

Proyecto-3

Desarrollar un programa utilizando los conceptos de algoritmos genéticos para llegar a la solución, o al estado más cercano de la solución del juego de las ocho reinas.

Utilizar como la función de idoneidad, la "fitness function" descrita en la página 127 de nuestro texto. Esta función nos da el número de pares de reinas que no se atacan directa o indirectamente. Si el número de pares es de 28, hemos llegado a la configuración del tablero que representa una solución. Utilizar estrictamente la codificación para los estados del tablero como se define en el libro. Las cuadrillas en cada columna comienzan el conteo desde el 1 no 0.

Utilizar estrictamente el pseudocódigo definido en nuestro texto:

function GENETIC-ALGORITHM(*population*, FITNESS-FN) **returns** an individual

inputs: *population*, a set of individuals

FITNESS-FN, a function that measures the fitness of an individual

repeat

new_population \leftarrow empty set

for *i* = 1 **to** SIZE(*population*) **do**

x \leftarrow RANDOM-SELECTION(*population*, FITNESS-FN)

y \leftarrow RANDOM-SELECTION(*population*, FITNESS-FN)

child \leftarrow REPRODUCE(*x*, *y*)

if (small random probability) **then** *child* \leftarrow MUTATE(*child*)

add *child* to *new_population*

population \leftarrow *new_population*

until some individual is fit enough, or enough time has elapsed

return the best individual in *population*, according to FITNESS-FN

function REPRODUCE(*x*, *y*) **returns** an individual

inputs: *x*, *y*, parent individuals

n \leftarrow LENGTH(*x*); *c* \leftarrow random number from 1 to *n*

return APPEND(SUBSTRING(*x*, 1, *c*), SUBSTRING(*y*, *c* + 1, *n*))

Figure 4.8 A genetic algorithm. The algorithm is the same as the one diagrammed in Figure 4.6, with one variation: in this more popular version, each mating of two parents produces only one offspring, not two.

Eso implica que a diferencia del ejemplo del libro, cada apareamiento de dos padres produce un resultado (hijo) no dos.

El algoritmo debe ser implementado con las siguientes características:

- (a) Cada población debe tener 30 cromosomas
- (b) Los 30 cromosomas iniciales deben ser generados aleatoriamente según el algoritmo que ha sido proveído
- (c) Los cromosomas seleccionados para el apareamiento deben seguir al algoritmo de selección rueda de ruleta o "Roulette Wheel Selection Algorithm"; este algoritmo está descrito en las diapositivas del capítulo 4.
- (d) La población de los cromosomas debe ser sorteado en orden de decrecimiento; desde los cromosomas más aptos hasta los menos aptos, según la evaluación que las cromosomas tengan del "fitness function".
- (e) El numero de generaciones creadas debe ser 5000 máximo, o hasta que la configuración del tablero sea correcta; o sea hasta obtener una cromosoma con una evaluación del "fitness function" de 28.
- (f) La probabilidad de mutación de un gen en la cromosoma debe ser de 0.01
- (g) La posición del gen en el cromosoma que va a mutar debe ser seleccionado aleatoriamente.
- (h) El valor de la mutación del gen debe ser aleatoriamente escogido entre los números validos para el gen, o sea los valores integrales en este intervalo: [1,8]

Detalles de la implementación:

- (1) Utilizar el algoritmo "Fisher-Yates shuffling algorithm", que es incluido como parte de las especificaciones del proyecto para obtener la población aleatoria inicial de cromosomas.
- (2) Utilizar el algoritmo "Roulette Wheel Selection Algorithm" para seleccionar dos cromosomas para el apareamiento.