Sistemas Inteligentes 3º

Grupo 1.2

Ricardo Ramírez García

Práctica 1

Fecha:7/7/2018

Índice:

A)

Comienzan con un conjunto de estados generados a partir de un operador de selección (explicaré mas adelante)(población) Cada estado (individuo) está representado (codificado) como una cadena sobre un alfabeto finito. Cada elemento de un individuo es denominado gen. El valor heurístico de individuo es denominado fitness. La función fitness mide la calidad de los individuos.
Se utiliza una codificación entera de los individuos.
Los operadores genéticos que se utilizan son :

-Selección: encargados de escoger qué individuos van a disponer de oportunidades de reproducirse y cuáles no. Habrá individuos que aparezcan más de una vez e individuos que no aparezcan.

Los usados en Sudoku son:
Ruleta : Se eligen con probabilidad proporcional a su función fitness.
Torneo : Se establecen k torneos aleatorios entre parejas de individuos y se eligen los que ganan en cada torneo (mejor función fitness).

-Cruce: una vez seleccionados los individuos, éstos son recombinados para producir la descendencia que se insertará en la siguiente generación.

El usado en Sudoku es:
CruceSudoku

-Mutación: la mutación provoca que algunos de los genes de un individuo varíe su valor.

El usado en Sudoku es:
MutaciónSudoku

El algoritmo genético :
Se comienza con una población de 100 o 150 estados en este caso generados aleatoriamente a partir de la plantilla inicial. Iterar hasta que la población converge o alcanza una condición de parada que en nuestro caso es cuando el valor fitness encontrado es cero.

1. Se seleccionan k individuos de la población actual para crear la población intermedia.
2. Se eligen los individuos con una probabilidad (pc) para ser cruzados. Se emparejan, y para cada pareja se aplica el operador de cruce. Se obtienen los nuevos individuos, que sustituyen a los padres.
3. Con una probabilidad (pm) se mutan los genes de los individuos de la población actual.
4. Esta nueva población obtenida (mediante selección, cruce y mutación) forman la población inicial de la siguiente generación.
5. Se evalúan (con una función objetivo que valora a los individuos)los nuevos individuos para la selección en la siguiente iteración.

B)

3)

Dada esa definición, explica de qué manera se están inicializando los individuos en el AG propuesto.

4)

Explica el funcionamiento de los operadores de selección indicados en la sección “Ajuste del Algoritmo Genético”.

7)

Dada esa definición, explica de qué manera se están cruzando los individuos. o Mutación
El operador de mutación está definido en la función MutacionSudoku() proporcionado en el fichero “MutacionSudoku.txt”. La implementación se proporciona de forma completa salvo el casting necesario para recuperar en la variable genome la variable genérica g. Incorporar el código de “MutacionSudoku.txt” a “Sudoku.cpp”.

9)

Dada esa definición, explica de qué manera se están mutando los individuos

10)

Define y explica la condición de parada que utilizarás.

12)

Diseña y explica la función fitness que utilizarás. Recuerda, como se indica al comienzo de este guión, que una solución del sudoku no puede repetir en una misma fila, columna o subcuadrícula ninguno de los números

C)

Selector	TamPoblacion	ProbCruce	ProbMutacion	Caso-A1.txt	Caso-A2.txt	Caso-A3.txt	Caso-A4.txt	Caso-A5.txt	Fitness
GARouletteWheelSelecto	100	0.8	0.01	4	2	0	0	0	3
GARouletteWheelSelecto	100	0.8	0.05	4	0	0	0	0	4
GARouletteWheelSelecto	100	0.8	0.1	6	2	0	2	0	2
GARouletteWheelSelecto	100	0.8	0,125	4	2	0	2	4	1
GARouletteWheelSelecto	100	0.8	0.15	0	2	0	0	0	3
GARouletteWheelSelecto	100	0.85	0.01	2	0	4	6	0	2
GARouletteWheelSelecto	100	0.85	0.05	2	2	4	0	0	2
GARouletteWheelSelecto	100	0.85	0.1	0	4	6	0	0	3
GARouletteWheelSelecto	100	0.85	0,125	2	2	4	0	0	2
GARouletteWheelSelecto	100	0.85	0.15	6	0	6	6	2	1
GARouletteWheelSelecto	100	0.9	0.01	0	0	4	0	0	4
GARouletteWheelSelecto	100	0.9	0.05	5	0	0	0	0	4
GARouletteWheelSelecto	100	0.9	0.1	6	2	0	0	0	3
GARouletteWheelSelecto	100	0.9	0,125	2	2	4	6	6	0
GARouletteWheelSelecto	100	0.9	0.15	0	4	4	4	0	2
GARouletteWheelSelecto	100	0.95	0.01	4	0	0	0	2	3
GARouletteWheelSelecto	100	0.95	0.05	0	2	0	0	0	4
GARouletteWheelSelecto	100	0.95	0.1	0	2	0	0	0	4
GARouletteWheelSelecto	100	0.95	0,125	0	0	0	4	0	4
GARouletteWheelSelecto	100	0.95	0.15	10	2	0	2	4	1
GARouletteWheelSelecto	150	0.8	0.01	0	0	0	2	0	4
GARouletteWheelSelecto	150	0.8	0.05	0	2	0	0	0	4
GARouletteWheelSelecto	150	0.8	0.1	0	2	4	0	4	2
GARouletteWheelSelecto	150	0.8	0,125	4	2	0	6	0	2
GARouletteWheelSelecto	150	0.8	0.15	11	2	4	2	0	1
GARouletteWheelSelecto	150	0.85	0.01	6	0	0	0	0	4
GARouletteWheelSelecto	150	0.85	0.05	6	0	0	0	0	4
GARouletteWheelSelecto	150	0.85	0.1	0	0	4	0	0	4
GARouletteWheelSelecto	150	0.85	0,125	0	0	0	9	4	3
GARouletteWheelSelecto	150	0.85	0.15	2	2	2	4	2	0
GARouletteWheelSelecto	150	0.9	0.01	0	2	0	0	0	4
GARouletteWheelSelecto	150	0.9	0.05	0	0	0	5	0	4

Nfitness	ConRuleta	ConTorneo
ProbC 0.8	30	39
ProbC 0.85	30	42
ProbC 0.9	33	40
ProbC 0.95	31	40
ProbM 0.01	31	4
ProbM 0.05	29	32
ProbM 0.1	21	39
ProbM 0.125	17	37
ProbM 0.15	13	38
NFitPob100	52	72
NFitPob150	60	78
	347	461

D)Análisis de las pruebas de ajuste (esto es, el análisis de la tabla de resultados del apartado c). El objetivo de este análisis es obtener el comportamiento del software diseñado.

Analizaremos los datos por separado fijando un parámetro y viendo su eficiencia en función al valor Fitness obtenido para ese parámetro.

Selección con torneo es mejor para este problema que la selección por ruleta , para los datos obtenidos excepto cuando escogemos una probabilidad de mutación de 0.01 que es significativamente peor escoger torneo.

Tamaño de población si fijamos la selección de esta población para estas valores de tamaño de la población, la diferencia entre ambos no es relevante, es decir; si escogemos Ruleta , el tamaño de población con 100 el valor Fitness obtenido comparando con 150 es tan solo de 5 y si nos fijamos en la selección con torneos es menor la diferencia sólo de uno en el valor Fitness , por lo que podemos decir que no hay una gran diferencia entre escoger un tamaño de población u otro, así que como con 100 el numero de operaciones que hará el algoritmo serán menores , la mejor según este criterio es tamaño de población 100.

Probabilidad de cruce , con los casos obtenidos , pasa algo similar que con el tamaño de población que la diferencia de Fitness obtenido no es tan diferenciada como para destacar una ante las otras, pero podemos llegar a la misma conclusión , así que basándonos en esto podríamos escoger o 0.85 o 0.90

Probabilidad de mutación ,aquí esta claro según estos casos que la peor probabilidad de mutación es de 0.01 cuando escogemos la selección por torneo y esto se debe a que es muy baja , y esto provoca que el algoritmo genético tarde mas en converger ya que no obtenemos diversidad en los individuos, pero hemos de tener cuidado tampoco hemos de subir mucho esta probabilidad , porque sino nos acercaríamos mas a una busca aleatoria que a otra cosa , con todo esto podemos dividir este análisis de probabilidades de mutación fijando el parámetro de selección . Dentro de este con ruleta se comporta mejor cuando la probabilidad de mutación es mas baja y cuando es con torneo al contrario , para los valores comprendidos entre 0.01 y 0.15. Si hemos de escoger uno , elegiríamos 0,01 para la ruleta y 0.1 para torneo , por dos motivos , uno porque son los valores con mejores resultados al obtener Fitness 0 en mas casos y otra cuestión planteada es que al bajar la probabilidad hacemos que realice menos operaciones nuestro algoritmo genético.

E) A partir de este análisis, indicar qué valores de los parámetros debemos utilizar, y el protocolo para asignarlos, para que el software tenga un comportamiento aceptable. Denominar a esta sección “manual-Asignación”.

Manuel de asignación:

Los parámetro que debemos de usar para que el software tenga un buen comportamiento son:

C → Fichero.txt donde se encuentra el sudoku que queremos resolver.

P → Tamaño de población.

S → Selección.

Pc → Probabilidad de cruce.

Pm → Probabilidad de mutación.

SELECCIÓN	TAMPOBACIÓN	PROBCRUCE	PROBMUTACIÓN
GATournamentSelector	100 150	0.85 0.9	0.1

Suele funcionar mejor cuanto mayor es el tamaño de población y mayor la probabilidad de cruce pero los otros valores tienen un buen rendimiento

Sudoku.exe C P S Pc Pm

F) Para cada uno de los “Casos del Usuario” (ver sección correspondiente), indicar el protocolo seguido para resolverlo (siguiendo el “manual-Asignación”), el valor para cada parámetro, la solución obtenida y su fitness.

Vas a la terminal al direcorio donde se encuentra el ejecutable
y escribes la siguiente linea en la terminal.
Sudoku.exe C P S Pc Pm
Caso de prueba: Sudoku-1.txt
Parametros: - Tamaño población: 100
- Numero de generaciones: 12000
- Probabilidad cruce: 0.85
- Probabilidad mutacion: 0.1

El GA encuentra la solucion (4 5 8 3 6 7 9 1 2 1 6 3 9 2 5 7 4 8 2 9 7 8 4 1 5 6 3 8 4 5 2 9 6 3 7 1 7 3 2 5 1 8 6 9 4 6 1 9 4 7 3 2 8 5 5 7 1 6 8 2 4 3 9 9 2 6 1 3 4 8 5 7 3 8 4 7 5 9 1 2 6)
Valor fitness 0

Caso de prueba: Sudoku-2.txt
Parámetros: - Tamaño población: 100
- Numero de generaciones: 12000
- Probabilidad cruce: 0.9
- Probabilidad mutación: 0.1

El GA encuentra la solución (6 7 1 8 9 2 3 5 4 8 5 9 4 1 3 7 2 6 3 2 4 6 5 7 1 8 9 4 3 6 7 8 5 2 9 1 5 1 2 3 6 9 8 4 7 9 8 7 1 2 4 5 6 3 7 6 3 5 4 8 9 1 2 2 4 5 9 7 1 6 3 8 1 9 8 2 3 6 4 7 5)
Valor Fitness 0

Caso de prueba: Sudoku-3.txt
Parámetros: - Tamaño población: 150
- Numero de generaciones: 12000
- Probabilidad cruce: 0.9
- Probabilidad mutación: 0.1

El GA encuentra la solución (1 5 9 7 2 4 8 6 3 8 2 4 1 3 6 5 9 7 7 6 3 9 8 5 1 2 4 9 4 2 3 6 8 7 1 5 3 7 8 5 1 9 2 4 6 5 1 6 2 4 7 9 3 8 4 8 7 6 9 1 3 5 2 6 3 1 8 5 2 4 7 9 2 9 5 4 7 3 6 8 1)
Valor Fitness 0