

Informe Presa vs Depredador

Integrantes:

- Ernesto Alfonso Hernández C-312
- Abraham González Rivero C-312
- Jorge Soler González C-312

Descripción general:

Nuestro proyecto consiste en una simulación por turnos de un entorno y las diferentes interacciones que ocurren entre sus pobladores; presas y depredadores, con el objetivo de crear el modelo de depredador más apto para el medio. La idea general del proyecto es analizar el comportamiento de los habitantes del ecosistema.

Elementos de la Simulación:

La simulación consiste en un mapa generado, una cantidad de presas y depredadores, una cantidad máxima de comida por casillas del mapa y una probabilidad de que una casilla contenga comida y una cantidad de turnos a simular. En cada turno los agentes obtienen información del entorno, a partir de la cual realizan una acción determinada.

Agentes:

Existen dos modelos de agentes: presa y depredador, las características generales de cada modelo son:

- **Visión:** radio de la vecindad del mapa conocida por el agente desde su posición.
- **Movilidad:** cantidad de casillas máximas que puede avanzar en un turno.
- **Resistencia:** energía del agente, en caso de llegar a cero muere.
- **Fuerza:** parámetro que indica que tan poderoso es el agente, sirve para simular los enfrentamientos entre estos.

Los agentes cuentan con una base de conocimientos que influyen en las acciones que van tomando a lo largo de la simulación. Esta se basa en tres aspectos fundamentales: percepción del entorno, sentimientos internos y acciones motoras. Esto se modela a través de un mapa cognitivo difuso que funciona de la siguiente manera:

Es un digrafo ponderado donde el conjunto de los nodos representan los conceptos de cada individuo, cada uno de estos es un conjunto difuso y el conjunto de aristas representan el grado de implicación entre un concepto y otro. Dos conceptos unidos por una arista con pesos positivos significa que un concepto influye positivamente en el otro y viceversa.

Los conceptos que se manejan en una presa(depredador) son:

Percepción:

- Depredador(Presa) cerca/lejos.
- Comida cerca/lejos.
- Energía baja/alta.
- Comida local baja/alta.

Sentimientos:

- Miedo(Perseguir).
- Hambre.
- Satisfacción.
- Molestia.

Acciones:

- Huir(Cazar).
- Comer.
- Explorar.
- Esperar.
- Buscar comida.

Luego de que cada individuo obtenga información del mapa esta se fuzifica en base a un intervalo determinado por nosotros que indica el grado de pertenencia que tiene esta información al conjunto difuso de cada percepción. Entonces cada concepto se puede ver como la combinación lineal de los conceptos que tienen influencia en él. El valor resultante se normaliza con la función sigmoide($\lambda=1$).

Al concluir este proceso cada concepto contiene un valor que expresa de forma abstracta el pensamiento del individuo en ese instante. Para determinar que acción va realizar el agente se normalizan los valores de los conceptos de acciones y se genera un valor entre 0 y 1, la cota superior del intervalo al cual pertenezca este valor generado será la acción a realizar. De esta forma se logra simular de forma estocástica el proceso de toma de decisiones de los agentes.

En caso de enfrentamientos se normalizan las fuerzas de los individuos y se genera un valor entre 0 y 1, para determinar el vencedor, se compara con la fuerza normalizada del individuo más débil. Con esto se modela el hecho de que una presa sobreviva al ataque de un depredador. Agregar que la energía del vencedor se ve afectada en base a la fuerza del contrario.

En caso de muerte de un individuo el valor de comida para depredador de la casilla en que se encuentra aumenta en dependencia de los parámetros del tipo de agente.

Entorno:

El entorno consiste en un mapa conformado por casillas de diferente naturaleza, dígame agua, montaña, llanura, bosque y sabana cada una de ellas con sus propias características e implicaciones sobre el comportamiento de los agentes, por ejemplo: la visibilidad de un agente en un bosque es afectada; la movilidad de un agente se ve reducida considerablemente en una montaña.

Para la generación de este entorno se realizó una implementación del algoritmo CSP. Este se encuentra en el archivo `CSP.py` de la carpeta `iAalgorithms`. Las restricciones se representan a través de una clase `Constraints` la cual tiene un método `satisfied` el cual recibe como parámetros un diccionario de variable:dominio y determina si el dominio de la variable es válido para el conjunto de restricciones impuestas a esta. Para agregar nuevas restricciones es necesario crear una nueva clase que herede de `Constraints`. Las restricciones impuestas para la generación del mapa consiste en distribuir las celdas por 5 listas cada una con tipo de casillas distintos. Para la distribución de las celdas se obtiene una casilla aleatoria del mapa y mediante una adaptación de DFS se obtienen las casillas restantes para la lista en cuestión.

La comida se genera de manera aleatoria en las casillas del mapa, inicialmente solo hay comida para presas.

Problema y propuesta de solución

Teniendo en cuenta que el objetivo de nuestro proyecto es encontrar un depredador con condiciones óptimas para sobrevivir en un entorno con un número prefijado de presas y depredadores, decidimos utilizar el algoritmo de Simulated Annealing.

Nuestra implementación se encuentra en el archivo `SimulatedAnnealing.py` de la carpeta `iAalgorithms` y es una adaptación del algoritmo visto en clase. Para nuestra implementación se usa como vector solución la matriz de conceptos que representa el mapa cognitivo difuso del depredador, consideramos que dado que esta matriz está fuertemente involucrada en el proceso de toma de decisiones del individuo, si toma las mejores decisiones tendrá las mayores probabilidades de sobrevivir. La función para evaluar la solución no es más que la propia simulación y la puntuación de la solución la suma de la cantidad de turnos sobrevividos por el individuo en la simulación y su energía final. Para determinar el sucesor de una solución se modifican aleatoriamente una cantidad de componentes en un intervalo que depende de la temperatura del algoritmo, el nuevo valor de la componente será seleccionado aleatoriamente en una vecindad del valor antiguo que también depende de la temperatura; al apoyarnos en la temperatura logramos que a medida que esta disminuye la función esté menos dispuesta a realizar grandes modificaciones a la solución actual.