O Problema do caixeiro viajante através de algoritmo genético

Lorena Mazia Enami (Universidade Estadual de Maringá) lorena.enami@gmail.com
Larissa Longo da Silva (Universidade Estadual de Maringá) larissalongo92@gmail.com
Paula Vasconcelos Stocco (Universidade Estadual de Maringá) paulavstocco@gmail.com
Rommy Komatsu (Universidade Estadual de Maringá) rommykomatsu@gmail.com
Alana Corsi (Universidade Estadual de Maringá) aaacorsi@gmail.com

Resumo: A meta-heurística é a execução de métodos para problemas que não possuem algoritmo estabelecido, empregue para encontrar soluções subótimas por intermédio da avaliação de soluções de qualidade com baixo custo computacional, bom desempenho, rapidez e simplicidade. O algoritmo genético é um desses métodos, e tem como fundamentação as premissas de Charles Darwin em que os indivíduos menos aptos são desprezados e os mais aptos selecionados. Nesse contexto, o problema do caixeiro viajante pretende descobrir o menor caminho de um número finito de vértices com distâncias entre si conhecidas. Assim, o objetivo desse trabalho é minimizar o custo da rota que contém todas as nove cidades simuladas, por intermédio da programação de um algoritmo genético. Utilizou-se da linguagem de programação C, com padrões de 3500 número de evoluções, uma amostra populacional de tamanho 15, e uma taxa de mortalidade de 0,5. As simulações apresentaram grande divergência quanto ao número de interações, devido a aleatoriedade do modelo, direcionando o resultado final ao algarismo baixo devido ao pequeno porte do problema.
Palavras chave: Caixeiro viajante, algoritmo genético, meta-heurística.

The Traveling Salesman Problem Through Genetic Algorithm

Abstract: A meta-heuristic is an execution's methods for problems that do not have an established algorithm, used to find suboptimal solutions by evaluating quality solutions with low computational cost, good performance, speed and simplicity. The genetic algorithm is one such method, and is based on Charles Darwin's assumptions that the least used are scorned and the most selected. In this context, the problem of the traveling salesman discovering the shortest path of a finite number of points with known distances from each other. Thus, the objective of this work is to reduce the cost of the route that contains all nine simulated cities by programming a genetic algorithm. Used in C programming language, with patterns of 3500 evolution numbers, a population sample of size 15, and a mortality rate of 0.5. As simulations showed great divergence in the number of interactions, due to the randomness of the model, the final result was directing to the low digit due to the small size of the problem.
Key-words: traveling salesman, genetic algorithm, mate-heuristic.

1. Introdução

Heurística deriva da palavra grega heuriskein, que significa descobrir (BECCENERI, 2012). A heurística também pode ser definida como um algoritmo que resolve problemas por abordagem intuitiva, gerando uma solução admissível, porém não garantindo que aquela seja a solução ótima (LOPES et al., 2013). Gomes (2006) afirma que os métodos heurísticos encontram soluções subótimas para os problemas, porém não garantem o resultado ótimo, visto que aliam soluções de qualidade com baixo custo computacional. Além disso, as heurísticas apresentam bom desempenho, rapidez e simplicidade.

Uma meta-heurística, por sua vez, é definida como conceitos utilizados para resolver um conjunto de problemas através de métodos heurísticos. (BECCENERI, 2012). Para Gomes (2006), uma meta-heurística pode ser conceituada como a junção de métodos heurísticos para encontrar soluções de qualidade, aplicada em problemas que não têm um algoritmo definido ou demandam um grande esforço computacional.

Otimização é a busca pela solução ótima de dado problema. Busca soluções cada vez melhores através da tentativa de várias soluções. Para isso, são utilizadas técnicas que apresentam um espaço de busca, onde se encontram as soluções potenciais do problema, e uma função objetivo que avalia as soluções encontradas. Ou seja, a otimização é a busca por um ponto de máximo ou mínimo da função objetivo. Porém, pode haver pontos, os chamados máximos ou mínimos locais, que dificultam o descobrimento dos máximos e mínimos globais. (LACERDA & CARVALHO, 1999).

2. Revisão bibliográfica

Essa sessão abordará a fundamentação teórico referente a algoritmo genérico e o problema do caixeiro viajante.

2.1. Algoritmo genético

Um algoritmo genético (AG) é um método de otimização baseado em mecanismos evolucionistas, que seguem a premissa de seleção natural e da sobrevivência de seres mais aptos, temas estes abordados por Charles Darwin. De acordo com essa teoria, quanto mais adaptado um ser é ao meio, maior a probabilidade de este sobreviver e de se reproduzir (LACERDA & CARVALHO, 1999).

Para Pacheco (1999), o princípio da seleção natural beneficia os indivíduos mais aptos que geram mais descendentes. Assim, seu código genético apresenta maior probabilidade de perpetuação, sendo os códigos representados por cromossomos. Dessa maneira, algoritmos computacionais procuram uma solução ótima para um problema através de populações de soluções codificadas por cromossomos artificiais, ou seja, dados que representam uma possível solução para o problema.

Para que, depois de muitas gerações, toda a população não tenha o mesmo material genético, a natureza modifica o material genético de alguns indivíduos através da mutação. Se os seres que sofrem mutação apresentam uma boa capacidade de sobrevivência, sua probabilidade de perpetuação aumente (SOARES, 1997). A analogia entre os sistemas natural e artificial é mostrada no quatro 1.

No Algoritmo Genético Simples (SGA), a população é fixa e as cadeias de caracteres são codificadas binariamente. Depois de analisado o problema, define-se a quantidade de indivíduos da população, a formação cromossômica do indivíduo e as probabilidades de aplicação dos operadores genéticos e, então, avalia-se cada indivíduo de acordo com seu desempenho, a partir do qual o processo de seleção define quais indivíduos se reproduzirão e sobre estes aos operadores genéticos, cruzamento e mutação agem (SOARES, 1997).

Natureza	Algoritmos Genéticos
Cromossoma	Palavra binária, vetor, etc.
Gene	Caractere
Alelo	Valor do caractere
Loco	Posição na cadeia, vetor
Genótipo	Estrutura
Fenótipo	Estrutura submetida ao problema
Indivíduo	Solução
Geração	Ciclo

Quadro 1. Analogia entre sistemas adaptado de Pacheco (1999)

Inicialmente, há a geração de uma população inicial de cromossomos formada aleatoriamente, que são soluções possíveis para o problema. A população é avaliada e cada um dos cromossomos recebe um valor, denominado aptidão, representando a qualidade das soluções. Seguindo ao princípio darwinista, os cromossomos menos aptos são rejeitados e os mais aptos, selecionados, que podem sofrer modificações através dos operadores genéticos de cruzamento e mutação, gerando descendentes. Esse processo se repete até que seja encontrada uma solução razoável (LACERDA & CARVALHO, 1999). A seguir, serão apresentados detalhes de cada estágio desse processo.

2.1.1. Formação da população

Para que se possam efetuar os operadores genéticos, o indivíduo é codificado em uma cadeia de caracteres de comprimento l, como um vetor, em que cada cromossomo representa uma variável (SOARES, 1997).

2.1.2. Seleção

São escolhidos os melhores cromossomos da população inicial para cruzamento e mutação, gerando cromossomos filhos. Os pais são selecionados de acordo com sua aptidão, ou seja, qualidade de solução. Neste trabalho a seleção é feita através de ordenamento (ranking) do cromossomo (LACERDA & CARVALHO, 1999).

2.1.3. Cruzamento

É a troca de material genético entre os cromossomos, o que possibilita a combinação das soluções. No SGA a cadeia de caracteres é cortada na mesma posição aleatória para os dois indivíduos selecionados e troca-se o material genético a partir da posição em que estes foram cortados (SOARES, 1997).

2.1.4. Mutação

Soares (1997) define a mutação como a inserção de novas características e restauração de material genético descartado no processo de seleção. A mutação inverte os valores dos bits dos dois filhos gerados de acordo com uma probabilidade. Esse operador intensifica a diversidade dos cromossomos, porém destrói os dados existentes previamente. Logo, a mutação se dá de acordo com uma taxa que garante a diversidade da população (LACERDA & CARVALHO, 1999).

2.1.5. Critérios de parada

Para analisar a convergência do método, podem-se adotar diversos critérios de parada. Para Lacerda e Carvalho (1999), os critérios de parada são:

a) Quando o AG atinge um determinado número de gerações ou verificações;
b) Quando o valor ótimo da função objetivo for atingido (caso este seja conhecido);
c) Quando não houver melhoramento razoável no melhor cromossomo por um determinado número de gerações

2.1.6. Problemas de convergência

A convergência prematura faz com que os indivíduos percam sua diversidade antes de atingir o objetivo principal. Isso faz com que os GAs atinjam um ápice local, o que diminui seu desempenho e produz resultados inferiores (SOARES, 1997).

Uma medida para suavizar a convergência prematura é limitar os filhos por cromossomo, através de escalonamento de aptidão, ordenamento, manutenção da diversidade, entre outras técnicas.

2.2. O problema do caixeiro viajante

O primeiro registro de um problema do caixeiro viajante (PCV) se deu em 1930 por Karl Menger. O problema consiste em encontrar o menor caminho de um número finito de pontos cujas distâncias entre si são dadas. Pode-se encontrar a solução por um número finito de tentativas, porém não há uma regra que reduz o número de tentativas para um valor menor do que o número de permutações dos pontos dados. O PCV é classificado como um problema NP-Hard, isto é, não são possíveis de se resolver através de um algoritmo polinomial (GAREY e JOHNSON, 1979).

Hoffman et al (2001), afirmam que o PCV tem chamado a atenção de matemáticos pelo fato de ser tão fácil de explicar e tão difícil de resolver. O problema consiste em encontrar o caminho que demanda menor custo em que um caixeiro viajante que deseja visitar apenas uma vez uma lista de m cidades (sendo o custo de transitar da cidade i para a j cij) e, então, retornar à cidade inicial.

Segundo Selong e Kripka (2009), um dos tipos mais importantes de PCV é o hamiltoniano, proposto por Willian Hamilton, que propôs um jogo chamado de “*Around the World*”. O jogo consistia em um dodecaedro cujos vértices representavam cidades e seu objetivo era encontram um caminho pelos vértices que começasse e terminasse no mesmo vértice sem repetir uma cidade. A solução do jogo denominou-se ciclo hamiltoniano. Assim, o problema do caixeiro viajante é de otimização que busca fluxos hamiltonianos em um grafo qualquer, objetivando-se encontrar o caminho hamiltoniano de menor custo. Dessa forma, como o ponto de início e de final é o mesmo, podem ser feitos (n-1)! caminhos.

A estrutura de formulação matemática do PCV se dá por meio de um grafo, no qual cada cidade é representada por um ponto ou nó e arcos conectam dois nós distintos. O grafo é dito completo quando o caixeiro é capaz de chegar a todas as cidades a partir de todas as outras diretamente.

3. Resultados

3.1 Descrição do problema

Supondo uma empresa de logística que busca otimizar seu processo de entrega de produtos em nove cidades. Para isso, foi analisada a distância entre as cidades percorridas. A área de vendas inclui 9 cidades e a cada viagem está associado um custo proporcional à distância que se percorre. O objetivo é minimizar o custo da rota, ou seja, selecionar a rota mais curta que leva ao menor custo. No quadro 2 temos as 9 cidades e a distância entre elas.

Cidade	A	B	C	D	E	F	G	H	I
A	0	42	61	30	17	82	31	11	27
B	42	0	14	87	28	70	19	33	13
C	61	14	0	20	81	21	8	29	15
D	30	87	20	0	34	33	91	10	20
E	17	28	81	34	0	41	34	82	22
F	82	70	21	33	41	0	19	32	27
G	31	19	8	91	34	19	0	59	30
H	11	33	29	10	82	32	59	0	32
I	27	13	15	20	22	27	30	32	0

Quadro 2. Matriz de distância entre cidades

O grafo desse problema é mostrado na figura 1.

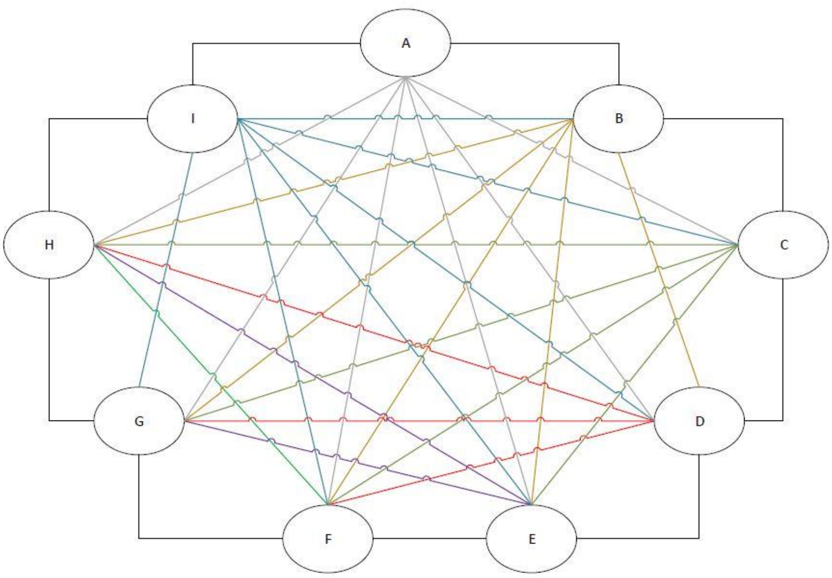

Figura 1. Grafo do PCV

3.2 Formulação matemática

Utilizando-se a formulação de Dantzig-Johnson e o grafo G = (N, A), sendo N os nós e A as arestas, temos a representação do grafo:

$$Minimizar\ z = \sum_{j=1}^{n} \sum_{i=1}^{n} c_{ij} x_{ij}$$

Sujeito a:

$$\sum_{i=1}^{n} x_{ij} = 1, \ \forall j \in N$$

$$\sum_{j=1}^{n} x_{ij} = 1, \ \forall i \in N$$

$$\sum_{i,j \in S}^{n} x_{ij} < |S| - 1, \ \forall S \subset N$$

$$x_{ij} \in \{0,1\} \ \forall i, j \in N$$

Em que

N: número de nós da rede;

$\sum_{i=1}^{n} x_{ij} = 1, \ \forall j \in N$: todo fluxo que chega ao nó *j* é igual a 1;

$\sum_{j=1}^{n} x_{ij} = 1, \ \forall i \in N$: todo fluxo que sai do nó *i* é igual a 1;

S: subgrafo de G;

|S|: número de vértices do subgrafo S;

$\sum_{i,j \in S}^{n} x_{ij} < |S| - 1, \ \forall S \subset N$: devem ser evitados subciclos;

$$x_{ij} \begin{cases} 1, & se\ o\ fluxo\ passar\ pelo\ arco\ (i,j) \\ 0, & caso\ contrário \end{cases}$$

Nesse caso, x_{ii} é nulo, pois nesse caso o caminho sai e chega no mesmo nó, o que não faz sentido para a resolução.

4. Simulações

A seguir, são apresentadas três simulações do programa que apresentam o número de iterações necessárias para atingir a convergência do gráfico, o custo e o caminho mais apto.

Iterações	Aptidão	Percurso
1	234	C -> G -> H -> F -> E -> A -> D -> I -> B -> C
2	197	D -> F -> E -> A -> H -> B -> G -> C -> I -> D
9	171	E -> A -> H -> D -> F -> C -> G -> I -> B -> E
14	165	E -> A -> H -> D -> C -> B -> I -> F -> G -> E
82	154	E -> A -> H -> D -> F -> C -> G -> B -> I -> E
397	153	E -> A -> H -> D -> F -> C -> G -> B -> I -> E
1270	147	A -> H -> D -> F -> G -> C -> B -> I -> E -> A

Quadro 3. Simulação 1

Figura 2. Evolução dos cromossomos da simulação 1

Iterações	Aptidão	Percurso
1	243	D -> E -> B -> H -> I -> C -> F -> G -> A -> D
2	238	I -> E -> B -> H -> D -> A -> C -> G -> F -> I
8	199	G -> F -> D -> H -> I -> C -> B -> E -> A -> G
21	194	A -> H -> F -> I -> D -> C -> B -> G -> E -> A
27	192	B -> I -> A -> H -> F -> G -> C -> D -> E -> B
38	183	D -> H -> A -> I -> B -> C -> F -> G -> E -> D
53	172	I -> E -> A -> D -> H -> F -> C -> G -> B -> I
130	165	A -> E -> F -> G -> B -> I -> C -> D -> H -> A
146	153	A -> E -> B -> I -> F -> G -> C -> D -> H -> A
662	152	A -> E -> I -> B -> G -> F -> C -> D -> H -> A
1059	147	A -> E -> I -> B -> C -> G -> F -> D -> H -> A

Quadro 4. Simulação 2

Figura 3. Evolução dos cromossomos da simulação 2

Iterações	Aptidão	Percurso
1	245	E -> G -> F -> B -> H -> D -> C -> I -> A -> E
2	231	B -> G -> A -> I -> C -> F -> E -> D -> H -> B
5	230	B -> H -> F -> G -> A -> D -> C -> I -> E -> B
6	223	G -> E -> B -> I -> F -> C -> H -> D -> A -> G
10	205	I -> E -> B -> H -> A -> G -> F -> C -> D -> I
16	195	A -> G -> F -> I -> D -> H -> C -> B -> E -> A
31	194	F -> G -> B -> E -> A -> H -> I -> C -> D -> F
94	181	I -> E -> F -> G -> A -> H -> D -> C -> B -> I
106	160	F -> G -> B -> E -> A -> H -> D -> I -> C -> F
147	153	G -> C -> B -> I -> D -> H -> A -> E -> F -> G
663	147	C -> B -> I -> E -> A -> H -> D -> F -> G -> C

Quadro 5. Simulação 3

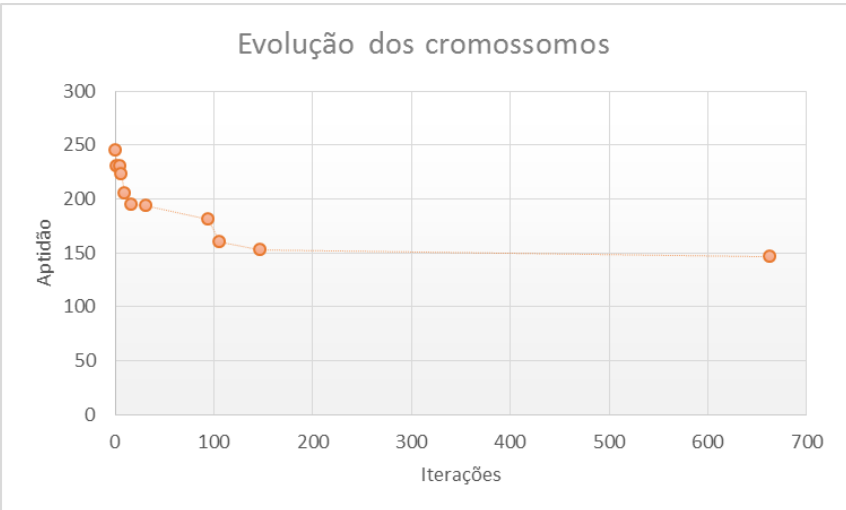

Figura 4. Evolução dos cromossomos da simulação 3

5. Considerações finais

Conhecendo os conceitos de algoritmo genético e os padrões utilizados para código apresentado nesse trabalho (número de evolução = 3500; tamanho da população = 15; e taxa de mortalidade = 0,5), é possível fazer a análise do gráfico e das tabelas identificados à cima.

Primeiramente, sabe-se que o algoritmo genético procura a solução ótima dentro de um determinado problema proposto, ou seja, a solução menos custosa. No presente trabalho, identificou-se o menor caminho como sendo o que oferece uma aptidão total de cento e quarenta e sete quilômetros (147 km), encontrada em todas as simulações realizadas. Porém, a cada simulação, o número de iterações para que fosse obtido o custo mínimo variou de seiscentos e sessenta e três (663) a mil duzentos e setenta (1270). Houve grande discrepância entre o número de interações, pelo fato de esse fator depender da aleatoriedade na geração da população inicial, e possivelmente só foi alcançado em um algarismo baixo, 663, devido ao pequeno porte do problema em questão.

Em ambos os casos, chegou-se à estagnação do gráfico demasiadamente rápido, comparado ao número de evolução adotado.

Referências

GAREY, M. R.; JOHNSON, D. S. Computers and Intractability: a Guide to the Theory Of NP_Completness. 1. ed. New York: W. H. Freeman, 1979.

HOFFMAN, K.L.; PADBERG, M.; RINALDI, G. Traveling Salesman Problem. 2001, 9 f. Kluwer Academic Publishers, 2001.

LACERDA, E.G.M, CARVALHO, A.C.P.L. Introdução aos algoritmos genéticos. In: Galvão, C.O., Valença, M.J.S. (orgs.) Sistemas inteligentes: aplicações a recursos hídricos e ciências ambientais. Porto Alegre: Ed. Universidade/UFRGS : Associação Brasileira de Recursos Hídricos, 1999. Disponível em: <http://www.leca.ufrn.br/~estefane/metaheuristicas/ag.pdf>.

MENGER, K. Ergebnisse eines Kolloquiums. 3, 11-12, 1930.

PACHECO, M.A.C. Algoritmos genéticos: princípios e aplicações. Disponível em: http://www.ica.ele.puc-rio.br/downloads/38/ce-apostila-comp-evol.pdf.

SELONG, L.M.; KRIPKA, R.M. Otimização de roteiros: estudo de caso de uma distribuidora de ferro de Passo Fundo/RS para a região. Revista CIATEC, vol.1 (1), p.p. 14-31, 2009.

SOARES, G.L. Algoritmos genéticos: estudos, novas técnicas e aplicações. 1997, 145 f. Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte, 1997.