

Projeto 3 IC

Aluno: Silvano Martins da Silva Junior - 12011BCC042

Artigo Escolhido

AG: [An Efficient Genetic Algorithm for Solving the Generalized Traveling Salesman Problem](#)

PSO : [A Discrete Particle Swarm Optimization Algorithm for the Generalized Traveling Salesman Problem](#)

Principais Elementos

Descrição do Problema

Uma variante do problema do Traveling Salesman Problem (TSP), onde um tour não visita necessariamente todos os nós, é chamado de problema Generalized Traveling Salesman Problem (GTSP). Mais especificamente, o conjunto de N nós é dividido em m conjuntos ou clusters.

Solução Proposta

O algoritmo proposto é um PSO discreto que utiliza soluções pessoais e globais ótimas para refinar iterativamente os caminhos.

Função Objetivo

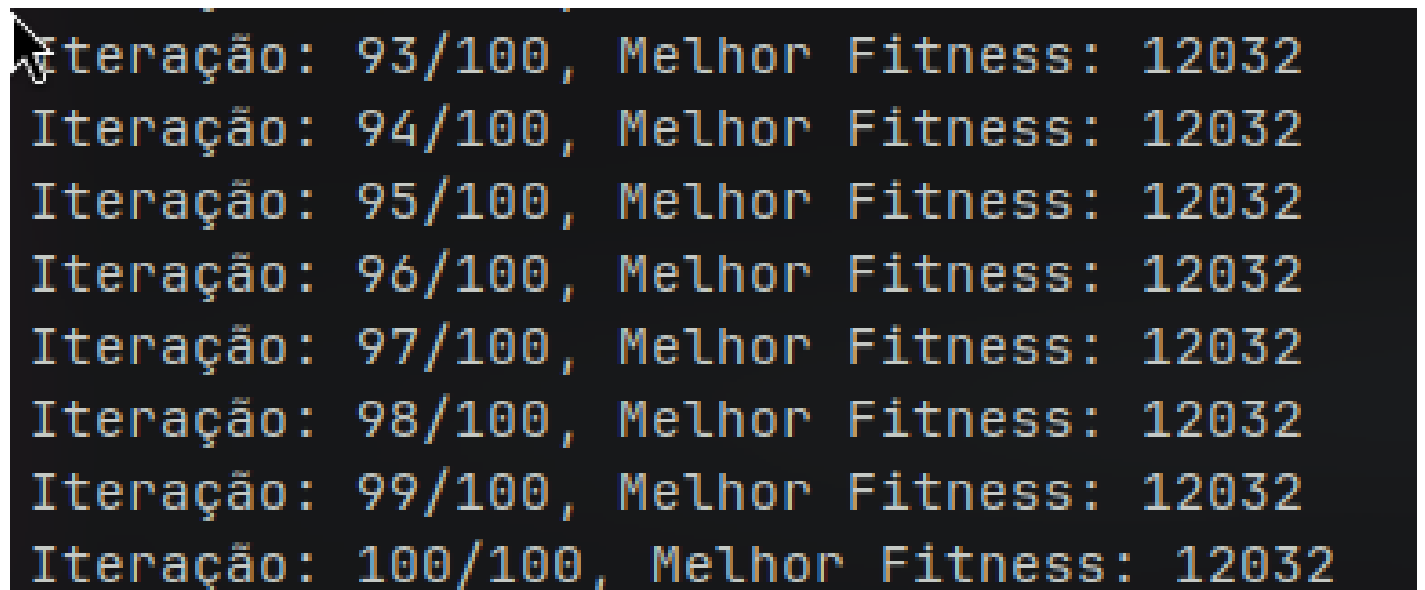
A função de objetivo é o comprimento total do tour, somando as distâncias entre os nós selecionados nos diferentes clusters.

Observações

- O artigo não diz nada a respeito do tamanho utilizado na população de partículas.
- O artigo do PSO utiliza do mesmo benchmark que o do AG
- O artigo diz ter utilizado todos os problemas do benchmark passado, porém, mostrou apenas 36 (?).

Experimentos

d198 com 40 clusters e 198 nós:



```
Iteração: 93/100, Melhor Fitness: 12032
Iteração: 94/100, Melhor Fitness: 12032
Iteração: 95/100, Melhor Fitness: 12032
Iteração: 96/100, Melhor Fitness: 12032
Iteração: 97/100, Melhor Fitness: 12032
Iteração: 98/100, Melhor Fitness: 12032
Iteração: 99/100, Melhor Fitness: 12032
Iteração: 100/100, Melhor Fitness: 12032
```

kroa200 com 40 clusters e 200 nós:

```
Iteração: 92/100, Melhor Fitness: 13481
Iteração: 93/100, Melhor Fitness: 13481
Iteração: 94/100, Melhor Fitness: 13481
Iteração: 95/100, Melhor Fitness: 13481
Iteração: 96/100, Melhor Fitness: 13481
Iteração: 97/100, Melhor Fitness: 13481
Iteração: 98/100, Melhor Fitness: 13481
Iteração: 99/100, Melhor Fitness: 13481
Iteração: 100/100, Melhor Fitness: 13481
```

krob200 com 40 clusters e 200 nós:

```
Iteração: 90/100, Melhor Fitness: 13111
Iteração: 91/100, Melhor Fitness: 13111
Iteração: 92/100, Melhor Fitness: 13111
Iteração: 93/100, Melhor Fitness: 13111
Iteração: 94/100, Melhor Fitness: 13111
Iteração: 95/100, Melhor Fitness: 13111
Iteração: 96/100, Melhor Fitness: 13111
Iteração: 97/100, Melhor Fitness: 13111
Iteração: 98/100, Melhor Fitness: 13111
Iteração: 99/100, Melhor Fitness: 13111
Iteração: 100/100, Melhor Fitness: 13111
```

gr229 com 46 clusters e 229 nós

```
Iteração: 92/100, Melhor Fitness: 82938
Iteração: 93/100, Melhor Fitness: 82938
Iteração: 94/100, Melhor Fitness: 82938
Iteração: 95/100, Melhor Fitness: 82938
Iteração: 96/100, Melhor Fitness: 82938
Iteração: 97/100, Melhor Fitness: 82938
Iteração: 98/100, Melhor Fitness: 82938
Iteração: 99/100, Melhor Fitness: 82938
Iteração: 100/100, Melhor Fitness: 82938
```

gil262 com 53 clusters e 262 nós:

```
Iteração: 92/100, Melhor Fitness: 2754
Iteração: 93/100, Melhor Fitness: 2754
Iteração: 94/100, Melhor Fitness: 2754
Iteração: 95/100, Melhor Fitness: 2754
Iteração: 96/100, Melhor Fitness: 2754
Iteração: 97/100, Melhor Fitness: 2754
Iteração: 98/100, Melhor Fitness: 2754
Iteração: 99/100, Melhor Fitness: 2754
Iteração: 100/100, Melhor Fitness: 2754
```

pr299 com 60 clusters e 299 nós:

```
Iteração: 90/100, Melhor Fitness: 24627
Iteração: 91/100, Melhor Fitness: 24627
Iteração: 92/100, Melhor Fitness: 24627
Iteração: 93/100, Melhor Fitness: 24627
Iteração: 94/100, Melhor Fitness: 24627
Iteração: 95/100, Melhor Fitness: 24627
Iteração: 96/100, Melhor Fitness: 24627
Iteração: 97/100, Melhor Fitness: 24627
Iteração: 98/100, Melhor Fitness: 24627
Iteração: 99/100, Melhor Fitness: 24627
Iteração: 100/100, Melhor Fitness: 24627
```

rd400 80 clusters e 400 nós:

```
Iteração: 92/100, Melhor Fitness: 13022
Iteração: 93/100, Melhor Fitness: 13022
Iteração: 94/100, Melhor Fitness: 13022
Iteração: 95/100, Melhor Fitness: 13022
Iteração: 96/100, Melhor Fitness: 13022
Iteração: 97/100, Melhor Fitness: 13022
Iteração: 98/100, Melhor Fitness: 13022
Iteração: 99/100, Melhor Fitness: 13022
Iteração: 100/100, Melhor Fitness: 13022
```

fl417 84 clusters e 417 nós:

```
Iteração: 92/100, Melhor Fitness: 15239
Iteração: 93/100, Melhor Fitness: 15239
Iteração: 94/100, Melhor Fitness: 15239
Iteração: 95/100, Melhor Fitness: 15239
Iteração: 96/100, Melhor Fitness: 15239
Iteração: 97/100, Melhor Fitness: 15239
Iteração: 98/100, Melhor Fitness: 15239
Iteração: 99/100, Melhor Fitness: 15239
Iteração: 100/100, Melhor Fitness: 15239
```

pcb442 89 clusters e 442 nós:

```
Iteração: 93/100, Melhor Fitness: 42913
Iteração: 94/100, Melhor Fitness: 42913
Iteração: 95/100, Melhor Fitness: 42913
Iteração: 96/100, Melhor Fitness: 42913
Iteração: 97/100, Melhor Fitness: 42913
Iteração: 98/100, Melhor Fitness: 42913
Iteração: 99/100, Melhor Fitness: 42913
Iteração: 100/100, Melhor Fitness: 42913
```

Com isso:

Problem	m	n	MA	RK-GA	GA	DPSO	Meu
40d198	40	198	10557	10557	10557	10557	12032
40kroa200	40	200	13406	13406	13406	13406	13481
40krob200	40	200	13111	13111	13111	13111	13111
46gr229	46	229	71641	71641	71832	71641	82938
53gil262	53	262	1013	1013	1014	1013	2754
60pr299	60	299	22615	22615	22618	25110	24627
80rd400	80	400	6361	6361	6389	13945	13022
84fl417	84	417	9651	9651	9651	10133	15239
89pcb442	89	442	21657	21657	21665	36792	42913

Conclusão

Bom, comparando os resultados com o AG anterior o resultado foi ruim, ao aumentarmos o número de clusters, os resultados variam cada vez mais, porém, se levarmos em consideração apenas os resultados do artigo, com os resultados do código implementado por mim, é um bom resultado, na prática, o DPSO foi um algoritmo que demandou muito menos tempo do que o AG, porém, ser mais rápido e dar resultados menos precisos não é exatamente algo muito bom.