Universidade Federal de Ouro Preto - UFOP
Instituto de Ciências Exatas e Biológicas - ICEB
Departamento de Computação - DECOM

Um Algoritmo Heurístico Aplicado à Homogeneização das Características Físicas de Produtos II

Bolsista: Jordi Alves Reinsma
Orientador: Marco Antonio Moreira de Carvalho – DECOM/UFOP

Nota: Relatório Final referente ao período de 01/08/2017 a 31/07/2018, apresentado à Universidade Federal de Ouro Preto, como parte das exigências do Programa de Iniciação à Pesquisa.

Local: Ouro Preto - Minas Gerais - Brasil
Data: 11 de julho de 2018

Método Exato para Resolução do Problema de Minimização de Blocos de Uns Consecutivos

O problema de Minimização de Blocos de Uns Consecutivos (ou COB, do inglês *Consecutive Ones Block Minimization Problem*) é um problema $\mathcal{NP}$-Difícil que consiste em determinar a permutação das colunas de uma matriz binária que minimize o espalhamento dos elementos não nulos em cada uma de suas linhas. Este problema é estudado de maneira independente sob diferentes nomes em diferentes áreas do conhecimento, como pesquisa operacional, arqueologia, microbiologia, sistemas de informação e matemática. Este trabalho reporta a criação de um método exato para solução do COB com base em resultados parciais anteriores presentes na literatura. O COB é reduzido a uma versão do conhecido Problema do Caixeiro Viajante (PCV) e resolvido de maneira exata pelo resolver Concorde, atual estado da arte para solução do PCV. Nos experimentos computacionais, foram geradas pela primeira vez as soluções ótimas para o único conjunto de instâncias *benchmark* da literatura, comparando as soluções obtidas com os métodos estado da arte do COB.

Assinatura do orientador(a): ______________________________
Marco Antonio Moreira de Carvalho

Assinatura do bolsista: ______________________________
Jordi Alves Reinsma

Sumário

Capítulo 1

Introdução

Em Ciência da Computação, define-se um *problema de otimização* como um tipo de problema computacional em que é necessário determinar a melhor solução entre todas possíveis. Este processo de busca envolve primeiramente a definição de uma função objetivo – uma função que associa um valor à uma solução, o que permite comparar intuitivamente soluções diferentes de modo a determinar a melhor solução possível, também chamada de *solução ótima*. O segundo passo consiste em pesquisar e analisar o *espaço de soluções*, ou seja, o conjunto de todas as soluções. Entretanto, para formar soluções para um problema computacional, é necessário respeitar as *restrições* impostas. Desta maneira, a busca deve também reconhecer e desconsiderar as soluções inviáveis contidas no espaço de soluções, ou seja, aquelas que não respeitam as restrições do problema.

A *Pesquisa Operacional* é a disciplina que estuda a aplicação de métodos científicos à solução dos problemas de otimização para auxiliar no processo de tomada de decisões. Como exemplo, cita-se projetar, planejar e operar sistemas em situações que requerem a alocação eficiente de recursos escassos. O desafio reside no fato de que tomar a decisão sobre qual o melhor projeto, planejamento ou operação é uma tarefa que, mesmo para situações envolvendo poucas dezenas de variáveis, torna-se inviável de se analisar todas as possibilidades até certificar-se de que foi determinada a melhor resposta. Isso é verdade mesmo considerando a utilização de supercomputadores para realização desta análise, uma vez que a dificuldade é intrínseca a boa parte dos problemas práticos de interesse [Goldbarg and Luna, 2005]. Problemas com tais características combinatórias surgem em contextos produtivos, logísticos, financeiros, militares, industriais, laboratoriais e sociais, por exemplo.

Segundo Arenales et al. [2017], para contornar as dificuldades da solução de problemas de *otimização combinatória*, são utilizados métodos específicos para determinação de soluções ótimas ou sub-ótimas, como a programação matemática, heurísticas e metaheurísticas. Todos os métodos exigem do pesquisador a compreensão das características do problema abordado e a habilidade de abstrair e traduzir as informações mais importantes em um modelo matemático ou modelo de simulação. Após a modelagem, é necessária a habilidade de desenvolver métodos de solução para o modelo em questão, considerando a qualidade das soluções geradas e também a rapidez do método como métricas de avaliação.

Dentre os vários problemas já abordados na literatura de Pesquisa Operacional, há uma série destes, encontrados principalmente em contextos industriais, que são abstraídos e modelados em matrizes binárias. O Problema de Determinação de Leiaute de Matrizes de Portas (*Gate Matrix Layout Problem*, GMLP), o Problema de Minimização de Espalhamento de Ordens de Compra (*Minimization of Order Spread Problem*, MORP) e o Problema de Minimização de Pilhas Abertas (*Minimization of Open Stacks Problem*, MOSP) são exemplos desta modelagem em contextos industriais. O GMLP está relacionado ao projeto de circuitos eletrônicos com dimensões otimizadas, o MORP está relacionado com o tempo de atendimento a pedidos de compra e o MOSP está relacionado com a utilização otimizada de estoque intermediário. Da mesma maneira, problemas combinatórios com a mesma modelagem podem ser encontrados no contexto da matemática pura. Cita-se o Problema de Minimização de Banda (*Bandwidth Minimization Problem*), Problema de Largura de Corte (*Cutwidth*) e Problema de Aumento de Uns Consecutivos (*Consecutive Ones Augmentation Problem*).

Neste trabalho será discutido o Problema de Minimização de Blocos de Uns Consecutivos (*Consecutive Ones Blocks*, COB), em que, a partir de uma matriz binária, deve-se obter o menor número de descontinuidades de uns nas linhas da matriz, utilizando-se de permutações das colunas. Define-se *descontinuidade* de uns como a presença de qualquer quantidade de zeros entre os uns. Também será discutida a relação deste problema com os demais citados anteriormente, assim como a aplicação direta do COB na indústria.

1.1 Justificativa

O COB é um problema da classe *NP-Difícil* [Garey and Johnson, 1979], possuindo, portanto, uma grande complexidade teórica. A dificuldade de se trabalhar com problemas com esta característica é uma motivação para encontrar métodos para sua melhor resolução, acrescentando conhecimento à literatura. Adicionalmente, este problema possui similaridades de resolução com outros problemas de matrizes binárias, como será comentado na revisão da literatura.

O Problema de Minimização de Descontinuidades (*Minimization of Discontinuities Problem*, MDP) é exemplo de uma aplicação do COB no contexto industrial. Neste problema, o objetivo é a redução do número de interrupções na fabricação sequencial de produtos. Estas interrupções podem ocasionar ociosidade da linha de produção, ou mesmo diferenças nas características físicas dos produtos. Em ambos os casos, as interrupções influenciam os custos relacionados à produção. Assim, o estudo de métodos de solução do COB tornam-o interessante economicamente para empresas de pequeno, médio e grande porte.

O problema de reduzir a quantidade de “espaços” entre “elementos” também ocorre na arqueologia [Kendall, 1969], microbiologia [Alizadeh et al., 1995], compressão de dados [Johnson et al., 2004] e na organização de arquivos [Kou, 1977]. A multidisciplinaridade do problema também é um fator motivacional para sua abordagem.

Capítulo 2

Objetivos

Em linhas gerais, o objetivo deste trabalho é elaborar um método exato derivado que pôde ser derivado através de observações já encontradas na literatura. Este método deve possuir tempo de execução hábil, no qual possa ser utilizado no contexto do problema abordado, mas com uma possibilidade de generalização. São objetivos específicos:

1. Propor um modelo de otimização que contemple apropriadamente as especificidades de diferentes aplicações do problema abordado com o uso dos modelos descritos na literatura;

2. Implementação computacional de um *software* para determinação de soluções de alta qualidade para o problema abordado usando ferramentas de Inteligência Computacional, o que inclui a utilização de métodos heurísticos e metaheurísticos;

3. Pesquisar técnicas para melhoria de soluções obtidas por heurísticas usando programação linear inteira;

4. Avaliação do *software* implementado considerando dados reais e também com problemas teste publicamente disponíveis;

5. Ampliar o conjunto de problemas testes com soluções ótimas já conhecidas através da execução de testes sistemáticos usando formulações já desenvolvidas;

6. Produzir trabalhos para serem publicados em periódicos e eventos nacionais e internacionais, a fim de contribuir para a promoção dos centros de pesquisas nacionais e também da tecnologia.

Capítulo 3

Revisão da Literatura

O problema de Blocos de Uns Consecutivos é estudado há mais de 40 anos por diversos pesquisadores, que consideram o problema em diferentes contextos e utilizam diferentes métodos de resolução. Neste capítulo, são consideradas as pesquisas mais relevantes encontradas na literatura sobre o COB e outros problemas relacionados. São mantidas as denominações originais utilizadas em cada trabalho.

A primeira pesquisa analisada relativa ao COB é a de Dyson and Gregory [1974]. No trabalho, foi realizado um estudo da produção de chapas de vidro de uma fábrica inglesa, visando solucionar o Problema de Corte de Estoque (*Cutting-Stock Problem*) em combinação do MDP, este último sendo abstraído como Problema de Alocação de Padrões (*Pattern Allocation Problem*). Primeiro, o Problema de Corte foi resolvido via programação linear e, a partir dos padrões de corte obtidos, a fase de sequenciamento dos padrões foi modelada como o Problema do Caixeiro Viajante (*Traveling Salesman Problem*, TSP). Esta modelagem foi descartada, visto que o TSP não possuía resolução praticável na época da pesquisa. Foi então proposto um algoritmo *branch-and-bound* modificado para gerar os padrões de corte e o sequenciamento. A função objetivo utilizada penaliza as descontinuidades e prioriza sequências de padrões consideradas difíceis de ordenar por análise empírica. Também, são priorizados os padrões que produzem peças já inseridas na sequência, afim de finalizá-las rapidamente, minimizando a ocorrência de interrupções.

Kou [1977], Garey and Johnson [1979] apresentaram provas formais de que o COB pertence à classe *NP-Difícil*. Anos depois, Haddadi [2002], estudando o COB sob o nome Problema de Minimização de Blocos Consecutivos (*Consecutive Blocks Minimization Problem*, CBMP), demonstra que o problema mantém tal propriedade mesmo em matrizes binárias restringidas a possuir exatamente dois elementos não nulos por linha.

O COB é abordado novamente em conjunto com o Problema de Corte de Estoque na pesquisa de Madsen [1979], no contexto de uma pequena empresa produtora de chapas de vidro. Após a obtenção dos padrões de corte via resolução do Problema da Mochila (*Knapsack Problem*), é utilizada a representação de matriz binária, quadrada e simétrica, para indicar se padrões possuem peças em comum ou não. Em seguida, é aplicada uma heurística originalmente aplicada ao problema de Largura de Banda de Matrizes Simétricas (*Matrix Bandwidth*) para a redução da distância de produção e, consequentemente, a minimização das interrupções. Em uma pesquisa seguinte sob o mesmo contexto, Madsen [1988] transforma a instância de COB advinda dos padrões de corte em uma instância de TSP, obtendo então uma solução utilizando uma heurística sub-ótima de TSP.

Haddadi and Layouni [2008] discutem o problema de Minimização de Blocos Consecutivos (*Consecutive Blocks Minimization*, CBM), uma aplicação do COB para a redução da complexidade de resolução de sistemas de equações lineares. Neste trabalho, é demonstrado como construir uma solução com fator de aproximação de 1,5 do valor da solução ótima para o problema. A transformação se baseia em reduções entre problemas que não alteram a solução do problema original. A instância de CBM, uma matriz binária de n linhas e m colunas é transformada no problema de Minimização de Blocos Circulares (*Circular Blocks Minimization*, CIR), que relaxa a interpretação da matriz no sentido de que as colunas são consideradas em ordem circular. Consequentemente, os blocos são também circulares. Em seguida, é feita a transformação do CIR para uma instância simétrica do TSP métrico. A matriz de distâncias criada, baseada na distância entre as colunas do problema original, é equivalente à *distância de Hamming*, o que caracteriza o *Hamming Traveling Salesman Problem* (HTSP). A partir desses dados, é utilizado o algoritmo de Christofides [1976], cujo fator de aproximação é de 1,5 para encontrar uma solução para o TSP na instância transformada.

Novamente sob o contexto de redução das dimensões de sistemas lineares, Haddadi et al. [2015] propõe duas buscas locais para o CBM com complexidade $O(n^2)$, sendo n a quantidade de linhas da matriz binária do problema. A primeira busca local é a troca da posição de duas colunas (método *2-swap*), e a segunda é a remoção de uma coluna de sua posição para inserção em outra posição, deslocando as demais colunas (método *shift*). Ambas as buscas locais são implementadas com um método de avaliação de soluções de complexidade de tempo $O(m)$, em que m representa o número de colunas da matriz binária. Esse método denominado Δ-avaliação, constitui uma contribuição importante, uma vez que a avaliação de uma matriz binária exige complexidade $\Theta(mn)$. Para comparar a performance das heurísticas, foram utilizadas 5 instâncias reais de uma companhia ferroviária alemã, além de 45 instâncias geradas aleatoriamente com diferentes características. As instâncias artificiais, gentilmente cedidas pelo autor, serão utilizadas para avaliar a performance do algoritmo produzido neste trabalho, comparando com os resultados obtidos por demais autores.

Nascimento and Carvalho [2017] introduzem uma representação em grafos do COB, estudado ao contexto do MDP. A modelagem considera as peças a serem produzidas como sendo os vértices, possuindo adjacência entre si se as peças estão presentes em um mesmo padrão. Adicionalmente, o peso de cada aresta indica o número de vezes em que cada par de peças é produzido a partir de um mesmo padrão. Utiliza-se a busca em largura com modificações para gerar uma solução inicial de maneira construtiva, priorizando vértices com menor número de adjacências. A partir da solução inicial, a metaheurística *Iterated Local Search* é empregada com diferentes buscas locais e um método de perturbação para obtenção de refinamento da solução. Foram considerados as instâncias artificiais produzidas por Haddadi et al. [2015] e seus resultados para a avaliação da performance do método proposto. A metaheurística encontrou melhores soluções para as instâncias densas, ou seja, matrizes com grande quantidade de elementos não nulos. Para instâncias pouco densas, o método não conseguiu se aproximar dos resultados da referência. O algoritmo produzido, por ser uma metaheurística, é executado em tempo consideravelmente maior que a heurística *local improvement* de Haddadi et al. [2015].

Há na literatura diversos problemas combinatórios envolvendo matrizes binárias. A extensa pesquisa de Dom [2009] apresenta esta diversidade, mostrando suas correlações. É válido destacar que há também problemas idênticos sendo estudados sob nomes e contextos diferentes, com pesquisas não possuindo conhecimento das partes. De maneira similar Linhares and Yanasse [2002] relacionaram diferentes problemas modelados em matrizes binárias. Neste trabalho, estabeleceu-se a equivalência entre alguns problemas, como MOSP e GMLP, entretanto, foi demonstrado que o MDP não é equivalente aos problemas MOSP, GMLP, MORP e ao Problema de Minimização de Trocas de Ferramentas (*Minimization of Tool Switches Problem*, MTSP), este último definido como o problema de minimizar a quantidade de operações de troca de ferramentas de uma máquina flexível em uma linha de produção.

O trabalho de Chakhlevitch et al. [2013] contempla o Problema do Aumento de Uns Consecutivos (*Consecutive Ones Augmentation Problem*, C1AP) que trata minimizar a quantidade de zeros entre uns em matrizes binárias. Foram introduzidos três algoritmos: um algoritmo exato de busca extensiva e duas heurísticas construtivas. A primeira heurística, denominada *GRIN*, descreve uma inserção gulosa (*GReedy INsertion*), onde é inserida uma nova linha à solução por vez, analisando a posição da inserção que menos aumenta a função objetivo. A segunda heurística, *MAZE*, constrói a solução pela primeira e última linha da matriz, inserindo demais linhas entre as mesmas visando maximizar o número de zeros nas extremidades verticais da matriz (*MAximizing the number of end-ZEros*). Para a análise dos três métodos, foram criadas 50 instâncias artificiais aleatórias para cada diferente tamanho de matriz, variando de 4×4 para 10×1023. Também, foram produzidas 5 instâncias de dimensões com 25 colunas e 20000 linhas. O algoritmo exato possui complexidade de tempo $O(mn(n!)^2)$ para m linhas e n colunas da matriz, respectivamente. Dada sua complexidade, este algoritmo foi executado apenas nas instâncias com 8 ou menos colunas, determinando seus resultados ótimos. Por fim, as heurísticas *GRIN* e *MAZE* foram comparadas entre si em todas as instâncias, nas quais o método *MAZE* obteve resultados majoritariamente melhores em relação ao *GRIN*, produzindo soluções ótimas globais em todas as instâncias com resultados ótimos conhecidos. É importante destacar que o C1AP é o mesmo problema estudado pelo MORP, que busca a redução do tempo entre a produção da primeira peça até a requisição da última peça de um mesmo tipo para todas as peças em uma linha de produção. Estes são exemplos de problemas idênticos porém com estudos não interligados até o momento deste trabalho.

Capítulo 4

Fundamentação Teórica

Este capítulo aborda a descrição detalhada do COB, criando uma rígida base de conceitos necessária para produção dos métodos computacionais deste trabalho. Também são descritos dois problemas relacionados, que, por possuírem forte relação ao COB, têm seus características específicas levadas em consideração para a elaboração do algoritmo.

4.1 Problema dos Blocos de Uns Consecutivos

Formalmente, o COB considera uma matriz binária $A_{m\times n}$, i.e., $A = \{a_{ij}\}$, com $a_{ij} \in \{0,1\}$, $i = \{1,2,...,m\}$ e $j = \{1,2,...,n\}$. Em uma dada linha de A, uma sequência contígua de elementos não nulos é denominada *bloco de uns consecutivos*. A existência de um elemento nulo entre dois blocos de uns consecutivos caracteriza uma *descontinuidade*. A existência de um único bloco de uns consecutivos em uma linha de A implica na inexistência de descontinuidades. De forma análoga, a existência de dois ou mais blocos de uns consecutivos em uma mesma linha implica na existência de uma ou mais descontinuidades, respectivamente.

Uma matriz $A_{6\times 6}$, exemplo de uma instância do COB, é apresentada a seguir. Nesta matriz, a linha 1 contém dois blocos de uns, separados por uma descontinuidade presente na coluna b. Na linha 4 há três blocos – o primeiro na coluna a, outro em d e o útlimo em f. No total, a matriz A possui 11 blocos de uns e 5 descontinuidades, sendo estas últimas destacadas em negrito.

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{pmatrix} 1 & \mathbf{0} & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & \mathbf{0} & \mathbf{0} & 1 & \mathbf{0} & 1 \\ 0 & 1 & \mathbf{0} & \mathbf{0} & 1 & 0 \\ 0 & 0 & 1 & \mathbf{0} & \mathbf{0} & 1 \end{pmatrix} \end{array}$$

Uma solução para o COB é representada por uma permutação π das colunas da matriz de entrada A. Desta forma, é possível determinar a matriz permutação A^{π}, que possui as colunas de A na ordem estabelecida por π. São apresentados dois exemplos de solução para a matriz A. A permutação da matriz A^{π_1} é representada pela sequência de colunas $\pi_1 = [b,a,c,e,d,f]$ e a permutação da matriz A^{π_2} pela sequência $\pi_2 = [d,f,c,a,b,e]$. Nota-se que em ambas as matrizes produzidas o número de descontinuidades se alterou, havendo três na matriz A^{π_1} e duas na matriz A^{π_2}. Sendo n o número de colunas da matriz binária, o número total de soluções existentes é de $\frac{n!}{2}$, levando em conta que permutações em ordem inversa produzem solução idêntica à sua contraparte.

$$A^{\pi_1} = \begin{array}{cc} & \begin{matrix} b & a & c & e & d & f \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & \mathbf{0} & \mathbf{0} & 1 & 1 \\ 1 & \mathbf{0} & \mathbf{0} & 1 & 0 & 0 \\ 0 & 0 & 1 & \mathbf{0} & \mathbf{0} & 1 \end{pmatrix} \end{array} \qquad A^{\pi_2} = \begin{array}{cc} & \begin{matrix} d & f & c & a & b & e \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 \\ 1 & 1 & \mathbf{0} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

No trabalho de Linhares and Yanasse [2002], uma equação que calcula o número de inversões dos valores da matriz binária é utilizada para fazer uma aproximação do número de descontinuidades. Este método, aqui chamado A_{COB} é válido para simular o número de interrupções, dado que A_{COB} é proporcional à quantidade de fato de blocos de 0s entre 1s. Esta equação é definida abaixo.

$$A_{COB} = \sum_{j=1}^{m} \sum_{i=1}^{n} a_{ij}(1 - a_{i-1j})$$

Considerando que o objetivo seja o de minimizar o número de blocos de 0s entre 1s, a função objetivo deste problema é descrita como calcular a permutação $\pi \in \Pi$ que produza a matriz A^{π} com o menor número de inversões entre todas as possibilidades Π, ou seja:

$$\min_{\pi \in \Pi} \quad A^{\pi}_{COB}$$

Sendo n o número de colunas da matriz binária, o número total de permutações existentes é de $\frac{n!}{2}$, levanto em conta que permutações em ordem inversa produzem solução idêntica à sua contraparte.

4.2 Problema de Minimização de Blocos Circulares

O *Circular Block Minimization* (*Circ*BM) também é um problema com o objetivo de minimizar o número de descontinuidades, ou seja, elementos nulos entre elementos não nulos, entretanto este apresenta a peculiaridade de que os elementos da última coluna da matriz são considerados adjacentes aos elementos na primeira coluna de suas respectivas linhas. Em outras palavras, as colunas são consideradas dispostas circularmente.

Esta característica adicional reduz o número de soluções únicas para o problema em relação ao COB, possuindo $\frac{(n-1)!}{2}$. Este valor é explicado da seguinte forma: dada uma instância de matriz binária, é possível deslocar, por exemplo, a primeira coluna da matriz para a última posição da matriz, empurrando todas as demais colunas para esquerda, sem que seu número de descontinuidades circulares seja alterada. Como este processo de deslocar a primeira coluna para a última posição da matriz pode ser realizada n vezes até que a matriz volte à seu estado original, é feita a divisão de $\frac{n!}{2}$ por n, obtendo este número reduzido de soluções. Este deslocamento é exemplificado pela matriz abaixo, produto da deslocação da coluna a para a última posição em relação à matriz A da Seção 4.1.

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}
\begin{array}{c}
\begin{array}{cccccc} b & c & d & e & f & a \end{array} \\
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0
\end{pmatrix}
\end{array}
$$

4.3 Problema do Caixeiro Viajante *Hamming*

O Problema do Caixeiro Viajante (*Traveling Salesman Problem*, TSP) é caracterizado por um conjunto de n cidades e um conjunto D de distâncias entre todas as cidades. Em sua versão de otimização, o problema é $\mathcal{NP}$-Difícil e consiste em determinar uma rota de visitação das cidades com menor distância total e de forma a visitar cada cidade exatamente uma vez, retornando à cidade inicial no fim da rota. Em outras palavras, uma solução do TSP pode ser entendida como uma permutação circular do conjunto de cidades de menor custo. Os tipos de distância consideradas caracterizam versões diferentes do TSP, e podem ser rodoviárias, euclidianas, *Manhattan* ou outras, dependendo da aplicação do problema.

A distância *Hamming*, que caracteriza o HTSP, é calculada pelo número de caracteres em que duas cadeias se diferenciam entre si, considerando duas cadeias de mesmo tamanho. Em outras palavras, essa métrica mede o mínimo numero de substituições de caracteres para transformar uma cadeia em uma outra. A distância *Hamming* é naturalmente simétrica, i.e., a distância de uma cadeia para outra é idêntica à distância inversa. Adicionalmente, a distância entre duas cadeias idênticas é nula.

Contextualizando para o HTSP, cada cidade é uma cadeia binária de caracteres e a distância de uma cidade para a outra é calculada pela métrica *Hamming*. O conjunto de distâncias entre as cidades pode ser armazenada em formato de matriz, onde o valor que ocupa a célula da posição $(1,2)$ representa a distância *Hamming* entre as cadeias 1 e 2, por exemplo.

Abaixo demonstra-se a matriz resultante do cálculo de distâncias para a matriz A dos exemplos anteriores. Para este exemplo, é considerada como uma cidade cada coluna de A.

$$
D = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array}
\begin{array}{c}
\begin{array}{cccccc} a & b & c & d & e & f \end{array} \\
\begin{pmatrix}
0 & 3 & 3 & 3 & 5 & 3 \\
3 & 0 & 4 & 4 & 2 & 4 \\
3 & 4 & 0 & 4 & 4 & 2 \\
3 & 4 & 4 & 0 & 2 & 2 \\
5 & 2 & 4 & 2 & 0 & 4 \\
3 & 4 & 2 & 2 & 4 & 0
\end{pmatrix}
\end{array}
$$

Capítulo 5

Desenvolvimento

Neste capítulo é abordada a modelagem utilizada para o Problema de Blocos de Uns Consecutivos, dividida em dois breves passos. Na sequência, é apresentada brevemente a ferramenta utilizada para obtenção da solução exata para o problema, a partir da modelagem empregada .

5.1 Modelagem

Nesta seção é apresentada a redução do COB ao HTSP, conforme introduzida e provada parcial ou totalmente por diferentes autores [Kou, 1977, Alizadeh et al., 1995, Greenberg and Istrail, 1995, Haddadi and Layouni, 2008]. Esta redução é realizada em tempo determinístico polinomial, de modo que o COB e o HTSP podem ser considerados problemas equivalentes. Desta maneira, um método de solução para o HTSP pode ser aplicado de maneira equivalente para solução do COB.

A redução é realizada em duas etapas. Na primeira delas, o COB é reduzido ao *Circ*BM, o Problema de Minimização de Blocos Circulares. Na segunda etapa, o *Circ*BM é reduzido ao HTSP, o Caixeiro Viajante *Hamming*.

5.1.1 Redução do COB ao *Circ*BM

Para esta transformação, modifica-se a matriz binária A com a inserção de uma coluna artificial preenchida apenas com elementos nulos fixa à esquerda, i.e., $B = [0 \mid A]$. Desta forma, a matriz $A_{m\times n}$ dá origem a uma matriz $B_{m\times n+1}$, que possui a primeira coluna fixa, impedindo a criação de blocos circulares. A solução do *Circ*BM para a instância $B_{m\times n+1}$ é equivalente à solução do COB para a instância $A_{m\times n}$, conforme provado por Haddadi and Layouni [2008]. A matriz B ilustra a redução do COB ao *Circ*BM a partir da matriz A utilizada nos exemplos anteriores. A coluna adicional é apresentada em destaque.

$$
B = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}
\begin{array}{c}
\begin{array}{ccccccc} \boldsymbol{0} & a & b & c & d & e & f \end{array} \\
\begin{pmatrix}
\mathbf{0} & 1 & 0 & 1 & 0 & 0 & 0 \\
\mathbf{0} & 1 & 1 & 0 & 0 & 0 & 0 \\
\mathbf{0} & 0 & 0 & 0 & 1 & 1 & 0 \\
\mathbf{0} & 1 & 0 & 0 & 1 & 0 & 1 \\
\mathbf{0} & 0 & 1 & 0 & 0 & 1 & 0 \\
\mathbf{0} & 0 & 0 & 1 & 0 & 0 & 1
\end{pmatrix}
\end{array}
$$

5.1.2 Redução do *Circ*BM ao HTSP

Nesta redução, considera-se cada uma das $n+1$ colunas da matriz B como uma cidade do HTSP. A matriz de distâncias D entre as cidades são dadas pelas distâncias *Hamming* entre as colunas de B, i.e., considera-se cada coluna como uma cadeia binária de caracteres e a distância é dada pelo número de não correspondências entre os elementos de cada uma de suas linhas. Intuitivamente, quanto menor a distância *Hamming* de duas colunas de B, maior o desejo de produzir uma permutação que junte estas colunas lado a lado na solução.

A matriz de distâncias D pode ser construída, a partir da matriz B, pelo seguinte cálculo: $D = B^T(M^1 - B) + (M^1 - B)^T B$ [Haddadi and Layouni, 2008], na qual M^1 representa uma matriz $m \times n+1$ constituída inteiramente de 1s, e ^{T} indica a função de transposição de matriz. Este algoritmo possui complexidade de tempo $O(n^3)$, por se tratar de uma sequência de multiplicação de matrizes. Contudo, é possível implementar o cálculo de distâncias *Hamming* em complexidade de tempo $O(n^2)$.

A solução do HTSP para a instância $(n+1, D)$ é equivalente à solução do *Circ*BM para a instância B, conforme provado por [Kou, 1977, Alizadeh et al., 1995, Greenberg and Istrail, 1995]. Por transitividade, a solução também é equivalente à solução do COB para a instância A. A matriz D ilustra a redução do *Circ*BM ao HTSP, considerando a matriz B.

$$D = \begin{array}{c} \\ \mathit{0} \\ a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{ccccccc} \mathit{0} & a & b & c & d & e & f \end{array} \\ \begin{pmatrix} 0 & 3 & 2 & 2 & 2 & 2 & 2 \\ 3 & 0 & 3 & 3 & 3 & 5 & 3 \\ 2 & 3 & 0 & 4 & 4 & 2 & 4 \\ 2 & 3 & 4 & 0 & 4 & 4 & 2 \\ 2 & 3 & 4 & 4 & 0 & 2 & 2 \\ 2 & 5 & 2 & 4 & 2 & 0 & 4 \\ 2 & 3 & 4 & 2 & 2 & 4 & 0 \end{pmatrix} \end{array}$$

5.2 Resolvedor Concorde

Dada a correspondência entre COB e HTSP, é possível utilizar qualquer algoritmo de resolução do TSP simétrico para produzir uma solução para o COB. A literatura de tais algoritmos é vasta, possuindo métodos de aproximação [Christofides, 1976], heurísticas sub-ótimas de bom desempenho [Lin and Kernighan, 1973] e métodos exatos [Held and Karp, 1962].

O método considerado estado-da-arte por mais de 15 anos entre os métodos exatos para solução do TSP é o resolvedor Concorde [Applegate et al., 2006]. Este resolvedor é escrito na linguagem ANSI C, e é disponível gratuitamente para fins acadêmicos, necessitando de uma licença para seu uso comercial. Sua biblioteca é constituída por mais de 700 funções, permitindo sua integração a códigos de algoritmos específicos de problemas relacionados ao TSP.

Desta forma, utiliza-se neste trabalho a modelagem do COB como o HTSP e resolve-se este último problema por meio do resolvedor exato Concorde e, posteriormente, a obtenção do número de descontinuidades. Embora alguns trabalhos da literatura tenham utilizado anteriormente esta modelagem em determinados contextos, as soluções foram geradas por meio de heurísticas.

O resolvedor Concorde produz soluções ótimas para o TSP devido à utilização de programação linear inteira, programação dinâmica, busca de limitantes inferiores para o valor solução e diversas heurísticas auxiliares para obtenção de limitantes superiores. Dentre os métodos empregados, destacam-se a produção de planos de cortes e relaxações, a triangulação de *Delaunay* e algoritmos para determinar árvores geradoras mínimas. Para a fase de produção de planos de cortes, é necessária a utilização de um resolvedor de programação linear auxiliar. O Concorde possui suporte para dois resolvedores: o *QSopt* e o CPLEX.

A solução ótima obtida pelo Concorde para o HTSP consiste em uma permutação π de $n+1$ elementos. A solução ótima π^* para o COB é obtida a partir de π pela remoção do elemento referente à coluna de elementos nulos da matriz B. Por fim, com a permutação π^*, agora com n elementos, permuta-se as colunas da matriz original A em razão da ordem estabelecida por π^*. Como ilustração, a solução ótima da instância do HTSP representada pela matriz D é $\pi = [\mathit{0}, d, e, b, a, f, c]$, que produz $\pi^* = [d, e, b, a, f, c]$, e em sequência a matriz abaixo.

$$A^{\pi^*} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} d & e & b & a & f & c \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 & \mathbf{0} & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & \mathbf{0} & \mathbf{0} & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

Capítulo 6

Experimentos, Resultados e Discussão

Os experimentos foram realizados em dois ambientes computacionais distintos. A redução do COB ao HTSP foi executada em um computador com processador *Intel i7-3520M* 2,90GHz, com 16GB de memória RAM e sistema operacional *Windows 10*. O algoritmo foi codificado na linguagem C++ e compilado com *gcc* 6.2.0 e *flag* de otimização -O3. Para geração das soluções ótimas para HTSP foi utilizada a infraestrutura computacional e a implementação do resolvedor Concorde disponíveis no servidor de otimização NEOS [Mittelmann, 2016], utilizando o resolvedor de programação linear CPLEX e o parâmetro "-s 99". A infraestrutura consiste em três computadores: *Thales* (processador *Intel i7-X990* com 3,47GHz, seis *cores* e 24GB de memória RAM); *Neos* (*Intel i7-7700K* de 4,20GHz, quatro *cores* e 32GB de memória RAM) e *Athene* (*Intel Xeon X5690* de 3,47GHz, seis *cores* e 32GB de memória RAM). Embora os computadores sejam *multicore*, o resolvedor Concorde foi executado de maneira serial.

São considerados os dois conjuntos de instâncias propostos por Haddadi et al. [2015]. O primeiro conjunto contém 5 instâncias reais advindas de um problema de alocação de estações de companhia ferroviária alemã. O segundo conjunto contém 45 instâncias geradas aleatoriamente, divididas em 9 grupos de 5 instâncias cada, nomeadas de *A* a *I*, tal que cada grupo se difere pela dimensão da matriz e pela quantidade média de uns em cada linha da matriz.

As Tabelas 6.1 e 6.2 apresentam os resultados ótimos para as instâncias consideradas, exceto por quatro. São apresentados os dados de cada instância, como identificador e dimensões. São também apresentados o valor ótimo para o número de descontinuidades (OPT) e o tempo de execução do método exato, em segundos. Foi imposto um limite de 14400 segundos (4 horas) para solução de cada instância. Os quatro resultados marcados com um asterisco indicam os casos em que este limite foi atingido e, portanto, as soluções não são comprovadamente ótimas.

Tabela 6.1: Resultados computacionais.

Instância	Dimensões	OPT	Tempo (s)
RCOV$_{1km}$	757×707	757	13,48
RCOV$_{2km}$	1196×889	1196	14,22
RCOV$_{3km}$	1419×886	1420	12,03
RCOV$_{5km}$	1123×593	1129	6,89
RCOV$_{10km}$	275×165	276	1,52
A1		210	63,00
A2		216	1,83
A3	100×200	204	45,55
A4		233	0,41
A5		226	1,74

Tabela 6.2: Resultados computacionais (cont.).

Instância	Dimensões	OPT	Tempo (s)
B1		632	1,03
B2		629	0,91
B3	100×200	619	0,96
B4		638	20,20
B5		613	0,94
C1		1245	0,92
C2		1242	1,29
C3	100×200	1247	103,37
C4		1253	1,03
C5		1311	111,68
D1		491	2,65
D2		502	4,90
D3	100×500	445	3,18
D4		472	2,83
D5		454	2,54
E1		1412	3,98
E2		1397	7,72
E3	100×500	1394	4,61
E4		1386	7,22
E5		1430	2,61
F1		2955	4,92
F2		3002	5,21
F3	100×500	2999	2,25
F4		3009	9,34
F5		3021	3,98
G1		893	15,89
G2		884	7,95
G3	100×1000	912	10,23
G4		932	7,56
G5		763	9,53
H1		2708	7,05
H2		2597*	14400
H3	100×1000	2688*	14400
H4		2516	14,66
H5		2680	6,82
I1		5714	15,06
I2		5767	6,35
I3	100×1000	5653*	14400
I4		5674	20,23
I5		5676*	14400

* Tempo limite de 4 horas excedido.

Os dados reportados comprovam a eficiência deste método de resolução da maior parte das instâncias consideradas. A redução do COB ao HTSP para cada instância é obtida em tempo desprezível, menor do que 0,3 segundos em média. A este tempo adiciona-se o tempo de execução do método exato, menor do que 10 segundos na maioria das instâncias. Em apenas sete instâncias, o tempo de execução supera um minuto. Dentre estes casos, há quatro instâncias nas quais o método exato excede o tempo limite imposto, reportando apenas o resultado parcial encontrado dentro do limite.

A Tabela 6.3 apresenta a comparação entre os resultados médios obtidos neste trabalho e os melhores resultados (BKS) encontrados na literatura até então, determinados por Haddadi et al. [2015] e Nascimento and Carvalho [2017]. É apresentada adicionalmente a distância percentual entre as duas soluções (*gap*), calculada como $gap = \frac{BKS-OPT}{OPT} \times 100$.

Tabela 6.3: Análise comparativa de performance.

Grupo	Dimensões	OPT	BKS	*gap* (%)
A	100×200	217,8	253,0[a]	16,16
B		626,2	671,6[b]	7,25
C		1259,6	1307,2[b]	3,78
D	100×500	472,8	552,0[a]	16,75
E		1403,8	1576,2[b]	12,28
F		2997,2	3162,2[b]	5,51
G	100×1000	876,8	1072,4[a]	22,31
H		2637,8	3067,2[b]	16,28
I		5696,8	6109,6[b]	7,25

[a] Haddadi et al. [2015]
[b] Nascimento and Carvalho [2017]

As melhores soluções heurísticas da literatura estão em média 11,95% dos resultados obtidos neste trabalho. Mesmo nos casos em que a solução obtida não é comprovadamente ótima, há distância considerável. Particularmente, as distâncias são maiores nas instâncias consideradas esparsas e menores nas consideradas densas. Este resultado evidencia a dificuldade das heurísticas em resolver o COB em matrizes esparsas, i.e., com menor quantidade de elementos não nulos e menor número de descontinuidades, consequentemente.

Não é possível uma comparação justa dos tempos de execução, dado que os métodos comparados foram executados em arquiteturas computacionais diferentes. Entretanto, é possível notar que o trabalho de Nascimento and Carvalho [2017] reporta tempos de execução médios da ordem de horas para solução das instâncias dos grupos F, H e I, ao passo que Haddadi et al. [2015] reporta tempo de execução máximo abaixo de 30 segundos.

Capítulo 7

Conclusões

Este trabalho introduz um método exato para o Problema de Minimização Blocos de Uns Consecutivos (ou COB, do inglês *Consecutive Ones Block Minimization*), um problema $\mathcal{NP}$-Difícil estudado em diferentes áreas do conhecimento, de maneira independente. O método exato consiste na redução do COB ao Problema de Minimização de Minimização de Blocos Circulares, posteriormente reduzido ao Problema do Caixeiro Viajante com distâncias *Hamming* (ou *Hamming distance Traveling Salesman Problem*, HTSP). Após realizar as reduções, o resolvedor exato Concorde é aplicado à solução do HTSP, cuja solução ótima é finalmente transformada na solução ótima do COB. Os experimentos computacionais envolveram 50 instâncias dos únicos dois conjuntos de instâncias *benchmark* disponíveis na literatura, que inclui instâncias reais. Em um baixo tempo de execução, foi possível obter as até então inéditas soluções ótimas para estas instâncias. Os resultados dos métodos considerados estado da arte foram comparados a estas novas soluções, a fim de se analisar o impacto das mesmas. Os trabalhos futuros serão concentrados em gerar novos conjuntos de instâncias e desenvolver um método do tipo *matheuristic* para os casos em que o método exato não seja suficiente.

Referências Bibliográficas

Farid Alizadeh, Richard M Karp, Lee Aaron Newberg, and Deborah K Weisser. Physical mapping of chromosomes: A combinatorial problem in molecular biology. *Algorithmica*, 13(1): 52–76, 1995.

David Applegate, Ribert Bixby, Vasek Chvatal, and William Cook. Concorde TSP solver, 2006. URL `http://www.math.uwaterloo.ca/tsp/concorde/`.

Marcos Arenales, Vinicius Armentano, Reinaldo Morabito, and Horacio Yanasse. *Pesquisa operacional: para cursos de engenharia*. Elsevier Brasil, 2017.

Konstantin Chakhlevitch, Celia A Glass, and Natalia V Shakhlevich. Minimising the number of gap-zeros in binary matrices. *European Journal of Operational Research*, 229(1):48–58, 2013.

Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group, 1976.

Michael Dom. *Recognition, Generation, and Application of Binary Matrices with the Consecutive Ones Property*. Cuvillier, 2009.

RG Dyson and AS Gregory. The cutting stock problem in the flat glass industry. *Journal of the Operational Research Society*, 25(1):41–53, 1974.

Michael R Garey and David S Johnson. *Computers and intractability*. WH Freeman New York, 1979.

Marco Cesar Goldbarg and Henrique Pacca L Luna. *Otimização combinatória e programação linear: modelos e algoritmos*. Elsevier, 2005.

David S Greenberg and Sorin Istrail. Physical mapping by STS hybridization: Algorithmic strategies and the challenge of software evaluation. *Journal of Computational Biology*, 2(2): 219–273, 1995.

Salim Haddadi. A note on the NP–hardness of the consecutive block minimization problem. *International Transactions in Operational Research*, 9(6):775–777, 2002.

Salim Haddadi and Zoubir Layouni. Consecutive block minimization is 1.5-approximable. *Information Processing Letters*, 108(3):132–135, 2008.

Salim Haddadi, S Chenche, Meryem Cheraitia, and F Guessoum. Polynomial-time local-improvement algorithm for consecutive block minimization. *Information Processing Letters*, 115(6):612–617, 2015.

Michael Held and Richard M Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962.

David Johnson, Shankar Krishnan, Jatin Chhugani, Subodh Kumar, and Suresh Venkatasubramanian. Compressing large boolean matrices using reordering techniques. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 13–23. VLDB Endowment, 2004.

David Kendall. Incidence matrices, interval graphs and seriation in archeology. *Pacific Journal of mathematics*, 28(3):565–570, 1969.

Lawrence T Kou. Polynomial complete consecutive information retrieval problems. *SIAM Journal on Computing*, 6(1):67–75, 1977.

Shen Lin and Brian W Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2):498–516, 1973.

Alexandre Linhares and Horacio Hideki Yanasse. Connections between cutting-pattern sequencing, VLSI design, and flexible machines. *Computers & Operations Research*, 29(12): 1759–1772, 2002.

Oli BG Madsen. Glass cutting in a small firm. *Mathematical Programming*, 17(1):85–90, 1979.

Oli BG Madsen. An application of travelling-salesman routines to solve pattern-allocation problems in the glass industry. *Journal of the Operational Research Society*, 39(3):249–256, 1988.

Hans Mittelmann. NEOS server: concorde, 2016. URL `https://neos-server.org/neos/solvers/co:concorde/TSP.html`.

Luis Henrique Leão Nascimento and Marco Antonio Moreira Carvalho. Uma heurística aplicada à uniformidade das características físicas de produtos. In *Anais do XLIX Simpósio Brasileiro de Pesquisa Operacional*, 2017.