DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Relatório Técnico

RT-MAC-9515

Anais da I Oficina Nacional em Problemas Combinatórios: Teoria, Algoritmos e Aplicações

Carlos Eduardo Ferreira
Yoshiko Wakabayashi

Novembro 95

Anais da
I Oficina Nacional em
Problemas Combinatórios:
Teoria, Algoritmos e Aplicações

I Oficina Nacional em Problemas Combinatórios
(I ON ProComb)

IME–USP 1 a 3 de fevereiro de 1995

O Departamento de Computação do IME-USP está organizando a primeira Oficina Nacional em Problemas Combinatórios – teoria, algoritmos e aplicações. Esse evento faz parte do projeto "ProComb" que integra pesquisadores do DCC-IME-USP, DCC-Unicamp, PUC-Rio e UFRJ que atuam na área de Combinatória. O projeto ProComb é financiado pelo CNPq, dentro do Programa Temático Multiinstitucional em Ciência da Computação, ProTeM-CC-II.

O objetivo desse encontro é integrar a comunidade nacional da área de Combinatória com os membros do projeto, assim como divulgar os resultados obtidos no projeto. Além desse encontro está prevista a realização de mais 2 oficinas nacionais e 1 workshop internacional nos próximos dois anos.

Integrantes do projeto ProComb: Arnaldo Mandel (USP), Carlos Eduardo Ferreira (USP), Celia P. Mello (UNICAMP), Celina M. H. Figueiredo (UFRJ), Celso C. Ribeiro (PUC/RJ), Cid C. de Souza (UNICAMP), Jayme L. Szwarcfiter (UFRJ), João Meidanis (UNICAMP), João Carlos Setubal (UNICAMP), José Augusto R. Soares (USP), José Coelho de Pina Jr. (USP), Maria Angela Gurgel (USP), Marcia Cerioli (UFRJ), Marcus Vinicius Poggi (UNICAMP), Oscar Porto (PUC/RJ), Paulo Feofiloff (USP), Pedro Sergio de Souza (UNICAMP), Ricardo Dahab (UNICAMP), Sulamita Klein (UFRJ), Yoshiharu Kohayakawa (USP), Yoshiko Wakabayashi (USP) - coordenadora.

I Oficina Nacional em Problemas Combinatórios (I ON ProComb)

Sala de Conferências Antonio Gilioli, IME-USP

Sexta feira – 3 de fevereiro de 1995

Programa

9:00 Yoshiharu Kohayakawa (USP)
Paradigma de amostragem

10:00 Marcos N. Arenales (ICMSC/USP) e Reinaldo Morábito (DEP/UFSCar)
Abordagem em grafo-E/OU para problemas de corte/empacotamento

11:00 Café

11:30 Tomasz Luczak (Academia Polonesa de Ciências e Univ. Poznan)
A lossy data compression based on an approximate pattern matching

12:30 Almoço

14:30 Jayme L. Szwarcfiter (UFRJ), Célia P. de Mello (UNICAMP) e Celina M. Figueiredo (UFRJ)
Fontes, Sumidouros, Duplas Pares e Ímpares em Grafos de Comparabilidade

15:30 Abilio P. de Lucena (LNCC-IMPA)
Um novo algoritmo para calcular custos reduzidos numa árvore geradora de custo mínimo

16:30 Café

17:00 Célia P. de Mello (UNICAMP), Celina M. Figueiredo (UFRJ) e João Meidanis (UNICAMP)
Coloração de arestas, grafos de indiferença e grafos de grau máximo ímpar

18:00 Discussão Final

Índice

Abordagem em Grafo-E/OU para Problemas de Corte e Empacotamento

Marcos Arenales
Departamento de Ciências de Computação e Estatística
Instituto de Ciências Matemáticas de São Carlos
Universidade de São Paulo
e.mail: arenales@icmsc.sc.usp.br

Reinaldo Morábito
Departamento de Engenharia da Produção
Centro de Ciências e Tecnologia
Universidade Federal de São Carlos
e.mail: drmo@power.ufscar.br

1 Introdução

O problema de corte consiste, basicamente, em determinar a "melhor" forma de cortar unidades de material (aqui chamadas de *objetos*) de maneira a produzir um conjunto de unidades menores (itens). Este problema aparece em diversos processos industriais onde os objetos, em geral disponíveis em estoque correspondem a barras de aço, bobinas de papel e alumínio, chapas metálicas e de madeira, placas de circuito impresso, lâminas de vidro e fibra de vidro, peças de couro, etc..., e os itens, com dimensões especificadas, são em geral encomendados através de uma carteira de pedidos.

Similarmente, o problema de empacotamento consiste em determinar a "melhor" forma de arranjar um conjunto de itens dentro de objetos. Exemplos ocorrem nas atividades de movimentação, armazenagem e transporte de materiais, ao carregar produtos embalados (itens) sobre paletes, ou dentro de contêineres ou caminhões-cofre (objetos). Evidentemente o problema de corte e o problema de empacotamento estão intimamente relacionados e têm sido referidos na literatura como problemas de corte e empacotamento (PCE).

Nestes últimos 30 anos centenas de artigos foram publicados em diversas revistas especializadas (veja, por exemplo, as pesquisas bibliográficas em Sweeney e Parternoster, 1992, e Dyckhoff e Finke, 1992), sendo sensível o aumento do interesse acadêmico nos PCE, medido por exemplo pelo número de comunicações nos útimos congressos científicos. Artigos de revisão podem ser encontrados em Dowsland e Dowsland (1992), Golden (1976), Hinxman (1980).

Vários outros problemas estão estreitamente relacionados com os PCE, ampliando ainda mais o interesse neste tópico, como por exemplo o problema do carregamento de veículos, o problema de programação de veículos, o problema de alocação de tarefas, o problema de balanceamento de linha de montagem, o problema de orçamento de capital, o problema de programação de multi-processador, o problema de alocação de memória computacional, e

outros citados em Golden (1976) e Dyckhoff (1990). Nestes problemas os objetos e os itens, ao invés de terem dimensões espaciais como comprimento, largura, etc..., podem ter dimensões não espaciais como tempo, peso, etc...

Parte da motivação para a pesquisa dos PCE é devido a importância técnica e econômica das aplicações práticas. Pequenos ganhos percentuais no desempenho das atividades de corte ou empacotamento de materiais podem levar a economias substanciais, dependendo da escala de produção e de transportes. No problema de corte, por exemplo, uma pequena redução percentual no nível de aparas (restos de papel) de uma indústria de papel pode resultar numa vantagem competitiva significativa dentro deste setor. Similarmente, com o aumento da prática de paletização de carga na armazenagem e distribuição de produtos, pequenos ganhos no aproveitamento do espaço disponível de paletes e contêineres tornam-se significativos ao envolverem toda a cadeia logística percorrida por estes paletes e contêineres.

Um aspecto fundamental que tem motivado tantos autores para a pesquisa dos PCE é a ausência de abordagens de solução gerais e eficientes, devido principalmente à diversidade de casos em que os problemas podem aparecer e à complexidade dos algoritmos exatos disponíveis. Os PCE são problemas de otimização combinatória em geral da classe NP e portanto, de difícil solução exata. Não existem até então (e é improvável que venham a existir) abordagens de solução gerais e eficientes para os PCE. As abordagens, na maioria das vezes, baseiam-se num conhecimento específico do estudo de caso, o que em parte justifica a coleção de métodos heurísticos encontrados na literatura. Dyckhoff (1990) propõe uma classificação dos PCE, procurando salientar as principais características de tais problemas. Tais características são:

1. *De dimensão:* um problema é classificado como unidimensional quando apenas uma dimensão é relevante no processo de cortagem, por exemplo, corte de barras de aço na construção civil. Da mesma forma, bidimensional, tridimensional ou multidimensional.

2. *De seleção:* ou um subconjunto dos itens deve ser selecionado, pois os objetos são insuficientes, ou um subconjunto dos objetos deve ser selecionado pois são fornecidos em excesso.

3. *De disponibilidade de objetos:* pode-se ter apenas um objeto, diversos objetos com a mesma figura, ou ainda diversos com diferentes figuras.

4. *De demanda de itens:* pode-se ter poucos itens (de figuras diferentes) muitos itens de muitas figuras diferentes muitos itens com relativamente poucas figuras diferentes; ou ainda com a mesma figura.

Naturalmente esta classificação salienta apenas alguns aspectos dos PCE e, problemas de uma mesma classe podem diferir sobremaneira, como por exemplo, quanto à regularidade das figuras, tanto nos dados do problema, como nas figuras obtidas no processo de cortagem.

Neste trabalho enfatizamos uma abordagem para PCE, tratando alguns problemas clássicos da literatura, onde apenas um objeto é disponível, sendo a seleção, portanto, de um subconjunto de itens (este é um subproblema fundamental quando muitos objetos são disponíveis). Além disso, consideramos a demanda de muitos itens com poucas figuras (conhecido na literatura como problema irrestrito) e estendemos a abordagem também para poucos itens (chamados problemas restritos).

A formulação combinatória destes problemas é definido a seguir. Considere que um objeto seja cortado para produção de possíveis m tipos de itens (m figuras diferentes, cujas geometrias e medidas são fornecidas), onde cada item tipo i tem um valor de utilidade $v_i, i = 1 \ldots m$.

Um arranjo de um subconjunto de itens no objeto é chamado padrão de corte. A um padrão de corte associamos um vetor y de inteiros não-negativos, onde y_i é o número de itens do tipo i presente no padrão. Seja $\mathbb{P}$ definido por:

$$\mathbb{P} = \{(y_1, \ldots, y_m) \text{ tal que exista pelo menos um padrão de corte com } y_i \text{ unidades do item tipo } i, \quad i = 1, \ldots, m\}$$

O problema irrestrito é então formulado por:

$$\begin{array}{l} \max \ v_1 y_1 + v_2 y_2 + \ldots + v_m y_m \\ \text{sujeito à:} \ \ (y_1, y_2, \ldots, y_m) \in \mathbb{P}. \end{array}$$

No problema restrito é adicionada as restrições: $y_i \leq b_i, \quad i = 1, \ldots, m$, onde b_i é o número máximo de itens permitido no padrão de corte.

2 Problemas Bidimensionais

2.1 Caso Guilhotinado e Irrestrito

Problema de corte bidimensional, guilhotinado e irrestrito trata de cortar um objeto retangular, de medidas (A, B), em m itens retangulares, de medidas (a_i, b_i), $i = 1, \ldots, m$. Os possíveis padrões de cortes são obtidos por uma sequência de "cortes elementares" que aplicados a um retângulo produzirá 2 novos retângulos (cortes guilhotinados), ou manterá intacto o retângulo (corte nulo, denotado por corte-0). Os cortes guilhotinados podem ser ainda verticais ou horizontais.

O processo de cortagem (e, por conseguinte, os padrões de corte) pode ser representado por um grafo-E/OU, onde os vértices representam os retângulos obtidos e os arcos representam os cortes. Note que um corte guilhotinado relaciona um vértice com outros dois vértices, os quais separadamente não fornecem uma solução para o predecessor. Este tipo de arco é chamado arco-E.

Um grafo-E/OU estende a noção usual de grafo, onde os arcos são pares de vértices. Num grafo-E/OU os arcos são pares de subconjunto de vértices. Assim, se os subconjuntos forem unitários, teremos o conceito usual de grafos.

No grafo-E/OU definido acima, um vértice inicial representa o retângulo (A, B) e vértices finais representam retângulos obtidos por corte-0, ou seja, um corte-0 interrompe o processo de cortagem. Os vértices finais podem representar itens ou perdas (ou espaço ocioso, no caso de empacotamento).

Um caminho completo, ligando o vértice inicial a vértices finais, corresponde a um padrão de corte, pois fornece uma sequência inequívoca de cortes e, portanto, representa uma solução factível para o problema. O problema pode então ser resolvido por uma busca exaustiva de todos os caminhos completos, isto é, uma enumeração explícita de todos os vértices do grafo. Embora o grafo seja finito (pois os cortes podem ser restritos a um conjunto finito, conforme Herz, 1972), este trabalho é infactível computacionalmente. Entretanto, um método do tipo "branch-and-bound", isto é, um método de enumeração implícita, pode ser descrito com o

uso de limitantes facilmente definidos a partir de uma solução factível trivial onde apenas um tipo de item é considerado, chamada solução homogênea e de uma relaxação do problema, onde apenas a restrição de área é considerada.

Estes limitantes podem ser úteis na produção de heurísticas muito eficazes. Em Morábito et al (1992) foi proposto um algoritmo para o problema de corte bidimensional, guilhotinado e irrestrito, baseado nesta abordagem em grafo-E/OU.

2.2 Caso Guilhotinado, Estagiado e Restrito

O problema restrito (embora as chamadas restrições sejam simples: $y_i \leq b_i$, onde b_i é o número máximo de itens permitido do tipo i no padrão de corte) torna-se muito mais difícil de resolução.

A dificuldade pode ser percebida pela abordagem em grafo-E/OU, pois os vértices decorrentes de uma ramificação (corte), chamados vértices-E, não são independentes. Um método heurístico, baseado na heurística gulosa e no grafo-E/OU da seção anterior, pode ser desenvolvido juntamente com uma busca primeiro-em-profundidade, onde um dos vértices-E é primeiro resolvido (isto é, determina-se o padrão de corte para um dos retângulos) com a limitação de retângulos imposta pelo vértice predecessor, em seguida, resolve-se o outro vértice-E com a limitação de retângulos imposto pela folga do outro vértice-E já resolvido. Os limitantes devem ser ligeiramente modificados por causa da restrição de limitação no número de retângulos admissíveis.

Um problema de corte guilhotinado é dito estagiado, se existir um número máximo de estágio admissível, onde um novo estágio é obtido sempre que um corte aplicado a um retângulo for ortogonal ao corte que gerou aquele retângulo. Este problema é facilmente tratado pela abordagem em grafo-E/OU, acrescentando-se por vértice uma variável lógica indicando se o corte, que gerou aquele vértice, foi vertical ou horizontal, e uma variável inteira indicando o estágio correspondente. Ligeiras modificações nos limitantes devem também ser feitas.

O problema de corte bidimensional, guilhotinado, estagiado e restrito foi estudado em Morábito e Arenales (1992), usando-se a abordagem em grafos-E/OU. As heurísticas baseadas nos limitantes foram usadas, produzindo excelentes resultados.

2.3 Caso Não-Guilhotinado e Irrestrito

O problema de corte não-guilhotinado, envolvendo apenas retângulos, admite padrões de cortes que não são representados pelo grafo-E/OU da seção 2.1.

Entretanto, é possível formular um novo grafo-E/OU incluindo-se um tipo especial de corte elementar não-guilhotinado, o qual produz 5 novos retângulos (4 nos cantos e um central). Assim, o grafo-E/OU deve considerar um outro tipo de arco que relaciona um vértice com outros 5 vértices.

O método branch-and-bound desenvolvido para o caso guilhotinado é naturalmente estendido. Pequenas alterações podem ser introduzidas nos limitantes e regras especiais para se evitar duplicação de padrões são desenvolvidas.

Este problema foi estudado em Arenales e Morábito (1995) utilizando a abordagem em grafo-E/OU com resultados promissores.

3 Problemas Tridimensionais

A extensão para problemas tridimensionais da abordagem em grafo-E/OU, desenvolvida para problemas bidimensionais, é também natural.

O objeto agora consiste na caixa (A, B, C) e os itens em caixas menores $(a_i, b_i, c_i), i = 1 \ldots m$. Um padrão de corte guilhotinado é obtido por aplicar-se sucessivamente cortes elementares que dividem uma caixa em duas outras caixas. Os vértices do grafo representam caixas e os arcos, como anteriormente, representam os cortes. Padrões não-guilhotinados podem também ser produzidos por considerar arcos-E que levam vértices em outros 5 vértices. Limitantes são definidos de modo similar ao caso bidimensional.

Em Morábito e Arenales (1994), a abordagem em grafo-E/OU, bem como o método branch-and-bound e heurísticas, foram estendidos para o problema de carregamento de contêineres Regras especiais para estabilidade da carga foram desenvolvidas em sintonia com a busca no grafo. Os resultados computacionais, mais uma vez indicam que tal abordagem é promissora na resolução de problemas práticos.

Um texto completo sobre a abordagem em grafo-E/OU para problemas uni-, bi- e tridimensionais pode ser encontrado em Morábito (1992).

4 Extensões

A abordagem em grafo-E/OU, aplicada a problemas bi e tridimensionais nas seções 2 e 3 se aplica a qualquer problema de corte que satisfaça as seguintes hipóteses:
Hipóteses sobre objeto e itens:

- O objeto e os itens são sub-conjuntos conexos e limitados, com interiores não-vazios, de um espaço n-dimensional, onde n é o número de dimensões relevantes.

Hipóteses sobre a formação de padrões de cortes (processo de cortagem):

- A cortagem se processa de forma sequencial:
  a partir da aplicação de cortes elementares (definidos a priore) sobre o objeto original, são produzidos novos objetos desconexos entre si, sobre os quais também serão aplicados cortes elementares, e assim sucessivamente.
- Todo item pode ser obtido por uma sequência finita de cortes elementares.
- São considerados equivalentes os padrões de corte que produzem o mesmo número de itens de cada tipo.

Desta forma, esta abordagem abre caminhos para tratar problemas ainda mais complexos. Naturalmente a viabilidade computacional depende do número de informações necessárias por vértice. O conceito de regularidade de um problema pode ser revisto a partir da limitação sobre o número de informações necessárias para armazenar os objetos intermediários.

Estas extensões estão em Arenales (1993), onde um método branch-and-bound e heurísticas foram desenvolvidos para um problema de corte qualquer que satisfaça às hipóteses acima.

Referências

[ARENALES] ARENALES, M. (1993). Uma teoria para o problema de corte. Tese de Livre-Docência. ICMSC-USP, São Carlos-SP.

[ARENALES] ARENALES, M. (1995). An AND/OR-Graph Approach to the Solution of Two-Dimensional Non-Guillotine Cutting Problems. European Journal of Operational Research (a aparecer).

[DOWSLAND] DOWSLAND, K. e DOWSLAND, W. (1992). Packing Problems. European Journal of Operational Research, v.56, pp. 2-14.

[DYCKHOFF] DYCKHOFF, H. (1990). A Typology of Cutting and Packing Problems. European Journal of Operational Research, v.44, p. 145-159.

[DYCKHOFF] DYCKHOFF, H. e FINKE, U. (1992). Cutting and Packing in Production and Distribution: Typology and Bibliography. Springer-Verlag Co, Heildelberg.

[GILMORE] GILMORE, P. e GOMORY, R. (1965). MultiStage Cutting Stock Problems of Two and More Dimensions. Operations Research, v. 14, p.1045-1074.

[GOLDEN] GOLDEN, B. (1976). Approaches to the Cutting Stock Problem. AIIE Transactions, v.8, p.265-274.

[HERZ] HERZ, J. (1972). Recursive Computational Procedure for Two Dimensional Stock Cutting. IBM Journal of Research and Development, v.16, p.462-469.

[HINXMAN] HINXMAN, A. (1980). The Trim-Loss and Assortment Problems: A Survey. European Journal of Operational Research, v. 5, p.8-18.

[MORÁBITO] MORÁBITO, R. (1992). Uma Abordagem em Grafo-E/OU para o Problema do Empacotamento: Aplicação ao Carregamento de Paletes e Contêineres. Tese de Doutorado. Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos.

[MORÁBITO] MORÁBITO, R. e ARENALES, M. (1992). Staged and Constrained Two-Dimensional Guillotine Cutting Problems: A New Approach. Notas do ICMSC nº 126, Universidade de São Paulo.

[MORÁBITO] MORÁBITO, R. e ARENALES, M. (1994). An AND/OR-Graph Approach to the Container Packing Problem. International Transactions in Operational Research, v.1, pp. 59-73.

[MORÁBITO] MORÁBITO, R., ARENALES, M. e ARCARO, V. (1992). An And/Or-Graph Approach for Two-Dimensional Cutting Problems. European Journal of Operational Research, v. 58(2), p. 263-271.

Edge-Colouring, Indifference Graphs and Odd Maximum Degree Graphs

Celina M. H. de Figueiredo* João Meidanis†
Célia Picinin de Mello†

An *edge-colouring* of a graph is an assignment of colours to its edges such that no adjacent edges have the same colour. The *chromatic index* of a graph is the minimum number of colours required to produce an edge-colouring for that graph.

An easy lower bound for the chromatic index is the maximum vertex degree. A celebrated theorem by Vizing states that these two quantities differ by at most one [12]. Graphs whose chromatic index equals the maximum degree are said to be *Class* 1; graphs whose chromatic index exceeds the maximum degree by one are said to be *Class* 2.

Despite the powerful restriction imposed by Vizing's result, it is very hard to compute the chromatic index in general. The problem is NP-complete for general graphs [6] and for various subclasses, such as perfect graphs and 3-regular graphs [1, 6]. Very little is known about the complexity of computing the chromatic index for other classes, and the our goal is to investigate this question. We focus our attention on the class of *indifference* graphs because this is a relatively small class and much is known about it, and on graphs with *odd maximum degree* because, perhaps surprisingly, these graphs tend to be Class 1 more often then graphs with even maximum degree.

Indifference graphs are graphs whose vertices can be linearly ordered so that the vertices contained in the same maximal clique are consecutive in this order [11]. We

*Universidade Federal do Rio de Janeiro, Instituto de Matemática, Caixa Postal 68530, 21944 Rio de Janeiro, RJ, Brasil. Fax: +55 21 2901015. celina@cos.ufrj.br. Partially supported by CNPq, grant 301160/91.0.

†Universidade Estadual de Campinas, Departamento de Ciência da Computação, Caixa Postal 6065, 13081-970 Campinas, SP, Brasil. Fax: +55 192 397470. {meidanis,celia}@dcc.unicamp.br, Partially supported by FAPESP and CNPq.

shall call such an order an *indifference order*. Indifference graphs form an important subclass of interval graphs.

We are also interested in the role played by *overfullness* in determining the chromatic index. By definition of edge-colouring, each colour determines a matching and can cover at most $\lfloor n/2 \rfloor$ edges, where n is the number of nodes. Therefore, if the total number of edges is greater than the product of the maximum degree by $\lfloor n/2 \rfloor$, the graph is necessarily Class 2. Graphs to which this argument can be applied are called *overfull*. More generally, if a graph has an overfull subgraph with the same maximum degree, then the same counting argument shows that the supergraph is Class 2. We call such graphs *subgraph-overfull*. We consider classes of graphs for which being Class 2 is equivalent to being subgraph-overfull. For such graphs the chromatic index problem is in P: a decomposition algorithm due to Padberg and Rao [8] checks in polynomial time whether a graph is subgraph-overfull. We note that historical results of König on bipartite graphs and of Tait on 3-regular planar graphs show that these two classes of graphs are trivial examples: any set of Class 1 graphs satisfies vacuously the above equivalence [3]. On the other hand, it has been shown recently that every Class 2 multipartite complete graph is subgraph-overfull [5].

Hilton conjectured that a lower bound on the maximum degree would ensure the equivalence between being Class 2 and being subgraph-overfull [4]. Every Class 2 graph with either a universal vertex or a quasi-universal vertex is subgraph-overfull [9, 10]. As noted by Ortiz [7], this result can be used to compute efficiently the chromatic index for the following two subclasses of indifference graphs: clique-complete indifference graphs and split-indifference graphs.

We consider a new version of overfullness that is not as powerful as subgraph-overfullness but is trivially checkable: a graph is said to be *neighbourhood-overfull* when it has a maximum degree vertex whose neighbourhood induces an overfull subgraph. Every neighbourhood-overfull graph is subgraph-overfull, but the converse is not valid (see Figure 1). We consider classes of graphs for which being Class 2 is equivalent to being neighbourhood-overfull.

We prove that every indifference graph with odd maximum degree is Class 1. Since graphs with an even number of vertices cannot be overfull, graphs with odd maximum degree cannot be neighbourhood-overfull. Hence, being Class 2 and being neighbourhood-overfull are vacuously equivalent for indifference graph with odd maximum degree. We extend these results by considering classes of graphs that are defined by special vertex *perfect elimination orders*. As a result, we prove that every doubly chordal graph with odd maximum degree is Class 1.

We also prove that Class 2 and neighbourhood-overfullness are equivalent for

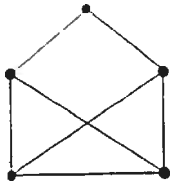

Figure 1: A subgraph-overfull graph which is not neighbourhood-overfull.

indifference graphs with three maximal cliques and no universal vertex. Since every non indifference graph with three maximal cliques must contain a universal vertex, our result actually says that Class 2 and neighbourhood-overfullness are equivalent for general graphs with at most three maximal cliques.

We state the following conjecture for indifference graphs.

Conjecture 1 *The following statements are equivalent for a given indifference graph:*

- *(1) the graph is neighbourhood-overfull;*
- *(2) the graph is subgraph-overfull;*
- *(3) the graph is Class 2.*

It should be noted that this conjecture, if true, provides a way of deciding in polynomial time whether an indifference graph is Class 1 or Class 2. We have described a linear-time algorithm for indifference graph recognition in [2].

On the other hand, in view of our results concerning graphs with odd maximum degree, it is interesting to ask what is the largest graph class for which odd maximum degree implies Class 1. We conjecture that this is true at least for chordal graphs.

Conjecture 2 *All chordal graphs with odd maximum degree are Class 1.*

References

[1] L. Cai and J. A. Ellis. NP-completeness of edge-colouring some restricted graphs. *Discrete Applied Mathematics*, 30:15–27, 1991.

[2] C. M. H. de Figueiredo, J. Meidanis, and C. P. de Mello. A lexbfs algorithm for proper interval graph recognition. Technical report, Departamento de Ciência da Computação - UNICAMP, 1993. Submitted to *Inform. Process. Lett.*

[3] S. Fiorini and R.J. Wilson. *Edge-colourings of graphs.* Pitman, London, 1977.

[4] A. J. W. Hilton. Two conjectures on edge-colouring. *Discrete Math.*, 74:61–64, 1989.

[5] D. G. Hoffman and C. A. Rodger. The chromatic index of complete multipartite graphs. *J. of Graph Theory*, 16:159–16364, 1992.

[6] I. Holyer. The NP-completeness of edge-coloring. *SIAM J. Comput.*, 10:718–720, 1981.

[7] C. Ortiz. *Sobre coloração de arestas.* PhD thesis, COPPE-UFRJ, Rio de Janeiro. 1994. In portuguese.

[8] M. W. Padberg and M. R. Rao. Odd minimum cut-sets ans b-matching. *Math. Oper. Res.*, 7:67–80, 1982.

[9] M. Plantholt. The chromatic index of graphs with a spanning star. *J. of Graph Theory*, 5:45–53, 1981.

[10] M. Plantholt. The chromatic index of graphs with large maximum degree. *Discrete Math.*, 47:91–96, 1983.

[11] F. S. Roberts. On the compatibility between a graph and a simple order. *J. Comb. Theory (B)*, 11:28–38, 1971.

[12] V. G. Vizing. On an estimate of the chromatic class of a p-graph. *Diket. Analiz.*, 3:25–30, 1964. In russian.

THE SAMPLING PARADIGM OF KARGER

Y. KOHAYAKAWA

§0. INTRODUCTION AND DISCLAIMER

In the past couple of years, two simple observations of Karger have been exploited by a few authors with quite surprising results. Our aim here is to say a few words on what the observations are and to discuss its consequences. We shall not try to be comprehensive; instead, we shall concentrate on the basics with the hope that the interested readers might be at a good starting point when turning to the literature, which seems to be growing quite rapidly.

At this point I should like to insert a disclamer: I have been actively interested in this area only quite recently, and I do not claim to present new ideas or insights here. The purpose of this short note is to stir up the readers' interest in Karger's method, with the hope that members of Project ProComb and possibly others will work on their problems with this new tool in hand.

In the sequel, we shall discuss both algorithms and purely combinatorial results. A common feature of the topics below is that probability will always be lurking in the background. In particular, we shall concentrate on randomised algorithms, although derandomisation has proved to fit in nicely in parts of what follows.

Let us recall the two most common types of randomised algorithms. A *Monte Carlo* type algorithm of time complexity $T(n)$ finishes its computation in time not exceeding $T(n)$ for inputs of size n, but might give an erroneous answer with some small probability, usually approaching 0 as $n \to \infty$. On the other hand, a *Las Vegas* type algorithm of time complexity $T(n)$ finishes its computation in time $T(n)$ with high probability (usually $1 - o(1)$ where the $o(1)$ term refers to $n \to \infty$), and its output is always correct.

For conciseness, we shall always assume that G is a graph of order $|G| = |V(G)| = n$ and size $e(G) = |E(G)| = m$. Our graphs will be allowed to have multiple edges.

§1. MINIMUM SPANNING TREES

A recent success in the theory of randomised computing was the development of a Las Vegas *linear* time algorithm for finding a minimum spanning tree in an edge-weighted graph G. The currently best deterministic algorithm, due to Gabow, Galil, and Spencer [3] (see also [4]), runs in time $O(m \log \beta(m, n))$, where $\beta(m, n) = \min\{i: \log^{(i)} n \leq m/n\}$ is the inverse Ackerman function. Thus here we have another piece of evidence that randomisation does help.

Work partially supported by FAPESP (Proc. 93/0603-1) and by CNPq (Proc. 300334/93-1 and Project ProComb).

The linear time algorithm above was developed by Klein and Tarjan [11], following a simple but powerful observation of Karger [6].

Theorem 1. *There is a Las Vegas algorithm that takes a connected edge-weighted graph G as input and outputs a minimum spanning tree for G in time $O(m)$ with probability $1 - \exp\{-m/\log^{O(1)} m\}$.*

In the result above, the model of computation is that of a unit-cost random-access machine with the restriction that only binary comparisons are allowed with the edge weights. Putting it very crudely (in fact, in an *oversimplified* manner), the algorithm of Klein and Tarjan behaves as follows: first, it randomly chooses a half of the edges of G to be discarded, and it concentrates its attention on the remaining edges. Thus the graph G is reduced to a graph G' of size roughly $e(G)/2$. The algorithm is then called recursively with input G'. The output T' of this recursive call is not necessarily a minimum spanning tree of G, and hence in the rest of the algorithm it is worked on to become one such tree T: here the crucial point is that with very high probability very little work will be needed to obtain T from T'. Thus, in short, the algorithm is based on *sampling* the edges of G with the aim of picking a substantially smaller subgraph G' that is a good 'picture' of G.

The key result connecting the cost of T' and the actual cost of a minimum spanning tree of G is a 'sampling lemma' of Klein and Tarjan. To state this result, let us say that an edge e of G is *T'-light* if it is either contaned in T' or else if it is not contained in T' but its weight is no greater than the weight of some edge of T' in the fundamental cycle of e with respect to T'. The sampling lemma of Klein and Tarjan [11] is as follows. We may and shall assume that the edge weights of our graph G are all distinct.

Theorem 2. *Let G' be obtained from the edge-weighted graph G by independently removing each edge of G with probability $1 - p$, and let F' be a minimum spanning forest of G'. Then, for any k, the probability that the number of F'-light edges in G exceeds k is at most*

$$\sum_{0 \le i \le n-2} \binom{k}{i} p^i (1-p)^{k-i}.$$

An immediate corollary of the sampling lemma is that the expected number of F'-light edges is at most n/p, and that the probability that the number of such edges exceeds $(1+\varepsilon)n/p$ is $\exp\{-\Omega(\varepsilon^2 n)\}$ for any $0 \le \varepsilon \le 1$.

We close this section with the following remark. The brief description of the algorithm of Klein and Tarjan above is far from complete: the sampling lemma, the algorithm sketched above, and some standard ideas only give an algorithm of time complexity $O(m + n \log n \log\log n)$. The linear algorithm actually makes use of an operation known as the Borůvka reduction and the transformation of T' into T involves the use of sophisticated data structures. Our aim in the presentation above was to focus on the sampling paradigm of Karger, which in this particular case amounts to the use of Theorem 2.

§2. Cuts, Flows, and Network Design

2.1 Cuts. Edge-contraction has been used by Nagamochi and Ibaraki [12] and Matula [13] to develop fast algorithms for determining or estimating the edge

connectivity or, equivalently, the minimum cut of a given graph. A simple but important remark of Karger [5] connected to edge-contraction and minimum cuts is best illustrated by an algorithm for finding a minimum cut in a given unweighted graph G. The discussion that follows concentrates on the unweighted case, but similar results may be obtained for weighted graphs. In §2, we shall always assume that G has minimum cut c. Here is the algorithm of Karger: given a graph G, pick an edge e of G uniformly at random and contract e. Remove all loops that arise. Proceed contracting the edges of our graph and removing loops until only two vertices remain. The cut naturally corresponding to the resulting 2-vertex graph is the output of the algorithm.

Of course, the algorithm above does not have much chance of producing a minimum cut of G unless one is lucky, but in fact one need not to be *too* lucky. This is the key observation of Karger.

Theorem 3. *Fix a minimum cut C of G. The algorithm above produces C as output with probability at least $\binom{n}{2}^{-1}$.*

Proof. Note first that the cut C is the output of our algorithm if and only if no edge of C is contracted in the process. Suppose we are executing our procedure and we have so far contracted t edges ($0 \le t < n-2$). The current graph G_t has minimal degree at least $|C|$, as contraction of edges does not decrease the minimum cut of our graph. Thus G_t has at least $|C|(n-t)/2$ edges. The probability that we chose an edge from C to be contracted is thus not larger than $2/(n-t)$. Therefore the probability that an edge of C is never chosen to be contracted is at least

$$\left(1-\frac{2}{n}\right)\left(1-\frac{2}{n-1}\right)\cdots\left(1-\frac{2}{3}\right)=\binom{n}{2}^{-1},$$

as required. □

An immediate corollary to the result above is a polynomial time Monte Carlo algorithm for finding a minimum cut of G. More interestingly, one deduces from Theorem 3 that any graph G on n vertices has at most $\binom{n}{2}$ minimum cuts. Karger in fact proves the following general result.

Theorem 4. *Let G be a graph with minimum cut c, and let $\alpha \ge 1$ be a real number. Then the number of cuts in G of cardinality at most αc is at most $n^{2\alpha}$.*

A moment's thought suggests that the result above must be very useful in sampling. Suppose we delete each edge of a graph G independently with probability $1-p$, and let G_p be the resulting graph. In this process, *any* fairly small cut C of G is bound to be reduced to a cut of size precisely $(1+o(1))p|C|$ provided only that the minimum cut of G is a little more than of logarithmic size: this is an immediate consequence of Theorem 4 and the exponential decay of the tail of the binomial distribution. A more careful argument gives the following result of Karger [7].

Theorem 5. *Let G be an n-vertex graph whose minimum cut has value c, let $p = 2(d+2)(\log n)/\varepsilon^2 c$, where d is some fixed constant, and suppose $0 < p < 1$. Then the*

cardinality c_p of a minimum cut of G_p satisfies $|c_p - pc| \le \varepsilon pc$ with probability $1 - O(n^{-d})$.

Theorem 5 suggests a simple algorithm for approximating the value of a minimum cut of a given graph G. Clearly, results similar to the above theorem may be proved for minimum s–t cuts for any fixed s and $t \in G$.

Theorem 6. *Let G be an n-vertex graph and fix s and $t \in G$. Suppose that G has minimum cut c and that the minimum s–t cut in G has value v. Suppose $0 < p < 1$ satisfies $p = \Theta((\log n)/\varepsilon^2 c)$. Then, with probability $1 - o(1)$ as $n \to \infty$, the value v_p of the minimum s–t cut in G_p is such that $|v_p - pv| \le \varepsilon pv$.*

Theorem 6 gives a Monte Carlo algorithm for finding a $(1+\varepsilon)$-approximation for the minimum s–t cut in a given graph G that runs in time roughly $mv/\varepsilon^4 c^2$. Faster algorithms were found by Karger, elaborating on the ideas presented in the sequel. Finally, we remark that, using more sophisticated techniques, Karger has obtained a Las Vegas algorithm for computing the value c of a minimum cut in time $\tilde{O}(mc^{1/2})$. (Here and in the sequel, $\tilde{O}(f)$ stands for $O(f \operatorname{polylog} f)$.)

2.2 Las Vegas algorithms for maximum flow. The results of the previous section have direct consequences in the study maximum flows in graphs. Consider the following randomised divide-and-conquer algorithm for finding a maximum family of edge-disjoint paths between two fixed vertices s and t of a given graph G. We first split G into two edge-disjoint spanning subgraphs G_1 and G_2, randomly colouring the edges of G red and blue. We then recursively run the algorithm with input $(G_i; s, t)$ $(i \in \{1,2\})$, put together the two families of s–t paths to form a family $\mathcal{F}'$, and use an algorithm for augmenting paths to obtain an optimal family $\mathcal{F}$. The crucial observation is that $\mathcal{F}'$ will actually be very close to being optimal as long as G has reasonably high edge-connectivity. Indeed, Theorem 6 implies that $|\mathcal{F}| - |\mathcal{F}'|$ is bounded by $\tilde{O}(|\mathcal{F}|c^{-1/2})$.

Theorem 7. *The procedure above gives a Las Vegas algorithm that finds a maximum edge-disjoint family of s–t paths in the graph G in time $\tilde{O}(m|\mathcal{F}|c^{-1/2})$.*

2.3 Network design and other problems. We start with 'other problems'. We only wish to point out that, given a $2k$-edge-connected graph G, a random orientation of G is very likely to have edge-connectivity $(1 - o(1))k$, provided only that k is superlogarithmic. The following is a more precise statement that may be proved with the aid of Theorem 4.

Theorem 8. *There exists a linear time Monte Carlo type algorithm for finding a $(k - O\{(k \log n)^{1/2}\})$-edge-connected orientation of any n-vertex $2k$-edge-connected graph.*

The algorithm above may be used as a component of a Las Vegas algorithm that runs faster than the currently best deterministic algorithm for this problem, due to Gabow [2]. The speeding up factor is of order $\tilde{O}(k^{1/2})$.

We now turn to network designs. It is best if we restrict our attention on one specific problem. Suppose a graph G is given. We wish to find a spanning k-edge-connected subgraph of G with the smallest possible number of edges. This problem is NP-complete even for $k = 2$. However, a *fractional* solution to our problem may

be found in polynomial time through the use of interior point algorithms, as shown by Williamson, Goemans, Mihail, and Vazirani [14]. In the fractional solution, we assign to each edge $e \in E(G)$ a weight $0 \le p_e \le 1$. Interpreting the edge weights as probabilities, we may obtain a random subgraph of G that is likely to be close to an optimal solution as long as k is superlogarithmic. Specifically, this 'randomised rounding' procedure gives the following result.

Theorem 9. *There exists a polynomial time Las Vegas algorithm for finding a* $(1 + O\{((\log n)/k)^{1/2}\})$*-approximation to the minimum* k*-connected subgraph.*

The proof of Theorem 9 above is based on an obvious variant of Theorem 5, on a simple 2-approximation algorithm for our problem (to transform the almost optimal solution given by the random rounding procedure into an optimal one), and, of course, on the interior point method of [14].

§3. Matroids

Theorem 2 has an immediate generalisation to matroids. Thus, the problem of finding a minimum cost basis in a matroid may be approached through sampling. Since the ideas involved in this more general setting are quite similar to the ones briefly discussed in §1, we shall not dwell on them. We shall, however, mention two further results of Karger [6] concerning random submatroids of a matroid.

Let M be a matroid on S, and write $\rho_M = \rho(M)$ for the rank of M. Suppose $0 \le p \le 1$ is given, and let M_p be the random matroid obtained from M by restricting M to S_p, the random subset of S in which each element of S is independently present with probability p. Thus to obtain M_p we simply delete elements from M independently with probability $1 - p$. Let $\pi(M)$ be the *packing number* of M, *i.e.*, the maximal number of pairwise disjoint bases it contains.

Theorem 10. *Let* $0 < p < 1$ *be given and suppose the matroid* M *has packing number* $\pi(M) = a + 2 + p^{-1}\log\rho(M)$. *Then* M_p *contains a basis of* M *with probability* $1 - O(e^{-pa})$.

We remark that the result above is strong enough to give the right order of magnitude for the connectivity threshold for the binomial random graph $G_{n,p}$. Indeed, Theorem 10 shows that $G_{n,p}$ is almost surely connected if $p = p(n) = (2 + \varepsilon)(\log n)/n$ and $\varepsilon > 0$ is a fixed constant.

In view of Theorem 10, one may ask whether if M has a large packing number, then M_p has a large packing number as well or, more precisely, whether M_p contains many pairwise disjoint bases of M. The following result of Karger answers this question.

Theorem 11. *Let* M *and* p *be as above, and write* $\pi'(M_p)$ *for the maximal number of pairwise disjoint bases of* M *that* M_p *contains. Then, for any* $0 \le \varepsilon \le 1$, *we have*

$$\mathbb{P}\{|\pi'(M) - p\pi(M)| > \varepsilon p\pi(M)\} < \rho(M)\exp\left\{-\frac{1}{2}\varepsilon^2 p\pi(M)\right\}.$$

An immediate corollary to Theorem 11 is the following.

Theorem 12. *Let $0 < p = p(n) < 1$ be given and suppose the n-vertex graph G contains π disjoint spanning trees. If π' denotes the maximal number of pairwise disjoint spanning trees of G that G_p contains, then we have $|\pi' - p\pi| \le 2(p\pi \log n)^{1/2}$ with probability $1 - o(1)$ as $n \to \infty$.*

§4. Concluding Remarks

We hope that the usefulness of Theorems 2 and 4 have been well illustrated by the above discussion. Further applications and extensions may be found in the work of Karger and others. In particular, many of the algorithms based on sampling are parallelisable (see, *e.g.*, Cole, Klein, and Tarjan [1]) and, furthermore, derandomisation has been achieved for a minimum cut algorithm, showing that determining a minimum cut in a graph is in NC (see Karger and Motwani [9]).

Perhaps more purely combinatorial results along the lines of Theorems 10 and 11 should be sought. Moreover, perhaps difficult algorithmic problems for matroids could be tackled with these methods; *e.g.*, can the intersection problem for three matroids be solved through sampling?

References

1. Cole, R., Klein, P.N., Tarjan, R.E., *A linear-work parallel algorithm for finding minimum spanning trees*, SPAA, 1994.
2. Gabow, H.N., *A frame-work for cost-scaling algorithms for submodular flow problems*, 34th FOCS, 1993, pp. 449–458.
3. Gabow, H.N., Galil, Z., Spencer T., *Efficient implementation of graph algorithms using contraction*, 25th FOCS, 1984, pp. 347–357.
4. Gabow, H.N., Galil, Z., Spencer T., Tarjan, R.E., *Efficient algorithms for finding minimum spanning trees in undirected and directed graphs*, Combinatorica **6** (1986), 109–122.
5. Karger, D.R., *Global min-cuts in $\mathcal{RNC}$, and other ramifications of a simple min-cut algorithm*, 4th SODA, 1993, pp. 21–30.
6. Karger, D.R., *Random sampling in matroids, with applications to graph connectivity and minimum spanning trees*, 34th FOCS, 1993, pp. 84–93.
7. Karger, D.R., *Random sampling in cut, flow, and network design problems*, 26th STOC, 1994, pp. 648–657.
8. Karger, D.R., *Using randomised sparsification to approximate minimum cuts*, 4th SODA, 1994.
9. Karger, D.R., Motwani, R., *Derandomization through approximation: an $\mathcal{NC}$-algorithm for minimum cuts*, 26th STOC, 1994, pp. 497–506.
10. Karger, D.R., Stein, C., *An $\tilde{O}(n^2)$ algorithm for minimum cuts*, 25th STOC, 1993, pp. 757–765.
11. Klein, P.N., Tarjan, R.E., *A randomized linear-time algorithm for finding minimum spanning trees*, 26th STOC, 1994, pp. 9–15.
12. Nagamochi, H., Ibaraki, T., *Computing edge connectivity in multigraphs and capacitated graphs*, SIAM J. Discrete Math. **5** (1992), 54–66.
13. Matula, D.W., *A linear $2+\varepsilon$ approximation algorithm for edge connectivity*, 4th SODA, 1993, pp. 500–504.
14. Williamson, D., Goemans, M.X., Mihail, M., Vazirani, V.V., *A primal-dual approximation algorithm for generalized Steiner problems*, 25th STOC, 1993, pp. 708–717.

Instituto de Matemática e Estatística, Universidade de São Paulo, Caixa Postal 20570, 01452-990 São Paulo SP, Brazil

E-mail address: <yoshi@ime.usp.br>

An algorithm to compute reduced costs on a spanning tree

Abilio Lucena

Laboratório Nacional de Computação Científica
Rio de Janeiro
Brasil 22290
E-mail:lucena@alpha.lncc.br

and

Yazid M. Sharaiha

The Management School
Imperial College of Science, Technology and Medicine
London, United Kingdom
E-mail:y.sharaiha@ic.ac.uk

February 1995

1 Introduction

The problem of computing reduced costs on a minimum spanning tree (MST) has a number of applications in graph theory and combinatorial optimization. The reduced costs are typically used within Lagrangean relaxation algorithms for solving some combinatorial optimization problems. For example, both Volgenant and Jonker [10] and Carpaneto et al [2] have used MST reduced costs and branch-chord exchanges in order to fix variables in the travelling salesman problem. Beasley [1] and Lucena [7] did the same for the Steiner problem in graphs. All those references compute MST reduced costs by using edge labelling procedures. In this paper, we present a new algorithm for the problem. The algorithm is based on the construction of a binary tree by sequential deletion of the branches in the MST in a descending order of their costs. The binary tree thus constructed is composed of leaf nodes representing the actual vertices in the graph and intermediate nodes representing the branches of the MST. The problem then naturally reduces to that of locating the nearest common ancestors of the leaf nodes in the binary tree.

2 Terminology

We first review some well-known terminology in graph theory. Let $G = (V, E)$ be a connected undirected graph with a vertex set V and an edge set E. Each edge e is given by an unordered pair of vertices (i, j) and has an associated cost denoted by $c(e)$ or $c(i, j)$. A subgraph of G in which all vertices are connected and there are no cycles is called a spanning tree of G. Clearly, any spanning tree of G has n vertices and $(n-1)$ edges. The diameter of a spanning tree, D, is the maximum cardinality of all paths in the tree, where cardinality is measured in terms of the number of edges on a path. A minimum spanning tree T of G is a spanning tree of minimum total edge cost. Given an arbitrary vertex of T, termed the root, each vertex $i \in V$ has a unique father given by the first vertex on the path in T from i to the root. The only exception is the root vertex itself which has no father. An edge of G in T is called a branch, and an edge of G not in T is called a chord. For two disjoint subsets V_1 and V_2 of V the cutset of edges in G separating G_1 and G_2 is given by $E(V_1, V_2) = \{(i, j) \in E : i \in V_1, j \in V_2\}$. The fundamental cutset of a branch (p, q) is the cutset of the two connected subgraphs of G that remain when (p, q) is removed from T. The fundamental path of a chord $(i, j) \in E$ is the chain of edges that connect i and j in T. A critical branch on the fundamental path of a chord is one with maximum edge cost. Let (p, q) be a critical branch on the fundamental path of a chord (i, j) , the reduced cost of (i, j) is given by: $\bar{c}(i, j) = c(i, j) - c(p, q)$.

The MST reduced costs problem (MSTRCP) is to compute the reduced costs of all chords in a graph by locating a critical branch for each chord. The number of chords in a graph, m_c , depends on its density. A complete graph has a maximum number of edges and, hence, chords. A graph is called complete if, for all pair of vertices $i, j \in V$ there exists one edge of the form (i, j). A complete graph with n vertices is denoted by K_n and has $M = \frac{n(n-1)}{2}$ edges. In general, a graph with m edges has density given by $\rho = \frac{m}{M} * 100\%$ and number of chords given by: $m_c = m - (n-1)$.

We now introduce some basic definitions for binary trees (see eg, Tarjan [9]). We distinguish the vertices of a graph from the nodes of the binary tree. A binary tree (BT) is a tree where each node has a maximum degree of three: at most one father and two sons. The unique node which has no father is called the root of BT. All nodes with no sons are called leaf nodes: all other nodes are called intermediate nodes. For our application, each intermediate node of BT has exactly two sons: a left son and a right son. The ancestors of a node are those that lie on the unique path between the given node and the root of the tree. The level of a node is given by

the number of its ancestors upto and including the root (ie, the cardinality of the path). Clearly, the root node has a level of zero. The nearest common ancestor of two given nodes is the unique node with maximum level which is an ancestor of both nodes. The height of BT , H, is given by the maximum level over all its leaf nodes (ie, the level of its deepest node). The average height of BT is given by the mean level over all its leaf nodes. A binary tree is full if every level that contains any node contains the maximum number of nodes that can occur on that level. The number of nodes in a full binary tree must be in the form $(2^h - 1)$ for some integer h. Since this condition is quite restrictive, the requirement for a full binary tree is relaxed to specify a minimum height binary tree. A binary tree has minimum height if there is no other binary tree containing the same number of nodes whose height is less. The minimum height of a binary tree with N nodes can be easily derived as: $H_{min} = \lceil log_2(N+1) \rceil - 1$.

3 The Algorithm

We first note a simple observation: any connected subtree of T is also a minimum spanning tree on the subgraph of G implied by the subtree. We will use this observation iteratively for developing an algorithm for the solution of the MSTRCP by performing a sequence of branch cuts on T in a descending order of their costs. Consider an ordered set of branches of T, $\{e_1, e_2, \cdots, e_{n-1}\}$, such that $c(e_k) \geq c(e_{k+1})$ for all $k = 1, 2, ..., n-2$. Let $G_{10}, G_{11} \subset G$ be, respectively the two subgraphs with MSTs $T_{10}, T_{11} \subset T$ that result from the exclusion of a maximum cost branch $e_1 = (p_1, q_1)$ from T. G_{10} has vertex set V_{10} while G_{11} has vertex set V_{11}. Let $E(V_{10}, V_{11})$ be the fundamental cutset associated with branch e_1. It is immediate that e_1 is the critical branch for all chords (i_1, j_1) in the cutset. The reduced cost of a chord (i_1, j_1) in the cutset is, therefore, given by: $\bar{c}(i_1, j_1) = c(i_1, j_1) - c(e_1)$. We now perform a cut by removing branch e_1 from T. Note that the first subscript in $G_{10}, G_{11}, T_{10}, T_{11}$ and V_{10}, V_{11} is the number of the cut in the order, while the second subscript reflects the binary division of the tree (0 for left son, 1 for right son). Since T_{10}, T_{11} are both minimum spanning trees on their respective subgraphs, we can use this idea again by applying the second branch cut in the descending order. Let $e_2 \subset T_{10}$, say. By removing branch e_2 from T_{10}, the two connected subgraphs $G_{20}, G_{21} \subset G_{10} (\subset G)$ are now separated by the fundamental cutset $E(V_{20}, V_{21})$. Since e_2 is clearly a highest cost branch in the minimum spanning tree T_{10}, then e_2 is the critical branch for all chords (i_2, j_2) in the cutset. Hence, their reduced costs can be readily computed. This procedure can then be repeated by removing the next branch in the ordered sequence, until no more branches remain in T. The procedure can be formalized using the following Lemma.

Lemma: Consider a MST T of a graph G, and let $\{e_1, e_2, \cdots, e_{n-1}\}$ be an ordered set of branches of T, such that $c(e_k) \geq c(e_{k+1})$ for all $k = 1, 2, ..., n-2$. Perform an ordered sequence of k branch cuts ($k = 1, 2, ..., n-1$) on T. If e_k is the $k-th$ branch to be removed from T, let C_k be the set of chords in the fundamental cutset separating the two most recent connected subgraphs of G. The reduced cost of a chord $(i_k, j_k) \in C_k$ for any $k = 1, 2, ..., n-1$ is given by: $\overline{c}(i_k, j_k) = c(i_k, j_k) - c(e_k)$.

Proof: The proof is immediate from the above observations.

The algorithm proposed for solving the MSTRCP is composed of following three main procedures.

Input: A connected undirected graph $G = (V, E)$ and costs $c(i, j), \forall (i, j) \in E$.

- I: MST Construction.
- II: Binary Tree Construction.
- III: Nearest Common Ancestor Location.

Output: Reduced costs of all chords in G.

4 Summary of Results

A number of implementations for constructing a minimum spanning tree of a graph have been proposed in the literature (eg, Cheriton and Tarjan [3], Haymond et al [5]). All these algorithms are essentially based on the greedy algorithm proposed by Kruskal [6], and later by Dijkstra [4] and Prim [8]. We adopted Prim's algorithm for our application since the father of each vertex in the minimum spanning tree is immediately established. The algorithmic complexity of this procedure is $O(n^2)$. In [5], the computational complexity can be reduced to $O(n + m)$ using the forward star structure for storing low density graphs.

Starting with the minimum spanning tree T, the binary tree is constructed by a sequential deletion of the branches in T in their descending order of their costs, given by the output of Procedure I. The new nodes inserted for the construction of BT will be referred to as artificial nodes (intermediate nodes), and the vertices of the minimum spanning tree will be referred to simply as nodes (finally all becoming leaf nodes). For every branch $e_k = (p_k, q_k)$ in the sequence $k = 1, ..., n-1$, delete e_k from T, and reconnect the two remaining subtrees via a new artificial node $(n + k)$ and two new artificial edges linking the new node to p_k and q_k. The subtree rooted at $(n + k)$ is then inserted immediately below the intermediate node which is the nearest

artificial ancestor to p_k and q_k. This simply involves replacing an artificial edge with another. After all $(n-1)$ branches are removed, resulting in the addition of $(n-1)$ intermediate nodes and $2*(n-1)$ edges, the binary tree is fully constructed. The number of nodes of BT is, therefore, $N = (2n-1)$, which is composed of n leaf nodes which are the vertices T, and $(n-1)$ intermediate (artificial) nodes representing the branches of T.

Once the binary tree is constructed, and the level of all intermeditae nodes is recorded, the location of the nearest common ancestors of any pair of leaf nodes becomes trivial, and hence the reduced costs can be readily computed.

The algorithm proposed in this paper was computationaly tested with that presented in Carpaneto et al [1]. Both algorithms were coded in Fortran and run on a Sun workstation. Graphs of three different sizes were considered (n = 100, 150 and 220 vertices). The edge costs of the complete graph for each size were selected randomly in the integer range [1-100]. For each size, subgraphs of the complete graph of varying densities were generated. This was achieved by deleting the edges of the complete graph which fall below a given edge cost, c_{max}, say. Since the range of edge costs for the complete graph are randomly selected from 1 to 100, the density r of the subgraph with edge costs between 1 to c_{max} therefore approximates to c_{max}%. For each $[n, r]$ combination, average results were achieved over ten runs, each with a random selection of edge costs.

We have noted that, for a given graph size, the algorithm by Carpaneto et al. [2] has a relatively costant computational time over the graph density range. Our algorithm has a monotonically decreasing computational time as the density of the graph falls. This is expected since the computational complexity of the algorithm is proportional to the number of chords in the graph. While the algorithm by Carpaneto et al. solves the RCP in considerabely smaller computational time than the proposed algorithm for high density graphs, the opposite effect is observed for low density graphs. In fact, empirical results indicate that the changeover point is for graphs of density below 25%. In the case of complete graphs, our algorithm can be approximately four times slower, while for $5-10$% density graphs, our algorithm can be three to four times faster. Clearly, the advantages of the proposed algorithm are achieved for large low-density graphs.

References

[1] J.E Beasley. An SST-based algorithm for the Steiner problem in graphs. *Networks*. 19:1–16. 1989.

[2] Carpaneto, G., Fichetti, M. and Toth. P. New Lower Bounds for the Symmetric Travelling Salesman Problem *Mathematical Programming* 5:233–254,1989.

[3] Cheriton, D. and Tarjan, R.E. Finding Minimum Spanning Trees *SIAM Journal of Computing* 5,1976.

[4] Dijkstra, E.W. A note on two problems in connection with graphs *Numerical Mathematics* 1,5:269–271,1959.

[5] Haymond, R.E., Jarvis, J.P. and Shier, D.F. Algorithm 613: Minimum Spanning Trees for Moderate Integer Weights *ACM Transactions on Mathematical Software*, 10:108–111, 1984.

[6] Kruskal, J. B. On the shortest spanning subtree of a graph and the Traveling Salesman Problem *Proceedings of the American Mathematical Society*, 2:48–50,1956.

[7] A. Lucena. Steiner problem in graphs: lagrangean relaxation and cutting-planes. *COAL Bulletin*. Mathematical Programming Society, 21:2–8, 1992.

[8] Prim, R.C. Shortest Connection Networks and some Generalisations *Bell Systems Technology Journal*. 36:1389–1401,1957.

[9] Tarjan, R.E. Data Structures and Network Algorithms *Regional Conference Series in Applied Mathematics*.

[10] Volgenant, T. and Jonker, R. The Symmetric Travelling Salesman Problem and Edge Exchanges in Minimal 1-trees *European Journal of Operations Research*, 12:394-403,1983.

A LOSSY DATA COMPRESSION BASED ON AN APPROXIMATE PATTERN MATCHING*

Tomasz Łuczak†
Mathematical Institute
Polish Academy of Science
60-769 Poznań
Poland
tomasz@math.amu.edu.pl

Wojciech Szpankowski‡
Department of Computer Science
Purdue University
W. Lafayette, IN 47907
U.S.A.
spa@cs.purdue.edu

Abstract

A practical suboptimal (variable source coding) algorithm for lossy data compression is presented. This scheme is based on an approximate string matching, and it extends lossless Lempel-Ziv data compression scheme. Among others we consider the typical length of an approximately repeated pattern (we allow $D\%$ of mismatches) within the first n positions of a stationary mixing sequence. We prove that there exists a constant $r_0(D)$ such that the length of such an approximately repeated pattern converges in probability to $1/r_0(D)\log n$ (pr.) but it *almost surely* oscillates between $1/r_{-\infty}(D)\log n$ and $2/r_1(D)\log n$, where $r_{-\infty}(D) > r_0(D) > r_1(D)/2$ are some constants. More importantly, we show that the compression ratio of a data compression scheme based on such an approximate pattern matching is asymptotically equal to $r_0(D)$, and it is asymptotically optimal for $D \to 0$. In addition, we establish the asymptotic behavior of the so called *approximate waiting time* N_ℓ which is defined as the time until a pattern of length ℓ repeats approximately for the first time, and we prove that $\log N_\ell/\ell \to r_0(D)$ (a.s.) as $\ell \to \infty$. This settles the problem investigated recently by Steinberg and Gutman.

Index Terms: Lossy data compression, approximate pattern matching, Hamming distance, rate distortion, generalized Lempel-Ziv scheme, generalized Rényi entropy, mixing probabilistic model.

---

*This research started off while both authors were visiting INRIA, Rocquencourt, France, and then continued during the first author visit at Purdue University and the second author visit in Poland. The authors wish to thank INRIA (project ALGO) and NSF Grant NCR-9206315 for a generous support.

†On leave from Department of Discrete Mathematics, Adam Mickiewicz University, Poznań, Poland.

‡Additional support was provided by NSF Grants NCR-9206315 and CCR-9201078 and INT-8912631, and in part by NATO Collaborative Grant 0057/89

EXTENDED ABSTRACT

We plan to adopt approximate pattern matching algorithms to lossy data compression. An approximate pattern matching searches for an *approximate* occurrence of a given pattern in a text string, where the "approximation" is measured by some distance (i.e., distortion) between the pattern and the text strings (e.g., Hamming distance, edit distance, etc.). We should point out that exact pattern matching was recently successfully applied to numerous problems in data compression (cf. [5, 8, 10, 11, 12, 13, 14]). For example, Wyner and Ziv [14] introduced "stringology" into data compression, and then suffix tree was used in [12] to solve an open problem posed by Wyner and Ziv [14] (cf. see also [10, 13]). Furthermore, digital search trees (and analytical analysis of algorithms on words) were used in [5, 8] to obtain second-order properties of the lossless Lempel-Ziv parsing scheme originally studied only for *symmetric* memoryless source by Aldous and Shields [1].

We study a stationary and ergodic sequence $\{X_k\}_{k=-\infty}^{\infty}$ taking values in a finite alphabet Σ. For simplicity of presentation, we consider only binary alphabet $\Sigma = \{0, 1\}$. Let X_m^n to denote $X_m X_{m+1} \dots X_n$ and let $x_m^n = x_m \dots x_n$ stand for a string which is a realization of the stochastic process X_m^n. For a lossy transmission, one needs to introduce a measure of *fidelity*, and we restrict our discussion to Hamming distance (however, other fidelity criteria can be easily accommodated into our main results) defined as $d_n(x_1^n, \tilde{x}_1^n) = (1/n)\sum_{i=1}^n d_1(x_i, \tilde{x}_i)$ where $d_1(x, \tilde{x}) = 0$ for $x = \tilde{x}$ and 1 otherwise ($x, \tilde{x} \in \Sigma$). We assume that the maximum allowed distortion is D, and by $R(D)$ we denote the rate-distortion (cf. [6, 7, 9]).

Kieffer [6, 7], and Ornstein and Shields [9] proved that the compression factor in a lossy data compression is asymptotically equal to rate-distortion $R(D)$ (a.s.), and this cannot be improved. Note, however, that to construct an optimal data compression one needs to "guess" the optimal (i.e., minimum) cover of the set Σ^n by D-balls. (This is actually identical of guessing a probability measure on the receiver side on Σ^n that minimizes the mutual information [6, 11] which is equally difficult task!).

We propose a practical suboptimal lossy data compression scheme that extends the Lempel-Ziv scheme and that achieves rate

$$r_0(D) = \lim_{k\to\infty} \frac{-E \log_2 P(B_D(X_1^k))}{k},$$

where $B_D(x_1^k) = \{\tilde{x}_1^k : d(\tilde{x}_1^k, x_1^k) \le D\}$ is a D-ball with the center at X_1^k. It can be shown that this scheme is asymptotically optimal for $D \to 0$, that is, $\lim_{D\to 0} r_0(D) = h$, and usually $r_0(D)$ does not differ significantly from $R(D)$ for $D > 0$. Our scheme reduces to the following approximate pattern matching problem: Let the "training sequence" or "database sequence" x_1^n be given. Find the longest L_n such that there exists $1 \le i_0 \le n$ in the database satisfying $d(x_{i_0}^{i_0-1+L_n}, x_{n+1}^{n+L_n}) \le D$. This naturally extends Wyner and Ziv [14] idea to lossy situation (cf. also [11]).

satisfying $d(x_{i_0}^{i_0-1+L_n}, x_{n+1}^{n+L_n}) \leq D$. This naturally extends Wyner and Ziv [14] idea to lossy situation (cf. also [11]).

For $D = 0$ (i.e., **lossless case**) Wyner and Ziv [14] proposed the following data compression scheme based on L_n: The encoder sends the position i_0 in the database, the length L_n and possible one more symbol, namely x_{n+L_n+1}. Using this information the decoder reconstructs the original message, and both the encoder and the decoder enlarge the database. Based on a probabilistic analysis the authors of [14] (cf. also [10, 12]) concluded that with high probability the compression ratio of such an algorithm is equal to the entropy, thus it is asymptotically optimal. In this case, L_n can be found in $O(n)$ steps.

The idea of Wyner and Ziv is extended in this paper to a **lossy compression**. We shall propose an algorithm that finds L_n (i.e., approximately repeated pattern) in $O(n^2)$ steps in the worst case but on average in $O(n \cdot poly(\log n))$ steps. We also suggest a lossy data compression scheme based on such a approximate pattern matching. Preliminary studies (cf. [4]) indicate that such a data compression works particularly well for image compression (e.g., the compression ratio is often at least as good as in JPEG).

We further generalize our problem and we search for longest $L_n^{(b)}$ such that there exist *at most* b substrings in the database within distance D, that is, for some $1 \leq i_1 < \ldots < i_b \leq n$ we have $d(x_{i_1}^{i_1-1+L_n^{(b)}}, x_{n+1}^{n+L_n^{(b)}}) \leq D, \ldots, d(x_{i_b}^{i_b-1+L_n^{(b)}}, x_{n+1}^{n+L_n^{(b)}}) \leq D$, where b is a parameter (cf. [13] for lossless equivalent of this scheme and its implementation through the so called b-suffix trees). Clearly, $b = 1$ corresponds to the original problem.

Actually, the real engine behind this study (and its algorithmic issues) is a probabilistic analysis of an approximate pattern matching problem (i.e., lossy data compression), which we discuss next. Our probabilistic results are confined to the *stationary mixing model* in which two random events defined on two σ-algebra separated by g symbols behave like independent events as $g \to \infty$. We introduce the *generalized Rényi entropies* denoted as $r_b(D)$ which we prove to exist in our mixing model, where $-\infty \leq b \leq \infty$ is a parameter. In such a mixing model, we show that $L_n / \log n \to 1/r_0(D)$ in probability (pr.) where $r_0(D)$ represents the rate distortion, and in general $r_0(D) \geq R(D)$, except for the symmetric (memoryless) Bernoulli model in which $r_0(D) = R(D)$. Also, $\lim_{D \to 0} r_0(D) = \lim_{D \to 0} R(D) = h$. Surprisingly enough, $L_n / \log n$ does not converge almost surely (a.s.) but rather oscillates between two *different* constants, namely $1/r_{-\infty}(D) < 2/r_1(D)$. This kind of behavior was already observed in the lossless case (cf. [12, 13]). Finally, for memoryless source (i.e., Bernoulli model) we compute explicitly the entropies $r_b(D)$. We should point out that even in this case there are no simple formulæ for $r_b(D)$.

It turns out that the oscillation of L_n is related to the probabilistic behavior of two other interesting parameters that we call *shortest path* s_n and *height* H_n due to an analogy between these parameters and similar ones studied in [12, 13] for the lossless case. Roughly speaking, s_n is the largest K such that *all* strings of lengths K occur approximately somewhere in

$1/r_{-\infty}(D)$ (a.s.) and $H_n/\log n \to 2/r_1(D)$ (a.s.). Observe that $s_n \leq L_n \leq H_n$.

In a related paper Steinberg and Gutman [11] analyzed the so called waiting time N_ℓ which is define as length of the shortest string that contains approximately a string of length ℓ at the beginning and at the end (or equivalently string of length ℓ reoccurs approximately for the first time after N_ℓ symbols). The authors of [11] proved that for a stationary ergodic sequence $\limsup_{\ell\to\infty} \log N_\ell/\ell \leq R(D/2)$ (pr.). As a corollary of one of our results we show that in the mixing model $\lim_{\ell\to\infty} \log N_\ell/\ell = r_0(D)$ (a.s.), which ultimately settles the problem of [11].

There is a substantial literature on probabilistic analysis of pattern matching problems (cf. [1, 2, 3, 5, 11, 12, 13] but with an expection of [2, 3, 11] only lossless case (i.e., exact pattern matching) is discussed. The two papers [2, 3] on the approximate pattern matching explore only the height H_n which is not of the prime interest to data compression. Thus, to the best of our knowledge our results are novel not only in the context of data compressions.

References

[1] D. Aldous and P. Shields, A Diffusion Limit for a Class of Random-Growing Binary Trees, *Probab. Th. Rel. Fields*, 79, 509-542 (1988).

[2] R. Arratia and M. Waterman, The Erdős-Rényi Strong Law for Pattern Matching with Given Proportion of Mismatches, *Annals of Probability*, 17, 1152-1169 (1989).

[3] R. Arratia, L. Gordon, and M. Waterman, The Erdős-Rényi Law in Distribution for Coin Tossing and Sequence Matching, *Annals of Statistics*, 18, 539-570 (1990)

[4] M. Atallah, Y. Genin, and W. Szpankowski, Experimental Studies of Image Compressions Based on an Approximate Pattern Matching, in preparation.

[5] P. Jacquet and W. Szpankowski Asymptotic Behavior of the Lempel-Ziv Parsing Scheme and Digital Search Trees, *Theoretical Computer Science*, to appear (1995).

[6] J.C. Kieffer, Strong Converses in Source Coding Relative to a Fidelity Criterion, *IEEE Trans. Information Theory*, 37, 257-262 (1991).

[7] J. C. Kieffer, Sample Converses in Source Coding Theory, *IEEE Trans. Information Theory*, 37, 263-268 (1991).

[8] G. Louchard and W. Szpankowski, Average Profile and Limiting Distribution for a Phrase Size in the Lempel-Ziv Parsing Algorithm, *IEEE Trans. Information Theory*, to appear (1995).

[9] D. Ornstein and P. Shields, Universal Almost Sure Data Compression, *Annals of Probability*, 18, 441-452 (1990).

[10] D. Ornstein and B. Weiss, Entropy and Data Compression Schemes, *IEEE Information Theory*, 39, 78-83 (1993).

[11] Y. Steinberg and M. Gutman, An Algorithm for Source Coding Subject to a Fidelity Criterion, Based on String Matching, *IEEE Trans. Information Theory*, 39, 877-886 (1993).

[12] W. Szpankowski, Asymptotic Properties of Data Compression and Suffix Trees, *IEEE Trans. Information Theory*, 39, 1647-1659 (1993).

[13] W. Szpankowski. A Generalized Suffix Tree and Its (Un)Expected Asymptotic Behaviors. *SIAM J. Computing*, 22, 1176-1198 (1993).

[14] A. Wyner and J. Ziv. Some Asymptotic Properties of the Entropy of a Stationary Ergodic Data Source with Applications to Data Compression. *IEEE Trans. Information Theory*, 35, 1250-1258 (1989).

Sources, Sinks, Even and Odd Pairs in Comparability Graphs

Jayme L. Szwarcfiter[1]
Célia P. Mello[2]
Celina M. H. de Figueiredo[3]

KEYWORDS: algorithms, comparability graphs, even pairs, odd pairs, sources, sinks.

ABSTRACT

A characterization of vertices of a comparability graph G which are sources of some transitive orientation of G has been presented by Gimbel and independently, by Olariu. We extend this characterization to subsets of vertices which are sources, and to pairs of subsets corresponding to sources and sinks, respectively. The results obtained relate sources and sinks to even and odd pairs. We describe conditions for an even pair to form a pair of sources, for an odd pair to be a pair of source-sink vertices, and conversely. Also, we formulate characterizations for two vertices of G to be an even or odd pair of it. The latter corresponds to a simple and efficient algorithm to decide if two given vertices form an even or odd pair of a comparability graph.

[1]Universidade Federal do Rio de Janeiro, Núcleo de Computação Eletrônica and Instituto de Matemática, Caixa Postal 2324, 20001 Rio de Janeiro, RJ, Brasil. Supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, grant 200.076-83. This work has been done while this author was visiting the L.I.R.M.M., Université de Montpellier II, Montpellier, France.

[2]Universidade Estadual de Campinas, Departamento de Ciência da Computação, Caixa Postal 6065, 13081 Campinas, SP, Brasil.

[3]Universidade Federal do Rio de Janeiro, Instituto de Matemática, Caixa Postal 68530, 21944 Rio de Janeiro, RJ, Brasil. Partially supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, grant 301.160/91.0.

1 Introduction

We describe characterizations for a subset of vertices of a comparability graph G to be formed by sources of some transitive orientation $\vec{G}$ of G. In addition, for a pair of subsets to constitute sources and sinks of $\vec{G}$, respectively. The results relate sources and sinks to even and odd pairs. We describe the conditions for an even pair to form a pair of sources, and for an odd pair to be a source-sink pair of vertices, and conversely. In addition, formulate simple characterizations for a pair of vertices to form an even or odd pair of the graph. It leads to efficient algorithms for solving the **parity path problems** in comparability graphs. That is, decide whether or not two given vertices form an even or an odd pair, respectively.. The complexity of the algorithms is the square of the number of vertices of G. However, it reduces to linear time, when a transitive orientation of the graph is given.

Olariu [13] and Gimbel [6] have formulated characterizations for a vertex of a comparability graph to be a source in some transitive orientation. In particular, [13] and [6] provided characterizations by forbidden induced subgraphs. On the other hand, [10] and [15] described properties relating sources and sinks to the maximal cliques of the graph. The problem of finding a transitive orientation of a graph, if existing, with prescribed sources and sinks can be solved using a method described by Möhring [12]. It generalizes Golumbic's transitive orientation algorithm [8]. Given any partial orientation of an undirected graph, [12] extends it to a transitive orientation, whenever possible. The complexity is that of comparability graph recognition.

Even pairs have been first considered by Meyniel [11], in the context of perfect graphs. They have motivated the definition of some special classes of perfect graphs, as parity, quasi-parity and perfectly contractile graphs. The parity path problems are Co-NP-complete, in general (Bienstock [4]). However, special classes of graphs admit polynomial time algorithms. They include chordal graphs [1], circular arc graphs [3], planar perfect graphs having the two given vertices in a same face [9] and perfectly orientable graphs [2], a superclass of the perfectly orderable ones. The latter contains comparability graphs. However, the presently proposed algorithms for comparability graphs are substancially simpler and their complexity lower than that for perfectly orientable graphs.

We employ the following notation and definitions.

G denotes a simple non-trivial connected undirected graph, with vertex set $V(G)$ and edge set $E(G)$. Write $n = |V(G)|$ and $m = |E(G)|$. A **chord** of a path P is an edge of G incident to two non-consecutive vertices of P. An **induced path** contains no chords. A path is **even** or **odd** according to the parity of its number of edges, respectively. The vertices v, w form **an even pair** when G contains no odd induced path between them. They constitute **an odd pair** when $(v, w) \notin E(G)$ and there is no even induced path between v and w.

$\vec{G}$ represents an acyclic orientation of G. For $v \in V(G)$, write $N^-_{\vec{G}}(v) = \{w | (w, v) \in E(\vec{G})\}$ and $N^+_{\vec{G}}(v) = \{w | (v, w) \in E(\vec{G})\}$. If $N^-_{\vec{G}}(v) = \emptyset$ then v is **a source**, and if $N^+_{\vec{G}}(v) = \emptyset$ then v is **a sink**. **A source (sink) set** of $\vec{G}$ is a subset of vertices formed by sources (sinks). S, T are **source-sink sets** when S is a source set and T a sink set of $\vec{G}$. $\vec{G}$ **is transitive** when $(v, w), (w, z) \in E(\vec{G})$ implies $(v, z) \in E(\vec{G})$, for all $v, w, z \in V(G)$. G **is a comparability graph** when it admits a transitive orientation. In this case, $s \in V(G)$ is **a source** of G when it is a source of some transitive orientation of G. Similarly, $S \subset V(G)$ is **a source set** of G when it is a source set of some transitive orientation of it. If $|S| = 2$, let $S = \{v, w\}$. Then v, w is called **a source pair** of G. Finally, $S, T \subset V(G)$ are **source-sink sets** of G when they are so for some transitive orientation of the graph. When $|S| = |T| = 1$, denote $S = \{v\}$, $T = \{w\}$ and call v, w **a source-sink pair (of vertices)**.

A well known characterization of comparability graphs by forbidden induced subgraphs is that by

Gallai [5]. The list of its forbidden subgraphs is called the **Gallai's list**.

2 Characterizations

Not all vertices of a comparability graph are sources. For example, vertex z of figure 1 is not. The proposition below characterizes sources. G^v denotes the graph obtained from graph G by adding a new vertex v' adjacent solely to the fixed vertex $v \in V(G)$.

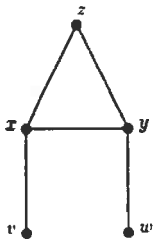

Figure 1

Theorem 1 [6,13]: Let G be a comparability graph and $v \in V(G)$. Then v is a source of G if and only if G^v is a comparability graph.

A set of sources is not necessarily a source set of a comparability graph. For example, $\{v, w\}$ is not a source set, although it is a set of sources of the graph of figure 1. The theorem below describes characterizations of source sets. For some $S \subset V(G)$, G^S denotes the graph constructed from G by including a new vertex v' and an edge (v, v'), for each $v \in S$.

Theorem 2: Let G be a connected comparability graph, $S \subset V(G)$. The following are equivalent.

(i) S is a source set.

(ii) G^v is a comparability graph and v, w an even pair, for all $v, w \in S$.

(iii) G^S is a comparability graph, $(v, w) \notin E(G)$ and v, w is not an odd pair, for all $v, w \in S$.

Proof (i) $\Longrightarrow$ (ii): Let S be a source set. By Theorem 1 it follows that G^v is a comparability graph, for each $v \in S$. To show the second condition of (ii), let $v, w \in S$. Suppose v, w is not an even pair of G. Then G contains an induced path $v = x_1, ..., x_{2k} = w$, $k \geq 1$, of odd length. Let $\vec{G}$ be a transitive orientation of G, having S as a source set. Then $(v, x_2) \in E(\vec{G})$ if and only if $(x_{2k-1}, w) \in E(\vec{G})$, contradicting $v, w \in S$. Hence v, w is an even pair.

(ii) $\Longrightarrow$ (iii): $S \subset V(G)$ is such that G^v is a comparability graph for each $v \in S$, and v, w is an even pair for each $v, w \in S$. We prove that G^S is a comparability graph. Suppose it is not. Then by [5], G^S contains an induced subgraph H isomorphic to one of the forbidden subgraphs of Gallai's list. Let $S' = V(G^S) - V(G)$. If H contains no vertex of S' then H is a subgraph of G, that is a comparability graph, a contradiction. If H contains exactly one vertex v' of S', $d_H(v') = 1$ and denote

by $v \in V(H)$ the vertex adjacent to v' in H. Clearly, H is a subgraph of G''. By hypothesis, G' is a comparability graph. So must be H, a contradiction. Hence H must contain at least two distinct vertices $v', w' \in S'$. Again $d_H(v') = d_H(w')=1$ and let v, w be respectively the vertices adjacent to v', w' in H. By hypothesis, v, w is an even pair of G. G^S and G have exactly the same induced paths between v and w. Hence v, w is an even pair of G^S, i.e., of H. Therefore H contains an induced subgraph formed by vertices v', w' of degree one, adjacent respectively to $v, w \in V(H)$, such that v, w is an even pair of it. It is a matter of routine checking Gallai's list to conclude that no such H exists. Hence G^S is a comparability graph. The second condition of (iii) is obvious.

(iii) $\Longrightarrow$ (i): G^S is a comparability graph and v, w is not an odd pair, for $v, w \in S$. We show that S is a source set. Let $\vec{G}^S$ be a transitive orientation of G^S. Each $v \in S$ is adjacent in G^S to at least one vertex of degree one. Hence v must be either a source or a sink. Since $(v, w) \notin E(G)$, G contains an even induced path between v and w. Then v is a source if and only if w is so. Hence, either all vertices of S are sources or all are sinks of $\vec{G}^S$. That is, S is a source set of G. □

The corollary below relates even pairs to source pairs.

Corollary 1: Let G be a comparability graph, $v, w \in V(G)$. If v, w is a source pair then it is an even pair. If v, w is an even pair then it is a source pair, except when v or w are not sources.

Proof: Equivalence (i) $\Leftrightarrow$ (ii) of Theorem 2, with $|S| = 2$. □

Two source sets are not always source-sink sets of a comparability graph. In figure 1, $\{v\}, \{y\}$ are source sets and not source-sink sets. Theorem 3 formulates characterizations of source-sink sets.

Theorem 3: Let G be a connected comparability graph, $|V(G)| > 1$ and $S, T \subset V(G)$. The following are equivalent:

(i) S, T are source-sink sets.

(ii) S, T are source sets, and for all $v \in S$ and $w \in T$
$(v, w) \in E(G) \Rightarrow G$ does not contain the complement of the graph of figure 2 as an induced subgraph.
$(v, w) \notin E(G) \Rightarrow v, w$ is an odd pair.

(iii) $G^{S \cup T}$ is a comparability graph, and for all $v, w \in S \cup T$
$v, w \in S$ or $v, w \in T \Rightarrow (v, w) \notin E(G)$ and v, w is not an odd pair.
$v \in S$ and $w \in T \Rightarrow v, w$ is not an even pair.

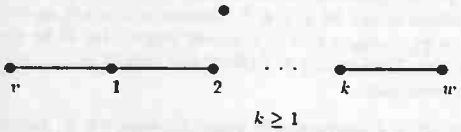

Figure 2

Proof (i) $\Longrightarrow$ (ii): S, T are source-sink sets, by hypothesis. Clearly, S and T are both source sets. To prove the second condition of (ii), let $v \in S$ and $w \in T$. The first case is $(v, w) \in E(G)$. Let

F be the graph of figure 2. Suppose that $\bar{F}$ is an induced subgraph of G. Since $v \in S$ and $w \in T$, $G^{\{v\}\cup\{w\}}$ must be a comparability graph. Let v', w' be the vertices of $V(G^{\{v\}\cup\{w\}}) - V(G)$, respectively. It follows that the complement of the subgraph induced in $G^{\{v\}\cup\{w\}}$ by the subset $V(F) \cup \{v', w'\}$ is precisely the graph of figure 3, whose complement is in Gallai's list. That is, $G^{\{v\}\cup\{w\}}$ is not a comparability graph, contradicting S, T to be a pair of source-sink sets. Hence $\bar{F}$ is not an induced subgraph of G. The second case is to consider $(v, w) \notin E(G)$. Suppose v, w is not an odd pair of G. Then G contains an induced path $v = x_1, \ldots x_{2k+1} = w$, $k \geq 1$, of even length. Let $\vec{G}$ be a transitive orientation of G, having S, T as sets of sources and sinks, respectively. Then $(v, x_2) \in E(\vec{G})$ if and only if $(w, x_{2k}) \in E(\vec{G})$, contradicting $v \in S$ and $w \in T$. Therefore v, w is an odd pair.

(ii) $\Longrightarrow$ (iii): First, we prove that $G^{S\cup T}$ is a comparability graph. Suppose it is not. Then by [5], $G^{S\cup T}$ contains an induced subgraph H isomorphic to one of Gallai's list. Let $S' = V(G^{S\cup T}) - V(G)$. If H contains no vertex of S' then G is not a comparability graph, a contradiction. For each $v' \in V(H) \cap S'$, $d_H(v') = 1$ and denote by v the vertex of $S \cup T$ adjacent to v' in H. If all such vertices v belong to S then by Theorem 2, $G^{S\cup T}$ is a comparability graph, a contradiction. Similarly, if all belong to T. Hence there exists a pair of vertices $v \in S$ and $w \in T$ such that $v', w' \in S'$, where v', w' are the degree one vertices of H adjacent to v and w, respectively. Therefore H has an induced subgraph containing two vertices v', w' of degree one. There are exactly two possible choices for H in Gallai's list, namely the graph of figure 4 and the complement of that of figure 3. In the first case, v and w correspond to vertices 2 and $2k + 1$ of the figure, respectively. Note that $(2, 2k + 1) \notin E(G)$. However, $2, 2k+1$ is not an odd pair of H, hence of G, contradicting the hypothesis. As for the second choice of H, observe that $H - \{v', w'\}$ is precisely the graph F of figure 2. In this case, $(v, w) \in E(G)$ which implies that $\bar{F}$ can not occur in G, a contradiction. Hence $G^{S\cup T}$ must be a comparability graph. The proof of the second condition of (iii) is immediate. If $v, w \in S$ or $v, w \in T$ then by Theorem 2 it follows (v, w) is not an odd pair. Let $v \in S$ and $w \in T$. If $(v, w) \in E(G)$ clearly v, w is not an odd pair. Otherwise it follows from the hypothesis.

(iii) $\Longrightarrow$ (i): Since $G^{S\cup T}$ is a comparability graph it admits a transitive orientation $\vec{G}^{S\cup T}$. Each vertex of $S \cup T$ must be either a source or a sink of $\vec{G}^{S\cup T}$. Without loss of generality, suppose some vertex $v \in S$ is a source. Let $w \in S$. Then $(v, w) \notin E(G)$ and v, w is not an odd pair. That is, w is a source of $\vec{G}^{S\cup T}$. Let $w \in T$. Then v, w is not an even pair and w is a sink of $G^{S\cup T}$. That is, S, T are source-sink sets of of G. □

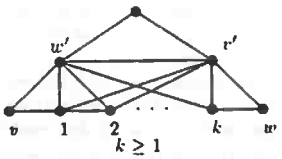

Figure 3

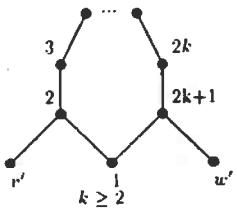

Figure 4

A different characterization of source-sink sets has been formulated by Gimbel [7].

A relation between odd pairs and source-sink pairs of vertices is given by the following corollary.

Corollary 2: Let G be a comparability graph, $v, w \in V(G)$. If v, w is a source-sink pair then it is an odd pair, except when $(v, w) \in E(G)$. If v, w is an odd pair then it is a source-sink pair, except

when v or w are not sources.

Proof: Equivalence (i) $\Leftrightarrow$ (ii) of Theorem 3, with $|S| = |T| = 1$. □

Let G be a comparability graph. A source set $S \subset V(G)$ is **exact** when G admits a transitive orientation $\vec{G}$ having the vertices of S as the only sources of $\vec{G}$. Similarly, we define **exact** source-sink sets S, T of G.

The observations below are clear.

Let S be a source set of G. Then S is exact if and only if S is a maximal independent set of G.

Let S, T be source-sink sets. Then S, T are exact if and only if both S and T are maximal independent sets of G.

By adding the above observations, theorems 2 and 3 can be reformulated so as to characterize exact source and source-sink sets, respectively.

Next, we describe characterizations of even and odd pairs of a comparability graph. The following definition is needed. Let G be a connected graph, $S \subset V(G)$ and $v, w \in V(G) - S$. S **separates** v, w in G when v and w lie in distinct connected components of $G - S$.

Theorem 4: Let G be a connected comparability graph, $\vec{G}$ any transitive orientation of it and $v, w \in V(G)$. Then v, w is an even pair if and only if

$$N^-_{\vec{G}}(v) \cup N^+_{\vec{G}}(w) \text{ and } N^-_{\vec{G}}(w) \cup N^+_{\vec{G}}(v)$$

both separate v, w in G.

Proof: $\Longrightarrow$: Let v, w be an even pair of G. Then all induced paths of G between v and w are even. Since G is a comparability graph, the orientations of the edges in any induced path alternate. That is, all induced paths between v and w are of the form $v = x_1, \ldots, x_{2k+1} = w$, $k \geq 1$, where $(v, x_2) \in E(G)$ is oriented from v to x_2 if and only if $(w, x_{2k}) \in E(G)$ is oriented from w to x_{2k}. Hence $N^-_{\vec{G}}(v) \cup N^+_{\vec{G}}(w)$ meets all induced paths between v and w. So does $N^-_{\vec{G}}(w) \cup N^+_{\vec{G}}(v)$. Every general path P between v and w contains an induced path between these two vertices formed by a subset of vertices of P. Hence $N^-_{\vec{G}}(v) \cup N^+_{\vec{G}}(w)$ and $N^-_{\vec{G}}(w) \cup N^+_{\vec{G}}(v)$ both separate v, w in G.

$\Longleftarrow$: By hypothesis $N^-_{\vec{G}}(v) \cup N^+_{\vec{G}}(w)$ and $N^-_{\vec{G}}(w) \cup N^+_{\vec{G}}(v)$ separate v, w in G. Because G is a comparability graph, all induced paths of G are alternating in $\vec{G}$. Partition the set of induced paths of G between v and w into four disjoint subsets E_1, E_2, O_1, O_2. $E_1 \cup E_2$ contains the even induced paths $v = x_1, \ldots, x_{2k+1} = w$, $k \geq 1$. E_1 is formed by the paths where $(v, x_2), (w, x_{2k}) \in E(\vec{G})$ while E_2 contains those satisfying $(x_2, v), (x_{2k}, w) \in E(\vec{G})$. $O_1 \cup O_2$ contains the odd induced paths $v = x_1, \ldots, x_{2k} = w$. The paths belonging to O_1 satisfy $(v, x_2), (x_{2k-1}, w) \in E(\vec{G})$, while those of O_2 satisfy $(x_2, v), (w, x_{2k-1}) \in E(\vec{G})$. $N^-_{\vec{G}}(v) \cup N^+_{\vec{G}}(w)$ meets all paths of $E_1 \cup E_2 \cup O_2$, but none of O_1. Because $N^-_{\vec{G}}(v) \cup N^+_{\vec{G}}(w)$ separates v, w in G it follows $O_1 = \emptyset$. Similarly, $N^-_{\vec{G}}(w) \cup N^+_{\vec{G}}(v)$ separating v, w implies $O_2 = \emptyset$. Therefore v, w is an even pair. □

Theorem 5: Let G be a connected comparability graph, $\vec{G}$ any transitive orientation of it and $v, w \in V(G)$. Then v, w is an odd pair if and only if

$$N^-_{\vec{G}}(v) \cup N^-_{\vec{G}}(w) \text{ and } N^+_{\vec{G}}(v) \cup N^+_{\vec{G}}(w)$$

both separate v, w in G.

Proof: Similar to that of Theorem 4. □

3 Conclusions

We have described characterizations for sources, sinks, even and odd pairs in comparability graphs. They emphasize the relationship among these concepts. Theorems 2 and 3 can be used to formulate an algorithm for finding transitive orientations with prescribed sources and sinks. However, the problem of finding a transitive orientation having a source set of maximum size remains to be solved.

The parity path problems for comparability graphs can be solved applying theorems 4 and 5. Given a comparability graph G and $v, w \in V(G)$, find a transitive orientation $\vec{G}$ of it. Then v, w form an even pair of G precisely when they lie in distinct connected components of $G - [N^-_{\vec{G}}(v) \cup N^+_{\vec{G}}(w)]$ and $G - [N^-_{\vec{G}}(w) \cup N^+_{\vec{G}}(v)]$. The odd pair problem is similar, except that the latter are $G - [N^-_{\vec{G}}(v) \cup N^-_{\vec{G}}(w)]$ and $G - [N^+_{\vec{G}}(v) \cup N^+_{\vec{G}}(w)]$. Finding a transitive orientation requires $O(n^2)$ time [14]. The remaining operations can be done in $O(n + m)$ time.

References

[1] S. R. Arikati and U.N. Peled (1993). A linear algorithm for the group path problem on chordal graphs. *Discrete Applied Maths.* 44, 185–190.

[2] S. R. Arikati and U.N. Peled (1992). *A Polynomial Algorithm for the Parity Problem on Perfectly Orientable Graphs* manuscript.

[3] S. R. Arikati, C. P. Rangan and G. K. Manacher (1991). Efficient reduction for path problems on circular-arc graphs. *BIT* 31, 182–193.

[4] D. Bienstock (1991). On the complexity of testing for odd holes and induced odd paths. *Discrete Maths.* 90, 85–92.

[5] T. Gallai (1967). Transitiv orienterbare graphen. *Acta Math. Acad. Hungar.* 18, 25–66.

[6] J. Gimbel (1992). Sources in posets and comparability graphs. *Order* 9, 361–365.

[7] J. Gimbel (1993). Private communication.

[8] M. C. Golumbic (1977). The complexity of comparability graph recognition and coloring. *Computing* 18, 199–208.

[9] W.-L. Hsu (1987). Recognizing planar perfect graphs. *J. of the ACM* 34, 255–288.

[10] C. P. Mello (1992) Sobre Grafos Clique-Completos, Ph.D. Thesis (in portuguese), COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ.

[11] H. Meyniel (1987). A new property of critical imperfect graphs and some consequences. *Europ. J. Combinatorics* 8, 313–316.

[12] R. H. Möhring (1985). Algorithmic aspects of comparability graphs and interval graphs. in I. Rival, ed., *Graphs and Orders*, 41–102, D. Reidl, Dordrecht.

[13] S. Olariu (1992) Sources in comparability graphs with aplications. *Discrete Maths.* 110, 289–292.

[14] J. Spinrad (1985) On comparability and permutation graphs. *SIAM J. Comput.* 14, 658–670.

[15] J. L. Szwarcfiter, C. P. Mello and C. M. H. de Figueiredo. On finding transitive orientations with prescribed sources and sinks, *Congressus Numerantium*, to appear.

RELATÓRIOS TÉCNICOS

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Instituto de Matemática e Estatística da USP

A listagem contendo os relatórios técnicos anteriores a 1992 poderá ser consultada ou solicitada à Secretaria do Departamento, pessoalmente, por carta ou e-mail(mac@ime.usp.br).

J.Z. Gonçalves, Arnaldo Mandel
COMMUTATIVITY THEOREMS FOR DIVISION RINGS AND DOMAINS
RT-MAC-9201, Janeiro 1992, 12 pp.

J. Sakarovitch
THE "LAST" DECISION PROBLEM FOR RATIONAL TRACE LANGUAGES
RT-MAC 9202, Abril 1992, 20 pp.

Valdemar W. Setzer, Fábio Henrique Carvalheiro
ALGOTRITMOS E SUA ANÁLISE (UMA INTRODUÇÃO DIDÁTICA)
RT-MAC 9203, Agosto 1992, 19 pp.

Claudio Santos Pinhanez
UM SIMULADOR DE SUBSUMPTION ARCHITECTURES
RT-MAC-9204, Outubro 1992, 18 pp.

Julio M. Stern
REGIONALIZAÇÃO DA MATRIZ PARA O ESTADO DE SÃO PAULO
RT-MAC-9205, Julho 1992, 14 pp.

Imre Simon
THE PRODUCT OF RATIONAL LANGUAGES
RT-MAC-9301, Maio 1993, 18 pp.

Flávio Soares C. da Silva
AUTOMATED REASONING WITH UNCERTAINTIES
RT-MAC-9302, Maio 1993, 25 pp.

Flávio Soares C. da Silva
ON PROOF-AND MODEL-BASED TECHNIQUES FOR REASONING WITH UNCERTAINTY
RT-MAC-9303, Maio 1993, 11 pp.

Carlos Humes Jr., Leônidas de O.Brandão, Manuel Pera Garcia
A MIXED DYNAMICS APPROACH FOR LINEAR CORRIDOR POLICIES
(A REVISITATION OF DYNAMIC SETUP SCHEDULING AND FLOW CONTROL IN MANUFACTURING SYSTEMS)
RT-MAC-9304, Junho 1993, 25 pp.

Ana Flora P.C.Humes e Carlos Humes Jr.
STABILITY OF CLEARING OPEN LOOP POLICIES IN MANUFACTURING SYSTEMS (Revised Version)
RT-MAC-9305, Julho 1993, 31 pp.

Maria Angela M.C. Gurgel e Yoshiko Wakabayashi
THE COMPLETE PRE-ORDER POLYTOPE: FACETS AND SEPARATION PROBLEM
RT-MAC-9306, Julho 1993, 29 pp.

Tito Homem de Mello e Carlos Humes Jr.
SOME STABILITY CONDITIONS FOR FLEXIBLE MANUFACTURING SYSTEMS WITH NO SET-UP TIMES
RT-MAC-9307, Julho de 1993, 26 pp.

Carlos Humes Jr. e Tito Homem de Mello
A NECESSARY AND SUFFICIENT CONDITION FOR THE EXISTENCE OF ANALYTIC CENTERS IN PATH FOLLOWING METHODS FOR LINEAR PROGRAMMING
RT-MAC-9308, Agosto de 1993

Flavio S. Corrêa da Silva
AN ALGEBRAIC VIEW OF COMBINATION RULES
RT-MAC-9401, Janeiro de 1994, 10 pp.

Flavio S. Corrêa da Silva e Junior Barrera
AUTOMATING THE GENERATION OF PROCEDURES TO ANALYSE BINARY IMAGES
RT-MAC-9402, Janeiro de 1994, 13 pp.

Junior Barrera, Gerald Jean Francis Banon e Roberto de Alencar Lotufo
A MATHEMATICAL MORPHOLOGY TOOLBOX FOR THE KHOROS SYSTEM
RT-MAC-9403, Janeiro de 1994, 28 pp.

Flavio S. Corrêa da Silva
ON THE RELATIONS BETWEEN INCIDENCE CALCULUS AND FAGIN-HALPERN STRUCTURES
RT-MAC-9404, abril de 1994, 11 pp.

Junior Barrera; Flávio Soares Corrêa da Silva e Gerald Jean Francis Banon
AUTOMATIC PROGRAMMING OF BINARY MORPHOLOGICAL MACHINES
RT-MAC-9405, abril de 1994, 15 pp.

Valdemar W. Setzer; Cristina G. Fernandes; Wania Gomes Pedrosa e Flavio Hirata
UM GERADOR DE ANALISADORES SINTÁTICOS PARA GRAFOS SINTÁTICOS SIMPLES
RT-MAC-9406, abril de 1994, 16 pp.

Siang W. Song
TOWARDS A SIMPLE CONSTRUCTION METHOD FOR HAMILTONIAN DECOMPOSITION OF THE HYPERCUBE
RT-MAC-9407, maio de 1994, 13 pp.

Julio M. Stern
MODELOS MATEMATICOS PARA FORMAÇÃO DE PORTFÓLIOS
RT-MAC-9408, maio de 1994, 50 pp.

Imre Simon
STRING MATCHING ALGORITHMS AND AUTOMATA
RT-MAC-9409, maio de 1994, 14 pp.

Valdemar W. Setzer e Andrea Zisman
*CONCURRENCY CONTROL FOR ACCESSING AND COMPACTING B-TREES**
RT-MAC-9410, junho de 1994, 21 pp.

Renata Wassermann e Flávio S. Corrêa da Silva
TOWARDS EFFICIENT MODELLING OF DISTRIBUTED KNOWLEDGE USING EQUATIONAL AND ORDER-SORTED LOGIC
RT-MAC-9411, junho de 1994, 15 pp.

Jair M. Abe, Flávio S. Corrêa da Silva e Marcio Rillo
PARACONSISTENT LOGICS IN ARTIFICIAL INTELLIGENCE AND ROBOTICS.
RT-MAC-9412, junho de 1994, 14 pp.

Flávio S. Corrêa da Silva, Daniela V. Carbogim
A SYSTEM FOR REASONING WITH FUZZY PREDICATES
RT-MAC-9413, junho de 1994, 22 pp.

Flávio S. Corrêa da Silva, Jair M. Abe, Marcio Rillo
MODELING PARACONSISTENT KNOWLEDGE IN DISTRIBUTED SYSTEMS
RT-MAC-9414, julho de 1994, 12 pp.

Nami Kobayashi
THE CLOSURE UNDER DIVISION AND A CHARACTERIZATION OF THE RECOGNIZABLE Z-SUBSETS
RT-MAC-9415, julho de 1994, 29pp.

Flávio K. Miyazawa e Yoshiko Wakabayashi
AN ALGORITHM FOR THE THREE-DIMENSIONAL PACKING PROBLEM WITH ASYMPTOTIC PERFORMANCE ANALYSIS
RT-MAC-9416, novembro de 1994, 30 pp.

Thomaz I. Seidman e Carlos Humes Jr.
SOME KANBAN-CONTROLLED MANUFACTURING SYSTEMS: A FIRST STABILITY ANALYSIS
RT-MAC-9501, janeiro de 1995, 19 pp.

C.Humes Jr. and A.F.P.C. Humes
STABILIZATION IN FMS BY QUASI- PERIODIC POLICIES
RT-MAC-9502, março de 1995, 31 pp.

Fabio Kon e Arnaldo Mandel
SODA: A LEASE-BASED CONSISTENT DISTRIBUTED FILE SYSTEM
RT-MAC-9503, março de 1995, 18 pp.

Junior Barrera, Nina Sumiko Tomita, Flávio Soares C. Silva, Routo Terada
AUTOMATIC PROGRAMMING OF BINARY MORPHOLOGICAL MACHINES BY PAC LEARNING
RT-MAC-9504, abril de 1995,16 pp.

Flávio S. Corrèa da Silva e Fabio Kon
CATEGORIAL GRAMMAR AND HARMONIC ANALYSIS
RT-MAC-9505, junho de 1995,17 pp.

Henrique Mongelli e Routo Terada
ALGORITMOS PARALELOS PARA SOLUÇÃO DE SISTEMAS LINEARES
RT-MAC-9506, junho de 1995, 158 pp.

Kunio Okuda
PARALELISAÇÃO DE LAÇOS UNIFORMES POR REDUCÃO DE DEPENDÊNCIA
RT-MAC-9507, julho de 1995, 27 pp.

Valdemar W. Setzer e Lowell Monke
COMPUTERS IN EDUCATION: WHY, WHEN, HOW
RT-MAC-9508, julho de 1995, 21 pp.

Flávio S. Corrèa da Silva
REASONING WITH LOCAL AND GLOBAL INCONSISTENCIES
RT-MAC-9509, julho de 1995, 16 pp.

Julio M. Stern
MODELOS MATEMÁTICOS PARA FORMAÇÃO DE PORTFÓLIOS
RT-MAC-9510, julho de 1995, 43 pp.

Fernando Iazzetta e Fabio Kon
A DETAILED DESCRIPTION OF MAXANNEALING
RT-MAC-9511, agosto de 1995, 22 pp.

Flávio Keidi Miyazawa e Yoshiko Wakabayashi
POLYNOMIAL APPROXIMATION ALGORITHMS FOR THE ORTHOGONAL Z-ORIENTED 3-D PACKING PROBLEM
RT-MAC-9512, agosto de 1995, pp.

Junior Barrera e Guillermo Pablo Salas
SET OPERATIONS ON COLLECTIONS OF CLOSED INTERVALS AND THEIR APPLICATIONS TO THE AUTOMATIC PROGRAMMINIG OF MORPHOLOGICAL MACHINES
RT-MAC-9513, agosto de 1995, 84 pp.

Marco Dimas Gubitoso e Jörg Cordsen
PERFORMANCE CONSIDERATIONS IN VOTE FOR PEACE
RT-MAC-9514, novembro de 1995, 18pp.

Carlos Eduardo Ferreira e Yoshiko Wakabayashi
ANAIS DA I OFICINA NACIONAL EM PROBLEMAS COMBINATÓRIOS: TEORIA, ALGORITMOS E APLICAÇÕES
RT-MAC-9515, novembro de 1995, 45 pp.