**MEMORIA**

**PRÁCTICA 4**

**ADINA HAN**

**YOUSSEF EL FAQIR EL RHAZOUI**

1. **NQueens**

La función de fitness está en la clase interna NQueensFitnessFunction de NQueensGenAlgoUtil que implementa la interfaz FitnessFunction. La función de fitness está relacionada con la variable booleana nonAttackingPair que se pone a false si hay parejas de reinas que se ataca y si no hay se le suma 1 a la función de fitness.

La función objetivo esta implementada en la clase NQueensGoalTest que implementa la interfaz GoalTest. El objetivo es representado por un valor booleano que devuelve el método isGoalState que indica si en el tablero de las n reinas no hay ninguna pareja de reinas que se ataquen.

La población inicial se genera mediante el método generateRandomIndividual, que devuelve una representación de los individuos. Los individuos se representan mediante una lista de enteros del tamaño del tablero. Esta lista está formada por enteros comprendidos entre 0 y el tamaño del tablero y son generados aleatoriamente.

El problema de las n reinas se representa en un tablero, que es un array bidimensional de enteros de 8x8 y las reinas se representa con el entero “1”, mientras las posiciones que están libres con “0”. La población está representada por un conjunto de individuos de tamaño 50 y la representación de los individuos se hace mediante una lista de enteros con tamaño igual al tamaño del tablero.

El algoritmo genético utiliza la selección por ruleta y esta implementada en el método randomSelection de la clase GeneticAlgorithm. Este método devuelve el individuo que cumple que al escoger un numero aleatorio, este sea menor que el valor de la función de fitness acumulado para cada individuo.

Una vez que estén seleccionados los individuos, se realiza el cruce para producir nuevos individuos que se insertaran en la siguiente generación. Se utiliza el cruce de un punto y la representación del hijo está dada por una parte que representa la cromosoma de uno de los padres desde el principio hasta ese punto que se ha elegido aleatoriamente, a la que se le concatena otra parte de la cromosoma del otro padre desde ese punto elegido hasta el final. El algoritmo genético inserta los hijos en la siguiente generación sin tener en cuenta si son mejores o no que sus padres porque utiliza el cruce de tipo destructivo.

El tipo de mutación que utiliza aima es el reemplazo aleatorio de un punto. La mutación se implementa en el método mutate de la clase GeneticAlgorithm y devuelve un individuo de la población después de que se le ha aplicado la mutación. Este método elige aleatoriamente el individuo a mutar de la población y también se elige aleatoriamente el valor por el cual será reemplazado.

Los valores que se pueden configurar para el problema de las n reinas en el algoritmo genético son: el tamaño del tablero, que en nuestro caso es 8, la probabilidad de mutación para el cual se ha elegido el valor de 0.15, el tamaño de la población se ha establecido en 50 y el tiempo máximo en milisegundos durante el cual el algoritmo se puede ejecutar (1000mls).

1. **Asignación de turnos a los exámenes CFI**
2. **Modificación del algoritmo genético de AIMA**
3. **Parte opcional**