

Tese apresentada à Divisão de Pós-Graduação do Instituto Tecnológico de
Aeronáutica como parte dos requisitos para obtenção do título de Mestre em Ciência no
Curso de Engenharia Eletrônica e Computação na Área de Informática

Fernando Masanori Ashikaga

Um Método Frugal para o Problema de Minimização de Pilhas Abertas

Tese aprovada em sua versão final pelos abaixo assinados

.....

Prof. Dr. Nei Yoshihiro Soma

Orientador

.....

Prof. Dr. Homero Santiago Maciel

Chefe da Divisão de Pós-Graduação

Campo Montenegro

São José dos Campos, SP – Brasil

2001

Um Método Frugal para o Problema de Minimização de Pilhas Abertas

Fernando Masanori Ashikaga

Composição da Banca Examinadora:

Prof. Dr. Sérgio Roberto Matiello Pellegrino	Presidente – ITA
Prof. Dr. Nei Yoshihiro Soma	Orientador (ITA)
Prof. Dr. Edgar Toshiro Yano	(ITA)
Prof. Dr. Horacio Hideki Yanasse	(INPE)
Prof. Dr. Reinaldo Morabito Neto	(UFSCar)

ITA

Índice de Figuras	iii
Índice de Tabelas	iv
Resumo	v
Abstract	vi
Agradecimentos	vii
Capítulo 1: Introdução	1
Capítulo 2: Problemas relacionados com o sequenciamento de padrões de corte	9
2.1 O sequenciamento de padrões através da Teoria dos Grafos	10
2.2 Heurísticas rápidas para o sequenciamento de padrões	12
2.3 Projeto de circuitos VLSI e o MOSP	14
2.4 O uso de meta-heurísticas para o sequenciamento de padrões	16
Capítulo 3: Definição do Problema	17
3.1 Objetivo da dissertação e análise crítica dos trabalhos anteriores	18
3.1.1 Objetivo da dissertação	18
3.1.2 Análise crítica dos trabalhos anteriores que utilizaram a Teoria dos Grafos	18
3.1.3 Análise crítica das heurísticas de Yuen	19
3.1.4 Análise crítica dos trabalhos sobre o PLA Folding	21
3.1.5 Análise crítica dos trabalhos que utilizam meta-heurísticas	22
3.2 Formulação matemática alternativa do MOSP	22
3.3 Modelagem por grafos do MOSP	24
3.3.1 O MOSP e o PLA Folding	26
3.4 Cliques, conjuntos independentes e circuitos hamiltonianos	30
3.4.1 Aproximação de cliques e conjuntos independentes maximais em grafos	31
3.4.2 Circuitos hamiltonianos e o problema do caixeiro viajante	33
3.4.3 Circuitos hamiltonianos e o algoritmo Extensão-Rotação	35
3.5 Avaliação crítica sobre o valor da abordagem sugerida para o MOSP	38
3.5.1 Análise crítica quanto à determinação do clique idealmente maximal para a resolução do problema	39
3.5.2 Análise crítica quanto à determinação de circuitos hamiltonianos para a resolução do problema	41
Capítulo 4: Nossa Contribuição para o MOSP	44
4.1 Análise empírica dos grafos MOSP quanto ao clique maximal e número de arestas	44
4.2 Método frugal sugerido: iniciar sempre pelo clique maximal encontrado pela busca gulosa por vértice de grau mínimo no grafo complementar	54
4.3 Descrição das heurísticas desenvolvidas através do método frugal sugerido	60
4.3.1 Heurística Extensão-Rotação (<i>her</i>)	62
4.3.2 Heurística de Contração Recursiva de Cliques (<i>hcrc</i>)	66
4.4 Implementação e Testes Computacionais	68
4.4.1 Testes iniciais comparando as heurísticas <i>her</i> , <i>hcrc</i> , <i>ybs99</i> e <i>yuen3</i>	70
4.4.2 Testes extensivos comparando as heurísticas <i>her</i> e <i>yuen3</i>	73
Capítulo 5: Conclusões	79
Referências Bibliográficas	82

Índice de Figuras

Figura 1 Exemplo de um padrão de corte da indústria de vidro.	2
Figura 2 Grafo MOSP e respectiva matriz de padrões.	4
Figura 3 Processo de formação do grafo MOSP através da união de padrões-cliques.	5
Figura 4 Possível seqüência de fechamento de peças para o grafo MOSP, obtida ao se aproximar um circuito hamiltoniano, e possível seqüência de padrões associada.	6
Figura 5 Seqüência de peças obtida pela contração recursiva de cliques e possível seqüência de padrões associada.	7
Figura 6 Grafo MOSP e padrões associados segundo modelo proposto por Yanasse, Becceneri e Soma (99)	11
Figura 7 Matriz de padrões e correspondente grafo MOSP.	25
Figura 8 Exemplo de um PLA das funções f_6 e f_7	27
Figura 9 Personality matrix do PLA associado às funções f_6 e f_7	27
Figura 10 Respectivamente, matriz inicial, matriz obtida pela permutação e matriz do PLA otimizado.	28
Figura 11 PLA otimizado após o “seqüenciamento de padrões”.	29
Figura 12 Dez primeiras linhas da matriz de um PLA associado ao processador 8085.	30
Figura 13 Algoritmo Extensão-Rotação.	37
Figura 14 Comparação entre as densidades relativas à $n \log n$, $3n \log n$ e das médias encontradas em grafos MOSP gerados com até 5 e 6 peças por padrão.	51
Figura 15 Médias dos cliques maximais encontrados para 100 padrões de corte.	54
Figura 16 Determinação do Clique Maximal via conjuntos independentes maximais no grafo complementar.	56
Figura 17 Parte inicial de um grafo complementar MOSP, que é um exemplo de pior caso para a busca gulosa por vértice de grau mínimo para detecção de conjuntos independentes maximais.	57
Figura 18 Pior caso citado por Johnson (74) de grafo MOSP para a busca gulosa.	58
Figura 19 Exemplo de como a contração de arcos executada pela heurística ybs99 pode produzir hipergrafos completos maiores que o clique maximal.	60
Figura 20 Os vértices 3 e 5 são conjugados em relação à seqüência parcial $Spe = \{7, 2,$ $6, 9, 4, 12, 10, 8, 11, 13, 1\}$	65
Figura 21 Exemplo do funcionamento da heurística her	65
Figura 22 Seqüência de peças Spe encontrada através heurística hcrc e possível seqüência de padrões Spa associada.	67

Índice de Tabelas

Tabela 1 Tamanho do clique maximal e porcentagem de arestas encontrados em cada caso de teste.....	46
Tabela 2 Respectivamente valores de $2x \frac{n \log n}{n(n-1)}$, $\sqrt{n \log n}$ e $2 \log_2 n$	49
Tabela 3 Valores de C necessários para superar $\varepsilon \sqrt{n \log n}$	52
Tabela 4 Valores de C necessários para superar $2 \log_2 n$	52
Tabela 5 Tempos médios para a busca gulosa detectar o clique maximal.	56
Tabela 6 Médias do número máximo de pilhas abertas e respectivos tempos em segundos.	71
Tabela 7 Valores de C necessários para obtermos uma determinada densidade em uma matriz MOSP de dimensão n	74
Tabela 8 Médias das pilhas abertas e tempos comparando her e yuen3	75
Tabela 9 Porcentagem de vértices que “sobraram” no final da heurística her	78

Resumo

Consideramos nesta dissertação um problema, NP-difícil, de sequenciamento de padrões, visando minimizar o número máximo de pilhas abertas em torno de uma máquina industrial de corte. Estamos interessados em métodos *frugais*, os quais, seguindo à terminologia de Halldórson (91), são aqueles – métodos – que além de utilizar poucos recursos computacionais – tempo e espaço – possuem ideias e implementações simples. A modelagem do problema pela Teoria dos Grafos foi a escolhida para a obtenção de tais métodos, na tentativa de se identificarem aspectos estruturais que, porventura, pudessem emergir e auxiliar na sua resolução. A partir daquela modelagem, descobrimos ser o grafo *complementar* bastante esparso e possuidor de um conjunto independente maximal surpreendentemente grande, se comparado ao número de vértices do grafo. Através da informação adicional, fornecida por estes dois aspectos estruturais encontrados no grafo modelado, um método geral, baseado no clique maximal, foi desenvolvido. A *frugalidade* do método está no uso de uma conhecida heurística gulosa, de tempo linear no número de vértices, para detecção de conjuntos independentes. A partir do método geral, duas heurísticas puderam ser desenvolvidas: a primeira, de detecção de circuitos hamiltonianos, obtidos através de uma versão do algoritmo extensão-rotação para grafos randômicos, com o circuito inicial composto pelos vértices do clique maximal; e a segunda, de contração recursiva de cliques. Realizamos testes computacionais comparando as nossas heurísticas com aquelas pertencentes ao estado da arte encontrado na literatura. Os resultados demonstram que as duas heurísticas desenvolvidas através do método proposto são competitivas, tanto em termos de tempo e espaço como de erro médio, além da facilidade de implementação.

Abstract

This dissertation is a study of an NP-hard problem of sequencing patterns to minimize the maximum number of open stacks around industrial saw machines (MOSP).

Our major aim was to develop a, so-called, *frugal* method, *i.e.* one that uses a polynomial amount of computer resources – time and space – and possesses a simple idea and ease of implementation, *cf.* Halldórson (91). To obtain such a method, two major ideas were used, the modeling of **MOSP** via Graph Theory (**MOSP** graph) and the correspondence between a maximal clique with the patterns sequencing. It is shown that, in the mean experimental case, a **MOSP** graph has very large cliques or, what is the same, that the complement of such graphs has very large independent (stable) sets. The frugality of the method comes from this observation, since it was possible to find large independent sets, ideally maximal ones, in linear time.

Based on the general method, two heuristics were developed: the first detects a hamiltonian circuit through a modified version of the extension-rotation algorithm for random graphs and the second through the recursive clique contraction. Moreover, both heuristics have as initial input the vertices of the maximum clique detected.

Extensive computational experiments were carried out, comparing the suggested two heuristics with those belonging to the state-of-art from the literature. The practical results confirm the expected frugal behavior of the two suggested heuristics, moreover, it was possible to infer that they are highly competitive, in time, space and mean error with those well known from the literature.

Agradecimentos

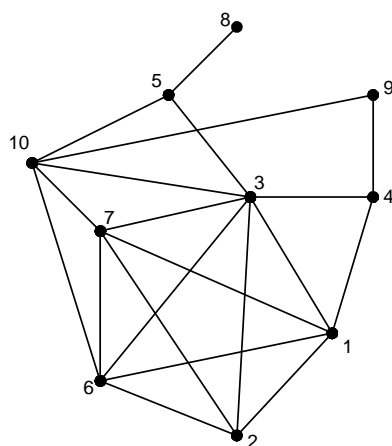
Ao José Carlos Becceneri e Horacio Hideki Yanasse do INPE/LAC, pelas proveitosas discussões e pelo auxílio empenhado durante o desenvolvimento deste trabalho.

Aos funcionários da biblioteca do ITA pelo auxílio na obtenção dos artigos para a revisão bibliográfica.

À professora Yoshiko Wakabayashi, que na graduação do IME-USP ajudou o autor a adquirir o interesse inicial pela área de otimização combinatória.

Ao professor Nei Yoshihiro Soma, pelo compromisso e interesse demonstrados durante a orientação desta dissertação.

À CAPES que ajudou financeiramente o desenvolvimento desta pesquisa.



Grafo relativo a um problema com 10 padrões de corte com até 4 peças diferentes cada

Capítulo 1: Introdução

A motivação para esta dissertação surgiu em um contexto de problemas industriais de corte, cuja análise pode ser encontrada em Dychoff e Finke (92), Dychoff, Abel e Gal (85) e, em vernáculo, em Morabito e Arenales (92). Um objetivo legítimo, de grande parte destes problemas, é atender aos pedidos dos consumidores e conseguir, ao mesmo tempo, minimizar o desperdício de matéria-prima. Podem existir outros objetivos, como, por exemplo, diminuir o movimento de peças em estoque. Tendo em vista a estes vários resultados pretendidos, determinados *padrões de corte* são gerados e sequencialmente cortados a partir de chapas maiores de matéria-prima em estoque. Em alguns casos específicos, como na indústria de vidro, se permitirmos uma maior quantidade e variedade de tipos diferentes de peças em cada um dos padrões de corte gerados, menor poderá vir a ser o desperdício de matéria-prima, *cf.* Yuen (95). Na **Figura 1** podemos ver um exemplo de padrão de corte da indústria de vidro, sendo a área hachurada o desperdício. Observe que neste estudo não tratamos apenas do caso guilhotinado ou somente dos problemas da indústria de vidro, são apenas exemplos ilustrativos.

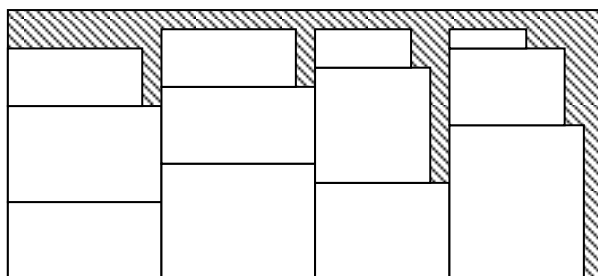


Figura 1 Exemplo de um padrão de corte da indústria de vidro.

Um problema subsequente e correlato, objeto de estudo nesta dissertação, é a obtenção de um bom sequenciamento dos padrões gerados. Na indústria de vidro, por exemplo, é desejável que não haja muitas pilhas de peças em torno da máquina de corte. As diferentes peças cortadas, em geral, são empilhadas (peças iguais com peças iguais) – pilhas abertas – e somente removidas quando todos os padrões que contenham àquela peça forem sequenciados. Frequentemente há uma limitação de espaço em torno da máquina de corte – limitação na quantidade de pilhas abertas. Evita-se, assim, confundir peças semelhantes por iguais. Problemas similares aos da indústria de vidro ocorrem no corte de madeira, *cf.* Yanasse (97a), onde também é desejável minimizar a quantidade de pilhas abertas em torno da máquina de corte industrial.

Assim, o foco desta dissertação será o de se minimizar o número máximo destas pilhas, doravante designado por *MOSP* (do acrônimo *Minimization of Open Stack Problem*). Neste problema esperamos obter uma sequência de corte, para os padrões gerados, que minimize o número máximo das pilhas de peças abertas que se formam ao redor da máquina de corte.

Faz-se mister notar que a geração de padrões e o posterior sequenciamento são estreitamente relacionados. Na indústria de vidro o problema de minimização de pilhas abertas, mencionado anteriormente, é tão importante que, caso se estime que um número proibitivo de pilhas seja aberto, os padrões são novamente gerados, *cf.* Yuen (95). Neste caso

é feita uma severa restrição da quantidade de peças permitidas em cada padrão, o que pode provocar um maior desperdício de matéria-prima, contrariamente ao previamente desejado.

O MOSP pode modelar muitas situações práticas, além das citadas nesta introdução, *cf.* Becceneri (99). Dentre estas várias situações, uma aplicação surpreendente do modelo MOSP aqui adotado é no projeto de circuitos VLSI, como veremos no Capítulo 3. Através de uma redução a um dos problemas desta área de circuitos VLSI, em aparência distante do problema de sequenciamento de padrões de corte, foi demonstrado recentemente, *cf.* Linhares e Yanasse (98, 00), que o MOSP é um problema NP-difícil. Desta forma, aliado ao aspecto prático a motivar o estudo do MOSP, existe também um interesse teórico associado ao problema.

Uma grande quantidade de abordagens para a resolução do MOSP, tanto de maneira exata quanto aproximada aparece em Becceneri (99), todavia, os algoritmos resultantes são computacionalmente pesados. A motivação inicial desta dissertação foi o de se tentar encontrar heurísticas frugais para o problema, *i.e.*, que sejam simples de serem implementadas; utilizem poucos recursos computacionais, e o mais importante, que gerem aproximações de qualidade ao problema. O ponto de partida na busca dos métodos frugais foi a modelagem do MOSP pela Teoria dos Grafos, onde algumas informações *estruturais* do problema foram encontradas, as quais nos permitiram gerar heurísticas muito rápidas, competitivas em erro com os melhores resultados da literatura. Descobrimos que o complementar do grafo MOSP é muito esparso e possui um conjunto independente maximal grande, *i.e.* maior do que aquele esperado para grafos randômicos com igual número de vértices. A partir dessas informações, duas heurísticas foram desenvolvidas: a primeira, de circuitos hamiltonianos obtidos através do método extensão-rotação, a ser descrito na Seção 3.4.3, onde o caminho inicial é fornecido pelos vértices do clique maximal; e a segunda, de

contração recursiva de cliques maximais. Estas duas heurísticas fazem parte de um método mais geral baseado no conjunto independente maximal encontrado no grafo complementar do problema.

Para uma melhor ilustração das ideias apresentadas até aqui, apresentamos na **Figura 2**, um exemplo de um grafo MOSP para um conjunto de 10 padrões contendo até 4 peças por padrão. Cada *linha* da matriz, ao lado do grafo, representa um *padrão*, onde um quadrado cheio com o número da peça indica que o padrão contém a *peça* associada à *coluna* correspondente. No grafo, as peças são representadas por vértices e uma aresta liga dois vértices se e somente se existe um padrão que contenha as peças por eles representadas. As arestas múltiplas são contadas como uma única. Em destaque aparece o clique maximal (vértices 1, 3, 7, 6 e 2), de tamanho 5, cuja importância será descrita com mais detalhes ao longo da dissertação.

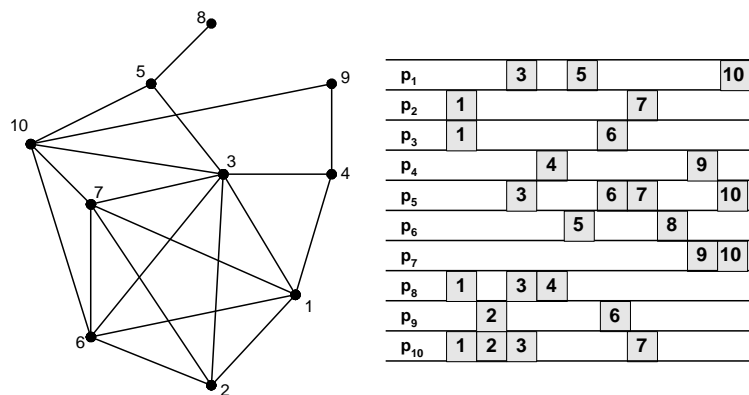


Figura 2 Grafo MOSP e respectiva matriz de padrões.

Na Teoria dos Grafos, podemos entender o grafo MOSP como uma *união de cliques*. Através da **Figura 3** podemos visualizar melhor como é formado o grafo MOSP da **Figura 2**. Cada padrão P_i aparece em destaque como um clique em relação ao grafo MOSP que será formado, em linhas tracejadas. A união deles irá formar o grafo MOSP que estudamos.

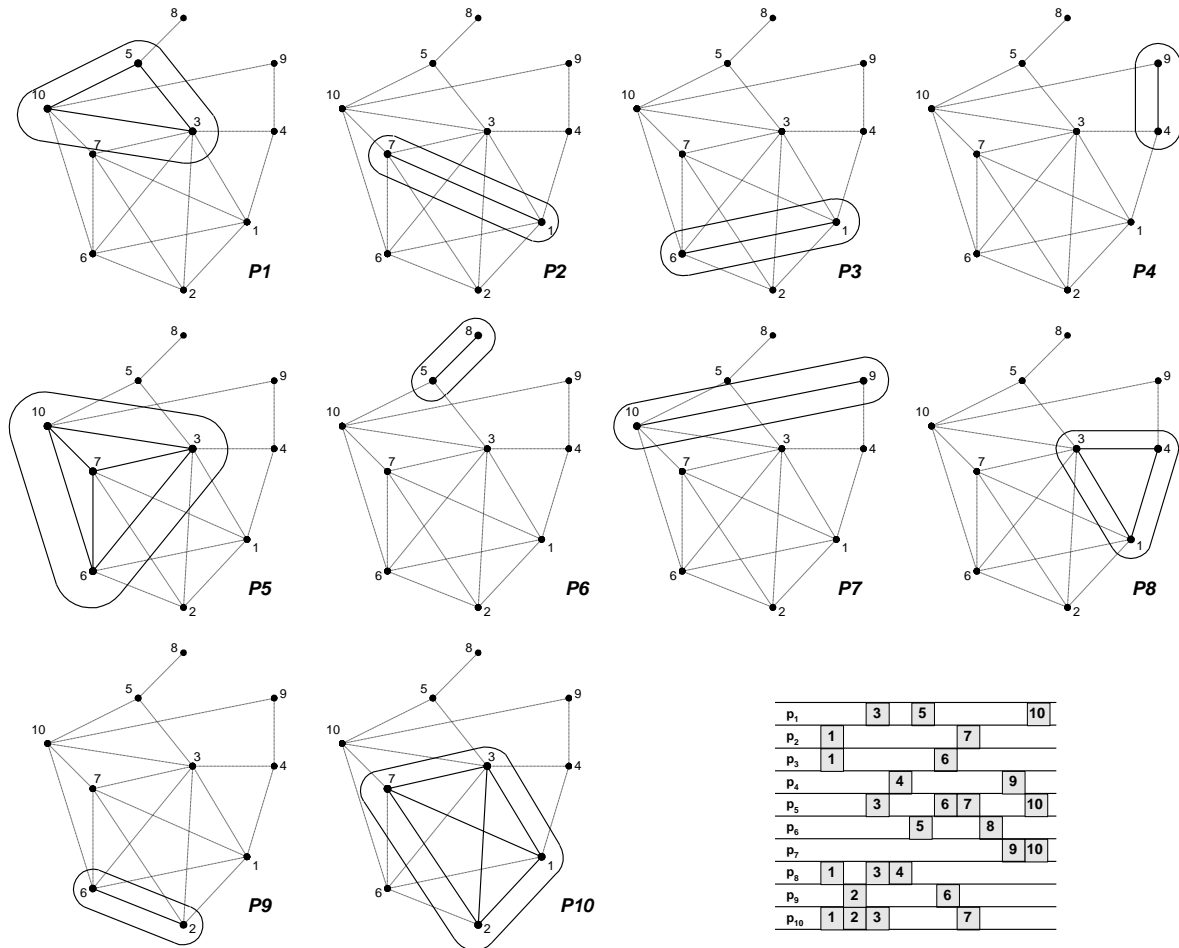


Figura 3 Processo de formação do grafo MOSP através da união de padrões-cliques.

Uma ideia de como funciona a heurística de circuitos hamiltonianos pode ser visualizada na **Figura 4** onde, à esquerda, vemos uma possível sequência de fechamento de peças, no caminho hamiltoniano $Spe = \{6, 7, 3, 2, 1, 4, 9, 10, 5, 8\}$. Uma peça é dita *aberta*

quando o primeiro arco do caminho incide sobre o vértice associado, e é *fechada*, quando todos os vértices associados aos arcos incidentes forem percorridos. Associada a esta sequência de peças, vemos à direita na **Figura 4**, uma possível sequência de padrões, $Spa = \{9, 10, 3, 2, 8, 4, 7, 5, 1, 6\}$, que irá abrir no máximo 5 pilhas ao longo do seu processamento. Esta sequência de padrões é obtida da seguinte forma: percorre-se a sequência de peças do final para o início, colocando-se os padrões que contenham a peça atual do fim para o começo na sequência de padrões. Por exemplo, como a peça 8 é a última na sequência de fechamento de pilhas, o padrão 6 será o último a ser processado.

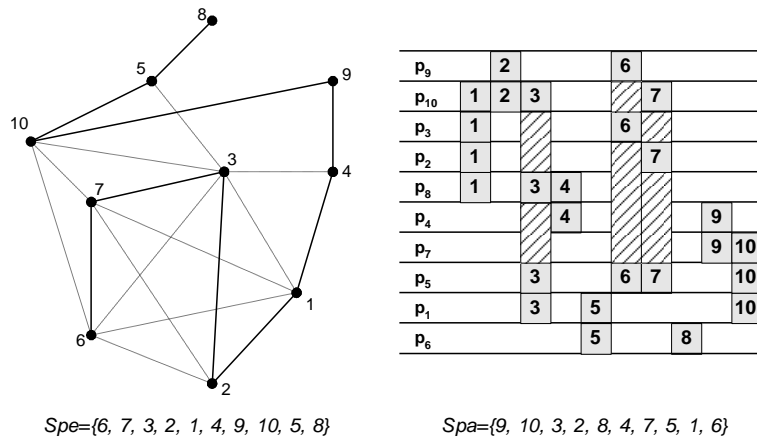


Figura 4 Possível sequência de fechamento de peças para o grafo MOSP, obtida ao se aproximar um circuito hamiltoniano, e possível sequência de padrões associada.

As áreas hachuradas indicam pilhas que permanecem abertas devido à existência de padrões não sequenciados e que contêm a peça representada pela coluna. O clique maximal é um limitante inferior para o MOSP, cf. Yanasse, Becceneri e Soma (99), logo a sequência de padrões exemplificada é ótima, pois o tamanho do clique maximal coincide com o número máximo de pilhas abertas na sequência de padrões. Note que, antes de termos a sequência de padrões Spa , já sabíamos que o número máximo de pilhas abertas seria 5 através da sequência de peças Spe .

Podemos ter uma ideia da heurística de contração recursiva de cliques através da **Figura 5**, onde C_1 , C_2 , C_3 e C_4 são os cliques contraídos a cada passo. A sequência de peças obtida através da contração recursiva é $Spe=\{3, 1, 2, 6, 7, 5, 10, 4, 9, 8\}$. Uma possível sequência de padrões para a sequência de peças é $Spa=\{9, 3, 10, 2, 5, 1, 8, 7, 4, 6\}$. Novamente foi obtida a sequência ótima, no exemplo dado, que abre no máximo 5 pilhas durante o processamento dos padrões.

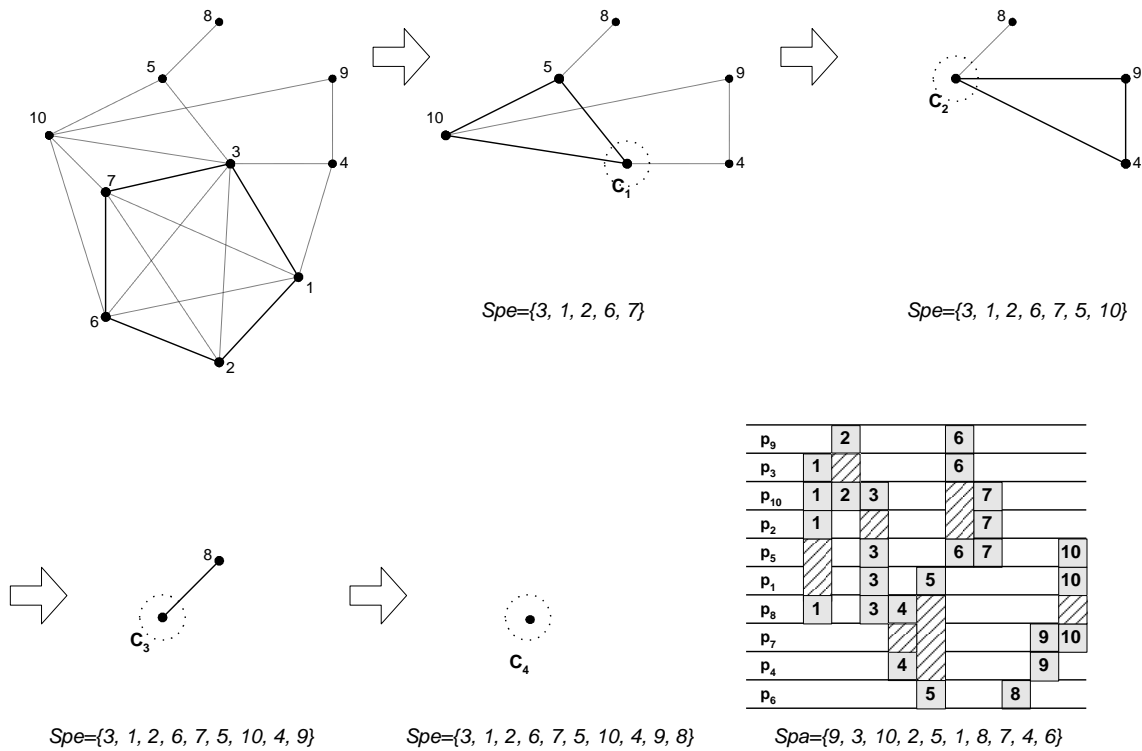
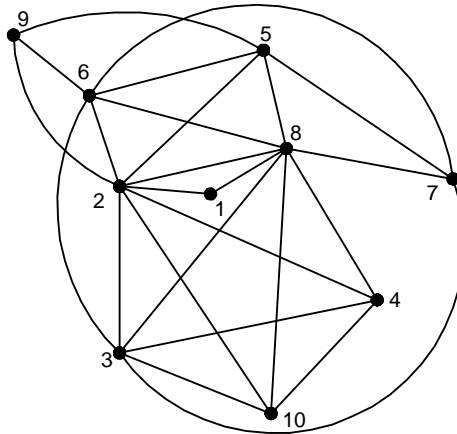


Figura 5 Sequência de peças obtida pela contração recursiva de cliques e possível sequência de padrões associada.

Este nosso método é *frugal*, pois utilizamos como base uma heurística de tempo linear e que tem o seu melhor comportamento exatamente em grafos MOSP. Esta heurística simples, *cf.* Halldórson (97), para detecção de grandes conjuntos independentes em grafos esparsos, se mostrou adequada ao problema de sequenciamento de padrões, como veremos ao longo desta dissertação.

No Capítulo 2 resumimos o estado da arte relacionado com o MOSP. Em seguida o problema é definido de maneira formal no Capítulo 3. Adicionalmente, apresentamos uma visão dos tópicos da Teoria dos Grafos diretamente relacionados à abordagem proposta, como cliques, conjuntos independentes maximais e circuitos hamiltonianos. Procuramos, ao mesmo tempo, fazer no Capítulo 3 uma análise crítica do estado da arte para o MOSP descrito anteriormente. Mostramos desta maneira ser original a forma de resolução desenvolvida nesta dissertação. No Capítulo 4 apresentamos a nossa contribuição para a resolução do problema, principalmente através das heurísticas propostas, suas implementações e o resultado de testes computacionais que sugerem a competitividade delas. Finalmente, no Capítulo 5, apresentamos as nossas conclusões e sugestões de problemas originais relacionados com o MOSP.



*Grafo Mosp de 10 padrões com
até 4 peças diferentes cada*

Capítulo 2: Problemas relacionados com o sequenciamento de padrões de corte

Neste capítulo procuraremos, de maneira tentativa, agrupar os resultados recentes relacionados com o nosso problema, por abordagens iguais e/ou similares. Alguns dos artigos descritos não são específicos para o MOSP, mas foram considerados pertinentes ao nosso estudo. Os trabalhos são apresentados quanto ao erro e/ou performance computacional, segundo a importância relativa que apresentam para a resolução do problema e não em ordem cronológica. Julgamos mais importantes para o nosso estudo, em primeiro lugar, abordagens que utilizam a Teoria dos Grafos e, em seguida, heurísticas que sejam rápidas, com erro médio pequeno. Uma análise crítica dos trabalhos que se encontram nesta revisão bibliográfica será feita *a posteriori* no Capítulo 3. Portanto, aqui nos limitaremos apenas à descrição dos artigos estudados.

Na Seção 2.1 são descritos, brevemente, trabalhos que utilizam a Teoria dos Grafos para atacar o problema de sequenciamento de padrões. Em seguida, apresentaremos na Seção 2.2 artigos que descrevem heurísticas que utilizem pouco tempo computacional, podendo assim fornecer um fator de comparação em relação ao tempo para o atual trabalho. Na Seção

2.3 analisaremos alguns resultados selecionados da área de projeto de circuitos VLSI, particularmente do problema Programmable Logic Array (PLA) Folding, devido a este problema possuir uma semelhança muito grande com o MOSP. Finalmente, na Seção 2.4, apresentamos artigos onde meta-heurísticas são utilizadas, especificamente busca tabu e *simulated annealing*, para resolver o sequenciamento de padrões.

Adotou-se aqui a representação sugerida em Blum (95) de se representar por n ao tamanho do *input* do problema, a menos de explicitamente dito ao contrário.

2.1 O sequenciamento de padrões através da Teoria dos Grafos

A melhor aproximação para o MOSP, até o presente momento, é a heurística de *contração* de arcos *cf.* Yanasse, Becceneri e Soma (99). Naquela heurística, de tempo $O(n^3)$, um MOSP é modelado por um grafo, onde um dado padrão (P) induz a um clique, e os vértices desse clique são as peças pertencentes a P . O grafo MOSP é, desta forma, uma *união* de cliques, onde arcos paralelos são contados como um único, e o problema visto como um percorrimto de arcos.

Exemplos de grafos MOSP segundo esta modelagem já foram apresentados no capítulo introdutório, outro exemplo adicional pode ser visualizado na **Figura 6**, com a correspondente matriz de padrões de corte. A *contração* de um arco se resume em uma junção dos dois vértices a ele incidentes. Os arcos ligados àqueles vértices passam a estar ligados ao novo vértice e arcos múltiplos que surgem também são considerados como um único. Nessa heurística, os arcos são contraídos até chegarmos a um hipergrafo completo, idealmente maximal. O sequenciamento dos padrões é feito a partir deste grafo inicial, cuja solução é imediata.

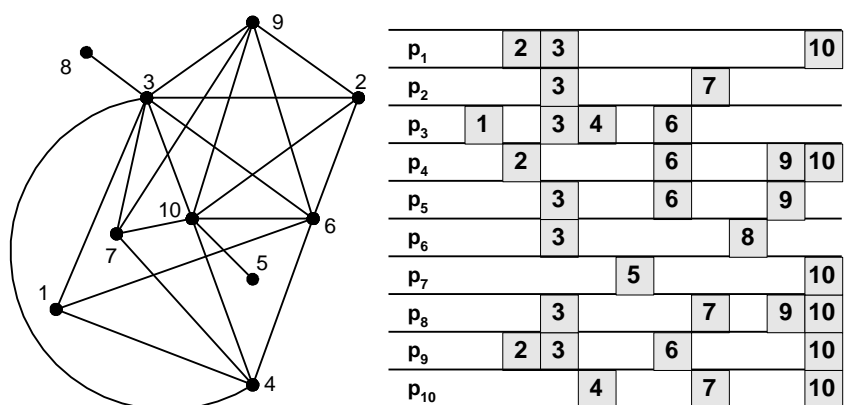


Figura 6 Grafo MOSP e padrões associados segundo modelo proposto por Yanasse, Becceneri e Soma (99)

A heurística de contração de arcos fornece também o melhor limitante inferior conhecido para o MOSP, dado pelo número de vértices do hipergrafo completo resultante da contração sucessiva. Dada a importância desta heurística para o MOSP, iremos fazer referência a ela ao longo desta dissertação como *ybs99*. Um estudo mais detalhado da heurística de contração de arcos pode ser encontrado em Becceneri (99), ou ainda em Yanasse, Becceneri e Soma (99).

Yanasse (98) prova que uma enumeração das possíveis soluções para o MOSP obtida pela permutação de pilhas completas é mais eficiente em relação à enumeração das possíveis sequências de padrões, mesmo para o caso de termos mais peças que padrões. Através deste resultado podemos trabalhar no sequenciamento de pilhas de peças terminadas para se obter a sequência desejada de padrões.

Em Yanasse (97b) uma heurística é sugerida para o problema do sequenciamento através da *eliminação* de arcos. O grafo é simplificado até se tornar uma árvore, onde existem algoritmos polinomiais, não exatos, para este caso particular, *cf.* Yanasse (96) para sua resolução.

Um problema clássico, dentro da Teoria dos Grafos, é o problema do caixeiro viajante. Madsen (88) e Dyson e Gregory (74) utilizam esta abordagem para o sequenciamento de padrões. Seu objetivo não era o de resolver o MOSP, mas sim, minimizar o número de descontinuidades, um problema semelhante, porém distinto do nosso, cf. Linhares e Yanasse (00). Uma descontinuidade aparece quando um item é parcialmente atendido em um padrão e não está incluído no padrão seguinte. Minimizar descontinuidades pode ser algo desejado em uma linha de produção, por exemplo, se o tempo de montagem é importante dentro do processo. Nestes dois trabalhos uma modelagem é feita via problema do caixeiro viajante, com resolução matricial. Dyson e Gregory (74), mencionam que a abordagem exata consumiu um grande tempo computacional, tornando inviável sua utilização prática para mais que 20 padrões. Já Madsen (88) apresentou resultados melhores utilizando uma heurística para aproximação do problema do caixeiro viajante modelado.

Observamos que, embora a modelagem utilizada em Dyson e Gregory (74) e Madsen (88) seja feita através do problema do caixeiro viajante, em sentido estrito, podemos considerar que não foi utilizada a abordagem por Teoria dos Grafos em ambas resoluções. Neste mesmo sentido, são recentes as abordagens pela Teoria dos Grafos. O primeiro trabalho por nós conhecido e que utiliza tal abordagem surge com Lins (89), que sugere uma heurística para resolver um caso particular do MOSP, onde os padrões possuem no máximo duas peças diferentes.

2.2 Heurísticas rápidas para o sequenciamento de padrões

Dentre as heurísticas por nós pesquisadas e que possuem boa aproximação para o MOSP, as mais rápidas são as heurísticas de Yuen (91, 95), particularmente a heurística de número 3 dentre as seis descritas pelo autor. Em uma indústria de vidro, via programação

linear, cf. Gilmore e Gomory (61), são gerados padrões de corte visando minimizar o desperdício de matéria-prima. Posteriormente, é atacado o sequenciamento de padrões. O autor menciona o problema particular desta indústria, onde é especialmente importante minimizar o número de pilhas abertas. Quando são abertas muitas pilhas ocorrem problemas no manuseio das peças, além de ser difícil diferenciar visualmente peças com dimensões próximas, o que pode provocar confusão. O procedimento usual para contornar o problema do sequenciamento é gerar novos padrões, restringindo severamente o número de peças permitidas, o que pode acarretar um maior desperdício de matéria-prima, contrário ao objetivo inicial. Foram seis as heurísticas propostas, sendo que os testes realizados apontam a terceira como sendo a melhor, em tempo e aproximação. Posteriormente, Yuen e Richardson (95) desenvolvem métodos exatos baseados naquelas seis heurísticas. Utilizaremos como fator de comparação a heurística número 3, visto ser a melhor delas, a qual passaremos a referenciar como **yuen3**. Observe, pela sua definição a seguir, como é simples esta heurística. Inicialmente são definidas as seguintes variáveis:

C_j : número de peças em comum que o padrão j possui com as pilhas já abertas

N_j : número de peças novas que o padrão j irá introduzir, caso escolhido

M_j : $C_j - N_j$

A heurística **yuen3** é definida da seguinte maneira:

Heurística yuen3:

Escolha o próximo padrão não sequenciado com o maior M_j

Se houver empate, escolha o padrão com menor N_j

Se persistir o empate escolha o padrão com menor índice j

Madsen (79) estuda o problema da geração de padrões de corte e posterior sequenciamento e em uma pequena indústria de vidro. Devido ao tamanho da empresa, o autor sustenta que o modelo deveria ser de fácil utilização e consumir pouco tempo de processamento. Uma matriz (0-1) A_{ij} , onde $a_{ij} = 1$ caso os padrões i e j possuam pelo menos uma peça em comum é definida, onde a ordem de linhas/colunas determina a sequência de corte dos padrões. Desta forma, uma permutação de linhas/colunas que torne os elementos diferentes de zero mais próximos da diagonal principal irá reduzir as discontinuidades entre os itens, fornecendo uma sequência para o corte dos padrões. Este problema, NP-difícil, cf. Garey *et al.* (78), é conhecido como *bandwidth minimization* e para resolvê-lo o autor utilizou o algoritmo de aproximação fornecido por Gibbs *et al.* (76).

2.3 Projeto de circuitos VLSI e o MOSP

Gate Matrix Layout (GML) e Programmable Logic Array (PLA) Folding fazem parte de uma classe de problemas de arranjo linear de circuitos VLSI. O termo linear se refere à otimização combinatória unidimensional das portas lógicas deste tipo de circuito. Uma breve explicação destes problemas, bem como o seu relacionamento com o MOSP, será dada na Seção 3.3.1.

A literatura para os problemas GML ou PLA é bem mais extensa que a encontrada para o MOSP, o que nos motivou, também, ao seu estudo, dado serem problemas semelhantes ao MOSP. Para não perder o foco da dissertação, procuramos nos restringir aos artigos que apresentam alguma semelhança com a abordagem por Teoria dos Grafos aqui estudada. Um detalhamento mais completo do relacionamento entre estes problemas com o MOSP deverá aparecer em Linhares (01).

As ideias para o PLA Folding são igualmente aplicáveis para o Gate Matrix Layout. Deste modo, optamos por utilizar a notação PLA Folding durante a dissertação, devido a uma motivação de ordem prática: em geral os artigos PLA Folding que escolhemos mostram a disposição da matriz utilizada na resolução do problema no mesmo sentido que a matriz de padrões/peças MOSP, enquanto os artigos GML utilizam a matriz transposta. Para evitar uma provável confusão preferimos a notação PLA.

Ferreira e Song (92) apresentam uma heurística de tempo $O(n^3)$, onde $n = \max(\text{lin}, \text{col})$ relativos a matriz associada ao problema, para o PLA Folding utilizando uma abordagem por Teoria dos Grafos. O artigo mostra um detalhamento da complexidade em cada etapa da heurística proposta e uma caracterização para a otimalidade das soluções.

Lecky *et al.* (89) utilizam a busca gulosa sugerida por Johnson (74) para detecção do clique maximal nas suas heurísticas propostas para o PLA Folding. As heurísticas apresentadas por estes autores são de tempo $O(n^5)$ e $O(n^4)$, $n = \max(\text{lin}, \text{col})$, para o PLA. O mérito de Lecky *et al.* (89) é, além de apresentar uma aproximação melhor, em tempo e em erro, que trabalhos prévios, ser o primeiro artigo que modela o problema em um grafo onde o conjunto procurado é o clique maximal. Desta forma é criado um relacionamento mais natural entre cada par de colunas na matriz PLA, tornando cada clique no grafo um conjunto viável para o *folding*.

Xu *et al.* (95) utilizam a mesma abordagem, com a diferença de usarem o dual da busca gulosa de Johnson (74) para o clique maximal. Foi feita esta opção após a verificação experimental que os grafos complementares do PLA possuem menos arestas que o grafo original.

Como o nosso problema central é o MOSP, e não o PLA Folding, nos deteremos apenas nos artigos citados na atual seção. Um apanhado de resultados anteriores aos descritos

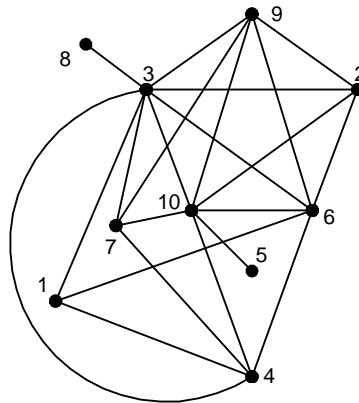
nesta seção para o PLA Folding e que utilizam a Teoria dos Grafos pode ser encontrado em Möhring (90).

2.4 O uso de meta-heurísticas para o sequenciamento de padrões

Um bom resultado para o MOSP através de meta-heurísticas é o de Fink e Voss (99). Estes autores estudaram o MOSP e o problema de minimizar o espalhamento de ordens de serviço utilizando *simulated annealing* e busca tabu. O espalhamento de uma peça pode ser medido através da quantidade de padrões processados entre o corte da primeira ocorrência da peça até a última ocorrência dela. É outro problema que apresenta semelhança com o MOSP, sendo, porém, distinto, *cf.* Linhares e Yanasse (00). Testes computacionais confrontando seus resultados com aqueles apresentados em Foester e Wäescher (98) – que também utilizaram a meta-heurística *simulated annealing* – são realizados. A comparação é feita apenas para o problema do espalhamento, visto que Foester e Wäescher (98) não analisam o MOSP.

Faggioli e Bentivoglio (98) sugerem um método exato para o MOSP. Uma solução inicial é obtida através de uma heurística semelhante às encontradas em Yuen (95), ou seja, estabelecendo critérios de escolha entre os padrões. Esta solução é depois aprimorada utilizando-se de uma busca tabu, a partir da qual é feita uma enumeração sobre o conjunto de padrões para a obtenção da solução exata.

Na sequência temos o capítulo que trata da definição do problema. Nele incluiremos uma análise crítica dos trabalhos enunciados de forma a permitir organizar melhor as informações até agora fornecidas.



“Problema bem definido, meio problema resolvido” Pólya

Capítulo 3: Definição do Problema

O presente capítulo procura definir bem o nosso problema, especificando o contexto, escopo e limites. Na Seção 3.1, explicitaremos a tese de que, através de Teoria dos Grafos, podemos obter heurísticas rápidas e com boa aproximação em relação à qualidade da solução para o MOSP. Concomitantemente, faremos uma análise crítica dos trabalhos da revisão bibliográfica apresentada no Capítulo 2, mostrando que a abordagem aqui adotada é diversa das anteriores. Uma formulação matemática alternativa é apresentada na Seção 3.2, onde iremos introduzir a nomenclatura que será utilizada ao longo da dissertação. Na Seção 3.3 iremos apresentar a modelagem por grafos adotada. Feito isto, na Seção 3.4, iremos apresentar os conceitos de clique, conjunto independente e circuitos hamiltonianos, com os seus resultados na Teoria de Grafos mais significativos para o MOSP. Note que a revisão bibliográfica do problema já foi efetuada no Capítulo 2, sendo que na Seção 3.4 apresentaremos resultados que influenciaram à nova abordagem sugerida pela nossa dissertação. Finalmente, na Seção 3.5, apresentaremos algumas ideias e/ou justificativas que nos induziram a considerar viável a abordagem sugerida para resolução do MOSP.

3.1 Objetivo da dissertação e análise crítica dos trabalhos anteriores

3.1.1 Objetivo da dissertação

Como já foi mencionado anteriormente, iremos tratar apenas do problema de sequenciamento de padrões visando minimizar o número máximo de pilhas abertas durante o processo. Desta forma, não tratamos o problema da geração destes mesmos padrões, apenas o seu sequenciamento. Tratar do problema da geração de padrões em conjunto com o sequenciamento é algo que ainda não possui muitos trabalhos na literatura e está fora do escopo aqui proposto.

A tese defendida nesta dissertação é de que, através da Teoria dos Grafos, iremos obter um método rápido e que forneça uma aproximação de qualidade para o MOSP. A modelagem por grafos utilizada por Yanasse, Becceneri e Soma (99) é a base para o nosso estudo. Naquela modelagem o grafo MOSP, conforme já comentado anteriormente, é uma união de cliques, que representam cada padrão estudado. Esta modelagem, peculiar, deve produzir um grafo resultante também com características peculiares, tendo configurado esta hipótese o ponto focal e inicial desta dissertação.

Observe que não é o objetivo da dissertação a Teoria dos Grafos, ela será apenas uma ferramenta auxiliar para a resolução de um problema de corte industrial, ainda que os resultados apresentados possam levar a conclusões mais gerais de interesse teórico.

3.1.2 Análise crítica dos trabalhos anteriores que utilizaram a Teoria dos Grafos

Yanasse, Becceneri e Soma (99) perseguiram o algoritmo exato para o MOSP, *cf.* Becceneri (99). Ao desenvolver a heurística de contração de arcos, buscavam um limitante inferior de qualidade para o método exato. Neste sentido, a heurística resultante possui uma

excelente aproximação, porém, em detrimento do tempo computacional. A heurística, implementada em C++ e executada em um Pentium 233 MHz, gasta até 138 segundos para processar 100 padrões, cf. Becceneri (99). O fator tempo é um limitante sério, mas não impeditivo, visto que o valor obtido pela heurística é quase sempre o ótimo, ou apresentando uma diferença de *uma ou duas* pilhas em relação a este valor. Mesmo desconsiderando o método exato, em um ambiente industrial de corte, o tempo da heurística de contração de arcos serve, com folga, para o propósito, já que os problemas possuem um tamanho moderado. Como esta heurística apresenta o menor erro médio, iremos utilizar **ybs99** como um dos fatores de comparação para as heurísticas desenvolvidas na dissertação. Como contraponto, é interessante notar que, dado que o problema de geração dos padrões de corte é, usualmente, associado ao MOSP, pode haver situações onde estes padrões são continuamente gerados até ser obtido um número ideal de pilhas abertas, cf. Yuen (95). Nesta situação, uma heurística rápida pode ser mais conveniente, mesmo tendo um erro médio *um pouco* maior.

Uma outra possibilidade é o esquema proposto de *eliminação* de arcos até chegarmos a uma árvore no grafo MOSP, conforme descrito em Yanasse (96). Em Becceneri (99) foi feita uma comparação entre a heurística de contração e a de eliminação de arcos, onde o número máximo de pilhas abertas, em média, foi menor na heurística de contração de arcos. Posteriormente à publicação do artigo Yanasse (96), foi constatado que alguns dos algoritmos apresentados não eram exatos. Portanto, não iremos considerar o esquema de eliminação de arcos nesta dissertação.

3.1.3 Análise crítica das heurísticas de Yuen

A heurística **yuen3**, proposta em Yuen (95), apresenta resultados bons, fornecendo sequenciamentos muito próximos do ótimo, não tanto quanto a heurística de contração de

arcos *ybs99*, porém em um tempo bem mais reduzido. O exemplo mencionado anteriormente, de sequenciamento de 100 padrões, em um Pentium 233 MHz, que custou 138 segundos para a heurística *ybs99*, é processado em 0,02 segundo pela heurística *yuen3*, abrindo apenas 3 pilhas a mais na média, cf. Becceneri (99). O grande mérito da heurística *yuen3* é a sua simplicidade, sendo relativamente fácil, também, a sua implementação.

No entanto, nos parece que, como *yuen3* não possui uma visão de conjunto de todos os padrões, aquela heurística pode ter sua qualidade de aproximação deteriorada, a medida em que aumentar o número de padrões/peças permitidos, fato que iremos averiguar nos nossos testes computacionais.

Se formos tentar imaginar o comportamento da heurística *yuen3* em um grafo MOSP, já teremos um início atípico: começar pelo padrão que contenha menos peças. Em seguida, *yuen3* irá tentar sequenciar o menor clique/padrão que contenha mais vértices em comum com as peças dos padrões já sequenciados. Durante este processo, em um determinado passo, os vértices do clique maximal terão sido sequenciados. Ou seja, existe a possibilidade de termos uma heurística mais rápida através de uma abordagem por grafos, pois podemos chegar ao sequenciamento das peças relativas ao clique maximal *antes* que *yuen3*, através, por exemplo, da busca gulosa para conjuntos independentes maximais proposta na nossa abordagem.

Iremos utilizar *yuen3*, em conjunto com *ybs99*, para auferir a qualidade da aproximação das nossas heurísticas. Adicionalmente *yuen3*, como é muito rápida, nos irá fornecer um fator de comparação em relação ao tempo computacional. Não existe até o momento uma heurística que possua uma aproximação melhor que *yuen3* em um tempo *também* inferior, na média. Encontrar uma heurística com estas características é um desafio que perseguimos nesta dissertação.

3.1.4 Análise crítica dos trabalhos sobre o PLA Folding

Os trabalhos para PLA Folding de Lecky *et al.* (89) e Xu *et al.* (95) mostram uma semelhança grande com o MOSP aqui tratado. Mais precisamente, como veremos na próxima seção, se trata do *mesmo* problema, se visto do ponto de vista da modelagem por grafos. Sendo assim, talvez devêssemos comparar as heurísticas para o PLA Folding com as heurísticas desenvolvidas pela abordagem aqui proposta. Contra essa comparação pesa o seguinte fato: o PLA Folding está inserido em um vasto contexto de arquitetura VLSI, com diferenciações não triviais entre as diversas fases, tipos de decodificação e outras idiossincrasias da área que estão fora do escopo delimitado nesta dissertação. Os únicos tempos fornecidos, *ref. op. cit.*, levam em conta ferramentas auxiliares de minimização em dois níveis, não proporcionando base para qualquer comparação. A título de exemplo, a função *in6*, utilizada na literatura como *benchmark* para o problema, contendo 33 entradas e 23 saídas, possui um tempo *total* de processamento para PLA Folding de mais de 580 segundos, executada em uma estação Sparc Sun, *cf.* Xu *et al.* (95), todavia não se tem o tempo gasto apenas pela heurística de sequenciamento. Devido a todas estas dificuldades, nesta dissertação nos limitaremos ao contexto dos problemas de corte industrial, como ocorre na indústria do vidro ou madeira. O que nos parece, é que apesar de terem a mesma modelagem, o grafo resultante é estruturalmente diferente. Na Seção 3.3.1 apresentaremos um exemplo de uma matriz associada a um PLA real utilizado no processador 8085. No exemplo apresentado, veremos que o grafo resultante é, em essência, distinto do grafo MOSP padrão no que se refere à estrutura, isto é, naquele caso, o PLA possui um clique maximal bem menor que o encontrado, na média, em grafos MOSP com o mesmo número de vértices.

3.1.5 Análise crítica dos trabalhos que utilizam meta-heurísticas

Não iremos levar em conta uma abordagem por meta-heurísticas nesta dissertação devido ao seu caráter, em geral, não *frugal*. Fink e Voss (99) mostram um tempo de 60 a 87 segundos para casos de 60 padrões para o *simulated annealing* e de mais de 100 segundos para a busca tabu, utilizando um Pentium 266 MHz. A heurística *ybs99* necessitaria para estes casos menos de 7 segundos e *yuen3* em uma pequena fração de segundo. Outros motivos que desaconselham o uso de meta-heurísticas decorrem da modelagem por grafos, adotada para o MOSP. Veremos que os cliques do grafo MOSP possuem uma função importante no sequenciamento ótimo dos padrões. Em relação a cliques, existem, por exemplo, resultados que mostram que a abordagem por *simulated annealing* tem uma degradação de performance quando o grafo original possui cliques grandes, *cf.* Jerrum (92) e Sasaki (91); ou que algoritmos genéticos para a determinação de cliques maximais têm como casos particularmente difíceis um conjunto de classes de grafos definidas em Marconi e Foster (98). Como veremos na Seção 3.4 sobre grafos, resultados teóricos recentes, *cf.* Arora *et al.* (92), mostram que, obter uma solução *aproximada* para o problema do clique maximal é tão difícil quanto obter sua solução exata. Esses indícios teóricos parecem apontar para abordagens que utilizem informações adicionais para seu bom desenvolvimento. Nos parece que esse conjunto de resultados, quando contextualizados para o caso específico do MOSP, não nos aconselha, sem mais justificativas, uma abordagem por meta-heurísticas.

3.2 Formulação matemática alternativa do MOSP

Na dissertação iremos utilizar a modelagem do problema através da Teoria dos Grafos. No entanto, achou-se prudente colocar uma formulação matemática antes, desta forma, poderemos aproveitar a nomenclatura introduzida nas seções seguintes.

O Problema da Minimização de Pilhas Abertas, MOSP, é definido por um conjunto $\Pi = \{1, \dots, n\}$ de *peças* e $\underline{\Lambda} = \{\underline{\lambda}_1, \dots, \underline{\lambda}_v\}$ de *padrões*, onde, cada um dos $\underline{\lambda}_j$ é um subconjunto de Π . As *peças* devem ser processadas por uma única máquina de corte e *pilhas* serão formadas ao redor desta. A cada *peça* $t \in \Pi$ associa-se uma pilha, única, a qual é *fechada*, *i.e.* removida do redor da máquina de corte, somente quando a demanda, por outras peças que estão presentes em padrões nos quais t aparece, for satisfeita e *após* o corte da peça em processamento. O objetivo a ser alcançado é, então, o de se obter uma sequência de padrões a serem cortados, no qual a quantidade máxima de pilhas abertas seja a menor possível.

Seja Π uma dada permutação de Π , $\Pi = \{\sigma(1), \dots, \sigma(n)\}$. Um padrão $\underline{\lambda}_i$ *domina* $\underline{\lambda}_j$, se $\underline{\lambda}_j \subset \underline{\lambda}_i$, onde $\underline{\lambda}_i, \underline{\lambda}_j \in \underline{\Lambda}$. Assume-se, também, sem perda de generalidade, que a inclusão é tomada em seu sentido próprio, *i.e.* padrões mutuamente dominantes são iguais. Define-se, ainda, ao conjunto $\Lambda = \{\lambda_1, \dots, \lambda_m\}$ como sendo formado por padrões maximais quanto à dominância. Sejam $\Pi' \subseteq \Pi$ e $\Lambda' \subseteq \Lambda$, o *maior fechamento* de Π' em Λ' ($\phi(\Pi', \Lambda')$) é definido como sendo um conjunto de padrões dentre os Λ' , cujas peças são de Π' e para o qual, pelo menos um padrão dominante maximal de Λ' é gerado.

Baseado nestas definições, uma possível formulação matemática para o problema, pode ser dada por:

$$z = \text{Minimize } \{ \{ \text{Máximo } (\Theta_1, \dots, \Theta_n) \}, \Pi = \{ \sigma(1), \dots, \sigma(n) \} \}$$

sujeito à:

$$\Theta_i = 1 + \theta_i;$$

$$\pi_i = \phi(\{ \sigma(1), \dots, \sigma(i) \} \cap \Pi_{i-1}, \Lambda_{i-1})$$

$$\Lambda_i = \Lambda_{i-1} \setminus \pi_i;$$

$$\Pi_i = \Pi_{i-1} \setminus \{ \forall p \mid p \in \lambda \subseteq \pi_i \text{ e } p \notin \lambda \subseteq \Lambda_i \}$$

$$\theta_i = \Theta_i - |\Pi_{i-1} \setminus \Pi_i|$$

$$i = 1, \dots, n$$

onde inicialmente:

$$\Lambda_0 = \Lambda; \Pi_0 = \Pi \text{ e } \theta_0 = 0;$$

Note-se que as variáveis Θ_k e θ_k indicam, respectivamente, a quantidade de pilhas abertas quando do k -ésimo corte e a quantidade de pilhas que permanece aberta após tal corte. Também é interessante notar que o espaço de busca das soluções ao problema é fatorial, configurando-se uma condição necessária, mas não suficiente, de uma dificuldade prática de resolução. Para uma outra formulação matemática, qual seja, modelo linear inteiro, cf. Yanasse (97a) e Becceneri (99).

3.3 Modelagem por grafos do MOSP

Aplicaremos a modelagem utilizada por Yanasse, Becceneri e Soma (99). Considere a matriz MOSP $m \times n$, de m padrões e n peças, definida por $M = (m_{ij})$, uma matriz (0,1) onde:

$$m_{ij} = \begin{cases} 1 & \text{caso } j \subseteq \Lambda_i \\ 0 & \text{c.c.} \end{cases}, \quad 1 \leq i \leq m, 1 \leq j \leq n$$

Note que existe a possibilidade de um padrão ter várias vezes uma mesma peça j . Nesta situação teremos igualmente o valor 1 na coluna associada, pois é indiferente ao modelo haver uma ou várias peças iguais dentro de um padrão. Em ambas as situações o número de pilhas abertas será idêntico.

Definimos o grafo MOSP associado como $G = (V, E)$, onde os vértices v_j de V correspondem às colunas j da matriz. Dois vértices v_i e v_j são adjacentes se ambas colunas i e j possuem 1 na mesma linha. As arestas múltiplas são contadas como uma única, salvo

referência em contrário. O grafo MOSP é conhecido na literatura como grafo de adjacência de colunas (*column adjacency graph*). A **Figura 7** apresenta uma matriz M de padrões, com o respectivo grafo MOSP associado.

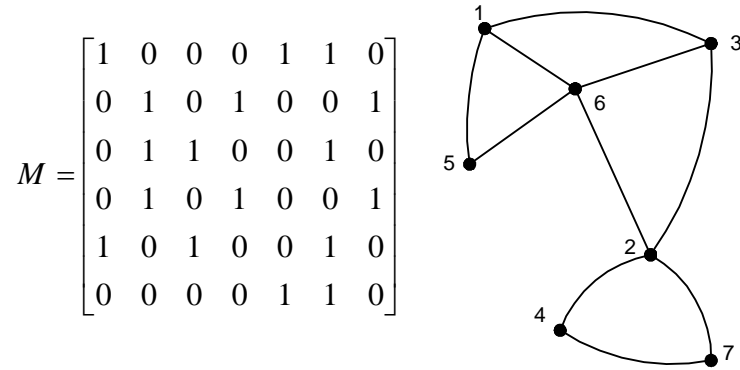


Figura 7 Matriz de padrões e correspondente grafo MOSP.

Uma peça é dita *aberta* quando o primeiro arco do caminho incide sobre o vértice associado, e é *fechada*, quando todos os vértices associados aos arcos incidentes forem percorridos. Uma possível sequência de fechamento de *peças* (*pilhas*) para o grafo MOSP da **Figura 7** seria o caminho hamiltoniano $Spe=(7, 4, 2, 6, 3, 1, 5)$. Uma sequência de *padrões* associada àquelas peças poderia ser $Spa=(2, 4, 3, 5, 1, 6)$ e o número máximo de pilhas abertas será de 3. Note que o clique maximal possui cardinalidade 3, e sendo ele um limitante inferior para o MOSP a solução exemplificada é ótima. Observe que os padrões 2 e 4 são iguais e que o padrão 1 domina o padrão 6, no sentido em que ele é um subconjunto do anterior. Heurísticas que trabalham diretamente com a matriz, algumas vezes efetuam um pré-processamento, visando a uma redução por dominância de linhas e/ou colunas, *i.e.*, de padrões e/ou peças. No grafo MOSP, isto é automaticamente feito na sua montagem, ao contar como únicas arestas múltiplas.

Note que, diferentemente da maior parte da literatura, consideramos uma permutação na sequência de *peças*, ao invés de uma permutação na sequência de padrões. Isto decorre do resultado demonstrado em Yanasse (98) indicando que é melhor resolver o sequenciamento

de padrões através de uma sequência de fechamento de pilhas associadas às peças. Pelo modelamento por grafos MOSP, sempre analisaremos sequências viáveis de peças, para posteriormente acharmos a sequência de padrões.

3.3.1 O MOSP e o PLA Folding

Iremos mostrar a seguir, que o MOSP, tal como foi formulado, é exatamente o PLA Folding, quando resolvido através da Teoria dos Grafos. O exemplo no qual nos basearemos, assim como as definições, são retirados de Ferreira e Song (92).

Considere duas funções lógicas f_6 e f_7 a serem implementadas em um PLA:

$$\begin{aligned} f_6 &= \bar{v}_1 v_5 + \bar{v}_2 \bar{v}_3 + v_1 v_3 + \bar{v}_5 \\ f_7 &= v_2 v_4 + \bar{v}_2 \bar{v}_4 \end{aligned}$$

onde v_1, v_2, v_3, v_4 e v_5 são variáveis booleanas e $v_i, \bar{v}_i, i=1, \dots, 5$ os literais associados.

O correspondente PLA está representado na **Figura 8**, onde cada ponto representa a presença de um circuito lógico. No plano AND as variáveis de entrada e seus complementos são responsáveis pelos termos relativos aos produtos, enquanto o plano OR efetua a soma dos produtos resultando nas funções desejadas f_6 e f_7 .

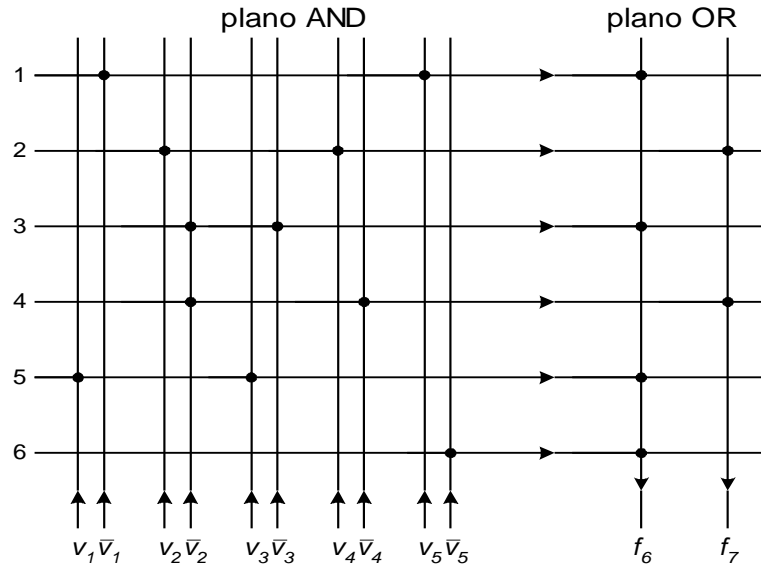


Figura 8 Exemplo de um PLA das funções f_6 e f_7 .

Considerando as variáveis v_1, v_2, \dots, v_k e as funções de saída $f_{k+1}, f_{k+2}, \dots, f_n$ perfazendo um total de n termos, define-se uma matriz (0,1) denominada *personality matrix*

$M = (m_{ij})$, onde:

$$m_{ij} = \begin{cases} 1 & \text{circuitológico presente em } v_j, \bar{v}_j \text{ ou } f_j \\ 0 & \text{c.c.} \end{cases}, \quad 1 \leq i \leq m, 1 \leq j \leq n$$

Observe que a definição da matriz contempla a variável e seu complemento em uma única coluna, apesar de que visualmente sejam duas. A *personality matrix* da **Figura 8** pode ser vista na **Figura 9**. Esta matriz associada ao PLA das funções f_6 e f_7 é idêntica à matriz MOSP exemplificada na Seção anterior, através da **Figura 7**.

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Figura 9 Personality matrix do PLA associado às funções f_6 e f_7 .

Para uma coluna lógica j de uma matriz $M'=(m'_{ij})$, obtida através da permutação das colunas de M , definimos $\text{Int}(M', j)$ ao intervalo $[a, b]$, onde:

$$a = \min i \mid m'_{ij} = 1 \text{ e}$$

$$b = \max i \mid m'_{ij} = 1$$

A partir desta definição, dizemos que um subconjunto de colunas lógicas de M' , denominado s_j ,

$$s_j = \{j_1, j_2, \dots, j_p\}, 0 \leq p \leq m$$

é “dobrável” (*foldable*) em uma simples coluna física j , se e somente se,

$$\text{Int}(M', j_1) \cap \text{Int}(M', j_2) \cap \dots \cap \text{Int}(M', j_p) = \emptyset$$

Assim, o PLA Folding, é um problema, que dada a *personality matrix* M , busca uma permutação de linhas, cujas colunas sejam “dobráveis” até ser obtida um número mínimo de colunas físicas ou subconjuntos s_j . Como são idênticas as matrizes exemplificadas para o MOSP (**Figura 7**) e PLA Folding (**Figura 9**), vamos aplicar o sequenciamento de padrões $Spa=(2, 4, 3, 5, 1, 6)$, obtendo, no exemplo, o PLA ótimo. Através da **Figura 10** o processo ficará mais claro. Na primeira parte visualizamos a matriz inicial, depois a matriz obtida por permutações, e finalmente a matriz do PLA otimizado. Os conjuntos obtidos por permutação são $s_1=\{1, 2\}$, $s_2=\{3, 4, 5\}$ e $s_3=\{6, 7\}$. O PLA otimizado é apresentado na **Figura 11**.

P ₁	1			5	6		P ₂				4		7		P ₂				2	4	7	
P ₂				4		7	P ₄	2			4		7		P ₄				2			
P ₃		2	3		6		P ₃			3					P ₃							
P ₄				4		7	P ₅	1							P ₅				1	3	6	
P ₅	1		3		6		P ₁					5			P ₁					5		
P ₆				5	6		P ₆								P ₆							

Figura 10 Respectivamente, matriz inicial, matriz obtida pela permutação e matriz do PLA otimizado.

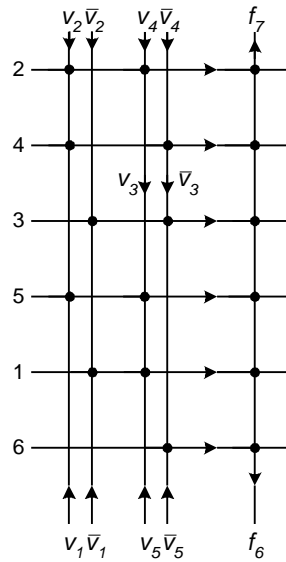


Figura 11 PLA otimizado após o “sequenciamento de padrões”.

Apresentamos neste exemplo, como o PLA Folding pode ser resolvido pelas heurísticas desenvolvidas nesta dissertação. Porém, temos razões para crer que, embora o PLA possa ser resolvido através de uma heurística MOSP, ele não terá um desempenho equivalente. Alguns PLA’s observados produzem grafos bem mais esparsos, se comparados com os grafos MOSP. Na **Figura 12** apresentamos um trecho da matriz 58×19 associada ao PLA real utilizado no processador 8085. O exemplo completo pode ser obtido em Ferreira e Song (92). Esta matriz é muito mais esparsa que a matriz MOSP. Linhas com apenas um único elemento não têm muito sentido para padrões de corte (uma única peça), além de que, na prática, é desejável uma quantidade e variedade grande nos padrões, cf. Yuen (95), já que queremos minimizar o desperdício de matéria-prima, entre outros objetivos. Caso a maior parte dos PLA’s produza matrizes esparsas, a aproximação aqui obtida pode ser não razoável para sua resolução.

```

00000000000000100000C
0000000001000000000C
000000100000000000C
000000000000000001C
000100000000000000C
000000010000100000C
000000010000000001
00000001000000000C
000000000000000001
00000000000000100C

```

Figura 12 Dez primeiras linhas da matriz de um PLA associado ao processador 8085.

3.4 Cliques, conjuntos independentes e circuitos hamiltonianos

Apresentaremos nesta seção alguns resultados que influenciaram o desenvolvimento desta dissertação. Na sua maior parte são resultados relativos às aproximações para alguns problemas clássicos de otimização combinatória e NP-difíceis, particularmente os problemas do clique maximal, conjunto independente maximal e caixeiro viajante. Para o MOSP modelado na forma de um grafo veremos que tanto o clique maximal – e indiretamente o conjunto independente maximal – como o circuito hamiltoniano desempenham um papel importante na elaboração das heurísticas. Foi feito, porquanto, um estudo prévio destes problemas.

Nos últimos anos tem havido muitos progressos no entendimento de como aproximações podem ser eficientemente computadas para problemas NP-difíceis. Um bom estudo sobre os desenvolvimentos recentes para vários problemas de otimização combinatória pode ser encontrado em Du e Pardalos (99). Uma outra coletânea, não tão recente, está em Shmoys (95) e em vernáculo, Campello e Maculan (94). Os tópicos que são relacionados

com o nosso trabalho também tiveram muitos progressos nos últimos anos. Iremos apresentar estes avanços com uma seção sobre cliques e conjuntos independentes, que possuem uma maior importância relativa na nossa abordagem, e prosseguiremos com duas seções sobre circuitos hamiltonianos.

3.4.1. Aproximação de cliques e conjuntos independentes maximais em grafos

Um *clique* é um conjunto de vértices mutuamente adjacentes. Um *conjunto independente* (ou *conjunto estável*) é um conjunto de vértices sem arestas que os unam. O estudo do conjunto independente, e do seu *alter ego* – clique – possui uma importância central na teoria combinatória, cf. Diestel (00).

O *complementar* de um grafo, representado por \overline{G} , possui os mesmos vértices que G , mas uma aresta é incidente a dois vértices, em \overline{G} , se e somente se, não existir uma aresta no grafo original G . Um clique no grafo original é um conjunto independente no grafo complementar, e vice-versa, sendo que este fato será muito utilizado na nossa abordagem.

O problema de *coloração* de um grafo consiste em atribuir o menor número possível de cores aos vértices de G , de tal forma que dois vértices adjacentes não possuam a mesma cor. Pelo motivo de que as cores induzem uma partição do grafo em conjuntos independentes, os problemas de aproximação do conjunto independente e da coloração são estreitamente relacionados.

Os problemas, de detecção do clique maximal e do conjunto independente maximal, são NP-difíceis, cf. Garey e Johnson (79). Sobre o problema do clique maximal existe um apanhado com várias referências em Pardalos e Xue (94). Caso haja interesse também em resultados sobre meta-heurísticas, cf. Bomze *et al.* (99). Em relação ao problema do conjunto independente maximal, muitas referências sobre aproximações podem ser obtidas em

Halldórson (91). Apresentamos a seguir um conjunto de ideias e resultados relevantes a estes dois problemas, que de maneira direta ou indireta, influenciaram os resultados obtidos por esta dissertação. Agrupamos os resultados em boas e más notícias, segundo o espírito de Crescenzi e Kann (98), como apresentado a seguir.

Boas notícias: ☺

1. Boppana e Halldórson (92) conseguem a melhor garantia de performance conhecida para o conjunto independente maximal. Através de um processo de exclusão de subgrafos é obtida uma garantia de performance de $O(n/(\log n)^2)$. O resultado vale *simultaneamente* para o problema do clique maximal e o da coloração. As técnicas utilizadas são estreitamente relacionadas com um limitante superior do número Ramsey $R(s, t)$, cf. Graham *et al.* (90), que é o menor inteiro n tal que, todos os grafos de ordem n ou contêm um clique de tamanho s , ou um conjunto independente de tamanho t .
2. A função *theta* de Lovász. Existe uma *fascinante*, cf. Knuth (94), função $\vartheta(G)$, determinada em tempo polinomial, introduzida por Lovász (79), que possui uma interessante propriedade *sandwich* entre duas funções NP-difíceis, $\omega(G) \leq \vartheta(\overline{G}) \leq \chi(G)$, onde $\omega(G)$ é o tamanho do clique maximal e $\chi(G)$ o mínimo número de cores necessários para o colorir o grafo G . Esta propriedade sugere que podemos obter boas aproximações para as duas funções NP-difíceis através da função *theta*. Todavia, após uma consulta à bibliografia da área, há indícios que é difícil obter tais aproximações a partir de $\vartheta(G)$.
3. *Minimum-degree greedy* ou heurística gulosa por vértice de grau mínimo para o conjunto independente maximal. Halldórson e Radhakrishnan (94a) fazem uma caracterização mais precisa da performance desta heurística linear, mostrando que a aproximação é bem melhor do que se apontava anteriormente em relação ao erro. Seguindo este trabalho

inicial, os mesmos autores aprimoram o resultado para grafos limitados pelo número de vizinhos (*bounded-degree*), cf. Halldórson e Radhakrishnan (94b). Em seguida, Demange e Paschos (97) realizam uma parametrização deste novo resultado para o mesmo tipo de grafos. Um trabalho mais completo sobre a *minimum-degree greedy* foi posteriormente publicado em Halldórson e Radhakrisnan (97). Paralelamente, implementações do algoritmo dual desta *greedy*, para obtenção do clique maximal, se mostraram bastante eficazes, na prática, para grafos que possuem cliques relativamente grandes, cf. Homer e Peinado (96), conseguindo resultados muito próximos do ótimo. Estes resultados já eram apontados teoricamente por Kučera (95), que observou que cliques de tamanho $O(\sqrt{n \log n})$ podem ser facilmente encontrados examinando-se apenas os vértices de maior grau no grafo.

Más notícias: ☹

1. Além dos problemas do clique maximal e conjunto independente maximal serem NP-difíceis, recentemente foi demonstrado que até mesmo a garantia de boas aproximações é um problema difícil em NP. Arora *et al.* (92) mostram que *aproximar* o clique maximal dentro de um fator N^ϵ ($\epsilon > 0$ e $|G| = N$) também é NP-difícil. Desta forma, obter uma solução aproximada para o clique maximal é tão difícil quanto obter a solução exata. Assim, este problema não admite um esquema eficiente de aproximação, a menos que $P = NP$, o que não parece ser o caso. Uma boa introdução sobre a impossibilidade de aproximações pode ser encontrada em Kohayakawa e Soares (95).

3.4.2 Circuitos hamiltonianos e o problema do caixeiro viajante

Um Grafo $G(V, E)$ não-direcionado (direcionado) é *hamiltoniano* se possuir um *circuito hamiltoniano*, isto é, se admitir um caminho indireto (direto) fechado que passe

exatamente uma única vez em cada vértice, com exceção do primeiro, que é igual ao último. Se atribuirmos pesos c_{ij} a cada aresta $(i, j) \in E$ do grafo, o problema de se achar um circuito hamiltoniano de custo *mínimo* é conhecido como *problema do caixeiro viajante*, *TSP* (*Traveling Salesman Problem*), pois se associa um vértice a uma “cidade” e um arco à estrada que liga duas cidades.

O problema do caixeiro viajante é NP-difícil, *cf.* Garey e Johnson (79) e tem sido utilizado como laboratório de testes para muitas das ideias que têm surgido na otimização combinatória. No livro texto básico sobre o TSP, Lawler *et al.* (85), encontramos uma boa introdução de cada uma delas. Uma análise probabilística recente sobre o TSP também pode ser encontrada em Frieze e Yukich (01). Os códigos para a resolução do TSP, tendo como meta a resolução de problemas contendo um número cada vez maior de cidades têm se tornado mais elaborados nos últimos anos, *cf.* compare-se a abordagem de Lin e Kernighan (73) exposta em Syslo *et al.* (83) com a recente, proposta em Applegate *et al.* (98).

Dantzig *et al.* (54) demonstraram que é possível resolver problemas com um número grande de cidades através da programação linear. Vários progressos foram alcançados a partir desta abordagem inicial, permitindo obter a solução exata para problemas bem maiores. Applegate *et al.* (98) descrevem vários refinamentos dentro de um método exato por planos de corte que levaram à resolução ótima de um problema com 13.509 cidades.

Johnson e McGeoch (97) analisam várias heurísticas para o problema do caixeiro viajante. Os autores destacam a heurística Lin e Kernighan (73) como sendo aquela que produz a melhor aproximação em termos de erro médio. Uma derivação é feita através do uso repetido da heurística – com uma perturbação entre cada processamento – resultando numa versão denominada *Chained Lin-Kernighan*. Uma ampla discussão sobre implementações e resultados computacionais *i.e.*, estruturas de dados possíveis, escolha de soluções iniciais

visando uma solução exata e formas de perturbação referentes a esta versão, é apresentado em Applegate *et al.* (01).

3.4.3 Circuitos hamiltonianos e o algoritmo Extensão-Rotação

Cada uma das abordagens, descritas anteriormente para o problema do caixeiro viajante também pode ser aplicada na determinação de circuitos hamiltonianos, já que o último problema é um caso particular do primeiro. No entanto, preferimos apresentar aqui um outro algoritmo, denominado extensão-rotação, *probabilístico*, que se mostrou mais eficaz para a resolução do problema de sequenciamento de padrões. A abordagem probabilística vem apresentando resultados algumas vezes surpreendentes na área combinatória, *e.g.* Chvátal (84) ilustra alguns exemplos da aplicação dessa abordagem para a Teoria dos Grafos. Para a nossa resolução do MOSP, uma implementação baseada no método extensão-rotação, para detecção de circuitos hamiltonianos, forneceu uma heurística que, em quase a totalidade das classes de testes, alcançou um erro médio menor e, surpreendentemente, um tempo médio também inferior à heurística 3 de Yuen (95).

Existem vários exemplos de algoritmos que são considerados eficientes, na prática, não obstante o seu comportamento não favorável quando da análise do pior caso. O mais conhecido exemplo é do algoritmo Quicksort, cujo comportamento, na prática, é bem mais rápido que o tempo quadrático previsto para o pior caso. Nesta dissertação iremos utilizar um outro exemplo desta classe de algoritmos, denominado extensão-rotação.

Existem algumas classes de grafos com elevado número de arestas e que não são hamiltonianos. Estes exemplos podem ser considerados excepcionais, *cf.* Chvátal (85). Um questionamento natural é saber qual a probabilidade de que um grafo randomicamente gerado seja hamiltoniano. Desde o clássico resultado de Dirac (52), que demonstrou que basta que

todos os vértices tenham um grau $\geq n/2$, muitos investigam, através do número de arestas necessário, as condições suficientes para um grafo ser hamiltoniano. Existem outros conjuntos de condições que não exigem que o grafo tenha muitas arestas para que ele possua um circuito hamiltoniano, porém não nos ateremos a esses conjuntos nesta dissertação, devido a não serem representativos ao estudo do MOSP. Para a nossa dissertação consideraremos principalmente o resultado de Pósa (76), onde é demonstrado que, para uma dada constante c , utilizando o método extensão-rotação, quase todos os grafos com n vértices e $cn \log n$ arestas são hamiltonianos. Paralelamente, Koršunov (76) demonstrou que este resultado vale para $3n \log n$ arestas. Estes resultados nos dão uma melhor noção do número de arestas necessárias para que a probabilidade de que o grafo contenha um circuito hamiltoniano tenda para 1 à medida que o número de vértices tenda a infinito. Descrevemos a seguir o algoritmo extensão-rotação para detectar circuitos hamiltonianos em grafos randômicos, tal como descrito em Pósa (76).

Algoritmo extensão-rotação

Passo 0: Seja P um caminho, inicial, consistindo de um único vértice x_1 de G .

Passo 1: Neste ponto teremos um caminho $P(x_1, x_2, \dots, x_k)$, $k \geq 1$

Se P é um circuito hamiltoniano então pare.

Passo 2: Escolha um vizinho y de x_k

Caso $y \notin P$, transforme P por *extensão*, sendo o novo caminho $P(x_1, x_2, \dots, x_k, y)$.

Caso $y \in P$ e $y = x_i$, $k-2 \geq i \geq 1$, transforme P por *rotação*, sendo o novo caminho $P(x_1, \dots, x_{i-1}, x_i, x_k, x_{k-1}, \dots, x_{i+2}, x_{i+1})$, se não existir tal y o algoritmo termina sem retornar um circuito hamiltoniano.

Retorne para o passo 1.

O algoritmo extensão-rotação procura, a partir de um dado caminho P , estendê-lo até onde isto for possível. Quando isto não for factível, tenta modificar o caminho P de tal maneira que possibilite mais extensões a partir do novo vértice final. Esta modificação, conhecida por *rotação* é ilustrada na **Figura 13**, a seguir.

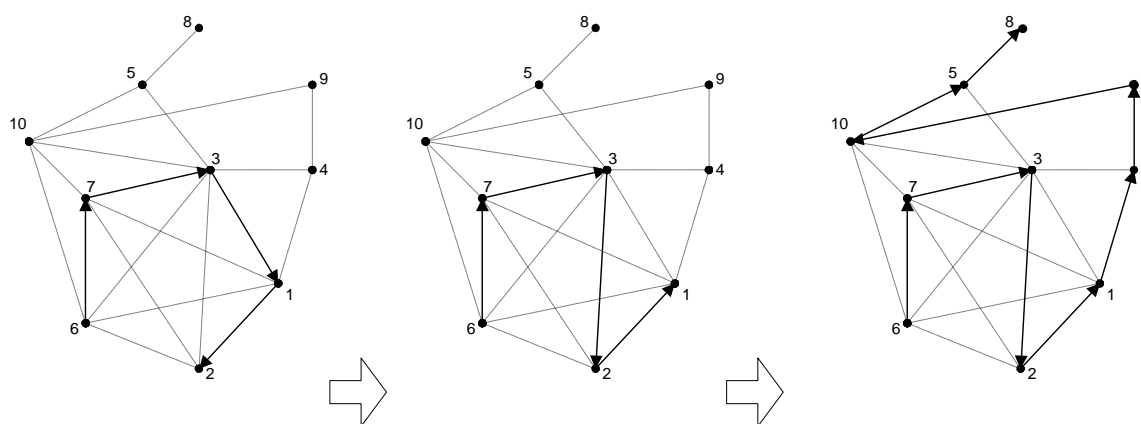


Figura 13 Algoritmo Extensão-Rotação

No exemplo, não é mais possível estender o caminho P (6, 7, 3, 1, 2), como existe uma aresta entre os vértices 3 e 2, é realizada uma rotação modificando o caminho para P (6, 7, 3, 2, 1), desta forma podemos continuar a fazer mais 5 extensões a partir do vértice 1 até obtermos o caminho final P (6, 7, 3, 2, 1, 4, 9, 10, 5, 8). Note que o grafo do exemplo é claramente não hamiltoniano.

Kučera (90) comenta, brevemente, alguns testes experimentais realizados sobre o algoritmo *Pósa*. Naqueles testes, foram inseridas, randomicamente, arestas em um conjunto de 500 vértices até surgir um grafo conexo sem vértices de grau 1 (condição necessária para a existência de um circuito hamiltoniano). O algoritmo *Pósa* extensão-rotação determinou um circuito hamiltoniano, na primeira tentativa, em 57 dos 60 experimentos, e nos 3 casos restantes bastou renumerar os vértices para obter um circuito hamiltoniano no grafo. Como

esperado pelo resultado teórico, o algoritmo foi bem sucedido em todos os casos testados, *ref. op. cit.*

Há muitas variações do algoritmo extensão-rotação, como, por exemplo, Angluin e Valiant (79), onde é apresentado um algoritmo com tempo $O(n(\log n)^2)$, para grafos com $cn \log n$ arestas, tratando também grafos direcionados. Como este, existem muitos resultados para grafos particulares: com muitas arestas, regulares, planares, etc. Em Bollobás, Fenner e Frieze (87) é descrito o primeiro algoritmo com tempo polinomial para grafos randômicos de características gerais. Outro algoritmo polinomial, também para o caso geral, está descrito em Broder, Frieze e Shamir (91), onde os autores propõem um algoritmo de tempo $O(n^3 \log n)$, com limitações impostas para as rotações, através de coloração de arestas, e de vértices finais para os caminhos parciais. Um apanhado sobre resultados de algoritmos probabilísticos em grafos randômicos, não se restringindo apenas aos circuitos hamiltonianos, pode ser encontrado em Frieze e McDiarmid (97).

3.5 Avaliação crítica sobre o valor da abordagem sugerida para o MOSP

Em Becceneri (99), conforme já havíamos descrito, uma grande quantidade de cenários práticos é apresentada. Dentre elas, explicitamos nas páginas anteriores o sequenciamento de padrões de corte em um ambiente industrial, *cf.* Seções 2.1, 2.2 e 2.4, e o projeto de circuitos VLSI, *cf.* Seção 2.3 e 3.3.1. Apesar de focarmos o problema no contexto de problemas de corte e empacotamento, os resultados podem ser igualmente aplicados para circuitos VLSI, *cf.* Seção 3.3.1. Lembramos que recentemente o MOSP foi demonstrado ser NP-difícil, *cf.* Linhares e Yanasse (98, 00). Faremos agora uma análise crítica da abordagem *frugal*, pretendida pela presente dissertação, através da Teoria dos Grafos. A análise será

dividida em duas partes: quanto ao uso do clique maximal e quanto ao emprego de circuitos hamiltonianos nos algoritmos para tratar o MOSP.

3.5.1 Análise crítica quanto à determinação do clique idealmente maximal para a resolução do problema

Seja K_l um clique no grafo MOSP onde $|K_l| = l$, sabemos que os padrões relacionados à K_l podem ser automaticamente sequenciados abrindo exatamente l pilhas, constituindo, porquanto, parte essencial para algoritmos de resolução do MOSP. Empiricamente observamos que os grafos MOSP possuem cliques grandes, se comparados com o seu número de vértices. Bollobás (85) aponta como sendo bem menor o tamanho dos cliques esperados, ou seja, para $G(n, 1/2)$ sendo um grafo com n vértices, onde cada aresta é gerada, de forma randômica e independente, com igual probabilidade de $1/2$. Com uma probabilidade que se aproxima de 1 quando n tende a infinito, o valor do clique é $\lceil r(n) \rceil$ ou $\lfloor r(n) \rfloor$, para $r(n) = (2 + o(1)) \log_2 n$. Para se ter uma ideia de como os grafos associados aos MOSP's possuem cliques grandes, um exemplo é fornecido em Abello, Pardalos e Resende (99), onde a partir de dados reais, modela-se uma rede de telefonia por intermédio de um grafo contendo 54 milhões de vértices e 170 milhões de arestas. Para esse exemplo foi encontrado um clique maximal de 32 nós $(32 \approx 1.245 \times \log_2(54 \times 10^6))$. Por outro lado, configurações aleatórias de MOSP's contendo apenas 50 padrões e, no máximo, igual quantidade de peças possuem, também, em média, cliques daquela ordem. Desta forma, pareceu-nos interessante explorar as consequências combinatórias deste fato, o que não foi realizado pelos trabalhos anteriores que abordaram ao problema.

Homer e Peinado (96) mostram experimentalmente que a busca gulosa para o clique maximal, cf. Johnson (74), possui ótima performance em grafos com cliques grandes, como é

o caso do grafo MOSP. Isto é corroborado com os resultados teóricos de Kučera (95). Seja $G(n, 1/2, k)$ o espaço de probabilidades cujos membros são gerados a partir de um grafo randômico $G(n, 1/2)$, inserindo-se aleatoriamente um clique de tamanho k , onde k deve ser maior que $c\sqrt{n \log n}$ para uma constante apropriada c . Os vértices do clique maximal, quase certamente, à medida que n tende a infinito, serão aqueles que possuem os maiores graus em G . Isto implica que a detecção daquele clique pode ser feita de forma eficiente. Alon, Krivelevich e Sudakov (98), melhoraram este resultado por um fator logarítmico, mostrando que existe um algoritmo polinomial que encontrará o maior clique de tamanho k , em $G(n, 1/2, k)$, com $k \geq \varepsilon \sqrt{n}$, dado $\varepsilon > 0$. Desta forma, apesar do problema teórico de se determinar o clique maximal ser extremamente difícil, cf. Arora *et al.* (92), vemos que o grafo MOSP *pode* ter aproximações do clique com boa performance apenas com uma busca gulosa de tempo linear. Outro resultado, que em certo sentido valoriza a abordagem aqui sugerida, apontado em Jerrum (92), mostra que a meta-heurística *simulated annealing* requer um processamento em tempo superpolinomial para detectar um clique pouco maior que o obtido pela busca gulosa, implementável em tempo linear. Todos estes resultados apontam, pelo menos em princípio, um valor, tanto teórico quanto prático, para o estudo de uma abordagem *frugal* na qual o fato de termos cliques tão grandes nos grafos MOSP é explorado.

É bem sabido que o clique de um grafo G é exatamente o conjunto independente nos mesmos vértices no grafo complementar \overline{G} . Recentes resultados para conjuntos independentes, cf. Boppana e Halldórson (92) e Halldórson e Radhakrishnan (94a, 94b, 97), apontam vantagens em aproximar o clique maximal através da busca gulosa pelo vértice de grau mínimo, buscando assim o conjunto independente maximal no grafo complementar, principalmente quando o tamanho deste conjunto é grande, se comparado ao número de vértices. Nesta dissertação iremos trabalhar com uma única estrutura de dados, onde teremos

simultaneamente o grafo original G e o seu correspondente complementar \overline{G} , e isto implica que poderemos utilizar heurísticas para detecção de conjuntos independentes no MOSP, ou seja, encontrar um clique através de seu *alter-ego* no grafo complementar, o conjunto independente.

O resultado teórico, contrário à nossa abordagem, mais forte é apresentado em Arora *et al.* (92), onde se prova que *aproximar* o clique maximal dentro de um fator N^ϵ ($\epsilon > 0$ e $|G| = N$) *também* é NP-difícil, a menos que $NP = P$. Desta maneira, temos neste desafio uma motivação adicional para estudar o MOSP segundo a proposta *frugal* desta dissertação.

3.5.2 Análise crítica quanto à determinação de circuitos hamiltonianos para a resolução do problema

É natural pensar em um caminho hamiltoniano quando estamos querendo um sequenciamento. Isto implica pensar em uma heurística para obter um sequenciamento de padrões via caixeiro viajante no caso MOSP. De certa forma, é o que faz a heurística *yuen3*, ao procurar o próximo padrão entre aqueles que possuam mais peças em comum com as já sequenciadas.

Madsen (88) e Dyson e Gregory (74) utilizam o caixeiro viajante para a resolução do problema das discontinuidades, que é semelhante ao MOSP. Nesta dissertação utilizaremos uma abordagem distinta destes autores. Iremos sequenciar as *peças* através de um circuito hamiltoniano e posteriormente sequenciar os *padrões*. Em um trabalho preliminar a esta dissertação, Ashikaga *et al.* (00), utilizando uma heurística de inserção de arcos, no caso a de maior distância (*farthest insertion*), *cf.* Syslo *et al.* (83), já obtiveram bons resultados em termos de tempo e erro médio na sua implementação realizada, indicando ser factível a

abordagem por circuitos hamiltonianos para sequenciar as peças com o intuito de se obter o sequenciamento de padrões.

O fato de empregarmos circuitos hamiltonianos como abordagem na resolução do MOSP não implica que os grafos MOSP sejam hamiltonianos. De fato, como foi exemplificado anteriormente, muitos grafos MOSP não são hamiltonianos. Utilizaremos algoritmos para aproximação de circuitos hamiltonianos tendo como meta a sequência de peças, não sendo necessário para o processo que a mesma seja um circuito no grafo.

Apesar do problema de se detectar circuitos hamiltonianos em um grafo geral ser NP-difícil, *cf.* Garey e Johnson (79), podemos encontrar facilmente circuitos hamiltonianos, se o número de arestas do grafo for suficientemente grande, *cf.* Pósa (76) e Koršunov (76). O grafo MOSP, como já comentado, possui este número mínimo necessário de arestas para a maior parte dos casos. Adotaremos como base para a detecção de circuitos hamiltonianos o algoritmo probabilístico extensão-rotação descrito na Seção 3.4.3. Uma possível melhoria no algoritmo *Pósa* será aproveitar os vértices do clique maximal para o circuito inicial, já que eles formam um caminho, pois fazem parte de um subgrafo completo, e com isto espera-se tornar menor o tempo de processamento.

Pode-se objetar que o processo de rotação pode deteriorar a qualidade da aproximação, caso não seja bem empregado, na resolução do MOSP. Para evitar este inconveniente tomaremos o cuidado de fazer somente as rotações *necessárias* e nas *proximidades* do clique maximal, apenas para dar início ao processo de extensões sucessivas. Ainda não foi efetuado um estudo teórico mais abrangente para aplicar rotações em todas os momentos do sequenciamento, sendo este uma sugestão para um trabalho futuro. É importante notar que a nossa versão do algoritmo extensão-rotação *também* é probabilística, no sentido em que não existe uma checagem de término a cada passo, *e.g.* hipergrafo

completo na contração de arcos na heurística *ybs99*, conforme ficará claro adiante, configurando-se, porquanto, uma outra contribuição original do trabalho.

Capítulo 4: Nossa Contribuição para o MOSP

No Capítulo 3 procuramos definir o nosso problema, apresentando uma formulação matemática alternativa, resumindo os aspectos da Teoria dos Grafos importantes para a nossa dissertação, bem como, avaliando de forma crítica a abordagem *frugal* proposta. Podemos introduzir agora com detalhes nossa contribuição para o problema. Na Seção 4.1 apresentaremos uma amostragem dos grafos MOSP, relacionando o valor encontrado do clique maximal com o número de arestas provenientes da modelagem do conjunto de padrões de corte, os quais foram gerados aleatoriamente. Em seguida, resumiremos na Seção 4.2 o método geral proposto. A base deste método é o clique maximal encontrado através da busca gulosa linear por vértice de grau mínimo no grafo MOSP complementar. Na Seção 4.3, descrevemos as duas heurísticas desenvolvidas, começando pela que apresentou os melhores resultados, modelada a partir de uma versão do algoritmo extensão-rotação para detecção de circuitos hamiltonianos em grafos randômicos, passando em seguida à heurística de contração recursiva de cliques. Finalmente, na Seção 4.4, faremos uma exposição sucinta dos testes computacionais relativos à implementação das nossas heurísticas, confrontando-as, quanto ao erro médio e tempo, com a heurística *yuen3*, apresentada em Yuen (95) e a heurística de contração de arcos *ybs99*, descrita em Yanasse, Becceneri e Soma (99) e Becceneri (99).

4.1 Análise empírica dos grafos MOSP quanto ao clique maximal e número de arestas

Apresentamos a seguir uma amostragem dos grafos MOSP em relação ao clique maximal e número de arestas encontrado. Os padrões utilizados para os testes foram propostos em Becceneri (99), onde, naquele trabalho, são geradas matrizes $n \times n$, com n

padrões e n peças diferentes, cada padrão contendo até no máximo C peças dentre as n possíveis. Para cada linha da matriz é gerado um número aleatório entre 1 e C . A seguir, também aleatoriamente, são escolhidas as colunas que deverão indicar a presença de um determinado tipo de peça dentro do padrão de corte representado pela linha da matriz. A seguir, é feita uma consistência de dados nas matrizes MOSP de modo a evitar que sejam gerados padrões de corte sem nenhuma peça (linhas contendo somente zeros), ou que um tipo de peça não seja utilizada por nenhum padrão (colunas contendo somente zeros). Também padrões que contenham apenas uma única peça são coibidos, já que neste caso é preferível cortar a peça isolada, sem necessidade de gerar um padrão de corte adicional.

Para cada caso de teste foram geradas 2000 matrizes aleatórias com os grafos MOSP correspondentes. Na **Tabela 1** podemos visualizar alguns dos resultados relacionados às médias encontradas do clique maximal e número de arestas nos testes gerados. Uma importante observação é a de que estamos contando arestas múltiplas como *uma única* em todos os cálculos efetuados.

A notação $\langle n^1 \rangle_c \langle n^2 \rangle$ indica que temos $\langle n^1 \rangle$ de padrões e peças, sendo $\langle n^2 \rangle$ a maior quantidade de peças em um dado padrão, *e.g.* $10c02$ têm-se 10 padrões e 10 peças distintas e 2 é a quantidade máxima de peças em qualquer um dos padrões. A aproximação para o clique maximal é fornecida pela busca gulosa por vértice de grau mínimo a ser descrita na Seção 4.2 e possui como título “ m_clique ”. O número de arestas é indicado pela porcentagem em relação ao grafo completo, ou seja $2x \frac{n_arestas}{n(n-1)}$ e é rotulado como

“ $\%arestas$ ”. Iremos omitir os dados a partir do momento em que a densidade do grafo chegar até aproximadamente 1, quando ele for quase um grafo completo, já que o clique maximal encontrado irá se aproximando do valor de n , tornando-se constante a partir daquele valor.

Caso	m_clique	$\%arestas$	Caso	m_clique	$\%arestas$	Caso	m_clique	$\%arestas$
-------------	-------------------------------	-------------------------------	-------------	-------------------------------	-------------------------------	-------------	-------------------------------	-------------------------------

10c02	2,80	0,25	40c03	3,55	0,11	50c25	44,68	0,99
10c03	3,58	0,40	40c04	4,32	0,16	50c26	46,01	0,99
10c04	4,71	0,55	40c05	5,27	0,23	50c27	47,16	1,00
10c05	5,95	0,70	40c06	6,32	0,31	60c02	3,07	0,04
10c06	7,16	0,82	40c07	7,46	0,38	60c03	3,53	0,07
10c07	8,22	0,90	40c08	8,73	0,46	60c04	4,24	0,11
10c08	9,05	0,95	40c09	10,25	0,54	60c05	5,12	0,16
10c09	9,60	0,98	40c10	11,91	0,61	60c06	5,97	0,21
10c10	9,87	0,99	40c11	13,75	0,68	60c07	6,89	0,27
20c02	2,91	0,13	40c12	15,82	0,74	60c08	7,92	0,34
20c03	3,59	0,21	40c13	18,09	0,80	60c09	9,05	0,40
20c04	4,58	0,31	40c14	20,41	0,84	60c10	10,34	0,47
20c05	5,76	0,42	40c15	22,73	0,88	60c11	11,79	0,53
20c06	7,08	0,53	40c16	25,08	0,91	60c12	13,48	0,59
20c07	8,52	0,64	40c17	27,42	0,93	60c13	15,31	0,65
20c08	10,12	0,73	40c18	29,71	0,95	60c14	17,33	0,70
20c09	11,87	0,81	40c19	31,66	0,97	60c15	19,62	0,74
20c10	13,50	0,87	40c20	33,41	0,98	60c16	22,14	0,79
20c11	15,09	0,91	40c21	34,96	0,98	60c17	24,86	0,83
20c12	16,43	0,95	40c22	36,32	0,99	60c18	27,70	0,86
20c13	17,59	0,97	40c23	37,38	0,99	60c19	30,49	0,89
20c14	18,48	0,98	40c24	38,19	1,00	60c20	33,48	0,91
20c15	19,11	0,99	50c02	3,06	0,05	60c21	36,59	0,93
20c16	19,54	1,00	50c03	3,51	0,09	60c22	39,19	0,95
30c02	3,01	0,08	50c04	4,26	0,13	60c23	41,98	0,96
30c03	3,58	0,14	50c05	5,16	0,19	60c24	44,86	0,97
30c04	4,42	0,21	50c06	6,11	0,25	60c25	47,38	0,98
30c05	5,46	0,30	50c07	7,13	0,32	60c26	49,60	0,98
30c06	6,62	0,39	50c08	8,26	0,39	60c27	51,51	0,99
30c07	7,96	0,48	50c09	9,55	0,46	60c28	53,27	0,99
30c08	9,46	0,57	50c10	10,98	0,53	60c29	54,88	0,99
30c09	11,07	0,65	50c11	12,64	0,60	60c30	56,08	1,00
30c10	12,99	0,73	50c12	14,53	0,66	70c02	3,07	0,04
30c11	14,92	0,79	50c13	16,64	0,72	70c03	3,50	0,06
30c12	16,97	0,84	50c14	18,95	0,77	70c04	4,20	0,10
30c13	19,02	0,89	50c15	21,28	0,81	70c05	5,07	0,14
30c14	20,94	0,92	50c16	23,82	0,85	70c06	5,90	0,19
30c15	22,79	0,94	50c17	26,58	0,88	70c07	6,78	0,24
30c16	24,52	0,96	50c18	29,22	0,91	70c08	7,73	0,29
30c17	25,99	0,98	50c19	31,88	0,93	70c09	8,71	0,35
30c18	27,12	0,99	50c20	34,48	0,95	70c10	9,86	0,41
30c19	28,07	0,99	50c21	37,04	0,96	70c11	11,12	0,47
30c20	28,76	0,99	50c22	39,13	0,97	70c12	12,57	0,53
30c21	29,26	1,00	50c23	41,43	0,98	70c13	14,20	0,59
40c02	3,03	0,06	50c24	43,18	0,99	70c14	16,07	0,64

Tabela 1 Tamanho do clique maximal e porcentagem de arestas encontrados em cada caso de teste.

Caso	m_clique	%arestas	Caso	m_clique	%arestas	Caso	m_clique	%arestas
70c15	18,13	0,69	80c29	59,41	0,98	100c04	4,19	0,07
70c16	20,38	0,73	80c30	62,44	0,98	100c05	4,97	0,10
70c17	22,89	0,77	80c31	65,11	0,99	100c06	5,80	0,13
70c18	25,71	0,81	80c32	67,38	0,99	100c07	6,64	0,17
70c19	28,62	0,85	80c33	69,69	0,99	100c08	7,45	0,22
70c20	31,66	0,87	80c34	71,73	0,99	100c09	8,28	0,26
70c21	34,60	0,90	80c35	73,37	1,00	100c10	9,06	0,31
70c22	38,02	0,92	90c02	3,11	0,03	100c11	10,00	0,36
70c23	41,21	0,93	90c03	3,50	0,05	100c12	11,01	0,41
70c24	44,24	0,95	90c04	4,18	0,08	100c13	12,14	0,46
70c25	47,53	0,96	90c05	5,00	0,11	100c14	13,47	0,51
70c26	50,48	0,97	90c06	5,83	0,15	100c15	14,95	0,56
70c27	53,19	0,98	90c07	6,67	0,19	100c16	16,72	0,60
70c28	55,91	0,98	90c08	7,51	0,24	100c17	18,62	0,65
70c29	58,22	0,99	90c09	8,33	0,29	100c18	20,73	0,69
70c30	60,41	0,99	90c10	9,22	0,34	100c19	23,08	0,73
70c31	62,43	0,99	90c11	10,24	0,39	100c20	25,60	0,76
70c32	64,00	1,00	90c12	11,42	0,44	100c21	28,36	0,79
80c02	3,11	0,03	90c13	12,70	0,50	100c22	31,44	0,82
80c03	3,49	0,05	90c14	14,16	0,55	100c23	34,43	0,85
80c04	4,22	0,08	90c15	15,77	0,59	100c24	37,84	0,87
80c05	5,04	0,12	90c16	17,76	0,64	100c25	41,43	0,89
80c06	5,85	0,16	90c17	19,84	0,69	100c26	45,20	0,91
80c07	6,71	0,21	90c18	22,11	0,72	100c27	49,07	0,92
80c08	7,58	0,26	90c19	24,71	0,76	100c28	52,87	0,94
80c09	8,49	0,32	90c20	27,41	0,80	100c29	56,98	0,95
80c10	9,51	0,37	90c21	30,45	0,83	100c30	60,88	0,96
80c11	10,61	0,43	90c22	33,49	0,85	100c31	64,57	0,97
80c12	11,89	0,48	90c23	36,73	0,88	100c32	68,41	0,97
80c13	13,37	0,54	90c24	40,35	0,90	100c33	71,92	0,98
80c14	15,04	0,59	90c25	44,01	0,92	100c34	75,82	0,98
80c15	16,95	0,64	90c26	47,69	0,93	100c35	78,74	0,99
80c16	18,98	0,69	90c27	51,40	0,94	100c36	81,74	0,99
80c17	21,24	0,73	90c28	55,16	0,96	100c37	84,51	0,99
80c18	23,74	0,77	90c29	58,74	0,96	100c38	87,05	0,99
80c19	26,54	0,80	90c30	62,26	0,97	100c39	89,32	1,00
80c20	29,46	0,83	90c31	65,72	0,98	110c02	3,13	0,02
80c21	32,42	0,86	90c32	68,93	0,98	110c03	3,48	0,04
80c22	35,83	0,89	90c33	71,79	0,99	110c04	4,18	0,06
80c23	39,12	0,91	90c34	74,59	0,99	110c05	4,97	0,09
80c24	42,64	0,92	90c35	77,13	0,99	110c06	5,81	0,12
80c25	46,35	0,94	90c36	79,43	0,99	110c07	6,61	0,16
80c26	49,50	0,95	90c37	81,54	1,00	110c08	7,38	0,20
80c27	52,98	0,96	100c02	3,12	0,03	110c09	8,20	0,24
80c28	56,23	0,97	100c03	3,50	0,04	110c10	8,98	0,29

Tabela 1 (cont1.) Tamanho do clique maximal e porcentagem de arestas encontrados em cada caso de teste.

Caso	m_clique	%arestas	Caso	m_clique	%arestas	Caso	m_clique	%arestas
110c11	9,83	0,33	120c16	15,08	0,53	130c19	19,30	0,63
110c12	10,76	0,38	120c17	16,64	0,58	130c20	21,24	0,66
110c13	11,76	0,43	120c18	18,35	0,62	130c21	23,48	0,70
110c14	12,89	0,47	120c19	20,33	0,66	130c22	25,80	0,73
110c15	14,28	0,52	120c20	22,50	0,69	130c23	28,54	0,76
110c16	15,77	0,57	120c21	24,88	0,73	130c24	31,30	0,79
110c17	17,54	0,61	120c22	27,53	0,76	130c25	34,41	0,81
110c18	19,50	0,65	120c23	30,37	0,79	130c26	37,80	0,84
110c19	21,56	0,69	120c24	33,36	0,82	130c27	41,21	0,86
110c20	23,90	0,73	120c25	36,76	0,84	130c28	45,06	0,88
110c21	26,48	0,76	120c26	40,08	0,86	130c29	48,84	0,90
110c22	29,27	0,79	120c27	43,82	0,88	130c30	53,26	0,91
110c23	32,24	0,82	120c28	47,65	0,90	130c31	57,45	0,93
110c24	35,57	0,84	120c29	51,86	0,92	130c32	61,66	0,94
110c25	39,03	0,87	120c30	55,91	0,93	130c33	66,39	0,95
110c26	42,63	0,89	120c31	60,02	0,94	130c34	71,13	0,96
110c27	46,47	0,90	120c32	64,49	0,95	130c35	75,11	0,96
110c28	50,51	0,92	120c33	68,83	0,96	130c36	79,89	0,97
110c29	54,62	0,93	120c34	73,10	0,97	130c37	84,16	0,97
110c30	58,49	0,94	120c35	77,63	0,97	130c38	88,50	0,98
110c31	62,67	0,95	120c36	81,57	0,98	130c39	92,83	0,98
110c32	66,75	0,96	120c37	85,74	0,98	130c40	96,90	0,99
110c33	70,88	0,97	120c38	89,39	0,99	130c41	100,52	0,99
110c34	74,97	0,98	120c39	93,04	0,99	130c42	104,34	0,99
110c35	78,65	0,98	120c40	96,70	0,99	130c43	107,65	0,99
110c36	82,43	0,98	120c41	99,78	0,99	130c44	110,96	0,99
110c37	85,77	0,99	120c42	102,82	0,99	130c45	113,76	1,00
110c38	89,03	0,99	120c43	105,39	1,00	140c02	3,15	0,02
110c39	92,25	0,99	130c02	3,14	0,02	140c03	3,49	0,03
110c40	94,68	0,99	130c03	3,51	0,03	140c04	4,20	0,05
110c41	96,94	1,00	130c04	4,19	0,05	140c05	4,97	0,07
120c02	3,15	0,02	130c05	4,97	0,08	140c06	5,80	0,10
120c03	3,52	0,04	130c06	5,79	0,10	140c07	6,60	0,13
120c04	4,20	0,06	130c07	6,61	0,14	140c08	7,39	0,16
120c05	4,95	0,08	130c08	7,38	0,17	140c09	8,18	0,19
120c06	5,78	0,11	130c09	8,12	0,21	140c10	8,90	0,23
120c07	6,59	0,15	130c10	8,90	0,25	140c11	9,56	0,27
120c08	7,40	0,18	130c11	9,61	0,29	140c12	10,33	0,31
120c09	8,16	0,22	130c12	10,40	0,33	140c13	11,02	0,35
120c10	8,90	0,27	130c13	11,27	0,38	140c14	11,92	0,40
120c11	9,66	0,31	130c14	12,16	0,42	140c15	12,92	0,44
120c12	10,55	0,35	130c15	13,28	0,46	140c16	14,00	0,48
120c13	11,47	0,40	130c16	14,46	0,51	140c17	15,34	0,52
120c14	12,56	0,45	130c17	15,93	0,55	140c18	16,69	0,56
120c15	13,74	0,49	130c18	17,42	0,59	140c19	18,28	0,60

Tabela 1 (cont2.) Tamanho do clique maximal e porcentagem de arestas encontrados em cada caso de teste.

<i>n</i>	<i>Pósa</i>	<i>Kučera</i>	<i>Bollobás</i>	<i>n</i>	<i>Pósa</i>	<i>Kučera</i>	<i>Bollobás</i>
10	0,22	3,16	6,65	110	0,04	14,99	13,58
20	0,14	5,10	8,65	120	0,03	15,80	13,81
30	0,10	6,66	9,82	130	0,03	16,58	14,06
40	0,08	8,01	10,66	140	0,03	17,33	14,26
50	0,07	9,22	11,30	150	0,03	18,07	14,46
60	0,06	10,33	11,83	160	0,03	18,78	14,65
70	0,05	11,36	12,28	170	0,03	19,47	14,83
80	0,05	12,34	12,64	180	0,03	20,15	14,99
90	0,04	13,26	13,00	190	0,02	20,81	15,15
100	0,04	14,14	13,29	200	0,02	21,45	15,30

Tabela 2 Respectivamente valores de $2x \frac{n \log n}{n(n-1)}$, $\sqrt{n \log n}$ e $2 \log_2 n$.

À título de comparação, na **Tabela 2**, são apresentados algumas estimativas teóricas em relação ao número de arestas e o tamanho do clique maximal. O campo titulado “*Pósa*” mostra o valor $2x \frac{n \log n}{n(n-1)}$; lembramos que em *Pósa* (76) é demonstrado que é altamente provável que encontremos um circuito hamiltoniano, à medida que o número de vértices tende a infinito, em um grafo com $\delta n \log n$ arestas, dada uma constante δ (no caso da **Tabela 2**, $\delta=1$). No campo denominado “*Kučera*”, temos o valor de referência para o clique maximal $\sqrt{n \log n}$, onde, segundo *Kučera*(95), é esperado que o clique maximal seja encontrado com facilidade olhando apenas para os vértices de maior grau, exatamente o que a nossa busca gulosa linear efetua. Por fim, o campo intitulado “*Bollobás*” temos o valor $2 \log_2 n$, para ter uma ideia do tamanho do clique maximal em grafos aleatórios. Em *Bollobás* (85), o clique maximal esperado em grafos randômicos de n vértices, onde cada aresta é gerada, de forma randômica e independente, com uma probabilidade igual a $\frac{1}{2}$, tenderá para $\lceil r(n) \rceil$ ou $\lfloor r(n) \rfloor$, para $r(n) = (2+o(1)) \log_2 n$, com uma probabilidade que se aproxima de 1 quando n tende a infinito. Os valores de referência “*Kučera*” e “*Bollobás*” são estimativas para o valor *exato* do clique maximal. Os dados que geramos são uma aproximação, já que o valor exato

demandaria um tempo longo para o volume de testes proposto. Por último, alertamos que estes valores são colocados apenas a título de referência, para maiores conclusões quanto às constantes multiplicativas será necessário um estudo mais teórico dos grafos MOSP.

Pelos dados da **Tabela 1**, podemos verificar que, nos testes efetuados, o número de arestas supera em quase a totalidade dos casos o valor $n \log n$, apresentado pela coluna “*Pósa*”, na **Tabela 2**. Somente nos casos de teste onde o valor C é muito pequeno (2 ou 3) este valor não é superado pela média encontrada nos grafos MOSP. Isto já era de se esperar, pois, na média, cada padrão irá gerar um clique com $\frac{1+C}{2}$ vértices, ou seja $\frac{C^2-1}{8}$ arestas.

Se formos dividir o número de arestas esperado para n padrões pelo valor $\delta n \log n$, teremos

$\frac{C^2-1}{8\delta \log n}$. Mesmo subtraindo os arcos múltiplos, esta proporção tende rapidamente a 1. Este

resultado pode ser interessante, caso circuitos hamiltonianos possam ser utilizados no sequenciamento das peças, já que com grande probabilidade, a medida em que cresce o número de vértices, encontraremos um circuito hamiltoniano em grafos MOSP utilizando o algoritmo extensão-rotação descrito na Seção 3.4.3. Poderíamos objetar que talvez o fator constante $\delta=1$ seja insuficiente para que o algoritmo funcione favoravelmente. Por exemplo, em Koršunov (76) é demonstrado serem necessárias $3n \log n$ arestas ($\delta=3$). Na **Figura 14** podemos comparar as densidades relativas aos números de arestas $n \log n$, $3n \log n$ e das médias encontradas nos grafos MOSP aleatórios com até 5 e 6 peças em cada padrão. Como podemos observar, a maior parte dos grafos MOSP parece gerar, na média, mais arestas que o valor indicado em Koršunov (76). Note que não é difícil termos mais de 6 peças por padrão, o que implica, pelo menos em teoria, um funcionamento razoável de heurísticas baseadas no algoritmo extensão-rotação.

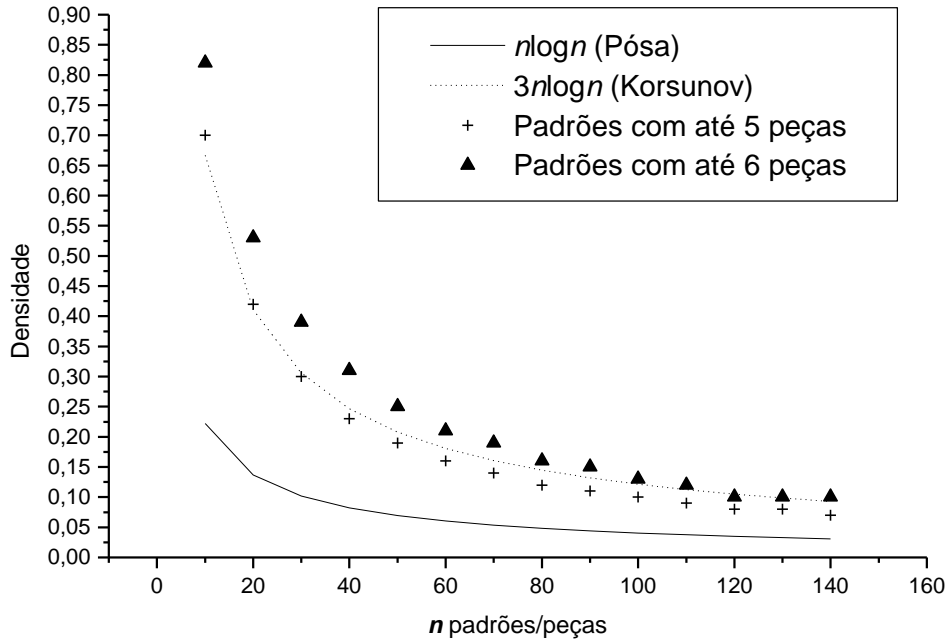


Figura 14 Comparação entre as densidades relativas à $n\log n$, $3n\log n$ e das médias encontradas em grafos MOSP gerados com até 5 e 6 peças por padrão.

Uma vez analisado o valor médio do número de arestas, passemos ao valor do clique maximal encontrado. Lembramos, novamente, que o valor do clique maximal apresentado nesta seção não é exato, sendo, portanto, subestimado. Pela **Tabela 1** calculamos o valor mínimo de C necessário para superar $\varepsilon\sqrt{n\log n}$, para $\varepsilon = 0.5, 1, 2, 3$ com o resultado apresentado na **Tabela 3**. Podemos concluir que, pelo menos nos nossos testes, com poucas peças permitidas em cada padrão alcançamos o clique da ordem apresentada em Kučera (95). É, portanto de se esperar que consigamos uma boa aproximação no clique maximal encontrado em grafos MOSP. Pelos dados apresentados, também podemos prever que a aproximação será menor nos casos MOSP esparsos, *e.g.* $100c05$ ou $100c08$, o que se confirmará nos testes computacionais.

n	$\varepsilon = 0.5$	$\varepsilon = 1$	$\varepsilon = 2$	$\varepsilon = 3$
10	2	3	6	9
20	2	5	8	12
30	3	6	11	14
40	4	8	13	16
50	5	9	14	18
60	6	10	15	19
70	6	11	17	21
80	7	13	19	23
90	7	14	20	24
100	8	15	21	26
110	9	16	22	27
120	9	17	23	28
130	10	18	25	30
140	10	19	26	31

Tabela 3 Valores de C necessários para superar $\varepsilon\sqrt{n\log n}$

Através do campo “*Bollobás*” podemos ter uma ideia, nos casos testados, de como até a aproximação encontrada do clique maximal MOSP é superior ao que é esperado em grafos randômicos com mesmo número de vértices e de como este valor tende a crescer rapidamente à medida em que cresce a entrada n . A **Tabela 4** possui apenas o intuito de mostrar que os cliques MOSP são grandes, sem as consequências algorítmicas dos trabalhos de Pósa (76) e Kučera (95), desta forma não iremos avaliar várias constantes multiplicativas. Por último, podemos observar que, como $2\log_2 n$ vai se tornando bem menor que $\sqrt{n\log n}$ a medida em que n cresce, teremos valores inferiores de C , por exemplo, para $n > 100$ e $\varepsilon = 1$, em comparação com a **Tabela 3**.

n	$2\log n$	n	$2\log n$
10	6	80	13
20	8	90	14
30	9	100	14
40	10	110	15
50	11	120	15
60	11	130	16
70	12	140	17

Tabela 4 Valores de C necessários para superar $2\log_2 n$

Poderíamos apresentar outras informações que obtivemos nos nossos testes, como, por exemplo, o tamanho do segundo e terceiro clique maximais encontrados. Estes cliques são obtidos após a contração do clique atual encontrado em um hipernó, ao procurarmos o novo clique maximal no hipergrafo induzido. Como poderíamos esperar, eles se apresentaram relativamente grandes, se comparados com grafos randômicos de igual número de vértices. Este fato motivou a heurística de contração recursiva de cliques que será descrita na Seção 4.3.2 do presente capítulo. Porém omitiremos estes dados para não sobrecarregar esta dissertação. O que é interessante é o fato de, pelo menos nos nossos testes realizados, se comprovar a constituição bastante peculiar dos grafos MOSP quanto ao tamanho do clique maximal e o número de arestas, característica que irá nos ajudar nas heurísticas desenvolvidas.

4.2 Método frugal sugerido: iniciar sempre pelo clique maximal encontrado pela busca gulosa por vértice de grau mínimo no grafo complementar

Como observamos nos testes realizados, o tamanho do clique maximal encontrado aumenta rapidamente em função do número de peças permitido em cada padrão. Na **Figura 15** podemos ver este crescimento nos casos de teste para 100 padrões de corte. Como exemplo, para 27 peças permitidas por padrão, o clique maximal, encontrado pela nossa aproximação, já alcança, na média dos 2000 testes aleatórios realizados, o valor 50, ou seja, metade do número de peças.

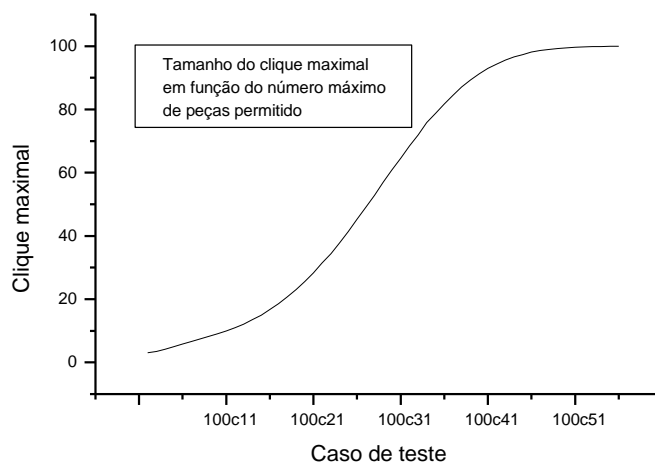


Figura 15 Médias dos cliques maximais encontrados para 100 padrões de corte.

Como foi visto anteriormente, a busca gulosa por vértice de grau mínimo consegue detectar quase sempre o clique maximal MOSP. É natural começar sempre o sequenciamento de padrões de corte olhando para as peças formadas pelo clique maximal encontrado. Como o grafo MOSP estruturalmente é um caso em que a busca gulosa citada funciona bem, estamos sugerindo, como um método frugal para elaboração de uma classe de heurísticas, começar

sempre o sequenciamento a partir dos padrões associados às peças associadas ao clique maximal. Para encontrar este clique maximal sugerimos a busca gulosa por vértice de grau mínimo no grafo complementar. Nela, um vértice de grau mínimo é removido, junto com todos os seus vizinhos, e o processo é repetido iterativamente até que o grafo seja vazio, como descrito a seguir.

Busca Gulosa por vértice de grau mínimo

Entrada: Grafo complementar MOSP \overline{G}

Saída: Clique idealmente maximal $S = \{v_1, v_2, \dots, v_k\}$

```

 $S = \{ \}$ 
while  $\overline{G} \neq \emptyset$  {
     $v = \text{vértice de grau mínimo em } \overline{G}$ 
     $S = S \cup \{ v \}$ 
     $\overline{G} = \overline{G} - (\{ v \} \cup \text{vizinhos}( v ))$ 
}

Retorne  $S$ 

```

A seguir a **Figura 16** mostra um exemplo de como calculamos o clique maximal no grafo MOSP a partir da busca gulosa no grafo complementar. Os vértices seleccionados aparecem em destaque e compõe o clique maximal $\{1, 2, 3, 4\}$ de tamanho 4, no grafo original. Caso na última iteração fosse escolhido o vértice 5, também detectaríamos outro clique maximal $\{1, 2, 3, 5\}$. Podemos reparar como há uma rápida diminuição do número de vértices e arestas a cada passo do algoritmo, o que torna esta heurística muito rápida para o grafo complementar MOSP, *i.e.* um grafo complementar esparso.

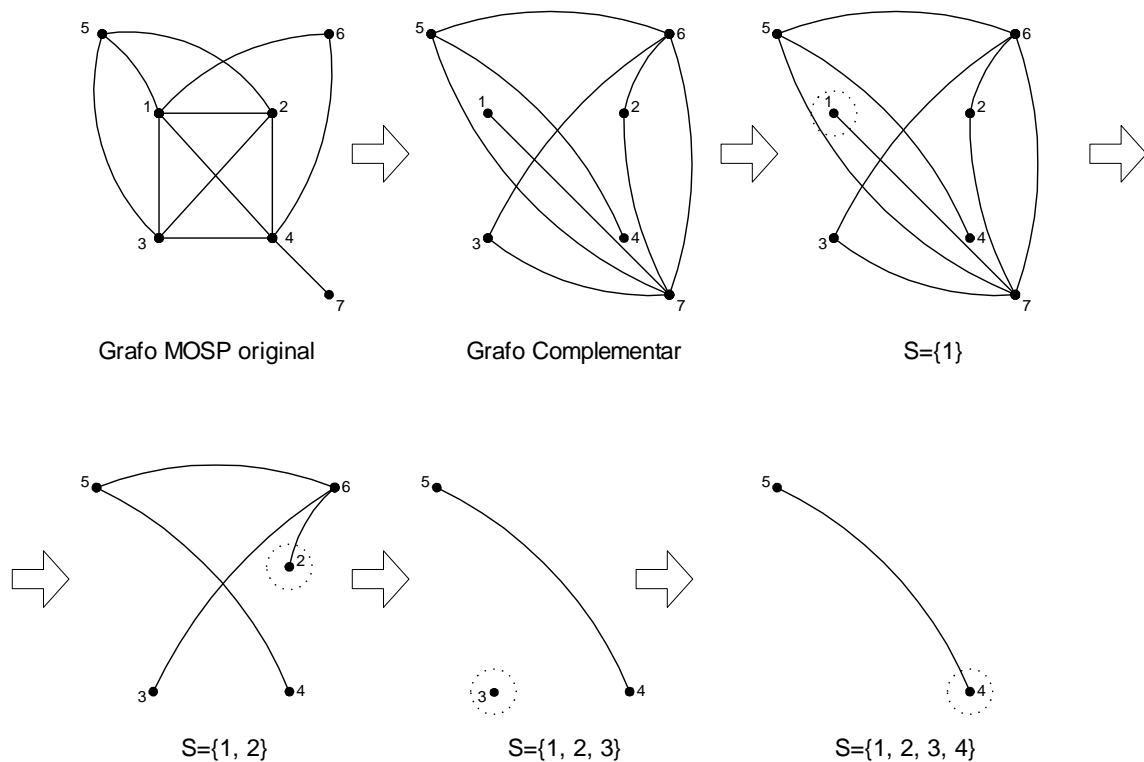


Figura 16 Determinação do Clique Maximal via conjuntos independentes maximais no grafo complementar.

Do ponto de vista da complexidade, Halldórson (97) mostra como a busca gulosa pode ser implementável em tempo linear no tamanho do grafo, o que parece ser confirmado pelos nossos testes computacionais, como podemos exemplificar através da **Tabela 5**. As médias são relativas a 2000 casos aleatórios gerados para uma densidade de 50% dos grafos correspondentes.

<i>n</i>	<i>tempo(s)</i>	<i>n</i>	<i>tempo(s)</i>
100	0,0005	600	0,0187
200	0,0019	700	0,0254
300	0,0044	800	0,0333
400	0,0081	900	0,0431
500	0,0131	1000	0,0519

Tabela 5 Tempos médios para a busca gulosa detectar o clique maximal.

Testamos ainda uma busca gulosa randômica, não escolhendo o vértice de grau mínimo, porém um vértice qualquer. Apesar desta busca gulosa randômica necessitar de apenas $\log_2 n$ passos, *cf.* Halldórson (91), o clique maximal encontrado teve pior qualidade, na média dos testes executados.

Vamos verificar agora alguns exemplos de piores casos para a abordagem proposta. Podemos visualizar na **Figura 17**, parte de um grafo *complementar* MOSP. Caso o primeiro vértice selecionado pela nossa busca gulosa for o que está representado mais à esquerda, retornaremos o conjunto representado pelos vértices em negrito, quando na verdade o maior conjunto independente maximal é o visualizado com vértices vazados. Neste caso não é encontrado o clique maximal no grafo original, mas um clique bem menor no seu lugar. Na verdade, este exemplo faz parte de uma família de piores casos, *cf.* Halldórson (97), onde uma cadeia de repetições de subgrafos, cada um deles composto por um clique totalmente conectado com um conjunto independente, é montada.

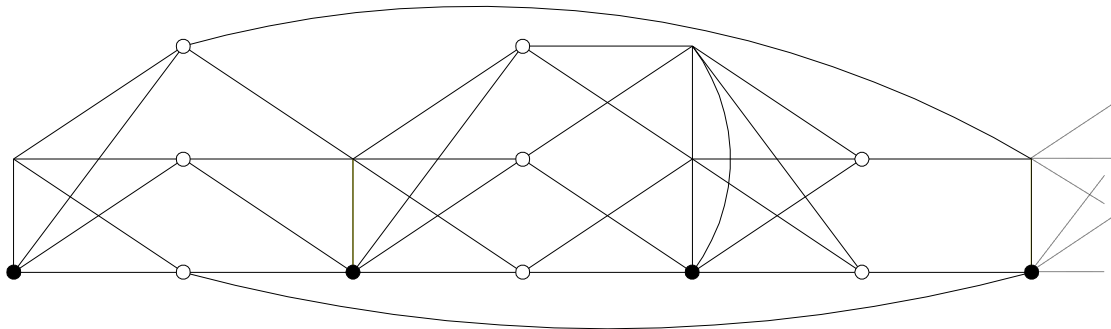


Figura 17 Parte inicial de um grafo complementar MOSP, que é um exemplo de pior caso para a busca gulosa por vértice de grau mínimo para detecção de conjuntos independentes maximais.

O segundo exemplo é retirado de Johnson (74), onde visualizamos na **Figura 18** um grafo MOSP, com $2k+1$ vértices, k deles formando um clique de tamanho k , e um vértice P_0 conectado com k vértices $\{P_1, P_2, \dots, P_k\}$, distintos dos que formam o k -clique. No grafo

complementar a busca gulosa irá escolher o vértice P_0 , encontrando um clique de tamanho 2, ao invés do k -clique.

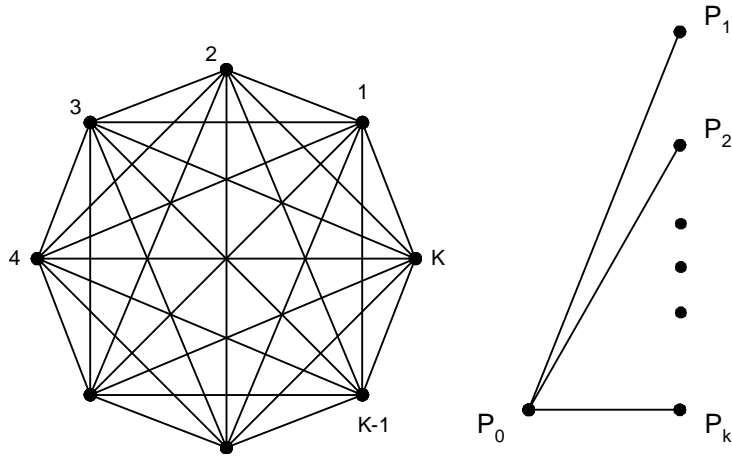


Figura 18 Pior caso citado por Johnson (74) de grafo MOSP para a busca gulosa.

Na prática o pior caso Johnson (74), intuitivamente, deve ser muito raro em grafos MOSP gerados a partir de casos reais da indústria. A situação seria algo como padrões de corte da indústria do papelão – grafo esparsos com $k+1$ vértices – concomitantes com padrões com muitas peças – k -clique – num *mesmo* sequenciamento. Nos nossos testes aleatórios este caso é impossível, já que os grafos MOSP gerados são conexos pelo fato de não permitirmos padrões de uma peça única e colunas vazias. O pior caso citado em Halldórson (97) pode ser evitado escolhendo-se convenientemente mais de um vértice inicial, mas não implementaremos esta otimização.

Uma dúvida, legítima, que pode surgir, é porque optamos por utilizar a busca gulosa no grafo complementar ao invés do dual no grafo original, já que conceitualmente são um único e mesmo algoritmo. No grafo original, o dual da busca gulosa iria escolher o vértice de *maior* grau, incluir o mesmo no conjunto solução, e repetir o processo recursivamente para o conjunto de seus vizinhos. O algoritmo *dual* é resumido a seguir.

Dual da Busca Gulosa adotada no MOSP

Entrada: Grafo original MOSP G

Saída: Clique idealmente maximal $S = \{v_1, v_2, \dots, v_k\}$.

```
S = {}  
while G ≠ ∅ {  
    v = vértice de grau máximo em G  
    S = S ∪ { v }  
    G = {vizinhos de v}  
}  
  
Retorne S
```

Alguns motivos que nos levaram a optar pela busca gulosa no grafo complementar:

a) Cronologicamente encontramos primeiro artigos mais significativos sobre a busca gulosa para conjuntos independentes maximais.

b) Em Halldórson (97) é sugerida uma generalização para a busca gulosa, onde em lugar do vértice de grau mínimo, é escolhida uma classe de vértices, cujo grau é pelo menos a média dos graus dos seus vizinhos, isto permite uma *paralelização* da busca, aspecto interessante para um estudo futuro.

c) Johnson (74) afirma que a busca gulosa no grafo original é implementável em $O(n \log n)$, enquanto o Halldórson (97) já apresenta uma implementação linear no tamanho do grafo da busca gulosa para conjuntos independentes maximais.

Para finalizar podemos tecer algumas considerações. No trabalho Yanasse, Becceneri e Soma (99) já era alertada a importância do clique maximal no sequenciamento. A ideia apresentada aqui, portanto, não é original, apenas o *modo* como estamos efetivando-a, de forma tentativa, esperando que seja mais rápida. Um outro ponto importante é de que a heurística **ybs99**, de contração de arcos, pode encontrar um hipergrafo completo *maior* que o clique maximal. Podemos entender como isto é possível pela **Figura 19**, onde o clique

maximal $\{1, 2, 3, 4\}$ é de tamanho 4, enquanto a contração do arco $\{5, 6\}$ produz o clique $\{1, 2, 3, 4, 7\}$ de tamanho 5. A heurística **ybs99**, consequentemente, deve produzir, em média, um número máximo de pilhas abertas menor que as heurísticas que iniciem pela busca gulosa por vértice de grau mínimo, já que possui a possibilidade de começar com um hipergrafo completo maior. De fato, como verificaremos na Seção 4.4.1 adiante, a heurística **ybs99** irá obter uma qualidade melhor no número máximo de pilhas abertas no sequenciamento de padrões de corte, sacrificando para isso o tempo de computação, dada que a contração de arcos é computacionalmente custosa de ser processada.

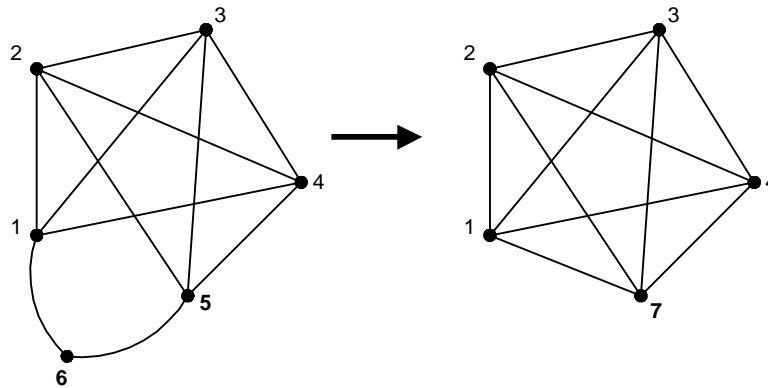


Figura 19 Exemplo de como a contração de arcos executada pela heurística **ybs99** pode produzir hipergrafos completos maiores que o clique maximal.

4.3 Descrição das heurísticas desenvolvidas através do método frugal sugerido

As ideias apresentadas até agora, para a resolução do MOSP, associam o problema a um grafo, com características especiais, e este à determinação do clique maximal, via conjunto independente, a aquele associado. Faz-se mister notar, que a menos do caso trivial, o problema ainda está por ser resolvido, no senso que, a totalidade dos padrões deve ser, ainda, sequenciada, *i.e.*, alguns vértices (peças no problema original) devem ser considerados no sequenciamento final dos padrões.

Para avaliar o método geral sugerido – começar o sequenciamento sempre pelo clique maximal encontrado pela busca gulosa por vértice de grau mínimo para posteriormente sequenciar aos vértices remanescentes – testamos computacionalmente várias implementações de possíveis heurísticas. Inicialmente modelamos o MOSP como um caixeiro viajante, *i.e.*, percorrimento dos vértices do grafo MOSP para sequenciar peças, onde os custos associados são os graus médios dos vértices, utilizando para a resolução do mesmo uma heurística de inserção de arcos, no caso a de maior distância (*farthest insertion*), *cf.* Syslo *et al.* (83) e Johnson e McGeoch (97). Concomitantemente implementamos a heurística de contração recursiva de cliques. Os resultados destas duas heurísticas podem ser encontrados em Ashikaga *et al.* (00). Posteriormente foi implementada uma versão da heurística via caixeiro viajante levando em conta os arcos múltiplos, obtendo resultados melhores em erro médio em relação às duas versões anteriores. Em seguida, outra ideia testada foi implementar heurísticas para árvores geradoras mínimas, a partir do clique maximal, já contraído, também levando em conta os arcos múltiplos. Novamente foi obtida uma melhora na aproximação – tempo e erro – em relação a todas as heurísticas anteriores.

Uma implementação recente, via circuitos hamiltonianos, acabou fornecendo nosso melhor resultado, tanto em termos de erro médio, como em relação ao tempo. Desta maneira, abandonamos momentaneamente as heurísticas anteriores para realizar testes computacionais mais exaustivos e concentrar esforços tendo em vista a dissertação. Sendo esta, porquanto, a primeira heurística a ser descrita. A segunda heurística escolhida foi a de contração recursiva de cliques. Ela foi implementada e testada na mesma época da primeira versão da heurística via caixeiro viajante. Ela será nossa segunda heurística a ser descrita na dissertação dada sua ideia central, pois como o clique estruturalmente é o relacionamento entre vários padrões, é razoável pensar no sequenciamento deles pela contração sucessiva de cliques, idealmente

maximais. Estão sendo iniciados testes em uma outra heurística, onde não fazemos uma contração de cada clique, operação computacionalmente pesada, como veremos. Nesta nova heurística, que não será descrita na presente dissertação, sequenciamos simplesmente os cliques idealmente maximais, sem contraí-los, obtendo resultados bem melhores em tempo e erro médios que a contração recursiva.

Uma dúvida que tivemos foi pensar se *qualquer* método de sequenciamento dos vértices produziria um resultado razoável, desde que o sequenciamento começasse pelo clique maximal. A diferença de erro médio e principalmente do tempo médio de resolução da heurística via circuitos hamiltonianos nos mostra que isto não era verdade. Também, implementamos outras heurísticas rápidas, como por exemplo, uma ordenação simples dos vértices pelo grau, após o clique maximal e o resultado foi ruim em termos de erro.

4.3.1 Heurística Extensão-Rotação (*her*)

Nesta heurística proposta, em primeiro lugar, o caminho inicial não será apenas um vértice isolado, mas o *conjunto dos vértices* que compõe o clique maximal. Para o sequenciamento dos vértices resultantes, via circuitos hamiltonianos, optou-se por uma modificação do algoritmo Extensão-Rotação, o qual foi apresentado na Seção 3.4.3. Note-se que em um clique a detecção de caminhos é executada de maneira trivial. A ideia é fazer as extensões a partir deste caminho inicial, economizando um número significativo de passos, dado o tamanho do clique maximal. Em segundo lugar, iremos restringir o número de rotações de duas formas: não faremos mais rotações que uma constante C_r e somente faremos rotações se ainda não tivermos feito C_e extensões. As rotações permitirão que as extensões consigam sair de “becos sem saída”, que sejam *próximos* do clique maximal. Os nossos testes computacionais nos mostraram que, para uma grande maioria de grafos MOSP, bastam

poucas rotações para que grande parte dos vértices seja percorrida. Nos nossos testes, constatamos de maneira prática, que $C_r = 2$ e $C_e = \min(n/10, 5)$. Estamos efetuando estudos sobre a influência do número de rotações permitido na aproximação obtida por **her**. Um fato observado após diversos experimentos é que em muitos casos basta apenas *uma* rotação no caminho formado pelos vértices do clique maximal para que seja obtida uma boa aproximação. Por último, iremos tornar as extensões determinísticas, procurando o vértice de maior grau para a extensão, visando escolher as peças com mais padrões associados para o sequenciamento. A heurística **her** é descrita em seguida, em um dialeto (*pidin*) à linguagem C, nos seus principais passos.

Heurística Extensão-Rotação (*her*)

Entrada: Grafo MOSP $G(E, V)$.

Saída: Sequência $Spa = \{pa_1, pa_2, \dots, pa_n\}$ ordenada de *padrões*.

```

Spe = {vértices do clique maximal}
while (true) {
     $x_k = \{\text{vizinho de maior grau do último vértice em } Spe\}$ 
    if  $x_k \neq \emptyset$ 
         $Spe = Spe \cup \{x_k\}$ 
         $C_e = C_e + 1$ 
        continue; // volta ao while (true)
     $j = \text{vértice\_para\_rotação}()$ 
    if  $j \neq \emptyset$ 
        transforme_Spe_por_rotação( $j$ ); //  $C_r = C_r + 1$ 
    else
        break; // sai do while (true)
}
otimiza_sequência_de_peças ();
sequenciamento_dos_vértices_que_sobraram ();
Spa = sequência_de_padrões_associada ();
Retorne Spa;

```

A sub-rotina *vértice_para_rotação* () procura tomar os seguintes cuidados ao escolher um vértice: não escolher o vizinho à esquerda em *Spe* (a rotação não teria efeito); não escolher o último vértice escolhido na rotação anterior; não permitir mais que C_r rotações e não permitir rotações caso já tivermos feito C_e extensões.

Devido às limitações impostas por C_r e C_e , *i.e.*, caso não tenha sido possível percorrer todos os vértices do grafo, efetuamos o sequenciamento a partir das duas rotinas restantes. Em *otimiza_sequência_de_peças* () procuramos um tipo especial de vértice que denominamos *conjugado*, que são os vértices restantes e que possuem todos seus vizinhos já sequenciados de forma contígua. Desta forma, este tipo de vértice deixa várias pilhas associadas aos vizinhos abertas, quando poderiam estar fechadas, caso o conjugado já houvesse sido sequenciado.

Na **Figura 20**, uma possível sequência de peças no grafo MOSP exemplificado seria $Spe=\{7, 2, 6, 9, 4, 12, 10, 8, 11, 13, 1\}$. Os vértices que sobraram neste exemplo são $\{5, 3, 14, 15\}$. São conjugados os vértices 5 e 3, que possuem todos os vizinhos na sequência Spe parcialmente construída de forma contígua. Não são conjugados os vértices 14 e 15. A nossa rotina *otimiza_sequência_de_peças* () irá colocar os vértices 5 e 3 antes dos seus vizinhos, tornando a sequência $Spe=\{3, 7, 2, 6, 5, 9, 4, 12, 10, 8, 11, 13, 1\}$. Por fim, os vértices restantes são sequenciados em ordem decrescente de grau na rotina *sequenciamento_dos_vértices_que_sobraram* (). Podemos prever que em grafos MOSP muito esparsos, a porcentagem de vértices que irá sobrar para esta última rotina será maior, como veremos nos testes.

A rotina *sequência_de_padrões_associada* () transforma a sequência de fechamento de peças na sequência ordenada de padrões da seguinte forma: percorre-se a sequência de peças do final para o início, colocando-se os padrões que contenham a peça atual do fim para o começo na sequência de padrões.

Na **Figura 21** temos um exemplo de como funciona a heurística *her*. O clique maximal encontrado inicia a sequência de peças $Spe = \{2, 8, 3, 4, 10\}$. É realizada uma rotação modificando a sequência para $Spe=\{2, 8, 10, 4, 3\}$, a partir de onde foram possíveis

mais 3 extensões. O vértice 1 é conjugado, sendo inserido antes dos vértices 2 e 8. Por último, restou o vértice 9, que foi inserido no final. A sequência de padrões associada, $Spa = \{4, 2, 8, 9, 6, 10, 7, 3, 1, 5\}$ abre no máximo 6 pilhas durante o seu processamento.

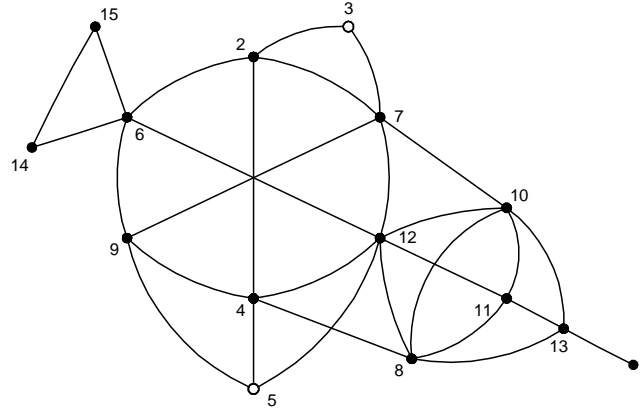


Figura 20 Os vértices 3 e 5 são conjugados em relação à sequência parcial $Spe=\{7, 2, 6, 9, 4, 12, 10, 8, 11, 13, 1\}$.

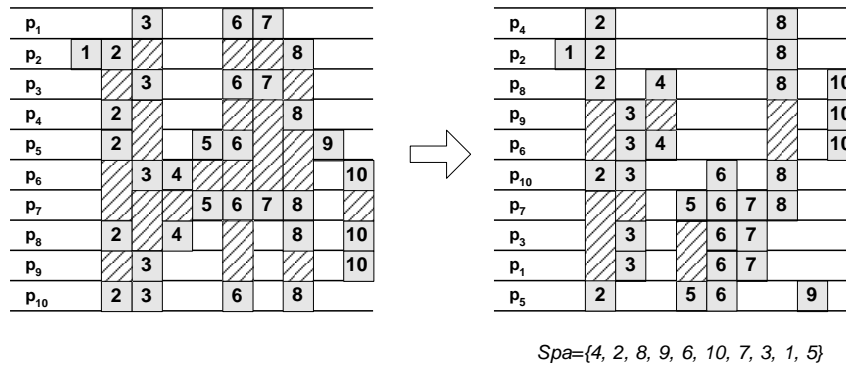
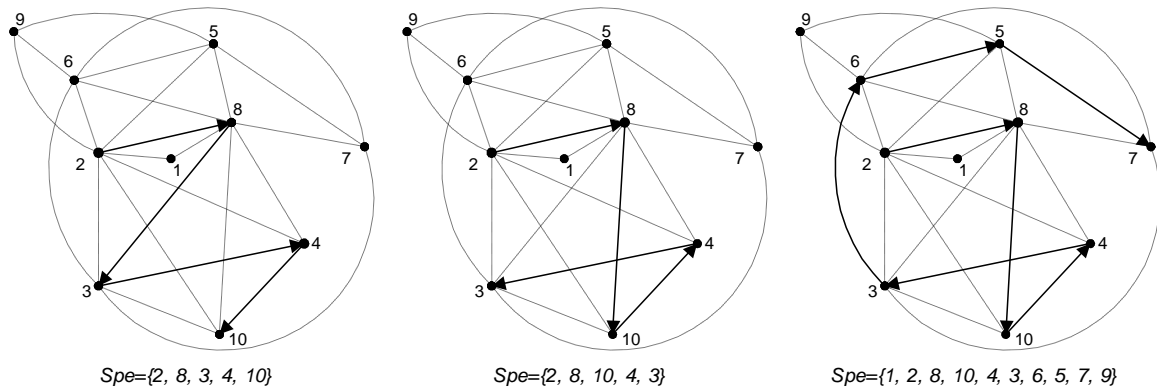


Figura 21 Exemplo do funcionamento da heurística *her*.

A complexidade, no pior caso, da heurística **her** pode ser analisada da seguinte maneira: o cálculo do clique maximal pode ser implementado em $O(n)$, cf. Halldórson (97), e cada rotação pode ser implementada em $O(\log n)$, cf. Angluin e Valiant (79), onde podemos ter no máximo C_r rotações. A totalidade das extensões terá um custo $O(n^2)$ se levarmos em conta que em geral os grafos MOSP são densos, e no pior caso, a cada extensão temos que percorrer n vizinhos. As duas últimas rotinas são igualmente $O(n^2)$, no pior caso. No cômputo geral teremos uma complexidade $O(n^2)$ para **her**. Nos testes computacionais veremos, no entanto, que esta heurística é muito mais rápida do que pode sugerir a análise da complexidade apresentada. Em geral, um fator de grande importância na redução do tempo, é o tamanho do clique maximal encontrado, como veremos na Seção 4.4, quando dos testes e resultados computacionais. Na sequência, a segunda heurística é introduzida.

4.3.2 Heurística de Contração Recursiva de Cliques (**hcrc**)

Nos testes realizados notamos que muitas vezes não havia um único grande clique, mas vários. Partiu-se, então, para a contração recursiva dos cliques, idealmente maximais, no grafo MOSP associado. Inicialmente um clique idealmente maximal é encontrado e este é contraído, tornando-se um hipernó. A partir do hipergrafo induzido, calcula-se um novo clique maximal, que inclua aquele hipernó anteriormente encontrado e assim sucessivamente. A partir da sequência de vértices pode-se gerar uma sequência de cortes, visto que os vértices de cada subsequência são todos cliques. A descrição da heurística **hcrc** é dada a seguir, novamente em um dialeto (*pidin*) à linguagem C .

Heurística Contração Recursiva de Cliques (hcrc)

Entrada: Grafo MOSP $G (E, V)$

Saída: Sequência $Spa=\{pa_1, pa_2, \dots, pa_n\}$ ordenada de *padrões*.

$Spe = \{ \}$

while (true) {

$Spe = Spe \cup \{ \text{vértices do clique maximal} \}$

$\text{contraí_clique_em_um_hipernó ()}$;

 if $|G| = 1$

 break;

}

$Spa = \text{seqüência_de_padrões_associada ()}$;

Retorne Spa ;

Para ilustrar o algoritmo temos a **Figura 22** onde os sucessivos cliques encontrados são $C_1=\{6, 9, 8, 7\}$, $C_2=\{C_1, 5, 3, 2\}$ e $C_3=\{C_2, 10, 4, 1\}$.

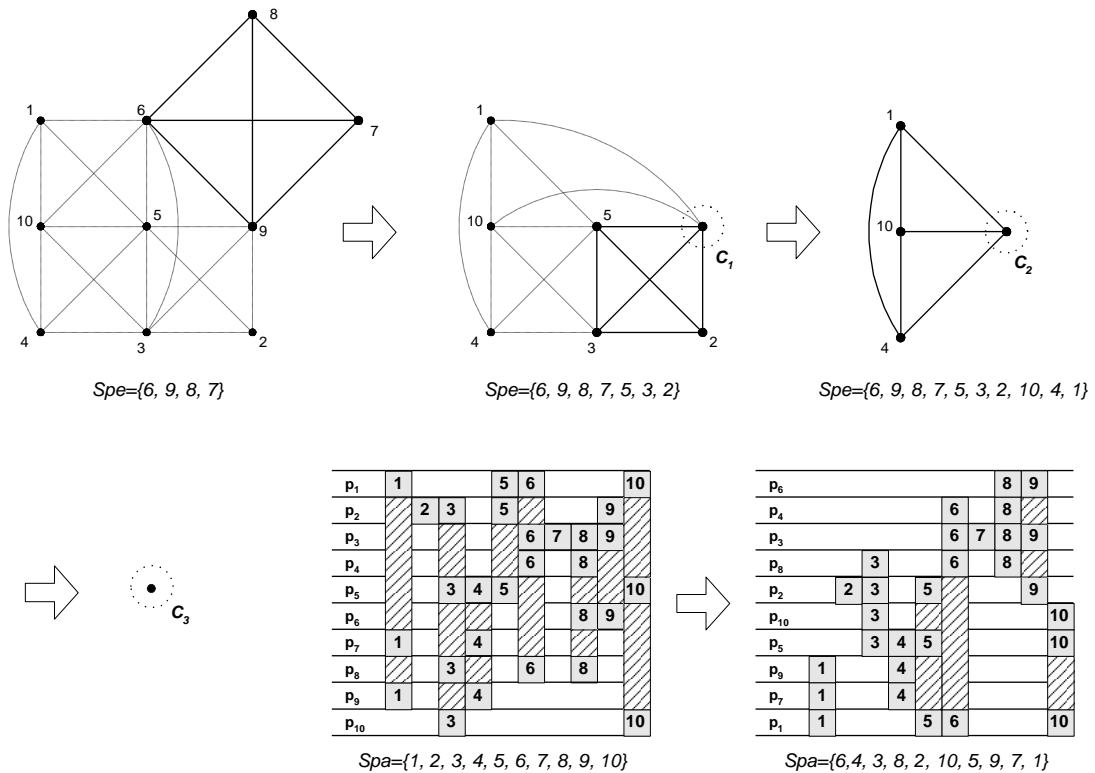


Figura 22 Sequência de peças Spe encontrada através heurística **hcrc** e possível sequência de padrões Spa associada.

A sequência de peças montada ao final do processamento será $Spe=\{6, 9, 8, 7, 5, 3, 2,$

10, 4, 1}, gerando uma possível sequência de padrões $Spa=\{6,4, 3, 8, 2, 10, 5, 9, 7, 1\}$, cujo número máximo de pilhas abertas será igual a 5, que no caso particular deste exemplo é o valor ótimo. Para ter uma base de comparação, caso o processamento seguisse a sequência léxico-crescente original seriam abertas 8 pilhas.

A complexidade, no pior caso, de *hcrc* pode ser analisada pelo custo da contração de cada clique. Para cada vértice de um clique, encontrado em tempo $O(n)$, será feita uma atualização da lista de adjacência de seus vizinhos, o que terá um custo no pior caso de $O(n^2)$. No máximo teremos n cliques, o que dá um tempo, no pior caso, de $O(n^3)$. Caso cada clique tenha um tamanho grande, o tempo real deverá ser bem menor que essa estimativa feita.

4.4 Implementação e Testes Computacionais

Para a avaliação da performance computacional das heurísticas aqui sugeridas e a contraposição destas com aquelas conhecidas da literatura, duas classes de testes foram consideradas. Inicialmente, iremos analisar uma amostragem limitada, na qual cada conjunto de testes, gerados de maneira aleatória, terá uma amostra de tamanho 20, e o número de padrões e peças se limitará a 150. Estes conjuntos de testes seguem à idealização sugerida em Becceneri (99) e servirá para termos uma ideia, imparcial, de como se comportam as heurísticas em casos próximos aos encontrados em um ambiente de corte industrial. Iremos confrontar *her*, *hcrc*, *yuen3* e *ybs99* quanto às médias encontradas em relação ao tempo de processamento computacional e número máximo de pilhas abertas. O segundo conjunto de testes é mais exaustivo, pois iremos gerar testes aleatórios em amostras de tamanho 2000 para até 1000 padrões e peças, confrontando particularmente a heurística extensão-rotação *her* com *yuen3*. Note-se, que não obstante sabermos que problemas contendo 1000 padrões e

peças não são usuais em ambientes de corte industrial, eles podem emergir em outros contextos, *e.g.*, projeto de circuitos *VLSI*. Também, há o intuito de se auferir o comportamento das heurísticas em problemas maiores para termos uma imagem mais fidedigna da qualidade das aproximações, assim como da utilização dos recursos computacionais.

Adicionalmente, poderíamos fazer, ainda, dois possíveis questionamentos em relação aos testes sugeridos. Em primeiro lugar, se não seria interessante executarmos, ao menos nos testes iniciais, o método exato proposto em Becceneri (99). Isto não foi efetuado, pois aparentemente descobrimos um erro no algoritmo de redução por dominância de linhas e/ou colunas, o que faz com que o método exato não funcione em alguns casos. Por exemplo, o problema apontado na **Figura 22** possui um sequenciamento de padrões que abre 5 pilhas no máximo. O método exato retornou 6 pilhas devido a um erro na rotina supracitada. Para termos uma ideia do quanto nos distanciamos da otimalidade podemos utilizar como referência os valores da heurística de contração de arcos *ybs99*, que fornece valores muito próximos do ótimo.

O segundo questionamento é se não deveríamos fazer testes exaustivos com a heurística *ybs99*. Devido ao problema na rotina citada, alguns, poucos, casos não podem ser processados, provocando um erro geral na heurística. Desta forma, é inviabilizado um processamento automático de um número amplo de casos, tal como propomos. Infelizmente, como este erro foi detectado recentemente, não houve tempo hábil para a sua correção pelos referidos autores. O excessivo tempo computacional exigido, por ambos, método exato de Becceneri (99) e heurística *ybs99*, também se configura como um outro fator proibitivo para a execução de testes mais extensivos.

Tal como nos testes, efetuados na Seção 4.1, em relação à busca gulosa, assumimos, sem perda de generalidade, um número igual de padrões e peças distintas, gerando matrizes quadradas. Todos os programas aqui implementados foram testados em um microcomputador com processador *Celeron* 533 MHz, 64Mb de memória principal e sistema operacional Windows98. A heurística *ybs99* foi gentilmente cedida para os nossos testes, e foi originalmente desenvolvida em C++, através do pacote *Visual C++* da Microsoft. As demais heurísticas foram implementadas em linguagem C e geradas através do *Borland C++ Builder*. Não habilitamos nenhuma opção de otimização para gerar as nossas heurísticas e procuramos seguir o mais perto possível o padrão C ANSI.

4.4.1 Testes iniciais comparando as heurísticas *her*, *hcrc*, *ybs99* e *yuen3*

Na **Tabela 6** apresentam-se as médias do número máximo de pilhas abertas e os seus respectivos tempos de execução em segundos. Os rótulos *m-ybs99*, *m-yuen3*, *m-her* e *m-hcrc* são as médias quanto ao número máximo de pilhas abertas no sequenciamento de padrões obtido pelas heurísticas e *t-ybs99*, *t-yuen3*, *t-her* e *t-hrc* são as, respectivas, médias dos tempos de processamento em segundos. Como já comentado no Capítulo 3, um conjunto denominado *10c04* significa que são 10 padrões com no máximo 4 peças em cada padrão entre 10 tipos possíveis.

Devemos tecer alguns breves comentários sobre a **Tabela 6**. As classes foram truncadas em relação ao número máximo de peças permitido, pois a partir de um dado valor a quantidade máxima de pilhas abertas será igual ao número de peças distintas. Por exemplo, para 30 padrões, com no máximo 24 peças dentre as 30 possíveis, todas as sequências de padrões acabam abrindo 30 pilhas. Outra observação a ser feita é sobre o truncamento após 60 padrões. Em geral, na indústria, cf. Becceneri (99), temos um sequenciamento para até este

número de padrões e peças. Adicionalmente foram testadas as classes de 100 e 150 padrões com o intuito de verificar o tempo de processamento das heurísticas para um valor maior que o usual.

classe	m-ybs99	m-yuen3	m-her	m-hcrc	t-ybs99	t-yuen3	t-her	t-hcrc
10c04	5,65	6,05	6,05	6,40	0,0145	0,0006	0,0000	0,0002
10c05	6,40	7,10	6,90	6,95	0,0165	0,0000	0,0000	0,0000
10c06	7,75	8,35	7,80	8,15	0,0170	0,0000	0,0000	0,0003
10c07	8,50	8,90	8,55	8,65	0,0100	0,0000	0,0000	0,0000
10c08	9,10	9,40	9,10	9,15	0,0100	0,0002	0,0000	0,0000
20c06	11,80	13,75	13,65	14,15	0,0320	0,0002	0,0000	0,0003
20c08	14,40	16,15	16,00	16,20	0,0235	0,0003	0,0000	0,0003
20c10	16,20	17,70	17,15	17,35	0,0370	0,0003	0,0000	0,0003
20c12	17,75	19,10	18,10	18,35	0,0365	0,0000	0,0000	0,0004
20c14	18,80	19,80	18,80	18,90	0,0395	0,0000	0,0000	0,0005
20c16	19,85	20,00	19,85	19,85	0,0325	0,0000	0,0000	0,0000
30c08	19,75	23,35	22,40	23,65	0,0460	0,0003	0,0000	0,0014
30c12	24,55	28,25	26,35	26,80	0,0940	0,0000	0,0000	0,0018
30c16	27,50	29,70	28,00	28,45	0,1255	0,0005	0,0000	0,0013
30c20	29,20	29,95	29,30	29,30	0,1535	0,0003	0,0004	0,0007
30c24	30,00	30,00	30,00	30,00	0,1120	0,0003	0,0003	0,0000
40c10	29,40	34,70	32,80	34,40	0,1460	0,0000	0,0000	0,0010
40c15	34,40	38,80	36,80	37,60	0,2480	0,0007	0,0005	0,0028
40c10	37,40	39,90	38,40	38,40	0,4010	0,0000	0,0000	0,0010
40c25	39,30	40,00	39,30	39,30	0,5150	0,0000	0,0000	0,0000
50c10	35,60	42,00	39,90	42,50	0,2800	0,0010	0,0008	0,0038
50c15	42,70	49,30	45,20	46,30	0,4910	0,0007	0,0005	0,0059
50c20	45,90	50,00	47,10	47,80	0,7180	0,0034	0,0019	0,0062
50c30	49,30	50,00	49,30	49,30	1,2270	0,0027	0,0005	0,0009
60c10	42,30	49,10	47,60	49,90	0,4300	0,0034	0,0002	0,0047
60c15	50,40	57,60	54,20	55,60	0,7960	0,0042	0,0000	0,0107
60c10	55,50	60,00	56,80	57,70	1,3510	0,0047	0,0008	0,0131
60c25	56,90	60,00	57,90	58,30	1,7390	0,0059	0,0004	0,0081
100c10	67,05	79,35	77,70	81,15	1,8175	0,0124	0,0009	0,0147
100c15	81,70	94,30	88,60	91,30	3,8620	0,0151	0,0010	0,0338
100c10	88,50	98,90	93,00	93,80	7,8740	0,0154	0,0011	0,0626
100c30	95,40	100,00	97,00	97,80	18,7570	0,0148	0,0015	0,0963
100c40	97,90	100,00	98,80	98,70	32,9110	0,0161	0,0009	0,0234
150c10	98,80	115,80	114,60	122,40	6,4130	0,0393	0,0028	0,0388
150c15	120,10	140,25	131,05	135,10	21,8825	0,0392	0,0022	0,0733
150c20	131,65	147,75	139,20	141,70	44,1570	0,0411	0,0035	0,1430
150c25	138,10	149,85	142,60	144,25	69,0080	0,0427	0,0031	0,2538
150c30	142,20	149,90	145,30	146,20	97,0360	0,0434	0,0027	0,3864

Tabela 6 Médias do número máximo de pilhas abertas e respectivos tempos em segundos.

Através deste limitado conjunto de testes podemos verificar que: primeiramente, percebemos que *her* é a mais rápida entre as heurísticas, confirmando a complexidade quadrática no pior caso, em contraste com o comportamento cúbico das demais. Por outro lado, *her*, é dominada pela heurística *ybs99* quanto à média de pilhas, no entanto apresenta um valor próximo em um tempo bem inferior na maior parte dos casos. Para a classe *150c30*, a heurística de contração de arcos consome da ordem de 97 segundos, na média, para o processamento, em contraste com a fração de segundo da heurística de extensão-rotação *her*. Uma das possíveis explicações da qualidade superior nas médias constatadas da heurística *ybs99* é o processo de contração de arcos, que leva a um hipergrafo completo maior que o clique maximal, como foi exemplificado na **Figura 19** da Seção 4.2, mas sendo também, como contraponto, o responsável pelo grande tempo de processamento apresentado pela heurística *ybs99*.

Surpreendentemente, a heurística de extensão-rotação *her* obteve, em todas as classes testadas, uma aproximação melhor que a heurística *yuen3*. Desta maneira, um fato a ser averiguado em testes mais exaustivos, é se a heurística de extensão-rotação conseguirá ou não obter uma média de pilhas menor que *yuen3*, em um tempo médio *também* inferior, resultado este considerado não trivial de ser obtido.

A heurística *hcrc* também é dominada por *ybs99* em relação às médias de pilhas abertas, obtendo também valores próximos em muitos casos em um tempo inferior. Como apontado na Seção 4.1, para casos mais esparsos a qualidade da aproximação é inferior a *yuen3*. Isto é devido aos cliques bem menores que são encontrados em grafos relativos a matrizes MOSP mais esparsas. Em relação à *yuen3*, podemos verificar uma melhor qualidade de aproximação de *hcrc*. Em 30, das 38 classes testadas, *hcrc* obteve uma melhor média no número de pilhas abertas, em um tempo próximo, que a heurística *yuen3*. Como já

comentamos anteriormente, estamos trabalhando em uma nova heurística, de sequenciamento simples de cliques idealmente maximais, sem fazermos a contração recursiva, obtendo resultados próximos à heurística *her* em tempo e qualidade de aproximação.

4.4.2 Testes extensivos comparando as heurísticas *her* e *yuen3*

Iremos agora realizar os testes de maneira mais extensiva, onde foram consideradas classes de problemas de 100 até 1000 padrões/peças, com 2000 testes aleatórios gerados para cada classe. Em Becceneri (99), foram analisados 20 testes para cada classe, estamos propondo 100 vezes mais casos. Talvez fosse desnecessário este número de testes; todavia a ideia de realizarmos os testes desta forma surgiu ao submetermos um artigo com resultados da heurística de extensão-rotação. Foi sugerido por um dos *referee's* fazer os testes desta maneira para podermos comparar com maior precisão os tempos das heurísticas *yuen3* e *her*. Assim iremos seguir a sugestão também para os dados apresentados nesta seção.

Para cada classe, em lugar de simplesmente irmos aumentando o número de peças permitido em cada padrão, preferimos ir aumentando a densidade do grafo MOSP, de 10% até 90%. Como não existe uma fórmula direta, que dado um número n de padrões/peças, retorne um valor C de peças permitidas para termos uma determinada densidade d , calculamos este valor gerando matrizes aleatórias, também com 2000 casos, incrementado o valor C , a cada valor de n . Desta forma conseguimos os valores da **Tabela 7**, onde podemos ver o valor C necessário para obtemos a densidade desejada para cada dimensão da matriz n .

n	10%	20%	30%	40%	50%	60%	70%	80%	90%
100	5	8	10	12	14	16	18	21	26
200	8	11	14	17	20	23	26	30	37
300	10	14	18	21	25	28	33	38	45
400	11	16	20	25	29	33	38	44	52
500	13	18	23	27	32	37	42	49	59
600	14	20	25	30	35	40	46	54	66
700	15	22	27	32	38	44	50	58	72
800	16	23	29	35	41	47	54	62	75
900	17	24	31	37	43	50	57	67	79
1000	18	26	33	39	45	52	60	70	83

Tabela 7 Valores de **C** necessários para obtermos uma determinada densidade em uma matriz MOSP de dimensão **n**.

Na **Tabela 8** temos os resultados da comparação entre **yuen3** e **her**. Foram gastos aproximadamente 10 dias de processamento, em um micro *Celeron* 533 MHz, para obtermos as médias apresentadas. As médias do número máximo de pilhas abertas são dadas pelos campos **m-her** e **m-yuen3** e os tempos obtidos representados pelas colunas **t-her** e **t-yuen3** em segundos. Podemos afirmar que, nos testes exaustivos, a heurística de extensão-rotação **her** domina **yuen3** em relação ao tempo em todas as classes, realizando o processamento em uma pequena fração do tempo gasto pela heurística **yuen3**. Como um exemplo, vemos que para as classes com 1000 padrões a heurística **yuen3** gasta aproximadamente 12 segundos para o processamento, contra 0,1 segundo da heurística de extensão-rotação. Também o domínio é constatado em relação à qualidade da aproximação. Em 87 das 90 classes de teste, **her** abriu menos pilhas, na média, que **yuen3**. As três únicas classes em que **yuen3** abre menos pilhas são as que geram grafos MOSP bem esparsos: para $n=100$, $d=10\%$, $d=20\%$ e para $n=200$, $d=10\%$. Adicionalmente verificamos como a qualidade de aproximação de **yuen3** vai se deteriorando a medida em que cresce a densidade média do grafo MOSP.

densidade (d)	classe (n)	m-her	m-yuen3	t-her	t-yuen3
10%	100	52,59	44,97	0,00	0,01
	200	136,97	131,17	0,00	0,09
	300	226,44	228,66	0,01	0,28
	400	313,12	321,59	0,02	0,72
	500	413,37	436,95	0,03	1,51
	600	506,20	539,30	0,04	2,69
	700	601,02	644,52	0,05	4,31
	800	697,46	750,85	0,07	6,45
	900	794,65	857,38	0,09	9,19
	1000	893,42	964,45	0,11	12,59
20%	100	69,21	67,72	0,00	0,01
	200	157,26	162,98	0,00	0,09
	300	253,98	271,70	0,01	0,28
	400	349,45	376,92	0,02	0,72
	500	447,36	482,99	0,03	1,53
	600	546,81	588,20	0,04	2,71
	700	647,07	692,00	0,06	4,34
	800	743,87	793,02	0,07	6,50
	900	841,08	893,89	0,09	9,32
	1000	943,10	996,13	0,12	12,68
30%	100	76,71	78,83	0,00	0,01
	200	170,06	182,34	0,00	0,10
	300	269,49	290,65	0,01	0,33
	400	365,44	392,63	0,02	0,74
	500	465,83	495,99	0,03	1,53
	600	564,31	597,17	0,04	2,73
	700	663,68	698,21	0,06	4,37
	800	763,22	798,79	0,08	6,57
	900	863,37	899,24	0,10	9,30
	1000	963,65	999,59	0,12	12,74
40%	100	82,36	87,08	0,00	0,01
	200	178,49	192,46	0,00	0,09
	300	276,82	296,29	0,01	0,30
	400	377,03	398,42	0,02	0,75
	500	474,69	498,87	0,03	1,55
	600	574,85	599,47	0,05	2,75
	700	673,76	699,65	0,06	4,39
	800	774,55	799,87	0,08	6,63
	900	874,12	899,92	0,10	9,35
	1000	973,93	999,97	0,13	12,79
50%	100	86,29	92,57	0,00	0,01
	200	183,99	196,98	0,00	0,09
	300	283,46	298,96	0,01	0,29
	400	382,88	399,62	0,02	0,75
	500	481,94	499,81	0,03	1,57
	600	581,57	599,92	0,05	2,77
	700	681,38	699,98	0,06	4,42
	800	781,64	799,98	0,08	6,62
	900	880,92	899,99	0,11	9,40
	1000	980,54	1.000,00	0,13	12,86

Tabela 8 Médias das pilhas abertas e tempos comparando **her** e **yuen3**.

<i>densidade (d)</i>	<i>classe</i>	<i>m-her</i>	<i>m-yuen3</i>	<i>t-her</i>	<i>t-yuen3</i>
60%	100	89,29	96,07	0,00	0,01
	200	187,75	198,92	0,00	0,09
	300	286,76	299,63	0,01	0,30
	400	386,86	399,91	0,02	0,75
	500	486,63	499,97	0,03	1,57
	600	585,96	599,99	0,05	2,79
	700	686,27	700,00	0,06	4,45
	800	786,20	800,00	0,08	6,66
	900	886,12	900,00	0,11	9,46
	1000	985,64	1.000,00	0,13	12,93
70%	100	91,50	97,97	0,00	0,01
	200	190,51	199,64	0,00	0,09
	300	290,59	299,95	0,01	0,30
	400	390,23	399,99	0,02	0,77
	500	489,78	500,00	0,03	1,59
	600	589,58	600,00	0,05	2,81
	700	689,57	700,00	0,06	4,49
	800	789,69	800,00	0,08	6,70
	900	889,57	900,00	0,11	9,51
	1000	989,37	1.000,00	0,13	13,01
80%	100	93,84	99,31	0,00	0,01
	200	192,93	199,92	0,00	0,09
	300	293,10	300,00	0,01	0,30
	400	392,89	400,00	0,02	0,78
	500	492,75	500,00	0,03	1,61
	600	592,71	600,00	0,05	2,84
	700	692,55	700,00	0,06	4,53
	800	792,42	800,00	0,08	6,76
	900	892,72	900,00	0,11	9,61
	1000	992,52	1.000,00	0,13	13,15
90%	100	96,17	99,92	0,00	0,01
	200	195,64	200,00	0,00	0,10
	300	295,37	300,00	0,01	0,33
	400	395,21	400,00	0,02	0,81
	500	495,22	500,00	0,03	1,64
	600	595,46	600,00	0,04	2,90
	700	695,50	700,00	0,06	4,60
	800	795,13	800,00	0,08	6,84
	900	895,04	900,00	0,10	9,70
	1000	994,97	1.000,00	0,13	13,25

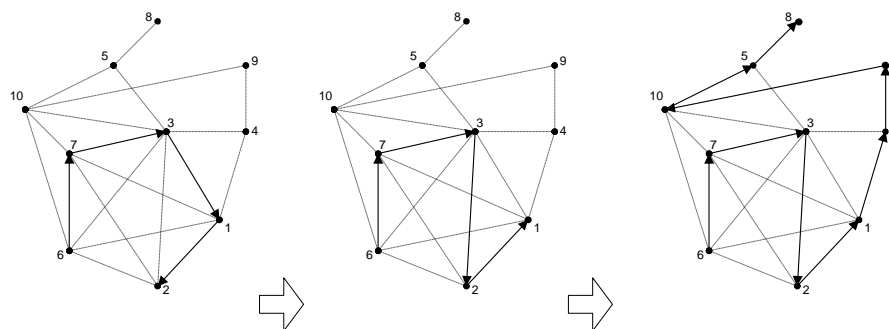
Tabela 8 (cont.) Médias das pilhas abertas e tempos comparando **her** e **yuen3**.

Por último, na próxima **Tabela 9** temos a porcentagem de vértices, rotulada como *%s*, que sobraram para o sequenciamento míope no final da heurística extensão-rotação, após a inserção de vértices conjugados. Como prevíamos, justamente nos 3 casos citados, em que a heurística **her** abriu menos pilhas que **yuen3**, temos um número maior de vértices que

sobraram. Em contrapartida, na grande maioria dos casos, este valor não ultrapassa 1%, o que indica que, o sequenciamento da maior parte dos vértices é realizado através do processo proposto. Podemos confirmar, o que era apenas apontado nos testes iniciais, que *her* se mostra uma opção *frugal* competitiva para o MOSP.

densidade (d)	classe (n)	%s	densidade (d)	classe (n)	%s
10%	100	6,9%	60%	100	1,8%
	200	3,9%		200	0,8%
	300	2,6%		300	0,6%
	400	2,1%		400	0,4%
	500	1,5%		500	0,3%
	600	1,3%		600	0,3%
	700	1,1%		700	0,2%
	800	0,9%		800	0,2%
	900	0,8%		900	0,2%
	1000	0,7%		1000	0,1%
20%	100	4,7%	70%	100	1,5%
	200	2,6%		200	0,7%
	300	1,6%		300	0,4%
	400	1,2%		400	0,3%
	500	0,9%		500	0,2%
	600	0,7%		600	0,2%
	700	0,6%		700	0,2%
	800	0,5%		800	0,1%
	900	0,5%		900	0,1%
	1000	0,4%		1000	0,1%
30%	100	3,6%	80%	100	1,1%
	200	1,9%		200	0,5%
	300	1,1%		300	0,3%
	400	0,9%		400	0,2%
	500	0,6%		500	0,2%
	600	0,5%		600	0,1%
	700	0,4%		700	0,1%
	800	0,4%		800	0,1%
	900	0,3%		900	0,1%
	1000	0,3%		1000	0,1%
40%	100	2,8%	90%	100	0,7%
	200	1,4%		200	0,3%
	300	0,9%		300	0,2%
	400	0,6%		400	0,2%
	500	0,5%		500	0,1%
	600	0,4%		600	0,1%
	700	0,3%		700	0,1%
	800	0,3%		800	0,1%
	900	0,2%		900	0,1%
	1000	0,2%		1000	0,1%
50%	100	2,2%			
	200	1,1%			
	300	0,6%			
	400	0,5%			
	500	0,4%			
	600	0,3%			
	700	0,3%			
	800	0,2%			
	900	0,2%			
	1000	0,2%			

Tabela 9 Porcentagem de vértices que “sobraram” no final da heurística *her*.



*Exemplificação do processo de extensão-rotação utilizado pela heurística **her***

Capítulo 5: Conclusões

Vimos que o problema de sequenciamento de padrões, dentro da modelagem pela Teoria dos Grafos aqui adotada, possui no clique maximal uma parte importante de qualquer boa solução a ser alcançada. Mas, em princípio, um método de ataque naquele baseado é desaconselhado pelo resultado teórico forte, *cf.* Arora *et al.* (92), contrário à adoção de cliques maximais como ponto de partida para aproximações. Para o caso específico do MOSP, esta hipótese é apoiada pelos tempos computacionais demandados pela heurística *ybs99* e abordagens por meta-heurísticas, *cf.* Seção 3.1.5. Além de todas essas dificuldades, a proposição de uma nova heurística deve ser melhor que a aproximação boa, em tempo reduzido, e extremamente simples como a de *yuen3*. Seria possível, então, encontrar algo melhor que a heurística *yuen3* em qualidade e tempo, mantendo sua simplicidade? Podemos concluir que a heurística extensão-rotação **her** parece ser uma forte candidata a isso, sendo talvez a contribuição mais interessante da presente dissertação e que merece um estudo teórico futuro.

Concluindo, como contribuições originais da dissertação, podemos citar:

1. O método geral aqui desenvolvido. Nele iniciamos sempre o sequenciamento através das peças associadas ao clique maximal do grafo MOSP. Este clique maximal é obtido a partir do conjunto independente maximal através da busca gulosa por vértice

de grau mínimo no grafo complementar MOSP. Este método é *frugal* pelo tempo linear da busca gulosa citada.

2. As heurísticas *her* inspirada no algoritmo extensão-rotação para detecção de circuitos hamiltonianos e *hcrc* de contração recursiva de cliques. As duas heurísticas partem do método geral proposto, com bons resultados apresentados nos testes computacionais.
3. Uma caracterização melhor dos grafos MOSP, quanto à densidade do número de arestas e tamanho do clique maximal.

Podemos também resumir as nossas sugestões para trabalhos originais futuros:

1. Um método *exato* partindo das contribuições feitas pela presente dissertação.
2. Um algoritmo *on-line* na geração de padrões visando o MOSP, que parta da busca gulosa por vértice de grau mínimo.
3. Adaptação do método e heurísticas sugeridas para os problemas *Gate Matrix Layout* e *PLA Folding*.
4. Estudar uma abordagem que utilize como solução inicial árvores geradoras (*spanning tree's*) obtidas através do algoritmo *Borůvka* (26).

Abaixo tecemos algumas considerações sobre as nossas sugestões para trabalhos futuros.

O algoritmo *on-line* trataria do problema da geração de padrões *em conjunto* com o sequenciamento dentro de uma única estrutura de dados, modelada através da Teoria dos Grafos, fortalecendo a unidade e harmonia de uma possível solução.

Para adaptar as heurísticas sugeridas para os problemas *PLA Folding* e *Gate Matrix*

Layout, podemos levar em conta a estrutura de dados denominada *PQ-trees*, sugerida em Booth e Lueker (76), prestando especial atenção ao relacionamento apontado pelos autores entre esta estrutura de dados e a Teoria dos Grafos.

Em Becceneri (99) foi sugerida a utilização de *spanning tree's* para um novo método enumerativo exato. Esta abordagem não foi bem sucedida devido ao grande número de árvores possíveis, *cf.* Aigner e Ziegler (99). A possível linha de pesquisa que sugerimos é distinta da apontada por Becceneri (99). Propomos uma heurística onde o *início* será uma árvore, obtida pelo algoritmo para árvores geradoras de peso mínimo (*minimum-weight spanning tree*), proposta inicialmente por Borůvka (26), e depois tornada popularmente conhecida, de forma independente, como algoritmo de Kruskal (56). Em Applegate *et al.* (01), esta rápida heurística foi escolhida para fornecer uma solução inicial para resolução do problema do caixeiro viajante, sendo interessante notar que a aproximação fornecida pela solução do algoritmo baseado em Borůvka (26), não é a melhor em erro, e no entanto, forneceu uma boa solução *inicial* para a resolução do caixeiro viajante, quando *em conjunto* com o algoritmo *Lin-Kernighan* encadeado.

Referências Bibliográficas

- Abello, J., Pardalos, P.M., Resende, M.G.C., On maximum clique problems in very large graphs, in: J. Abello and J. Vitter (eds.) *External Memory Algorithms, DIMACS Series, AMS*, 1999.
- Aigner, M. and Ziegler, G.M., Cayley's formula for the number of trees, chapter 22, 141-146, *Proofs from The Book*, Springer Verlag, Berlin, 1999.
- Alon, N., Krivelevich, M., Sudakov, B., Finding a large hidden clique in a random graph, *Random Structures and Algorithms*, 13: 457-466, 1998.
- Angluin, D., Valiant, L.G., Fast probabilistic algorithms for hamiltonian circuits and matchings, *Journal of Computer and System Sciences*, 18: 155-193, 1979.
- Applegate, D., Bixby, R., Chvátal, V., Cook, W., On the solution of traveling salesman problem, *Documenta Mathematica J. DMV – Extra Volume ICM III*, 645-656, 1998.
- Applegate, D., Bixby, R., Chvátal, V., Cook, W., Finding tours in the TSP, em preparação, 2001.
- Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M., Proof verification and hardness of approximation problems, *Proc. 33rd Ann. Symp. Found. Computer Sci.*, 14-23, 1992.
- Ashikaga, F.M., Fabrin A.P., Gil J.C., Fernandes L.R., Soma, N.Y., O problema da minimização de pilhas abertas: uma abordagem por conjunto independente maximal e circuito hamiltoniano, in: *Simpósio Brasileiro de Pesquisa Operacional*, 1872-1882, Viçosa, Minas Gerais, 2000.
- Becceneri, J.C., *O problema de sequenciamento de padrões para a minimização do número máximo de pilhas abertas em ambientes de cortes industriais*, Tese de Doutorado, Instituto Tecnológico de Aeronáutica, São José dos Campos, 1999.

- Blum, M., *Lectures Notes in Analysis and Complexity of Algorithms*, Berkeley University, Berkeley, 1995.
- Bollobás, B., *Random Graphs*, Academic Press, 1985.
- Bollobás, B., Fenner, T.I., Frieze, A.M., An algorithm for finding hamiltonian paths and cycles in random graphs, *Combinatorica*, 7(4): 327-341, 1987.
- Bomze, I.M., Budinich, M., Pardalos, P.M. e Pelillo, M., The maximum clique problem, in: D.-Z. Du and P.M. Pardalos (eds.) *Handbook of Combinatorial Optimization*, 1-74, Kluwer Academic Publishers, 1999.
- Booth, K.S., Lueker, G.S., “Testing for the consecutive ones property, interval graph, and graph planarity using PQ-trees algorithms”, *Journal of Computer Systems and Science*, 13: 335-379, 1976.
- Boppana, R.B., Halldórsson, M.M., Approximating maximum independent sets by excluding subgraphs. *BIT*, 32(2): 180-196, June 1992.
- Borůvka, O., On a certain minimal problem (em tcheco), *Práce Moravské Přírodovědecké Společnosti*, 3:37-58, 1926.
- Broder, A., Frieze, A.M., Shamir, E., Finding hidden hamiltonian cycles, *Random Structures and Algorithms*, 5: 395-410, 1995.
- Campello, R.E., Maculan, N., *Algoritmos e Heurísticas: desenvolvimento e avaliação de performance*, Editora Universitária da Universidade Federal Fluminense, Niterói, Rio de Janeiro, 1994.
- Chvátal, V., Probabilistic methods in graph theory, *Annals of Operations Research*, 1: 171-182, 1984.
- Chvátal, V., Hamiltonian cycles, in: Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B. (eds.), *The traveling salesman problem*, 403-429, John Wiley&Sons, 1985.

- Crescenzi, P., Kann, V., How to find the best approximation results – a follow-up to Garey and Johnson, *ACM Sigact news*, 29(4): 90-97, 1998.
- Dantzig, G.B., Fulkerson, R., Johnson, S.M., Solution of a large-scale traveling salesman problem, *Operations Research*, 2: 393-410, 1954.
- Demange, M., Paschos, V.Th., Improved approximations for maximum independent set via approximation chains, *Appl. Math. Lett.*, 10(3): 105-110, 1997.
- Diestel, R., *Graph Theory*, Graduate Texts in Mathematics, sec. ed., Springer Verlag, Berlin, 2000.
- Dirac, G.A., Some theorems on abstract graphs, *Proc. London Math. Soc.*, 2: 69-81, 1952.
- Du, D.-Z., e Pardalos, P.M. (eds.), *Handbook of Combinatorial Optimization*, Kluwer Academic Publishers, 1999.
- Dyckhoff, H., Abel, D., Gal, D., Trim loss and related problems, *Omega, The Journal of Management Sciences*, 13(1): 59-72, 1985.
- Dyckhoff, H. and Finke, U., *Cutting and packing in production and distribution: typology and bibliography*, Springer Verlag, Heidelberg, 1992.
- Dyson, R.G., Gregory, A.S., The cutting stock problem in the flat glass industry. *Operational Research Quarterly*, 25: 41-54, 1974.
- Faggioli, E., Bentivoglio, C.A., Heuristic and exact methods for the cutting sequencing problem, *European Journal of Operational Research*, 110(3): 564-575, 1998.
- Ferreira, A.G., Song, S.W., Achieving optimality for gate matrix layout and PLA folding: a graph theoretic approach, *Integration, The VLSI Journal*, 14(2): 173-195, 1992.
- Fink, A., Voss, S. Applications of modern heuristic search methods to pattern sequencing problems, *Computer and Operations Research*, 26(1): 17-34, 1999.

- Foerster, H.E., Wäescher G., Simulated annealing for order spread minimization in sequencing cutting patterns, *European Journal of Operational Research*, 110: 272-281, 1998.
- Frieze, A., McDiarmid, C., Algorithmic theory of random graphs, *Random Structures and Algorithms*, 10: 5-42, 1997.
- Frieze, A., Yukich, J., Probabilistic analysis of the traveling salesman problem, em preparação, 2001.
- Garey, M.R., Graham, R.L., Johnson, D.S., Knuth, D.E., Complexity results for bandwidth minimization, *SIAM Journal of Applied Mathematics*, 34(3): 477-495, 1978.
- Garey, M.R. and Johnson, D.S., *Computers and Intractability: A Guide to the theory of NP-Completeness*, Freeman, San Francisco, 1979.
- Gibbs, N.E., Poole, W.G., Stockmeyer, P.K., An algorithm for reducing the bandwidth and profile of a sparse matrix, *SIAM Journal of Numerical Analysis*, 13(236-250), 1976.
- Gilmore, P., Gomory, R., A linear programming approach to the cutting-stock problem, *Operations Research*, 9: 849-859, 1961.
- Graham, R.L., Rothschild, B.L., Spencer, J.H., *Ramsey Theory*, John Wiley&Sons, 1990.
- Halldórsson, M.M., *Frugal Methods for the Independent Set and Graph Coloring Problems*, Tese de Doutorado, Rutgers, The State University of New Jersey, New Jersey, 1991.
- Halldórsson, M.M., Radhakrishnan, J., Greed is good: approximating independent sets in sparse and bounded-degree graphs, *Proc. STOC'94*, 439-448, 1994a.
- Halldórsson, M.M., Radhakrishnan, J., Improved approximations of independent sets in bounded-degree graphs via subgraph removal, *Nordic Journal of Computing*, 1(4): 475-492, 1994b.

- Halldórsson, M.M., Radhakrishnan, J., Greed is good: approximating independent sets in sparse and bounded-degree graphs, *Algorithmica*, 18: 145-163, 1997.
- Homer, S., Peinado, M., Experiments with polynomial-time clique approximation algorithms on very large graphs, in D.S. Johnson and M.A. Trick (eds.), *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, DIMACS, 26: 146-167, 1996.
- Jerrum, M., Large cliques elude the Metropolis process, *Random Structures and Algorithms*, 3: 347-359, 1992.
- Johnson, D.S., Approximation algorithms for combinatorial problems, *J. Comput. Syst. Sci.*, 9: 256-278, 1974.
- Johnson, D.S., McGeoch, L.A., The travelling salesman problem: a case study, in: E.H.L. Aarsts and J.K. Lenstra (eds.), *Local Search in Combinatorial Optimization*, 215-310, John Wiley & Sons, Chichester, 1997.
- Knuth, D.E., The Sandwich Theorem, *The Electronic Journal of Combinatorics*, v.1:A1, 1994.
- Kohayakawa, Y., Soares, J.A., Demonstrações transparentes e a impossibilidade de aproximações, *Livro do 20º Colóquio Brasileiro de Matemática*, IMPA, 1995.
- Koršunov, A.D., Solution of a problem of Erdős and Rényi on hamiltonian cycles in nonoriented graphs, *Dokl. Akad. Nauk SSSR*, 228(3):760-764, 1976.
- Kruskal Jr., J. B., On the shortest spanning subtrees of a graph and the travelling salesman problem, *Proc. Amer. Math. Soc.*, 7:48-50, 1956.
- Kučera, L., *Combinatorial Algorithms*, Adam Hilger, 1990.
- Kučera, L., Expected complexity of graph partitioning problems, *Discrete Applied Mathematics*, 57: 193-212, 1995.

- Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B., *The traveling salesman problem*, John Wiley&Sons, 1985.
- Lecky, J.E., Murphy, O.J., Absher, R.G., Graph theoretic algorithms for the PLA folding problem, *IEEE Trans. on Computer-Aided Design*, 8(9): 1014-1021, 1989.
- Lin, S., Kernighan, B.W., An effective heuristic algorithm for the traveling-salesman problem, *Operations Research*, 21: 498-516, 1973.
- Linhares, A., *Exploring the Connections Between the Sequencing of Cutting Patterns and VLSI Design*, PhD Thesis, INPE, em preparação, 2001.
- Linhares, A., Yanasse, H.H., Minimização de pilhas em aberto e problemas relacionados: alguns resultados de complexidade. In: *Simpósio Brasileiro de Pesquisa Operacional*, 30, Curitiba, Paraná, 1998.
- Linhares, A., Yanasse, H.H., On the minimization of open stacks problem, *personal communication*, 2000.
- Lins, S. Traversing trees and scheduling tasks for duplex corrugator machines, *Pesquisa Operacional*, 9: 40-54, 1989.
- Lovász, L., On the Shannon capacity of a graph, *IEEE Trans. Inform. Theory*, 25: 1-7, 1979.
- Madsen, O.B.G., Glass cutting in a small firm, *Mathematical Programming*, 17: 85-90, 1979.
- Madsen, O.B.G., An application of travelling-salesman routines to solve pattern-allocation problems in the glass industry, *Journal of Operational Research Society*, 39: 249-256, 1988.
- Marconi, J., Foster, J.A., A hard problem for genetic algorithms: finding cliques in Keller graphs, *IEEE Evolutionary Computations Proceedings*, 650-655, 1998.

- Möhring, R.H., Graph problems related to gate matrix layout and PLA folding, in: G. Tinhofer et al. (eds.), *Computational Graph Theory*, Computing Supplement 7, Springer Verlag, Wien, 1990.
- Morabito, R.N., Arenales, M. N. Um exame dos problemas de corte e empacotamento, *Pesquisa Operacional*, 12(1):1-20, 1992.
- Pardalos, P.M., Xue, J. The maximum clique problem, *J. Global Optimization*, 4: 301-328, 1994.
- Pósa, L., Hamiltonian circuits in random graphs. *Discrete Mathematics*, 14: 359-364, 1976.
- Sasaki, G., The effect of the density of states on the Metropolis algorithm, *Information Processing Letters*, 37: 159-163, 1991.
- Shmoys, D.B., Computing near-optimal solutions to combinatorial optimization problems. In: Cook, W., Lovász, L., Seymour, P.D. (eds.), *Combinatorial Optimization*, DIMACS Series, 355-397, 1995.
- Syslo, M.M., Deo, N. e J.S. Kowalick , *Discrete optimization algorithms with Pascal programs*, Prentice Hall, Englewood Cliffs, 1983.
- Xu, Y., Abd-El-Barr, M., McCrosky, C., Graph-based output phase assignment for PLA minimization, *IEEE Trans. on Computer-Aided Design*, 14(5): 613-622, 1995.
- Yanasse, H.H., Minimization of open orders – polynomial algorithms for some special cases, *Pesquisa Operacional*, 16(1): 1-26, junho 1996.
- Yanasse, H.H., On a pattern sequencing problem to minimize the maximum number of open stacks, *European Journal of Operational Research*, 100: 456-463, 1997a.
- Yanasse, H.H., A transformation for solving a pattern sequencing problem in the wood cut industry, *Pesquisa Operacional*, 17(1): 57-70, junho 1997b.

- Yanasse, H.H., A note on generating solutions of a pattern sequencing problem to minimize the maximum of number of open orders, *Technical Report LAC-002/98*, INPE, São José dos Campos, SP, Brazil, 1998.
- Yanasse, H.H, Becceneri, J.C., Soma, N.Y. Bounds for a problem of sequencing patterns, *Pesquisa Operacional*, 19(2): 1-29, dezembro 1999.
- Yuen, B.J., Heuristics for sequencing cutting patterns, *European Journal of Operational Research*, 55: 183-190, 1991.
- Yuen, B.J., Improved heuristics for sequencing cutting patterns, *European Journal of Operational Research*, 87: 57-64, 1995.
- Yuen, B.J., Richardson, K.V, Establishing the optimality of sequencing heuristics for cutting stock problems, *European Journal of Operational Research*, 84: 590-598, 1995.