

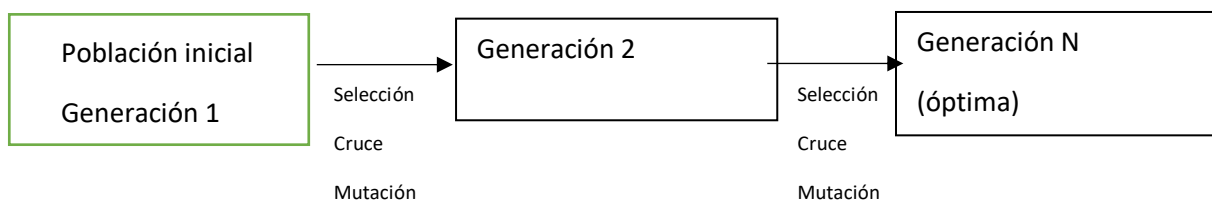
Algoritmos genéticos

Los algoritmos genéticos están basados en la Teoría Evolucionista de Darwin, la cual enuncia que aquellos individuos que mejor se adaptan al medio tienen más probabilidades de dejar descendencia y esta descendencia heredará sus genes, a su vez agrega, que las especies más fuertes tendrán adaptaciones genéticas que le permitirán adaptarse al medio, situación que hará que perduren en el tiempo.

- (1) La idea principal de un algoritmo genético es resolver problemas de optimización, donde se realiza una búsqueda guiada a partir de un conjunto inicial de posibles soluciones, a esto se lo denomina población inicial, el cual va evolucionando (mejorando) en cada iteración del algoritmo. A las iteraciones se las llama generaciones y la última generación incluye la mejor o las mejores soluciones al problema de optimización planteado. Cada posible solución al problema de conoce como individuo, y cada individuo codifica las variables independientes del problema de optimización. Estas variables representan los genes de la cadena cromosómica que representa a un individuo.

En cada iteración, se crean nuevos individuos, que son originados a partir de las operaciones genéticas: selección, cruce y mutación. En resumen, a partir de una población inicial, se generan iteraciones y el resultado final es la población final, la cual incluye las mejores soluciones al problema.

El siguiente esquema mostrará de una forma más clara lo antedicho.



Los pasos a seguir son los siguientes:

- 1) Se parte de una población inicial, la misma puede generarse en forma aleatoria
- 2) Se analiza la aptitud de cada individuo (de esa población individual), esto se hace calculando el fitness, lo que termina determinando un porcentaje.
- 3) Se crea la población 2, a partir de las operaciones de selección, cruce y mutación. Se seleccionan de la población 1 los que tengan el fitness más alto, se cruzan y se mutan, esto constituye la generación 2.
- 4) Esta secuencia se repetirá hasta que se cumple la condición que se indicó para que se corte la iteración. Aquí se devuelve la población final, con los individuos solución o el individuo solución.

Problema de las N reinas

Consiste en ubicar en un tablero de ajedrez de $n \times n$, una cantidad n de reinas de forma que éstas no se ataquen, es decir, no pueden estar en la misma línea horizontal, ni vertical ni diagonal. Se debe indicar, de acuerdo al valor de n , cual es la generación N y de cuántos individuos, es decir, soluciones está compuesta.

Este problema puede resolverse con Programación orientada a objetos. Uno de los pilares que tendrán lugar es la abstracción, pues se definirán los atributos y métodos de una clase y estos serán una abstracción del problema del mundo real.

Patrones de diseño

Los patrones de diseño son plantillas prefabricadas que se pueden personalizar y permiten resolver un problema de diseño recurrente en el código.

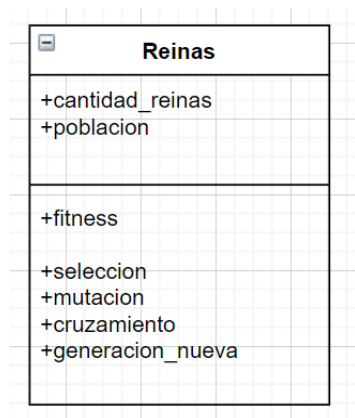
Los patrones de diseño se clasifican en 3 patrones de creación, patrones estructurales y patrones de comportamiento.

En código para resolver el problema de N-reinas se hace uso de estos patrones:

- Patrones creacionales: Builder. Permite construir objetos complejos paso a paso
- Patrones estructurales: Decorador . Permite añadir funcionalidades a objetos
- Patrones de comportamiento: iterator. Permite recorrer elementos de una colección.

Lenguaje Unificado de Modelado (UML)

Diagrama de clases



Resolución del problema

- 1) Se define la clase Reinas
 - 2) Para la clase Reina es necesario indicar la cantidad de reinas con la que se va a trabajar (debe coincidir con las dimensiones del tablero) y también se debe definir una secuencia de cromosomas, es decir de lugares donde se ubicarían las reinas.
-

Cabe destacar que podría trabajarse con matrices pero es más óptimo trabajar con vectores, pues se puede considerar que el número de posición dentro del vector indica la columna en la que se encuentra la reina y el valor numérico que se pone en el vector indica la posición en la fila.

- 3) Se analizan los cromosomas, es decir, se aplica fitness a ellos. Ese entrenamiento está definido por la función a maximizar, que en este caso esa función está dada por una probabilidad. Para cada vector que se generó en el paso 2 (como se trabaja con iteraciones, se generan muchos vectores), se calcula el número de pares de reinas que se atacan.

Previamente hay que considerar que hay un número máximo de reinas que se pueden atacar y está dado por la fórmula:

$númeroMáximoAtaques = \frac{n \cdot (n-1)}{2}$ siendo n la cantidad de reinas. A este número máximo se le resta para cada cromosoma la cantidad de reinas que efectivamente se atacan. (cuantas menos reinas se ataquen, mayor quedará este resultado) y luego se calcula la probabilidad que tiene cada cromosoma de sobrevivir.

- 4) Se seleccionan los cromosomas que tienen probabilidad más alta.
- 5) Se realiza el cruzamiento, esto es tomar parte de un cromosoma y colocarla en otro, lo mismo con los demás cromosomas, esto haría que posiblemente estos cromosomas mejoren.
- 6) Luego se aplica la mutación, la cual implica seleccionar un elemento del vector (cromosoma) y cambiarlo. Esta modificación debe darse dentro del rango de valores permitidos (de 1 a n)
- 7) Se vuelve a aplicar el fitness a estos nuevos cromosomas generados a partir de: selección, cruzamiento y mutación.
- 8) Si aún no se cumple con la condición de que la cantidad de ataques a las reinas sea 0, se continua con el procedimiento.
- 9) Luego el programa debe separar a todos los cromosomas que cumplen con que tienen cantidad de ataques cero. Estos serán los que formen la población final.