Projeto 2 IC

Aluno: Silvano Martins da Silva Junior - 12011BCC042

Artigo Escolhido

An Efficient Genetic Algorithm for Solving the Generalized Traveling Salesman Problem

---

Principais Elementos

Descrição do Problema

O artigo trata do Generalized Traveling Salesman Problem (GTSP), que é uma variação do problema clássico do Caixeiro Viajante (TSP). No GTSP, o objetivo é encontrar um ciclo de custo mínimo em um grafo onde os nós estão agrupados em cluster. O ciclo deve incluir exatamente um nó de cada cluster. Os problemas Generalized são tipicamente pertencem a classe NP-completo e são mais difíceis que o problema clássico, atualmente estão sendo bastante estudados por conta das possíveis aplicações no mundo real. Resumindo tudo isto, o GTSP é basicamente o problema de encontrar o custo minimo em um circuito Hamiltoniano, porém, tratamos os "nodes" como "node sets" (cluster).

Representação do Problema

O grafo $G = (V, E)$ possui V (conjunto de nós) e E (conjunto de arestas com custos), o conjunto V é particionado em m subconjuntos, chamados de clusters.

Representação do Indivíduo (Cromossoma)

Cada indivíduo é representado por uma tupla (C, N).
C é uma lista que representa a sequência de clusters visitados, descrita como um ciclo Hamiltoniano sobre os clusters.
N é um conjunto de nós selecionados de cada cluster, ou seja $N = (N_1, N_2, \ldots, N_m)$ onde N_i é o nó selecionado do cluster C_i.

Operadores Genéticos

1. Crossover
   - Mantém a ordem e a posição dos símbolos (clusters) de um dos pais, preservando a sequência relativa dos símbolos do outro pai.
2. Mutação
   - Intra-cluster mutation: Um cluster é escolhido aleatoriamente, e o nó selecionado é trocado por outro nó dentro do mesmo cluster.
   - Inter-cluster mutation: Dois clusters têm suas posições trocadas aleatoriamente na sequência C.

Função de Adaptação

A função de adaptação calcula o custo total do ciclo de Hamilton generalizado com base no custo das arestas entre os nós selecionados de cada cluster.

Parâmetros de Funcionamento do Algoritmo

Número de Iterações (Critério de Parada)

O número de gerações varia entre 400 e 1000, dependendo do experimento.

Tamanho da População e Offspring

- Tamanho da População: Definido como **5 vezes o número de clusters**.
- Tamanho do Offspring Minímo: Definido como **2 vezes o tamanho da população (λ = 2 * μ)**.

Taxa de Mutação

- A **probabilidade de mutação** foi definida em **5%**.

Benchmarks Utilizados

- O algoritmo foi testado em nove problemas de benchmark da **TSPLIB**, com instâncias que variam de **198 a 442 nós**, divididos em um número específico de clusters.

Forma de Apresentação dos Resultados e Medidas Estatísticas

- Os resultados são comparados com dois outros algoritmos meta-heurísticos: **Random-Key Genetic Algorithm (RK-GA)** e **Memetic Algorithm (MA)**.
- Os benchmarks incluem medidas de custo mínimo de diferentes instâncias e são apresentados em uma tabela que compara os resultados dos três algoritmos para cada instância testada

---

Código

Classe GTSP

- **evaluate:** Calcula o custo total do tour representado por um indivíduo (que consiste em uma ordem de clusters e uma seleção de nós). O custo é calculado somando as distâncias entre nós, conforme a ordem especificada e, finalmente, adicionando o custo de retorno ao ponto inicial.

initialize_population

Inicializa uma população de indivíduos aleatórios

- Uma ordem aleatória de clusters (`cluster_order`).
- Um conjunto de nós (`node_selection`), selecionados com base em uma distribuição de probabilidades inversamente proporcional ao custo dos nós dentro de cada cluster.

calculate_selection_probabilities

Calcula as probabilidades de seleção de nós dentro de um cluster, calculando a distância mínima para outros nós usando a matriz de custos.

crossover

Realiza a operação de crossover entre dois indivíduos (pais).

intra_cluster_mutation

Realiza uma mutação que troca um nó selecionado dentro de um cluster aleatório por outro nó dentro do mesmo cluster.

inter_cluster_mutation

Realiza uma mutação que troca a posição de dois clusters na ordem.

selection

Implementa um método de seleção por torneio binário para escolher os indivíduos que serão os pais da próxima geração.

genetic_algorithm

Função principal onde iremos chamar todas as outras para utilizar de fato o algoritmo (main).

load_tsplib_instance

Carrega uma instância do problema a partir de um arquivo TSPLIB e cria a matriz de custo, que armazena as distâncias entre todos os pares de nós.

create_clusters

Cria clusters a partir de uma lista de nós, distribuindo-os aleatoriamente em grupos.

Experimentos

Dados pelo autor:

Problem	*m*	*n*	*MA*	*RK-GA*	*GA*
40d198	40	198	10557	10557	10557
40kroa200	40	200	13406	13406	13406
40krob200	40	200	13111	13111	13111
46gr229	46	229	71641	71641	71832
53gil262	53	262	1013	1013	1014
60pr299	60	299	22615	22615	22618
80rd400	80	400	6361	6361	6389
84fl417	84	417	9651	9651	9651
89pcb442	89	442	21657	21657	21665

Obtidos:

d198 com 40 clusters e 198 nós:

```
Geração 998: Melhor Fitness = 9960.0
Geração 999: Melhor Fitness = 9960.0
Melhor Solução:  ([19, 23, 39, 8, 42, 13, 31, 5, 7, 36, 44, 6, 20, 35, 2, 43, 3, 38, 32, 17, 40, 26, 1,
 14, 30, 4, 48, 25, 46, 27, 21, 10, 22, 16, 41, 0, 24, 34, 37, 47, 12, 11, 9, 29, 18, 15, 33, 28, 45, 4
9], [130, 73, 76, 115, 42, 16, 70, 55, 36, 75, 161, 182, 177, 26, 59, 31, 135, 95, 38, 80, 66, 157, 141
, 64, 136, 69, 87, 156, 108, 62, 56, 22, 94, 32, 146, 81, 45, 145, 122, 51, 97, 131, 27, 105, 54, 109,
155, 178, 12, 91])
Valor total:  9960.0
```

kroa200 com 40 clusters e 200 nós:

```
Geração 997: Melhor Fitness = 13223.0
Geração 998: Melhor Fitness = 13223.0
Geração 999: Melhor Fitness = 13223.0
Melhor Solução:  ([4, 36, 8, 3, 24, 10, 22, 19, 7, 39, 23, 29, 1, 31, 13, 35, 9, 17, 27, 28, 37, 14, 16
, 38, 12, 2, 26, 30, 34, 5, 15, 20, 33, 21, 18, 25, 32, 11, 6, 0], [93, 27, 128, 10, 149, 57, 121, 66,
175, 67, 11, 174, 65, 160, 83, 192, 54, 120, 145, 188, 4, 85, 52, 46, 92, 1, 74, 75, 71, 40, 156, 123,
53, 101, 33, 20, 99, 152, 6, 18])
Valor total:  13223.0
```

krob200 com 40 clusters e 200 nós:

```
Geração 998: Melhor Fitness = 14799.0
Geração 999: Melhor Fitness = 14799.0
Melhor Solução:  ([28, 25, 2, 37, 5, 3, 19, 23, 16, 33, 12, 30, 36, 22, 20, 31, 38, 8, 10, 13, 14, 0, 1
, 17, 26, 7, 11, 21, 15, 4, 9, 27, 29, 34, 24, 18, 32, 6, 35, 39], [109, 120, 87, 79, 152, 188, 84, 159
, 181, 20, 72, 186, 69, 187, 126, 146, 150, 125, 22, 81, 70, 25, 198, 101, 23, 168, 183, 80, 133, 173,
40, 53, 58, 30, 88, 7, 39, 37, 49, 165])
Valor total:  14799.0
```

gr229 com 46 clusters e 229 nós

```
Geração 997: Melhor Fitness = 71848.0
Geração 998: Melhor Fitness = 71848.0
Geração 999: Melhor Fitness = 71848.0
Melhor Solução:  ([21, 41, 29, 38, 5, 12, 4, 8, 14, 2, 11, 34, 18, 37, 1, 35, 3, 13, 28, 40, 30, 10, 24
, 36, 27, 19, 23, 16, 0, 42, 32, 26, 6, 22, 7, 45, 31, 17, 44, 20, 9, 39, 15, 33, 43, 25], [101, 79, 12
9, 82, 135, 169, 126, 105, 134, 37, 60, 122, 163, 24, 132, 47, 93, 16, 102, 85, 91, 182, 108, 92, 61, 1
78, 114, 83, 57, 176, 59, 36, 117, 193, 112, 81, 66, 77, 174, 34, 58, 180, 106, 187, 89, 96])
Valor total:  71848.0
```

gil262 com 53 clusters e 262 nós:

```
Geração 998: Melhor Fitness = 1014.0
Geração 999: Melhor Fitness = 1014.0
Melhor Solução:  ([50, 53, 0, 16, 51, 31, 29, 32, 57, 52, 2, 20, 9, 64, 30, 11, 48, 58, 45, 12, 26, 33,
 19, 43, 44, 28, 8, 10, 13, 37, 3, 15, 22, 41, 40, 65, 46, 61, 34, 56, 14, 1, 38, 39, 5, 55, 18, 63, 47
, 21, 60, 35, 6, 62, 4, 36, 25, 59, 17, 49, 42, 7, 27, 54, 23, 24], [112, 225, 105, 243, 140, 90, 165,
229, 163, 17, 177, 113, 84, 44, 74, 43, 250, 6, 115, 198, 127, 261, 195, 73, 153, 235, 151, 125, 76, 70
, 136, 238, 191, 11, 38, 20, 230, 224, 58, 102, 33, 56, 256, 81, 137, 138, 21, 28, 167, 31, 15, 63, 75,
 99, 134, 27, 199, 216, 19, 262, 55, 106, 116, 194, 85, 246])
Valor total:  1014.0
```

pr299 com 60 clusters e 299 nós:

```
Geração 998: Melhor Fitness = 22618.0
Geração 999: Melhor Fitness = 22618.0
Melhor Solução:  ([14, 44, 59, 18, 58, 3, 24, 15, 25, 20, 53, 30, 73, 47, 54, 69, 17, 10, 2, 28, 12, 67
, 19, 52, 62, 39, 5, 57, 29, 4, 33, 63, 0, 55, 41, 9, 1, 68, 22, 56, 65, 13, 48, 11, 31, 21, 26, 72, 37
, 32, 61, 38, 71, 23, 34, 45, 43, 51, 60, 35, 64, 70, 42, 46, 8, 74, 36, 6, 7, 66, 16, 49, 27, 40, 50],
 [250, 136, 163, 83, 127, 43, 268, 266, 37, 162, 97, 183, 66, 253, 34, 82, 255, 89, 96, 62, 138, 186, 7
6, 231, 79, 102, 245, 40, 134, 111, 208, 185, 278, 179, 299, 69, 60, 275, 199, 41, 42, 202, 169, 8, 99,
 9, 38, 150, 182, 121, 33, 11, 45, 223, 14, 251, 74, 39, 91, 98, 19, 277, 46, 189, 105, 170, 261, 107,
100, 88, 194, 230, 273, 158, 36])
Valor total:  22618.0
```

rd400 80 clusters e 400 nós:

```
Geração 998: Melhor Fitness = 6392.0
Geração 999: Melhor Fitness = 6392.0
Melhor Solução:  ([75, 70, 11, 78, 44, 38, 29, 24, 57, 63, 6, 3, 27, 72, 62, 39, 53, 18, 71, 12, 49, 54
, 4, 66, 56, 15, 50, 23, 47, 48, 51, 33, 13, 30, 64, 31, 19, 61, 69, 7, 73, 17, 36, 21, 26, 20, 55, 10,
 46, 0, 35, 2, 60, 5, 8, 34, 43, 65, 37, 77, 16, 45, 74, 1, 22, 9, 67, 40, 68, 58, 14, 28, 52, 59, 25,
79, 42, 76, 32, 41], [251, 359, 293, 399, 40, 329, 180, 46, 245, 112, 130, 318, 349, 6, 339, 234, 69, 1
38, 375, 141, 80, 126, 85, 117, 143, 306, 27, 172, 15, 274, 36, 135, 9, 151, 363, 327, 131, 341, 270, 3
24, 377, 111, 39, 206, 75, 301, 311, 17, 173, 181, 282, 288, 21, 105, 55, 228, 258, 61, 212, 107, 62, 2
21, 81, 94, 142, 35, 237, 28, 52, 325, 63, 121, 267, 144, 102, 220, 304, 76, 370, 317])
Valor total:  6392.0
```

fl417 84 clusters e 417 nós:

```
Geração 998: Melhor Fitness = 9651.0
Geração 999: Melhor Fitness = 9651.0
Melhor Solução:  ([75, 38, 72, 62, 80, 96, 81, 34, 25, 3, 33, 11, 87, 76, 69, 26, 40, 28, 59, 60, 14, 2
7, 53, 0, 48, 56, 6, 9, 95, 99, 20, 102, 83, 51, 67, 24, 50, 70, 21, 94, 1, 78, 90, 30, 61, 55, 12, 8,
91, 57, 52, 74, 63, 13, 19, 23, 18, 93, 44, 43, 15, 58, 46, 31, 36, 103, 82, 64, 98, 101, 5, 17, 29, 10
4, 77, 22, 41, 68, 66, 92, 73, 88, 100, 39, 97, 4, 42, 85, 86, 49, 79, 35, 7, 54, 89, 71, 10, 47, 2, 65
, 84, 32, 45, 16, 37], [229, 300, 189, 265, 325, 201, 413, 65, 250, 278, 164, 258, 56, 161, 388, 127, 2
8, 193, 335, 158, 259, 344, 204, 337, 141, 264, 95, 359, 83, 62, 108, 2, 178, 285, 286, 240, 13, 12, 94
, 79, 78, 208, 351, 145, 125, 19, 129, 167, 236, 40, 117, 146, 55, 361, 51, 112, 384, 289, 151, 87, 403
, 132, 277, 38, 174, 163, 215, 118, 213, 66, 138, 162, 253, 223, 41, 97, 75, 383, 306, 238, 279, 290, 1
92, 144, 180, 349, 17, 98, 70, 160, 333, 287, 217, 150, 299, 284, 274, 326, 176, 262, 86, 177, 68, 315,
 364])
Valor total:  9651.0
```

pcb442 89 clusters e 442 nós:

```
Geração 998: Melhor Fitness = 21669.0
Geração 999: Melhor Fitness = 21669.0
Melhor Solução:  ([110, 83, 7, 80, 104, 1, 97, 65, 100, 98, 50, 9, 17, 96, 92, 58, 68, 107, 99, 48, 6,
23, 87, 69, 75, 61, 94, 32, 34, 49, 30, 20, 21, 12, 5, 29, 4, 51, 10, 8, 28, 95, 93, 16, 72, 52, 56, 10
9, 67, 33, 41, 15, 62, 76, 106, 60, 59, 74, 73, 84, 103, 89, 47, 53, 19, 0, 22, 37, 42, 66, 78, 46, 79,
 11, 81, 39, 91, 25, 40, 13, 71, 105, 70, 26, 18, 54, 3, 108, 43, 38, 24, 44, 31, 86, 88, 57, 64, 35, 1
02, 45, 36, 77, 85, 90, 101, 55, 27, 63, 2, 14, 82], [178, 435, 399, 38, 204, 246, 42, 440, 206, 320, 1
97, 22, 244, 145, 262, 226, 233, 296, 71, 166, 393, 175, 398, 11, 292, 147, 105, 136, 413, 248, 157, 37
1, 146, 207, 133, 134, 198, 191, 238, 64, 169, 411, 190, 234, 368, 112, 53, 177, 73, 158, 324, 194, 254
, 155, 39, 103, 216, 209, 439, 322, 298, 91, 420, 396, 159, 329, 118, 132, 20, 79, 179, 394, 232, 354,
294, 81, 306, 205, 130, 54, 312, 57, 264, 325, 316, 395, 334, 12, 333, 201, 127, 383, 153, 258, 101, 40
9, 176, 367, 327, 16, 304, 441, 124, 425, 434, 192, 303, 19, 173, 180, 328])
Valor total:  21669.0
```

Com isso obtemos a seguinte tabela final:

Problem	m	n	MA	RK-GA	GA	Meu
40d198	40	198	10557	10557	10557	9960
40kroa200	40	200	13406	13406	13406	13223
40krob200	40	200	13111	13111	13111	14799
46gr229	46	229	71641	71641	71832	71848
53gil262	53	262	1013	1013	1014	1575
60pr299	60	299	22615	22615	22618	22618
80rd400	80	400	6361	6361	6389	6392
84fl417	84	417	9651	9651	9651	9651
89pcb442	89	442	21657	21657	21665	21669

Conclusão

Apesar de não termos conseguido melhorar os resultados apresentados pelo autor no artigo, os valores obtidos pelo nosso código se aproximam significativamente dos resultados originais. Essa proximidade indica que a implementação está correta? Com um número maior de clusters produz soluções de alta qualidade, alinhadas com os métodos utilizados pelo autor. Embora existam ainda margens para ajustes e refinamentos no algoritmo, o desempenho atual é bastante satisfatório e demonstra a eficácia da abordagem adotada.