGB 194238	450.1082	7.7683
3	$(AA*AA+BB*CCC)-(ABC+BEF)=ABFGC$	No solution	No solution	14457.0317	15180.7089
4	$(SNED+MOER)*MONEY=SMOERG$	No solution	No solution	369528.6653	15069.4548
5	$AA*(C+D)=EFG$	ACDEFG 235176	DGECFA 751462	1195.3342	45.2371

8 REFERENCES

- [Cryptarithmic Problem in AI](#)
- Solving Cryptarithmic Problems Using Parallel Genetic Algorithm - Reza Abbasian
- An Evolutionary Algorithm to Solve Cryptarithmic Problem - Abu Sayef Md. Ishaque, Md. Bahlul Haider, Muhammad Al Mahmud Wasid
- Theoretical slides