Minimum Hitting Set (weighted)

Mínimo Conjunto Transversal (unweighted)

- **Instância:** Um conjunto universo $U = \{u_1, \ldots, u_n\}$ e uma família $\mathcal{S} = \{S_1, \ldots, S_m\}$ de subconjuntos de U.
- **Objetivo:** Obter o menor subconjunto T (transversal) de U tal que todo conjunto $S_j \in \mathcal{S}$ tem algum elemento de T $(S_j \cap T \neq \emptyset)$.

Mínimo Conjunto Transversal (weighted)

- **Instância:** Idem + cada elemento $u \in U$ tem um custo c_u
- **Objetivo:** Obter um conjunto transversal $T \subseteq U$ $(S_j \cap T \neq \emptyset\ \forall j)$ de custo mínimo ($\sum_{u \in T} c_u$ mínimo).
- **Generalização:** Basta tomar custo 1 para cada elemento $u \in U$.

Hitting Set (CT) e Vertex Cover (CV)

Cobertura de Vértices (Vertex Cover)

- **Instância:** Grafo G e um custo $c(v)$ para cada vértice v de G
- **Objetivo:** Obter uma cobertura das arestas por vértices $v_1, \ldots, v_k$ de G de custo mínimo $(c(v_1) + \ldots + c(v_k)$ mínimo$)$.
- **Observação:** Cobrir arestas usando vértices

Conjunto Transversal CT (Hitting Set)

- **Instância:** Conj univ $U = \{u_1, \ldots, u_n\}$, família $\mathcal{S} = \{S_1, \ldots, S_m\}$ de subconjuntos de U e um custo c_u p/ cada elem $u \in U$
- **Objetivo:** Obter um conjunto transversal $T \subseteq U$ $(S_j \cap T \neq \emptyset\ \forall j)$ de custo mínimo $(\sum_{u \in T} c_u$ mínimo$)$.
- **Observação:** Cobrir conjuntos usando elementos
- **Redução CV→CT:** Para cada vértice v de G, crie um elemento v em U. Para cada aresta uv de G, crie o conjunto $\{u, v\}$ em $\mathcal{S}$.

Hitting Set (CT) e Set Cover (CC)

Set Cover (CC): cobrir elementos usando conjuntos

- **Instância:** Conj. univ. $U = \{u_1, \ldots, u_n\}$, família $\mathcal{S} = \{S_1, \ldots, S_m\}$ de subconjuntos de U e um custo $c(S_i)$ para cada conjunto S_i
- **Objetivo:** Obter uma cobertura $\mathcal{C} = \{S_{i_1}, \ldots, S_{i_k}\} \subseteq \mathcal{S}$ de U de custo mínimo ($c(S_{i_1}) + \ldots + c(S_{i_k})$ mínimo).

Hitting Set (CT): cobrir conjuntos usando elementos

- **Instância:** Conj univ $U = \{u_1, \ldots, u_n\}$, família $\mathcal{S} = \{S_1, \ldots, S_m\}$ de subconjuntos de U e um custo c_u p/ cada elem $u \in U$
- **Objetivo:** Obter um conjunto transversal $T \subseteq U$ ($S_j \cap T \neq \emptyset\ \forall j$) de custo mínimo ($\sum_{u \in T} c_u$ mínimo).

•• **Redução CC $\rightarrow$ CT:** P/ cada conj. S em Set Cover, crie elem. em Hitting Set. P/ cada elem. u em Set Cover, crie conj. em Hitting Set com os conj. de Set Cover que contém u.
•• **Exemplo:** $U = \{1, 2, 3, 4, 5\}$, $\mathcal{S} = \{\{1, 2, 3\}, \{2, 3, 4\}, \{3, 4, 5\}\}$
•• $\Rightarrow U' = \{a, b, c\}$, $\mathcal{S}' = \{\{a\}, \{a, b\}, \{a, b, c\}, \{b, c\}, \{c\}\}$
•• Freq. máx. f (Set Cover) $=$ tam. s maior conj de $\mathcal{S}$ (Hitting Set)

Min Hitting Set - Transversal custo min - Método Primal

Mínimo Conjunto Transversal (weighted)

- **Instância:** Conj univ $U = \{u_1, \ldots, u_n\}$, família $\mathcal{S} = \{S_1, \ldots, S_m\}$ de subconjuntos de U e um custo c_u para cada elemento $u \in U$
- **Objetivo:** Obter um conjunto transversal $T \subseteq U$ $(S_j \cap T \neq \emptyset \ \forall j)$ de custo mínimo $(\sum_{u \in T} c_u$ mínimo$)$.

Método Primal - Formulação de Programação Inteira

- Vetores c e x, indexados por U. **Ex:** c_u e x_u para cada $u \in U$
- Interpretação: $x_u = 1$ se $u \in T$; caso contrário $x_u = 0$
- **Minimizar** $c \cdot x \ = \ \sum_{u \in U} \ c_u x_u$, **restrito a**:
- **Para cada** $u \in U$: $\qquad x_u \ \in \ \{0, 1\}$
- **Para cada** $S \in \mathcal{S}$: $\quad \displaystyle\sum_{u \in S} x_u \ \geq \ 1$

- É viável; basta tomar $x_u = 1$ para todo elemento $u \in U$
- Obtém soℓ. exata $opt(U, \mathcal{S}, c) = c \cdot x_{opt}$, mas problema NP-Difícil.

Min Hitting Set - Transversal custo min - Método Primal

Mínimo Conjunto Transversal (weighted)

- **Instância:** Conj univ $U = \{u_1, \ldots, u_n\}$, família $\mathcal{S} = \{S_1, \ldots, S_m\}$ de subconjuntos de U e um custo c_u para cada elemento $u \in U$
- **Objetivo:** Obter um conjunto transversal $T \subseteq U$ ($S_j \cap T \neq \emptyset \forall j$) de custo mínimo ($\sum_{u \in T} c_u$ mínimo).

Método Primal - Relaxação (Programação Linear)

- Vetores c e x, indexados por U. **Ex:** c_u e x_u para cada $u \in U$
- **Minimizar** $c \cdot x = \sum_{u \in U} c_u x_u$, **restrito a**:
- **Para cada** $u \in U$: $\quad x_u \in [0,1]$
- **Para cada** $S \in \mathcal{S}$: $\quad \sum_{u \in S} x_u \geq 1$

- É viável; basta tomar $x_u = 1$ para todo elemento $u \in U$
- $opt(U, \mathcal{S}, c) \geq c \cdot x_{opt}$, pois a relaxação pode obter valores menores.

Min Hitting Set - Transversal custo min - Método Primal

Técnica do arredondamento

Algoritmo MinTrasversal(U, $\mathcal{S}$, c)

1. **seja** x uma solução ótima racional da PL (relax. PI)
2. **seja** $s = \max\{|S| : S \in \mathcal{S}\}$
3. **seja** $T = \{u \in U : x_u \geq 1/s\}$
4. **retorne** T

LEMA: T é um transversal de $\mathcal{S}$
Prova: Para cada $S \in \mathcal{S}$ vale a restrição $\sum_{u \in S} x_u \geq 1$
$\Rightarrow \exists u \in S : x_u \geq 1/|S| \geq 1/s$.

TEOREMA: O algoritmo MinTransversal é ***s*-aproximativo**
Prova: Como $s \cdot x_u \geq 1, \forall u \in U$, então:
$c(T) = \sum_{u \in T} c_u \leq s \cdot \sum_{u \in T} c_u x_u \leq s \cdot \sum_{u \in U} c_u x_u \leq s \cdot opt(U, \mathcal{S}, c)$

Min Hitting Set - Transversal custo min - Método Dual

Problema Primal

$$\text{Max} \quad Z = 3x_1 + 5x_2$$

Sujeito a

$$\begin{cases} x_1 & & \leq 4 \\ & 2x_2 & \leq 12 \\ 3x_1 & + \; 2x_2 & \leq 18 \end{cases}$$

$$x_1 \text{ e } x_2 \geq 0$$

Notação Matricial

$$\text{Max} \quad Z = \begin{bmatrix} 3 & 5 \end{bmatrix} . \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Sujeito a

$$\begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 3 & 2 \end{bmatrix} . \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix}$$

$$\text{e} \quad \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Problema Dual

$$\text{Min} \quad W = 4y_1 + 12y_2 + 18y_3$$

Sujeito a

$$\begin{cases} y_1 & + & & 3y_3 & \geq 3 \\ & & 2y_2 \; + & 2y_3 & \geq 5 \end{cases}$$

$$y_1, \; y_2 \text{ e } y_3 \geq 0$$

Notação Matricial

$$\text{Min} \quad W = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix}$$

Sujeito a

$$\begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 3 & 2 \end{bmatrix} \geq \begin{bmatrix} 3 & 5 \end{bmatrix}$$

$$\text{e} \quad \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix} \geq \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

Teorema Forte da Dualidade: opt do primal = opt do dual

Min Hitting Set - Transversal custo min - Método Dual

Método Primal - Relaxação (Programação Linear)

- Vetores c e x, indexados por U. **Ex:** c_u e x_u para cada $u \in U$
- **Minimizar** $c \cdot x \;=\; \sum_{u \in U} \; c_u x_u$, **restrito a**:
- **Para cada** $u \in U$: $\quad x_u \;\in\; [0, 1]$
- **Para cada** $S \in \mathcal{S}$: $\quad \sum_{u \in S} x_u \;\geq\; 1$

Método Dual - Relaxação (Programação Linear)

- Vetor y, indexado por $\mathcal{S}$. **Ex:** y_S para cada $S \in \mathcal{S}$
- **Maximizar** $y \cdot \mathbf{1} \;=\; \sum_{S \in \mathcal{S}} \; y_S$, **restrito a**:
- **Para cada conj** $S \in \mathcal{S}$: $\quad y_S \;\geq\; 0$
- **Para cada elemento** $u \in U$: $\quad \sum_{S \in \mathcal{S}(u)} y_S \;\leq\; c_u \qquad (R1)$

onde $\mathcal{S}(u) = \{S \in \mathcal{S} : \; u \in S\}$.

- É viável; basta tomar $y_S = 0$ para todo conjunto $S \in \mathcal{S}$
- $opt(U, \mathcal{S}, c) \;\geq\; c \cdot x_{opt} \;=\; y_{opt} \cdot \mathbf{1}$, pelo Teorema da Dualidade.

Min Hitting Set - Transversal custo min - Método Dual

Interpretação do Dual

- Custo de cada elem u é repartido entre ele e os conj S que o contém
- Se um conjunto S não tem um elemento com igualdade na restrição R1, podemos aumentar o valor de y_S. Ou seja, no ótimo, todo conjunto S tem um elemento u com igualdade na restrição R1.
- Colocar no transversal os elem $u \in U$ com igualdade na restr R1.

Método Dual - Relaxação (Programação Linear)

- Vetor y, indexado por $\mathcal{S}$. **Ex:** y_S para cada $S \in \mathcal{S}$
- **Maximizar** $y \cdot \mathbf{1} \;=\; \sum_{S \in \mathcal{S}} y_S$, **restrito a**:
- **Para cada conj** $S \in \mathcal{S}$: $\quad y_S \;\geq\; 0$
- **Para cada elemento** $u \in U$: $\quad \displaystyle\sum_{S \in \mathcal{S}(u)} y_S \;\leq\; c_u \qquad (R1)$

onde $\mathcal{S}(u) = \{S \in \mathcal{S} : \; u \in S\}$.

- É viável; basta tomar $y_S = 0$ para todo conjunto $S \in \mathcal{S}$
- $opt(U, \mathcal{S}, c) \;\geq\; c \cdot x_{opt} \;=\; y_{opt} \cdot \mathbf{1}$, pelo Teorema da Dualidade.

Min Hitting Set - Transversal custo min - Método Dual

Algoritmo MinTrasversal-dual (U, $\mathcal{S}$, c)

1. **seja** y uma solução ótima racional da PL (dual, relax. PI)
2. **seja** $T = \{u \in U : \sum_{S \in \mathcal{S}:\ u \in S} y_S = c_u\}$ (igualdade em R1)
3. **retorne** T

LEMA: T é um transversal de $\mathcal{S}$ com elementos de U
Prova: Já visto.

TEOREMA: MinTransversal-dual é ***s*-aproximativo**
Prova: Seja $s = \max\{|S| : S \in \mathcal{S}\}$ e $\mathcal{S}(u) = \{S \in \mathcal{S} : u \in S\}$.

$$c(T) = \sum_{u \in T} c_u = \sum_{u \in T} \sum_{S \in \mathcal{S}(u)} y_S \leq \sum_{u \in U} \sum_{S \in \mathcal{S}(u)} y_S = \sum_{S \in \mathcal{S}} |S| \cdot y_S$$

$$c(T) \leq s \cdot \sum_{S \in \mathcal{S}} y_S \leq s \cdot opt(U, \mathcal{S}, c)$$

O Esquema Primal-Dual Clássico

Programa Primal Clássico: n variáveis x_j com m restrições b_i

- **Minimizar** $c \cdot x = \sum_{j=1}^{n} c_j x_j$, **onde** $x_j \geq 0, \forall j = 1, \ldots, n$
- Restrito à $\sum_{j=1}^{n} a_{ij} x_j \geq b_i$ para todo $i = 1, \ldots, m$

Programa Dual Clássico: m variáveis y_i com n restrições c_j

- **Maximizar** $b \cdot y = \sum_{i=1}^{m} b_i y_i$, onde $y_i \geq 0, \forall i = 1, \ldots, m$
- **Restrito à** $\sum_{i=1}^{m} a_{ij} y_i \leq c_j$ para todo $j = 1, \ldots, n$

Teoremas da Dualidade e das Folgas Complementares

- **Dualidade Forte:** Se viáveis, $c \cdot x \geq b \cdot y$ e $c \cdot x_{opt} = b \cdot y_{opt}$
- **Folgas Complementares:** $c \cdot x = b \cdot y$ se e só se valem as folgas completares primais e duais:
  - Primais: $x_j = 0$ ou $\sum_{i=1}^{m} a_{ij} y_i = c_j$, para todo $j = 1, \ldots, n$
  - Duais: $y_i = 0$ ou $\sum_{j=1}^{n} a_{ij} x_j = b_i$, para todo $i = 1, \ldots, m$

- **Esquema PL:** Começa com solução dual y viável e solução primal x inviável. Observando as folgas complementares, vai melhorando a otimalidade da dual y e a viabilidade da primal x **até** $c \cdot x = b \cdot y$.

O Esquema Primal-Dual de Aproximação

Programa Primal Clássico: n variáveis x_j com m restrições b_i

- **Minimizar** $c \cdot x = \sum_{j=1}^{n} c_j x_j$, **onde** $x_j \geq 0, \forall j = 1, \ldots, n$
- Restrito à $\sum_{j=1}^{n} a_{ij} x_j \geq b_i$ para todo $i = 1, \ldots, m$

Programa Dual Clássico: m variáveis y_i com n restrições c_j

- **Maximizar** $b \cdot y = \sum_{i=1}^{m} b_i y_i$, onde $y_i \geq 0, \forall i = 1, \ldots, m$
- **Restrito à** $\sum_{i=1}^{m} a_{ij} y_i \leq c_j$ para todo $j = 1, \ldots, n$

Folgas Complementares (α, β)-aproximadas

- Primais: $x_j = 0$ ou $c_j/\alpha \leq \sum_{i=1}^{m} a_{ij} y_i \leq c_j$, para todo $j = 1, \ldots, n$
- Duais: $y_i = 0$ ou $b_i \leq \sum_{j=1}^{n} a_{ij} x_j \leq \beta b_i$, para todo $i = 1, \ldots, m$
- **Lema:** Se x e y satisfazem folgas complementares (α, β)-aproximadas, então $c \cdot x \leq (\alpha\beta)\ b \cdot y$. **Prova:**

$$\sum_{j=1}^{n} c_j x_j \leq \alpha \sum_{j=1}^{n} \left(\sum_{i=1}^{m} a_{ij} y_i \right) x_j = \alpha \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij} x_j \right) y_i \leq \alpha\beta \sum_{j=1}^{n} b_i y_i$$

O Esquema Primal-Dual de Aproximação

Programa Primal Clássico: n variáveis x_j com m restrições b_i

- **Minimizar** $c \cdot x \; = \; \sum_{j=1}^{n} c_j x_j$, **onde** $x_j \geq 0, \forall j = 1, \ldots, n$
- Restrito à $\sum_{j=1}^{n} a_{ij} x_j \geq b_i$ para todo $i = 1, \ldots, m$

Programa Dual Clássico: m variáveis y_i com n restrições c_j

- **Maximizar** $b \cdot y = \sum_{i=1}^{m} b_i y_i$, onde $y_i \geq 0, \forall i = 1, \ldots, m$
- **Restrito à** $\sum_{i=1}^{m} a_{ij} y_i \leq c_j$ para todo $j = 1, \ldots, n$

Folgas Complementares (α, β)-aproximadas

- Primais: $x_j = 0$ ou $c_j/\alpha \leq \sum_{i=1}^{m} a_{ij} y_i \leq c_j$, para todo $j = 1, \ldots, n$
- Duais: $y_i = 0$ ou $b_i \leq \sum_{j=1}^{n} a_{ij} x_j \leq \beta b_i$, para todo $i = 1, \ldots, m$
- **Aprox Primal-Dual:** Começa com soluções y dual viável e x primal inviável (mas sempre inteira). Observ folgas compl (α, β)-aprox, vai melhorando otimal. dual y e viab. primal x. No fim, a solução dual vezes $\alpha\beta$ serve como limite superior para o do primal inteiro.

Hitting Set - Transversal custo min - Método Primal-Dual

Programa Primal - Programação Inteira

- Vetor x, indexado por U. **Ex:** x_u para cada $u \in U$
- **Minimizar** $c \cdot x \;=\; \sum_{u \in U} c_u x_u$, **restrito a**:
- **Para cada elemento** $u \in U$: $\quad x_u \;\in\; \{0,1\}$
- **Para cada conjunto** $S \in \mathcal{S}$: $\quad \displaystyle\sum_{u \in S} x_u \;\geq\; 1$

- É viável; basta tomar $x_u = 1$ para todo elemento $u \in U$

Programa Dual - Relaxação (Programação Linear)

- Vetor y, indexado por $\mathcal{S}$. **Ex:** y_S para cada $S \in \mathcal{S}$
- **Maximizar** $y \cdot \mathbf{1} \;=\; \sum_{S \in \mathcal{S}} y_S$, **restrito a**:
- **Para cada conj** $S \in \mathcal{S}$: $\quad y_S \;\geq\; 0$
- **Para cada elemento** $u \in U$: $\quad \displaystyle\sum_{S \in \mathcal{S} : u \in S} y_S \;\leq\; c_u \qquad (R1)$

- É viável; basta tomar $y_S = 0$ para todo conjunto $S \in \mathcal{S}$
- Primal-Dual início: $x_u = 0\ \forall u \in U$ e $y_S = 0\ \forall S \in \mathcal{S}$.

Hitting Set - Transversal custo min - Método Primal-Dual

Algoritmo MinTransversal-primal-dual $(U, \mathcal{S}, c)$

1. **seja** y o vetor com $y_S = 0$ para todo conjunto $S \in \mathcal{S}$
2. **enquanto** $\exists$ conj S' sem nenhum elemento com (=R1)
3. **faça** $y_{S'} \leftarrow y_{S'} + \min\left\{c_u - \sum_{S \in \mathcal{S}: u \in S} y_S : \; u \in S'\right\}$
4. **seja** $T = \{u \in U : \sum_{S \in \mathcal{S}: u \in S} y_S = c_u\}$ (igualdade em R1)
5. **retorne** T

- Vetor x não está explícito, mas $x_u = 1$ se u com (=R1) e $x_u = 0$ cc.

Programa Dual - Relaxação (Programação Linear)

- Vetor y, indexado por $\mathcal{S}$. **Ex:** y_S para cada $S \in \mathcal{S}$
- **Maximizar** $y \cdot \mathbf{1} = \sum_{S \in \mathcal{S}} y_S$, **restrito a**:
- **Para cada conj** $S \in \mathcal{S}$: $y_S \geq 0$
- **Para cada elemento** $u \in U$: $\sum_{S \in \mathcal{S}: u \in S} y_S \leq c_u$ $(R1)$

Hitting Set - Transversal custo min - Método Primal-Dual

Algoritmo MinTransversal-primal-dual $(U, \mathcal{S}, c)$

1. **seja** y o vetor com $y_S = 0$ para todo conjunto $S \in \mathcal{S}$
2. **enquanto** $\exists$ conj S' sem nenhum elemento com (=R1)
3. **faça** $y_{S'} \leftarrow y_{S'} + \min\left\{c_u - \sum_{S\in\mathcal{S}:u\in S} y_S : \ u \in S'\right\}$
4. **seja** $T = \{u \in U : \sum_{S\in\mathcal{S}:u\in S} y_S = c_u\}$ (igualdade em R1)
5. **retorne** T

LEMA: T é um transversal de $\mathcal{S}$ por elementos de U

TEOREMA: MinTransversal-primal-dual é **s-aproximativo**
Prova: Seja $s = \max\{|S| : \ S \in \mathcal{S}\}$ e $\mathcal{S}(u) = \{S \in \mathcal{S} : \ u \in S\}$.
Bastaria $\alpha = 1$ e $\beta = s$. **Outra prova:** P/ transversal opt T^*:

$$\sum_{S\in\mathcal{S}} y_S \leq \sum_{u\in T^*} \sum_{S\in\mathcal{S}(u)} y_S \leq \sum_{u\in T^*} c_u = opt.$$ Logo, p/ transv. T retorn p/ algoritm

$$c(T) = \sum_{u\in T} c_u = \sum_{u\in T} \sum_{S\in\mathcal{S}(u)} y_S \leq \sum_{u\in U} \sum_{S\in\mathcal{S}(u)} y_S \leq s \cdot \sum_{S\in\mathcal{S}} y_S \leq s \cdot opt(U, \mathcal{S}, c)$$

Conclusão: Set Cover, Vertex Cover e Hitting Set

Set Cover (Cobertura por Conjuntos)

- $(1 + \ln n)$-aproximação: Algoritmo MinCC-Chvátal.
- f-aprox: Alg. MinCC-Hochbaum (versões primal, dual e primal-dual)
- onde $f = \max_{u \in U}\{f(u)\}$ é a frequência máx entre os elem. de U,
- onde $f(u)$ é o número de subconjuntos $S \in \mathcal{S}$ que contém u

Vertex Cover (Cobertura de Vértices)

- 2-aprox: Alg. MinCV-Hochbaum (versões primal, dual e primal-dual)

Hitting Set (Conjunto Tranversal)

- $(1 + \ln |\mathcal{S}|)$-aproximação: Algoritmo MinCT-Chvátal.
- s-aprox: Alg. MinTransversal (versões primal, dual e primal-dual)
- onde $s = \min\{|S| : S \in \mathcal{S}\}$ (tam. do maior conj. em $\mathcal{S}$)