

Algoritmo Heurístico Otimização de Colônia de Formigas e Programação Paralela para a resolução do Problema do Máximo Conjunto Independente

Luciana P. Nedel¹, Rafael H. Bordini², Flávio Rech Wagner¹, Jomi F. Hübner³

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

²Department of Computer Science – University of Durham
Durham, U.K.

³Departamento de Sistemas e Computação
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brazil

{nedel, flavio}@inf.ufrgs.br, R.Bordini@durham.ac.uk, jomi@inf.furb.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 trabalho tem como objetivo na elaboração de algoritmo sequencial e paralelizado para encontrar soluções viáveis do problema do Máximo Conjunto Independente utilizando algoritmos da família Colônia de Formigas introduzida por Marco Dorigo. Além da elaboração, é realizada a análise e comparação entre os algoritmos confeccionados, proporcionando dados sobre a efetividade na utilização de múltiplas threads para o algoritmo apresentado e também ao problema solucionado.*

1. Introdução

O Problema do Máximo Conjunto Independente, traduzido de Maximum Set Independent Problem (MISP), pode ser considerado um problema de importância para a computação devido a sua aplicabilidade a várias áreas, tais como, Reconhecimento de Padrões, Escalonamento, Biologia Molecular e *Map Labeling*.

Para a elaboração do algoritmo solucionador do MISP, foi utilizado a técnica da Otimização de Colônia de Formigas, do inglês, Ant Colony Optimization (ACO), introduzido por Marco Dorigo */**Referencia**/*. O ACO desde sua criação vem resolvendo diversos problemas difíceis, como – citar problemas mais recentes –.

//Falar de algoritmos paralelos

Este artigo é organizado da seguinte forma. Na Seção 2, será descrito de forma detalhada o Problema do Máximo Conjunto Independente. Na Seção 3, será apresentado trabalho relacionados, seja pela resolução do Problema do Máximo Conjunto Independente, a utilização da Otimização de Colônia de Formigas ou por trabalhos utilizando algoritmo paralelos.

2. Problema

Dado um Grafo $G = (V, E)$, em V representa o conjunto de vértices e E representa o conjunto de arestas do grafo. O Problema do Máximo Conjunto Independente tem como objetivo encontrar um subconjunto $V^* \subseteq V$, em que $\forall i, j \in V^*$, a aresta $(i, j) \notin E$, além de que V^* deve ser máximo.

A formulação da programação inteira para o trabalho, pode ser definido da seguinte maneira:

$$\begin{aligned} \max \sum_{i=1}^{|V|} c_i x_i \\ x_i + x_j \leq 1, \quad \forall (i, j) \in E \\ x_i \in \{0, 1\}, \quad i = 1, 2, \dots, |V| \end{aligned}$$

O problema de decisão para verificar se há um conjunto de vértices independentes de tamanho n é classificado como NP-Completo. Enquanto o Problema Máximo Conjunto Independente é classificado como NP-Difícil.

O MISP é um problema muito similar ao problema do clique máximo, que resumidamente, busca encontrar o maior subconjunto de vértices adjacentes, em outras palavras, a cada par de vértices dentro do subconjunto, é necessário que haja uma aresta os interligando. É possível resolver o MISP através do clique máximo, dado o grafo de entrada no MISP, basta acharmos o seu complemento e utilizá-lo como entrada para o clique máximo. Também é válido a operação inversa, utilizarmos o MISP para resolvermos o clique máximo. Com isso, dado um grafo G qualquer e seu complemento o grafo X , temos que:

$$\begin{aligned} \text{MISP}(G) &= \text{CliqueMaximo}(X) \text{ ou então,} \\ \text{CliqueMaximo}(G) &= \text{MISP}(X) \end{aligned}$$

//falar do clique

3. Trabalhos Relacionados

A meta-heurística Ant Colony Optimization foi apresentado por Marco Dorigo */**Referencia**/* a partir de observações sobre o comportamento de formigas reais a fim de descobrir como animais de pouca visão conseguiam se locomover e atingir seus destinos. Como o resultado das observações foi modelada e desenvolvida a ACO. Ao longo do tempo foi desenvolvida extensões para a ACO, como por exemplo, Recursive Ant Colony Optimization por */**REFERENCIA**/* e Elitist Ant System por */**REFERENCIA**/*.

Foram encontrados diversos trabalhos com o Máximo Conjunto Independente, sendo o mais antigo do ano 1977, desenvolvido por RE Tarjan AE Trojanowski que apresentou um algoritmo de solução. Também foi encontrado trabalhos que apresentavam soluções utilizando heurísticas, bem como o trabalho de MAURICIO G.C. RESENDE AND CELSO C. RIBEIRO */**REFERÊNCIA**/* que utilizaram a meta-heurística GRASP e Algoritmos Evolucionários por Back e Khuri */**REFERENCIA**/*

Direcionando as pesquisas para encontrar trabalhos que utilizaram a meta-heurística Ant Colony Optimization para a resolução do Máximo Conjunto Independente, foi encontrado dois trabalhos que propôs esse desafio. */**Referenciar os 2**/*. Para o

presenta trabalho, utilizaremos propostas enunciadas neste trabalho, bem como a função probabilidade /ver nome e parâmetros.

Devido a sua semelhança, também foi pesquisado trabalhos envolvendo ACO e Clique Máximo, como destaque, houve os trabalhos /*referencia*/

4. Proposta

Para a resolução do problema, poderíamos utilizar algoritmos determinísticos, porém ao utilizar algoritmos heurísticos provavelmente encontraremos boas soluções com menor tempo em relação aos algoritmos determinísticos e com o uso de heurísticas adequadas a queda na qualidade das soluções não será tão sentida.

O único conhecimento que temos para resolver o problema é o grafo, e assim temos acesso a sua constituição, como por exemplo, o número de vértices e arestas, densidade e quantidade de cada vértices adjacentes de cada vértice.

Para resolvermos o problema, precisamos utilizar três conjuntos ao longo do algoritmo, os conjuntos são:

S(t): Conjunto de Vértices presente na resposta em um determinado tempo t.

I(t): Conjunto de Vértices que **não** podem estar mais na solução em um determinado tempo t.

D(t): Conjunto de Vértices que **ainda** podem estar solução em um determinado tempo t.

Com nos três conjuntos, sabemos que $S(t) + I(t) + D(t) = V$ para qualquer instante de tempo, e que $S(t) \subseteq V^*$. //Figuras com os três conjuntos dentro de V e da figura de S(t) dentro de V^*

A partir do grafo abaixo, explicaremos como funciona a heurística utilizada, que foi abstraída do trabalho /*REFERÊNCIA*/.

//Grafo

//explicar que pega o vertice com mais vert não adjacentes

5. Avaliação

Para a realização dos testes foi utilizado um computador com processador XXX de xx núcleos, com memória RAM xx, Memória cache e Sistema Operacional Ubuntu 16.04 64 bits.

No código sequencial ocorreu tantas miss caches, uso de memória. Enquanto no código em paralelo

//gráfico

Os resultados que foram obtidos, mostra que houve..

6. Conclusões e Trabalhos Futuros

Os resultados desse trabalho

A utilização de programação paralela se mostra bastante útil quando utilizada de forma correta e sua utilização nos mostra o ganho de performance na execução dos problemas e ainda é uma forma eficiente de utilizar toda potência de um computador multi-core. /verificar

Devido a inexperiência com a programação paralela, um trabalho futuro desejado seria a revisão do código e a melhoria do mesmo, devido ao aumento de conhecimento na área.

Outra hipótese envolvendo programação paralela é a utilização de placas gráficas para a execução de códigos paralelos, área crescente no cenário e que possui resultados animadores. A junção da programação paralela entre CPU e GPU já foi proposta e deve ser bastante explorada nos tempos atuais.

7. Referências

8. General Information

All full papers and posters (short papers) submitted to some SBC conference, including any supporting documents, should be written in English or in Portuguese. The format paper should be A4 with single column, 3.5 cm for upper margin, 2.5 cm for bottom margin and 3.0 cm for lateral margins, without headers or footers. The main font must be Times, 12 point nominal size, with 6 points of space before each paragraph. Page numbers must be suppressed.

Full papers must respect the page limits defined by the conference. Conferences that publish just abstracts ask for **one**-page texts.

9. First Page

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.

10. 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).

11. 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 1.27 cm.

11.1. Subsections

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

12. 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

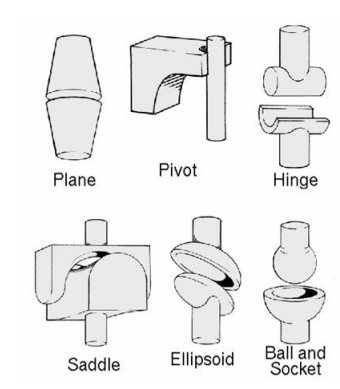


Figure 2. This figure is an example of a figure caption taking more than one line and justified considering margins mentioned in Section 12.

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

	Chessboard top view	Chessboard perspective view
Selection with side movements	6.02 \pm 5.22	7.01 \pm 6.84
Selection with in- depth movements	6.29 \pm 4.99	12.22 \pm 11.33
Manipulation with side movements	4.66 \pm 4.94	3.47 \pm 2.20
Manipulation with in- depth movements	5.71 \pm 4.55	5.37 \pm 3.28

13. 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.

14. 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], and [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 0.5 cm.

References

- Boulic, R. and Renault, O. (1991). 3d hierarchies for animation. In Magnenat-Thalmann, N. and Thalmann, D., editors, *New Trends in Animation and Visualization*. John Wiley & Sons Ltd.
- Knuth, D. E. (1984). *The T_EX Book*. Addison-Wesley, 15th edition.
- Smith, A. and Jones, B. (1999). On the complexity of computing. In Smith-Jones, A. B., editor, *Advances in Computer Science*, pages 555–566. Publishing Press.