Universidade Estadual de Campinas

Instituto de Computação

Projeto de Pesquisa de Mestrado

# Um problema de distritamento aplicado à antecipação do faturamento em redes de serviço

*Luis Henrique Pauleti Mendes*
*Orientador: Prof. Dr. Fábio Luiz Usberti*
*Co-orientador: Prof. Dr. Celso Cavellucci*

Campinas, 21 de Agosto de 2018

**Resumo**

Este projeto de pesquisa irá investigar um Problema de Distritamento Capacitado (CDP), cujo objetivo consiste em encontrar subconjuntos de arestas em um grafo que definam distritos conexos, balanceados e com uma capacidade máxima. A motivação deste trabalho consiste na aplicação prática do CDP para empresas que administram redes de distribuição de serviços; em particular, no processo de definição dos distritos para leitura do consumo de seus clientes. Neste projeto de pesquisa, serão propostas metodologias de solução exata e heurística para uma variante do CDP que visa maximizar o faturamento das empresas de distribuição.



**Palavras-chaves**: Meta-heurística; Otimização Combinatória; Problemas de Distritamento; Programação Linear Inteira.

**Abstract**

This research project will study a Capacitated Districting Problem (CDP), whose goal is to find subsets of edges in a graph defining districts that are connected, balanced and with a maximum capacity. The motivation of this work consists in the practical application of CDP for utilities; particularly, in their process of defining the districts to meter read customers. In this research project, we propose methodologies to solve exactly and through heuristics a variant of CDP that aims to maximiza the utilities profit.

**Key-words**: Metaheuristics; Combinatorial Optimization; Districting Problems, Integer Linear Programming.

# Sumário

# 1 Introdução

Em otimização combinatória, particularmente no que se refere a problemas de roteamento e de particionamento, existem dois problemas representativos na literaura: o Problema do Carteiro Chinês, do inglês *Chinese Postman Problem* (CPP) e o Problema de Distritamento, do inglês *Districting Problem* (DP).

O CPP consiste em encontrar o menor circuito que atravessa cada aresta do grafo ao menos uma vez. Este problema está relacionado com um dos problemas mais antigos em teoria dos grafos: encontrar um ciclo euleriano em um grafo conexo, isto é, encontrar uma maneira de percorrer toda aresta exatamente uma vez (EDMONDS; JOHNSON, 1973).

O CPP modela o problema do mundo real enfrentado por um carteiro que deve entregar correspondências em cada rua de uma certa região e retornar ao seu ponto de origem. Porém, o CPP pode também ser aplicado em outros contextos, como o dos leituristas das companhias provedoras de água e energia, que devem medir o consumo de todos os clientes de suas redes de distribuição.

O DP consiste em particionar um território, ou seja, agrupar pequenas áreas geográficas, chamadas unidades básicas, em aglomerados geográficos maiores, chamados distritos, de forma que estes estejam de acordo com critérios de planejamento pré-estabelecidos. Exemplos típicos de unidades básicas são clientes, ruas ou códigos postais. Três critérios de planejamento importantes são balanceamento, conexidade e compacidade. Balanceamento refere-se ao objetivo de estabelecer distritos que tenham tamanhos iguais. Um distrito é conexo se é possível transitar entre suas unidades básicas sem ter de sair dele. Finalmente, um distrito é compacto se ele tiver um formato arredondado, não distorcido e sem buracos (KALCSICS, 2015).

O DP é um modelo natural para vários problemas do mundo real. Kalcsics (2015) ilustra várias aplicações interessantes do DP, incluindo distritamento político e planejamento de território de vendas. Distritamento político consiste no problema de dividir uma área governamental em distritos eleitorais a partir dos quais candidatos políticos são eleitos. Já o planejamento de território de vendas é uma tarefa comum a todas as companhias que operam com uma equipe de vendas e consiste em subdividir o mercado em regiões que são atendidas por um ou mais representantes de vendas.

Uma variante do DP é o Problema de Distritamento Capacitado, do inglês *Capacitated Districting Problem* (CDP), onde cada distrito possui uma capacidade máxima que pode estar relacionada, por exemplo, ao número máximo de unidades agrupadas por distrito, ou ainda a um limite de atendimento de demanda por distrito. Uma das aplicações práticas do CDP consiste em definir as regiões a serem percorridas pelos leituristas das companhias de redes de serviço. Assis, Franca e Usberti (2014) estudaram um problema similar: o Problema de Redistritamento Capacitado Multicritério, do inglês *Multicriteria Capacitated Redistricting Problem* (MCRP). Note que no MCRP um conjunto de distritos originais já existe e o critério de balanceamento é incorporado na função objetivo, enquanto que no CDP estudado neste projeto o desbalanceamento é incorporado nas restrições e a função objetivo tem um viés econômico.

Neste projeto será estudado o Problema de Distritamento Econômico e Capacitado, do inglês *Capacitated and Economic Districting Problem* (CEDP), que consiste em uma variante do CDP onde os distritos, além de possuírem uma capacidade máxima, são formados a partir de um critério econômico, visando, por exemplo, a redução de custos ou o aumento do faturamento. Assim como o CDP, uma aplicação prática consiste em definir regiões a serem percorridas pelos leituristas das companhias de redes de serviço, porém também leva-se em consideração a antecipação de faturamento quando se aloca consumidores com maiores consumos em distritos que são faturados previamente.

Este documento está organizado da seguinte maneira. Na Seção 2 o CEDP é formalmente descrito. Na Seção 3 os objetivos deste projeto são apresentados. Na Seção 4 um modelo matemático do CEDP é apresentado. A Seção 5 descreve as metodologias propostas neste projeto para a solução do problema. A Seção 6 descreve as formas de análise de resultados. Finalmente, a Seção 7 traz o cronograma de atividades deste projeto.

# 2 O Problema de Distritamento Econômico e Capacitado

Uma definição formal do CEDP é apresentada a seguir. Seja $m$ um inteiro positivo representando o número de distritos; $D$ um inteiro positivo representando a capacidade máxima permitida para cada distrito; $B$ um real positivo representando o desbalanceamento máximo permitido para cada distrito; $G = (V, E)$ um grafo conexo não-orientado, onde $V$ representa o conjunto de vértices e $E$ representa o conjunto de arestas. Associado a cada aresta $e \in E$ há um inteiro não-negativo $d_e$, que representa a demanda de $e$. Associado a cada $j \in \{1, \ldots, m\}$ e a cada aresta $e \in E$ há um inteiro não-negativo $c_{e,j}$ que representa o lucro obtido ao se colocar a aresta $e$ no $j$-ésimo distrito. O objetivo do CEDP consiste em particionar $E$ em uma coleção $\mathcal{E} = \{E_1, \ldots, E_m\}$ de $m$ distritos conexos, maximizando o lucro total e respeitando a capacidade e o desbalanceamento máximo.

Um distrito $E_j \in \mathcal{E}$ é dito ser conexo se o subgrafo induzido $G[E_j]$ for conexo. Distritos perfeitamente balanceados geralmente não são possíveis devido à natureza discreta do problema. Porém, existem diversas abordagens na literatura para quantificar o desbalanceamento e incorporar esse critério no processo de distritamento (KALCSICS, 2015).

A medida local de desbalanceamento de um distrito $E_j \in \mathcal{E}$ adotada neste trabalho é baseada no desvio relativo da demanda do distrito, denotada por $d_j$, em relação à demanda média dos distritos, dada por $\bar{d} = \frac{\sum_{j=1}^{m} d_j}{m}$:

$$bal_j = \left| \frac{d_j - \bar{d}}{\bar{d}} \right|, \forall j \in \{1, \ldots, m\}.$$

Um distrito $E_j \in \mathcal{E}$ está perfeitamente balanceado se $bal_j = 0$.

Uma medida global de desbalanceamendo de um distritamento $\mathcal{E}$ é calculada como o desbalanceamento máximo de um distrito:

$$bal_{\mathcal{E}}^{max} = \max_j \{bal_j\}.$$

A Figura 2.1 mostra um exemplo de distritamento em um grafo onde todas as arestas possuem a mesma demanda. Neste exemplo, as arestas foram particionadas em três subconjuntos: as vermelhas pontilhadas, as amarelas contínuas e as azuis tracejadas. Note que os distritos definidos pelos subconjuntos de arestas são conexos, pois o subgrafo induzido por cada subconjunto de arestas é conexo. Note também que, como todas as arestas possuem demandas iguais, os distritos são perfeitamente balanceados, pois cada subconjunto de arestas possui exatamente 4 arestas.

Uma aplicação do CEDP consiste em um problema enfrentado por companhias provedoras de redes de serviço como água e energia: definir em que dia útil do mês será realizada a leitura de cada um de seus clientes. Esta decisão deve considerar vários critérios logísticos, como a carga horária dos leituristas, o consumo médio, a localização geográfica e, no caso de um redistritamento, o dia útil do mês em que atualmente é realizada a leitura do consumo de cada um de seus clientes. A solução desse problema tem potencial de aumentar o faturamento dessas companhias. Por exemplo, clientes que apresentam um alto consumo médio devem ter maior prioridade para serem faturados antes daqueles com menor consumo médio. Por outro lado, a fim de respeitar a carga horária dos leituristas, as companhias devem dar preferência por realizar no mesmo dia a leitura do consumo de clientes que estão localizados próximos uns ao outros, pois, desta forma, os leituristas despenderão menos tempo se locomovendo entre os clientes.

É possível modelar este problema como um CEDP onde o número de distritos $m$ corresponde ao número de dias úteis do mês necessários para a companhia realizar a

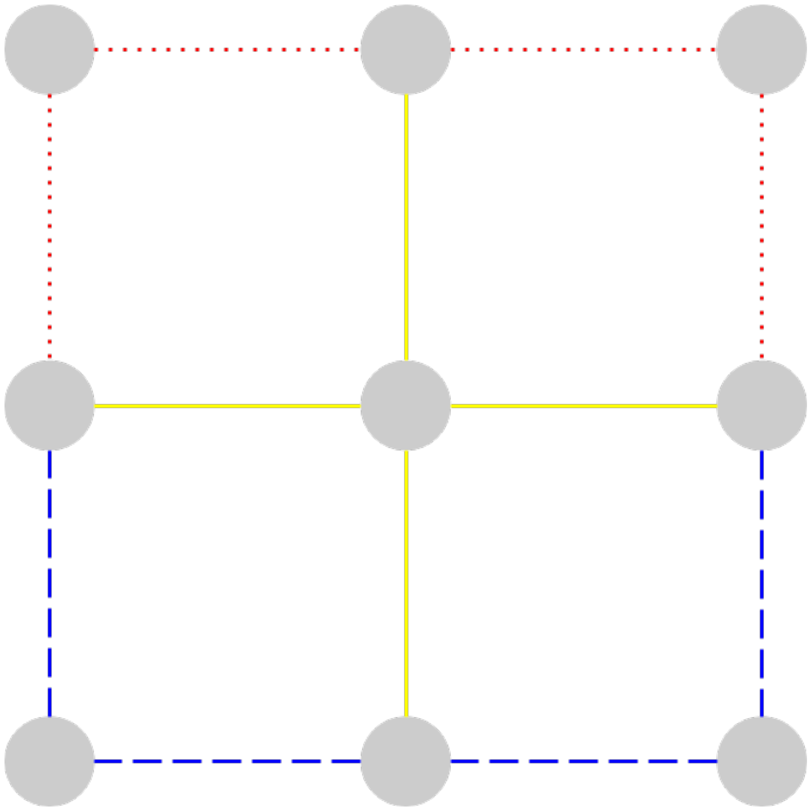

Figura 2.1: Exemplo de um distritamento.

leitura do consumo de seus consumidores; $D$ corresponde à força de trabalho (em homens-hora) disponível por distrito; B corresponde à tolerância de desbalanceamento permitida pela companhia; e o grafo $G = (V, E)$ corresponde à topologia da rede, sendo que cada vértice $v \in V$ corresponde a um cruzamento de ruas e cada aresta $e \in E$ corresponde a um segmento de rua conectando dois cruzamentos. A demanda $d_e$, associada ao segmento de rua $e$, corresponde ao tempo necessário para os leituristas percorrerem $e$, realizando a leitura do consumo dos clientes nele localizados. Já o custo $c_{e,j}$ associado ao segmento de rua $e$ e ao $j$-ésimo distrito corresponde ao ganho de capital da companhia ao realizar a leitura do consumo de todos os clientes localizados no segmento de rua $e$ no dia útil $j$.

Os valores $d_e$ e $c_{e,j}$ são parâmetros de entrada. Seja $v$ a velocidade média dos leituristas e $s_e$ o comprimento do segmento de rua $e$, tem-se que $d_e = \dfrac{s_e}{v}$. Seja $C_e$ o conjunto de clientes localizados no segmento de rua $e$, $j_i$ o dia útil do mês em que atualmente é realizada a leitura do consumo do cliente $i$, $p_i$ o faturamento médio do cliente $i$ e $r$ a taxa de juros diária, tem-se que $c_{e,j} = \sum_{i \in C_e} p_i(1+r)^{(j-j_i)}$.

A demanda $d_j$ de um distrito $E_j \in \mathcal{E}$ pode ser calculada como o tamanho de um ciclo de carteiro chinês mínimo no subgrafo induzido $G[E_j]$. Porém, resolver um CPP para cada subgrafo induzido $G[E_j]$ pode ser computacionalmente custoso. Uma outra forma de sobreestimar a demanda $d_j$ é somar duas vezes a demanda de cada aresta em $E_j$: $d_j = 2\sum_{e \in E_j} d_e$. Note que $d_j = 2\sum_{e \in E_j} d_e$ é uma 2-aproximação do CPP em $G[E_j]$.

Neste projeto será estudado apenas o problema de definir os distritos que serão atendidos pelos leituristas. O problema de definir os caminhos dos leituristas dentro de cada distrito já foi investigado por Maziero, Usberti e Cavellucci (2015) e Mendes e Usberti (2017).

# 3 Objetivos

Nesta seção são apresentados os objetivos gerais e os objetivos específicos deste projeto de pesquisa de mestrado:

## 3.1 Objetivos Gerais

- Investigar formalmente o CEDP, propondo formulações matemáticas e metodologias de solução exata e heurística.
- Conduzir experimentos computacionais das metodologias propostas em um conjunto representativo de instâncias.

## 3.2 Objetivos Específicos

- Propôr e implementar metodologias de solução exatas, fundamentadas em programação linear inteira, branch-and-bound e branch-and-cut.
- Propôr e implementar metaheurísticas visando a solução de instâncias de porte real.
- Elaborar um conjunto representativo de instâncias, geradas artificialmente e extraídas de mapas reais, para avaliar computacionalmente as metodologias propostas.
- Avaliar o desempenho das metodologias propostas com base em métodos analíticos de avaliação comparativa de desempenho.

# 4 Formulação em Programação Linear Inteira

Um Programa Linear, do inglês *Linear Program* (LP), consiste no problema de minimizar ou maximizar uma função objetivo linear na presença de restrições lineares (desigualdades e/ou igualdades) (BAZARAA; JARVIS, 1977).

Considere o seguinte LP genérico:

$$\max\{cx : Ax \leq b, x \in \mathbb{R}^n_+\}$$

onde $A$ é uma matriz de dimensões $m \times n$ com os coeficientes das restrições, $c$ é um vetor linha de dimensão $n$ com os coeficientes da função objetivo, $b$ é um vetor coluna de dimensão $m$ com os coeficientes do lado direito das desigualdades (ou igualdades) e $x$ é um vetor coluna de dimensão $n$ com as variáveis de decisão do problema. Um Programa Linear Inteiro, do inglês *Integer Linear Program* (ILP), consiste em um LP onde algumas (ou todas) variáveis de decisão devem assumir valores inteiros (WOLSEY, 1998).

Muitas vezes são utilizadas formulações de ILP para modelar problemas de otimização combinatória, uma vez que as variáveis discretas podem ser representadas como número inteiros. Em Wolsey (1998) são apresentadas formulações de vários problemas de otimização combinatória, incluindo problemas clássicos da literatura, como o Problema da Mochila, do inglês *Knapsack Problem* (KP), e o Problema do Caixeiro Viajante, do inglês *Traveling Salesman Problem* (TSP).

Uma formulação em ILP para o CEDP é apresentada a seguir.

$$x_{e,j} = \begin{cases} 1, \text{ se a aresta } e \text{ for alocada ao distrito } E_j, \\ 0, \text{ caso contrário.} \end{cases}$$

$$\max \left\{ \sum_{e \in E} \sum_{j=1}^{m} c_{e,j} x_{e,j} \right\} \tag{4.1}$$

sujeito a

$$\sum_{j=1}^{m} x_{e,j} = 1, \forall e \in E \tag{4.2}$$

$$x_{e,j} + x_{f,j} - \sum_{g \in [S, V \setminus S]} x_{g,j} \leq 1, \forall e, f \in E, S \subset V \setminus f, S \supset e, j \in \{1, \ldots, m\} \tag{4.3}$$

$$\sum_{e \in E} d_e x_{e,j} \leq \min \left\{ D, \frac{(1+B) \sum_{e \in E} d_e}{m} \right\}, \forall j \in \{1, \ldots, m\} \tag{4.4}$$

$$\sum_{e \in E} d_e x_{e,j} \geq \frac{(1-B) \sum_{e \in E} d_e}{m}, \forall j \in \{1, \ldots, m\} \tag{4.5}$$

$$x_{e,j} \in \{0, 1\}, \forall e \in E, j \in \{1, \ldots, m\} \tag{4.6}$$

A função objetivo (4.1) busca por uma solução de lucro máximo, onde o lucro de uma solução é dado pelo somatório dos lucros associados a colocar cada aresta em cada distrito. As restrições (4.2) garantem que cada aresta vai ser alocada a exatamente um distrito. As restrições (4.3) garantem que se duas arestas $e, f \in E$ são alocadas a um mesmo distrito $E_j$ ($x_{e,j} = x_{f,j} = 1$), então ao menos uma aresta do corte de arestas $[S, V \setminus S]$ que separa $e$ de $f$ também é alocada a esse distrito ($\sum_{g \in [S, V \setminus S]} x_{g,j} \geq 1$), ou seja, essas restrições garantem que os distritos serão conexos. As restrições (4.4) garantem que a capacidade de cada distrito é respeitada. As restrições (4.5), juntamente com as restrições (4.4), garantem que o desbalanceamento máximo é respeitado. Por fim, as restrições (4.6) garantem a integralidade das variáveis.

Na formulação proposta primeiramente é possível notar que a demanda de um distrito $E_j$ é dada por $d_j = \sum_{e \in E} d_e y_{e,j}$. Como cada distrito $E_j$ deve respeitar a capacidade máxima, chega-se ao primeiro termo da função de mínimo das restrições (4.4). A demanda média dos distritos é dada por $\bar{d} = \frac{\sum_{j=1}^{m} d_j}{m} = \frac{\sum_{e \in E} \sum_{j=1}^{m} d_e y_{e,j}}{m} = \frac{\sum_{e \in E} d_e \sum_{j=1}^{m} y_{e,j}}{m}$, porém, como cada aresta é alocada a exatamente um distrito (restrições (4.2)), tem-se que $\bar{d} = \frac{\sum_{e \in E} d_e}{m}$. Logo, o desbalanceamento de cada distrito $E_j$ é dado por $bal_j = \left| \frac{d_j - \bar{d}}{\bar{d}} \right| = \left| \frac{m \sum_{e \in E} d_e y_{e,j}}{\sum_{e \in E} d_e} - 1 \right|$. Considerando que o desbalanceamento do distritamento é dado pelo desbalanceamento máximo de um distrito, tem-se que, para todo distrito $E_j$, $bal_j \leq B$ $\Rightarrow \left| \frac{m \sum_{e \in E} d_e y_{e,j}}{\sum_{e \in E} d_e} - 1 \right| \leq B \Rightarrow -B \leq \frac{m \sum_{e \in E} d_e y_{e,j}}{\sum_{e \in E} d_e} - 1 \leq B \Rightarrow \frac{(1-B) \sum_{e \in E} d_e}{m} \leq \sum_{e \in E} d_e y_{e,j} \leq \frac{(1+B) \sum_{e \in E} d_e}{m}$. Com isso, deduz-se o segundo termo da função de mínimo das restrições (4.4) e, de forma análoga, deduz-se também as restrições (4.5).

# 5 Metodologias

Nesta seção serão apresentados conceitos introdutórios de metodologias exatas e heurísticas. Sobre métodos exatos, serão apresentados o Branch-and-Bound e o Branch-and-Cut e, sobre heurísticas, a Heurística Lagrangiana, o GRASP ("*Greedy Randomized Search Procedure*") e Algoritmos Genéticos.

## 5.1 Metodologias Exatas

Branch-and-Bound e Branch-and-Cut são métodos para resolver problemas de otimização combinatória formulados como ILP (WOLSEY, 1998; NEMHAUSER; WOLSEY, 1988; BERTSIMAS; TSITSIKLIS, 1997).

O método Branch-and-Bound utiliza uma abordagem de divisão e conquista para explorar o conjunto de soluções inteiras viáveis. Porém, como explorar todo o conjunto é impraticável para a maioria do problemas, são utilizados limitantes primais e duais no valor ótimo para evitar explorar certas partes do conjunto.

Seja $S = \{x \in \mathbb{B}^{|E| \times m} : x \text{ satisfaz a (4.2), (4.3), (4.4), (4.5) e (4.6)}\}$ o conjunto de soluções inteiras viáveis para o CEDP

$$z = \max \left\{ \sum_{e \in E} \sum_{j=1}^{m} c_{e,j} x_{e,j} : x \in S \right\}$$

Divide-se recursivamente o conjunto $S$ em uma coleção finita de subconjuntos $\mathcal{S} = \{S_1, \ldots, S_k\}$ e resolve-se separadamente cada um dos subproblemas

$$z_i = \max \left\{ \sum_{e \in E} \sum_{j=1}^{m} c_{e,j} x_{e,j} : x \in S_i \right\}, i \in \{1, \ldots k\}$$

As soluções ótimas dos subproblemas são comparadas, retornando-se a melhor. Cada subproblema pode ser quase tão difícil quanto o problema original e isto sugere tentar resolver cada subproblema pelo mesmo método, isto é, dividindo-o em outros subproblemas. Esta é a parte de ramificação do método, que induz uma árvore de subproblemas. Esta ramificação pode ser vista como uma enumeração total do elementos de $S$. Porém, como a enumeração total não é viável para a maioria dos problemas, precisa-se evitar dividir o conjunto inicial $S$ em muitos subconjuntos. Se for possível estabelecer que não são mais necessárias divisões no subconjunto $S_i \in \mathcal{S}$, então diz-se que a árvore de subproblemas pode ser podada no nó correspondende a $S_i$.

Sejam $\underline{z}_i$ e $\overline{z}_i$, respectivamente, limitantes primal e dual de $z_i$, o valor de uma solução ótima de cada subproblema $S_i \in \mathcal{S}$. Então $\underline{z} = \max_{i=1}^{k}\{\underline{z}_i\}$ e $\overline{z} = \max_{i=1}^{k}\{\overline{z}_i\}$ são, respectivamente, limitantes primal e dual de $z$, o valor de uma solução ótima para o problema original $S$. Os limitantes primais podem ser obtidos por meio de soluções viáveis para os subproblemas, enquanto que os limitantes duais podem ser obtidos por meio de relaxações ou dualidade.

Pode-se podar a árvore de subproblemas e, assim, enumerar um grande número de soluções implicitamente, se qualquer uma das seguintes condições é satisfeita:

- Poda por inviabilidade: $S_i = \emptyset$.

- Poda por dominância de valor: $\overline{z}_i \leq \underline{z}$.

- Poda por otimalidade: uma solução ótima para $S_i$ é conhecida.

O Algoritmo 1 exibe um pseudo-código do Branch-and-Bound para um problema de maximização.

**Algoritmo 1:** Branch-and-Bound

**Entrada:** Conjunto de soluções inteiras viáveis $S$

**Saída:** Solução ótima $x^\star$

$\mathcal{S} \leftarrow \{S\}$; $x^\star \leftarrow \emptyset$; $\underline{z} \leftarrow -\infty$; $\overline{z} \leftarrow -\infty$;
**enquanto** $\mathcal{S} \neq \emptyset$ **faça**
    Selecione um problema ativo $S_i$ de $\mathcal{S}$;
    **se** $S_i = \emptyset$ **então**
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S_i\}$;
    **senão**
        Compute um limitante dual $\overline{z}_i$ e uma solução dual $x'$ para $S_i$;
        **se** $\overline{z}_i \leq \underline{z}$ **então**
            $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S_i\}$;
        **senão**
            **se** *uma solução ótima* $x_i^\star$ *para* $S_i$ *de valor* $\underline{z}_i$ *é conhecida* **então**
                **se** $\underline{z} \leq \underline{z}_i$ **então**
                    $x^\star \leftarrow x_i^\star$;
                    $\underline{z} \leftarrow \underline{z}_i$;
                **fim**
                $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S_i\}$;
            **senão**
                Divida o problema $S_i$ em uma coleção de subproblemas $\mathcal{S}_i$;
                $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S_i\}$;
                $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_i$;
            **fim**
        **fim**
    **fim**
**fim**
**retorna** $x^\star$

O método Branch-and-Cut é uma variante do método Branch-and-Bound no qual planos de corte são gerados ao longo da árvore de subproblemas. Ao invés de reotimizar rapidamente em cada nó dessa árvore, a ideia desse método é trabalhor o quanto for necessário para obter um bom limitante dual no nó. Existe, na prática, um trade-off, pois, se muitos cortes forem adicionados em cada nó, a reotimização pode ficar muito mais lenta do que antes.

Note que, como as restrições (4.3) são em número exponencial em $|V|$, é inviável resolver instâncias do CEDP de interesse prático utilizando um algoritmo de Branch-and-Bound puro. Porém, é possível utilizar uma abordagem Branch-and-Cut, na qual as restrições (4.3) são adicionadas ao longo da árvore de subproblemas, conforme necessário.

## 5.2 Metodologias Heurísticas

Para instâncias grandes, onde a aplicação de uma metodologia exata se torna computacionalmente inviável, uma heurística torna-se uma alternativa para a obtenção de soluções viáveis de boa qualidade, porém sem garantia de otimalidade, em tempo computacional polinomial no tamanho da entrada.

### 5.2.1 Heurística Lagrangeana

Uma técnica bem conhecida na literatura para encontrar limitantes duais para problemas de otimização combinatória formulados como ILP é a relaxação Lagrangeana (NEMHAUSER; WOLSEY, 1988; WOLSEY, 1998; BEASLEY, 1993).

Considere o ILP genérico:

$$z = \max\{cx : x \in S\}, \text{ onde } S = \{x \in \mathbb{Z}^n_+ : Ax \leq b\}$$

que pode ser reescrito como:

$$
\begin{aligned}
z &= \max\{cx\} \\
A_1x &\leq b_1 (\text{restrições complicadoras}) \\
A_2x &\leq b_2 (\text{restrições tratáveis}) \\
x &\in \mathbb{Z}^n_+
\end{aligned}
\tag{IP}
$$

onde $A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$ e $b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$. Suponha que $A_2x \leq b_2$ são $m - m_1$ "restrições tratáveis", no sentido que um programa com apenas estas restrições é fácil. Deste modo, removendo-se as $m_1$ "restrições complicadoras" $A_1x \leq b_1$, a relaxação resultante é mais fácil de se resolver do que o problema original. Porém, o limitante dual obtido por esta relaxação pode ser fraco, pois algumas restrições importantes são completamente ignoradas.

Uma maneira de atacar esta dificuldade é introduzindo um vetor de multiplicadores de Lagrange $u \in \mathbb{R}^{m_1}_+$, o qual é anexado a este conjunto de restrições e levado para a função objetivo, resultando no seguinte problema:

$$
\begin{aligned}
z(u) &= \max\{cx + u(b_1 - A_1x)\} \\
A_2x &\leq b_2 \\
x &\in \mathbb{Z}^n_+
\end{aligned}
\tag{RL(u)}
$$

O problema RL(u) é chamado de relaxação Lagrangeana de IP com parâmetro $u$. RL(u) não contém as restrições complicadoras. Em vez disso, estas restrições foram incluídas na função objetivo com a "penalidade" $u(b_1 - A_1x)$. Visto que $u \geq 0$, violações de $A_1x \leq b_1$ fazem a penalidade ser negativa, assim, intuitivamente, $A_1x \leq b_1$ será satisfeita se $u$ for adequadamente grande.

Para encontrar o melhor limitante dual dentre a infinidade de valores possíveis para

$u$, torna-se necessário resolver o Problema Dual Lagrangiano:

$$w_{DL} = \min\{z(u) : u \in \mathbb{R}_+^{m_1}\}. \tag{DL}$$

Uma técnica disponível para resolver o problema DL é o método do subgradiante, que é um procedimento iterativo o qual, a partir de um conjunto de multiplicadores de Lagrange inicial, gera novos multiplicadores de forma sistemática.

Em uma heurística Lagrangeana, toma-se uma solução para o problema RL(u) e tenta-se convertê-la em uma solução viável para o problema IP por meio um método de factibilização. Esta solução viável fornece um limitante primal no valor de uma solução ótima para o problema IP.

A característica chave da heurística Lagrangeana é que, assim como o valor da solução para o RL(u) fornece informações úteis (um limitante dual no valor de uma solução ótima para o problema IP), a estrutura da solução para o problema RL(u) (isto é, o valor das variáveis) também pode fornecer informações úteis sobre a estrutura da solução ótima para o problema IP. Assim, cada vez que resolve-se o problema RL(u) dentro do método do subgradiente, a heurística Lagrangeana tem uma oportunidade para transformar a solução para o problema RL(u) em uma solução para o problema IP.

### 5.2.2 GRASP

Feo e Resende (1995) apresentam o Procedimento de Busca Adaptativo Aleatório Guloso, do inglês *Greedy Randomized Adaptive Search Procedure* (GRASP), como uma meta-heurística de dois estágios. O primeiro estágio é responsável pela construção de uma solução inicial a partir de uma heurística construtiva aleatório-gulosa. O segundo estágio realiza uma busca local no intuito de explorar a vizinhança da solução inicial até atingir um mínimo local do espaço de soluções. Após certo número de iterações, a melhor solução encontrada é retornada como a solução final da heurística.

No estágio de construção da solução inicial, os elementos candidatos a participarem da solução inicial são escolhidos de modo aleatório e guloso. Este processo se realiza a partir de uma Lista Restrita de Candidatos, do inglês *Restricted Candidate List* (RCL), que recebe os melhores candidatos a entrar na solução. A partir da RCL, os candidatos que efetivamente entram na solução são escolhidos aleatoriamente com distribuição uniforme.

As soluções encontradas no primeiro estágio do GRASP não são necessariamente ótimos locais, logo o segundo estágio passa a explorar a vizinhança dessas soluções. Uma vizinhança é definida como um conjunto de soluções que são alcançáveis a partir de uma solução inicial após a aplicação de um operador de busca local.

O Algoritmo 2 abaixo exibe um pseudo-código do GRASP para um problema de maximização.

**Algoritmo 2:** GRASP

$S^\star \leftarrow \emptyset$;
**enquanto** *não atingiu critério de parada* **faça**
    $S \leftarrow$ `Solução-Gulosa-Aleatória()`;
    $S' \leftarrow$ `Busca-Local(`$S$`)`;
    **se** `Custo(`$S^\star$`)` $<$ `Custo(`$S'$`)` **então**
        $S^\star \leftarrow S'$;
    **fim**
**fim**
**retorna** $S^\star$

### 5.2.3 Algoritmos Genéticos

Um Algoritmo Genético, do inglês *Genetic Algorithm* (GA), é uma meta-heurística inspirada pelo processo de seleção natural na evolução biológica. Ao contrário das heurísticas que geram uma única solução e atuam exaustivamente para melhorá-la, em um GA, uma população de soluções candidatas (chamadas indivíduos) evoluem em direção a soluções melhores. Cada solução candidata possui uma codificação (cromossomo), que pode ser combinada e mutada.

A evolução começa a partir de uma população inicial de indivíduos gerados aleatoriamente. Trata-se de um processo iterativo, onde a população de cada iteração é denominada geração. Em cada geração, a aptidão dos indivíduos é avaliada; para problemas de maximização, a aptidão é diretamente proporcional ao valor da solução, ou seja, soluções de maior valor são mais aptas. Os indivíduos mais aptos são selecionados estocasticamente da população atual e têm seu genoma modificado por operadores genéticos para formar uma nova geração. O algoritmo então se repete e após um certo critério de parada ser atingido, a melhor solução da população atual é retornada como a solução final da heurística.

O Algoritmo 3 abaixo exibe um pseudo-código de um algoritmo genético para um problema de otimização.

**Algoritmo 3:** Algoritmo-Genético

$t \leftarrow 0$;<br>
$S_0 \leftarrow$ `População-Inicial-Aleatória()`;<br>
`Avaliar-Aptidão(`$S_0$`)`;<br>
**enquanto** *não atingiu critério de parada* **faça**<br>
    $t \leftarrow t + 1$;<br>
    $S_t \leftarrow$ `Selecionar-Mais-Aptos(`$S_{t-1}$`)`;<br>
    $S_t \leftarrow$ `Alterar(`$S_t$`)`;<br>
    `Avaliar-Aptidão(`$S_t$`)`;<br>
**fim**<br>
**retorna** *indivíduo mais apto em* $S_t$

# 6 Análise dos Resultados

## 6.1 Limitantes Primal e Dual

Em problemas de otimização, os limitantes primais e duais são valores que limitam o valor de uma solução ótima. Normalmente o valor de uma solução ótima é desconhecido, porém através dos limitantes é possível definir um intervalo onde este valor se encontra. Em problemas de maximização, o limitante primal é o limitante inferior e o limitante dual é o limitante superior. Neste trabalho, limitantes primais serão obtidos explorando o espaço de soluções viáveis do problema a partir das heurísticas propostas (Seção 5.2). Já os limitantes duais serão obtidos através da relaxação linear dos modelos de programação linear inteira para o problema (Seção 5.1).

Os limitantes primais e duais podem provar a otimalidade de uma solução caso seus valores se igualem. Através desses dois limitantes também é possível definir um desvio de otimalidade (GAP) que representa o quão distante, no pior caso, o valor de uma solução está em relação ao valor ótimo.

## 6.2 Geração de Instâncias

Com o intuito de analisar os resultados das metodologias, será criado um "benchmark" de instâncias representativas para o CDP. Tal conjunto terá instâncias extraídas da literatura e do mundo real, bem como instâncias geradas computacionalmente.

A fim de obter um conjunto representativo de instâncias, serão utilizadas instâncias de diversos tamanhos e densidade. O tamanho de uma instância corresponde ao tamanho do grafo, ou seja, ao número de vértices e arestas do grafo. Já a densidade de um grafo corresponde a um número real no intervalo $[0, 1]$, e diz respeito à quantidade de arestas no grafo com relação a um grafo completo. Por exemplo, um grafo conexo com $n$ vértices e densidade 0 possui apenas $n - 1$ arestas, ou seja, consiste em uma árvore. Um grafo conexo com $n$ vértices e densidade 1 consiste em um grafo completo, onde existe uma aresta interligando cada um dos $\frac{n(n-1)}{2}$ pares distintos de vértices.

## 6.3 Comparação das Metodologias

Os algoritmos desenvolvidos utilizando as metodologias descritas anteriormente serão testados com o conjunto de instâncias descritos na seção anterior. Serão avaliados os valores das soluções obtidas pelas meta-heurísticas propostas, comparando-os com os limitantes duais extraídos pelas solução de formulações matemáticas para o problema. Finalmente, serão avaliados os desvios de otimalidade (GAP) e o tempo de execução, comparando a eficiência das metodologias propostas.

# 7 Cronograma

As etapas deste projeto de mestrado são enumeradas a seguir e apresentadas cronologicamente na Tabela 7.1.

1. Cumprimento de créditos de pós-graduação.
2. Revisão bibliográfica.
3. Formulação matemática do problema.
4. Qualificação de mestrado.
5. Desenvolvimento de metodologias exatas.
6. Desenvolvimento de metodologias heurísticas.
7. Geração de instâncias.
8. Experimentos computacionais.
9. Avaliação e discussão dos resultados.
10. Redação da dissertação.
11. Submissão de artigo para periódico indexado.
12. Defesa de mestrado.

Tabela 7.1: Cronograma do projeto

| | Trimestre | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Etapa | 1º | 2º | 3º | 4º | 5º | 6º | 7º | 8º |
| 1 | X | X | X | | | | | |
| 2 | X | X | | | | | | |
| 3 | | X | | | | | | |
| 4 | | | X | | | | | |
| 5 | | | X | X | | | | |
| 6 | | | | X | X | | | |
| 7 | | | | | X | | | |
| 8 | | | | | X | X | | |
| 9 | | | | | | X | | |
| 10 | | | | | | X | X | X |
| 11 | | | | | | | X | |
| 12 | | | | | | | | X |

# Referências Bibliográficas


EDMONDS, J.; JOHNSON, E. L. Matching, euler tours and the chinese postman. *Mathematical programming*, Springer, v. 5, n. 1, p. 88–124, Dec 1973. ISSN 1436-4646. Disponível em: <https://doi.org/10.1007/BF01580113>.

KALCSICS, J. Districting problems. In: LAPORTE, G.; NICKEL, S.; GAMA, F. Saldanha da (Ed.). *Location science*. Cham: Springer International Publishing, 2015. p. 595–622. ISBN 978-3-319-13111-5. Disponível em: <https://doi.org/10.1007/978-3-319-13111-5_23>.

ASSIS, L. S. de; FRANCA, P. M.; USBERTI, F. L. A redistricting problem applied to meter reading in power distribution networks. *Computers & Operations Research*, Elsevier, v. 41, p. 65–75, 2014. ISSN 0305-0548. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0305054813002086>.

MAZIERO, L. P.; USBERTI, F. L.; CAVELLUCCI, C. O problema do ciclo dominante. In: *SBPO XLVII Proceedings–Brazilian Symposium of Operations Research*. [s.n.], 2015. Disponível em: <http://cdsid.org.br/sbpo2015/wp-content/uploads/2015/08/142930.pdf>.

MENDES, L. H. P.; USBERTI, F. L. Heurısticas para o problema do ciclo dominante com coleta de prêmios. 2017. Disponível em: <http://www.sbpo2017.iltc.br/pdf/169094.pdf>.

BAZARAA, M. S.; JARVIS, J. J. *Linear programming and network flows*. [S.l.]: John Wiley & Sons, 1977.

WOLSEY, L. *Integer Programming*. [S.l.]: Wiley-Interscience, 1998. (Wiley-Interscience Series in Discrete Mathematics and Optimization). ISBN 9780471283669.

NEMHAUSER, G. L.; WOLSEY, L. A. Integer and combinatorial optimization. Wiley-Interscience, 1988.

BERTSIMAS, D.; TSITSIKLIS, J. N. *Introduction to Linear Optimization*. [S.l.]: Athena Scientific Belmont, MA, 1997. v. 6. ISBN 1886529191.

BEASLEY, J. E. Modern heuristic techniques for combinatorial problems. In: REEVES, C. R. (Ed.). New York, NY, USA: John Wiley & Sons, Inc., 1993. cap. Lagrangian Relaxation, p. 243–303. ISBN 0-470-22079-1. Disponível em: <http://dl.acm.org/citation.cfm?id=166648.166660>.

FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedures. *Journal of global optimization*, Springer, v. 6, n. 2, p. 109–133, Mar 1995. ISSN 1573-2916. Disponível em: <https://doi.org/10.1007/BF01096763>.