

Diseño de un sistema de riego eficiente

A.Badilla-Olivas, Brandon Mora, Gabriel Molina

7 de julio de 2023

Enunciado del problema

Imagine que esta trabajando para una cooperativa agrícola, y se le ha asignado la tarea de diseñar un sistema de riego eficiente para un campo grande e irregularmente formado. El campo tiene diferentes condiciones del suelo y diferentes partes del campo necesitan diferentes cantidades de agua.

Componentes del problema

- Número de aspersores: Determinar el número óptimo de aspersores.
- Posicionamiento de los aspersores: Determinar los mejores lugares para colocar los aspersores.
- Cobertura del terreno: Dada una configuración de aspersores, determinar que tan bien se cubre el campo.

Función de uso de agua de un aspersor

$$\text{Uso de agua} = 180 * \text{aspersores_tipo_1} + 350 * \text{aspersores_tipo_2} + 400 * \text{aspersores_tipo_3} + 650 * \text{aspersores_tipo_4} + 800 * \text{aspersores_tipo_5}$$

Función de cobertura del terreno

Dada una matriz de terreno A de $n \times m$ y una matriz de aspersores S de $n \times m$ se irriga el terreno con respecto a cada aspersor (posicion y tipo) finalmente se calcula la cobertura del terreno como:

```
for i : 0 in n:  
  for j : 0 in m:  
    cobertura = cobertura + abs(A[i][j])
```

Función objetivo

Objetivo = $\alpha * \text{uso de agua} + \beta * \text{Cobertura del terreno}$

Solución con Whale Optimization Algorithm

El Whale Optimization Algorithm (WOA) puede ayudar a encontrar la mejor solución a este problema. Las variables de decisión en este problema (es decir, el número, las posiciones y los tipos del aspersor) se pueden codificar en un vector de solución, y el WOA se puede utilizar para buscar el vector de solución que minimiza el uso de agua mientras cumple con todos los requisitos de riego.

Inicialización

El WOA comienza inicializando una población de posibles soluciones (search agents). En este caso, cada solución especificaría una cierta configuración de aspersores (una matriz, donde cada fila es un vector con la posición del aspersor y el tipo $[x, y, \text{tipo}]$).

Exploración y Explotación (continuación)

El WOA utiliza tres operadores para explorar y explotar el espacio de soluciones. Estos son: buscar, encirclar, ataca (red de burbujas). Según las funciones de búsqueda, encirclar y atacar, el WOA actualiza las soluciones en cada iteración. Estas, definidas en el paper original de la metaheurística, funcionan con multiplicación de vectores. Pues normalmente se trabaja con vectores de soluciones. Sin embargo dado que las soluciones en este problema son matrices, se debe adaptar la metaheurística ligeramente el algoritmo para que funcione con matrices.

Modelo de Fuerza Bruta y Heurística Voraz

El objetivo es determinar las posiciones y tipos óptimos de aspersores que se colocarán en un terreno dado. El algoritmo toma dos entradas: terrain y sprinklers.

- terrain es una matriz 2D que representa el terreno donde se colocarán los aspersores.
- sprinklers es una matriz 1D que representa la cantidad de aspersores disponibles de cada tipo.

Algoritmo de Fuerza Bruta para el problema de riego (Parte 1)

- La función `brute_force_algorithm` toma como entrada el terreno (`terrain`) y los aspersores disponibles (`sprinklers`). El terreno se copia para mantenerse al comparar qué tan viable son las soluciones creadas.
- La función interna `generate_sprinkler_combinations` se encarga de generar todas las combinaciones posibles de colocaciones de aspersores en el terreno.
- Se verifica si no hay más aspersores disponibles. En ese caso, se genera una combinación vacía y se devuelve. De lo contrario, se selecciona el tipo de aspersor actual.

Algoritmo de Fuerza Bruta para el problema de riego

(Parte 2)

- Se obtiene una copia de los aspersores restantes, excluyendo el tipo actual y se itera sobre cada posición en el terreno. Para cada posición, se itera sobre cada columna en el terreno.
- Se selecciona un aspersor candidato, representado por una matriz de NumPy con las coordenadas de la posición y el tipo de aspersor.
- Se verifica si el aspersor candidato es válido utilizando una función externa llamada `check_if_valid_sprinkler`.

Algoritmo de Fuerza Bruta para el problema de riego

(Parte 3)

- Si el aspersor candidato es válido, se realiza una llamada recursiva a `generate_sprinkler_combinations` con los aspersores restantes y se itera sobre cada combinación generada.
- Se devuelve la combinación actual junto con las combinaciones generadas en las llamadas recursivas.
- Se itera sobre todas las combinaciones generadas por `generate_sprinkler_combinations`.

Algoritmo de Fuerza Bruta para el problema de riego

(Parte 4)

- En cada iteración, se calcula la aptitud de la combinación actual utilizando una función externa llamada `fitness_function`.
- Si la aptitud de la combinación actual es mejor que la mejor aptitud encontrada hasta el momento, se actualiza la mejor aptitud y se guarda la combinación como la mejor solución.
- Finalmente, se devuelve la mejor solución encontrada como una matriz de coordenadas.

Heurística Voraz para el problema de riego (Parte 1)

- Inicializa una lista vacía `solution` para almacenar las posiciones y tipos de los aspersores colocados.
- Establece la variable `best_fitness` en infinito para realizar un seguimiento del mejor valor de aptitud encontrado hasta el momento.
- Crea una copia del terreno original.

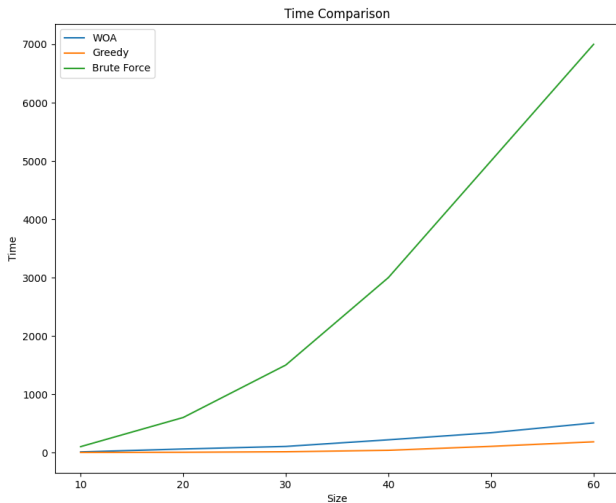
Heurística Voraz para el problema de riego (Parte 2)

- Itera sobre cada posición en el terreno y para cada posición, considera cada tipo de aspersor.
- Verifica si agregarlo a la solución mantiene una solución válida.
- Si la colocación del aspersor candidato es válida, calcula su valor de aptitud.
- Si el valor de aptitud es mejor que el mejor valor de aptitud actual, actualiza el mejor valor de aptitud y establece el aspersor candidato como el mejor aspersor.
- Después de evaluar todos los aspersores candidatos posibles, si no se encuentra alguno que mejore la solución actual, sale del bucle.

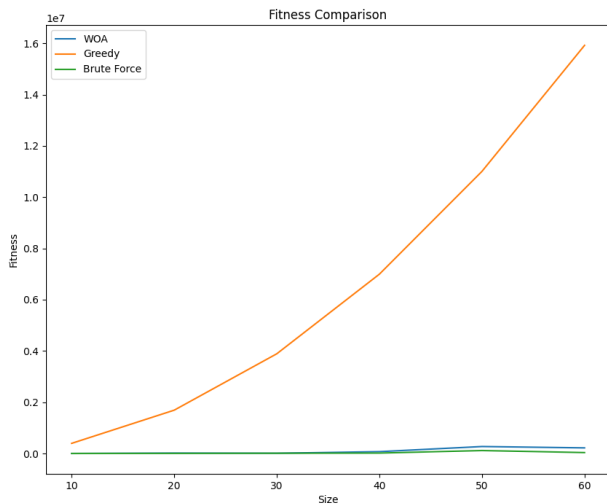
Heurística Voraz para el problema de riego (Parte 3)

- De lo contrario, actualiza el terreno para reflejar la colocación del mejor aspersor.
- Agrega el mejor aspersor a la lista de solución y reduce la cantidad de aspersores disponibles de ese tipo.
- Repite los pasos 4 al 11 hasta que no haya más colocaciones válidas de aspersores.
- Finalmente, devuelve la lista de solución.

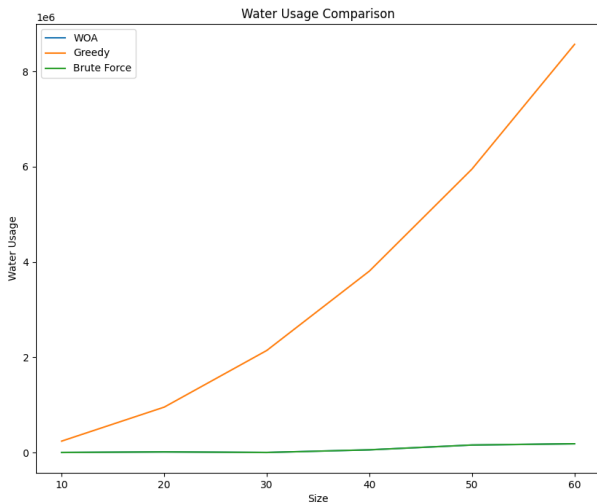
Comparaciones



Comparaciones



Comparaciones



Comparaciones

