

Actividad en clase - Algoritmos aleatorios

Victor misael Escalante Alvarado, A01741176

Parte 1

Programa un algoritmo random sampler para producir 1000 soluciones aleatorias del problema del bandwidth para ash85.txt

```
file_dir = "/Users/misa_v/Library/CloudStorage/OneDrive-InstitutoTecnologicoyde
numero_nodos, mejor_solucion, numero_aristas, aristas = read_file(file_dir)

num_iteraciones = 1000

distancias = []
mejor_distancia = float('inf')
peor_distancia = 0

for i in range(num_iteraciones):
    solution, distancia = random_solution(aristas, numero_nodos)
    if distancia < mejor_distancia:
        mejor_distancia = distancia
    if distancia > peor_distancia:
        peor_distancia = distancia
    distancias.append(distancia)

print('#####')
print('Mejor distancia: ', mejor_distancia)
print('Peor distancia: ', peor_distancia)
print('Distancia promedio: ', sum(distancias) / len(distancias))
print('#####')
✓ 0.0s

#####
Mejor distancia: 71
Peor distancia: 84
Distancia promedio: 79.974
#####
```

Responde:

¿Cómo se podría mejorar este algoritmo para producir mejores soluciones con las mismas iteraciones? Podríamos empezar con los randoms y saber si es posible mejorar poco a poco la solución o si conviene más reemplazar toda la solución completa

¿Cuántas soluciones posibles hay? Para todo número n de nodos existe un número $n!/2$ que solucionan el problema (no los mejores)

¿Cuál es la dificultad del problema?

Parte 2

Crea un algoritmo que realice el random walk para el problema del bandwidth para ash85.txt .Ejecuta tu programa con 1000 iteraciones para cada ejecución.

```
absolut_solution = []
absolut_distance = float('inf')
all_distance = []
worst_distance = 0

for i in range(1000):
    final_solution, final_distance, all_distan, worst = local_search(random_sol_inicial, aristas, 1000)
    if final_distance < absolut_distance:
        absolut_distance = final_distance
        absolut_solution = final_solution

    if worst > worst_distance:
        worst_distance = worst

    all_distance.extend(all_distan)

print('#####')
print('Mejor distancia: ', absolut_distance)
print('Peor distancia: ', worst_distance)
print('Distancia promecio: ', sum(all_distance) / len(all_distance))
print('#####')

60] ✓ 25.7s

#####
Mejor distancia: 65
Peor distancia: 84
Distancia promecio: 79.94418
#####
```

Responde

Compáralo con el random sampler. ¿Cuál es mejor y porque?

En base a todas las iteraciones que he realizado con los 2, trate de correr los samples completos varias veces y he notado que el random sampler tiende a ganar en ciertas ocasiones (corri ambos con un millon), sin embargo al momento de hacerlos con menos la tendencia de el random walker es que gana pues con pocos pasos conviene mas mejorar la solucion antes que solo reiniciar

Enlace a colab :

[https://drive.google.com/file/d/1ECa8dAW--L_YU4R2RhqCB_pyMyw19_8O/view?usp=sharing]