JORDI ALVES REINSMA

Orientador: Marco Antonio Moreira de Carvalho

# APLICAÇÃO DE MÉTODOS EXATO E HEURÍSTICO PARA RESOLUÇÃO DO PROBLEMA DE MINIMIZAÇÃO DE BLOCOS DE UNS CONSECUTIVOS

Ouro Preto

Julho de 2018

Universidade Federal de Ouro Preto
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

# APLICAÇÃO DE MÉTODOS EXATO E HEURÍSTICO PARA RESOLUÇÃO DO PROBLEMA DE MINIMIZAÇÃO DE BLOCOS DE UNS CONSECUTIVOS

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

JORDI ALVES REINSMA

Ouro Preto
Julho de 2018

UNIVERSIDADE FEDERAL DE OURO PRETO

FOLHA DE APROVAÇÃO

# Aplicação de Métodos Exato e Heurístico para Resolução do Problema de Minimização de Blocos de Uns Consecutivos

JORDI ALVES REINSMA

Monografia defendida e aprovada pela banca examinadora constituída por:

Dr. Marco Antonio Moreira de Carvalho – Orientador
Universidade Federal de Ouro Preto

Dr. Marcone Jamilson Freitas Souza
Universidade Federal de Ouro Preto

Dr. Túlio Ângelo Machado Toffolo
Universidade Federal de Ouro Preto

Ouro Preto, Julho de 2018

# Resumo

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 para minimizar as descontinuidades de blocos de 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 estudo reporta a aplicação de dois métodos para abordagem do COB. O primeiro consiste em um método exato baseado em resultados parciais anteriores presentes na literatura. Nele, o COB é reduzido a uma versão do conhecido Problema do Caixeiro Viajante (*Traveling Salesman Problem*, TSP) e resolvido de maneira exata pelo resolvedor Concorde, atual estado da arte para solução do TSP. O segundo método utiliza a mesma redução e a aborda de maneira heurística para gerar soluções para as instâncias em que o algoritmo exato não conseguir comprovar a otimalidade das soluções em tempo hábil. Nos experimentos computacionais, foram geradas pela primeira vez as soluções ótimas para o único conjunto de instâncias *benchmark* da literatura, as quais foram comparadas aos resultados do estado da arte do COB. Adicionalmente, foi gerado um novo conjunto de instâncias, gerando uma nova bateria de testes.

# Abstract

The Consecutive Ones Block (COB) Minimization is an $\mathcal{NP}$-Hard problem which consists in finding a permutation of the columns of a binary matrix such that the discontinuities of its blocks of nonzero elements are minimized in each row. This problem is independently studied under different names in different knowledge areas, such as operations research, archaeology, microbiology, information systems and mathematics. This study reports the application of two methods for approaching the COB. The first is an exact method based on previous partial results found in the literature. The COB problem is reduced to a version of the well-known Traveling Salesman Problem (TSP) and solved to optimality by the Concorde solver, the current state of the art for solving the TSP. The second method uses the same reduction and heuristically generate solutions for the instances in which the exact algorithm does not prove the optimality of the solution. Computational experiments generated for the first time the optimal solutions for the only set of benchmark instances available in the literature and compared them to the the state of the art results for the COB. Additionally, a new set of instances is proposed, alongside a new battery of tests.

# Sumário

# Lista de Tabelas

# Siglas

**CBM** *Consecutive Blocks Minimization* (Minimização de Blocos Consecutivos).

**COB** *Consecutive Ones Block Minimization* (Minimização de Blocos de Uns Consecutivos).

**GMLP** *Gate Matrix Layout Problem* (Problema de Determinação de Leiaute de Matrizes de Portas).

**MDP** *Minimization of Discontinuities Problem* (Problema de Minimização de Descontinuidades).

**MORP** *Minimization of Order Spread Problem* (Problema de Minimização de Espalhamento de Ordens de Compra).

**MOSP** *Minimization of Open Stacks Problem* (Problema de Minimização de Pilhas Abertas).

**TSP** *Traveling Salesman Problem* (Problema do Caixeiro Viajante).

# Capítulo 1

# Introdução

Dentre os vários problemas já abordados na literatura de otimização inteira, há uma série destes, encontrados principalmente em contextos industriais, que são abstraídos e modelados em matrizes binárias. Particularmente, alguns desses problemas possuem natureza permutacional, ou seja, sua solução é determinada pela seleção ordenada dos elementos da matriz a partir de um critério de comparação específico do problema.

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 minimizadas; o MORP está relacionado com o ritmo de atendimento a pedidos de compra e o MOSP está relacionado com a minimização da utilização de estoque intermediário em indústria de corte de matéria-prima.

Da mesma maneira, problemas permutacionais modelados em matrizes binárias podem ser encontrados no âmbito da ciência da computação. Citam-se o Problema de Minimização de Banda (*Bandwidth Minimization Problem*), que consiste em reduzir a distância entre os elementos não-nulos e a diagonal principal, útil para resolução de sistemas lineares; o Problema de Largura de Corte em Grafos (*Cutwidth*), relacionado à organização linear dos vértices de um grafo e o Problema de Aumento de Uns Consecutivos (*Consecutive Ones Augmentation Problem*), que envolve minimizar a distância entre os elementos não-nulos da matriz de uma maneira geral, também aplicável na resolução de sistemas lineares.

O objeto de análise deste trabalho é 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 blocos de uns nas linhas da matriz, utilizando-se de permutações das colunas. Define-se *blocos de uns* como uma sequência de elementos não nulos consecutivos em uma mesma linha de uma matriz binária, na qual a presença de elementos nulos interrompe tal sequência.

Neste trabalho é discutida a relação do COB com alguns dos demais problemas citados anteriormente, assim como a aplicação direta do COB na indústria e em outros contextos. Para a resolução deste problema, é proposta a aplicação de um método exato a partir de uma modelagem clássica da literatura para o COB. Desta maneira, é possível gerar os resultados ótimos para instâncias da literatura pela primeira vez. Entretanto, o método exato eventualmente não consegue comprovar a otimalidade em tempo hábil para alguns dos problemas considerados. Sendo assim, propõe-se adicionalmente a aplicação de um método heurístico para a solução do COB a partir da mesma modelagem clássica. Ambos os métodos têm seu desempenho analisado utilizando-se instâncias da literatura e comparados com os métodos considerados o estado da arte.

## 1.1 Motivação

O COB é um problema da classe $\mathcal{NP}$-Difícil (Garey e Johnson, 1979), possuindo, portanto, uma relevante 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 reduzir a quantidade de "espaços" entre "elementos" também ocorre na arqueologia (Kendall, 1969) – com a rotulação de sítios, microbiologia (Alizadeh et al., 1995) – com a reconstrução de moléculas de DNA a partir de clones dos fragmentos da molécula, compressão de estruturas de dados (Johnson et al., 2004), na organização de arquivos (Kou, 1977) e na engenharia de produção (Dyson e Gregory, 1974) – na minimização de interrupções na fabricação de produtos. A multidisciplinaridade do problema também é um fator motivacional para sua abordagem.

## 1.2 Objetivos

Em linhas gerais, o objetivo deste trabalho é, a partir de algoritmos eficientes e consistentes que possam ser utilizados no contexto do problema abordado, produzir soluções ótimas ou próximas da solução ótima, inicialmente considerando problemas específicos, mas com a possibilidade de generalização. São objetivos específicos:

1. Realizar pesquisa para geração de embasamento teórico e revisão bibliográfica sobre o tema tratado;

2. Realizar pesquisa sobre a modelagem do Problema de Minimização de Blocos de Uns Consecutivos como o Problema do Caixeiro Viajante.

3. Implementar a modelagem e aplicar pela primeira vez um método exato na resolução do problema sob estudo;

4. Avaliar o método utilizado considerando dados reais e também com problemas teste publicamente disponíveis;

5. Utilizar método heurístico para avaliação de instâncias patológicas, ou seja, instâncias em que o método exato não consegue provar a otimalidade da solução em tempo hábil;

6. Ampliar o conjunto de problemas testes com soluções ótimas conhecidas; e

7. Produzir trabalhos para serem publicados em periódicos e eventos científicos.

## 1.3 Organização do Trabalho

O restante deste trabalho está organizado como segue. O Capítulo 2 apresenta a literatura relevante ao problema estudado. Em seguida, é abordada a fundamentação teórica sobre o COB, no Capítulo 3. O Capítulo 4 detalha a modelagem utilizada, assim como as ferramentas usadas para apoiar a resolução do problema. Em seguida, são apresentados os resultados computacionais advindos da aplicação dos métodos descritos, no Capítulo 5. Por fim, as considerações finais deste trabalho são expostas no Capítulo 6.

# Capítulo 2

# Revisão da Literatura

O Problema de Minimização de Blocos de Uns Consecutivos é estudado há mais de 40 anos, sendo aplicado em diferentes contextos e resolvidos por diferentes métodos. 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. Também é discutida a relação entre o COB e estes outros problemas permutacionais que também envolvem matrizes binárias.

A primeira pesquisa analisada relativa ao COB é a de Dyson e 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 com o COB, este último sendo abstraído como Problema de Alocação de Padrões (*Pattern Allocation Problem*). Neste problema, uma matriz binária indica quais padrões de corte produzem quais tipos de peças de vidro, e a ideia é minimizar as descontinuidades na produção das peças, o que pode gerar diferenças nas suas tonalidade. 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 mesmo para instâncias pequenas, o TSP não possuía resolução ótima realizável em tempo hábil 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, a fim de finalizá-las rapidamente, minimizando a ocorrência de interrupções.

Kou (1977) e Garey e Johnson (1979) apresentaram provas formais de que o COB pertence à classe $\mathcal{NP}$-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 permanece sendo $\mathcal{NP}$-Difícil 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. Neste cenário, o COB é estudado sob o nome de Problema de Minimização de Descontinuidades (*Minimization of Discontinuities Problem*, MDP). 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. No referido trabalho, após a obtenção dos padrões de corte via resolução do Problema da Mochila, 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 descontinuidades na produção das peças. 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 projetada para o TSP.

Haddadi e 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 desenvolvido um algoritmo de aproximação que constrói uma solução que está ao máximo 50% distante da solução ótima. 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 realizada a transformação do CIR para uma instância simétrica do TSP. 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 dessas transformações, 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) propõem 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 do impacto de modificar soluções com complexidade de tempo $O(m)$, em que $m$ representa o número de colunas da matriz binária. Esse método, denominado $\Delta$-avaliação, constitui uma contribuição importante, uma vez que a avaliação completa de uma matriz binária exige complexidade $\Theta(mn)$. Para comparar o

desempenho 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 o desempenho do algoritmo produzido neste trabalho, comparando-o com os resultados obtidos por demais autores.

Nascimento e Carvalho (2017) introduzem uma representação em grafos do COB, estudado sob o 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 um algoritmo de busca em largura modificado 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 consideradas as instâncias artificiais produzidas por Haddadi et al. (2015) e seus resultados originais para a avaliação do desempenho do método proposto. A metaheurística encontrou melhores soluções para as instâncias densas, ou seja, matrizes com grande quantidade de elementos não nulos. Para instâncias esparsas, o método não conseguiu se aproximar dos resultados do estudo de referência.

Há na literatura diversos problemas permutacionais envolvendo matrizes binárias. A extensa pesquisa de Dom (2009) apresenta esta diversidade, mostrando suas correlações. De maneira similar, Linhares e Yanasse (2002) relacionaram diferentes problemas permutacionais modelados em matrizes binárias. Nesse trabalho, estabeleceu-se a equivalência entre alguns destes problemas, como MOSP e GMLP, entretanto. No mesmo trabalho, 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) aborda o Problema do Aumento de Uns Consecutivos (*Consecutive Ones Augmentation Problem*, ou C1AP) que visa 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, fazendo a inserção de 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 tamanho diferente de matriz, variando entre $4\times4$ e $10\times1023$. Também foram produzidas 5 instâncias de 25 colunas e

20000 linhas. O algoritmo exato possui complexidade de tempo $O(mn(n!)^2)$ para $m$ linhas e $n$ colunas da matriz. Dada sua complexidade, esse 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 conhecido como MORP, que busca minimizar o comprimento das descontinuidades, o que não implica necessariamente em eliminá-las. Esses são exemplos de problemas idênticos porém com estudos não interligados até o momento deste trabalho.

# Capítulo 3

# Fundamentação Teórica

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

## 3.1 O Problema de Minimização de 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 qualquer número de elementos não nulos, inclusive um único, é denominada *bloco de uns consecutivos*. A existência de um elemento nulo entre dois blocos de uns consecutivos caracteriza uma *descontinuidade*. De forma análoga, a existência de um único bloco de uns consecutivos em uma linha de $A$ implica na inexistência de descontinuidades. Como decorrência, 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 entre a coluna $a$ e $b$, outro em $d$ e o último em $f$. No total, a matriz $A$ possui 11 blocos de uns e 5 descontinuidades, sendo estas últimas destacadas em negrito.

Uma solução para o COB é representada por uma permutação $\pi$ das colunas da matriz de entrada $A$. Desta forma, é possível determinar a matriz permutação $A^{\pi}$, que possui as colunas de $A$ na ordem estabelecida por $\pi$. São apresentados dois exemplos de solução para a matriz $A$. A permutação da matriz $A^{\pi_1}$ é representada pela sequência de colunas $\pi_1 = [b, a, c, e, d, f]$ e a permutação da matriz $A^{\pi_2}$ pela sequência $\pi_2 = [d, f, c, a, b, e]$. Nota-se que em ambas as

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

matrizes produzidas o número de blocos consecutivos de uns e de descontinuidades se alterou, havendo três descontinuidades na matriz $A^{\pi_1}$ e duas na matriz $A^{\pi_2}$. Considerando isso, o objetivo do COB é a minimização do número de blocos de uns em uma matriz binária, por meio da permutação das suas colunas. O tamanho do espaço de soluções do COB é limitado por $O(n!)$. 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.

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

## 3.2 O Problema de Minimização de Blocos de Uns Circulares

Sendo um problema intimamente relacionado ao COB, o CIR também tem como objetivo minimizar o número de blocos de uns consecutivos em uma matriz binária. Entretanto, este problema tem 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.

Ao analisar a mesma matriz $A$ utilizada no exemplo da Seção 3.1, porém, interpretando-a de acordo com o CIR, o número de blocos consecutivos torna-se 10, considerando que os uns da primeira e última coluna da quarta linha de $A$ são adjacentes entre si, exemplificado abaixo.

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

Conforme apresentado no Capítulo 4, o COB será transformado no CIR como primeiro passo da modelagem utilizada neste trabalho.

## 3.3 O Problema do Caixeiro Viajante

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. Os tipos de distância consideradas caracterizam versões diferentes do TSP, e podem ser rodoviárias, euclidianas, *Manhattan* ou outras, dependendo de sua aplicação.

Abaixo, observa-se um exemplo de instância do TSP, uma matriz $D_{5\times5}$ de distâncias entre 5 cidades. O elemento $d_{ij}$ representa a distância entre a cidade $i$ e a cidade $j$. Nota-se que, neste exemplo, a distância entre uma cidade e outra é idêntica à distância de volta, i.e. $d_{ij} = d_{ji}$ para toda combinação de $i$ e $j$. Isto caracteriza uma instância do TSP *simétrico*.

$$D = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 30 & 60 & 90 & 50 \\ 30 & 0 & 60 & 70 & 40 \\ 60 & 60 & 0 & 40 & 80 \\ 90 & 70 & 40 & 0 & 50 \\ 50 & 40 & 80 & 50 & 0 \end{pmatrix} \end{array}$$

Uma solução para o TSP é representada por uma permutação $s$ das $n$ cidades e seu custo total de distância. O custo total da permutação $s$ é determinado pela Equação (3.1).

$$d_{s_n s_1} + \sum_{i=1}^{n-1} d_{s_i s_{i+1}} \tag{3.1}$$

Para a matriz $D$ a permutação $s = [1, 2, 5, 4, 3]$, por exemplo, é avaliada como $d_{31} + d_{12} + d_{25} + d_{54} + d_{43}$, resultando em uma rota de custo de 220 unidades.

Uma solução do TSP pode ser entendida como uma permutação circular do conjunto de cidades de menor custo. Esta característica relaciona o TSP ao CIR e é utilizada convenientemente na modelagem empregada neste trabalho.

# Capítulo 4

# Desenvolvimento

Uma forma comum de contornar as dificuldades de solucionar um problema computacional é transformando-o em um outro problema para o qual haja mais recursos disponíveis para sua abordagem. Esta transformação deve permitir que a solução obtida possa ser transformada de volta ao problema original, com a garantia de que a qualidade da solução adquirida se mantenha. No texto a seguir, a redutibilidade de um problema $A$ a um problema $B$ com preservação de aproximação em tempo polinomial (*polynomial-time approximation scheme*) é denotada por $A \leq_{\text{PTAS}} B$.

Após a descrição das reduções entre problemas relacionadas à modelagem empregada, dois resolvedores, um exato e um heurístico, são aplicados para a solução do problema resultante. Ambos os resolvedores são também brevemente descritos neste capítulo.

## 4.1 Redução do COB ao TSP

É apresentada a redução do COB ao TSP, conforme introduzida e provada parcial ou totalmente por diferentes autores (Kou, 1977; Alizadeh et al., 1995; Greenberg e Istrail, 1995; Haddadi e Layouni, 2008). Esta redução é realizada em tempo determinístico polinomial e de modo que o COB e o HSP podem ser considerados problemas equivalentes. Desta maneira, um método de solução para o TSP 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 Problema de Minimização de Blocos Circulares (*Circular Blocks Minimization*, CIR). Na segunda etapa, o CIR é reduzido ao HTSP.

### 4.1.1 Redução ao Problema de Minimização de Blocos Circulares

Conforme Haddadi e Layouni (2008), para reduzir o COB ao CIR, modifica-se a matriz binária $A$ com a inserção de uma nova coluna artificial preenchida apenas com elementos nulos,

dando origem a uma nova matriz $B$, i.e., $B = [0 \cup A]$. Desta forma, a matriz $A_{m \times n}$ dá origem a uma matriz $B_{m \times n+1}$, contendo uma coluna nula. Esta transformação é simples, assim como a demonstração de sua equivalência.

**Lema 1.** *$COB \leq_{PTAS} CIR$.*

**Demonstração.** *Sem perda de generalidade, utilizaremos como exemplo a matriz $A$. A permutação $\pi_1 = [a, f, c, d, e, b]$ é uma solução ótima de $A$ para o CIR, denotada por $A^{\pi 1}$. Para o COB, esta permutação não é ótima, justificada pela existência de soluções melhores, como a apresentada na Seção 3.1.*

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

*Adicionando-se uma coluna de elementos nulos a $A$ e resolvendo o CIR nesta nova matriz $B$, encontramos o resultado ótimo $\pi_2 = [a, f, c, 0, d, e, b]$. Pela característica circular do problema, a permutação $\pi_3 = [0, d, e, b, a, f, c]$ é idêntica à $\pi_2$. Devido à descontinuidade em todas as linhas causada pela coluna nula, não ocorrem blocos de uns circulares. Isto torna $\pi_3$ uma permutação ótima para o COB também. Adicionalmente, a remoção da coluna nula da matriz resultante não altera esta solução, considerando que não há blocos de uns à esquerda desta coluna. Assim, demonstra-se que uma instância COB pode ser transformada em uma CIR com a preservação da otimalidade de sua solução.*

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

### 4.1.2 Redução ao Problema do Caixeiro Viajante

Para redução do CIR ao TSP, utiliza-se a distância de *Hamming*. Esta distância originalmente é utilizada para determinar o grau de similaridade entre duas cadeias de caracteres de mesmo tamanho. A distância é calculada em função do número de caracteres em que duas cadeias se diferenciam entre si e é naturalmente simétrica, i.e., a distância de uma cadeia para outra é idêntica à distância inversa. Adicionalmente, a distância de *Hamming* obedece à desigualdade triangular (Robinson, 2003, pp. 255-257).

A prova formal da preservação da qualidade da solução proveniente desta transformação é descrita detalhadamente por Haddadi e Layouni (2008).

**Lema 2.** *CIR* $\leq_{PTAS}$ *TSP.*

**Demonstração.** *Nesta redução, considera-se cada uma das* $n+1$ *colunas da matriz* $B$ *como uma cidade do TSP. As distâncias* $D_h$ *entre as cidades são dadas pelas distâncias de 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 de Hamming de duas colunas de* $B$*, maior a expectativa de que que estas colunas apareçam próximas, ou mesmo contiguamente, nas permutações que representam as soluções do CIR e do TSP. Adicionalmente, a simetria nas distâncias de Hamming e a observância à desigualdade triangular caracterizam também o TSP simétrico.*

A matriz de distâncias $D_h$ pode ser construída, a partir da matriz $B$, pelo seguinte cálculo: $D_h = B^T(M^1 - B) + (M^1 - B)^T B$, conforme Haddadi e 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 de *Hamming* em complexidade de tempo $O(n^2 m)$, ao interpretar cada coluna da matriz de entrada como sendo uma cadeia de $m$ bits e aplicando a operação XOR para cada combinação de pares de cadeias de bits. O número de elementos não-nulos na cadeia de bits resultante equivale à distância de *Hamming* entre tais colunas.

Nos casos em que o número de linhas é menor do que o número de colunas, a forma proposta para cálculo das distâncias de *Hamming* possui menor complexidade assintótica. Outra vantagem computacional é a utilização de operações bit a bit, mais rápidas na prática quando comparadas à multiplicações de números inteiros.

A solução do TSP para a instância $(n+1, D_h)$ é equivalente à solução do CIR para a instância $B$, conforme provado por (Kou, 1977; Alizadeh et al., 1995; Greenberg e Istrail, 1995). Haddadi e Layouni (2008) comprovam também que o custo da rota da solução TSP é igual ao dobro do número de blocos circulares consecutivos de uns para o CIR. Por transitividade, esta solução também é equivalente à solução do COB para a instância $A$, como desejou-se demonstrar.

**Teorema.** *COB* $\leq_{PTAS}$ *TSP.*

A matriz $D_h$ ilustra a redução do CIR ao TSP, considerando a matriz $B$.

Dada a correspondência entre COB e TSP, é possível utilizar qualquer algoritmo de resolução do TSP simétrico para produzir uma solução para o COB. A literatura de tais algoritmos

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

é vasta, possuindo métodos de aproximação (Christofides, 1976), heurísticas sub-ótimas de bom desempenho (Lin e Kernighan, 1973) e métodos exatos (Held e Karp, 1962).

## 4.2 O resolvedor Concorde

O método considerado estado-da-arte por mais de 15 anos entre os métodos exatos para solução do TSP simétrico é o resolvedor Concorde (Applegate et al., 2006). Este programa 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.

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

Desta forma, utiliza-se neste trabalho a modelagem do COB como o TSP e resolve-se este último problema por meio do resolvedor exato Concorde e, posteriormente, obtém-se o número de descontinuidades. A partir do número de descontinuidades, obtém-se o valor da solução para o COB, e a partir da permutação do TSP, obtém-se a permutação para o COB.

A escolha do Concorde como resolvedor é adequada devido à propriedade das distâncias de *Hamming* serem simétricas e obedecerem a desigualdade triangular. Embora alguns trabalhos da literatura tenham utilizado anteriormente esta modelagem em determinados contextos, as soluções foram geradas por meio de heurísticas.

A solução ótima obtida pelo Concorde para o TSP consiste em uma permutação $\pi^*$ de $n+1$ elementos. A solução ótima para o COB é obtida a partir da permutação das colunas da matriz $B$ em razão da ordem estabelecida por $\pi^*$ e pela remoção da coluna de elementos nulos. A solução ótima da instância de TSP para a matriz $D_h$ da Seção anterior é $\pi = [0, d, e, b, a, f, c]$. Esta sequência é então aplicada à matriz $B^{\pi^*}$, com a coluna extra. A redução para o COB

é enfim realizada pela remoção desta coluna, produzindo a matriz $A^{\pi^*}$. As soluções ótimas para o CIR e COB considerando a matriz de exemplo $A$ são apresentadas abaixo.

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

## 4.3 A heurística LKH

O método LKH é uma implementação aprimorada da heurística *Lin-Kernighan* (Lin e Kernighan, 1973), devida a Helsgaun (2000). Assim como o Concorde, seu código é escrito em ANSI C e de uso gratuito para fins acadêmicos e para uso não-comercial. O método se mostra eficaz, apesar de se tratar de um algoritmo heurístico, detendo recordes de melhor solução encontrada para instâncias de dimensões muito grandes, cujos valores ótimos das soluções são desconhecidos. Entre estas instâncias cita-se a *world TSP*, que contém 1.904.711 cidades. Outro recorde é devido à solução da instância *pla85900*, que contém 85.900 cidades, para a qual o método obteve a solução ótima, determinada anteriormente pelo resolvedor Concorde.

A heurística *Lin-Kernighan* considera a entrada do problema representada como um grafo, em que os vértices representam as cidades e as arestas ponderadas indicam a conexão e a distância entre as cidades. O método consiste na utilização de uma tradicional busca local do TSP denominada *k-opt variável*, em que $k$ arestas pertencentes à atual solução são removidas e substituídas por outras $k$ arestas.

A cada passo da heurística *Lin-Kernighan*, considera-se a execução de um ($k$+1)-*opt* através de regras heurísticas de seleção de vértices, a fim de melhorar a solução. Este processo continua, reiniciando o valor de $k$ ao fim de cada passo para um valor padrão, até que o critério de parada do algoritmo seja satisfeito. Em uma implementação ingênua, uma busca *k-opt* possui complexidade de tempo $O(n^k)$, o que torna o algoritmo computacionalmente inviável para grandes valores de $n$ ou de $k$. Na etapa de análise da execução do ($k$+1)-*opt*, são selecionadas heuristicamente "cidades-chave" para a participação da busca, no intuito de acelerar o tempo computacional.

O autor aprimora as regras heurísticas originais para a seleção de cidades-chave para a busca, e emprega estratégias de buscas maiores e mais complexas. Por exemplo, menciona-se o particionamento em subproblemas, a implementação eficiente da busca *k-opt variável*, a utilização de outras buscas locais auxiliares e a utilização de análise de sensibilidade para direcionar e restringir a busca. Desta forma, produziu-se um método híbrido robusto.

Como mencionado anteriormente em detalhes na Seção 4.2, utiliza-se o LKH para determinar soluções heurísticas $\pi^h$ para instâncias COB transformadas em instâncias HTSP, possibilitando a análise de tempo e qualidade de soluções em relação ao resolvedor Concorde.

# Capítulo 5

# Experimentos Computacionais

Os experimentos computacionais foram realizados em dois ambientes computacionais distintos. A redução do COB ao HTSP e a aplicação do LKH foram executadas em um computador com processador *Intel Core i7-4790* 3,60GHz, com quatro *cores*, 16GB de memória RAM e sistema operacional *Ubuntu 16.04 LTS*. O processador possui uma pontuação de *benchmark* de 9990, vide PassMark CPU Benchmark (2018). A redução foi implementada na linguagem C++ e o código 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 i7-990X* 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). Os processadores possuem pontuação de *benchmark* de 9101, 12050 e 8944, respectivamente. Embora os computadores sejam *multicore*, o resolvedor Concorde foi executado de maneira sequencial.

Para averiguar a qualidade dos métodos, foi analisado o conjunto de instâncias *benchmark* encontrado no trabalho de Haddadi et al. (2015). Desta forma, comparam-se as novas soluções encontradas neste trabalho com os resultados considerados estado-da-arte do COB. Adicionalmente, é proposto um novo conjunto de instâncias, a fim de ampliar a abrangência dos experimentos realizados.

## 5.1 Instâncias *Haddadi*

São considerados os dois grupos de instâncias propostos por Haddadi et al. (2015). O primeiro contém 5 instâncias reais advindas de um problema de alocação de estações de uma 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 por linha da matriz.

As Tabelas 5.1 e 5.2 apresentam os resultados comprovadamente 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 blocos consecutivos encontrados pelo Concorde e o tempo de execução do método exato, em segundos. A plataforma *NEOS* impõe um limite de 14400 segundos (4 horas) para resolução 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. As duas últimas colunas indicam a solução encontrada pelo algoritmo LKH e o seu tempo de execução.

Tabela 5.1: Resultados computacionais, instâncias *Haddadi*.

| Instância | Dimensões | Concorde | Tempo (s) | LKH | Tempo (s) |
|---|---|---|---|---|---|
| $RCOV_{1km}$ | $757 \times 707$ | 757 | 13,48 | 757 | 3,17 |
| $RCOV_{2km}$ | $1196 \times 889$ | 1196 | 14,22 | 1196 | 19,69 |
| $RCOV_{3km}$ | $1419 \times 886$ | 1420 | 12,03 | 1420 | 32,61 |
| $RCOV_{5km}$ | $1123 \times 593$ | 1129 | 6,89 | 1130 | 15,55 |
| $RCOV_{10km}$ | $275 \times 165$ | 276 | 1,52 | 276 | 0,95 |
| A1 | | 210 | 63,00 | 210 | 1,59 |
| A2 | | 216 | 1,83 | 216 | 1,44 |
| A3 | $100 \times 200$ | 204 | 45,55 | 204 | 1,38 |
| A4 | | 233 | 0,41 | 233 | 1,41 |
| A5 | | 226 | 1,74 | 226 | 1,98 |
| B1 | | 632 | 1,03 | 632 | 1,56 |
| B2 | | 629 | 0,91 | 629 | 0,35 |
| B3 | $100 \times 200$ | 619 | 0,96 | 619 | 0,69 |
| B4 | | 638 | 20,20 | 638 | 0,90 |
| B5 | | 613 | 0,94 | 613 | 0,58 |
| C1 | | 1245 | 0,92 | 1245 | 0,20 |
| C2 | | 1242 | 1,29 | 1242 | 0,24 |
| C3 | $100 \times 200$ | 1247 | 103,37 | 1247 | 0,34 |
| C4 | | 1253 | 1,03 | 1253 | 0,25 |
| C5 | | 1311 | 111,68 | 1311 | 0,38 |
| D1 | | 491 | 2,65 | 491 | 15,46 |
| D2 | | 502 | 4,90 | 502 | 15,52 |
| D3 | $100 \times 500$ | 445 | 3,18 | 445 | 14,43 |
| D4 | | 472 | 2,83 | 472 | 13,89 |
| D5 | | 454 | 2,54 | 454 | 12,19 |
| E1 | | 1412 | 3,98 | 1412 | 1,02 |
| E2 | | 1397 | 7,72 | 1397 | 3,95 |
| E3 | $100 \times 500$ | 1394 | 4,61 | 1394 | 4,75 |
| E4 | | 1386 | 7,22 | 1386 | 6,41 |
| E5 | | 1430 | 2,61 | 1430 | 3,23 |

Os dados reportados comprovam a eficiência do método exato para a resolução da maior

Tabela 5.2: Resultados computacionais, instâncias *Haddadi* (cont.).

| Instância | Dimensões | Concorde | Tempo (s) | LKH | Tempo (s) |
|---|---|---|---|---|---|
| F1 | | 2955 | 4,92 | 2955 | 5,35 |
| F2 | | 3002 | 5,21 | 3002 | 3,81 |
| F3 | $100 \times 500$ | 2999 | 2,25 | 2999 | 3,09 |
| F4 | | 3009 | 9,34 | 3009 | 9,36 |
| F5 | | 3021 | 3,98 | 3021 | 0,75 |
| G1 | | 893 | 15,89 | 893 | 75,37 |
| G2 | | 884 | 7,95 | 884 | 137,32 |
| G3 | $100 \times 1000$ | 912 | 10,23 | 912 | 145,97 |
| G4 | | 932 | 7,56 | 932 | 130,60 |
| G5 | | 763 | 9,53 | 763 | 134,34 |
| H1 | | 2708 | 7,05 | 2708 | 105,17 |
| H2 | | 2597* | 14400 | 2597 | 58,95 |
| H3 | $100 \times 1000$ | 2688* | 14400 | 2688 | 49,12 |
| H4 | | 2516 | 14,66 | 2516 | 20,59 |
| H5 | | 2780 | 6,82 | 2780 | 59,88 |
| I1 | | 5714 | 15,06 | 5714 | 9,23 |
| I2 | | 5767 | 6,35 | 5767 | 18,50 |
| I3 | $100 \times 1000$ | 5653* | 14400 | 5653 | 13,65 |
| I4 | | 5674 | 20,23 | 5674 | 14,54 |
| I5 | | 5676* | 14400 | 5676 | 7,19 |

* Tempo limite de 4 horas atingido.

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 os limitantes superiores obtidos.

O algoritmo LKH consegue resultados satisfatórios também, sendo mais rápido que o Concorde em pouco mais da metade das instâncias e alcançando as soluções ótimas, exceto por uma. O algoritmo não conseguiu atingir o ótimo para a instância $\text{RCOV}_{5km}$, na qual o valor da solução se difere da ótima em uma unidade. As instâncias que causam o pior desempenho do método exato são processadas sem dificuldade pelo método heurístico, para as quais são geradas soluções de valor igual por ambos os métodos.

A Tabela 5.3 apresenta a comparação entre os resultados obtidos neste trabalho (Concorde) e os melhores soluções conhecidas na literatura até então (MSC), determinados por Haddadi et al. (2015) e Nascimento e Carvalho (2017). Estes trabalhos apresentam a média dos resultados das 5 instâncias de cada grupo. Para fins de análise, apresenta-se a distância

percentual entre as soluções (*gap*), calculada como $\frac{MSC-Concorde}{Concorde} \times 100$.

Tabela 5.3: Análise comparativa de performance.

| Instância | Dimensões | Concorde | MSC | *gap* (%) |
|---|---|---|---|---|
| $RCOV_{1km}$ | $757 \times 707$ | 757 | 757[a] | 0,00 |
| $RCOV_{2km}$ | $1196 \times 889$ | 1196 | 1206[a] | 0,84 |
| $RCOV_{3km}$ | $1419 \times 886$ | 1420 | 1461[a] | 2,89 |
| $RCOV_{5km}$ | $1123 \times 593$ | 1129 | 1143[a] | 1,24 |
| $RCOV_{10km}$ | $275 \times 165$ | 276 | 280[a] | 1,45 |
| A | $100 \times 200$ | 217,8 | 253,0[a] | 16,16 |
| B | | 626,2 | 671,6[b] | 7,25 |
| C | | 1259,6 | 1307,2[b] | 3,78 |
| D | $100 \times 500$ | 472,8 | 552,0[a] | 16,75 |
| E | | 1403,8 | 1576,2[b] | 12,28 |
| F | | 2997,2 | 3162,2[b] | 5,51 |
| G | $100 \times 1000$ | 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 e Carvalho (2017)

As melhores soluções heurísticas da literatura para os nove grupos de instância estão em média 11,95% distantes 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.

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 e 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.

## 5.2 Novas instâncias

Almejando expandir o alcance dos testes para averiguar a eficiência dos métodos, é proposto um novo conjunto de instâncias, geradas aleatoriamente. Para a geração destas instâncias, foi explorada com maior ênfase a disparidade na proporção entre as dimensões das matrizes, dado que no conjunto de instâncias anterior o número de linhas é fixo em 100 para todas as instâncias. Foram geradas matrizes de dimensões variadas, nas quais os elementos não nulos são dispostos de forma desorganizada. Adicionalmente, não há linhas ou colunas completa-

mente nulas. As matrizes resultantes possuem a propriedade de ter, em média, uma proporção idêntica na quantidade de elementos nulos e de elementos não-nulos.

Escolheu-se produzir 5 instâncias diferentes para cada dimensão alternativa das matrizes. Determinou-se a escolha de três dimensões diferentes de linhas e três dimensões diferentes de colunas para as matrizes, resultando em um total de 45 instâncias. O tamanho das matrizes e o resultado final de número de blocos de cada instância são dados nas Tabelas 5.4 e 5.5. Novamente, os resultados marcados com um asterisco indicam os casos em que o limite de tempo foi atingido e, portanto, as soluções obtidas pelo Concorde não são comprovadamente ótimas. Os mesmos ambientes computacionais do experimento anterior são utilizados.

Tabela 5.4: Resultados computacionais, novas instâncias.

| Instância | Dimensões | Concorde | Tempo (s) | LKH | Tempo (s) |
|---|---|---|---|---|---|
| COB-050-0500A | | 3796 | 0,68 | 3796 | 0,44 |
| COB-050-0500B | | 3826 | 1,90 | 3826 | 0,91 |
| COB-050-0500C | 50 × 500 | 3821 | 0,91 | 3821 | 2,85 |
| COB-050-0500D | | 3792 | 1,65 | 3792 | 0,49 |
| COB-050-0500E | | 3806 | 1,79 | 3806 | 0,75 |
| COB-050-1000A | | 7226 | 2772,07 | 7226 | 9,79 |
| COB-050-1000B | | 7212 | 3,66 | 7212 | 10,31 |
| COB-050-1000C | 50 × 1000 | 7224* | 14400 | 7224 | 7,49 |
| COB-050-1000D | | 7254 | 1,83 | 7254 | 6,46 |
| COB-050-1000E | | 7197* | 14400 | 7197 | 6,39 |
| COB-050-1500A | | 10486 | 6,64 | 10486 | 21,59 |
| COB-050-1500B | | 10514 | 8,53 | 21028 | 27,16 |
| COB-050-1500C | 50 × 1500 | 10522 | 11,59 | 10522 | 15,81 |
| COB-050-1500D | | 10509* | 14400 | 10509 | 16,20 |
| COB-050-1500E | | 10526 | 37,07 | 10526 | 15,44 |
| COB-100-0500A | | 9043 | 3,41 | 9043 | 1,07 |
| COB-100-0500B | | 9047 | 2,06 | 9047 | 1,82 |
| COB-100-0500C | 100 × 500 | 9048 | 0,79 | 9048 | 0,86 |
| COB-100-0500D | | 9000 | 1,52 | 9000 | 1,68 |
| COB-100-0500E | | 9028 | 0,63 | 9028 | 1,15 |
| COB-100-1000A | | 17518 | 4,29 | 17518 | 9,51 |
| COB-100-1000B | | 17544 | 2,49 | 17544 | 3,76 |
| COB-100-1000C | 100 × 1000 | 17442 | 3,35 | 17442 | 3,93 |
| COB-100-1000D | | 17486 | 4,25 | 17486 | 6,01 |
| COB-100-1000E | | 17452 | 2,52 | 17452 | 6,82 |

* Tempo limite de 4 horas excedido.

A performance do método exato é destacada novamente pela rápida convergência para a solução ótima para a maioria das instâncias. Um grupo de cinco instâncias apresentou um

Tabela 5.5: Resultados computacionais, novas instâncias (cont.).

| Instância | Dimensões | Concorde | Tempo (s) | LKH | Tempo (s) |
|---|---|---|---|---|---|
| COB-100-1500A | | 25796 | 8,48 | 25796 | 11,87 |
| COB-100-1500B | | 25853 | 4,94 | 25853 | 11,70 |
| COB-100-1500C | $100 \times 1500$ | 25750 | 4,15 | 25750 | 13,54 |
| COB-100-1500D | | 25843 | 32,68 | 25843 | 13,42 |
| COB-100-1500E | | 25796 | 7,82 | 25796 | 3,25 |
| COB-150-0500A | | 14467 | 0,74 | 14467 | 0,15 |
| COB-150-0500B | | 14488 | 65,85 | 14488 | 1,00 |
| COB-150-0500C | $150 \times 500$ | 14494 | 0,57 | 14494 | 0,17 |
| COB-150-0500D | | 14510 | 1,13 | 14510 | 0,64 |
| COB-150-0500E | | 14439 | 1,54 | 14439 | 0,96 |
| COB-150-1000A | | 28308 | 5,70 | 28308 | 6,28 |
| COB-150-1000B | | 28330 | 1,31 | 28330 | 6,46 |
| COB-150-1000C | $150 \times 1000$ | 28287 | 7,90 | 28287 | 7,80 |
| COB-150-1000D | | 28301 | 5,82 | 28301 | 7,04 |
| COB-150-1000E | | 28318* | 14400 | 28318 | 5,28 |
| COB-150-1500A | | 41939 | 13,64 | 41939 | 21,45 |
| COB-150-1500B | | 41900 | 4,50 | 41900 | 32,23 |
| COB-150-1500C | $150 \times 1500$ | 41931 | 2,36 | 41931 | 15,01 |
| COB-150-1500D | | 41898 | 65,28 | 41898 | 6,75 |
| COB-150-1500E | | 41895 | 13,66 | 41895 | 18,28 |

* Tempo limite de 4 horas excedido.

crescimento acelerado em seus tempos de execução, na qual quatro excederam o tempo limite de 4 horas para sua resolução. É possível observar que, mesmo com a proporção entre as dimensões das matrizes e o número de blocos de uns crescendo de forma linear assintótica, o tempo de execução não se altera por razão destes valores.

Em contrapartida, o algoritmo LKH gera soluções para as instâncias em tempo de execução proporcional ao número de colunas das instâncias. Verifica-se adicionalmente que o tempo de execução, na maior parte dos casos, se comporta de forma inversamente proporcional à densidade das instâncias e ao número de linhas das matrizes. Este comportamento sugere que o pior caso do algoritmo LKH é encontrado ao resolver instâncias com um menor desvio padrão em sua distância de *Hamming*, isto é, matrizes com as distâncias muito próximas umas das outras. Matrizes esparsas ou com poucas linhas possuem um menor desvio padrão de distância de *Hamming* em relação a matrizes densas ou com número maior de linhas. É razoável imaginar que as colunas das matrizes esparsas se distanciarão das demais colunas por poucas unidades, em média, dada a baixa quantidade de elementos não-nulos por coluna. Por outro lado, a distância de *Hamming* máxima entre duas colunas de uma matriz é igual ao número de linhas e, ao diminuir o número de linhas, diminui-se também a distância de *Hamming* máxima.

Para este conjunto de instâncias, os dois métodos terminam a execução em tempos infe-

riores em relação ao conjunto de instâncias anterior, se ignorarmos os casos patológicos. O método LKH conseguiu atingir o ótimo ou o mesmo valor de solução obtido pelo Concorde em todas as instâncias.

# Capítulo 6

# Conclusões

Este trabalho apresenta a aplicação de um método exato e um método heurístico para abordagem do Problema de Minimização de 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. Os métodos consistem inicialmente na redução do COB ao Problema de Minimização de Blocos Circulares, posteriormente reduzido ao Problema do Caixeiro Viajante com distâncias de *Hamming* (ou *Hamming distance Traveling Salesman Problem*, HTSP). Após realizar as reduções, aplica-se o resolvedor exato Concorde e o método heurístico LKH à solução do HTSP, cujas soluções são finalmente transformadas nas soluções ótimas e aproximadas do COB, respectivamente.

Os experimentos computacionais envolveram dois conjuntos de instâncias. O primeiro contém 50 instâncias *benchmark* disponíveis na literatura, incluindo instâncias reais. O segundo conjunto de instâncias foi produzido neste trabalho, no intuito de gerar uma maior abrangência da análise de desempenho dos métodos, cobrindo dimensões maiores e diferentes proporções entre as dimensões. Em um baixo tempo de execução, foi possível obter as até então inéditas soluções ótimas para a grande maioria das instâncias *benchmark* com o método exato. O método heurístico demonstrou ser competitivo em comparação ao método exato, considerando o tempo de execução e os resultados obtidos. Os resultados dos métodos considerados estado da arte foram comparados a estas novas soluções, a fim de se analisar o impacto dos métodos propostos. O baixo tempo computacional e eficácia de ambos os métodos apresentados nesta monografia indicam que o uso da modelagem empregada deve ser encorajado para a abordagem do COB, uma vez que os resultados se mostraram consideravelmente melhores do que os encontrados na literatura até então.

# Referências Bibliográficas


Alizadeh, F.; Karp, R. M.; Newberg, L. A. e Weisser, D. K. (1995). Physical mapping of chromosomes: A combinatorial problem in molecular biology. *Algorithmica*, 13(1):52–76.

Applegate, D.; Bixby, R.; Chvatal, V. e Cook, W. (2006). Concorde TSP solver.

Chakhlevitch, K.; Glass, C. A. e Shakhlevich, N. V. (2013). Minimising the number of gap-zeros in binary matrices. *European Journal of Operational Research*, 229(1):48–58.

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

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

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

Garey, M. R. e Johnson, D. S. (1979). *Computers and intractability*. WH Freeman New York.

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

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

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

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

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

Helsgaun, K. (2000). An effective implementation of the lin–kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130.

Johnson, D.; Krishnan, S.; Chhugani, J.; Kumar, S. e Venkatasubramanian, S. (2004). Compressing large boolean matrices using reordering techniques. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pp. 13–23. VLDB Endowment.

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

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

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

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

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

Madsen, O. B. (1988). 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.

Mittelmann, H. (2016). NEOS server: concorde.

Nascimento, L. H. L. e Carvalho, M. A. M. (2017). Uma heurística aplicada à uniformidade das características físicas de produtos. In *Anais do XLIX Simpósio Brasileiro de Pesquisa Operacional*.

PassMark CPU Benchmark (2018). PassMark CPU Benchmark. `https://www.cpubenchmark.net/cpu_test_info.html`. Acessado em: 2018-09-10.

Robinson, D. J. (2003). *An introduction to abstract algebra*. Walter de Gruyter.