# COMPUTATIONAL INTELLIGENCE REPORT (2022-2023) (14-06-2023)


| | | | |
|---|---|---|---|
| **Name** | **Mohamed Khaled Hassan Aly** | **GitHub Account** | https://github.com/MoMido1 |
| **Surname** | **MOTRASH** | **Repository** | https://github.com/MoMido1/Computational_Intellegence_2022 |
| **Email** | S295805@studenti.polito.it | **Exam Date** | 14-06-2023 |


## TOTAL WORK THROUGH THE YEAR

| LABS | Details | Details | Contributors |
|---|---|---|---|
| **Lab1**<br>**Set Covering Problem** | Breadth First | Developed without any external resources | **Alone** |
| | Depth First | Developed without any external resources | **Alone** |
| | Greedy BF | Developed without any external resources | **Alone** |
| | A* | Developed without any external resources | **Alone** |
| | Link | https://github.com/MoMido1/Computational_Intellegence_2022/tree/main/Lab1 | |
| **Lab1_HillClimbing** | Description | Solving the same problem of set covering using the same problem function with the same available function. | |
| | Tweak | The tweak function removes an already covered element and places a new one that is not covered and searches for the new solution if there's any | **Alone** |
| | Type | 2 types of Hill Climbing were attempted: -<br>SS -> Steepest-step<br>FI -> First improvement | **Alone** |
| | All this code was developed without any external resource or assistance | | |
| | Link | https://github.com/MoMido1/Computational_Intellegence_2022/tree/main/Lab1_Hill_Climping | |
| **Lab2_setCovering+ea**<br>**Evolutionary Algorithms** | Description | Solving the set covering problem using the same problem specifications with Evolutionary Algorithms.<br>No external resources were used | |
| | Types | Different types of Algorithms were used: -<br>( 1 + 1 ) Algorithm<br>( 1 + $\lambda$ ) Algorithm<br>( 1 , $\lambda$ ) Algorithm<br>( $\mu$ , $\lambda$ ) Algorithm | **Alone** |
| | Link | https://github.com/MoMido1/Computational_Intellegence_2022/tree/main/lab2_setCovering%2Bea | |
| **GA Set Covering Problem**<br>**Genetic Algorithm** | Details | Applying genetic Algorithm for set covering problem and modifying the hyper parameters | |
| | Types of Operators | Several operators were used with different probability of selection in each generation:<br>1- Cross Over i.e **(prob = 0.1)**<br>2- Mutation i.e **(prob = 0.1)**<br>3- Elitism i.e **(prob = 0.8)** **(ALONE)** | **Inspired from Lecture Code** |
| | Parent Selection | Tournament selection (pressure = 2)<br>Random parent selection | **Inspired from Lecture Code** |
| | Link | https://github.com/MoMido1/Computational_Intellegence_2022/tree/main/GA_SetCovering_Problem | |
| **Lab3_NIM_Game** | Details | 4 tasks were attempted to play against random player | |
| | Expert Player | Using the Nim sum technique to decide | **Inspired from Lecture Code** |

| | | |
|---|---|---|
| Base-Nim | Here it makes a check based on a base-3 NIM sum | **Alone** |
| Min Max | Uses the min max strategy to find the best solution | **Inspired from Lecture Code** |
| Reinforcement Leaning | Using Markov Decision Process algorithm for learning | **Inspired from online code presented in lecture** |
| Link | https://github.com/MoMido1/Computational_Intellegence_2022/tree/main/Lab3_NIM_Game | |

## PROJECT (2 ALGORITHMS WERE ATTEMPTED)

| Quarto Game | details | Contribution |
|---|---|---|
| Expert Player | Places the piece in the best place possible to guarantee the win and if not, it will place it in a place to make the opponent not win in any case.<br>Then selects the piece for the opponent that won't guarantee for the opponent the win | **Alone** |
| Minmax Player | Uses Min Max algorithm for better selection of the piece and a better selection of the position. | **Alone** |
| Link | https://github.com/MoMido1/Computational_Intellegence_2022/tree/main/quarto | |