

8 Vezir Probleminin Tepe Tırmanma Algoritması ile Çözümü

1. Problemin Tanımı

8x8'lik satranç tahtasına hiçbirini birbirini yemeyecek şekilde tepe tırmanma yöntemi kullanarak 8 adet vezirin dizilmesi.

2. Programın Özellikleri

Taşların konumlarının etkin şekilde tutulması için 8x1 dizi kullanıldı.

h değerinin düşük başlaması için dizinin her indeksindeki değer birbirinden farklı olacak şekilde başlatıldı.

h değeri daha iyiye gitmediği anda anda taşlar rastgele yeniden dizilerek çözüme yeniden başlandı.

3. Programın Ekran Görüntüleri

Rastgele taş dizimini gerçekleştiren metod:

```
def random_restart():
    numbers = [1, 2, 3, 4, 5, 6, 7, 8]
    location_list = []

    for i in range(len(numbers)):
        rnd_index = random.randint(0, len(numbers)-1)
        location_list.append(numbers[rnd_index])
        numbers.remove(numbers[rnd_index])

    return location_list
```

h değerinin bulunması:

```
def find_h(k):
    h = 0
    for i in range(len(k)):
        for j in range(i+1, len(k)):
            if k[i] == k[j] or abs(k[i]-k[j]) == abs(i-j):
                h += 1

    return h
```

Tepe tırmanma algoritması:

```
def hill_climb():
    random_restart_c = 0
    swap_c = 0
    t_start = process_time()

    location_list = random_restart()

    h = find_h(location_list)
    new_location_list = []
    min_h = 1000
    while h != 0:
        stuck = 1
        for i in range(8):
            original = location_list[i]
            for j in range(8):
                location_list[i] = j+1
                current_h = find_h(location_list)
                if current_h < min_h:
                    stuck = 0
                    min_h = current_h
                    new_location_list = location_list[:]

            location_list[i] = original

        swap_c += 1
        location_list = new_location_list[:]
        h = find_h(location_list)

        if stuck == 1:
            swap_c -= 1
            random_restart_c += 1
            new_location_list = []
            min_h = 1000
            location_list = random_restart()

    t_end = process_time()
    return [random_restart_c, swap_c, t_end - t_start]
```

Main:

```
def main():
    print("Random Restart Count, Swap Count, Time")
    for i in range(15):
        print(hill_climb())
```

Program Çıktısı:

```
Random Restart Count, Swap Count, Time
[13, 32, 0.03125]
[9, 30, 0.046875]
[0, 5, 0.015625]
[0, 5, 0.0]
[13, 35, 0.03125]
[3, 11, 0.015625]
[3, 13, 0.0]
[5, 18, 0.03125]
[5, 15, 0.015625]
[5, 15, 0.015625]
[4, 15, 0.015625]
[0, 4, 0.0]
[0, 5, 0.0]
[11, 28, 0.03125]
[2, 7, 0.015625]
```