**Inteligência Artificial**

**Problema da Mínima Latência**



Bianca Faria Dutra Fragoso

Pedro Felipe Magalhães

**Introdução**

Esse trabalho implementa alguns algoritmos que chegam perto da melhor solução para o problema da mínima latência.

O MLP é uma variante do problema do caixeiro viajante, sendo definido a seguir. Seja G=(V, A) um grafo direcionado completo, onde V = {v0, …, vn} é o conjunto de vértices, v0 representa o depósito e os outros vértices, os clientes, e A={(i,j) : i, j ∈ V, i ≠ j} é o conjunto de arcos, sendo que cada um é associado com o tempo de viagem entre os vértices i e j. O objetivo do MLP é encontrar um circuito hamiltoniano em G que minimiza o tempo total de espera (latência) dos clientes. O tempo de latência do i-éssimo cliente, ou l(i), é a soma de todos os tempos de viagem do depósito até o i-éssimo cliente presente no circuito Hamiltoniano.

1. **Busca Gulosa com profundidade limitada e poda**
   1. O algoritmo
   2. Os resultados

|  |  |  |
| --- | --- | --- |
|  | Melhor solução já encontrada | Nossa melhor solução |
| *dantzig42* |  |  |
| *gr48* |  |  |
| *brazil58* |  |  |
| *gr120* |  |  |

* 1. O tempo

|  |  |  |
| --- | --- | --- |
|  | Tempo para achar melhor solução já encontrada | Tempo para acharmos nossa melhor solução |
| *dantzig42* |  |  |
| *gr48* |  |  |
| *brazil58* |  |  |
| *gr120* |  |  |

1. **Simulated Annealing**
   1. O algoritmo
      1. A solução inicial
      2. A temperatura
      3. A probabilidade
      4. As vizinhanças
         1. 2 – Opt
         2. Swap
   2. Os resultados

|  |  |  |
| --- | --- | --- |
|  | Melhor solução já encontrada | Nossa melhor solução |
| *dantzig42* |  |  |
| *gr48* |  |  |
| *brazil58* |  |  |
| *gr120* |  |  |
| *pa561* |  |  |

* 1. O tempo

|  |  |  |
| --- | --- | --- |
|  | Tempo para achar melhor solução já encontrada | Tempo para acharmos nossa melhor solução |
| *dantzig42* |  |  |
| *gr48* |  |  |
| *brazil58* |  |  |
| *gr120* |  |  |

1. **Outros algoritmos**

Alguns algoritmos foram implementados enquanto desenvolvíamos o trabalho, no entanto não deram resultados tão bons quanto os citados acima.

* 1. Busca gulosa usando Kruskal
  2. Busca Local