Computational Optimization

Linear Programming: Refresher

Pascal Van Hentenryck

Volsay Problem

- The Volsay company produces
  - ammoniac gas (NH_3)
  - ammonium chloride (NH_4Cl)
- Volsay has in stock
  - 50 units of nitrogen (N)
  - 180 units of hydrogen (H)
  - 40 unit of chlorine (Cl)
- Volsay make a profit of
  - 40 dollars for each unit of ammoniac gas
  - 50 dollars for each unit of ammonium chloride
- Find an optimal production plan

Volsay Problem (OPL)

```
dvar float+ gas;
dvar float+ chloride;

maximize 40 * gas + 50 * chloride;
subject to {
  gas + chloride <= 50;
  3 * gas + 4 * chloride <= 180;
  chloride <= 40;
}
```

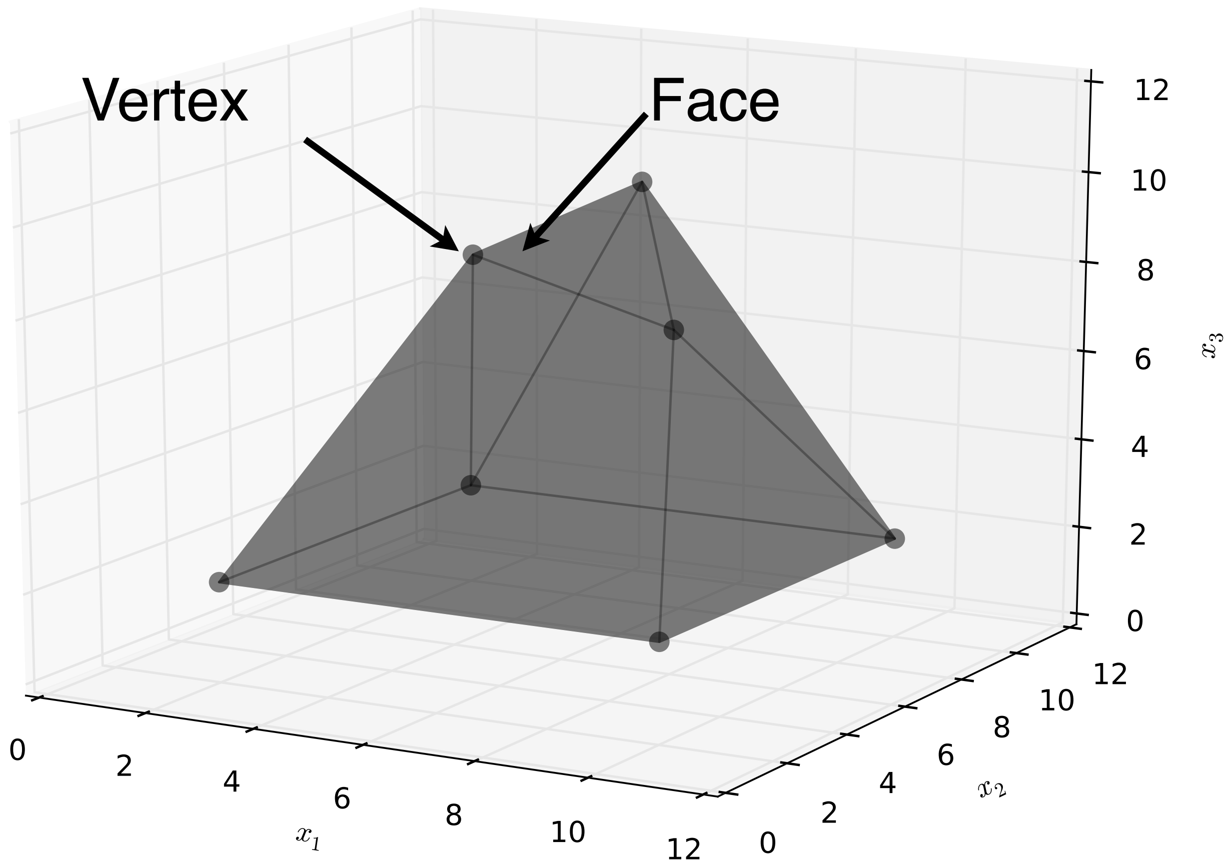
```
Solution

Objective value: 2300
gas = 20
chloride = 30
```

What is a Linear Program?

$$
\begin{aligned}
&\min c_1 x_1 + \ldots + c_n x_n \\
&\text{subject to} \\
&\quad a_{11} x_1 + \ldots + a_{1n} x_n \leq b_1 \\
&\quad \ldots \\
&\quad a_{m1} x_1 + \ldots + a_{mn} x_n \leq b_m \\
&\\
&\quad x_i \geq 0 \quad (1 \leq i \leq n)
\end{aligned}
$$

- n variables, m constraints
- variables are nonnegative
- inequality constraints

Hyperplane, Facets, and Vertices



Why I Love These Vertices

$$
\begin{aligned}
&\min c_1 x_1 + \ldots + c_n x_n \\
&\text{subject to} \\
&\quad a_{11} x_1 + \ldots + a_{1n} x_n \leq b_1 \\
&\quad \ldots \\
&\quad a_{m1} x_1 + \ldots + a_{mn} x_n \leq b_m \\
&\\
&\quad x_i \geq 0 \quad (1 \leq i \leq n)
\end{aligned}
$$

- Theorem: At least one of the points where the objective value is minimal is a vertex.

How to Obtain a Basic Feasible Solution?

- Consider $Ax = b$
- Choose m linearly independent columns A_B

$$
\begin{aligned}
&A_B x_B + A_N x_N = b \\
&A_B x_B = b - A_N x_N \\
&x_B = A_B^{-1} b - A_B^{-1} A_N x_N \\
&x_B = b' - A'_N x_N
\end{aligned}
$$

- Feasible if $b' \geq 0$
- The matrix A_B is called a *basis.*

Matrix Notations

$$
\begin{array}{ccccccccccccc}
3x_1 & + & 2x_2 & + & x_3 & & & & & & & = & 1 \\
2x_1 & & & & & + & x_4 & & & + & x_6 & = & 2 \\
x_1 & & & & & & & + & x_5 & + & x_6 & = & 3
\end{array}
$$

Basic variables = $\{x_3, x_4, x_5\}$

$$
\underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{A_B}
\underbrace{\begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix}}_{x_B}
+
\underbrace{\begin{pmatrix} 3 & 2 & 0 \\ 2 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}}_{A_N}
\underbrace{\begin{pmatrix} x_1 \\ x_2 \\ x_6 \end{pmatrix}}_{x_N}
=
\underbrace{\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}}_{b}
$$

Matrix Notations

$$\begin{array}{ccccccccccccc} 3x_1 & + & 2x_2 & + & \boxed{\begin{array}{ccccc} x_3 & & & & \\ & + & x_4 & & \\ & & & + & x_5 \end{array}} & & & = & 1 \\ 2x_1 & & & & & + & x_6 & = & 2 \\ x_1 & & & & & + & x_6 & = & 3 \end{array}$$

Basis = {3,4,5}

$$\begin{array}{ccccccccc} x_3 & = & 1 & - & 3x_1 & - & 2x_2 & & \\ x_4 & = & 2 & - & 2x_1 & & & - & x_6 \\ x_5 & = & 3 & & & & & - & x_6 \end{array}$$

How to Obtain a Basic Feasible Solution?

- Linear programming

$$\begin{array}{ll} \min & cx \\ \text{subject to} & \\ & Ax = b \end{array}$$

- Basic feasible solution: Basis B

$$x_B = A_B^{-1} b - A_B^{-1} A_N x_N$$

- What is the cost for basis B?

$$cx \quad = \quad c_B x_B + c_N x_N$$

How to Obtain a Basic Feasible Solution?

- What is the cost for basis B?

$$
\begin{aligned}
cx &= c_B x_B + c_N x_N \\
&= c_B(A_B^{-1}b - A_B^{-1}A_N x_N) + c_N x_N \\
&= c_B A_B^{-1} b + (c_N - c_B A_B^{-1} A_N) x_N \\
&= c_B A_B^{-1} b + (c_N - c_B A_B^{-1} A_N) x_N \\
&\quad + (c_B - c_B A_B^{-1} A_B) x_B \\
&= c_B A_B^{-1} b + (c - c_B A_B^{-1} A) x
\end{aligned}
$$

- Define $\Pi = c_B A_B^{-1}$

$$cx = \Pi b + (c - \Pi A)x$$

Testing if a Basis is Optimal

- What are the costs in the basic feasible solution?

$$cx = c_B A_B^{-1} b + \boxed{(c - c_B A_B^{-1} A)} x$$

$$cx = \Pi b + \boxed{(c - \Pi A)} x$$

- The basis is optimal if these costs are non-negative

[redacted]

Computational Optimization

Linear Programming Duality: Refresher

Pascal Van Hentenryck

Goals of the Video

- Linear programming
  – duality theory

Duality

$$\begin{array}{ll} \min & c\ x \\ \text{subject to} & \\ & Ax \geq b \\ & x_j \geq 0 \end{array}$$

primal

$$\begin{array}{ll} \max & y\ b \\ \text{subject to} & \\ & yA \leq c \\ & y_i \geq 0 \end{array}$$

dual

Volsay Dual

Primal

```
dvar float gas;
dvar float chloride;

maximize 40 * gas + 50 * chloride;
subject to {
  gas + chloride <= 50;
  3 * gas + 4 * chloride <= 180;
  chloride <= 40;
}
```

Dual

```
dvar float+ y1;
dvar float+ y2;
dvar float+ y3;

minimize 50*y1 + 180*y2 + 40*y3;
subject to {
   y1 + 3*y2 >= 40;
   y1 + 4*y2 + y3 >= 50;
}
```

Weak Duality

primal

$$\begin{array}{ll} \min & c\,x \\ \text{subject to} & \\ & Ax \geq b \\ & x_j \geq 0 \end{array}$$

dual

$$\begin{array}{ll} \max & y\,b \\ \text{subject to} & \\ & yA \leq c \\ & y \geq 0 \end{array}$$

Let $\hat{x}$ and $\hat{y}$ be feasible solutions to the primal and dual respectively. We have that

$$c\hat{x} \geq \hat{y}A\hat{x} \geq \hat{y}b.$$

Testing if a Basis is Optimal

- What are the costs in the basic feasible solution?

$$cx \quad = \quad c_B A_B^{-1} b + \boxed{(c - c_B A_B^{-1} A)} x$$

$$cx = \Pi b + \boxed{(c - \Pi A)} x$$

- The basis is optimal if these costs are non-negative
- Ha Ha!
  - The simplex multipliers are a feasible solution to the dual

dual

$$\begin{array}{ll} \text{max} & y\,b \\ \text{subject to} & \\ & yA \leq c \\ & y \geq 0 \end{array}$$

Strong Duality

- Theorem: If the primal has an optimal solution, the dual has an optimal solution with the same cost

Consider the optimal solution x^*.
It has an associated basis B

$$x^*_B = A_B^{-1} b.$$

The dual has a feasible solution

$$y^* = c_B A_B^{-1}$$

by the optimality of the primal. Hence,

$$y^* b = c_B A_B^{-1} b = c_B x^*.$$

Economic Interpretation

- What do these dual variables mean?

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{n} c_j x_j \\
\text{subject to} \quad & \\
& \sum_{j=1}^{n} a_{ij} x_j \quad \leq \quad b_i + t_i \quad (1 \leq i \leq m)
\end{aligned}
$$

- for some "small" t_i, this linear program has an optimal

$$z^* + \sum_{i=1}^{m} y_i^* \ t_i$$

optimal primal objective

dual solution

Volsay Problem: Dual Values

```
dvar float+ gas;
dvar float+ chloride;

maximize 40 * gas + 50 * chloride;
subject to {
  gas + chloride <= 50;
  3 * gas + 4 * chloride <= 180;
  chloride <= 40;
}
```

```
Solution

Objective value: 2300
gas = 20
chloride = 30
```

What happens if

```
dvar float+ gas;
dvar float+ chloride;

maximize 40 * gas + 50 * chloride;
subject to {
  gas + chloride <= 50;
  3 * gas + 4 * chloride <= 180 + 1;
  chloride <= 40;
}
```

```
Solution

Objective value: 2310
gas = 19
chloride = 31
```

Volsay Problem

```
dvar float+ gas;
dvar float+ chloride;

maximize 40 * gas + 50 * chloride;
subject to {
  gas + chloride <= 50;
  3 * gas + 4 * chloride <= 180;
  chloride <= 40;
}
```

Dual values

$y_1 = 10$
$y_2 = 10$
$y_3 = 0$

```
Solution

Objective value: 2300
gas = 20
chloride = 30
```

Volsay Problem

```
dvar float+ gas;
dvar float+ chloride;

maximize 40 * gas + 50 * chloride;
subject to {
  gas + chloride <= 51;
  3 * gas + 4 * chloride <= 180;
  chloride <= 40;
}
```

```
Solution

Objective value: 2310
gas = 24
chloride = 27
```

[redacted]

Computational Optimization

Mixed Integer Programming: Refresher

Pascal Van Hentenryck

Goals of the Video

- Mixed Integer Linear Programming (MIP)
  - introduction
  - branch and bound

What is a Mixed Integer Program?

$$
\begin{array}{llllllll}
\min & c_1x_1 & + & \dots & + & c_nx_n & & \\
\text{subject to} & & & & & & & \\
 & a_{11}x_1 & + & \dots & + & a_{1n}x_n & \leq & b_1 \\
 & \dots & & & & & & \\
 & a_{m1}x_1 & + & \dots & + & a_{mn}x_n & \leq & b_m \\
 & & & & & & & \\
 & x_i \geq 0 & & & & & & \\
 & x_i \text{ integer} & (i \in I) & & & & &
\end{array}
$$

Mixed Integer Versus Linear Programs?

- Integrality constraints
  – the gap between P and NP

The Knapsack Problem

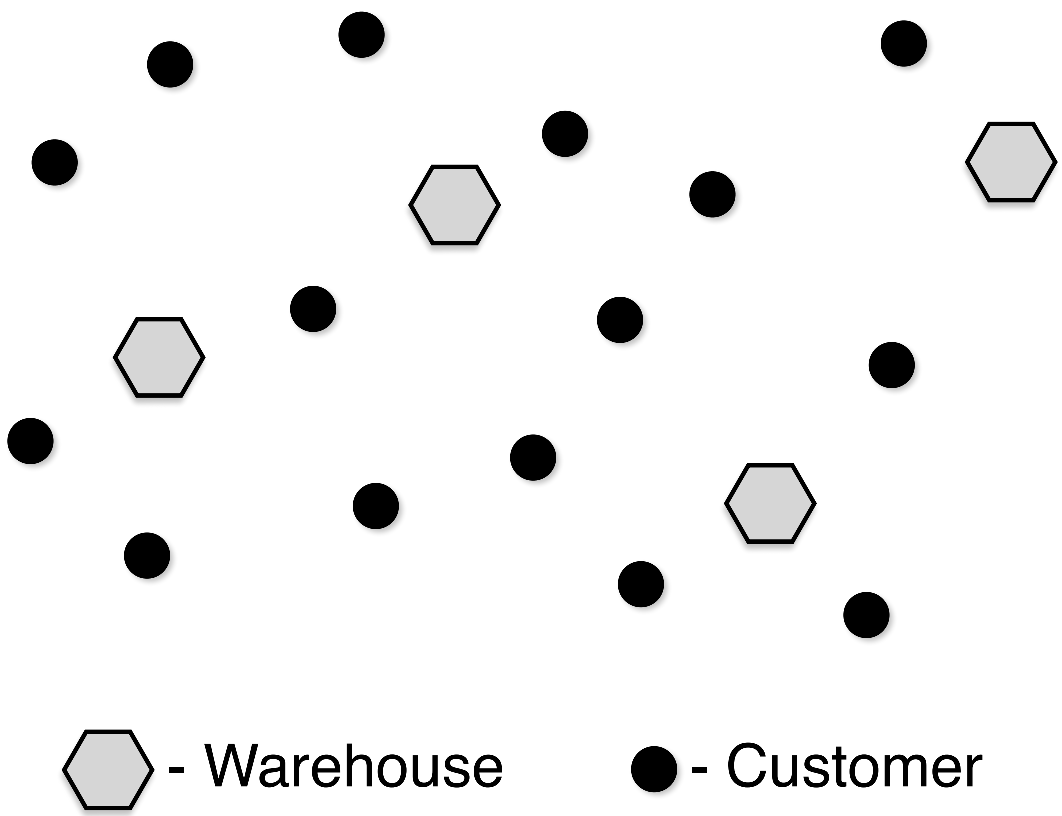
$$\text{maximize} \quad \sum_{i \in I} v_i \; x_i$$

subject to

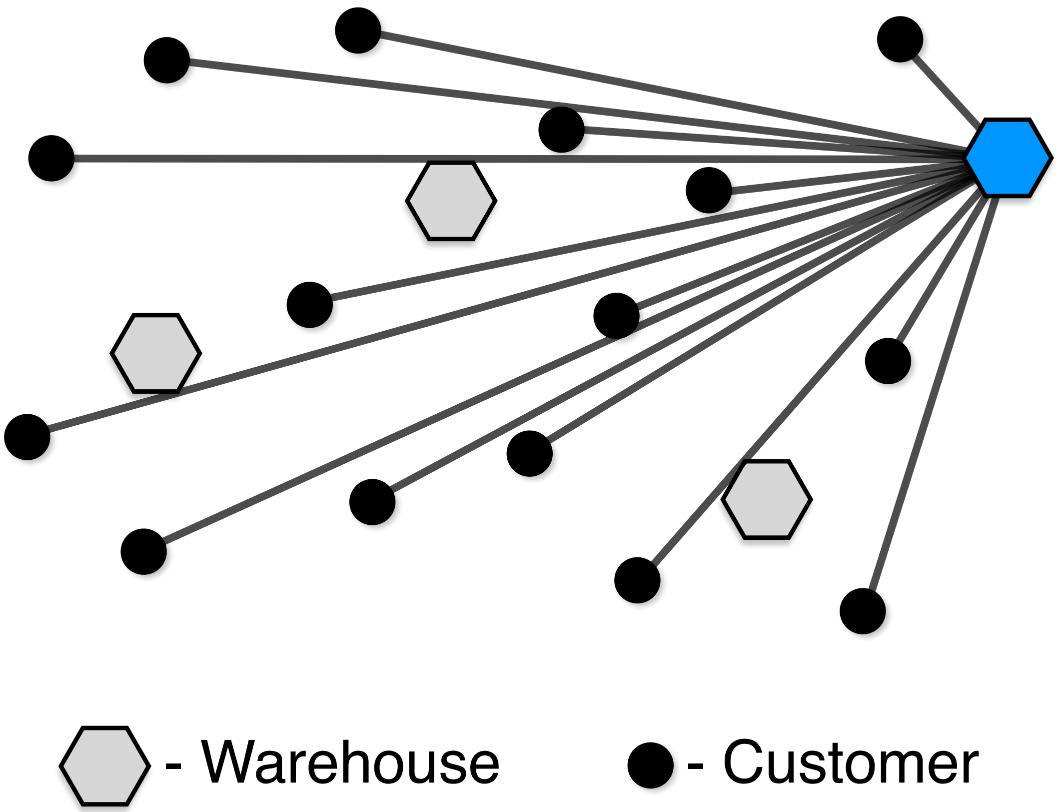
$$\sum_{i \in I} w_i x_i \leq K$$

$$x_i \in \{0, 1\} \quad (i \in I)$$

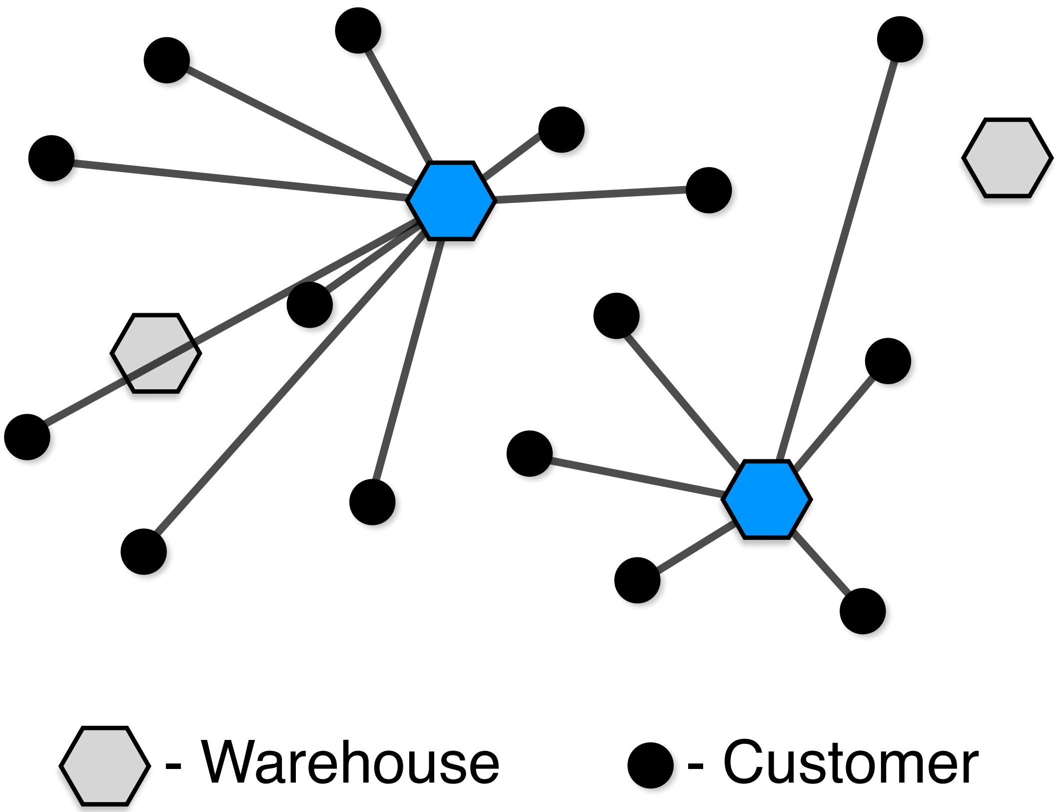
Warehouse Location



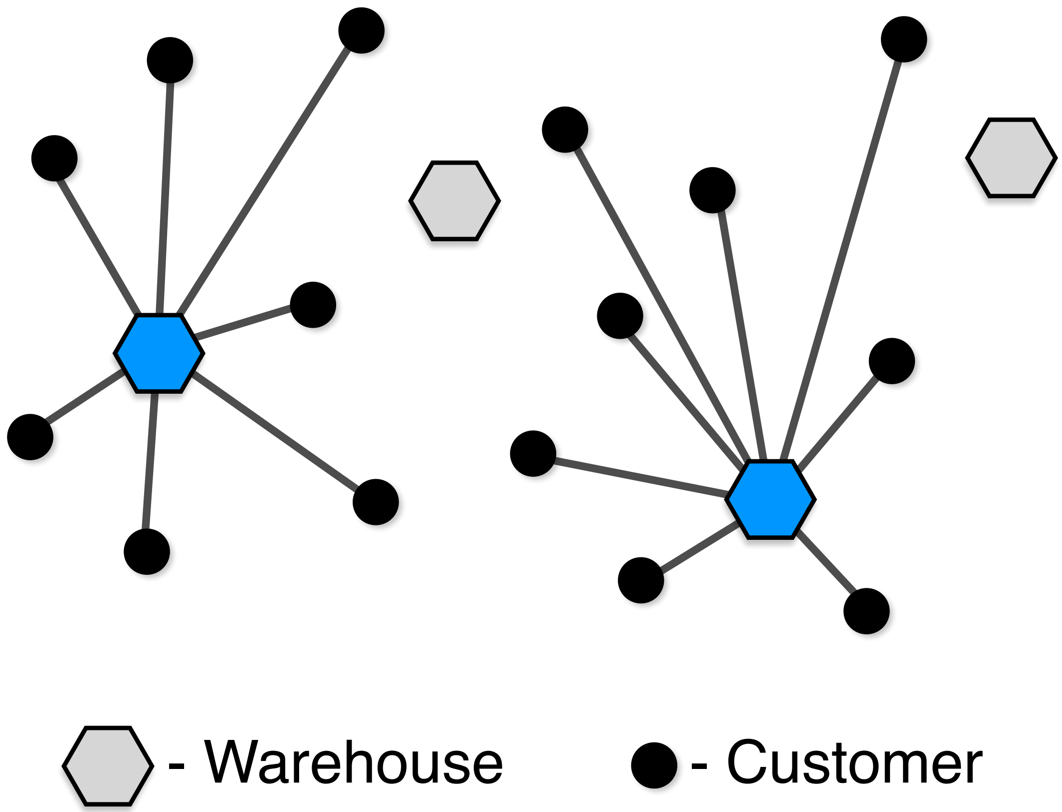
Warehouse Location



Warehouse Location



Warehouse Location



Warehouse Location

- Can we find a MIP model?
  - what are the decision variables?
  - what are the constraints?
  - what is the objective function?
- Decision variables
  - for each warehouse, decide whether to open it
    - $x_w = 1$ if warehouse w is open
  - decide whether a warehouse serves a customer
    - $y_{wc} = 1$ if warehouse w serves customer c

Warehouse Location

- Decision variables
  - for each warehouse, decide whether to open it
    - $x_w = 1$ if warehouse w is open
  - decide whether a warehouse serves a customer
    - $y_{wc} = 1$ if warehouse w serves customer c
- What are the constraints?
  - a warehouse can serve a customer only if it is open
  - a customer must be served by exactly one warehouse

Warehouse Location

- Decision variables
  - for each warehouse, decide whether to open it
    - $x_w = 1$ if warehouse w is open
  - decide whether a warehouse serves a customer
    - $y_{wc} = 1$ if warehouse w serves customer c
- What are the constraints?
  - a warehouse can serve a customer only if it is open

$$y_{w,c} \leq x_w$$

  - a customer must be served by exactly one warehouse

$$\sum_{w \in W} y_{w,c} = 1$$

Warehouse Location

- Decision variables
  - for each warehouse, decide whether to open it
    - $x_w = 1$ if warehouse w is open
  - decide whether a warehouse serves a customer
    - $y_{wc} = 1$ if warehouse w serves customer c
- What is the objective function?

$$\sum_{w \in W} c_w \, x_w + \sum_{w \in W, c \in C} t_{w,c} \, y_{w,c}$$

fixed cost   transportation cost

Warehouse Location

$$
\begin{array}{lll}
\min & \sum_{w \in W} c_w \; x_w + & \sum_{w \in W, c \in C} t_{w,c} \; y_{w,c} \\
\text{subject to} & & \\
& y_{w,c} \leq x_w & (w \in W, c \in C) \\
& \sum_{w \in W} y_{w,c} = 1 & (c \in C) \\
& x_w \in \{0,1\} & (w \in W) \\
& y_{w,c} \in \{0,1\} & (w \in W, c \in C)
\end{array}
$$

Warehouse Location

- Decision variables
  - for each warehouse, decide whether to open it
    - $x_w = 1$ if warehouse w is open
  - decide whether a warehouse serves a customer
    - $y_{wc} = 1$ if warehouse w serves customer c

- Why not use
  - y_c denotes the warehouse serving customer c?

The Role of 0/1 Variables

- MIP models love 0/1 variables
  - integer variables are typically 0/1 variables
- Linear constraints are easy to state
  - when using 0/1 variables
- Still many possible models to consider
  - decision variables
  - constraints
  - objectives

Branch and Bound

- How to solve MIP models?
  - active research area for many many decades
- Branch and bound
  - Bounding: finding an optimistic relaxation
  - Branching: splitting the problem in subproblems
- MIP models have a natural relaxation
  - the linear relaxation
  - remove the integrality constraint on variables

Branch and Bound for MIP Models

- Solve the linear relaxation
- If the linear relaxation is worse than the best solution found so far, prune this node
  - the associated problem is suboptimal
- If the linear relaxation is integral, we have found a feasible solution
  - update the best feasible solution if appropriate
- Otherwise, find an integer variable x that has a fractional value f in the linear relaxation
  - create two subproblems $\mathrm{x} \leq \lfloor f \rfloor$ and $\mathrm{x} \geq \lceil f \rceil$
  - repeat the algorithm on the subproblems

The Knapsack Problem

$$\text{maximize} \quad \sum_{i \in I} v_i \; x_i$$

$$\text{subject to}$$

$$\sum_{i \in I} w_i x_i \leq K$$

$$x_i \in \{0, 1\} \quad (i \in I)$$

The Knapsack Problem: Linear Relaxation

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i \in I} v_i \; x_i \\ \text{subject to} & \\ & \displaystyle\sum_{i \in I} w_i x_i \leq K \\ & 0 \leq x_i \leq 1 \quad (i \in I) \end{array}$$

Branch and Bound for MIP models

- Linear relaxation
  - a greedy algorithm
- How do we branch?
  - variable with a fractional value
  - i.e., most valuable item that cannot be fit entirely
- What do the subproblems mean?
  - do not take that item
    - what is the linear relaxation going to do?
  - take this item
    - what is the linear relaxation going to do?

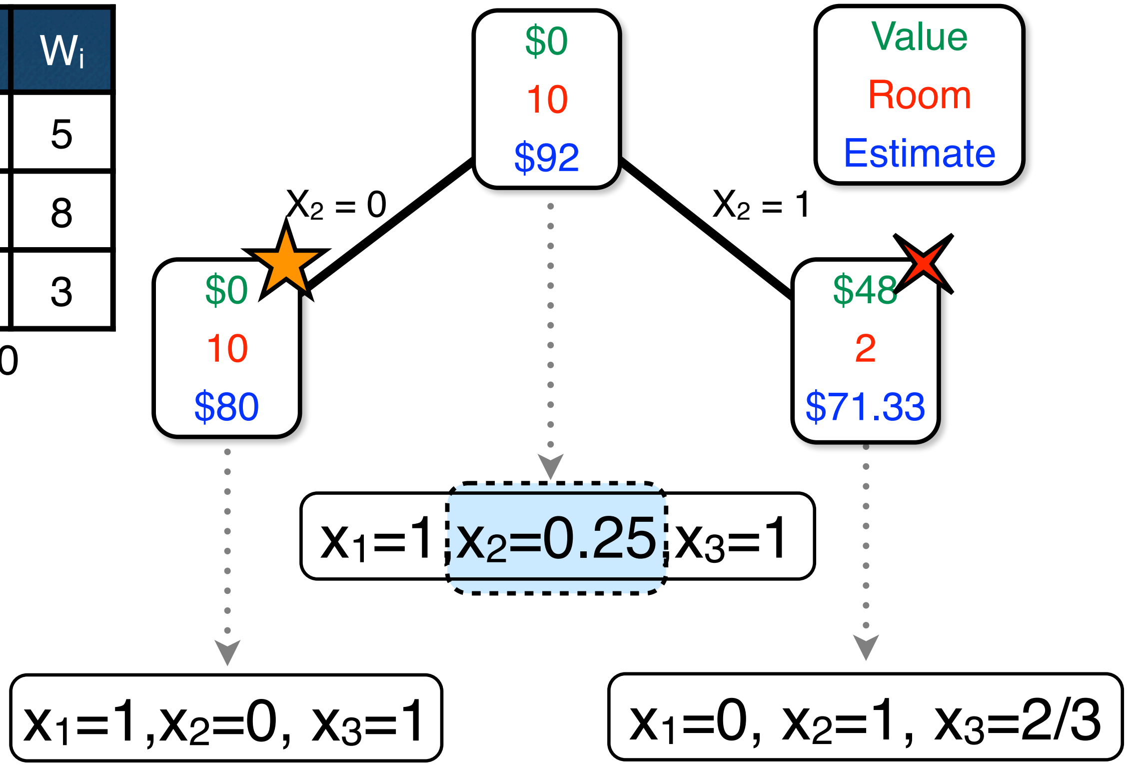
Branch and Bound for MIP models

- Linear relaxation
  - greedy algorithm
- How do we branch?
  - variable with a fractional value
  - i.e., most valuable item that cannot be fit entirely
- What do the subproblems mean?
  - do not take that item
    - which item is now fractional?
  - take this item
    - which item is now fractional?

Depth-First Branch and Bound

i	V_i	W_i
1	45	5
2	48	8
3	35	3

K = 10



Branch and Bound for MIP models

- When is Branch and Bound effective?
  - necessary condition: the linear relaxation is strong
    - is it sufficient?

- What is a good MIP model?
  - one with a good linear relaxation

- Which variable should one branch on?
  - most fractional value
    - why? exaggerate ...

Branch and Bound for MIP models

- When is Branch and Bound effective?
  - need to prune suboptimal solutions early
  - necessary condition: the linear relaxation is strong
    - is it sufficient?

- What is a good MIP model?
  - one with a good linear relaxation

Warehouse Location

- Decision variables
  - for each warehouse, decide whether to open it
    - $x_w = 1$ if warehouse w is open
  - decide whether a warehouse serves a customer
    - $y_{wc} = 1$ if warehouse w serves customer c
- What are the constraints?
  - a warehouse can serve a customer only if it is open
  - a customer must be served by exactly one warehouse

Warehouse Location

- Decision variables
  - for each warehouse, decide whether to open it
    - $x_w = 1$ if warehouse w is open
  - decide whether a warehouse serves a customer
    - $y_{wc} = 1$ if warehouse w serves customer c
- What are the constraints?
  - a warehouse can serve a customer only if it is open

$$y_{w,c} \leq x_w$$

  - a customer must be served by exactly one warehouse

$$\sum_{w \in W} y_{w,c} = 1$$

Warehouse Location

$$
\begin{array}{lll}
\min & \sum_{w \in W} c_w \; x_w + & \sum_{w \in W, c \in C} t_{w,c} \; y_{w,c} \\
\text{subject to} & & \\
& \sum_{c \in C} y_{w,c} \leq |C| x_w & (w \in W) \\
& \sum_{w \in W} y_{w,c} = 1 & (c \in C) \\
& x_w \in \{0,1\} & (w \in W) \\
& y_{w,c} \in \{0,1\} & (w \in W, c \in C)
\end{array}
$$

Warehouse Location

- Which of the two models is best?
  - our new model has a single constraint instead of |C| constraints for each warehouse

- What about the quality of linear relaxation?

Warehouse Location

- A solution to

$$y_{w,c} \leq x_w \quad (w \in W, c \in C)$$

- is also a solution to

$$\sum_{c \in C} y_{w,c} \leq |C| x_w \quad (w \in W)$$

- but not vice-versa.

- So the initial model has a better linear relaxation!

Warehouse Location Relaxations

$$y_{w,c} \leq x_w \quad (w \in W, c \in C)$$

W	C	OBJ$_1$	OBJ$_2$	%
16	50	932,615	844,807	9.5
16	50	1,010,641	853,434	15.6
25	50	796,648	659,341	17.2
50	50	793,439	631,421	20.4

$$\sum_{c \in C} y_{w,c} \leq |C| x_w \quad (w \in W)$$

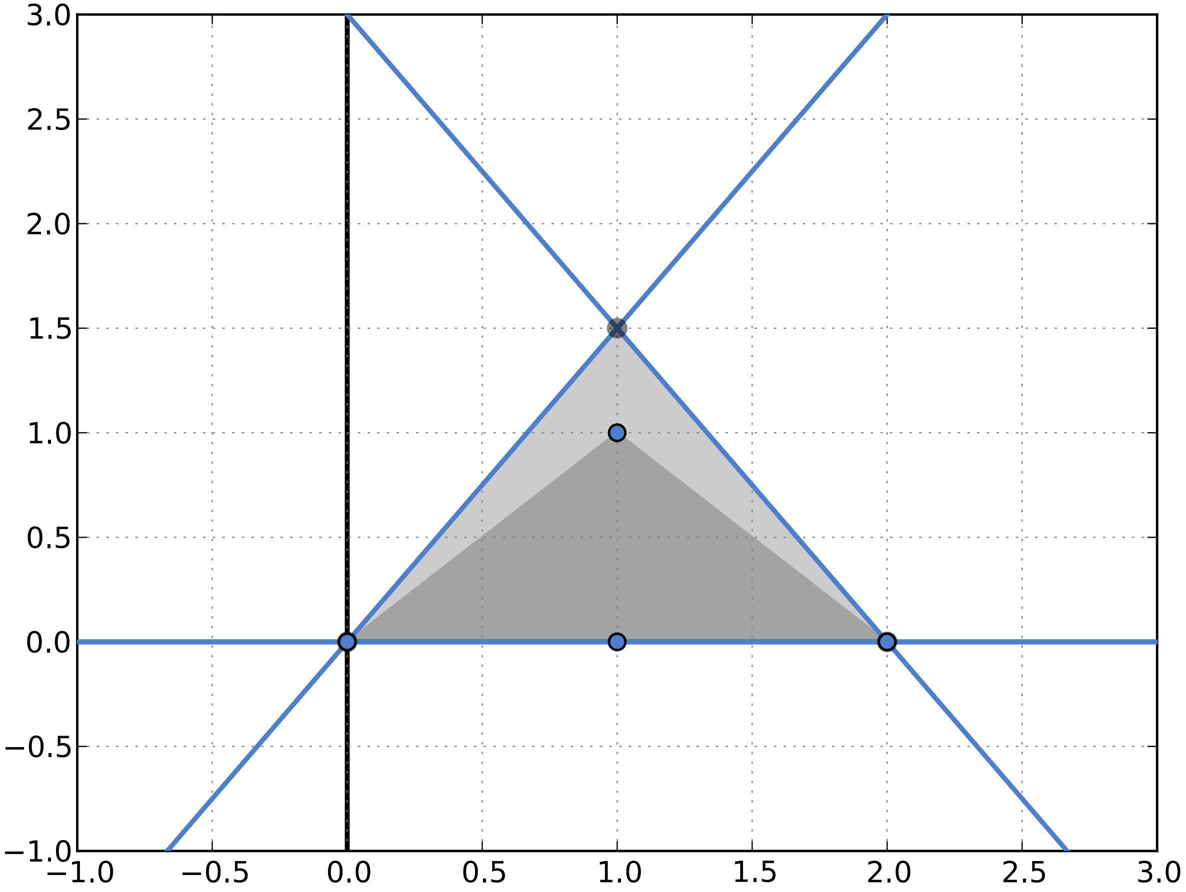
[redacted]

Computational Optimization

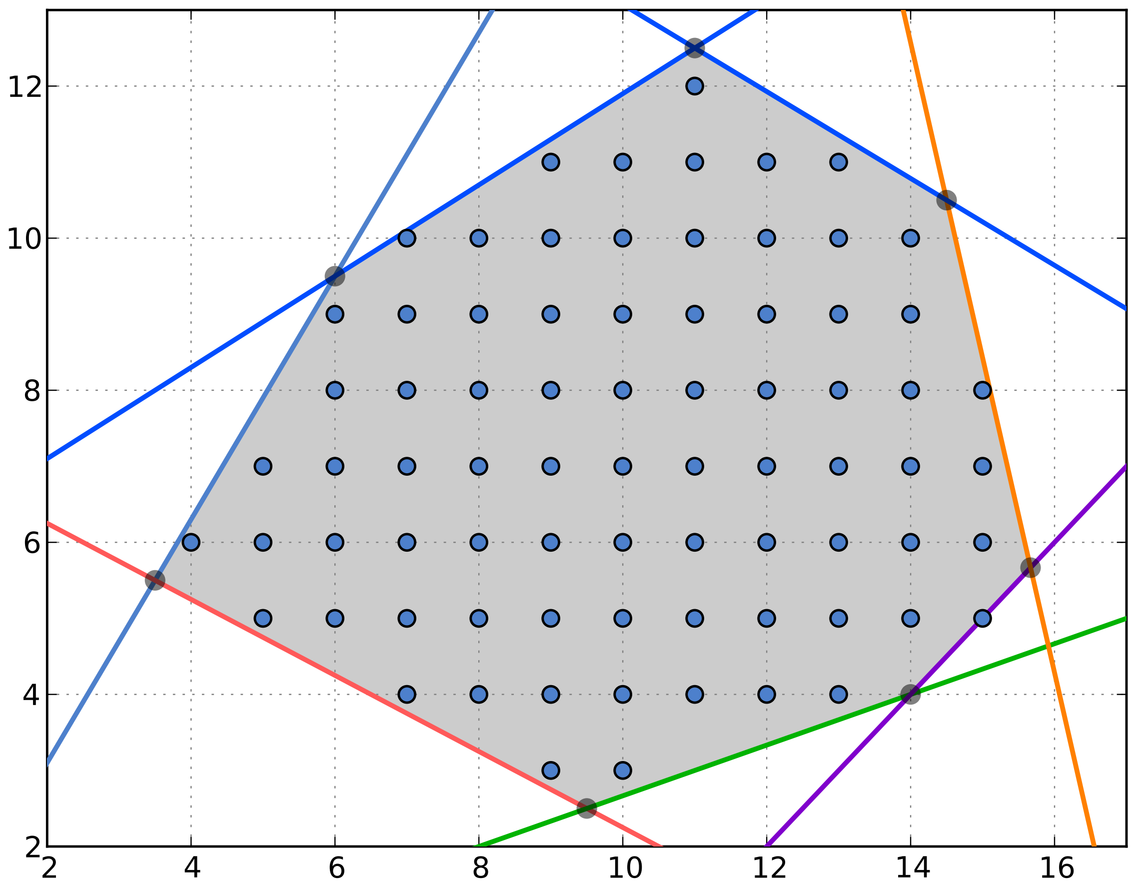
Mixed Integer Programming: Refresher II

Pascal Van Hentenryck

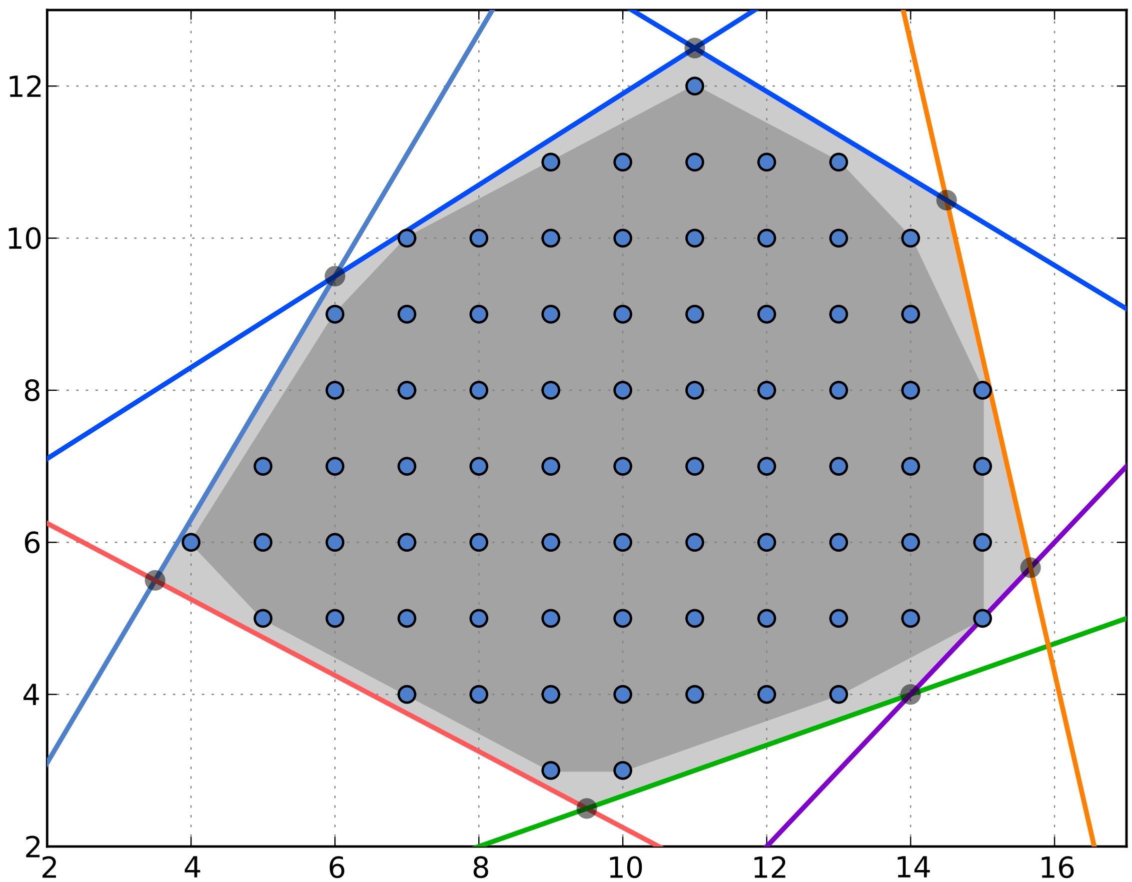
Convex Hull



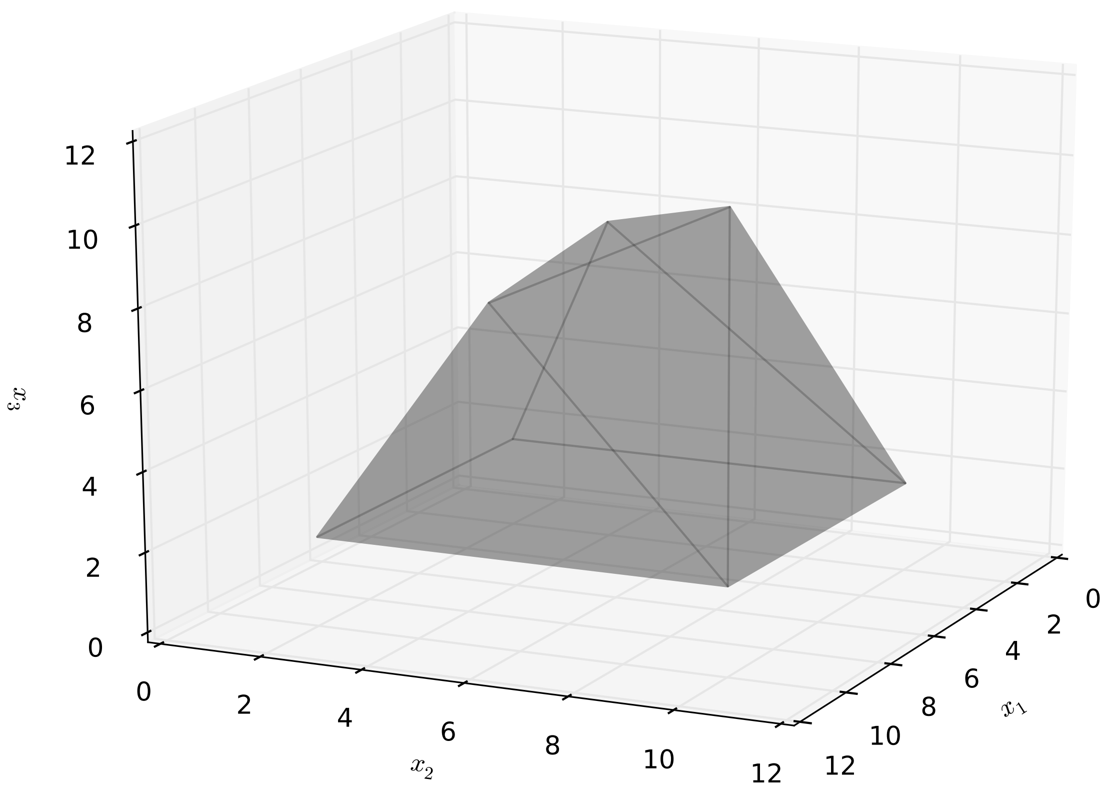
Convex Hull



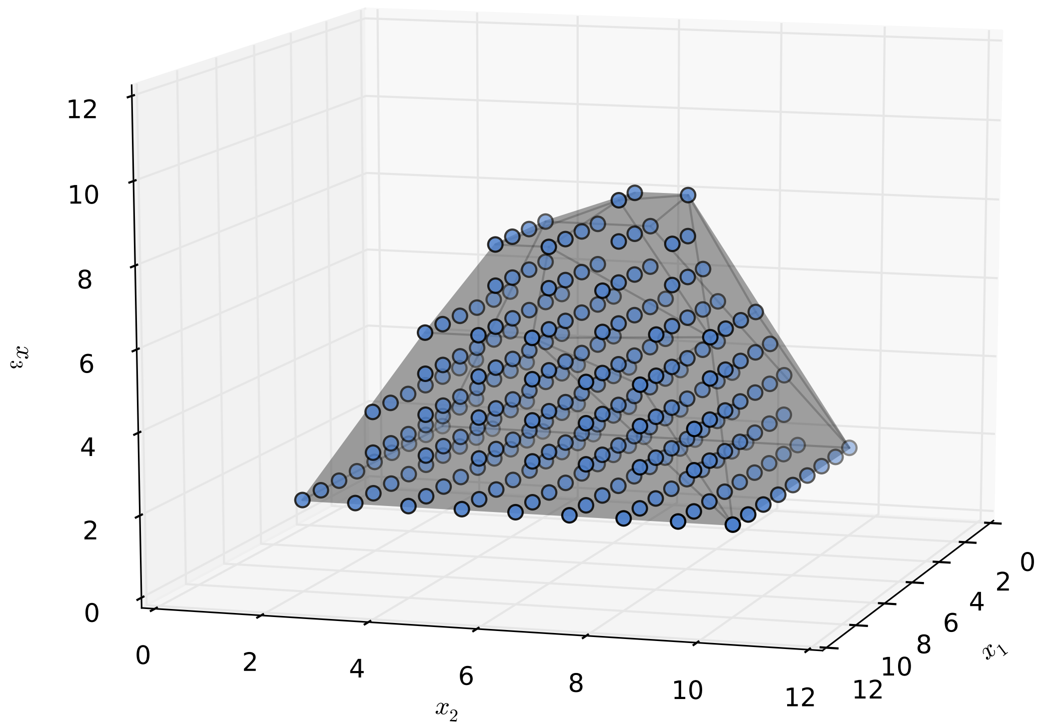
Convex Hull



Convex Hull



Convex Hull



Why the Convex Hull?

- If I had the convex hull of the integer solutions
  - I could use linear programming to solve the problems
- Why?
  - because an optimal solution will be on a vertex
  - each vertex is feasible

Polyhedral Cuts

- Polyhedral cuts
  - cuts that represent the facets of the convex hull of the integer solutions
- These cuts are valid
  - they do not remove any solution
- The cuts are as strong as possible
  - if we have all of them, we could use linear programming to solve the problem

Branch and Cut

- Basic idea
  1. formulate the application as a MIP;
  2. solve the linear relaxation; if the linear relaxation is integral, terminate;
  3. find a polyhedral cut which prunes the linear relaxation and is a facet if possible; if you can find such beautiful mathematical object, go back to step 2;
  4. otherwise, settle for the poor man’s choice and branch

The Separation Problem

- Consider a solution x^* to the linear relaxation possibly enhanced by a number of cuts
- We wish to know whether there exists a polyhedral cut that cut x^*


“That’s all folks!”

Computational Optimization

Mixed Integer Programming: Refresher III

Pascal Van Hentenryck

Warehouse location

- Assume that I have opened the warehouses
  – but there is a limit on how many customers can be served by a warehouse

$$
\begin{aligned}
\min \quad & \sum_{w \in W, c \in C} t_{w,c} \, y_{w,c} \\
\text{subject to} \quad & \\
& y_{w,c} \leq \bar{x}_w && (w \in W, c \in C) \\
& \sum_{w \in W} y_{w,c} = 1 && (c \in C) \\
& \sum_{c \in C} y_{w,c} \leq n_w && (w \in W) \\
& y_{w,c} \in \{0, 1\} && (w \in W, c \in C)
\end{aligned}
$$

Warehouse location

- This is a MIP

$$
\begin{array}{ll}
\min & \displaystyle\sum_{w\in W, c\in C} t_{w,c}\; y_{w,c} \\
\text{subject to} & \\
& y_{w,c} \leq \bar{x}_w \qquad\qquad (w \in W, c \in C) \\
& \displaystyle\sum_{w\in W} y_{w,c} = 1 \qquad\qquad (c \in C) \\
& \displaystyle\sum_{c\in C} y_{w,c} \leq n_w \qquad\qquad (w \in W) \\
& y_{w,c} \in \{0,1\} \qquad\qquad (w \in W, c \in C)
\end{array}
$$

- I claim that it is easy and can be solved by linear programming
  - will be important for decomposition algorithms later on

Total Unimodularity

- What is total unimodularity?
  - a property that guarantees that a linear program is guaranteed to have only integer vertices
- Consequences
  - you do not need to solve the MIP
  - solving the LP is sufficient

Warehouse location

- I only need to solve this LP

$$
\begin{aligned}
\min \quad & \sum_{w \in W, c \in C} t_{w,c} \; y_{w,c} \\
\text{subject to} \quad & \\
& y_{w,c} \leq \bar{x}_w && (w \in W, c \in C) \\
& \sum_{w \in W} y_{w,c} = 1 && (c \in C) \\
& \sum_{c \in C} y_{w,c} \leq n_w && (w \in W) \\
& 0 \leq y_{w,c} \leq 1 && (w \in W, c \in C)
\end{aligned}
$$

Total Unimodularity

- How do we define total unimodularity?
- Consider the polyhedra

$$P = \{x \mid Ax \leq b\ ,\ \ x \geq 0\}$$

- What are the conditions on A and b that guarantee that the vertices of P are all integral?
  - A is totally unimodular
  - b is integer

Total Unimodular Matrices

- A matrix A is unimodular if det(A) = 1 or -1.
- A matrix A is Totally Unimodular (TU) if each square submatrix B of A has det(B) = 0, 1, or -1.
- Note that, if A is TU
  – entries of A must be 0, -1, 1 (why?)
  – adding a row (0,…0,1,0,…,0) to A gives a TU matrix

Sufficient Condition

- A matrix A is totally unimodular if there are at most 2 non-zeros in each column and if the rows can be partitioned into two sets I_1 and I_2 such that
  - If a column has two entries of the same sign, their rows are in different sets of the partition
  - If a column has two entries of different signs their rows are in the same set of the partition

Corollary

- A matrix A is totally unimodular if
  - it has almost two non zero entries per column
  - the sum of the entries of a column is zero

Warehouse location

- After removing the warehouse that are closed

$$
\begin{aligned}
\min \quad & \sum_{w \in W, c \in C} t_{w,c}\, y_{w,c} \\
\text{subject to} \quad & \\
& \sum_{w \in W} y_{w,c} = 1 && (c \in C) \\
& \sum_{c \in C} y_{w,c} \leq n_w && (w \in W) \\
& 0 \leq y_{w,c} \leq 1 && (w \in W, c \in C)
\end{aligned}
$$

- How do the constraints look like?

Looking at the Matrix

[redacted]

Customers

	1	2	3	1	2	3	1	2	3	1	2	3
1	$y_{1,1}$	$y_{2,1}$	$y_{3,1}$									
2				$y_{1,2}$	$y_{2,2}$	$y_{3,2}$						
3							$y_{1,3}$	$y_{2,3}$	$y_{3,3}$			
4										$y_{1,4}$	$y_{2,4}$	$y_{3,4}$
1	$y_{1,1}$			$y_{1,2}$			$y_{1,3}$			$y_{1,4}$		
2		$y_{2,1}$			$y_{2,2}$			$y_{2,3}$			$y_{2,4}$	
3			$y_{3,1}$			$y_{3,2}$			$y_{3,3}$			$y_{3,4}$

Warehouses

[redacted]

Customers

	1	2	3	1	2	3	1	2	3	1	2	3
1	1	1	1									
2				1	1	1						
3							1	1	1			
4										1	1	1
1	1			1			1			1		
2		1			1			1			1	
3			1			1			1			1

Warehouses

[redacted]

	1	2	3	1	2	3	1	2	3	1	2	3
1	1	1	1									
2				1	1	1						
3							1	1	1			
4										1	1	1
1	1			1			1			1		
2		1			1			1			1	
3			1			1			1			1

Customers

Warehouses

[redacted]

		1	2	3	1	2	3	1	2	3	1	2	3
Customers	1	1	1	1									
	2				1	1	1						
	3							1	1	1			
	4										1	1	1
Warehouses	1	1			1			1			1		
	2		1			1			1			1	
	3			1			1			1			1

Total Unimodularity

- Typical totally unimodular problems
  - shortest paths
  - flow problems
  - assignment problems


"That's all folks!"