MATHEMATICAL FORMULATIONS FOR INTEGER PROGRAMMING PROBLEMS

Luciana Salete Buriol
buriol@inf.ufrgs.br

XXI ELAVIO, Miramar - Argentina — 2017

- Thanks Willy!

- Thanks Willy!
- Enjoy, ELAVIO is an unique event.

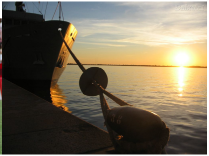
- Thanks Willy!
- Enjoy, ELAVIO is an unique event.

OUTLINE

1. Basic restrictions with binary variables
2. Non-linear and piecewise linear functions
3. Flow and path formulations
4. Subtour elimination
5. Hard vs. Soft constraints
6. Historical Notes
7. Historical Developments

COMPUTER SCIENCE AT UFRGS (FEDERAL UNIVERSITY OF RIO GRANDE DO SUL)

COMPUTER SCIENCE AT UFRGS (FEDERAL UNIVERSITY OF RIO GRANDE DO SUL)



- UFRGS is located in Porto Alegre, the Capital of Rio Grande do Sul;
- About 1.4M in habitants

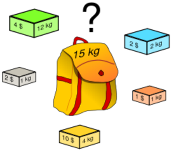
COMPUTER SCIENCE AT UFRGS (FEDERAL UNIVERSITY OF RIO GRANDE DO SUL)

- Post Graduation in Computer Science (PPGC) was created in 1972 and is among the first graduate programs in Computer Science in Brazil;
- Has about 330 PhD and MScs students, and already formed about 220 PhDs and 1330 MScs;
- 75 full-professors graduated in important institutions around the world;
- Ranked among the top-5 PPGC in Brazil.

MATHEMATICAL MODEL

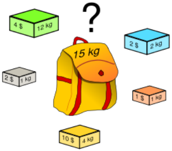
- Decision variables: quantified decisions of the problem;
- Objective function: performance measure;
- Constraints: limit the values of variables;
- Parameters: input data.

0-1 KNAPSACK PROBLEM



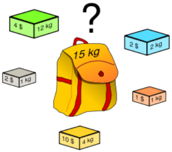
- Given n itens $N = \{1, 2, ...n\}$,
- each with a profit p_i and a weight w_i, and a knapsack weight restriction K.
- Select a subset of the items so that the total weight is less than or equal to K, and the total value is as large as possible.

0-1 KNAPSACK PROBLEM



- Given n itens $N = \{1, 2, ...n\}$,
- each with a profit p_i and a weight w_i, and a knapsack weight restriction K.
- Select a subset of the items so that the total weight is less than or equal to K, and the total value is as large as possible.
- Which are the decision variables?

0-1 KNAPSACK PROBLEM

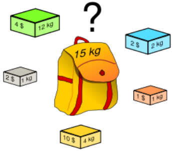


$$max \sum_{i=1}^{n} v_i x_i$$

$$s.t. \sum_{i \in N} w_i x_i \leq K$$

$$x_i \in \{0, 1\}$$

UNBOUNDED KNAPSACK PROBLEM

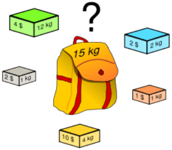


$$max \sum_{i=1}^{n} v_i x_i$$

$$s.t. \sum_{i \in N} w_i x_i \leq K$$

$$x_i \in \mathcal{Z}$$

(BOUNDED) KNAPSACK PROBLEM



$$max \sum_{i=1}^{n} v_i x_i$$

$$s.t. \sum_{i \in N} w_i x_i \leq K$$

$$x_i \leq d_i \quad \forall i$$

$$x_i \in \mathcal{Z}$$

INTEGER LINEAR PROGRAMMING

- Linear Programing (LP)

$$\begin{aligned} max \quad & c^t x \\ & Ax \leq b \\ & x \in \mathcal{R}^n \geq 0 \end{aligned}$$

- Integer Programing (IP)

$$\begin{aligned} max \quad & h^t y \\ & Gy \leq b \\ & y \in \mathcal{Z}^n \geq 0 \end{aligned}$$

MIXED INTEGER PROGRAMMING (MIP)

- Mixed Integer Programming

$$\begin{aligned} max \quad & c^t x + h^t y \\ & Ax + Gy \leq b \\ & x \in \mathcal{R}^n \geq 0, y \in \mathcal{Z}^n \geq 0 \end{aligned}$$

- LP and IP are special cases of MIP.
- Other special cases: 0-1-MIP e 0-1-IP.

$$x \in \mathcal{B}^n$$

Variables $x, y \in \mathcal{B}$: selection of objects.

- Or:

$$x + y \geq 1 \qquad x, y \in \mathcal{B}$$

Variables $x, y \in \mathcal{B}$: selection of objects.

- Or:

$$x + y \geq 1 \qquad x, y \in \mathcal{B}$$

- Exclusive-or:

$$x + y = 1 \qquad x, y \in \mathcal{B}$$

Variables $x, y \in \mathcal{B}$: selection of objects.

- Or:

$$x + y \geq 1 \qquad x, y \in \mathcal{B}$$

- Exclusive-or:

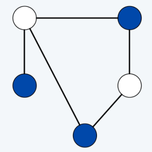
$$x + y = 1 \qquad x, y \in \mathcal{B}$$

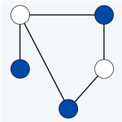
- Select n objects from m itens $x_1, \ldots, x_m \in \mathcal{B}$

$$\sum_{i}^{m} x_i \left\{ \begin{matrix} = \\ \geq \end{matrix} \right\} n$$

MAXIMUM INDEPENDENT SET

- Given a undirected graph $G = (V, E)$
- Objective: Find the larger set S of nodes such that no edge $e \in E$ has both endpoints in S





Variables:

$x_u \in \{0,1\}$: 1 if node u is in the solution and 0 otherwise

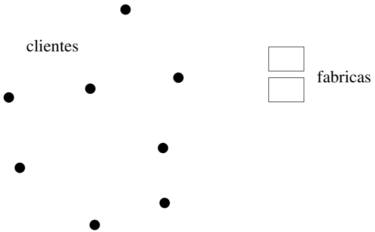
$$\begin{aligned} \textbf{max} \quad & \sum_{u \in V} x_u \\ \textbf{s.a} \quad & x_u + x_v \leq 1 \qquad \{u,v\} \in E \\ & x_u \in \{0,1\} \end{aligned}$$

- Implication: If x be selected, then y should be selected

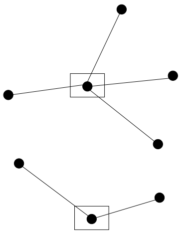
$$x \leq y \qquad x, y \in \mathcal{B}$$

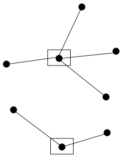
- Select one or more locations to install a factory each such that the total weighted distances from factories to customers (c_{ij}) is minimized. Moreover, a fix cost for each factory (f_i) installed is also summed up to the objective function.

An input:

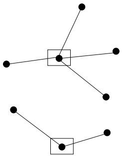


A possible solution:





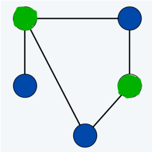
x_{ij}: 1 if location i is attended by customer j, and 0 otherwise.
y_i: 1 if there is a facility installed in location i, and 0 otherwise.

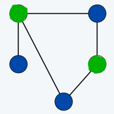


x_{ij}: 1 if location i is attended by customer j, and 0 otherwise.
y_i: 1 if there is a facility installed in location i, and 0 otherwise.

$$\begin{aligned}
\textbf{min} \quad & \sum_{j=1}^{n} f_j y_i + \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_{ij} & \\
\textbf{s.a} \quad & \sum_{j=1}^{n} x_{ij} = 1 & \forall i = 1...n \\
& x_{ij} \leq y_i & \forall i, j = 1...n \\
& x_{ij} \in \{0, 1\} & i, j = 1, ..., n \\
& y_j \in \{0, 1\} & j = 1, ..., n
\end{aligned}$$

- Given a undirected graph $G = (V, E)$
- Objective: Assign colors to all nodes such that no edge $e \in E$ has the same color.

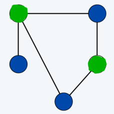




Variables:

$x_{uc} \in \{0,1\}$: 1 if node u is colored with color c, and 0 otherwise.

$y_c \in \{0,1\}$: 1 if color c is used, and 0 otherwise.



Variables:

$x_{uc} \in \{0, 1\}$: 1 if node u is colored with color c, and 0 otherwise.

$y_c \in \{0, 1\}$: 1 if color c is used, and 0 otherwise.

$$
\begin{aligned}
\textbf{min} \quad & \sum_{c=1}^{n} y_c & \\
\textbf{s.a} \quad & \sum_{c=1}^{n} x_{uc} = 1 & \forall u \in \mathcal{V} \\
& x_{uc} + x_{vc} \leq 1 & \forall (u, v) \in E, c \in V \\
& x_{uc} \leq y_c & \forall u, c \in \mathcal{V} \\
& x_{uc} \in \{0, 1\}, u_c \in \{0, 1\} & \forall u, c
\end{aligned}
$$

- Implication: If x_i and x_{i+1} be selected, then y should be selected

$$x_i + x_{i+1} \leq 1 + y \qquad x_i, x_{i+1}, y \in \mathcal{B}$$

Satisfiability problems: min-SAT, max-SAT, 3-SAT.

- Given n variables and m clausules, and a formula F in the conjunctive normal form.
- Objective: Find binary values for the variables such that the larger number of clausules be satisfied.

$$F = (x_1 \vee \bar{x_2} \vee \bar{x_4}) \wedge (x_2) \wedge (\bar{x_1} \vee x_3 \vee x_4)$$

Possible solution: $x_1 = x_2 = x_4 = 1$ and $x_3 = 0$

Input data:

n, m: number of variables and clausules, respectively

C_j: set of variables from clausule j

$\bar{C}_j$: set of negated variables from clausule j

Variables:

$x_i \in \{0, 1\}$: if the value of the variable is 0 or 1

$y_j \in \{0, 1\}$: if clausule j is satisfied or not

$$\max \sum_{j=1}^{m} y_j$$

$$s.t. \sum_{i \in C_j} x_i + \sum_{i \in \bar{C}_j} (1 - x_i) \geq y_j \qquad \forall j = 1, ..., m \qquad (1)$$

$$x_i \in \{0, 1\} \qquad \forall i = 1, ..., n \qquad (2)$$

$$y_j \in \{0, 1\} \qquad \forall j, ..., m \qquad (3)$$

- Implication in case x is an integer variable:

$$x \leq My \qquad x, y \in \mathcal{B}$$

In the cutting stock problem we are given an unlimited number of rolls of length c and m different types of items. At least b_i rolls of length $w_i, i = 1, ..., m$ have to be cut from the base rolls. The objective is to minimize the number of rolls used.

Bin Packing Problem: the case where $b_i = 1, i = 1, ...m$

Variables:

$x_{ij} \in Z^+$: denotes how many times item type i is cut in roll j

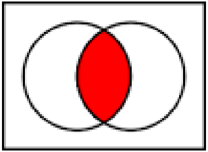
$y_j \in \{0,1\}$: denotes whether roll j is used for cutting or not

$$\textbf{min} \quad \sum_{j=1}^{U} y_j$$

$$s.t. \sum_{i=1}^{m} w_i x_{ij} \leq c y_j, \quad j = 1, ..., U$$

$$\sum_{j=1}^{U} x_{ij} \geq b_i, \quad i = 1, ..., m$$

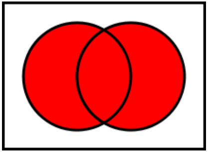
$$x_{ij} \in Z^+, y_j \in B, \quad i = 1, ..., m; j = 1, ..., U$$



Conjunction: $z = xy = x \wedge y$

$$z \leq (x+y)/2$$
$$z \geq x + y - 1$$

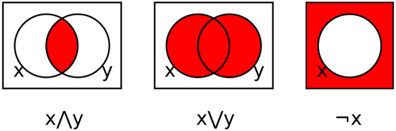
x	y	z
0	0	0
0	1	0
1	0	0
1	1	1



Disjunction: $z = x \vee y$

$$z \geq (x + y)/2$$
$$z \leq x + y$$

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

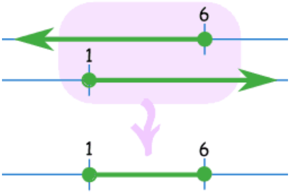


- Complement: $z = \neg x$

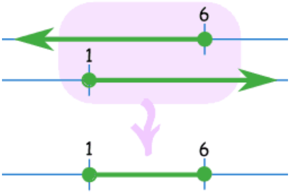
$$z = 1 - x$$

INTERVALS: $x \geq 1$ AND $x \leq 6$

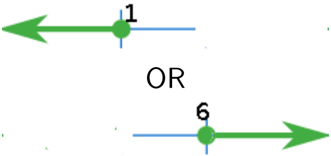
$$x \geq 1$$
$$x \leq 6$$



$$x \geq 1$$
$$x \leq 6$$



But how about: $x \leq 1$ or $x \geq 6$?



Choose M large enough such that for every constraint:

$$\sum_{i=1}^{n} a_i x_i \leq b + M$$

That is, we will be able to satisfy any $\leq$ constraint by adding M to the RHS.

Choose M large enough such that for every constraint:

$$\sum_{i=1}^{n} a_i x_i \leq b + M$$

That is, we will be able to satisfy any $\leq$ constraint by adding M to the RHS.
And we can satisfy any $\geq$ constraint by subtracting M from the RHS.

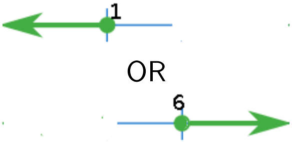
$$\sum_{i=1}^{n} a_i x_i \geq b - M$$

$x \leq 1$ OR $x \geq 6$

Choose $w \in \mathcal{B}$ such that:

- if $w = 1$ then $x \leq 1$
- if $w = 0$ then $x \geq 6$

$$x \leq 1 + M(1 - w)$$
$$x \geq 6 - Mw$$
$$w \in \mathcal{B}$$



Choose $w \in \mathcal{B}$ such that:

- if $w = 1$ then $A_1 x \leq b_1$
- if $w = 0$ then $A_2 x \leq b_2$

$$A_1 x \leq b_1 + M(1 - w)$$
$$A_2 x \leq b_2 + Mw$$
$$w \in \mathcal{B}$$

OR

Two restaurants are willing to produce empanadas and pizzas to sell during ELAVIO, but CELFI can only pay one restaurant. The revenue is $12 for each empanada and $8 for each pizza.
Restaurante A spends 7 minutes producing each empanada and 3 minutes per pizza, and has a total amount of 3h of production.
Restaurante B spends 4 minutes producing each empanada and 2 minutes per pizza, and has a total amount of 2h of production.
Which restaurante would obtain the larger revenue?
Variables: e and p number of empanadas and pizzas produced

$$\begin{aligned} \textbf{max} \quad & 12e + 8p \\ & 7e + 3p \leq 180 \\ & 4e + 2p \leq 120 \\ & e, p \in \mathcal{Z} \end{aligned}$$

Two restaurants are willing to produce empanadas and pizzas to sell during ELAVIO, but CELFI can only pay one restaurant. The revenue is $12 for each empanada and $8 for each pizza.

Restaurante A spends 7 minutes producing each empanada and 3 minutes per pizza, and has a total amount of 3h of production.

Restaurante B spends 4 minutes producing each empanada and 2 minutes per pizza, and has a total amount of 2h of production.

Which restaurante would obtain the larger revenue?

Variables: e and p number of empanadas and pizzas produced

$$\begin{aligned} \textbf{max} \quad & 12e + 8p \\ & 7e + 3p \leq 180 \\ & 4e + 2p \leq 120 \\ & e, p \in \mathcal{Z} \end{aligned}$$

This formulation imposes both restrictions, and then it does not model the problem correctly.

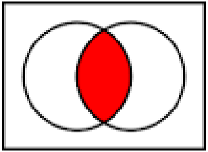
Two restaurants are willing to produce empanadas and pizzas to sell during ELAVIO, but CELFI can only pay one restaurant. The revenue is \$12 for each empanada and \$8 for each pizza. Restaurante A spends 7 minutes producing each empanada and 3 minutes per pizza, and has a total amount of 3h of production. Restaurante B spends 4 minutes producing each empanada and 2 minutes per pizza, and has a total amount of 2h of production. Which restaurante would obtain the larger revenue?

Variables: e and p number of empanadas and pizzas produced; and w is the Restaurant that will be chosen.

$$
\begin{aligned}
\textbf{max} \quad & 12e + 8p \\
& 7e + 3p \leq 180 + Mw \\
& 4e + 2p \leq 120 + M(1 - w) \\
& e, p \in \mathcal{Z} \\
& w \in \mathcal{B}
\end{aligned}
$$

$$
\begin{aligned}
A_1 x &\leq b_1 + M(1 - w_1) \\
A_2 x &\leq b_2 + M(1 - w_2) \\
\dots \\
A_n x &\leq b_n + M(1 - w_n) \\
\sum_{i=1}^{n} w_i &= k \quad i = \{1, ..n\} \\
w_i &\in \mathcal{B} \quad i = \{1, ..n\}
\end{aligned}
$$

$$A_1 x(1-w) \leq b_1(1-w)$$
$$A_2 x w \leq b_2 w$$
$$w \in \mathcal{B}$$



Conjunction: $z = xy = x \wedge y$

$$z \leq (x + y)/2$$
$$z \geq x + y - 1$$

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

- Minimize costs with a fix entry c

$$f(x) = \begin{cases} 0 & x = 0 \\ c + l(x) & 0 < x \leq \bar{x} \end{cases}$$

with $l(x)$ linear.

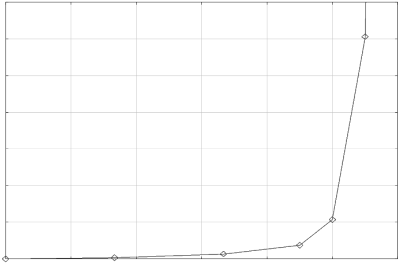
- Minimize costs with a fix entry c

$$f(x) = \begin{cases} 0 & x = 0 \\ c + l(x) & 0 < x \leq \bar{x} \end{cases}$$

with $l(x)$ linear.

- Linear model:

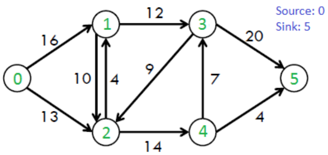
$$f(x) = cy + l(x)$$
$$x \leq \bar{x}y$$
$$x \in R, y \in \mathcal{B}$$



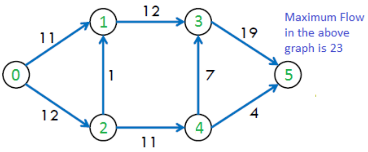
- Disagreg. Convex Combination - DCC (Sherali, 2001);
- Special Ordered Set of type 2 - SOS2 (Beale and Tomlin, 1970);

- Given a directed graph $G = (V, A)$
  - arcs with limited capacity $l : A \to \mathcal{Z}^+$,
- Which is the max flow?

An input:



A possible solution:



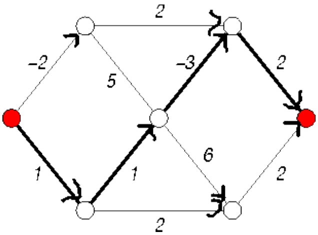
Variables:

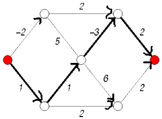
$x_a \in \mathcal{Z}^+$: flow on arc $a \in A$

$$
\begin{aligned}
\textbf{max} \quad & f \\
\textbf{s.a} \quad & f = \sum_{a \in N^+(s)} x_a \\
& \sum_{a \in N^+(v)} x_a - \sum_{a \in N^-(v)} x_a = 0 && \forall v \in V \backslash \{s, d\} \\
& 0 \leq x_a \leq l_a && \forall a \in A \\
& x_a \in \mathcal{Z} && \forall a \in A
\end{aligned}
$$

Flow conservation constraint

- Given a directed weighted graph $G = (V, A, w)$ with $w_a \in \mathcal{R}^+$, a source node s, and a destination node t
- Objective: Find the shortest path between s and t.





Variables:

$x_a \in \{0,1\}$: 1 if a is in the shortest path, and 0 otherwise

$$\begin{aligned}
\mathbf{min} \quad & \sum_{a \in A} c_a x_a \\
\mathbf{s.a} \quad & \sum_{a \in N^+(s)} x_a - \sum_{a \in N^-(s)} x_a = 1 \\
& \sum_{a \in N^+(t)} x_a - \sum_{a \in N^-(t)} x_a = -1 \\
& \sum_{a \in N^+(v)} x_a - \sum_{a \in N^-(v)} x_a = 0 && \forall v \in V \setminus \{s,t\} \\
& x_a \in \{0,1\} && \forall a \in A
\end{aligned}$$

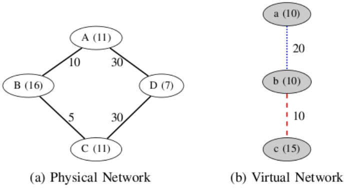
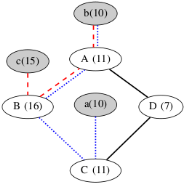


Fig. 1: An input instance for the VNEP.



- Leonardo Moura, Luciana S. Buriol, "A Column Generation Approach for the Virtual Network Embedding Problem", Conference on Combinatorial Optimization, 2014, Montevideo. Proceedings of the VIII ALIO/EURO Workshop on. Applied Combinatorial Optimization, 2014. p. 1-6.

Variables:
$x_{v,s} = 1$ iff the substrate node s hosts the virtual node v
$y_{v,w,s,j} = 1$ iff the physical link (s, j) hosts the virtual link (v, w)

$$\min \sum_{(s,j)\in E^S} \sum_{(v,w)\in E^V} y_{v,w,s,j} B_{v,w}$$

minimizes the amount of bandwidth used

$$s.t. \sum_{v\in V^V} x_{v,s} C_v \le C_s \qquad \forall s \in V^S \qquad (4)$$

$$\sum_{s\in V^S} x_{v,s} = 1 \qquad \forall v \in V^V \qquad (5)$$

$$\sum_{v\in V^V} x_{v,s} \le 1 \qquad \forall s \in V^S \qquad (6)$$

$$\sum_{j\in V^S} y_{v,w,s,j} - \sum_{j\in V^S} y_{v,w,j,s} = x_{v,s} - x_{w,s} \qquad \forall (v, w) \in E^V, s \in V^S \qquad (7)$$

$$\sum_{(v,w)\in E^V} y_{v,w,s,j} B_{v,w} \le B_{s,j} \qquad \forall (s, j) \in E^S \qquad (8)$$

$$x_{v,s} \in \{0, 1\} \qquad \forall v \in V^V, s \in V^S \qquad (9)$$

$$y_{k,l,m,n} \in \{0, 1\} \qquad \forall (k, l) \in E^V, (m, n) \in E^S \qquad (10)$$

Variables:
$x_{v,s} = 1$ iff the substrate node s hosts the virtual node v
$y_{v,w,s,j} = 1$ iff the physical link (s,j) hosts the virtual link (v,w)

$$\min \sum_{(s,j)\in E^S} \sum_{(v,w)\in E^V} y_{v,w,s,j} B_{v,w}$$

$$s.t. \sum_{v\in V^V} x_{v,s} C_v \le C_s \qquad \forall s \in V^S \qquad (4)$$

ensure substrate capacities are not surpassed

$$\sum_{s\in V^S} x_{v,s} = 1 \qquad \forall v \in V^V \qquad (5)$$

$$\sum_{v\in V^V} x_{v,s} \le 1 \qquad \forall s \in V^S \qquad (6)$$

$$\sum_{j\in V^S} y_{v,w,s,j} - \sum_{j\in V^S} y_{v,w,j,s} = x_{v,s} - x_{w,s} \qquad \forall (v,w) \in E^V, s \in V^S \qquad (7)$$

$$\sum_{(v,w)\in E^V} y_{v,w,s,j} B_{v,w} \le B_{s,j} \qquad \forall (s,j) \in E^S \qquad (8)$$

$$x_{v,s} \in \{0,1\} \qquad \forall v \in V^V, s \in V^S \qquad (9)$$

$$y_{k,l,m,n} \in \{0,1\} \qquad \forall (k,l) \in E^V, (m,n) \in E^S \qquad (10)$$

Variables:
$x_{v,s} = 1$ iff the substrate node s hosts the virtual node v
$y_{v,w,s,j} = 1$ iff the physical link (s,j) hosts the virtual link (v,w)

$$\min \sum_{(s,j)\in E^S} \sum_{(v,w)\in E^V} y_{v,w,s,j} B_{v,w}$$

$$s.t. \sum_{v\in V^V} x_{v,s} C_v \leq C_s \qquad \forall s \in V^S \qquad (4)$$

$$\sum_{s\in V^S} x_{v,s} = 1 \qquad \forall v \in V^V \qquad (5)$$

every virtual node is mapped to a substrate node

$$\sum_{v\in V^V} x_{v,s} \leq 1 \qquad \forall s \in V^S \qquad (6)$$

$$\sum_{j\in V^S} y_{v,w,s,j} - \sum_{j\in V^S} y_{v,w,j,s} = x_{v,s} - x_{w,s} \qquad \forall (v,w) \in E^V, s \in V^S \qquad (7)$$

$$\sum_{(v,w)\in E^V} y_{v,w,s,j} B_{v,w} \leq B_{s,j} \qquad \forall (s,j) \in E^S \qquad (8)$$

$$x_{v,s} \in \{0,1\} \qquad \forall v \in V^V, s \in V^S \qquad (9)$$

$$y_{k,l,m,n} \in \{0,1\} \qquad \forall (k,l) \in E^V, (m,n) \in E^S \qquad (10)$$

Variables:
$x_{v,s} = 1$ iff the substrate node s hosts the virtual node v
$y_{v,w,s,j} = 1$ iff the physical link (s,j) hosts the virtual link (v,w)

$$\min \sum_{(s,j)\in E^S} \sum_{(v,w)\in E^V} y_{v,w,s,j} B_{v,w}$$

$$s.t. \sum_{v\in V^V} x_{v,s} C_v \le C_s \qquad \forall s \in V^S \qquad (4)$$

$$\sum_{s\in V^S} x_{v,s} = 1 \qquad \forall v \in V^V \qquad (5)$$

every substrate node hosts at most one virtual node

$$\sum_{v\in V^V} x_{v,s} \le 1 \qquad \forall s \in V^S \qquad (6)$$

$$\sum_{j\in V^S} y_{v,w,s,j} - \sum_{j\in V^S} y_{v,w,j,s} = x_{v,s} - x_{w,s} \qquad \forall (v,w) \in E^V, s \in V^S \qquad (7)$$

$$\sum_{(v,w)\in E^V} y_{v,w,s,j} B_{v,w} \le B_{s,j} \qquad \forall (s,j) \in E^S \qquad (8)$$

$$x_{v,s} \in \{0,1\} \qquad \forall v \in V^V, s \in V^S \qquad (9)$$

$$y_{k,l,m,n} \in \{0,1\} \qquad \forall (k,l) \in E^V, (m,n) \in E^S \qquad (10)$$

Variables:
$x_{v,s} = 1$ iff the substrate node s hosts the virtual node v
$y_{v,w,s,j} = 1$ iff the physical link (s,j) hosts the virtual link (v,w)

$$\min \sum_{(s,j)\in E^S} \sum_{(v,w)\in E^V} y_{v,w,s,j} B_{v,w}$$

$$s.t. \sum_{v\in V^V} x_{v,s} C_v \leq C_s \quad \forall s \in V^S \quad (4)$$

$$\sum_{s\in V^S} x_{v,s} = 1 \quad \forall v \in V^V \quad (5)$$

every virtual link is mapped to a path into the substrate graph

$$\sum_{v\in V^V} x_{v,s} \leq 1 \quad \forall s \in V^S \quad (6)$$

$$\sum_{j\in V^S} y_{v,w,s,j} - \sum_{j\in V^S} y_{v,w,j,s} = x_{v,s} - x_{w,s} \quad \forall (v,w) \in E^V, s \in V^S \quad (7)$$

$$\sum_{(v,w)\in E^V} y_{v,w,s,j} B_{v,w} \leq B_{s,j} \quad \forall (s,j) \in E^S \quad (8)$$

$$x_{v,s} \in \{0,1\} \quad \forall v \in V^V, s \in V^S \quad (9)$$

$$y_{k,l,m,n} \in \{0,1\} \quad \forall (k,l) \in E^V, (m,n) \in E^S \quad (10)$$

Variables:
$x_{v,s} = 1$ iff the substrate node s hosts the virtual node v
$y_{v,w,s,j} = 1$ iff the physical link (s,j) hosts the virtual link (v,w)

$$\min \sum_{(s,j)\in E^S} \sum_{(v,w)\in E^V} y_{v,w,s,j} B_{v,w}$$

$$s.t. \sum_{v\in V^V} x_{v,s} C_v \leq C_s \qquad \forall s \in V^S \qquad (4)$$

$$\sum_{s\in V^S} x_{v,s} = 1 \qquad \forall v \in V^V \qquad (5)$$

ensures that the bandwidth capacities of the physical edges are not violated

$$\sum_{v\in V^V} x_{v,s} \leq 1 \qquad \forall s \in V^S \qquad (6)$$

$$\sum_{j\in V^S} y_{v,w,s,j} - \sum_{j\in V^S} y_{v,w,j,s} = x_{v,s} - x_{w,s} \qquad \forall (v,w) \in E^V, s \in V^S \qquad (7)$$

$$\sum_{(v,w)\in E^V} y_{v,w,s,j} B_{v,w} \leq B_{s,j} \qquad \forall (s,j) \in E^S \qquad (8)$$

$$x_{v,s} \in \{0,1\} \qquad \forall v \in V^V, s \in V^S \qquad (9)$$

$$y_{k,l,m,n} \in \{0,1\} \qquad \forall (k,l) \in E^V, (m,n) \in E^S \qquad (10)$$

- Because your solution approach needs a math formulation.

- Because your solution approach needs a math formulation.
- To formalize a clear definition of the problem;

- Because your solution approach needs a math formulation.
- To formalize a clear definition of the problem;

$$
\begin{aligned}
\textbf{Min}\quad & \left[\sum_{d\in D}\sum_{s\in S}\sum_{k\in K} a^1_{dsk}\omega^1\right] + \\
& \left[\sum_{n\in N}\sum_{d\in D}\sum_{t\in T}\sum_{i=2,4} b^i_{ndt}\omega^i\right] + \\
& \left[\sum_{n\in N}\sum_{d\in D}\sum_{i=3,5} c^i_{nd}\omega^i\right] + \\
& \left[\sum_{n\in N}\sum_{d\in D}\sum_{s\in S}\sum_{t\in T} e^6_{ndst}\omega^6\right] + \\
& \left[\sum_{n\in N}\sum_{d\in D}\sum_{s\in S} f^7_{nds}\omega^7\right] + \\
& \left[\sum_{n\in N}\sum_{d\in D}\sum_{s\in S} g^8_{nds}\omega^8\right] + \\
& \left[\sum_{n\in N}\sum_{w\in W} h^9_{nw}\omega^9\right] + \\
& \left[\sum_{n\in N}\sum_{i=10..12} j^i_n\omega^i\right]
\end{aligned}
$$

Subject to

$$\sum_{s\in S}\sum_{k\in K} x_{ndsk} \le 1 \qquad \forall n\in N, d\in D \tag{2}$$

$$\sum_{n\in N} x_{ndsk} \ge r_{dsk} \qquad \forall d\in D, s\in S, k\in K \tag{3}$$

$$\sum_{s\in S}\sum_{k\in K}(x_{ndsk}p_{ask}) + (x_{nd+1sk}p_{ask}) \le 1 \qquad \forall a\in P, n\in N, d\in D-1, d+1\in D \tag{4}$$

$$(k - N_{nk})x_{ndsk} = 0 \qquad \forall n\in N, d\in D, s\in S, k\in K \tag{5}$$

$$\sum_{n\in N} x_{ndsk} + a^1_{ndsk} \ge \alpha^1_{ndsk} \qquad \forall d\in D, s\in S, k\in K \tag{6}$$

$$S1_{ndt} + b^2_{ndt} \ge \beta^2_n \qquad \forall n\in N, d\in D, t\in T \tag{7}$$

$$\sum_{d2=d}^{\beta^3_n+1}\sum_{s\in S}\sum_{k\in K} x_{nd2sk} - c^3_{nd} \le \beta^3_n \qquad \forall n\in N, d\in D-\beta^3_n-1 \tag{8}$$

$$S2_{ndt} + b^4_{ndt} \ge \beta^4_n \qquad \forall n\in N, d\in D, t\in T \tag{9}$$

$$\sum_{d2=d}^{\beta^5_n+1}\sum_{s\in S}\sum_{k\in K} 1 - x_{nd2sk} - c^5_{nd} \le \beta^5_n \qquad \forall n\in N, d\in D-\beta^5_n-1 \tag{10}$$

$$S3_{ndst} + e^6_{ndst} \ge \gamma^6_s \qquad \forall n\in N, d\in D, s\in S, t\in T \tag{11}$$

$$\sum_{d2=d}^{\gamma^7_s+1}\sum_{k\in K} x_{nd2sk} - f^7_{nds} \le \gamma^7_s \qquad \forall n\in N, d\in D-\gamma^7_s-1, s\in S \tag{12}$$

$$u_{nds} - \left[\sum_{k\in K}(x_{ndsk}g^8_{nds})\right] = 0 \qquad \forall n\in N, d\in D, s\in S \tag{13}$$

$$\sum_{s\in S}\sum_{k\in K} x_{ndsk} + x_{nd+1sk} + h^9_{nw} \le 2By_{nw} \qquad \forall n\in N, w\in W \tag{14}$$

$$\sum_{d\in D}\sum_{s\in S}\sum_{k\in K} x_{ndsk} + j^{10}_n \ge \beta^{10}_n \qquad \forall n\in N \tag{15}$$

$$\sum_{d\in D}\sum_{s\in S}\sum_{k\in K} x_{ndsk} - j^{11}_n \le \beta^{11}_n \qquad \forall n\in N \tag{16}$$

$$\sum_{s\in S}\sum_{k\in K} x_{ndsk} + x_{nd+1sk} \le 2y_{nw} \qquad \forall n\in N, d\in W, w\in W \tag{17}$$

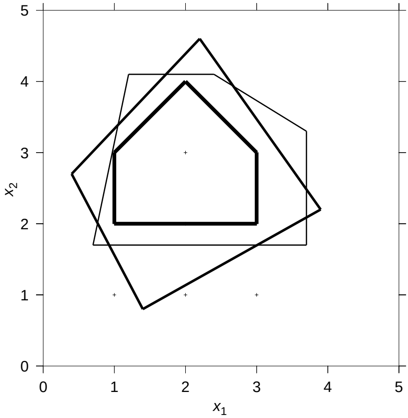
$$\sum_{w\in W} y_{nw} - j^{12}_n \le \beta^{12}_n \qquad \forall n\in N \tag{18}$$

WHY TO FORMULATE IP PROBLEMS?

- Because your solution approach needs a math formulation.
- To formalize a clear definition of the problem;
- To provide a comparison against CPLEX results (or from other solver);

- Because your solution approach needs a math formulation.
- To formalize a clear definition of the problem;
- To provide a comparison against CPLEX results (or from other solver);
- To play with restrictions when defining a problem;

- Because your solution approach needs a math formulation.
- To formalize a clear definition of the problem;
- To provide a comparison against CPLEX results (or from other solver);
- To play with restrictions when defining a problem;
- To guide decisions on further solution approaches for the problem (maybe a solver solution is enough);
- To explore bounds and properties of different formulations.



See for example *LP models for bin packing and cutting stock problem* by José Valerio de Carvalho, European Journal of Operational Research 141(2):253-273, 2002.

Subtour elimination.

- Given a directed weighed graph $G = (V, A, w)$ with $w_a \in \mathcal{R}^+$
- Objective: Find the shortest directed Hamiltonian cycle.

$$
\begin{aligned}
\textbf{min} \quad & c_{ij}x_{ij} \\
\textbf{s.a} \quad & \sum_{j=1}^{n} x_{ij} = 1 & i \in \mathcal{V} \\
& \sum_{i=1}^{n} x_{ij} = 1 & j \in \mathcal{V} \\
& \sum_{i \in S, j \in S} x_{ij} \leq |S| - 1, \qquad S \in V : 2 \leq |S| \leq (n-1) \\
& x_{ij} \in \{0, 1\}, & \forall i, j \in N.
\end{aligned}
$$

subtour elimination

Variables:
$x_{ij} \in \{0,1\}$: 1 if (i,j) is in the tour, and 0 otherwise
$u_i \in \mathcal{R}^+$: the order the node is visited

$$\begin{aligned}
\textbf{min} \quad & \sum_{i,j} c_{ij} x_{ij} & \\
\textbf{s.a} \quad & \sum_{j=1}^{n} x_{ij} = 1 & i \in \mathcal{V} \\
& \sum_{i=1}^{n} x_{ij} = 1 & j \in \mathcal{V} \\
& u_i - u_j + n x_{ij} \leq n-1, & \forall i,j \in \mathcal{V} \backslash 1, i \neq j \\
& x_{ij} \in \{0,1\}, u_i \in \mathcal{R}^+ & \forall i,j
\end{aligned}$$

subtour elimination

PS: This formulation is weaker than the standard one.

- There are n clients to visit, each with demand d_i, K vehicles with capacity C with routes leaving from node 1, and the costs c_{ij} between each pair (i,j)
- Find the K routes with minimum total cost, attending all client demands without surpassing the vehicle capacities
- More info about VRP find in `http://neo.lcc.uma.es/vrp/`.

Variables:
$x_{ij} \in \{0,1\}$: 1 if (i,j) is in a route, and 0 otherwise
$u_i \in \mathcal{R}^+$: load of vehicle after visiting node i

$$\min \sum_{i,j} c_{ij} x_{ij}$$

$$\text{s.a} \sum_{j=1}^{n} x_{ij} = 1 \qquad i \in \mathcal{V} \backslash \{1\}$$ //each customer has and incoming arc

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad j \in \mathcal{V} \backslash \{1\}$$ //each customer has and incoming arc

$$\sum_{i=1}^{n} x_{i1} = K; \sum_{i=1}^{n} x_{1j} = K$$ //there are K arcs incoming and outgoing the deposit

$$\sum_{i=1}^{n} x_{ii} = 0$$ //avoid self-loops

$$u_j - u_i + C(1 - x_{ij}) \geq d_j, \qquad \forall i,j \in \mathcal{V} \backslash \{1\}, i \neq j$$ //avoid subcicles

$$u_i \leq C \qquad i \in \mathcal{V} \backslash \{1\}$$ //the vehicle capacity cannot be surpassed

$$x_{ij} \in \{0,1\}, u_i \in \mathcal{R}^+ \qquad \forall i,j$$

- Given n games, each with a starting time and a finishing time; a start-end point p, and a time distance between each pair of points
- Objective: Find a tour that starts and ends at node p, and attends the larger number of games.

Hard Constraints

H1 : The workload defined in each event must be satisfied.

H2 : A teacher cannot be scheduled to more than one lesson in a given period.

H3 : Lessons cannot be taught to the same class in the same period.

H4 : A teacher cannot be scheduled to a period in which he/she is unavailable.

H5 : The maximum number of daily lessons of each event must be respected.

H6 : Two lessons from the same event must be consecutive when scheduled for the same day, in case it is required by the event.

Soft Constraints

S1 Avoid teachers' idle periods.

S2 Minimize the number of *working days* for teachers. In this context, working day means a day that the teacher has at least one lesson assigned to him/her.

S3 Provide the number of double lessons requested by each event.

- Árton Dorneles, Olinto Araújo, Luciana S. Buriol. "A fix-and-optimize heuristic for the high school timetabling problem", Computers & Operations Research, v. 52, p. 29-38, 2014.

Symbol	Definition
Sets	
$d \in D$	days of week.
$p \in P$	periods of day.
$t \in T$	set of teachers.
$c \in C$	set of classes.
$e \in E$	set of events.
E_t	set of events assigned to teacher t.
E_c	set of events assigned to class c.
Parameters	
R_e	workload of event e.
L_e	maximum daily number of lessons of event e.
Variables	
x_{edp}	binary variable that indicates whether event e is scheduled to timeslot (d, p).
y_{td}	has value 1 if at least one lesson is assigned to teacher t on day d, and zero otherwise.

$$\textbf{Min} \sum_{t \in T} \sum_{d \in D} y_{td} \tag{11}$$

H1: The workload defined in each event must be satisfied

$$\sum_{d \in D, p \in P} x_{edp} = R_e \qquad \forall e \quad //\text{H1} \tag{12}$$

$$\sum_{e \in E_c} x_{edp} \leq 1 \qquad \forall c, d, p \quad //\text{H3} \tag{13}$$

$$\sum_{p \in P} x_{edp} \leq L_e \qquad \forall e, d \quad //\text{H5} \tag{14}$$

$$\sum_{e \in E_t} x_{edp} \leq y_{td} \qquad \forall t, d, p \quad \text{S2, H4} \tag{15}$$

$$x_{edp} \in \{0, 1\} \qquad \forall e, d, p \tag{16}$$

$$y_{td} \in \{0, 1\} \qquad \forall t, d \tag{17}$$

$$\textbf{Min} \sum_{t \in T} \sum_{d \in D} y_{td} \tag{11}$$

H3: Lessons cannot be taught to the same class in the same period.

$$\sum_{d \in D, p \in P} x_{edp} = R_e \qquad \forall e \quad //\text{H1} \tag{12}$$

$$\sum_{e \in E_c} x_{edp} \leq 1 \qquad \forall c, d, p \quad //\text{H3} \tag{13}$$

$$\sum_{p \in P} x_{edp} \leq L_e \qquad \forall e, d \quad //\text{H5} \tag{14}$$

$$\sum_{e \in E_t} x_{edp} \leq y_{td} \qquad \forall t, d, p \quad \text{S2, H4} \tag{15}$$

$$x_{edp} \in \{0, 1\} \qquad \forall e, d, p \tag{16}$$

$$y_{td} \in \{0, 1\} \qquad \forall t, d \tag{17}$$

$$\textbf{Min} \sum_{t \in T} \sum_{d \in D} y_{td} \tag{11}$$

H5: The maximum number of daily lessons of each event must be respected

$$\sum_{d \in D, p \in P} x_{edp} = R_e \qquad \forall e \quad //\text{H1} \tag{12}$$

$$\sum_{e \in E_c} x_{edp} \leq 1 \qquad \forall c, d, p \quad //\text{H3} \tag{13}$$

$$\sum_{p \in P} x_{edp} \leq L_e \qquad \forall e, d \quad //\text{H5} \tag{14}$$

$$\sum_{e \in E_t} x_{edp} \leq y_{td} \qquad \forall t, d, p \quad \text{S2, H4} \tag{15}$$

$$x_{edp} \in \{0, 1\} \qquad \forall e, d, p \tag{16}$$

$$y_{td} \in \{0, 1\} \qquad \forall t, d \tag{17}$$

$$\textbf{Min} \sum_{t \in T} \sum_{d \in D} y_{td} \tag{11}$$

H2: A teacher cannot be scheduled to more than one lesson in a given period.
S2: Accounts the number of working days for teachers.

$$\sum_{d \in D, p \in P} x_{edp} = R_e \qquad \forall e \quad //\text{H1} \tag{12}$$

$$\sum_{e \in E_c} x_{edp} \leq 1 \qquad \forall c, d, p \quad //\text{H3} \tag{13}$$

$$\sum_{p \in P} x_{edp} \leq L_e \qquad \forall e, d \quad //\text{H5} \tag{14}$$

$$\sum_{e \in E_t} x_{edp} \leq y_{td} \qquad \forall t, d, p \quad \text{S2, H4} \tag{15}$$

$$x_{edp} \in \{0, 1\} \qquad \forall e, d, p \tag{16}$$

$$y_{td} \in \{0, 1\} \qquad \forall t, d \tag{17}$$

$$\textbf{Min} \sum_{t \in T} \sum_{d \in D} y_{td} \tag{11}$$

Minimizes the number of working days for teachers.

$$\sum_{d \in D, p \in P} x_{edp} = R_e \qquad \forall e \quad //\text{H1} \tag{12}$$

$$\sum_{e \in E_c} x_{edp} \leq 1 \qquad \forall c, d, p \quad //\text{H3} \tag{13}$$

$$\sum_{p \in P} x_{edp} \leq L_e \qquad \forall e, d \quad //\text{H5} \tag{14}$$

$$\sum_{e \in E_t} x_{edp} \leq y_{td} \qquad \forall t, d, p \quad \text{S2, H4} \tag{15}$$

$$x_{edp} \in \{0, 1\} \qquad \forall e, d, p \tag{16}$$

$$y_{td} \in \{0, 1\} \qquad \forall t, d \tag{17}$$

Symbol	Definition
Sets	
$v \in V$	set of all nodes.
$a \in A_t$	set of all arcs of the commodity t ($A_t \subset A$).
$a \in A_{tcdp}$	set of lesson arcs of the commodity t on class c, day d, and period p.
$a \in A^{-}_{tv}$	set of all arcs incoming node v for the commodity t.
$a \in A^{+}_{tv}$	set of all arcs outgoing node v for the commodity t.
$a \in Y_t$	set of all working day arcs of teacher t.
Parameters	
b_v	assumes 1 when v is the source, -1 when v is the sink, otherwise 0.
$H_{tc} \in \mathbb{N}$	number of lessons that teacher t must taught to class c.
$L_{tc} \in \{1, 2\}$	maximum daily number of lessons that teacher t can taught to class c.
$S_{ta} \in \{1, 2\}$	size of arc a for the commodity t
$\gamma = 9$	cost for each working day.
Variables	
$x_{ta} \in \{0, 1\}$	indicates whether commodity t uses arc a.

- Árton Dorneles, Olinto de Araújo, Luciana S. Buriol, “A Column Generation Approach to High School Timetabling Modeled as a Multicommodity Flow Problem". European Journal of Operational Research, p. 1-28, 2017.

$$\textbf{Minimize} \sum_{a \in Y_t} \gamma x_{ta} \tag{18}$$

Subject to

$$\sum_{a \in A^+_{tv}} x_{ta} - \sum_{a \in A^-_{tv}} x_{ta} = b_v \qquad \forall t \in T, v \in V \quad //\text{H2} \tag{19}$$

$$\sum_{t \in T} \sum_{a \in A_{tcdp}} x_{ta} \leq 1 \qquad \forall c \in C, d \in D, p \in P \quad //\text{H3} \tag{20}$$

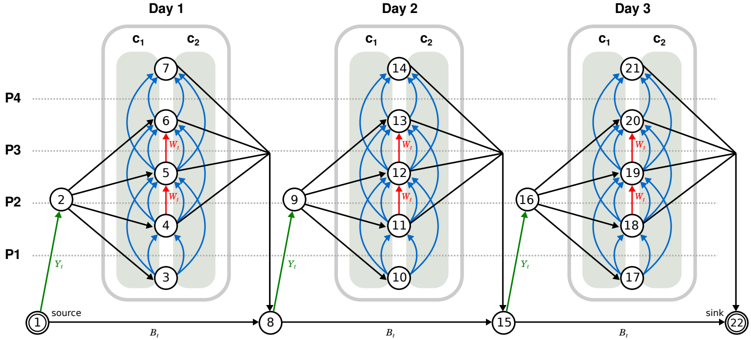
$$\sum_{a \in \bigcup_{d \in D, p \in P} A_{tcdp}} S_{ta} x_{ta} = H_{tc} \qquad \forall t \in T, c \in C \quad //\text{H1} \tag{21}$$

$$\sum_{a \in \bigcup_{p \in P} A_{tcdp}} S_{ta} x_{ta} \leq L_{tc} \qquad \forall t \in T, c \in C, d \in D \quad //\text{H5} \tag{22}$$

$$x_{ta} \in \{0, 1\} \qquad \forall t \in T, a \in A_t \tag{23}$$

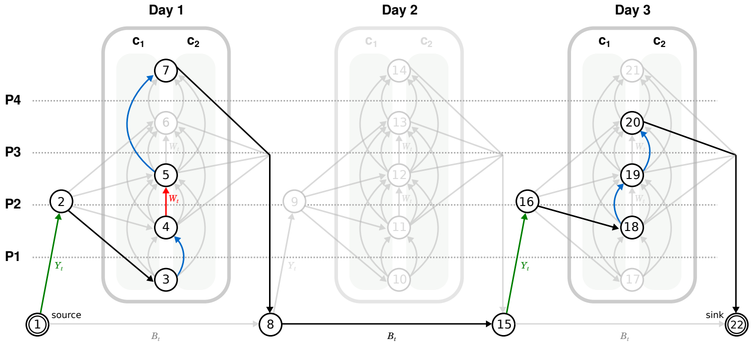
$$\tag{24}$$

HST: FLOW FORMULATION (AN INSTANCE)



Example of a network graph in a toy instance composed by three days, four periods by day (P1, P2, P3, P4), and two classes (c_1, c_2). Each day of the week is represented by a rounded rectangle where lesson arcs and idle period arcs are located. Inside each, lesson arcs appear in two groups represented by a shaded rectangle, where each group represents the lesson arcs for classes c_1 and c_2.

HST: FLOW FORMULATION (A SOLUTION)



Example of a feasible schedule for a teacher t represented by a path in the network. In this example, a teacher works only on days 1 and 3. On day 1, she/he teaches a single lesson for the class c_2 in the period P1, becomes idle in the period P2, and then gives a double lesson starting in the period P3 for the class c_1. On day 3, she/he teaches a single lesson for class c_1 in the period P2 and another one for class c_2 in the period P3.

```
/*******************************************************************
set VERTICES;
set ARCS within (VERTICES cross VERTICES);

param capacity{ARCS};
param weight{ARCS};
param demand{VERTICES} default 0;

var x{(i,j) in ARCS} >= 0;

minimize cost: sum{(i,j) in ARCS} x[i,j]*weight[i,j];

s.t. CAP {(i,j) in ARCS}: x[i,j]<=capacity[i,j];

s.t. BALANCE{i in VERTICES}:
          sum{j in VERTICES: (i,j) in ARCS} x[i,j]
        - sum{j in VERTICES: (j,i) in ARCS} x[j,i]
        = demand[i];
end;
/*******************************************************************
```

Operational Research = Operations Research

Operational Research is in British usage, while that Operations Research is in American usage.

Before the II World War OR did not exist as a research area. However, some of the basic OR techniques were developed before the IIWW: inventory control, queuing theory, and statistical, quality control, among others.

For example, Charles Babbage produced results for sorting mail and for defining the cost of transportation.

During the II World War scientists were contracted to **research** how to better perform military **operations**

During the II World War scientists were contracted to **research** how to better perform military **operations**

Operational Research

During the II World War scientists were contracted to **research** how to better perform military **operations**

Operational Research

As a formal discipline, OR was originated in the efforts of military planners during the II World War.

- About 1000 man and woman were were engaged in operational research in UK
- About 200 of them were scientists working in Operations Research for the British Army
- The Army Operational Research Group (AORG) was divided into 21 Operations Research Sections (ORS): BC-ORS (Bomber Command), CC-ORS (Coastal Command), etc.

The Army Operational Research Group (AORG) was responsible for strategic decisions:

- the color of the plains (white ones could arrive 20% closer than the black ones)
- the trigger depth of aerial-delivered charges (changing from 100 feet to 25 feet the percentage of success on sunking submarines changed from 1% to 7%)
- size of the convoys (large ones were more defensible)
- comparing the number of flying hours of aircrafts to the number of U-boat sightings in a given area, it was possible to redistribute aircraft to more productive patrol areas

The Army Operational Research Group (AORG) was responsible for strategic decisions:

- the color of the plains (white ones could arrive 20% closer than the black ones)
- the trigger depth of aerial-delivered charges (changing from 100 feet to 25 feet the percentage of success on sunking submarines changed from 1% to 7%)
- size of the convoys (large ones were more defensible)
- comparing the number of flying hours of aircrafts to the number of U-boat sightings in a given area, it was possible to redistribute aircraft to more productive patrol areas

The work developed by the AORG was very important for tactic and strategic decisions during the war.

- After the war, researchers kept on working in the area.
- OR was applied to many different problems in business, industry and society.
- The results attracted new researchers to the area.

OR: 1945-1960

- After the war, researchers kept on working in the area.
- OR was applied to many different problems in business, industry and society.
- The results attracted new researchers to the area.
- Several significant contributions

Several significant contributions:

- 1947: George Dantzig created the Simplex algorithm,
- 1948: Duality (conjecture by John von Neumann and proved by Albert Tucker in 1948)
- 1956: Alan Hoffman and Joseph Kruskal: importance of unimodularity to find integer solutions
- 1958: Cutting Planes algorithm by Ralph Gomory
- 1960: (Branch-and-Bound) A.H. Land and A.G. Doig, "An automatic method for solving discrete programming problems", Econometrica 28 (1960) 497-520.

Several significant contributions:

- 1946-1950: the Monte Carlo method was developed (John von Neumann and Stanislaw Ulam)
- 1950: The Nash Equilibrium (Ph.D. of John Nash)
- 1951: Karush-Kuhn-Tucker (KKT)
- 1953: Metropolis Algorithm
- 1953: Dynamic programming (Richard Bellman)
- 1956: Dijkstra algorithm for calculating shortest paths in graphs
- 1956: Ford-Fulkerson algorithm O($E.maxflow$)

Creation of OR Societies and Journals. Operational Research Societies:
1957: The first International Federation of Operational Research Societies (IFORS), in Oxford/England
1959: IFORS: International Federation of Operational Research Societies

Creation of OR Societies and Journals. Operational Research Societies:
1957: The first International Federation of Operational Research Societies (IFORS), in Oxford/England
1959: IFORS: International Federation of Operational Research Societies

- 1959: France, UK, USA

Creation of OR Societies and Journals. Operational Research Societies:
1957: The first International Federation of Operational Research Societies (IFORS), in Oxford/England
1959: IFORS: International Federation of Operational Research Societies

- 1959: France, UK, USA
- 1960: Australia, Belgium, Canada, India, The Netherlands, Norway, Sweden
- 1961: Japan
- 1962: Argentina, Germany, Italy
- 1963: Denmark, Spain, Switzerland
- 1966: Greece, Ireland, Mexico
- 1969: Brazil, Israel

- 1970: New Zealand
- 1972: Korea
- 1973: South Africa
- 1975: Chile, Finland
- 1976: Egypt
- 1977: Turkey
- 1978: Singapore
- 1979: Austria
- 1982: China, Portugal
- 1983: Hong Kong, Yugoslavia
- 1986: Iceland
- 1988: Malaysia

- 1990: Philippines
- 1992: Hungary
- 1993: Bulgaria
- 1994: Croatia, Czech Republic, Slovakia
- 1998: Belarus
- 2002: Bangladesh, Colombia, Lithuania
- 2007: Slovenia

- 1976: EURO (Association of European OR Societies) was constituted, currently with 31 countries

IFORS: REGIONAL GROUPINGS

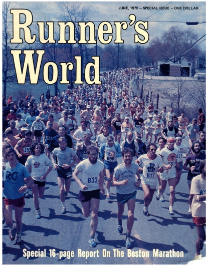
- 1976: EURO (Association of European OR Societies)
- 1982: ALIO (Association of Latin American OR Societies). Argentina, Brazil, Chile, Colombia, Cuba, Equador, Mexico, Peru, Portugal, Spain, Uruguay.

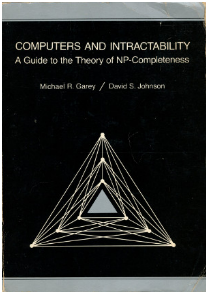
- 1976: EURO (Association of European OR Societies)
- 1982: ALIO (Association of Latin American OR Societies)
- 1985: APORS: Association of Asian-Pacific OR Societies, currently with 10 countries

- 1976: EURO (Association of European OR Societies)
- 1982: ALIO (Association of Latin American OR Societies)
- 1985: APORS: Association of Asian-Pacific OR Societies
- 1987: NORAM: Association of North American OR Societies

- 1960: Dantzig-Wolf decomposition
- 1961: Gilmore P. C., R. E. Gomory, “A linear programming approach to the cutting-stock problem”. Operations Research 9: 849-859
- 1962: Gale-Shapley algorithm for solve the Stable Matching Problem
- 1963: First OR book - “Linear programming and extensions”, by George Dantzig
- 1969: The four color problem theorem, a method for solving the problem using computers by Heinrich Heesch

- Notion of **problem complexity**: importance of polynomial algorithms for combinatorial algorithms reached a broader audience
    - 1971: the Cook-Levin theorem
    - Cook-Karp: 21 NPC problems
    - 1979 “Computers and Intractability”, by Garey and Johnson


JUNE, 1975 – SPECIAL ISSUE – ONE DOLLAR
Runner's World
Special 16-page Report On The Boston Marathon


COMPUTERS AND INTRACTABILITY
A Guide to the Theory of NP-Completeness
Michael R. Garey / David S. Johnson

- Notion of **problem complexity**: importance of polynomial algorithms for combinatorial algorithms reached a broader audience
  - 1971: the Cook-Levin theorem
  - Cook-Karp: 21 NPC problems
  - 1979 “Computers and Intractability”, by Garey and Johnson
- In 1977 the **microprocessors** were introduced. From mid 60’s to mid 70’s computers were generally large, costly systems owned by large corporations, universities, government agencies, and similar-sized institutions.

- Zionts, S.; Wallenius, J. (1976). “An Interactive Programming Method for Solving the Multiple Criteria Problem”. Management Science 22 (6): 652
- First solvers: MINOS - Modular In-Core Nonlinear Optimization System (1976), XMP (1979)
- 1979: The linear-programming problem was first shown to be solvable in polynomial time by Leonid Khachiyan

- 1984: Narendra Karmarkar introduced a new interior-point method for solving linear-programming problems

OR: 1980-1990

- 1984: Narendra Karmarkar introduced a new interior-point method for solving linear-programming problems
- Metaheuristics were able to provide near-optimal solutions to large problems
  - 1975: Genetic algorithms become through the work of John Holland in the early 1970s
  - 1983: Simulated annealing by Kirkpatrick
  - 1986: Tabu search by Glover
  - 1989: GRASP by Feo and Resend

  - Different approaches were proposed
  - Applied to different problems
  - Different set of parameters

- Branch and cut: Cornuejols and co-workers showed how to combine Gomory cuts with branch-and-bound overcoming numerical instabilities
- Branch and price: column generation combined with branch-and-bound (Nemhauser and Park (1991) and Vanderbeck (1994))
- Problem decompositions

- CPLEX performance
  - 1988: CPLEX 1.0
  - 1992: CPLEX 2.0 with branch-and-bound and limited cuts
  - 1998: CPLEX 6.0 added by heuristics and faster dual simplex
  - 1999: CPLEX 6.6 with 7 types of cutting planes and several node heuristics
  - 2010: CPLEX 12.2 full-version is available free-of-charge to academics.
  - 2014: CPLEX 12.6

- CPLEX performance
  - 1988: CPLEX 1.0
  - 1992: CPLEX 2.0 with branch-and-bound and limited cuts
  - 1998: CPLEX 6.0 added by heuristics and faster dual simplex
  - 1999: CPLEX 6.6 with 7 types of cutting planes and several node heuristics
  - 2010: CPLEX 12.2 full-version is available free-of-charge to academics.
  - 2014: CPLEX 12.6
- Matheuristics: interoperation of metaheuristics and mathematical programming techniques
- 2002: Agrawal, Manindra; Kayal, Neeraj; Saxena, Nitin "PRIMES is in P". Annals of Mathematics 160 (2): 781-793(2004). 2006 Gödel Prize and 2006 Fulkerson Prize.
- 2012: "Max flows in O(nm) time, or better", James Orlin.

FINAL CONSIDERATIONS

- To formulate mathematically a problem is an art!
- The number of variables and restrictions matters, but it is not the only factor that determines the performance of the solver.
- Explore different mathematical formulations for the problem you are solving.
- You are lucky for having so many solvers available...

Thanks for your attention!

buriol@inf.ufrgs.br