



Algoritmo Genético - Robot Plano de 5 ejes

10/12/2022

Javier Alonso Guerra

Computación Bioinspirada

Grado Ingeniería Informática en Ingeniería de Computadores

Introducción

Los algoritmos genéticos son aquellos algoritmos que basados en los procesos de evolución y selección natural, se centran en la búsqueda y optimización de parámetros para obtener la mejor solución posible a un problema concreto.

Es decir, partiendo de una población inicial de individuos (parámetros a optimizar), estos algoritmos mediante procesos que simulan las mutaciones y recombinaciones serán capaces de hacer evolucionar dicha población hacia la solución del problema.

Objetivos y Descripción del Problema

En esta ocasión, con un brazo robótico de 5 ejes y representado en un plano de dos dimensiones, trataremos de alcanzar una posición objetivo mediante el menor número de movimientos posibles haciendo uso de un algoritmo genético.

El brazo contará con 5 ejes de diferentes longitudes, para evitar posibles complicaciones tanto en la aplicación del algoritmo como en la representación de este, se ha decidido que el brazo tenga las siguientes características:

- Las longitudes de los diferentes ejes del robot deberán ser más pequeñas que las de sus predecesores.
- Cada eje puede moverse libremente 360 grados, esto implica que pueda darse algún caso en el que los ejes queden superpuestos.

Las posiciones de los 5 ejes, y por lo tanto la del brazo, se inicializan de forma aleatoria. Se definen un total de 200 posiciones del brazo diferentes, se ordenan en función de la cercanía al objetivo y solamente conservamos las 100 mejores.

Estas posiciones serán ahora nuestra población inicial de individuos las cuales mediante diferentes técnicas de selección, cruces, y mutaciones deberemos hacer evolucionar para alcanzar el objetivo.

Llegados a este punto, pueden darse dos casos, bien el brazo robótico ha sido capaz de capturar el objetivo o pasadas las 100 iteraciones del algoritmo el brazo no ha sido capaz de atraparlo, en cualquier caso el objetivo se mueve hacia una nueva posición y nuestra población de individuos (sin volver a inicializarse) debe volver a capturarlo y así sucesivamente.

Pruebas y Resultados

Con el objetivo de comprobar cómo afectan los diferentes parámetros y variaciones a los algoritmos genéticos, se han realizado diversas pruebas intentando conseguir el mejor resultado posible.

Mutaciones

En nuestro algoritmo, una mutación consiste en cambiar la posición de uno de los 5 ejes que conforman el brazo, escogido de forma aleatoria, por otra posición, también escogida de forma aleatoria

Para determinar qué probabilidad de mutación se ajusta mejor al algoritmo, se han realizado diferentes pruebas con diferentes probabilidades:

- Probabilidad de mutación **baja**: Se ha considerado una probabilidad de mutación baja cuando solo se realizan mutaciones al 10% o menos de la población, con una mutación baja nuestro algoritmo será menos propenso a explorar nuevos caminos, lo que puede acarrear soluciones con mínimos locales.
- Probabilidad de mutación **alta**: Se ha considerado una probabilidad de mutación alta cuando más del 50% de la población sufre mutaciones, con una mutación alta nuestros algoritmos no se quedarán atascados en mínimos locales, pero puede que la solución tarde mucho más en converger.
- Probabilidad de mutación **media**: Se ha considerado una probabilidad de mutación media cuando entre el 10% y 50% de la población sufren mutaciones, es una opción intermedia a las dos anteriores ya que por lo general evita mejor los mínimos locales y converge rápidamente en una solución.

A priori puede parecer que una mutación con probabilidad media es la mejor opción, pero no olvidemos que un algoritmo genético depende de muchos otros factores y que dependiendo de su implementación, puede ser que nos interese una mayor o menor probabilidad de mutación.

Elitismo

El elitismo en un algoritmo genético, consiste en guardar sin modificar a los mejores individuos de cada interacción.

Se han realizado diferentes pruebas aplicando elitismo guardando a los mejores individuos de cada generación, esto sirve a modo de "*checkpoint*", ya que en el caso que una generación no mejore a la anterior, siempre conservaremos a los mejores individuos de la generación pasada sobre los que probar diferentes evoluciones.

Por lo general, el uso de elitismo hace que el algoritmo converja más rápido que otro algoritmo que no lo aplique, pero también cabe destacar que un elitismo muy alto con

respecto al número de población puede hacer que el algoritmo quede atascado en mínimos locales, es por ello que a través de las pruebas realizadas, la opción más recomendable sería aplicar elitismo siempre y cuando este no sea superior al 5% con respecto de la población.

Cruces y Cortes

En un algoritmo genético, los cruces son la forma que tienen los individuos de evolucionar heredado del padre y de la madre, para definir qué partes heredan de uno y cuales de otro se definen los puntos de corte que dividirán a los individuos en diferentes trozos

En nuestro caso se han definido dos funciones diferentes para realizar los cortes:

- **Cruce básico con un corte:** A los individuos de entrada (el padre y la madre), se les realiza un corte a la mitad, a partir de dicho corte, los individuos de salida (los dos hijos que se generan) tendrán una mitad del padre y la otra mitad de la madre.

El problema que se genera con esta forma de realizar los cortes es que cada individuo tiene 5 componentes (correspondientes a los 5 ejes del brazo), y que al no poder dividir un componente a la mitad, con un solo corte se le estaría dando más importancia al padre o a la madre.

Para solucionar este problema recurriremos a una función aleatoria que nos generará 3 posibles casos:

- Se realiza el corte después de la tercera componente, en este caso el primer hijo tendrá más dependencia del padre, mientras que el segundo de la madre.
 - Se realiza el corte después de la segunda componente, es el caso contrario al anterior, el primer hijo tendrá más dependencia de la madre, mientras que el segundo del padre.
 - Se realiza una media de la tercera componente con los valores del padre y de la madre, en este caso ambos hijos tendrán la misma dependencia tanto del padre como de la madre.
- **Cruce ponderado sin cortes:** A diferencia del anterior, a los individuos de entrada (el padre y la madre), no se les realiza ningún corte, para realizar el cruce simplemente se multiplican los diferentes componentes por unos pesos.

En nuestro caso, el primer hijo heredará un 75% del padre y el 25% restante de la madre, en el segundo hijo se da el caso contrario, un 25% del padre y un 75% de la madre.

A la hora de comparar el rendimiento ambos métodos, parecen comportarse de forma parecida, aunque el cruce ponderado parece mostrar cierta tendencia hacia mínimos locales.

Selección

A la hora de realizar un algoritmo genético no solo hay que tener en cuenta la forma en la que vamos a realizar los cortes y cruces entre los individuos de la población, también hay que tener en cuenta la forma en la que seleccionamos a dicho individuos.

Como en los casos anteriores, hemos realizado diferentes pruebas para comprobar cuáles de los siguientes métodos se ajusta mejor a nuestra solución.

- **Selección mediante el mejor:** En este caso se ordena la lista de individuos en función de la cercanía con el objetivo, de esta forma siempre se van seleccionando las mejores parejas, esta solución es la más fácil de entender e implementar pero tiene la desventaja de que puede tender hacia mínimos locales.
- **Selección mediante torneo:** Se crean parejas de individuos de forma aleatoria y se van seleccionando aquellas que están más cerca de cumplir el objetivo, de esta forma al ser las parejas seleccionadas de forma aleatoria hay una mayor variedad a la hora de realizar los cruces y por lo tanto tienden a evitarse los mínimos locales.

Las diferentes pruebas realizadas parecen indicar que pese a que la selección mediante el mejor encuentra las soluciones, este no tiene ninguna ventaja acerca de realizar la selección mediante torneo que funciona de manera mucho más eficiente

Conclusiones y Problemas

I. Conclusiones

Tras todas las pruebas realizadas tanto en mutación, elitismo, cruces y cortes como la función de selección, podemos concluir que prácticamente todas las combinaciones anteriormente mencionadas encuentran la solución en menos de 100 iteraciones, sin embargo la combinación que lo hace de manera más rápida y que a priori evita los mínimos locales de manera más efectiva es :

- Una mutación **media-alta** alrededor del 40%.
- Haciendo uso de **elitismo**, guardando siempre las 3 mejores soluciones.
- Realizando el **cruce básico** de 1 corte
- Realizando la selección mediante **torneo**.

Con esta combinación conseguimos que el brazo robótico alcance el objetivo en 15 iteraciones aproximadamente.

II. Problemas

Durante la realización de las pruebas se ha detectado un error que causa que el algoritmo en momentos puntuales se quede atascado en un mínimo local cuando el objetivo se sitúa en el tercer cuadrante.

Este problema se ha dado con todas las combinaciones probadas, por lo que probablemente se trate de un error a la hora de llevar la representación del brazo a 2 dimensiones.

Aun así este error se ha dado de formas puntuales y no ha afectado a la realización de las pruebas del algoritmo genético.