

# **MCT-242L COMPUTER PROGRAMMING-I**

## **SEMESTER PROJECT** **N KNIGHT PROBLEM**



Submitted By:                      Muhammad Anss                      2022-MC-01

Submitted to:                      Mam Qurrat-ul-Ain

DEPARTMENT OF MECHATRONICS & CONTROL ENGINEERING

University of Engineering and Technology, Lahore.

January 7, 2023

# **Table Of Contents:-**

- Introduction
- Genetic Algorithm
- Possible Applications
- Pseudo Code
- Result
- Things I Learn
- Conclusion

## **INTRODUCTION:-**

Using a defined algorithm helps us to solve many complex problems. To learn how to use an algorithm we were assigned to solve a problem n knight problem which is what will be the orientation and number of knights that can be placed on the chess board so that the board is filled by the knight and their attacks using Genetic Algorithm.

## **Genetic Algorithm:**

It is a type of search algorithm that helps us to find the best solution to a problem.

Here the generic Genetic Algorithm to solve any problem.

function Genetic-Algorithm( population, Fitness-fn) returns an individual

inputs: population -> set of individuals with different genes

Fitness-fn -> a function that measures the fitness of an individual required survival

repeat

parents <-- Select parents from Population

(Survival of fittest)

population <-- Offsprings created by parents

(cross-over, mutation)

repeat until desired fit individual is obtained

return the best individual in the population

## **Possibe Applications:**

Here our possible applications of this problem

- it can be used in the army where a particular troop or a weapon can attack to somewhere similar to a knight can attack so it will tell Esther minimum number of troops and their orientation that can cover the area to be attacked
- it can also be used in placing the set lights which emits infrared waves or any waves of frequency such that wave generated by neighbouring satellite can superimpose meaning they can perform interference with each other so this code will tell us the minimum number of satellites and their orientation which will cover the targeted area by the waves

## **Pseudo Code:**

Here's the Pseudo code of my code

Pseudo Code of n\_KNIGHT\_problem using Genetic Algorithm

Declaration of arrays like boards(3D array), chromosome(2D array), fitness(1D array) and sol

Initialization of some macros `__BOARD_SIZE__` to 8 (or any number whose answer is required), `__EMPTY__` to ' ', `__KNIGHT__` to 'O', `__ATTACK__` to 'X', `__POP__` to 200, `__MAX_KNIGHT__` to a number which would calculate from the formula  $2 * (((__BOARD\_SIZE__ * __BOARD\_SIZE__) / 9) + 4)$

Declaration and Initialization of variable like `nParents` to `__POP__ / 2` and trying knights to  $((__BOARD\_SIZE__ * __BOARD\_SIZE__) / 9) + 4$ , `isOver` to 0, `iter` to 0 and `maxIter` to 5000

Initialize all the elements of chromosome(2D array) to a random number excluding the number which will result in placing the knight in first or last column or first row or last row  
And make sure no 2 elements in 1D array are same

Reset all the elements of `boards`(3D array) to `__EMPTY__`

Convert random numbers in chromosome array to row and column by dividing and taking remainder operator on that number and place there knight

Place attacks of each knight by giving relative coordinates of knights (2,1),(2,-1), (-2,1),(-2,-1), (1,2),(1,-2), (-1,2)and(-1,-2)

Calculate No. of spaces in each board and put that in fitness array

Sort in ascending order the fitness and also sort the chromosome array on the basis of fitness

Check if first element of fitness is 0

If Yes

Print Board

Else

Select the first half of the population as parents

Give even childrens the even values of even parents

Give even childrens the odd values of odd parents

Give odd childrens the odds values of even parents

Give odd childrens the even values of odd parents

Give random value at random position to every chromosome element by making sure that the random may not result in placing knight first row or last row or first column or last columns and no two values in that array are the same.

Add one plus value to `iter` variable

Repeat the process just after randome initialization of chromosome array until `iter` become equal to `maxIter`

Print “Solution not found”

## Result:

Here’s the most optimum solution I got from my code

```
-----  
| 1|2|3|4|5|6|7|8|  
-----  
1|X|X|X|X|X|X|X|  
-----  
2|X|X|O|X|X|X|X|  
-----  
3|X|X|O|O|X|O|O|X|  
-----  
4|X|X|X|X|X|O|X|X|  
-----  
5|X|X|O|X|X|X|X|X|  
-----  
6|X|O|O|X|O|O|X|X|  
-----  
7|X|X|X|X|X|O|X|X|  
-----  
8|X|X|X|X|X|X|X|X|  
-----
```

Knights were placed at the following positions:

(2,3),(3,3),(3,4),(3,6),(3,7),(4,6),  
(5,3),(6,2),(6,3),(6,5),(6,6)and (7,6),

No. of knights in this solution = 12

## Things I Learn:

Solving this problem opened a door to new learning beyond the coursework. Key Takeaways:

- Problem definition: I understood how to identify problems ideally suited for computers, not humans.
- Computer communication: I learned how to instruct computers to perform tedious or impossible tasks for humans, overcoming the communication barrier.
- Advanced C skills: I explored new C features like: Colorful outputs: Creating text with different colors. 3D data visualization: Understanding and manipulating 3D arrays.
- Header files: Mastering their creation and usage for modular programming. Dynamic animation: Simulating movement through screen clearing and repeated printing.

## **Conclusion:**

So overall I learned so much things from for this complex engineering problem it helped me improving my problem solving skills and programming skills and open gate for me to see the world in a different way and making computer my best friend to do my tasks