Aplicação de algoritmos de busca na resolução do Problema do Caixeiro Viajante

João Paulo Vargas da Fonseca

Departamento Acadêmico de Engenharia Eletrônica

Universidade Tecnológica Federal do Paraná (UTFPR) – Curitiba, PR -Brazil

jfonseca@alunos.utfpr.edu.br

**Abstract.** This meta-paper describes the style to be used in articles and short papers for SBC conferences. For papers in English, you should add just an abstract while for the papers in Portuguese, we also ask for an abstract in Portuguese (“resumo”). In both cases, abstracts should not have more than 10 lines and must be in the first page of the paper.

**Resumo.** Este meta-artigo descreve o estilo a ser usado na confecção de artigos e resumos de artigos para publicação nos anais das conferências organizadas pela SBC. É solicitada a escrita de resumo e abstract apenas para os artigos escritos em português. Artigos em inglês deverão apresentar apenas abstract. Nos dois casos, o autor deve tomar cuidado para que o resumo (e o abstract) não ultrapassem 10 linhas cada, sendo que ambos devem estar na primeira página do artigo.

1. Problema do Caixeiro Viajante

O problema do caixeiro viajante é um problema simples, consistente de encontrar o menor percurso dentre um conjunto de cidades, partindo de uma cidade origem onde cada cidade deve ser visitada apenas uma vez, e após voltar à origem. Porém, ele é complexo computacionalmente, da ordem de O(n!), sendo n o número de cidades. Por isso utilizam-se algoritmos de busca, que têm como objetivo otimizar o custo computacional, como os algoritmos têmpera simulada e o algoritmo genético, para encontrar soluções boas, mas sem otimalidade, que é a garantia de se obter a melhor solução, no caso, obter o caminho com a menor distância possível.

1. Fundamentação Teórica

O algoritmo da têmpera simulada implementa uma busca em estados vizinhos ao do atual estado, sendo uma melhoria ao algoritmo de subida de encosta pois, enquanto este acessa um estado inicial aleatório e implementa uma ida somente a estados vizinhos que tenham alguma característica melhor, que é medida através de uma função, aquele permite a ida a estados piores com uma certa probabilidade que decai com o tempo, ou número de iterações ou temperatura. Sendo esta somente uma forma de inserir aqueles no cálculo de probabilidade, que, quando chega a zero, transforma o algoritmo em uma subida de encosta, parando quando não houver melhores estados vizinhos que o atual.

Já o algoritmo genético tenta implementar algo totalmente diferente, pois se assemelha ao processo de evolução natural para se chegar a um estado melhor que o atual. Ele se utiliza de uma população, ou seja, um número arbitrário de indivíduos (estados inicialmente gerados aleatoriamente), com poucos escolhidos de acordo com seu valor de fitness, uma função avaliativa do próprio estado, para reproduzir, combinando genes dos pais, e gerar a próxima geração, sendo que há uma possibilidade de cada novo indivíduo gerado sofra alguma mutação. E a condição de parada normalmente é uma convergência dos genes da população.

1. Metodologia

O ambiente considerado é um plano, com cidades puntiformes. Desta forma, uma cidade pode ser representada por um número único e uma posição em um plano cartesiano. E um caminho pode ser facilmente representado por um conjunto de números únicos, referentes às cidades na ordem em que devem ser percorridas pelo caixeiro antes de retornar à inicial.

Como é possível ter um número n de cidades, cada ambiente será gerado aleatoriamente com as n cidades e armazenado, até para ter uma referência de comparação entre os algoritmos de têmpera simulada e o algoritmo genético.

Computacionalmente, cada ambiente será uma matriz n x 2, sendo o primeiro índice o número único da cidade e o segundo referente à posição x (1) e y (2), e cada conjunto de cidades será armazenado em um arquivo texto (“.txt”). O caminho armazenado em um vetor de tamanho n, sendo o índice responsável pela posição da cidade no caminho, e seu armazenamento será o número único da cidade. E como a comparação de um caminho a outro é feita pela distância total, que é a soma da distância entre as cidades adjacentes no caminho, será armazenado em uma matriz n x n a distância entre duas cidades i e j, prevendo com que o mesmo cálculo seja feito milhares de vezes pelo custo de mais armazenamento.

A implementação será feita utilizando do Matlab R2021a.

* 1. Têmpera Simulada

No têmpera simulada, o primeiro estado é gerado aleatoriamente, e os vizinhos são ditos aqueles nos quais cidades adjacentes, no estado atual, são trocadas. Ou seja, trocar a posição do número único no vetor de caminho entre as posições e , com exceção das trocas com a cidade origem, que seguem a mesma lógica, mas em um vetor são representadas de outra forma.

A cada iteração, é visitado um estado vizinho e caso o valor do mesmo, representado pela soma das distâncias no caminho, for menor, tal se torna o estado atual. Caso contrário, é gerado um número aleatório entre 0 e 1 que, se for menor que a probabilidade indicada por (1)

No têmpera simulada, vizinhos são considerados como estados cujas cidades adjacentes ou consecutivas no caminho, pois esta é a forma de ter o menor número possível de vizinhos.

O valor de um estado é considerado como a soma das distâncias dentre cidades em seu caminho. E a probabilidade do estado atual ir a um estado de valor maior (pior) é dado por (1), desta forma é mais provável ir a estados menos piores do que muito piores.

E como a temperatura está se relacionando com a diferença do valor entre estados, foi definido que ela teria valor inicial como a média da matriz de distâncias entre as cidades. E ao longo das iterações, a temperatura é dada por (2), chegando à temperatura final somente quando chegar ao número máximo de iterações.

A probabilidade do estado atual ir a um estado cujo valor, dado pela soma das distâncias no caminho, é maior

A temperatura inicial é dada pela média das distâncias entre as cidades

The first page must display the paper title, the name and address of the authors, the abstract in English and “resumo” in Portuguese (“resumos” are required only for papers written in Portuguese). The title must be centered over the whole page, in 16 point boldface font and with 12 points of space before itself. Author names must be centered in 12 point font, bold, all of them disposed in the same line, separated by commas and with 12 points of space after the title. Addresses must be centered in 12 point font, also with 12 points of space after the authors’ names. E-mail addresses should be written using font Courier New, 10 point nominal size, with 6 points of space before and 6 points of space after.

The abstract and “resumo” (if is the case) must be in 12 point Times font, indented 0.8cm on both sides. The word **Abstract** and **Resumo**, should be written in boldface and must precede the text.

1. CD-ROMs and Printed Proceedings

In some conferences, the papers are published on CD-ROM while only the abstract is published in the printed Proceedings. In this case, authors are invited to prepare two final versions of the paper. One, complete, to be published on the CD and the other, containing only the first page, with abstract and “resumo” (for papers in Portuguese).

1. Sections and Paragraphs

Section titles must be in boldface, 13pt, flush left. There should be an extra 12 pt of space before each title. Section numbering is optional. The first paragraph of each section should not be indented, while the first lines of subsequent paragraphs should be indented by 1.27cm.

* 1. Subsections

The subsection titles must be in boldface, 12pt, flush left.

1. Figures and Captions

Figure and table captions should be centered if less than one line (Figure 1), otherwise justified and indented by 0.8cm on both margins, as shown in Figure 2. The caption font must be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.



Figure 1. A typical figure



Figure 2. This figure is an example of a figure caption taking more than one line and justified considering margins mentioned in Section 5.

In tables, try to avoid the use of colored or shaded backgrounds, and avoid thick, doubled, or unnecessary framing lines. When reporting empirical data, do not use more decimal digits than warranted by their precision and reproducibility. Table caption must be placed before the table (see Table 1) and the font used must also be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

Table 1. Variables to be considered on the evaluation of interaction techniques



1. Images

All images and illustrations should be in black-and-white, or gray tones, excepting for the papers that will be electronically available (on CD-ROMs, internet, etc.). The image resolution on paper should be about 600 dpi for black-and-white images, and 150-300 dpi for grayscale images. Do not include images with excessive resolution, as they may take hours to print, without any visible difference in the result.

1. References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references in brackets, e.g. [Knuth 1984], [Boulic and Renault 1991]; or dates in parentheses, e.g. Knuth (1984), Smith and Jones (1999).

The references must be listed using 12 point font size, with 6 points of space before each reference. The first line of each reference should not be indented, while the subsequent should be indented by 0.5cm.

References

Boulic, R. and Renault, O. (1991) “3D Hierarchies for Animation”, In: New Trends in Animation and Visualization, Edited by Nadia Magnenat-Thalmann and Daniel Thalmann, John Wiley & Sons ltd., England.

Dyer, S., Martin, J. and Zulauf, J. (1995) “Motion Capture White Paper”, http://reality.sgi.com/employees/jam\_sb/mocap/MoCapWP\_v2.0.html, December.

Holton, M. and Alexander, S. (1995) “Soft Cellular Modeling: A Technique for the Simulation of Non-rigid Materials”, Computer Graphics: Developments in Virtual Environments, R. A. Earnshaw and J. A. Vince, England, Academic Press Ltd., p. 449-460.

Knuth, D. E. (1984), The TeXbook, Addison Wesley, 15th edition.

Smith, A. and Jones, B. (1999). On the complexity of computing. In *Advances in Computer Science*, pages 555–566. Publishing Press.