Alternate *ACM* SIG Proceedings Paper in LaTeX Format*

[Extended Abstract] †

Ben Trovato‡
Institute for Clarity in Documentation
1932 Wallamaloo Lane
Wallamaloo, New Zealand
trovato@corporation.com

G.K.M. Tobin§
Institute for Clarity in Documentation
P.O. Box 1212
Dublin, Ohio 43017-6221
webmaster@marysville-ohio.com

Lars Thørväld¶
The Thørväld Group
1 Thørväld Circle
Hekla, Iceland
larst@affiliation.org

Lawrence P. Leipuner
Brookhaven Laboratories
Brookhaven National Lab
P.O. Box 5000
lleipuner@researchlabs.org

Sean Fogarty
NASA Ames Research Center
Moffett Field
California 94035
fogartys@amesres.org

Charles Palmer
Palmer Research Laboratories
8600 Datapoint Drive
San Antonio, Texas 78229
cpalmer@prl.com

ABSTRACT

This paper provides a sample of a LaTeX document which conforms, somewhat loosely, to the formatting guidelines for ACM SIG Proceedings. It is an *alternate* style which produces a *tighter-looking* paper and was designed in response to concerns expressed, by authors, over page-budgets. It complements the document *Author's (Alternate) Guide to Preparing ACM SIG Proceedings Using LaTeX2$_\epsilon$ and BibTeX*. This source file has been written with the intention of being compiled under LaTeX2$_\epsilon$ and BibTeX.

The developers have tried to include every imaginable sort of "bells and whistles", such as a subtitle, footnotes on title, subtitle and authors, as well as in the text, and every optional component (e.g. Acknowledgments, Additional Authors, Appendices), not to mention examples of equations, theorems, tables and figures.

To make best use of this sample document, run it through LaTeX and BibTeX, and compare this source code with the printed output produced by the dvi file. A compiled PDF version is available on the web page to help you with the 'look and feel'.

CCS Concepts

•**Computer systems organization** → **Embedded systems;** *Redundancy;* Robotics; •**Networks** → Network reliability;

Keywords

ACM proceedings; LaTeX; text tagging

1. INTRODUÇÃO

Um dos assuntos fundamentais da área de Teoria dos grafos são as árvores geradoras de rótulos mínimos. Esta pode ser entendida como uma árvore geradora obtida através de um grafo conectado não direcionado []. Os Problemas das árvores geradoras de rótulos mínimos (PAGRM) podem ser formalmente definidos como: "Dado um grafo rotulado não dirigido $G = (V, E, L)$ no qual V é o conjunto n de vértices, E é o conjunto de m arestas e L é o conjunto de l rótulos, encontrar uma árvore geradora T de G tal que $|L_T|$ é minimizado. Define-se $|L_T|$ como o conjunto de diferentes rótulos das arestas em uma árvore geradora T."

Tal é utilizado para representar situações presentes no mundo real como redes de comunicação, elétrica, transporte e dentre outros. Sua compreensão e estudo é fundamental para a elaboração de soluções que visam reduzir custos e ampliar eficiência. Em outras palavras, o objetivo do estudo e aplicações de algoritmos para resolver o PAGRM é gerar - a partir de um grafo não dirigido - uma árvore que contenha o menor número possível de rótulos e ao mesmo tempo cobrindo todos os seus vértices.

Este trabalho tem por finalidade analisar o conjunto de instâncias – o mesmo conjunto utilizado em [] – através da implementação (realizada na linguagem de programação JAVA) e execução do algoritmo JPSO (*Jumping Particle Swarm Optimization*) [], e dessa forma estabelecer uma comparação entre os resultados obtidos desta aplicação e aqueles provindos das heurísticas apresentadas em [].

*(Produces the permission block, and copyright information). For use with SIG-ALTERNATE.CLS. Supported by ACM.

†A full version of this paper is available as *Author's Guide to Preparing ACM SIG Proceedings Using LaTeX2$_\epsilon$ and BibTeX* at `www.acm.org/eaddress.htm`

‡Dr. Trovato insisted his name be first.

§The secretary disavows any knowledge of this author's actions.

¶This author is the one who did all the really hard work.

WOODSTOCK '97 El Paso, Texas USA

DOI: 10.475/123_4

Tal comparação levou em consideração fatores como a complexidade dos grafos (número de vértices e rótulos) , valor médio da função objetivo e o tempo computacional. Dessa forma, os resultados são comparados da mesma forma como reportado em [].

O restante deste trabalho está organizado em seções da seguinte forma: Na seção 2 é realizada uma revisão literária, na 3 é apresentada a proposta de algoritmo exato para solução do PAGRM (no caso o JPSO), em 3.2 os resultados da implementação são mostrados, em 4 é explicitada uma comparação entre os resultados da literatura e os obtidos neste trabalho e por último são apresentadas as conclusões em 5.

2. REVISÃO LITERÁRIA

3. PROPOSTA DE ALGORITMO – O JPSO

3.1 Heurística - A ideia do JPSO

O PSO (Do inglês *Particle Swarm Optmization*) é um algoritmo que simula o comportamento de revoada de indivíduos na natureza. Cada solução do problema é tido como um indivíduo que pode voar dentro do espaço de soluções. A melhoria das posições, ou seja, as melhores respostas são obtidas através da continua movimentação das partículas que constituem tal revoada. É um algoritmo proposto por [] que utiliza o conceito de inteligência coletiva e é capaz de auxiliar na busca por respostas em problemas de PAGRM. Entretanto, esse algoritmo é formulado para resolver problemas contínuos, o que contradiz o uso para tal. Isto pois, um PAGRM constitui um cenário de resolução discreta.

Portanto, para a utilização da ideia provinda do PSO é necessário que o algoritmo foque problemas discretos. Assim, uma adaptação é utilizada: O JPSO (*Jumping Particle Swarm Optimization*). O JPSO é um DPSO (Do inglês,*Discrete Particle Swarm Optimization*) [] , ou seja, possui a mesma essência da heurística aplicada pelo PSO com a diferença de ser voltado para problemas discretos. A ideia é que a cada iteração as possíveis resposta estejam mais próximas do que seria considerado como ótimo.

3.2 Explicações da implementação do JPSO

A implementação para este trabalho está disposta na linguagem de programação JAVA e está de acordo com o pseudocódigo descrito em []. Assim, a entrada para o JPSO é um grafo não dirigido e conexo $G = (V, E, L)$, com n vértices, m arestas, l rótulos e $Q \subseteq V$ nós. Além disto, tem como saída uma árvore geradora a qual objetiva ser mínima, ou seja, conter o menor número de rótulos possível e ao mesmo tempo cobrir todos os vértices do grafo.

Inicialmente foram carregadas as instâncias presentes nos arquivos disponibilizados em ambiente virtual referente aos grafos sobre os quais a análise foi aplicada e que são as mesmas utilizadas por []. Estas foram carregadas em estruturas as quais retornam um grafo não dirigido e conexo provindo da base de dados.

Em seguida, a partir de cada grafo abstraído da base de dados foram geradas 100 partículas. Estas partículas são a representação das possíveis soluções e são inicialmente formadas de forma randômica: Para cada partícula é induzido um grafo aleatoriamente a partir dos rótulos do grafo original.

A partir desse momento iniciam-se as iterações que buscam a melhor resposta, elas são caracterizadas pelos seguintes passos: 1. atualiza-se a melhor resposta (componente conexa com menor número de rótulos) para cada partícula , 2. atualiza-se a melhor resposta global já obtida, 3. Para cada partícula há a tentativa de reduzir o número de rótulos mantendo as propriedades necessárias, 4. tentativa de combinar duas soluções gerando um grafo com menor número de rótulos.

As tentativas de modificações de cada partículas são randômicas. Além disso é importante ressaltar que a cada iteração as respostas possíveis nunca apresentarão maior número de rótulos do que na iteração anterior, pois toda vez que ocorrer aumento no número de rótulos há o descarte da solução mantendo a anterior. Outro ponto importante é o critério de parada para o algoritmo:É finalizado a partir do momento em que se completam 100 iterações sem modificação no melhor resultado global.

Assim, o JPSO deve retornar a melhor resposta obtida durante sua execução. Apesar de garantir o retorno da melhor solução, não garante que esta seja a solução ótima. Em outras palavras, não retorna necessariamente a arvore com menor rótulos possível de ser formada. Isto devido ao fato de basear-se na aleatoriedade inicial das soluções, o que também implica que execuções distintas sobre a mesma base de dados retorne uma soluções diferentes.

Resultado da execução - O Algoritmo e sua Ação sobre as Instâncias

A aplicação do JPSO para PAGRM (que é um problema NP completo []) sobre a base disponibilizada gerou dados de saída que foram organizados em um arquivo texto em formato *txt*. Eles podem ser verificados de forma organizada nas tabelas a seguir em 1, 2, 3 e 4 e posteriormente são comparados aos resultados dos algoritmos exibidos em [], estes sobre a mesma base de dados.

4. COMPARAÇÃO DE RESULTADOS

5. CONCLUSÃO

.

6. ADDITIONAL AUTHORS

Additional authors: John Smith (The Thørväld Group, email: `jsmith@affiliation.org`) and Julius P. Kumquat (The Kumquat Consortium, email: `jpkumquat@consortium.net`).

7. REFERENCES

[1] M. Bowman, S. K. Debray, and L. L. Peterson. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.*, 15(5):795–825, November 1993.

[2] J. Braams. Babel, a multilingual style-option system for use with latex's standard document styles. *TUGboat*, 12(2):291–301, June 1991.

[3] M. Clark. Post congress tristesse. In *TeX90 Conference Proceedings*, pages 84–89. TeX Users Group, March 1991.

[4] M. Herlihy. A methodology for implementing highly concurrent data objects. *ACM Trans. Program. Lang. Syst.*, 15(5):745–770, November 1993.

[5] L. Lamport. *LaTeX User's Guide and Document Reference Manual*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

N	L	D	F. Objetiva	Tempo (ms)
20	20	0.8	2.4	1002
		0.5	3.1	1543
		0.2	6.8	2156
30	30	0.8	2.8	1890
		0.5	3.7	4291
		0.2	7.5	6191
40	40	0.8	2.9	3021
		0.5	3.7	8018
		0.2	7.7	14583
50	50	0.8	3.0	6729
		0.5	4.5	15001
		0.2	8.8	3113
		Total	56.9	67538

Table 1: N é o número de nós, L é o número de rótulos e D é a densidade. A tabela mostra o tempo de execução de cada

[6] S. Salas and E. Hille. *Calculus: One and Several Variable*. John Wiley and Sons, New York, 1978.

APPENDIX

N	L	D	F. Objetiva	Tempo (ms)
100	25	0.8	1.8	11440
		0.5	2.0	16664
		0.2	4.6	46836
	50	0.8	2.0	31146
		0.5	3.0	34667
		0.2	6.9	107502
	100	0.8	3.8	37138
		0.5	5.3	96218
		0.2	11.3	192166
	125	0.8	4.1	60090
		0.5	6.2	113139
		0.2	13.1	239019
	Total –		64.1	986025

Table 2: N é o número de nós, L é o número de rótulos e D é a densidade. A tabela mostra o tempo de execução de cada

N	L	D	F. Objetiva	Tempo (ms)
200	50	0.8	2.0	53185
		0.5	2.3	130555
		0.2	5.7	306403
	100	0.8	2.9	99256
		0.5	4.1	157123
		0.2	9.4	574387
	200	0.8	4.6	254051
		0.5	6.8	655515
		0.2	15.3	1184675
	250	0.8	5.1	566244
		0.5	8.0	933404
		0.2	17.9	1441719
		Total –	84.1	6856514

Table 3: N é o número de nós, L é o número de rótulos e D é a densidade. A tabela mostra o tempo de execução de cada

N	L	D	F. Objetiva	Tempo (ms)
500	125	0.8	2.0	580854
		0.5	3.1	1535064
		0.2	7.3	4508425
	250	0.8	3.1	2351650
		0.5	4.9	6289072
		0.2	12.9	9052493
	500	0.8	5.8	6751334
		0.5	8.7	16560601
		0.2	21.3	16661087
	625	0.8	6.7	18957465
		0.5	10.5	17878234
		0.2	25.4	18310678
		Total –		

Table 4: N é o número de nós, L é o número de rótulos e D é a densidade. A tabela mostra o tempo de execução de cada