Sistemas de Inteligencia Artificial

Trabajo Práctico Especial N°3

Algoritmos Genéticos

Integrantes:

- Alejandro Bezdjian, 52108
- Horacio Miguel Gómez, 50825
- Juan Pablo Orsay, 49373
- Sebastián Maio, 50386

Tabla de contenidos

Introducción

Se desea implementar un motor de algoritmos genéticos para obtener las mejores configuraciones de personajes para un juego.

El juego consiste en personajes que tienen cierta clase, ciertas propiedades y cierto equipamiento. El objetivo es lograr la mejor configuracion de ellas para optimizar el desempeño del personaje en el juego.

En el juego actualmente existen 4 tipos de personajes: Guerreros, Arqueros, Defensores y Asesinos.

Cada personaje debe lograr diferentes objetivos en cuanto a su desempeño en el ataque y la defensa.

Implementación

Para resolver el problema se implementó un programa que consta de varios módulos.

Models

Este módulo tiene todos los modelos que se implementaron para la representación del problema.
Los mismos son:

- Characters: esta clase representa un personaje, guardando sus atributos (fuerza, agilidad, etc…), sus modificadores a los mismos y además realiza las operaciones necesarias para calcular el aptitud (fitness) del individuo.
- Ítems: este modelo representa los objetos (ítems) que el individuo puede equiparse, junto con sus atributos.

Selection

Este módulo contiene las implementaciones de todos los métodos de selección vistos: elite, ruleta, universal, torneos (determinístico y estocástico), Boltzmann y ranking.

Crossover

Este módulo contiene las implementaciones de los algoritmos de cruza, ellos son: anular, un punto, dos puntos y uniforme.

Para los algoritmos de cruce es importante definir cómo serán los alelos. En nuestro caso, se decidió optar porque los alelos sean la altura y los ítems completos. No se utilizó una representación de bits o similar, debido a que era poco probable que un intercambio de bits que modifique las propiedades de un objeto genere un nuevo objeto que sea válido en el contexto del juego. Creemos que los items del juego no deberían poder ser modificados de cualquier manera, además esa es la forma que se manejan normalmente los RPGs.

Entonces quedan definidos los alelos: ALTURA, ARMADURA, BOTAS, GUANTES, CASCO y ARMA.

Genetic

El módulo contiene la implementación del algoritmo genético en sí. Recibe todos los parámetros de la configuración y corre el algoritmo genético mediante el método “natural selection”. En la configuración está incluido que algoritmos de cruza y selección debe utilizar en cada etapa, para esto hace llamados a los módulos descritos anteriormente.

Es parte del algoritmo generar nuevas generaciones, por lo que, este módulo se encarga de ello. Además calcula la aptitud de los individuos (promedio, máxima, mínima) por generación hasta alcanzar alguna condición de corte.

Mutaciones

Se decidió que las mutaciones se realicen por cada alelo, los cuales son intercambiados por otro alelo válido al azar (por ejemplo, se cambian los guantes por otros de la base de datos). La decisión de cambiar o no un alelo se hace de forma aleatoria con una cierta probabilidad que se pasa por parámetro.

Conclusiones

De acuerdo a los resultados obtenidos, no podemos identificar una combinación de métodos de selección y cruce que parezca mejor que otra. Esto se debe a:

- Todos los métodos contienen un factor de aleatoriedad.
- El dataset de ítems y las posibles alturas tienen un tamaño inmanejable.

El tamaño de los datos hace que no se puedan correr la cantidad suficiente de pruebas para obtener un resultado estadísticamente

significativo, por lo tanto se obtienen corridas buenas y malas con todas las combinaciones de algoritmos.

El tamaño de la población es importante dado que poblaciones muy chicas se estancan rápidamente, pero a partir de un cierto tamaño (100 en nuestro caso), no se notan diferencias significativas en los valores obtenidos.

Resultados

Todos los resultados fueron generados utilizando una misma configuración base y modificando ciertos parámetros (que serán explicados en breve) y utilizando una semilla en el generador de números aleatorios previo a correr cada prueba para que la diferencia entre cada una sea únicamente su configuración. Esta configuración se probo con poblaciones de tamaño 200 y 400 (con 50% de hijos).

Parámetros de las pruebas

Decidimos variar los siguientes parámetros que terminaron generando 400 distintas configuraciones:

- Función de *cruce*:
  - Cruce de un punto
  - Cruce de dos puntos
  - Cruce annular
  - Cruce uniforme parametrizado (p = 0.5)
- 1er método de selección:
  - Boltzmann
  - Torneo determinístico
  - Ruleta
  - Muestreo directo: Elite
- 2do método de selección:
  - Muestreo aleatorio
  - Universal
  - Ranking
  - Torneo estocástico
  - Muestreo directo: Elite
- Ratio del método de selección: 0, 0.25, 0.50, 0.75, 1.0
  - Un ratio de 0.25 significa que se seleccionará a un 25% de la población con el 1er método y un 75% de la población con el 2do método.
- Métodos de reemplazo (1ero y 2do):

- Para cada configuración se eligió utilizar el inverso que los usados para la selección, por lo que si el 1er método de selección es Boltzmann y el 2do es Universal, los métodos de reemplazo serán:
  - 1ro: Universal
  - 2do: Boltzmann
  - Ratio: 1 - RATIO_SELECCIÓN

Resultados

Cada archivo de configuración de prueba contiene un nombre formado por:

- Ratio de selección (0.25)
- Función de cruza (anular)
- 1er método de selección (boltzmann)
- 2do método de selección (Muestreo elite)

Por lo que si utilizamos de ejemplo los valores anteriores será generada la siguiente configuración: **0.0_annular_boltzmann_elite-sample.json**.

- Las configuraciones se encuentran dentro del directorio *genetics/tp3/data/configs/reports*
- Los resultados se encuentran dentro del directorio *genetics/tp3/data/results/reports*