Gems of TCS

Integer Linear Programming

Sasha Golovnev
October 4, 2021
AVOIDING SCURVY

• Orange costs $1, grapefruit costs $1; we have budget of $2/day

• Orange weighs 100gm, grapefruit weighs 200gm, we can carry 300gm

• Orange has 100gm of vitamin C, grapefruit has 150gm of vitamin C, maximize daily vitamin C intake.
AVOIDING SCURVY. PLOT

\[
\max 2x + 3y \\
\]

\[
x + y \leq 2 \\
x + 2y \leq 3 \\
x \geq 0 \\
y \geq 0 \\
\]
AVOIDING SCURVY. PLOT

\[
\begin{align*}
\text{max } & 2x + 3y \\
x + y & \leq 2 \\
x + 2y & \leq 3 \\
x & \geq 0 \\
y & \geq 0
\end{align*}
\]
AVOIDING SCURVY. PLOT

\[
\begin{align*}
\text{max } 2x + 3y \\
x + y &\leq 2 \\
x + 2y &\leq 3 \\
x &\geq 0 \\
y &\geq 0
\end{align*}
\]
\textbf{AVOIDING SCURVY. PLOT}

\[
\begin{align*}
\text{max} & \ 2x + 3y \\
 x + y & \leq 2 \\
 x + 2y & \leq 3 \\
 x & \geq 0 \\
 y & \geq 0
\end{align*}
\]
AVOIDING SCURVY. PLOT

\[
\begin{align*}
\text{max } 2x + 3y \\
\text{s.t. } x + y &\leq 2 \\
x + 2y &\leq 3 \\
x &\geq 0 \\
y &\geq 0
\end{align*}
\]
Avoiding Scurvy. Plot

\[
\begin{align*}
\text{max} & \quad 2x + 3y \\
\text{x + y} & \leq 2 \\
x + 2y & \leq 3 \\
x & \geq 0 \\
y & \geq 0
\end{align*}
\]
Avoiding Scurvy. Plot

\[ \text{max } 2x + 3y \]
\[ x + y \leq 2 \]
\[ x + 2y \leq 3 \]
\[ x \geq 0 \]
\[ y \geq 0 \]
AVOIDING SCURVY. PLOT

\[ \begin{align*}
\text{max } & 2x + 3y \\
x + y & \leq 2 \\
x + 2y & \leq 3 \\
x & \geq 0 \\
y & \geq 0
\end{align*} \]
AVOIDING SCURVY. PLOT

\[
\begin{align*}
\text{max } & 2x + 3y \\
\text{s.t. } & x + y \leq 2 \\
& x + 2y \leq 3 \\
& x \geq 0 \\
& y \geq 0
\end{align*}
\]
Avoiding Scurvy. Plot

\[
\text{max } 2x + 3y \\
\begin{align*}
x + y &\leq 2 \\
x + 2y &\leq 3 \\
x &\geq 0 \\
y &\geq 0
\end{align*}
\]
AVOIDING SCURVY. PLOT

\[
\begin{align*}
\text{max} & \quad 2x + 3y \\
\text{s.t.} & \quad x + y \leq 2 \\
& \quad x + 2y \leq 3 \\
& \quad x \geq 0 \\
& \quad y \geq 0
\end{align*}
\]
Avoiding Scurvy II

\[
\begin{align*}
\text{max } 2x + 3y \\
x + y & \leq 2 \\
x + 2y & \leq 2.5 \\
x & \geq 0 \\
y & \geq 0 
\end{align*}
\]
**AVOIDING SCURVY II**

\[
\text{max } 2x + 3y \\
x + y \leq 2 \\
x + 2y \leq 2.5 \\
x \geq 0 \\
y \geq 0
\]
max \(2x + 3y\)

\[x + y \leq 2\]
\[x + 2y \leq 2.5\]
\[x \geq 0\]
\[y \geq 0\]
\[ \text{max } 2x + 3y \]
\[ x + y \leq 2 \]
\[ x + 2y \leq 2.5 \]
\[ x \geq 0 \]
\[ y \geq 0 \]
**Avoiding Scurvy II**

\[
\begin{align*}
\text{max} \ 2x + 3y \\
& \quad x + y \leq 2 \\
& \quad x + 2y \leq 2.5 \\
& \quad x \geq 0 \\
& \quad y \geq 0
\end{align*}
\]
AVOIDING SCURVY II

\[ \text{max } 2x + 3y \]

\[ x + y \leq 2 \]
\[ x + 2y \leq 2.5 \]
\[ x \geq 0 \]
\[ y \geq 0 \]
Linear programming

**Input:** A set of linear inequalities $Ax \leq b$.

**Output:** Real solution that optimizes the objective function.
<table>
<thead>
<tr>
<th><strong>Integer linear programming</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> A set of linear inequalities $Ax \leq b$.</td>
</tr>
<tr>
<td><strong>Output:</strong> Integer solution that optimizes the objective function.</td>
</tr>
</tbody>
</table>
Example

\[ x_1 \geq 0.5 \]
\[ -x_1 + 8x_2 \geq 0 \]
\[ -x_1 - 8x_2 \geq -8 \]
Example

\[ x_1 \geq 0.5 \]

\[-x_1 + 8x_2 \geq 0 \]

\[-x_1 - 8x_2 \geq -8 \]
$x_1 \geq 0.5$

$-x_1 + 8x_2 \geq 0$

$-x_1 - 8x_2 \geq -8$
Example

\[ x_1 \geq 0.5 \]
\[ -x_1 + 8x_2 \geq 0 \]
\[ -x_1 - 8x_2 \geq -8 \]
Example

\[ x_1 \geq 0.5 \]

\[ -x_1 + 8x_2 \geq 0 \]

\[ -x_1 - 8x_2 \geq -8 \]
<table>
<thead>
<tr>
<th><strong>LP</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Find a <strong>real</strong> solution of a system of linear inequalities</td>
<td></td>
</tr>
</tbody>
</table>
**LP**

Find a *real* solution of a system of linear inequalities

Can be solved efficiently (Lecture 9)
<table>
<thead>
<tr>
<th>LP</th>
<th>Find a <strong>real</strong> solution of a system of linear inequalities</th>
<th>Can be solved efficiently (Lecture 9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILP</td>
<td>Find an <strong>integer</strong> solution of a system of linear inequalities</td>
<td></td>
</tr>
<tr>
<td>LP</td>
<td>ILP</td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td></td>
</tr>
<tr>
<td><strong>Find a real solution of a system of linear inequalities</strong></td>
<td><strong>Find an integer solution of a system of linear inequalities</strong></td>
<td></td>
</tr>
<tr>
<td>Can be solved efficiently (Lecture 9)</td>
<td>No polynomial algorithm known!</td>
<td></td>
</tr>
</tbody>
</table>
Algorithm for ILP

\[
\begin{align*}
\text{max } 2x + y \\
4x + y &\leq 33 \\
3x + 4y &\leq 29 \\
x &\geq 0 \\
y &\geq 0 \\
x, y &\in \mathbb{Z}
\end{align*}
\]
Algorithm for ILP

\[
\begin{align*}
\text{max} & \quad 2x + y \\
4x + y & \leq 33 \\
3x + 4y & \leq 29 \\
x & \geq 0 \\
y & \geq 0 \\
x, y & \in \mathbb{Z}
\end{align*}
\]
Linear Programming

(max: 9 variables)

Optimize: Max

Objective Function: 2x + y

Subject to: 4x + y <= 33, 3x + 4y <= 29, x >= 0, y >= 0

and: y >= 0

More constraints (optional):

Solve

Global maximum:

\[
\max \{2x + y \mid 4x + y \leq 33 \land 3x + 4y \leq 29 \land x \geq 0 \land y \geq 0\} \approx 17.1538 \text{ at } (x, y) \approx (7.92308, 1.30769)
\]
Branching on $x$

Original
$OPT \approx 17.1538$
BRANCHING ON $x$

Original
$OPT \approx 17.1538$

- $x \leq 7$
  - Prob 1
- $x \geq 8$
  - Prob 2
\[ \max 2x + y \]

\[ 4x + y \leq 33 \]

\[ 3x + 4y \leq 29 \]

\[ x \geq 0 \]

\[ y \geq 0 \]

\[ x, y \in \mathbb{Z} \]
\begin{align*}
\text{max } 2x + y \\
4x + y &\leq 33 \\
3x + 4y &\leq 29 \\
x &\geq 0 \\
y &\geq 0 \\
x, y &\in \mathbb{Z}
\end{align*}
Linear Programming

(max: 9 variables)

Optimize: Max

Objective Function: 2x + y

Subject to: 4x + y ≤ 33, 3x + 4y ≤ 29,
x ≥ 0,
y ≥ 0,

and:

More constraints (optional): x ≤ 7

Global maximum:

\[
\max\{2x + y | 4x + y \leq 33 \land 3x + 4y \leq 29 \land x \geq 0 \land y \geq 0 \land x \leq 7\} = 16
\]

at (x, y) = (7, 2)
BRANCHING ON $x$

Original
OPT $\approx 17.1538$

$x \leq 7$
Prob 1
OPT = 16

$x \geq 8$
Prob 2
Linear Programming Solver

Linear Programming

(max: 9 variables)

Optimize: Max

Objective Function: 2x+y

Subject to:
4x+y<=33,
3x+4y<=29,
x>=0,
y>=0,

and:

More constraints(optional):
x>=8

More constraints(optional):

(Solve)

www.ordsworks.com** *(constraints separator: ",")

Global maximum:

\[
\max\{2x+y\mid 4x+y \leq 33 \land 3x+4y \leq 29 \land x \geq 0 \land y \geq 0 \land x \geq 8\} = 17
\]

at \((x, y) = (8, 1)\)
BRANCHING ON $x$

Original
OPT $\approx 17.1538$

$x \leq 7$

Prob 1
OPT = 16

$x \geq 8$

Prob 2
OPT = 17
HEURISTIC ALGORITHMS FOR ILP
Applications
APPLICATIONS

• Scheduling
• Planning
• Networks
• ...
VERTEX COVERS

• A **Vertex Cover** of a graph $G$ is a set of vertices $C$ such that every edge of $G$ is connected to some vertex in $C$. 
A **Vertex Cover** of a graph $G$ is a set of vertices $C$ such that every edge of $G$ is connected to some vertex in $C$.

A **Minimum Vertex Cover** is a vertex cover of the smallest size.
VERTEX COVERS: EXAMPLES
VERTEX COVERS: EXAMPLES
**Vertex Cover as ILP**

- Introduce binary variable for every vertex: $x_1, \ldots, x_n$:
  - $x_i = 1$ iff $x_i$ belongs to Vertex Cover
Vertex Cover as ILP

- Introduce binary variable for every vertex: $x_1, \ldots, x_n$:
  - $x_i = 1$ iff $x_i$ belongs to Vertex Cover
- $\forall i \in \{1, \ldots, n\}$, $0 \leq x_i \leq 1$, $x_i \in \mathbb{Z}$
Vertex Cover as ILP

• Introduce binary variable for every vertex: $x_1, \ldots, x_n$:
  • $x_i = 1$ iff $x_i$ belongs to Vertex Cover
  • $\forall i \in \{1, \ldots, n\}, \ 0 \leq x_i \leq 1, x_i \in \mathbb{Z}$

• $\min \sum_i x_i$
INTRODUCE BINARY VARIABLE FOR EVERY VERTEX:

• $x_1, \ldots, x_n$:
  • $x_i = 1$ iff $x_i$ belongs to Vertex Cover

• $\forall i \in \{1, \ldots, n\}$, $0 \leq x_i \leq 1$, $x_i \in \mathbb{Z}$

• $\min \sum_i x_i$

• For every edge $(u, v)$ in the graph: $x_u + x_v \geq 1$
import networkx as nx
from mip import *

g = nx.Graph()
g.add_edges_from([(1, 2), (1, 3), (1, 5), (1, 6), (2, 5), (2, 0),
                    (3, 4), (3, 5), (3, 6), (5, 6), (7, 0)])

m = Model()
n = g.number_of_nodes()
x = [m.add_var(var_type=BINARY) for i in range(n)]
for u, v in g.edges() :
    m += x[u]+x[v] >= 1
m.objective = minimize(xsum(x[i] for i in range(n)))
m.optimize()

selected = [i for i in range(n) if x[i].x >= 0.99]
print("selected items: {}".format(selected))
Is it possible to place $n$ queens on an $n \times n$ board such that no two of them attack each other?
N QUEENS AS ILP

- $n^2$ 0/1-variables: for $0 \leq i, j < n$, $x_{ij} = 1$ iff queen is placed into cell $(i, j)$
N QUEENS AS ILP

- \( n^2 \) 0/1-variables: for \( 0 \leq i, j < n \), \( x_{ij} = 1 \) iff queen is placed into cell \((i, j)\)
- For \( 0 \leq i < n \), \( i \)th row contains exactly \( 1 \) queen:
  \[
  \sum_{j=1}^{n} x_{ij} = 1.
  \]
N QUEENS AS ILP

- $n^2$ 0/1-variables: for $0 \leq i, j < n$, $x_{ij} = 1$ iff queen is placed into cell $(i, j)$
- For $0 \leq i < n$, $i$th row contains $= 1$ queen:
  $$\sum_{j=1}^{n} x_{ij} = 1.$$  
- For $0 \leq j < n$, $j$th column contains $= 1$ queen:
  $$\sum_{j=1}^{n} x_{ij} = 1.$$
N QUEENS AS ILP

• $n^2$ 0/1-variables: for $0 \leq i, j < n$, $x_{ij} = 1$ iff queen is placed into cell $(i, j)$
• For $0 \leq i < n$, $i$th row contains $= 1$ queen:
  \[ \sum_{j=1}^{n} x_{ij} = 1. \]
• For $0 \leq j < n$, $j$th column contains $= 1$ queen:
  \[ \sum_{i=1}^{n} x_{ij} = 1. \]
• Each diagonal contains $\leq 1$ queen:
  \[ \sum_{i=1}^{n} \sum_{j=1: i-j=k}^{n} x_{ij} \leq 1; \quad \sum_{i=1}^{n} \sum_{j=1: i+j=k}^{n} x_{ij} \leq 1 \]