UNICAMP

MS777: PROJETO SUPERVISIONADO

# Problema do Caixeiro Viajante com Coleta e Entrega

ALUNO: Fabricio C. Machado
fabcm1@gmail.com

ORIENTADOR: Flávio K. Miyazawa
fkm@ic.unicamp.br

6 de dezembro de 2012

## Resumo

Nesse semestre, foram estudados problemas de otimização combinatória e diversos resultados e ferramentas úteis para a combinatória poliédrica. Destacando considerações sobre a complexidade computacional de problemas, propriedades e descrições de poliedros e a equivalência entre otimização e separação. O estudo do Problema do Caixeiro Viajante com Coleta e Entrega é um caso particular dos problemas de otimização combinatória, assim, os conceitos estudados serão úteis na implementação de um algoritmo, futuramente.

# Sumário

# 1 Introdução

Problemas de otimização, na sua forma geral, tratam de maximizar ou minimizar uma função definida sobre um certo domínio. Nos problemas de otimização combinatória este domínio é finito e, em princípio, bastaria testar todos os elementos em busca de um que seja ótimo. Ainda assim, esta estratégia ingênua costuma ser impraticável, pois o tamanho do domínio pode ser muito grande, mesmo para instâncias de tamanho moderado. Dessa maneira, surge a necessidade de usar técnicas mais elaboradas para encontrar soluções de valor ótimo. Uma das estratégias usadas é associar um politopo ao problema e buscar um sistema de inequações que o descrevam. A subárea de pesquisa que surge, denominada combinatória poliédrica, é um dos temas de estudo desse projeto.

É conveniente definirmos formalmente a classe dos PROBLEMAS DE OTIMIZAÇÃO COMBINATÓRIA (POC): Dada uma tripla $(E, c, \mathcal{S})$, onde $E$ é um conjunto finito, $c$ uma função que associa a cada elemento de $E$ um valor real e $\mathcal{S}$ um conjunto de soluções viáveis, formado por subconjuntos de $E$. Deseja-se encontrar $S \in \mathcal{S}$ que maximiza (ou minimiza) a função objetivo $c(S) := \sum_{e \in S} c(e)$.

Enumerando os elementos de $E$, podemos associar a $S$ um vetor em $\mathbb{R}^{|E|}$ correspondente ao seu vetor de incidência $\chi^S$ definido como: $\chi^S_e = 1$, se $e \in S$ e $\chi^S_e = 0$, caso contrário. Considerando o politopo formado pelo fecho convexo dos vetores de incidência dos elementos de $\mathcal{S}$ e usando o resultado de que seus vértices são precisamente esses vetores de incidência, podemos considerar o problema equivalente de otimização sobre o poliedro. Caso encontremos um sistema de inequações que descrevam o poliedro, obtemos um problema de programação linear. A dificuldade está em obter esse sistema de inequações (o que nem sempre é possível), ou ainda na possibilidade desse sistema ter tamanho exponencial em relação às entradas do problema original.

Ao definir os POC, restringimos a função objetivo ao caso linear. Ainda que essa classe possa parecer restrita, vários problemas interessantes podem ser formulados dessa maneira. Como exemplo, o problema do Caixeiro Viajante, onde temos um grafo $G = (V, A)$, em que $E$ corresponde a seu conjunto de arestas $A$, $c$ uma função de custos sobre as arestas e $\mathcal{S}$ contêm os conjuntos de arestas que representam ciclos hamiltonianos (ciclos que passam por todos os vértices do grafo exatamente uma vez). O problema título deste projeto é uma variação deste, em que impomos restrições extras sobre a ordem em que os vértices devem ser visitados.

Nesse semestre, foram estudados diversos resultados e ferramentas úteis para a combinatória poliédrica. Destacando considerações sobre a complexidade computacional de problemas, propriedades e descrições de poliedros e a equivalência entre otimização e separação. Nas próximas seções, serão apresentados breves resumos desses tópicos.

A principal referência usada nesse estudo foi o livro [3], além dos livros [2], [4] e [5].

# 2 Complexidade Computacional

Para comparar os tempos gastos por um algoritmo, vamos definir o tamanho de uma instância $I$ para um problema, como o número de bits necessários para codificá-la. Assim, dizemos que um algoritmo resolve um problema em tempo $O(T(n))$ se, quando é fornecida uma instância de tamanho $n$, o algoritmo pode produzir a solução no tempo máximo $O(T(n))$. Dizemos ainda que um problema pode ser resolvido em tempo polinomial se existe um algoritmo para resolvê-lo em tempo $O(n^k)$ para alguma constante $k$.

Como muitos problemas tem como resposta soluções de difícil comparação, restringiremos nossa atenção a problemas de decisão, aqueles cuja resposta é simplesmente sim ou não. Muitos problemas de interesse não são assim, como os de otimização, mas em geral é simples reformular um problema de otimização como um problema de decisão não mais difícil que o primeiro.

Definimos, assim, a classe de complexidade $\mathcal{P}$ como o conjunto de problemas de decisão que podem ser resolvidos em tempo polinomial. Outra classe de interesse no estudo de complexidade computacional é a classe $\mathcal{NP}$ que é a classe de problemas que podem ser "verificados" em tempo polinomial, isto é, caso a resposta seja sim, deve existir um certicado e um algoritmo capaz de usar esse certificado para comprovar que a resposta é sim em tempo polinomial.

Claramente, $\mathcal{P}$ está contido em $\mathcal{NP}$, pois se somos capazes de resolver um problema em tempo polinomial, somos capazes de verificá-lo sem sequer receber um certificado. A grande questão em aberto nessa teoria é saber se verificar uma solução é de fato muito mais fácil que encontrar uma solução, isto é, se $\mathcal{P} \neq \mathcal{NP}$. A maioria dos pesquisadores acredita que $\mathcal{P}$ e $\mathcal{NP}$ não sejam o mesmo conjunto devido a existência de uma subclasse de $\mathcal{NP}$ chamada $\mathcal{NP}$-completo. Esta classe é formada pelos problemas de decisão para os quais todo problema em $\mathcal{NP}$ pode ser reduzido polinomialmente a ele, isto é, caso exista um algoritmo polinomial para qualquer problema em $\mathcal{NP}$-completo, todos os problemas em $\mathcal{NP}$ poderiam ser resolvidos em tempo polinomial. Desde que Cook [1] mostrou que o problema da satisfabilidade de fórmula booleanas é $\mathcal{NP}$-completo, muitos problemas foram mostrados ser $\mathcal{NP}$-Completos.

Por fim, dizemos que um problema é $\mathcal{NP}$-difícil se for possível reduzir polinomialmente um problema $\mathcal{NP}$-completo a ele. De certa forma, esta é uma extensão da classe $\mathcal{NP}$-completo para problemas que não são necessariamente de decisão mas que, devido a existência dessa redução, são "tão difíceis" quanto qualquer problema $\mathcal{NP}$.

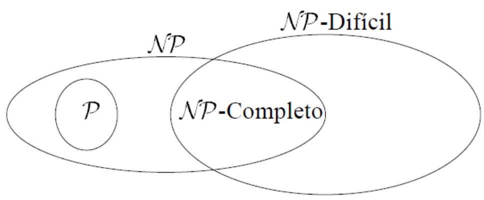


Figura 1: Possível configuração para as classes $\mathcal{P}$, $\mathcal{NP}$ e $\mathcal{NP}$-difícil.

Os Problemas de Otimização Combinatória em geral e o problema do Caixeiro Viajante são problemas $\mathcal{NP}$-dificeis. Isso significa que não é provável a existência de um algoritmo capaz de resolvê-los em tempo polinomial. Apesar da intrínseca dificuldade de tais problemas, muitos destes ocorrem em diversas situações práticas e necessitam de soluções, mesmo que não sejam de valor ótimo (de preferência que tenham valor próximo do ótimo) ou que sejam obtidas após muito processamento computacional. A abordagem poliédrica tem sido usada com sucesso na resolução de instâncias reais desses problemas, gerando soluções exatas para instâncias relativamente grandes.

# 3 Teoria de Poliedros

## 3.1 Conceitos Básicos

Um subconjunto $P \subset \mathbb{R}^n$ é chamado POLIEDRO se:

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\},$$

para alguma matriz $A \in \mathbb{R}^{mxn}$ e algum vetor $b \in \mathbb{R}^m$.

Se o poliedro for limitado é chamado POLITOPO.

Se $a \in \mathbb{R}^m \backslash \{0\}$ e $\alpha \in \mathbb{R}$, então o poliedro $\{x \in \mathbb{R}^n \mid a^\top x \leq \alpha\}$ é chamado SEMI-ESPAÇO. Se substituirmos a desigualdade na definição por uma igualdade, obtemos um HIPERPLANO.

Observando que cada uma das linhas da matriz $A$ juntamente com a correspondente entrada do vetor $b$ define um semi-espaço, segue que um poliedro é a intersecção de um número finito de semi-espaços.

Apesar de não aparecer explicitamente na definição dada, equações também podem ser usadas na definição de poliedros. Vejamos, como exemplo, o sistema:

$$\begin{aligned} Bx + Cy &= c \\ Dx + Ey &\leq d \\ x &\geq 0 \\ x \in \mathbb{R}^p,\ y &\in \mathbb{R}^q \end{aligned}$$

Podemos escrevê-lo como um poliedro em $\mathbb{R}^{p+q}$ tomando:

$$A = \begin{pmatrix} B & C \\ -B & -C \\ D & E \\ -I & 0 \end{pmatrix} \text{ e } b = \begin{pmatrix} c \\ -c \\ d \\ 0 \end{pmatrix}.$$

Em certas formulações, é conveniente definir um poliedro usando apenas igualdades e restrições de não negatividade nas variáveis. Isso pode ser feito introduzindo novas variáveis de folga. Por exemplo, uma inequação da forma:

$$a^\top x \leq \alpha$$

pode ser inicialmente transformada em:

$$a^\top x + y = \alpha,\ y \leq 0$$

e como a variável $x$ é livre de sinal, pode ser substituída pela diferença de duas variáveis não negativas $x^+$ e $x^-$, obtendo:

$$a^\top x^+ - a^\top x^- + y = \alpha,\ x^+ \geq 0,\ x^- \geq 0,\ y \geq 0$$

que é da forma desejada:

$$a'^\top z = \alpha,\ z \geq 0$$

Com as transformações mostradas nos dois últimos exemplos, vemos que as duas representações de poliedros são equivalentes, no sentido que uma pode ser obtida a partir da outra, acrescentando-se variáveis ou restrições, e suas soluções estão relacionadas.

## 3.2 Lema de Farkas e Dualidade

Um resultado muito importante na teoria de programação linear é o Lema de Farkas. Este lema estabelece condições para a factibilidade de poliedros e é usado no estabelecimento de condições de otimalidade para soluções de problemas de programação linear. Este lema aparece sob várias formas alternativas, que podem ser obtidas umas das outras através de transformações semelhantes às apresentadas na última seção. Apresentemos duas versões:

Lema de Farkas - VERSÃO PROJETIVA

*Seja* $A \in \mathbb{R}^{mxn}$ *uma matriz e* $b \in \mathbb{R}^m$ *um vetor. Então existe um vetor* $x$ *com* $Ax \leq b$ *se, e somente se, para cada vetor* $y \geq 0$ *com* $y^\top A = 0$ *temos* $y^\top b \geq 0$.

Essa versão é chamada versão projetiva porque a demonstração da parte "se" do lema pode ser feita de maneira elegante usando-se considerações sobre projeções de poliedros e o método de eliminação de Fourier-Motzkin.

Lema de Farkas - VERSÃO GEOMÉTRICA

*Existe* $x \geq 0$ *tal que* $Ax = b$ *se, e somente se,* $A^\top u \leq 0$ *implica* $u^\top b \leq 0$ *para todo* $u$.

Essa versão possui uma interpretação geométrica interessante. O conjunto dos vetores $b$ para os quais o sistema $Ax = b$, $x \geq 0$ tem solução (ou seja, $b$ pertence ao fecho cônico das colunas de $A$) é precisamente o conjunto de todos os vetores $b$ que formam um ângulo obtuso com cada um dos vetores do cone $A^\top u \leq 0$.

Usando o lema apresentado, podem ser obtidas as relações de dualidade entre problemas lineares. Mais especificamente, denominando primal o problema:

$$\begin{aligned} (P)\ \ max\ c^\top x & \\ & Ax \leq b \end{aligned}$$

associamos o problema dual:

$$\begin{aligned} (D)\ \ min\ y^\top b& \\ y^\top A &= c^\top \\ y &\geq 0 \end{aligned}$$

É fácil ver que se $(P)$ e $(D)$ forem viáveis, então o valor de uma solução do problema de minimização é sempre maior ou igual ao valor de uma solução do problema de maximização. Esse fato é chamado de TEOREMA FRACO DE DUALIDADE, em oposição ao TEOREMA FORTE DE DUALIDADE, de demonstração um pouco mais difícil:

*Sejam $(P)$ e $(D)$ os problemas primal e dual enunciados acima. Então ambos os problemas têm solução ótima e os valores ótimos são iguais se, e somente se, esses problemas têm soluções viáveis.*

Existe ainda o TEOREMA DAS FOLGAS COMPLEMENTARES:

*Sejam $x$ e $y$ soluções viáveis para $(P)$ e $(D)$, respectivamente. Essas soluções são ótimas se, e somente se,, para todo $j$, $y_j > 0$ implica que $A_{j*}x = b_j$.*

As folgas complementares junto com as condições de factibilidade do primal e do dual são conhecidas como CONDIÇÕES DE KUHN-TUCKER e, como enuncidado no último teorema, são condições necessárias e suficientes para a otimalidade.

## 3.3 Faces, Dimensão e Facetas

O estudo das faces de um poliedro envolve algumas definições:

Seja $S \subset \mathbb{R}^n$, $a \in \mathbb{R}^n$ e $\alpha \in \mathbb{R}$. Uma inequação $a^\top x \leq \alpha$ é válida em relação a $S$ se $S \subset \{x \in \mathbb{R}^n \mid a^\top x \leq \alpha\}$. Um hiperplano $H = \{x \mid a^\top x = \alpha\}$ é um hiperplano suporte de $S$ se $a^\top x \leq \alpha$ é válida em relação a $S$ e $S \cap H \neq \emptyset$.

Seja $P \subset \mathbb{R}^n$ um poliedro. Um conjunto $F \subset P$ é uma face de $P$, se existe uma inequação $a^\top x \leq \alpha$ válida em relação a $P$, tal que $F = P \cap \{x \mid a^\top x = \alpha\}$.

$F$ é não trivial se $\emptyset \neq F \neq P$ e se $a^\top x \leq \alpha$ é válida em relação a $P$, dizemos que $P \cap \{x \mid a^\top x = \alpha\}$ é a face induzida ou definida por $a^\top x \leq \alpha$.

O conjunto de soluções ótimas do problema de maximizar ou minimizar uma função linear sobre um poliedro $P$ é sempre uma face de $P$.

Dado um poliedro $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$, denotamos por $ind(A)$ o conjunto dos índices das linhas da matriz $A$ e definimos o conjunto igualdade de $F$ como sendo $ig(F) = \{i \in ind(A) \mid A_{i*}x = b_i,\ \forall x \in F\}$.

Pode-se mostrar que a dimensão de uma face (isto é, o maior número de vetores afim-independentes na face menos um) é dada pela dimensão do núcleo de $A_{ig(F)*}$. Também podemos pensar em uma face como subconjunto do poliedro induzido por um subconjunto de restrições onde é imposta igualdade. Assim, quanto maior o posto de $ig(F)$, menor a dimensão da face.

Certas faces merecem destaque em um poliedro: as facetas. São definidas como faces próprias maximais, que não estão contidas em nenhuma outra face própria dele. Possuem dimensão igual à dimensão do poliedro menos um. Para poliedros de dimensão cheia ($P \subset \mathbb{R}^n$ com $dim(P) = n$) pode-se mostrar que as desigualdades que definem facetas são necessárias e suficientes na descrição do poliedro e, em geral, temos que a descrição de $P = \{x \mid A_{J*}x = b_J, A_{I*}x \leq b_I\}$ é irreduntante se, e somente se,, $A_{J*}$ é uma matriz de posto completo e as desigualdades em $I$ definem facetas distintas de $P$.

Esta característica das facetas faz com que, ao procurarmos desigualdades válidas para um poliedro, ser de especial importância encontrarmos desigualdades que definam facetas.

# 4 Equivalência entre Otimização e Separação

O método simplex é o método mais usado na resolução de problemas lineares sobre poliedros, mas, apesar de seu bom desempenho na prática, possui pior caso exponencial. A polinomialidade da programação linear só foi estabelecida em 1979, com o surgimento do método elipsóide, desenvolvido por Khachiyan. Apesar das importantes consequências teóricas desse algoritmo, ele não se mostra útil do ponto de vista prático, sendo o método simplex ainda usado nas implementações. Existe ainda a classe de algoritmos de pontos interiores, criada por Karmarkar em 1984 que, apesar de ser uma versão polinomial mais eficiente que o método elipsóide, não possui suas importantes consequências teóricas.

O problema considerado pelo método elipsóide é, dado um conjunto convexo limitado $P' \subset \mathbb{R}^n$, encontrar $x \in P'$. O problema de minimização de uma função objetivo $c^\top x$ (podemos assumir $c \in \mathbb{Z}^n$ sem perda de generalidade) sobre um poliedro $P$ pode ser reduzido polinomialmente ao problema de factibilidade através de sucessivas execuções sobre o conjunto $P' = P \cap \{x \mid c^\top x \leq d + 1/2\}$ em que $d \in \mathbb{Z}$ e o valor ótimo corresponde ao menor $d$ em que $P'$ é factível. Para encontrar esse $d$, precisamos fazer uma busca binária sobre o intervalo de valores possíveis da função objetivo. Se $P$ for o fecho convexo de um conjunto $S \subset \{0,1\}^n$ (caso de interesse dos POC), esse intervalo é $[-nc_{max}, nc_{max}]$, onde $c_{max} = max_i\ c_i$. O que leva a $O(log(nc_{max}))$ iterações, o que é polinomial.

Resumidamente, o algoritmo elipsóide funciona da seguinte maneira. Inicia com um grande elipsóide $E$ que contêm $P'$ (no caso de $S$ como descrito anteriormente, $E$ pode ser uma bola centrada em $(\frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2})$ com raio $\frac{1}{2}\sqrt{2}$) e verifica se o centro do elipsóide pertence a $P'$. Caso não pertença, encontra uma inequação $c^\top x \leq d_i$ válida para $P'$ e que não seja satisfeita pelo centro e gera um novo elipsóide que contêm a intersecção do elipsóide anterior com o semi-espaço definido pela desigualdade. Este novo elipsóide possui a importante propriedade de possuir volume menor que o anterior por um fator constante. Se $P'$ possuir dimensão completa, eventualmente o centro do elipsóide estará em $P'$.

Existem diversos detalhes e dificuldades que estão sendo omitidas nessa explicação, como o tratamento para o caso de $P'$ não possuir dimensão completa e problemas com a precisão e a estabilidade numérica do método, por exemplo. Mas o ponto mais importante para que o algoritmo seja executável em tempo polinomial é a etapa de se encontrar a inequação separadora. Este problema, em que dados um poliedro e um ponto não pertencente a este, deseja-se encontrar uma inequação válida para o poliedro e inválida para o ponto é chamado de PROBLEMA DA SEPARAÇÃO. O algoritmo elipsóide é, então, uma redução polinomial do problema de otimização para o problema da separação.

No caso em que conhecemos uma descrição completa do poliedro, ou seja, uma matriz $A$ e um vetor $b$ tais que $P = \{x \mid Ax \leq b\}$, o problema da separação é fácil (basta verificarmos qual desigualdade da descrição de $P$ o ponto viola) e o algoritmo elipsóide é, portanto, polinomial. Já nos problemas de programação inteira, em que não conhemos uma descrição completa do poliedro, a etapa da separação é mais difícil.

É interessante que em certos casos onde a descrição do poliedro envolve um número exponencial de desigualdades, ainda é possível realizar a separação em tempo polinomial. Isso ocorre no poliedro do problema da arborescência de custo mínimo (arborescência é uma árvore em um grafo orientado, com raiz $r$ determinada e que cobre todo o grafo), no poliedro do problema do emparelhamento de peso máximo e em uma classe de facetas do poliedro do problema do caixeiro viajante. Em todos esses casos, é usado um algoritmo para o cálculo de fluxos máximos e uma relação entre fluxo máximo e corte mínimo de uma rede, tema da próxima seção.

# 5 Fluxos Máximos e Cortes Mínimos

O problema de fluxo máximo considera um grafo orientado $G = (V, A)$ de $m$ vértices e $n$ arestas, com dois vértices destacados: $s$ e $t$. Cada aresta $(i, j)$ possui uma capacidade $d_{ij}$, sendo que $d_{ts} = \infty$. Procura-se um fluxo $x : A \to \mathbb{R}$ que solucione o problema:

$$
\begin{aligned}
& max\ x_{ts} \\
& \qquad \sum_{j \in \delta^+(i)} x_{ij} - \sum_{j \in \delta^-(i)} x_{ji} = 0 && \forall i \in V \\
& \qquad x_{ij} \leq d_{ij} && \forall (i,j) \in A \\
& \qquad x_{ij} \geq 0 && \forall (i,j) \in A
\end{aligned}
$$

onde $\delta^+(i) = \{j \mid (i, j) \in A\}$ e $\delta^-(i) = \{j \mid (j, i) \in A\}$.

A primeira restrição impõe conservação de fluxo, isto é, o fluxo que entra em cada nó deve ser igual ao fluxo que sai. As outras restrições dizem respeito à capacidade de cada arco. Observe que para um fluxo que maximiza $x_{ts}$, temos $x_{is} = 0$ para $i \neq t$ e $x_{tj} = 0$ para $j \neq s$ e assim, $x_{ts}$ é igual ao fluxo que está sendo enviado de $s$ para $t$, usando a rede.

No problema de corte mínimo, consideramos uma partição $(U, \bar{U})$ de $V$ tal que $s \in U$ e $t \in \bar{U}$. O conjunto de arestas $\delta^+(U) = \{(i, j) \in A \mid i \in U, j \in \bar{U}\}$ é chamado um corte $s - t$ e procuramos um corte de menor capacidade, isto é, $U \subset V$ tal que $\sum_{(i,j) \in \delta^+(U)} d_{ij}$ seja mínimo. Note que para qualquer fluxo factível, devemos ter $x_{ts} \leq \sum_{(i,j) \in \delta^+(U)} d_{ij}$ para qualquer corte $s - t$. E portanto, o fluxo máximo é menor ou igual ao corte mínimo.

Aqui, já podemos suspeitar que existe uma relação de dualidade entre ambos os problemas. Isso de fato ocorre, com as soluções ótimas dos dois problemas tendo valor igual. Isso pode ser demonstrado construindo o problema dual do problema de fluxo máximo, como apresentado anteriormente, e relacionando uma solução ótima deste com um corte mínimo. A seguir, será mostrado um algoritmo polinomial para o problema de fluxo máximo capaz de encontrar simultaneamente o corte mínimo.

Dado um fluxo factível $x$ (podemos inicializar o algoritmo com $x = 0$, por exemplo), considere o grafo auxiliar $G' = (V, A'(x))$ com $A'(x) = \{(i,j) \mid (i,j) \in A, x_{ij} < d_{ij}\} \cup \{(i,j) \mid (j,i) \in A, x_{ji} > 0\}$. O primeiro subconjunto será denominado conjunto dos arcos diretos e o segundo, conjunto dos arcos reversos.

A idéia principal no algoritmo é encontrar um caminho de $s$ a $t$ nesse grafo. Caso exista, podemos calcular $\Delta = \min\left\{\min_{(i,j) direto}(d_{ij} - x_{ij}), \min_{(i,j) reverso} x_{ji}\right\}$ e gerar um fluxo melhor aumentando o fluxo em $x_{ts}$ e nas arestas diretas por $\Delta$ e diminuindo o fluxo das arestas reversas por $\Delta$. Pela escolha de $\Delta$, podemos ver que o novo fluxo é factível.

Esse caminho pode ser encontrado via busca em largura no grafo, um processo de complexidade linear. Tomando o cuidado de visitar os vértices com alguma ordem determinada, pode-se mostrar que serão necessários no máximo $mn$ aumentos de fluxo para se atingir o máximo, de forma que o processo todo é polinomial.

O fluxo máximo é encontrado quando não é mais possível atingir $t$ a partir de $s$ no grafo $G'$. Faça $U$ igual ao conjunto de vértices que podem ser atingidos por $s$. Temos que $t \notin U$ e, como o algoritmo se encerrou, $U$ e $\bar{U}$ são desconexos. Assim, pela construção de $A'(x)$, temos que $x_{ij} = d_{ij}$ para todo $(i,j) \in \delta^+(U)$ e $x_{ij} = 0$ para todo $(i,j) \in \delta^-(U)$. Pelas restrições de conservação de fluxo, conclui-se que $x_{ts} = \sum_{(i,j) \in \delta^+(U)} x_{ij} = \sum_{(i,j) \in \delta^+(U)} d_{ij}$, o que determina o corte mínimo e mostra que o fluxo encontrado é máximo.

# 6 Conclusão

Neste estudo foi possível ver diversas técnicas usadas nos métodos de planos de corte em combinatória poliédrica. Além de introdução às técnicas de otimização combiinatória, essas ferramentas serão úteis para o estudo do problema do caixeiro viajante com coleta e entrega.

Para o próximo semestre, planeja-se o estudo do poliedro de problemas específicos e a implementação de um método de planos de corte usando as facetas estudadas.

# Referências Bibliográficas

[1] S. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing*, pages 151–158, 1971.

[2] Charles E.; Rivest Ronald L.; Stein Clifford Cormen, Thomas H.; Leiserson. *Introduction to Algorithms (third ed.)*. MIT Press, 2009.

[3] C.E. Ferreira and Y. Wakabayashi. *Combinatória Poliédrica e Planos-de-Corte Faciais*. X Escola de Computação, Unicamp, 1996.

[4] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.

[5] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, Chichester, 1986.