Discrete Optimization
(at IBM's Mathematical Sciences Department)

Sanjeeb Dash
IBM Research

Lecture, ORF 363
Princeton University, Dec 15, 2015

Outline

▷ Real-world optimization (and at IBM)

▷ Discrete Optimization basics
- Problems
- Computational Complexity
- Formulations

▷ Solution techniques: integer programming

▷ Applications

Real-world Optimization

IBM Research

IBM: 379,592 employees (end of 2014)
IBM Research: 12 labs, 1800+ researchers

IBM's Math. Sciences Dept.

IBM Mathematical Sciences Department:
◇ 50+ years old
◇ 50+ people
◇ 50 % funding from contracts, 50% from IBM grants

- 40% of time spent on applied work $\equiv$ need to publish 2-3 papers (or perish)
- 100% of time spent on applied work $\equiv$ need to publish 0 papers

Discrete Optimization

Discrete optimization is the study of problems where the goal is to select a minimum cost alternative from a finite (or countable) set of alternatives.

Application areas

Airlines	route planning, crew scheduling	American, United
	revenue management	Air New Zealand, British Airways
Package Delivery	vehicle routing	UPS, Fedex, USPS
Trucking	route planning, vehicle routing	Schnieder
Transportation	network optimization	Amazon
Telecommunication	network design	AT&T
Shipping	route planning	Maersk
Pipelines	batch scheduling	CLC
Steel Industry	cutting stock	Posco
Paper Industry	cutting stock	GSE mbH
Finance	portfolio management	Axioma
Oil & Gas		ExxonMobil
Petrochemicals		SK Innovation
Power generation	unit commitment, resource management	BC Hydro
Railways	Timetabling, crew-scheduling	BNSF, CSX, Belgian Railways, Deutsche Bahn, Trenitalia

Recent jobs in optimization

2015

Apple - *Operations Research Scientist*

Supply chain optimization - Cplex/Gurobi/XpressMP (M.S./Ph.D.)

Amazon - *Operations Research Scientist*

Network optimization, statistics/mathematical programming (R, SPSS, CPLEX,LINDO or Xpress) (Ph.D.)

BNSF -*OR & Advanced Analytics Specialist I*

Railroad logistics - CPLEX, Gurobi, ProModel, ARENA, Frontline Solver (Ph.D.)

FedEx - *Senior Operations Research Analyst*

Mixed-integer programming software such as CPLEX/Gurobi (Ph.D.)

Ford - *Operations Research Analyst*

Capacity planning, plant scheduling - “mixed integer programming formulations and computationally efficient methods for obtaining optimal or near-optimal solutions" - CPLEX, Python, R (Ph.D.)

GrubHub - *Operations Research Scientist*

"Optimize driver dispatch and routing", vehicle routing, and facility location - AMPL (Ph.D.)

Sears - *Operations Research Data Scientist - Supply Chain*

Supply chain management, data mining, mathematical programming (Ph.D.)

Turner Broadcasting Systems - *Senior Operations Research Analyst*

Decision support models, R, MATLAB, CPLEX, SAS (M.S./Ph.D.)

Uber - *Operations Research & Data Science*
Operations Research, Optimization, ... (M.S.)

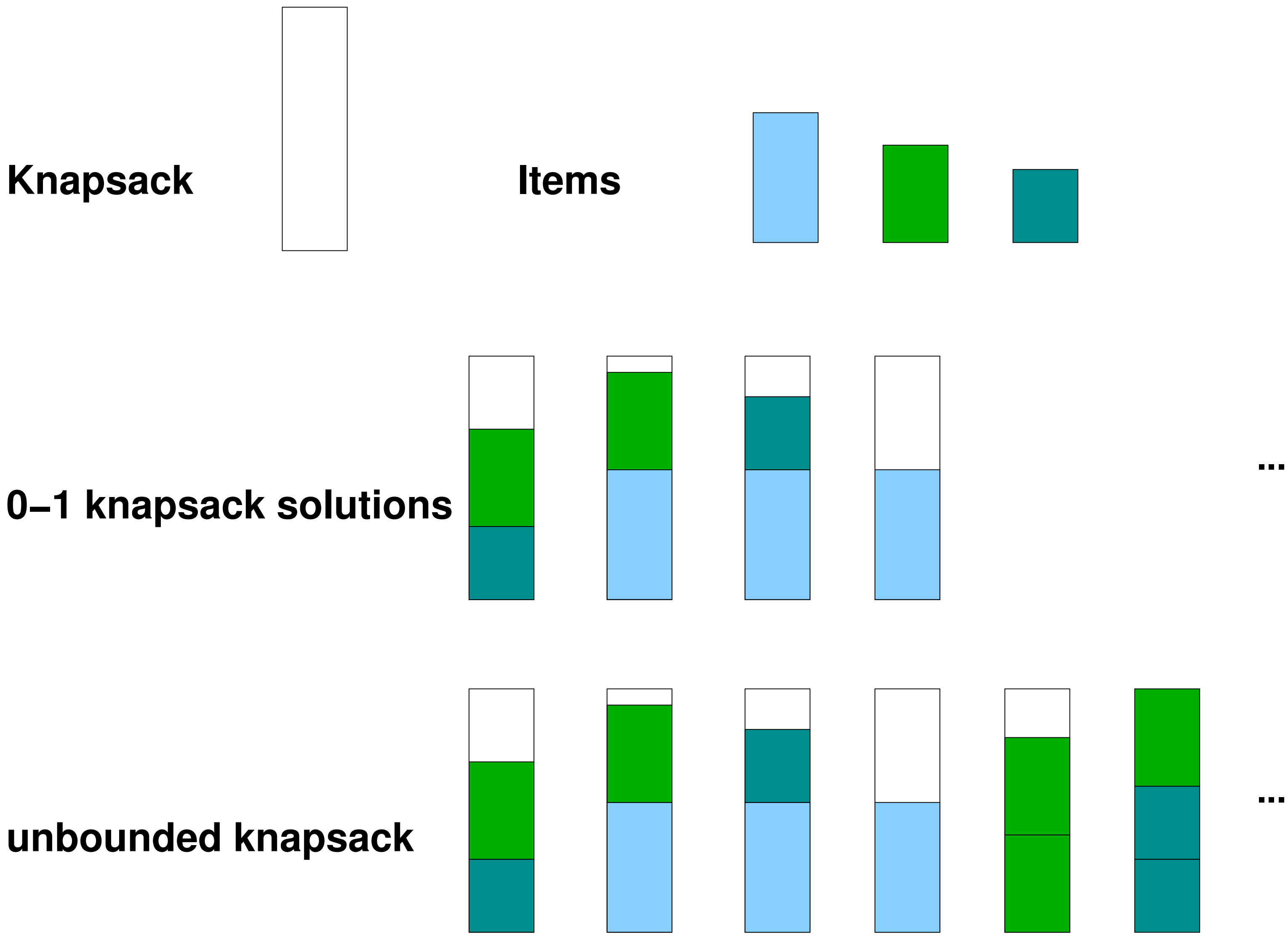
IBM, SAS, Gurobi, Mosek, ORTEC

Problems

Knapsack Problem

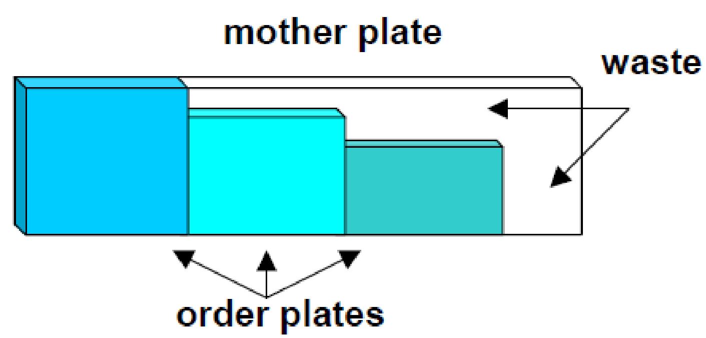
Maximize the value of items packed in a knapsack while not exceeding its capacity

Knapsack Problem



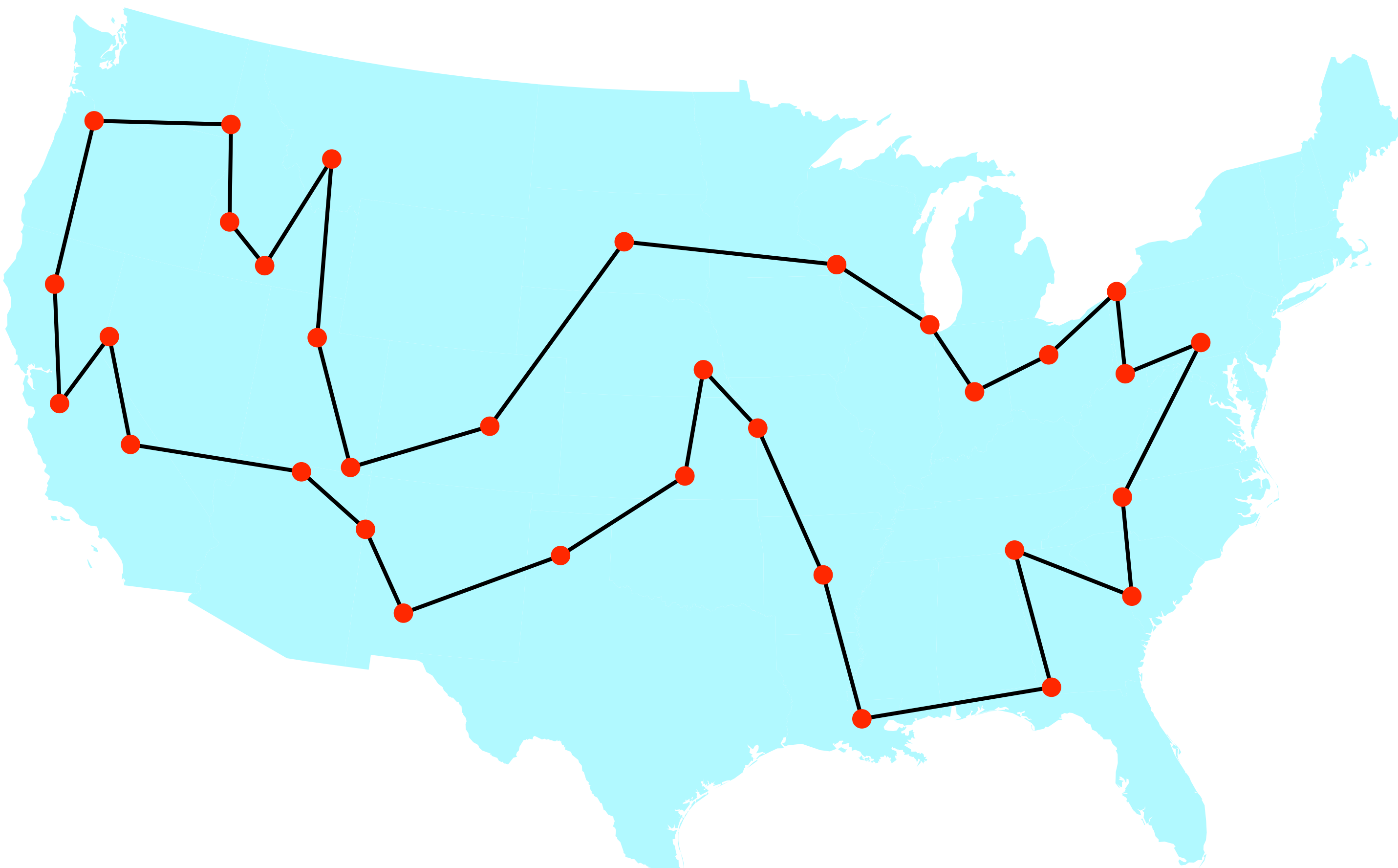
Cutting stock

Pack items into as few identical knapsacks as possible:
Used in steel, paper industry)



Traveling Salesman Problem

TSP: Minimize distance traveled while visiting a collection of cities and returning to the starting point.



33-city TSP instance from a 1962 Procter and Gamble competition ($10,000 prize won by Gerald Thompson of CMU)


HELP! WE'RE LOST!
HELP "CAR 54"...AND WIN CASH
54...$1,000 PRIZES
ONE...$10,000 GRAND PRIZE
START and FINISH
Lewiston
Portland
Butte
Boise
Twin Falls
Redding
Reno
Salt Lake City
Gustine
Lone Pine
Mexican Hat
Colorado Springs
Marble Canyon
Manuelito
Truth or Consequences
Amarillo
Blunt
Lincoln
Wichita
Kansas City
La Crosse
Chicago
Marion
Indianapolis
Erie
Carlisle
Wana
Wilkesboro
Chattanooga
Little Rock
Barnwell
Bainbridge
Baton Rouge
Map by Rand McNally
Help Toody and Muldoon find the shortest round trip route to visit all 33 locations shown on the map.
All you do is draw connecting straight lines from location to location to show the shortest round trip route.
HERE'S THE CORRECT START...
Begin at Chicago, Illinois. From there, lines show correct route as far as Erie, Pennsylvania. Next, do you go to Carlisle, Pennsylvania or Wana, West Virginia? Check the easy instructions on back of this entry blank for details.

10-city instance

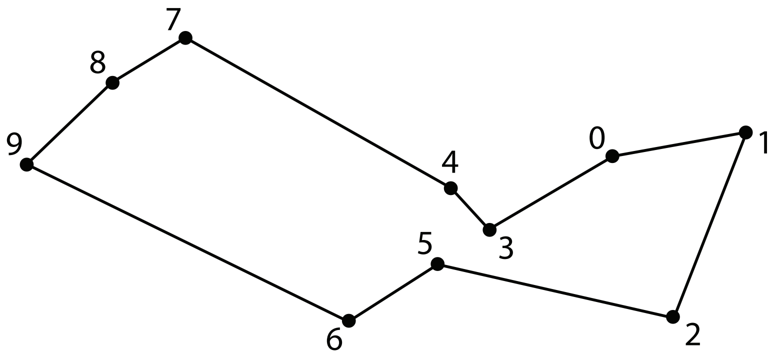
$(n-1)! = 362,880$ possible tours

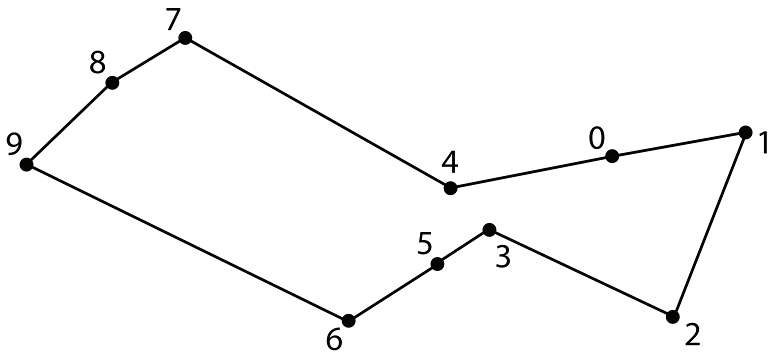
10-city instance

	0	1	2	3	4	5	6	7	8	9
0 Chicago	0									
1 Erie	449	0								
2 Chattanooga	618	787	0							
3 Kansas City	504	937	722	0						
4 Lincoln	529	1004	950	219	0					
5 Wichita	805	1132	842	195	256	0				
6 Amarillo	1181	1441	1080	563	624	368	0			
7 Butte	1538	2045	2078	1378	1229	1382	1319	0		
8 Boise	1716	2165	2217	1422	1244	1375	1262	483	0	
9 Reno	2065	2514	2355	1673	1570	1507	1320	842	432	0

10-city instance: solutions

Tours of length 6633 and 6514 miles

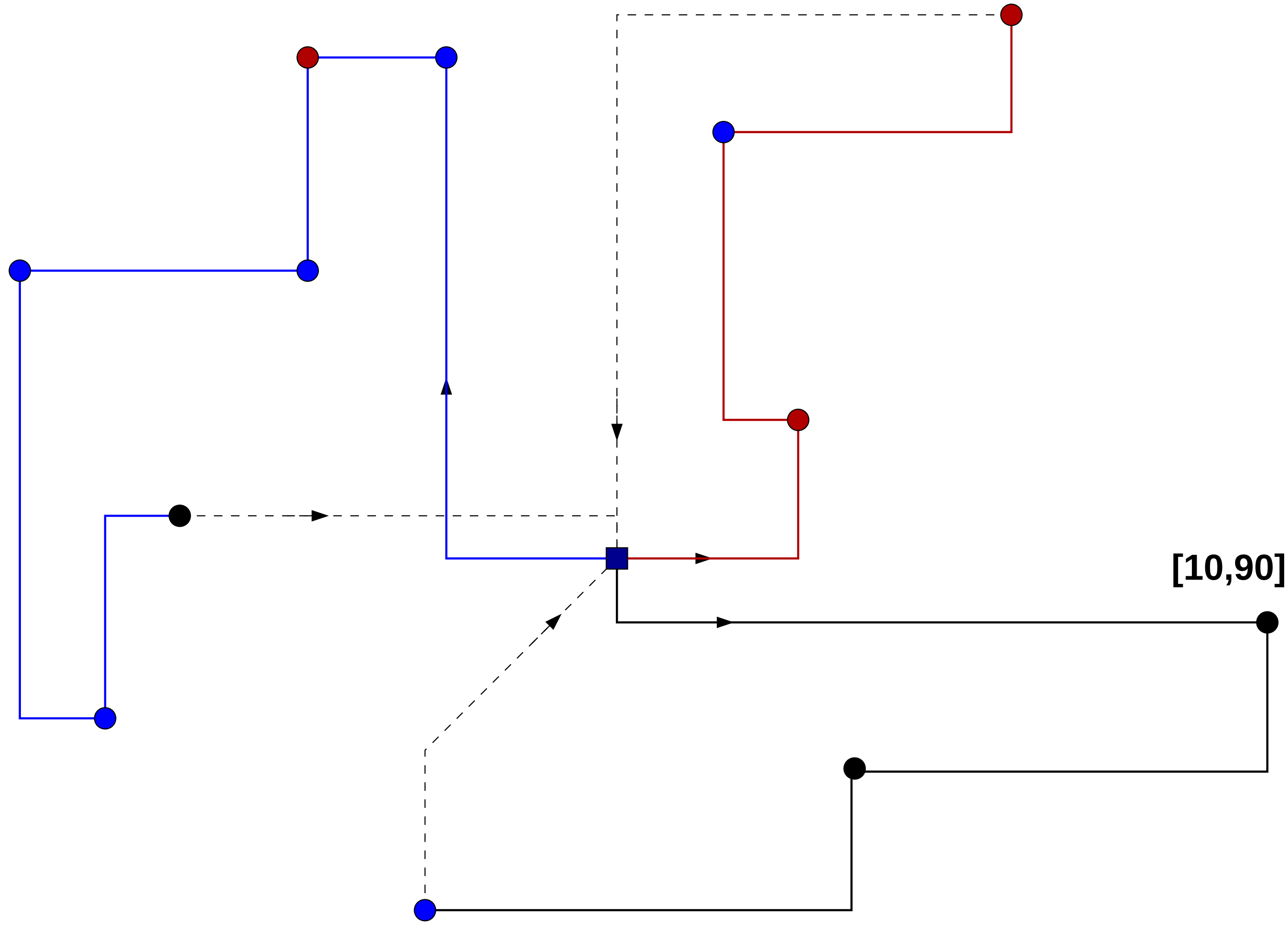




Shortest tour: 0, 1, 2, 3, 5, 6, 9, 8, 7, 4

Shortest tour length: 6514

Vehicle Routing



Minimize distance traveled by trucks at a depot delivering to a set of customers within prescribed time windows (used in package delivery by Fedex, USPS etc.)

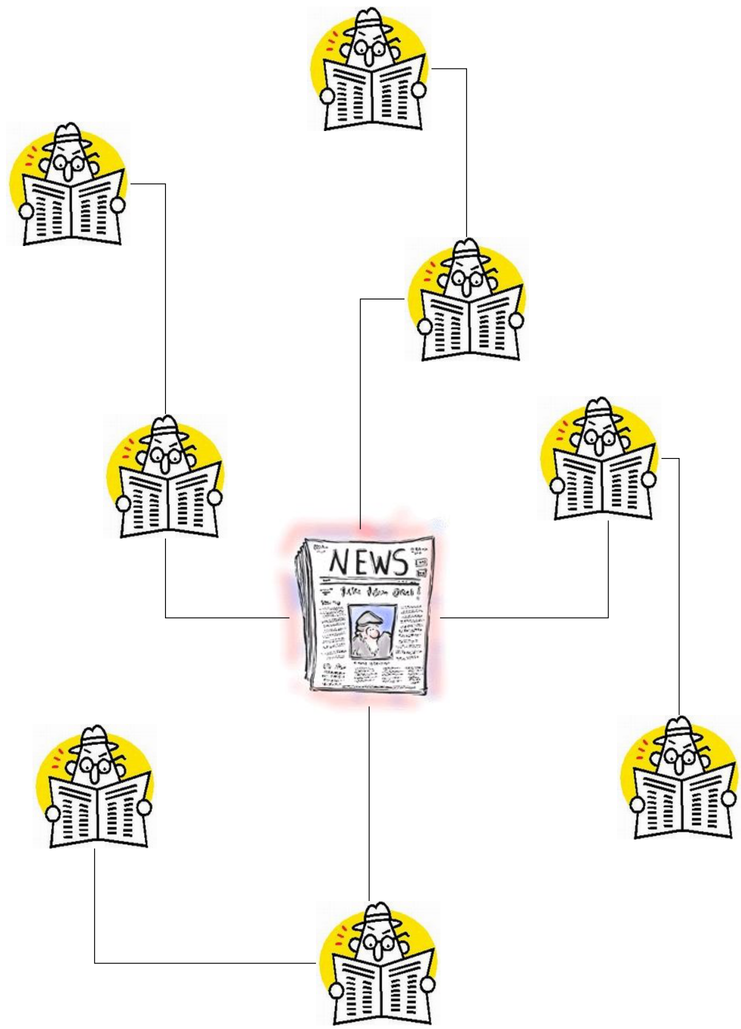
2014 survey in OR-MS Magazine lists 15+ vendors of VRP software.

Min-max vehicle routing

1 

4 

120 



1996 Whizzkids challenge

▷ 5000 Dutch Guilders prize sponsored by CMG

▷ Winners: Hemel, van Erk, Jenniskens (U. Eindhoven students)
▷ Max path length of 1183
▷ Local search techniques, 15,000 hours of computing time.

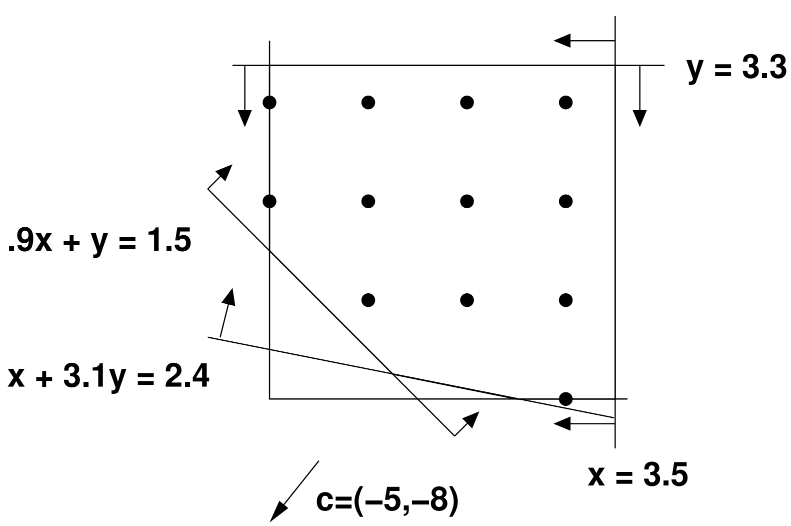
Optimal solution? Lower bound of 1160 given by Hurkens '97.

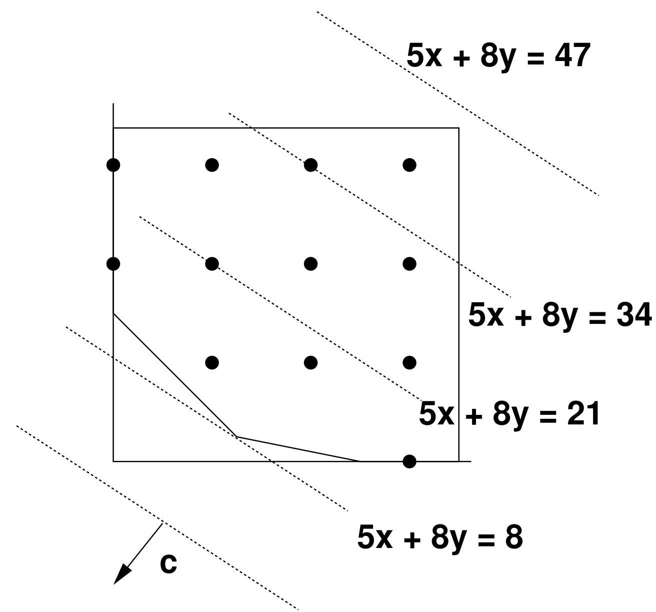
Integer programming

$\min 5x + 8y$ subject to

$.9x + y \geq 1.5, \; x + 3.1y \geq 2.4$

$0 \leq x \leq 3.5, \; 0 \leq y \leq 3.3, \; x, y$ integral

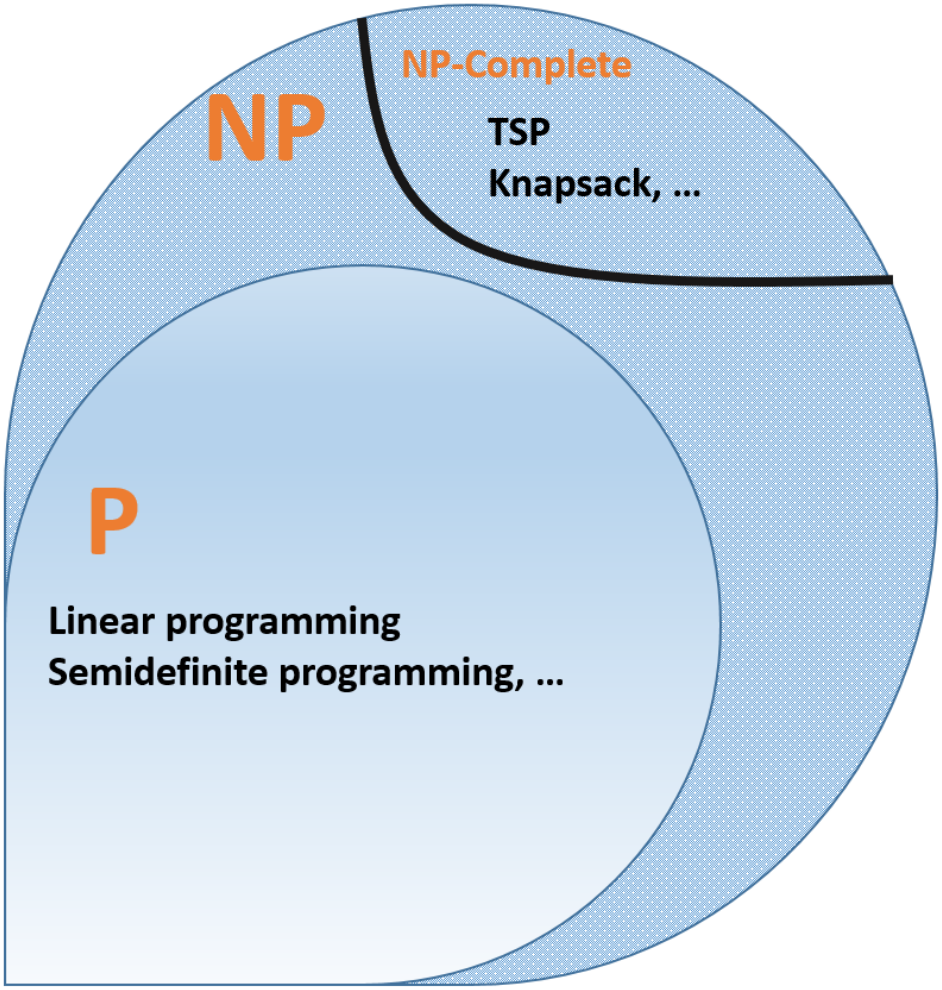




Computational Complexity

NP-completeness

The problem of determining if there exists a TSP tour of length less than k is NP-complete.



Running time growth

▷ Traveling salesman problem: $O(n^2 2^n)$ algorithm by Held and Karp

function	5	10	30	64
n^2	25	100	900	4096
$n^2 \log n$	58.0	332.2	4,416.2	24,576
2^n	32	1024	1,073,741,824	18,446,744,073,709,551,616
1.1^n	1.6	2.6	17.4	445.8

Important: For real-life applications, the data/problem size are restricted.

Time taken by Pisinger's MINKNAP algorithm on knapsack instances with n items and item weights chosen uniformly at random from $1, \ldots, R$.

	uncorrelated			strongly correlated		
n/R	100	1000	10000	100	1000	10000
100	.002	.002	.002	.002	.002	.076
1000	.002	.002	.003	.019	.078	.172
10000	.004	.005	.010	.050	1.19	25.2

Formulations

0-1 Knapsack formulations

Profits p_i and weights w_i are assumed to nonnegative

integer program:

$$\begin{array}{ll} \text{Maximize} & p_1x_1 + p_2x_2 + \ldots + p_nx_n \\ \text{s.t.} & w_1x_1 + w_2x_2 + \ldots w_nx_n \leq c \\ & x_1, x_2, \ldots, x_n \in \{0, 1\}. \end{array}$$

For *unbounded knapsack* replace $\{0, 1\}$ by $\{$integers$\}$ above.

nonlinear integer program:

$$\begin{array}{ll} \text{Maximize} & p_1x_1 + p_2x_2 + \ldots + p_nx_n \\ \text{s.t.} & w_1x_1^2 + w_2x_2^2 + \ldots w_nx_n^2 \leq c \\ & x_1, x_2, \ldots, x_n \in \{0, 1\}. \end{array}$$

0-1 Knapsack relaxations

Maximize $2x_1 + x_2$

s.t. $x_1 + x_2 \leq 1$

$x_1, x_2 \in \{0, 1\}$.

Maximize $2x_1 + x_2$

s.t. $x_1^2 + x_2^2 \leq 1$

$x_1, x_2 \in \{0, 1\}$.

Maximize $2x_1 + x_2$

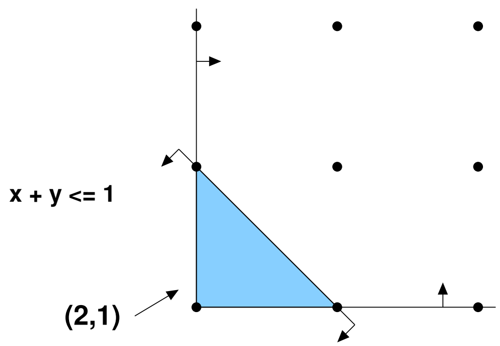
s.t. $x_1 + x_2 \leq 1$

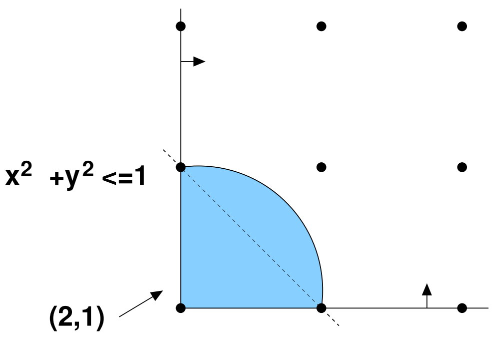
$x_1, x_2 \in [0, 1]$.

Maximize $2x_1 + x_2$

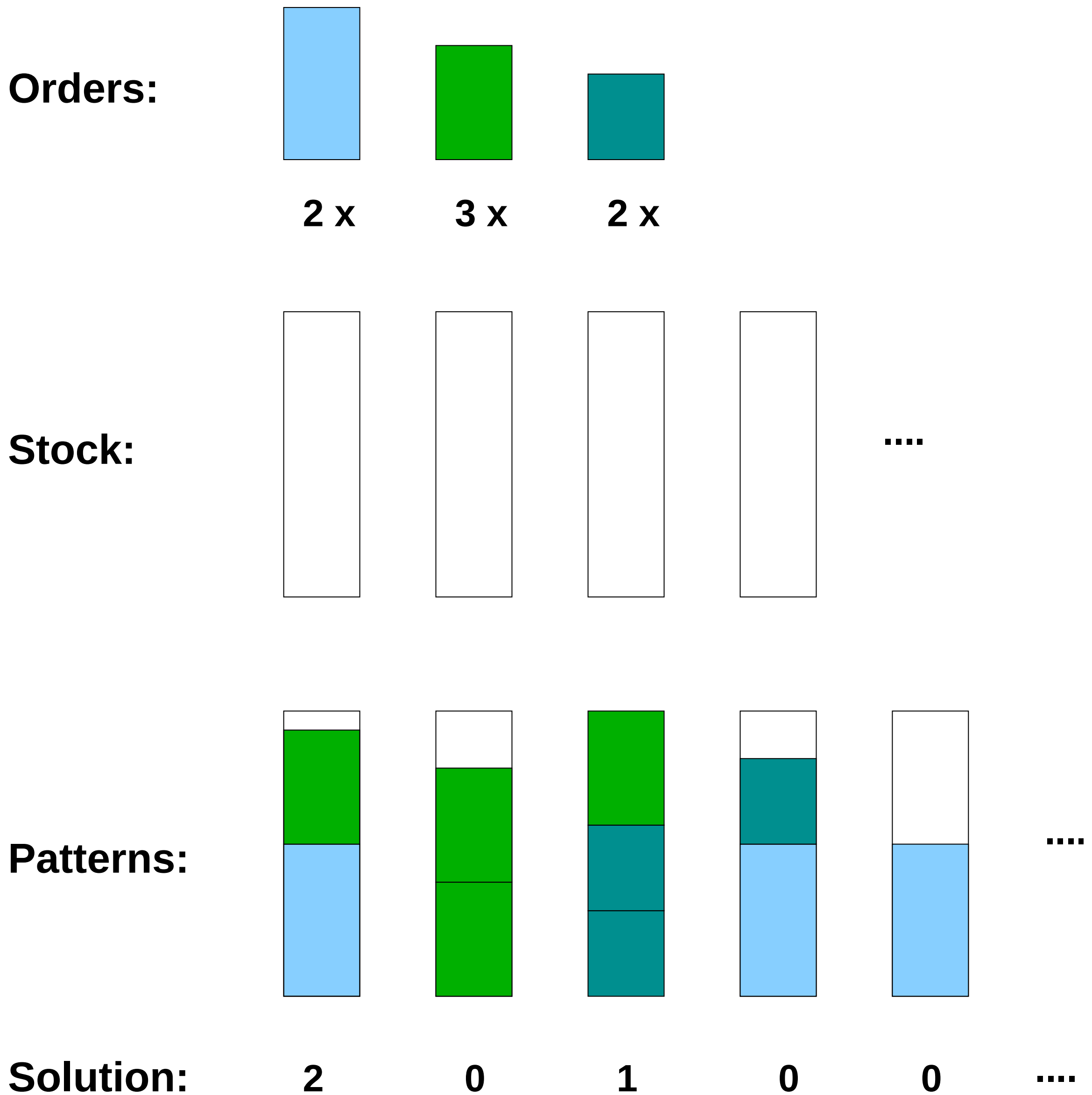
s.t. $x_1^2 + x_2^2 \leq 1$

$x_1, x_2 \in [0, 1]$.





Cutting stock



Cutting stock formulations

Two ways of representing cutting stock solution:

1) Item/stock piece combinations: e.g., 5 copies of ith item are placed in jth stock piece.

$$
\begin{aligned}
\text{Minimize} \quad & y_1 + y_2 + \ldots + y_m \\
\text{s.t.} \quad & l_1 x_{1j} + l_2 x_{2j} + \ldots l_n x_{nj} \le L, \ \text{for } j = 1, \ldots, m \\
& x_{i1} + x_{i2} + x_{im} \ge d_i, \ \text{for } i = 1, \ldots, n \\
& x_{ij} \in \{0, \ldots, d_i\}, \ \text{for } i = 1, \ldots, n, j = 1, \ldots m, \\
& y_1, \ldots, y_m \in \{0, 1\}. \qquad (1)
\end{aligned}
$$

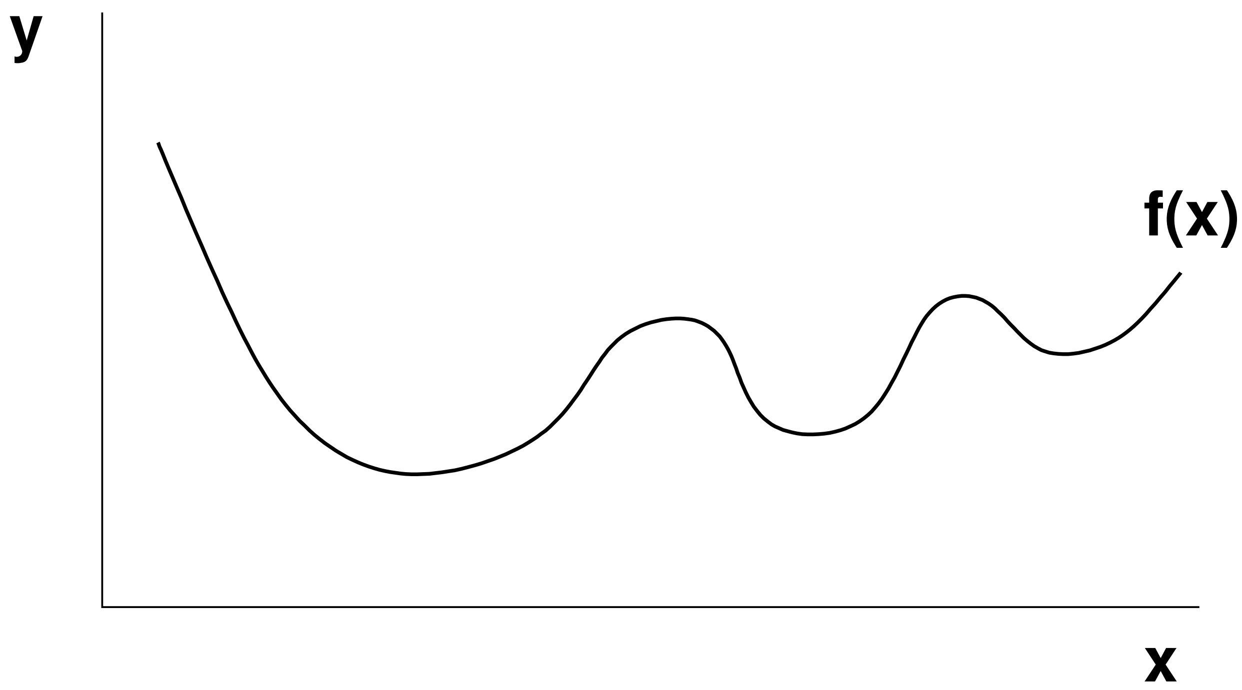
2) Number of copies of each possible "cutting pattern" (Gilmore, Gomory '61).

$$
\begin{aligned}
\text{Minimize} \quad & x_1 + x_2 + \ldots \\
\text{s.t.} \quad & a_{i1} x_1 + a_{i2} x_2 + \ldots \ge d_i, \ \text{for } i = 1, \ldots, n, \\
& x_1, x_2 \ldots \ge 0 \text{ and integral.}
\end{aligned}
$$

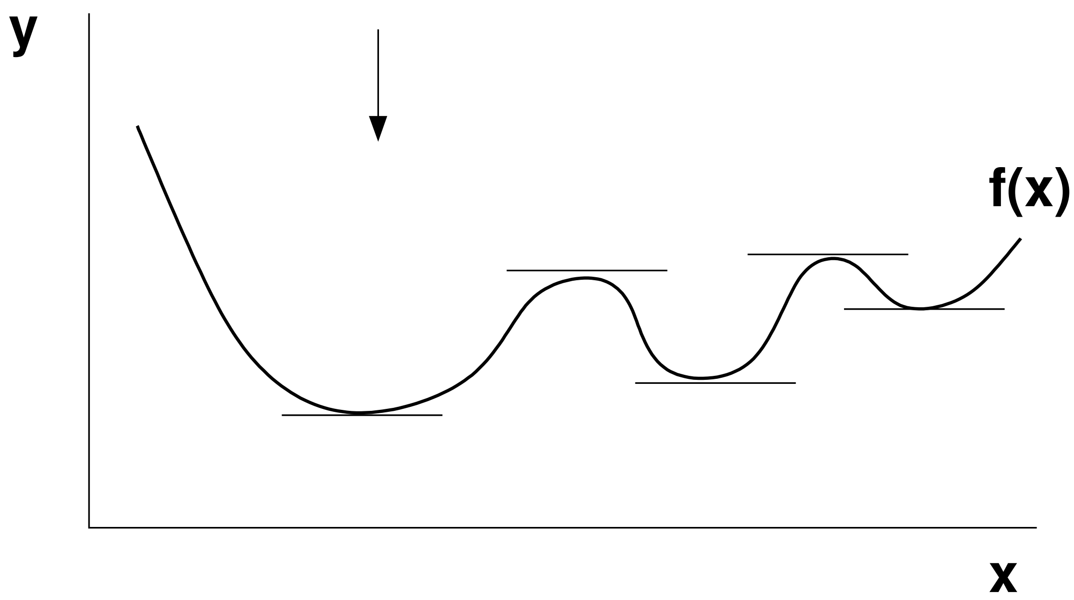
Solution techniques

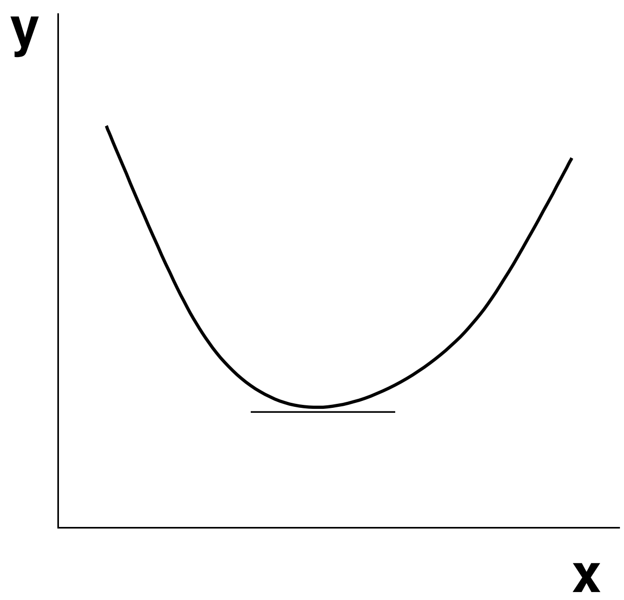
Basic optimization

Minimize $f(x)$ for x in some domain



Optimality conditions





Necessary condition for optimality of x is $f'(x) = 0$. $f''(x) > 0$ is sufficient condition for local optimality. For convex functions, first condition is sufficient.

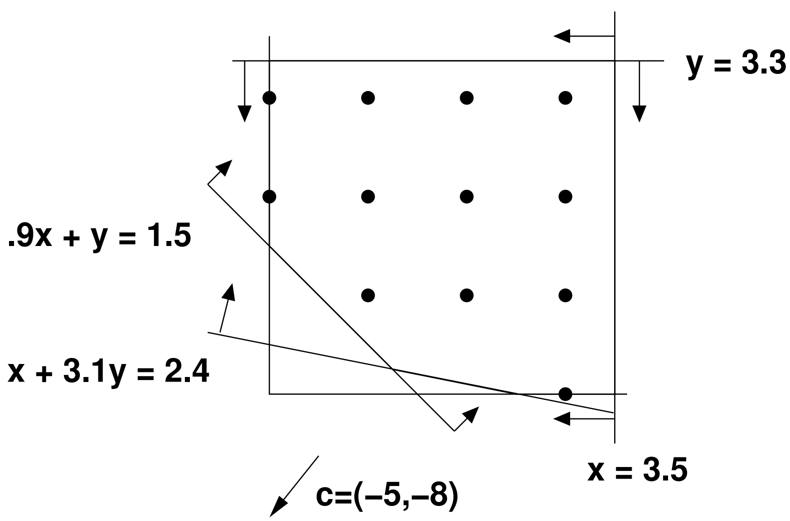
For constrained optimization, KKT conditions are necessary (Kuhn, Tucker '54, Karush '39).

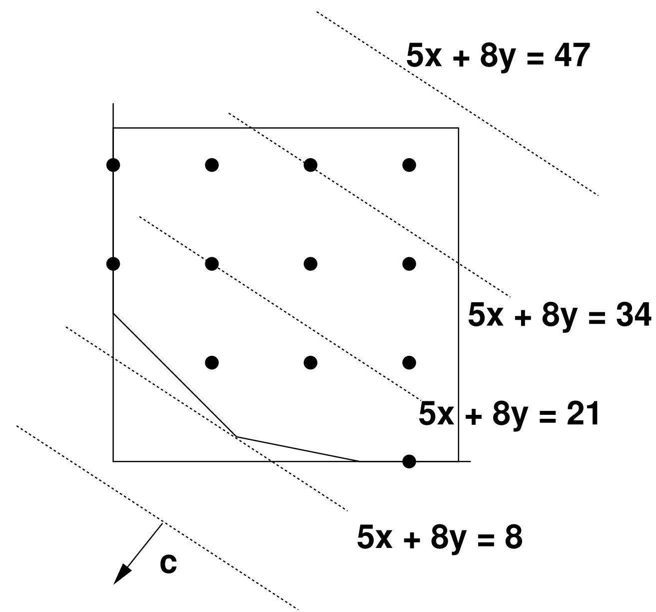
Integer programming

$$\min 5x + 8y \text{ subject to}$$

$$.9x + y \geq 1.5, \;\; x + 3.1y \geq 2.4$$

$$0 \leq x \leq 3.5, \;\; 0 \leq y \leq 3.3, \;\; x, y \;\; \text{integral}$$



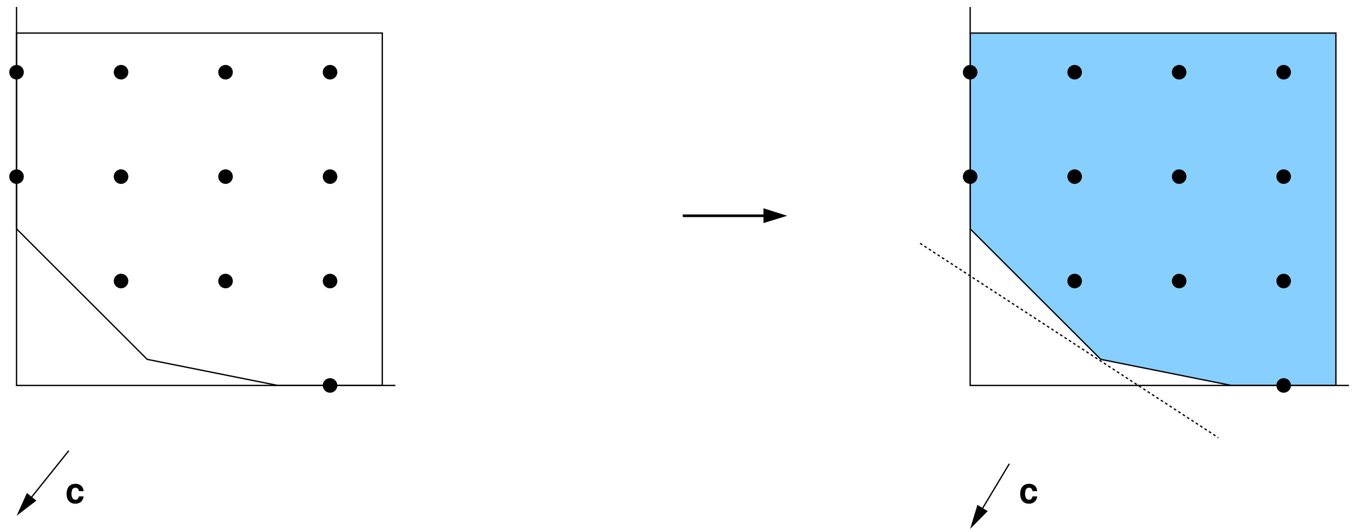


LP relaxation

$$\min 5x + 8y \text{ subject to}$$

$$.9x + y \geq 1.5,\ \ x + 3.1y \geq 2.4$$

$$0 \leq x \leq 3.5,\ \ 0 \leq y \leq 3.3$$



LP relaxation + branching

$\min 5x + 8y$ subject to

$.9x + y \geq 1.5, \; x + 3.1y \geq 2.4$

$0 \leq x \leq 3.5, \; 0 \leq y \leq 3.3$

$\min 5x + 8y$ subject to

$.9x + y \geq 1.5, \; x + 3.1y \geq 2.4$

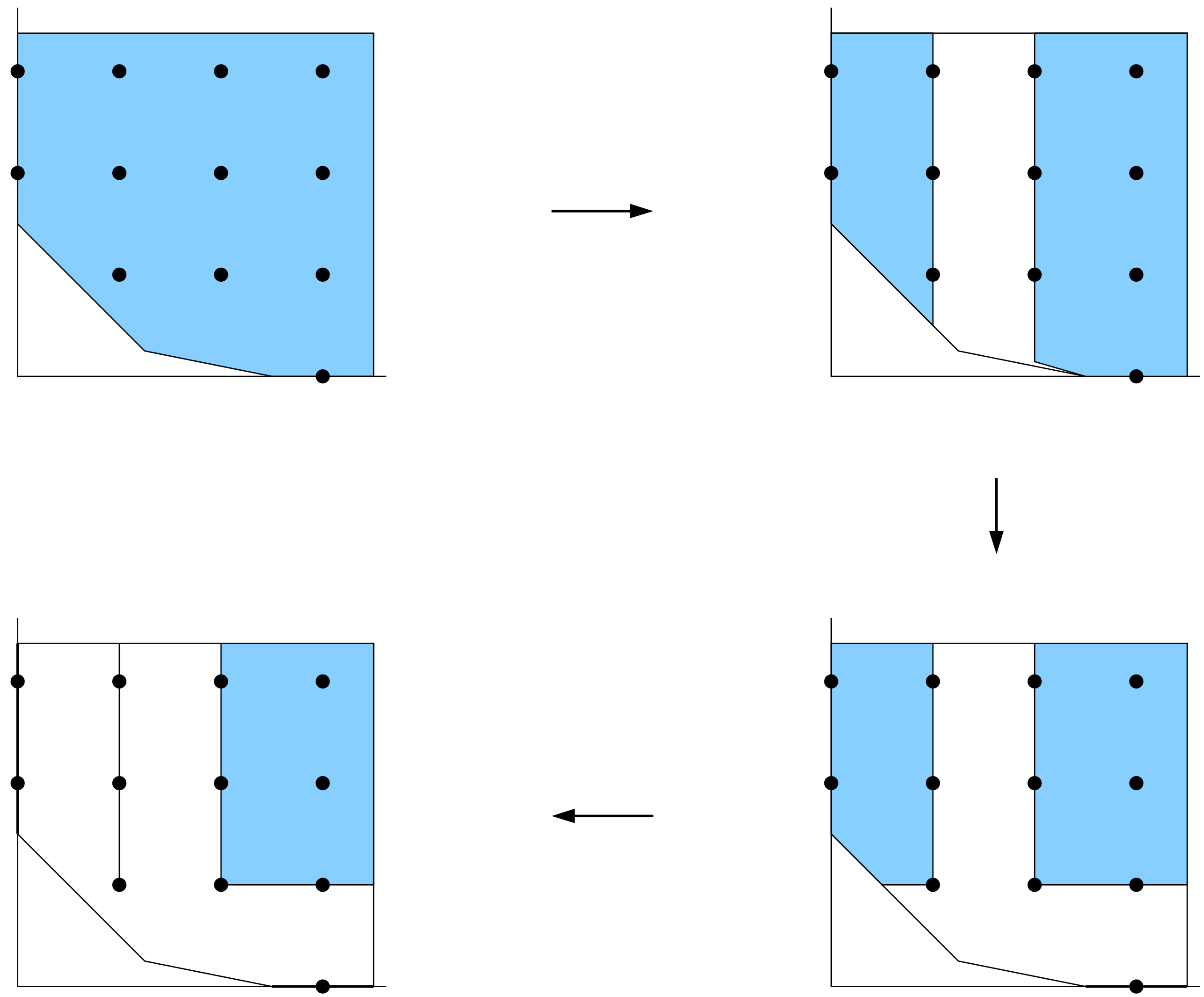
$0 \leq x \leq 1, \; 0 \leq y \leq 3.3$

$\min 5x + 8y$ subject to

$.9x + y \geq 1.5, \; x + 3.1y \geq 2.4$

$2 \leq x \leq 3.5, \; 0 \leq y \leq 3.3$

Branch and bound



cplex-log2.txt

```
Problem 'pp08a' read.
....
Reduced MIP has 133 rows, 234 columns, and 468 nonzeros.
Reduced MIP has 64 binaries, 0 generals, 0 SOSs, and 0 indicators.
....
        Nodes                                         Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Bound    ItCnt     Gap

*     0+    0                       27080.0000                      77     ---
      0     0     2748.3452    51    27080.0000     2748.3452       77   89.85%
*     0+    0                       14300.0000     2748.3452       77   80.78%
*     0+    0                        7950.0000     2748.3452       77   65.43%
      0     2     2748.3452    51     7950.0000     2748.3452       77   65.43%
Elapsed real time =   0.03 sec. (tree size =  0.00 MB, solutions = 3)
*   100+   94                        7860.0000     2848.3452      428   63.76%
*   100+   90                        7640.0000     2848.3452      428   62.72%
   2862  2111     6556.5595    28     7640.0000     3981.3452     9387   47.89%
   6557  5339     6788.4524    21     7640.0000     4254.2976    20447   44.32%
* 10017+ 8320                        7630.0000     4369.3452    30879   42.73%
* 10017+ 8067                        7520.0000     4369.3452    30879   41.90%
* 10017+ 8047                        7510.0000     4369.3452    30879   41.82%
* 10017+ 7947                        7480.0000     4369.3452    30879   41.59%
  10017  7949     7152.1667    16     7480.0000     4369.3452    30879   41.59%
....
 467260 381944     6279.9524    23     7480.0000     5330.2500  1336479   28.74%
Elapsed real time =  76.80 sec. (tree size = 86.82 MB, solutions = 9)
 488008 398616     6870.4881    16     7480.0000     5340.1310  1393871   28.61%
 508767 415262     7018.3810    21     7480.0000     5350.3452  1451784   28.47%
 529510 431893     5359.7738    26     7480.0000     5359.7738  1509653   28.35%
 550267 448498     5819.7024    30     7480.0000     5368.3929  1567040   28.23%
 570955 465047     7091.7738    13     7480.0000     5377.4405  1624524   28.11%
....
 760995 616110     6726.4405    24     7480.0000     5445.6548  2152219   27.20%
 778020 629628     6542.1548    30     7480.0000     5451.3214  2199840   27.12%
 794094 642371     6215.4881    25     7480.0000     5456.2024  2244463   27.06%
 811975 656559        cutoff            7480.0000     5461.4405  2294026   26.99%
 829297 670288     6740.9167    28     7480.0000     5466.6786  2342402   26.92%
 846366 683716     6716.6786    22     7480.0000     5471.6786  2389544   26.85%
Elapsed real time = 143.55 sec. (tree size = 155.11 MB, solutions = 9)
```

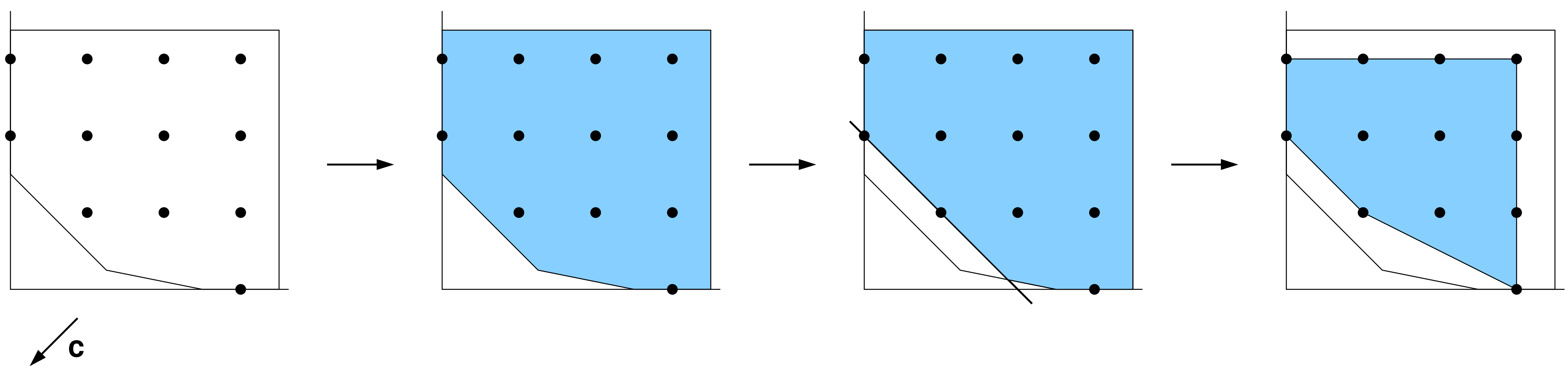
Cutting planes

cutting plane: an inequality satisfied by integral solutions of linear inequalities.

$$\min 5x + 8y \text{ subject to}$$

$$.9x + y \geq 1.5,\ \ x + 3.1y \geq 2.4$$

$$0 \leq x \leq 3.5,\ \ 0 \leq y \leq 3.3,\ \ x, y \ \ \text{integral}$$



Gomory-Chvátal cutting planes (cuts)

$$x \leq 3.5 \Rightarrow x \leq 3$$
$$y \leq 3.3 \Rightarrow y \leq 3$$

$$(.9x + y \geq 1.5) + (.1x \geq 0) \rightarrow$$
$$x + y \geq 1.5 \Rightarrow x + y \geq 2$$

$$(x + y \geq 2) \times .6 + (x + 3.1y \geq 2.4) \times .4 \rightarrow$$
$$x + 1.84y \geq 2.16 \rightarrow$$
$$x + 2y \geq 2.16 \Rightarrow x + 2y \geq 3.$$

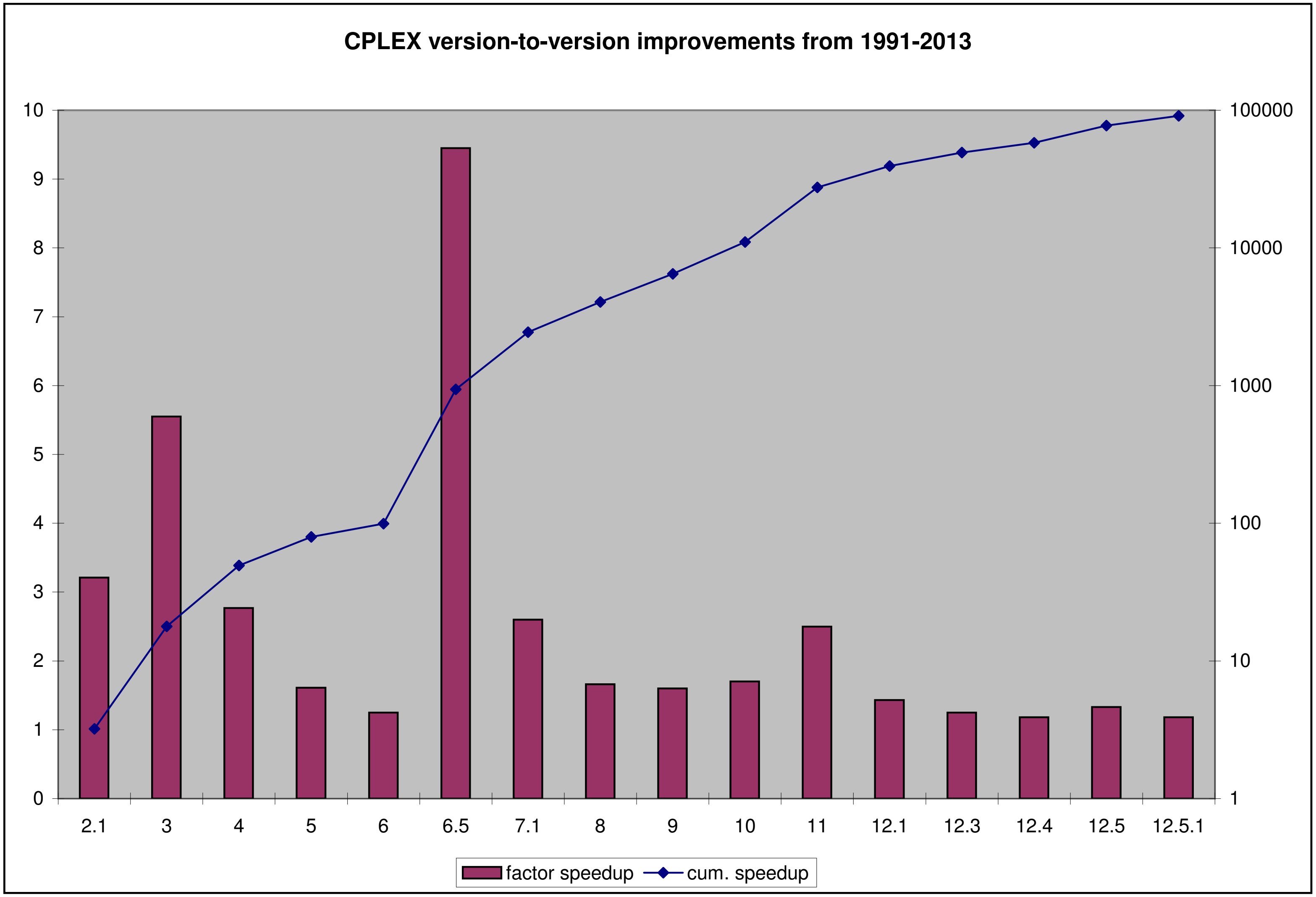
Every integer program can be solved by Gomory-Chvátal cuts (Gomory '60), though it may take exponential time in the worst case (Pudlák '97).

cplex-log.txt

```
Problem 'pp08a' read.
....
Reduced MIP has 133 rows, 234 columns, and 468 nonzeros.
Reduced MIP has 64 binaries, 0 generals, 0 SOSs, and 0 indicators.
....
        Nodes                                         Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Bound    ItCnt     Gap

*     0+    0                      27080.0000                     77      ---
      0     0     2748.3452    51  27080.0000     2748.3452       77   89.85%
*     0+    0                      14300.0000     2748.3452       77   80.78%
      0     0     5046.0422    48  14300.0000     Cuts: 133      153   64.71%
      0     0     6749.5837    24  14300.0000     Cuts: 130      265   52.80%
*     0+    0                      10650.0000     6749.5837      265   36.62%
      0     0     7099.1233    27  10650.0000      Cuts: 53      327   33.34%
      0     0     7171.1837    28  10650.0000      Cuts: 35      356   32.66%
*     0+    0                       7540.0000     7171.1837      356    4.89%
      0     0     7176.2716    31   7540.0000      Cuts: 19      370    4.82%
      0     0     7187.8155    33   7540.0000      Cuts: 20      388    4.67%
      0     0     7188.4198    28   7540.0000       Cuts: 4      398    4.66%
      0     0     7189.5182    30   7540.0000       Cuts: 9      409    4.65%
      0     0     7189.5877    30   7540.0000   Flowcuts: 5      413    4.65%
      0     0     7189.9535    26   7540.0000   Flowcuts: 2      420    4.64%
      0     2     7189.9535    26   7540.0000     7190.0161      420    4.64%
Elapsed real time =   0.27 sec. (tree size =  0.00 MB, solutions = 4)
*    50+   40                       7530.0000     7218.8496     1733    4.13%
*    55    44      integral     0   7520.0000     7218.8496     1783    4.00%
*    60+   45                       7490.0000     7218.8496     1892    3.62%
*    60+   38                       7420.0000     7218.8496     1892    2.71%
*   110+   53                       7400.0000     7238.6753     2712    2.18%
*   210    64      integral     0   7350.0000     7255.3139     4760    1.29%

Implied bound cuts applied:  1
Flow cuts applied:  149
Flow path cuts applied:  23
Multi commodity flow cuts applied:  5
Gomory fractional cuts applied:  34
....
Total (root+branch&cut) =    0.95 sec.
```


CPLEX version-to-version improvements from 1991-2013
0
1
2
3
4
5
6
7
8
9
10
1
10
100
1000
10000
100000
2.1
3
4
5
6
6.5
7.1
8
9
10
11
12.1
12.3
12.4
12.5
12.5.1
factor speedup
cum. speedup

Applications

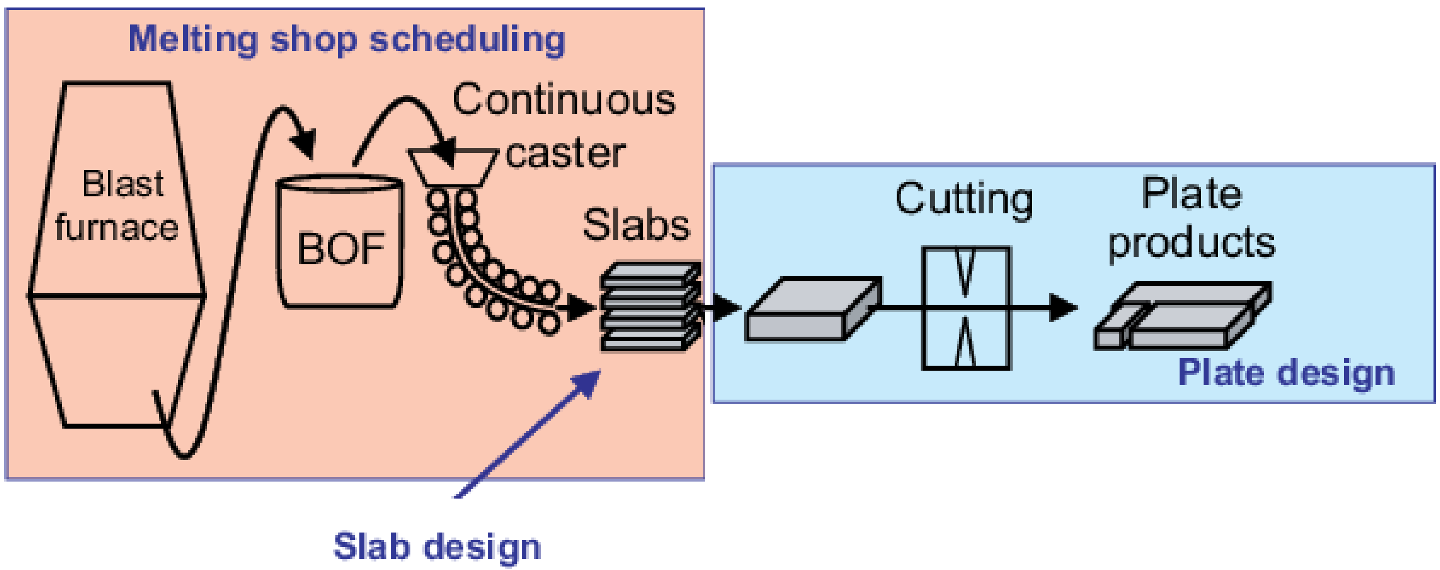
Steel industry application

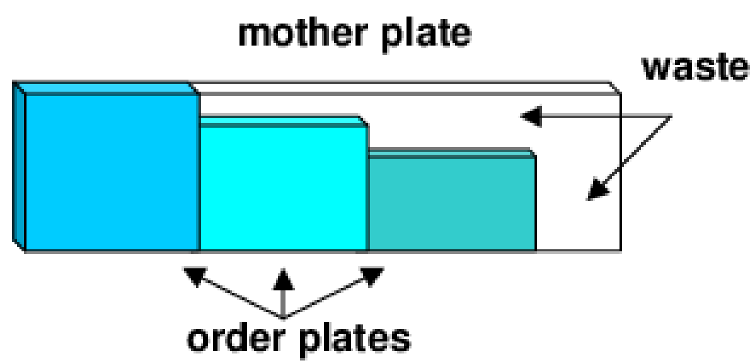
Context: Large steel plant (3 million tons of plates/year $\approx$ 10,000 tons/day) in East Asia moving from a producer-centric model to a customer-centric model

Goal: Optimization tool to generate a production design – a detailed desciption of production steps and related intermediate products

Timeline: 1.5 years
(5 man years on optimization, 25 man years on databases/GUI/analysis)
(joint work with J. Kalagnanam, C. Reddy, M. Trumbo)

Manufacturing process





Consulting Issues

◇ 2+ research man years spent defining problem (high complexity)
- Very large number of constraints including objectives masked as constraints
- 500+ pages of specifications: scope of problem not known at contract signing

◇ High level problem has non-linearities

◇ Software/data issues - 1000+ files

◇ 30 minutes of computing time allowed
- We create 100+ candidate casts = 100+ complex cutting stock problems with up to 2000 orders solved via integer programming column generation

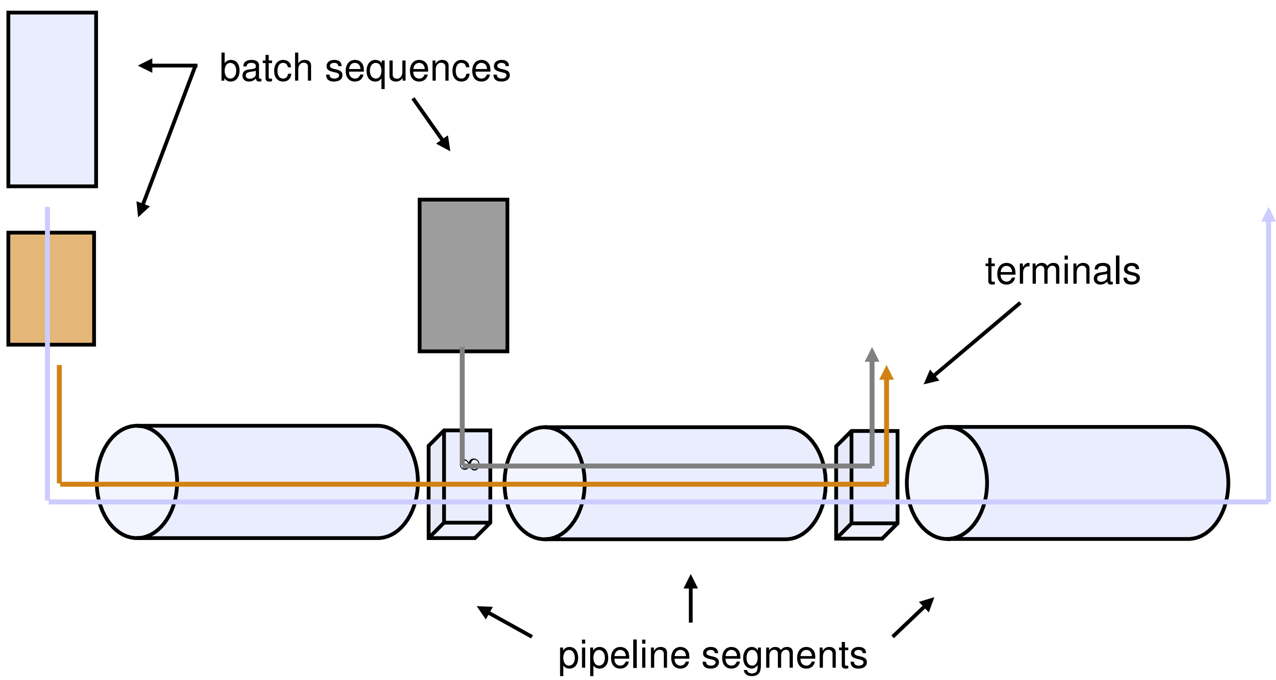
Pipeline management

Schedule injections of batches of oil on a pipeline network while minimizing interface costs, delays, and power costs and satisfying tank constraints

(joint work with V. Austel, O. Günlük, P. Rimshnick, B. Schieber)

A pipeline network has many pipelines, each with multiple segments, each of which can run at multiple 'natural rates'.

Inputs to Batch Sequencing Problem



Interface cost table

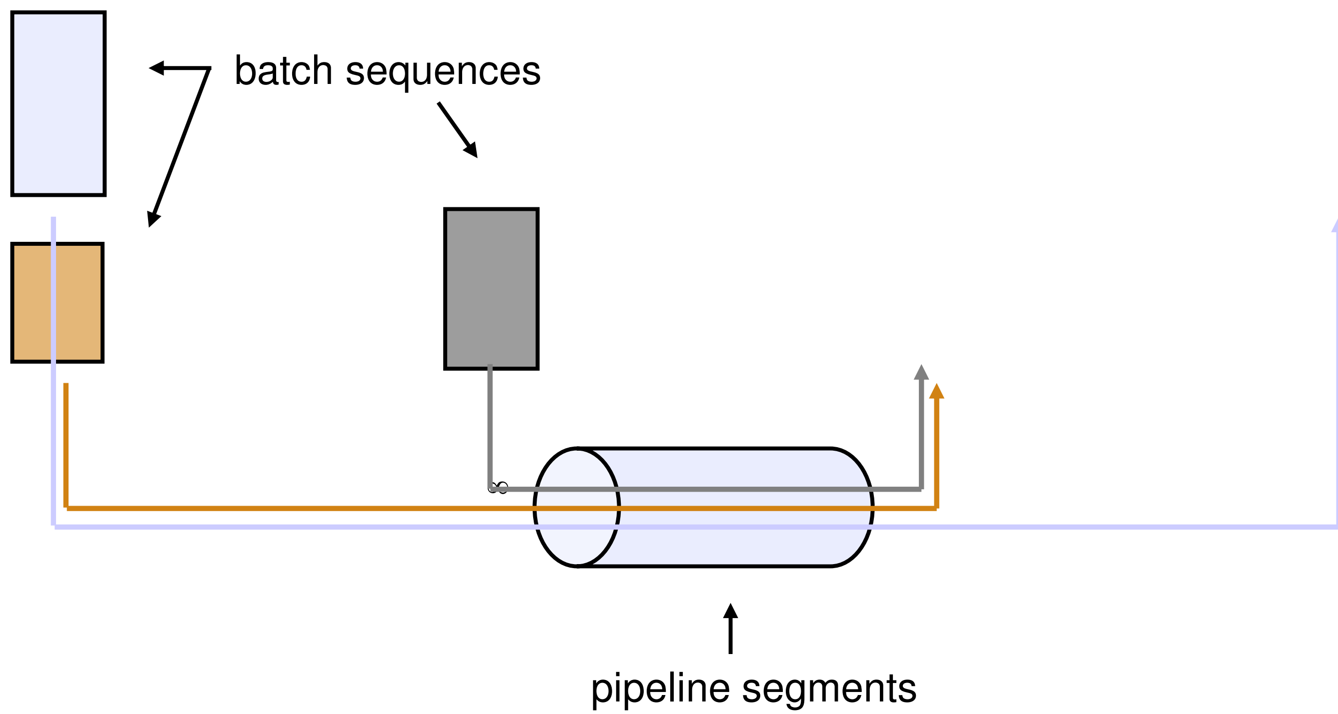
	0	0.9	∞
	1.5	0	2.5
	2.7		0

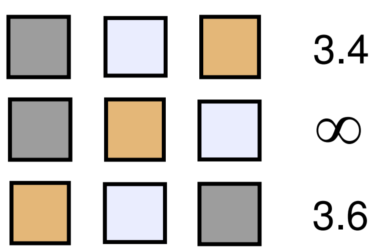
Batch sequencing

When the pipeline consists of single segment, the cost of a batch sequence depends only on interface costs of adjacent batch pairs: batch sequencing reduces to the Asymmetric TSP problem.

	0	1	2	3	4	5	6	7	8	9
0 Chicago	0									
1 Erie	449	0								
2 Chattanooga	618	787	0							
3 Kansas City	504	937	722	0						
4 Lincoln	529	1004	950	219	0					
5 Wichita	805	1132	842	195	256	0				
⋮										

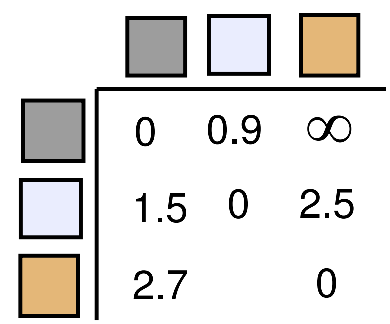
Batch Sequencing at one terminal



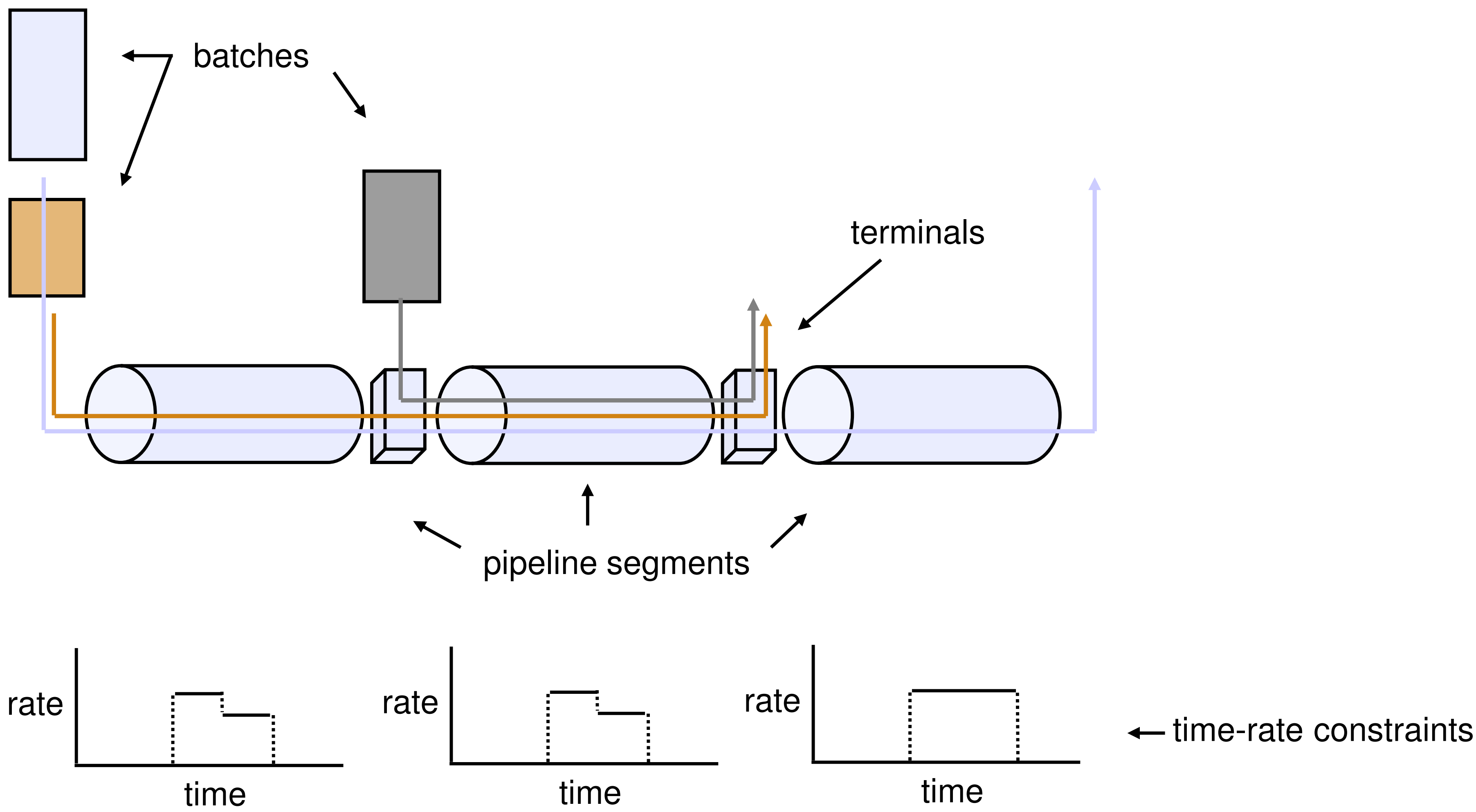


Model as TSP

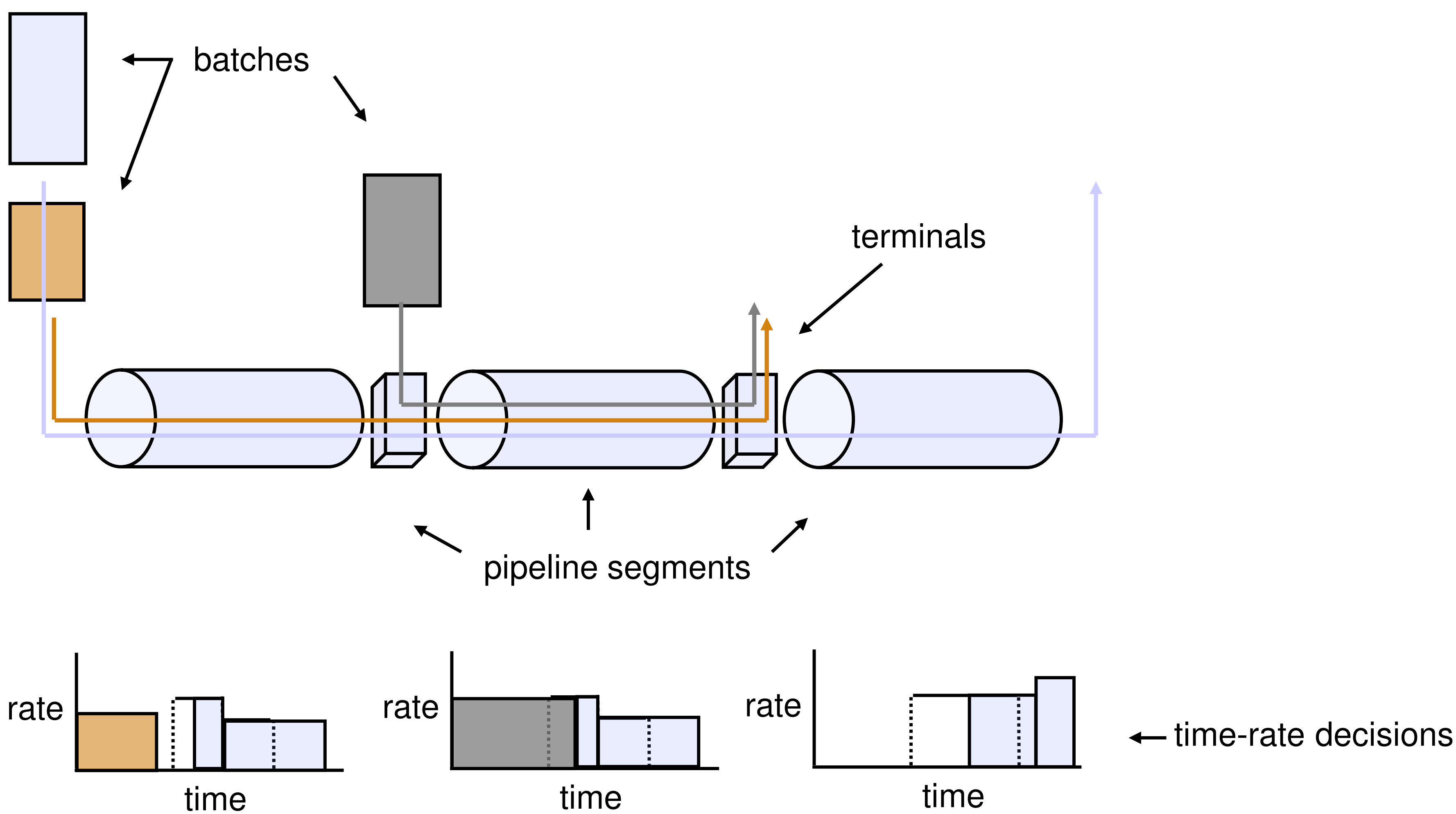
Interface cost table



Inputs to Flow Rate Optimization Problem



Output of Flow Rate Optimization Problem



Flow Rate Optimization

Let there be n possible flow rates.
Let $x_1, \ldots, x_k$ be variables indicating the rate for batches $1, \ldots, k$.
Let $c(k, x_k)$ be the cost of pumping batch k at rate x_k,
and let the time taken be $t(k, x_k)$.

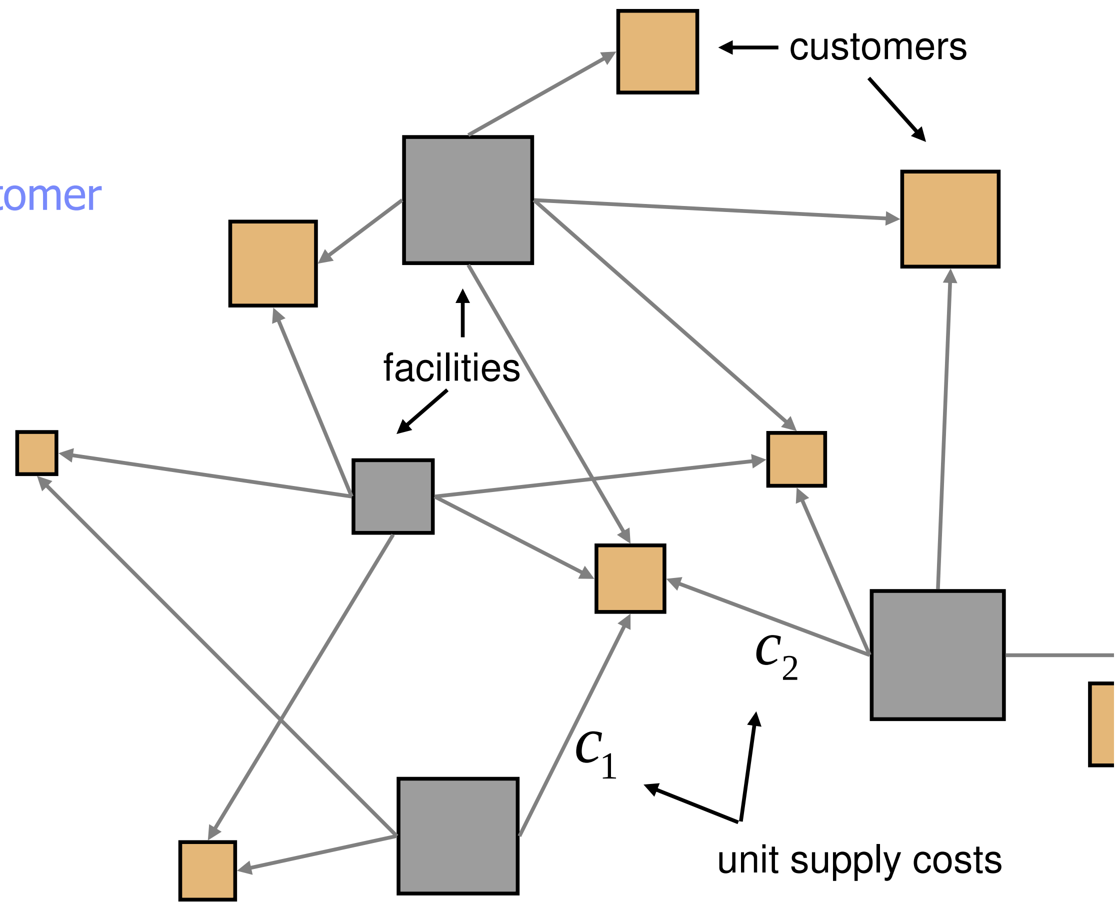
If a pipeline has a single segment, then the problem of minimizing pumping costs while pumping in all batches by a given deadline d is:

$$\begin{array}{ll} \text{Minimize} & c(1, x_1) + c(2, x_2) + \ldots + c(k, x_k) \\ s.t. & t(1, x_1) + t(2, x_2) + \ldots + t(k, x_k) \leq d \\ & x_1, x_2, \ldots, x_k \in \{1, 2, \ldots, n\} \end{array}$$

Facility location problem

Inputs:
- Facilities + capacities
- Cost of opening each facility
- Customers + demands
- Unit cost of supplying a customer from each facility

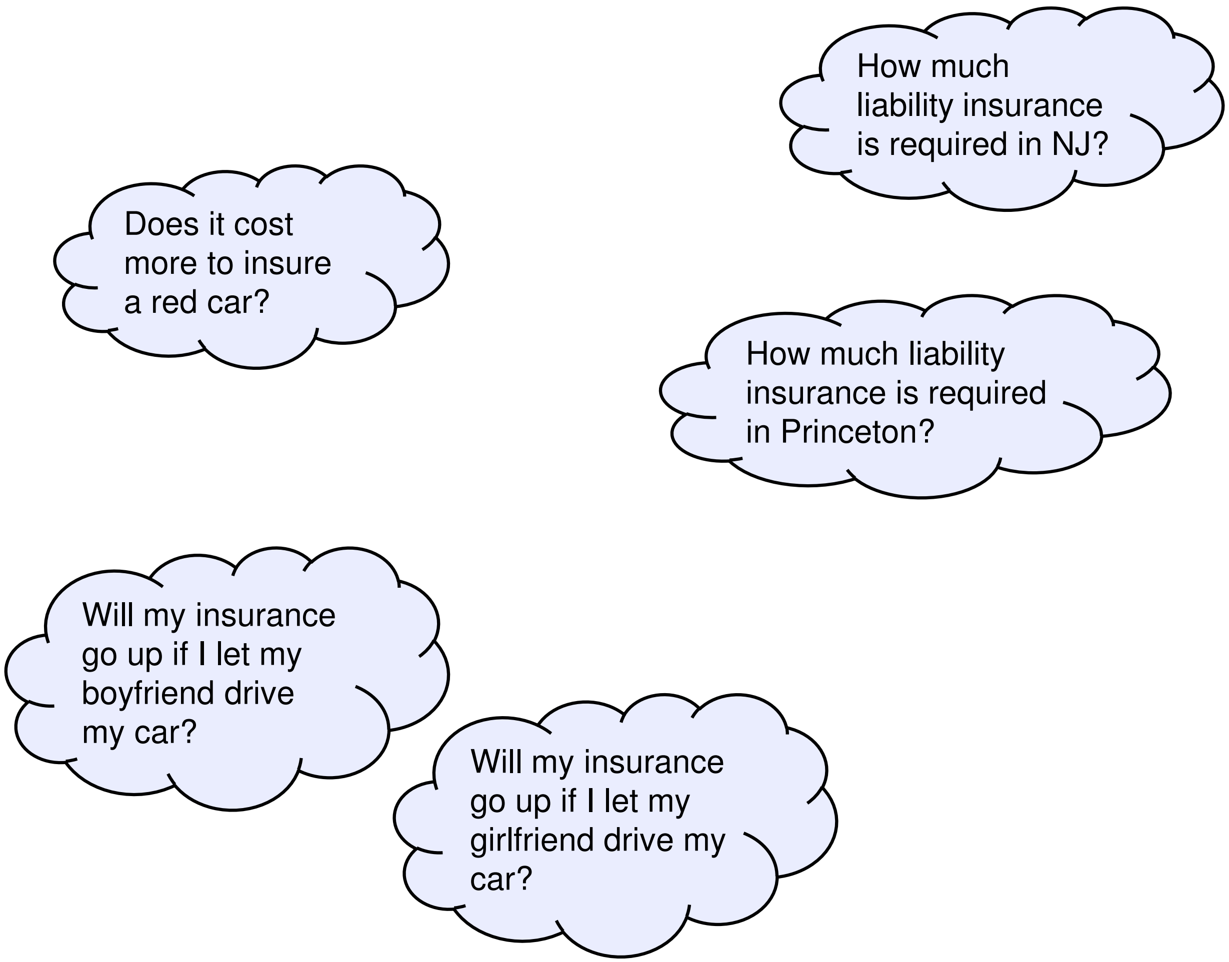
Goal:
- Minimize cost of opening facilities to satisfy demand



Related to Fermat-Weber problem

Machine learning application

▷ Insurance company wants to answer a long list of customer questions, but has a budget for only 500 answers (Dmitry Malioutov).



▷ The problem is an “active learning” problem: try to optimize which questions to answer.

▷ Balance “information gain” vs. “diversity” for each answered question.

Inputs:

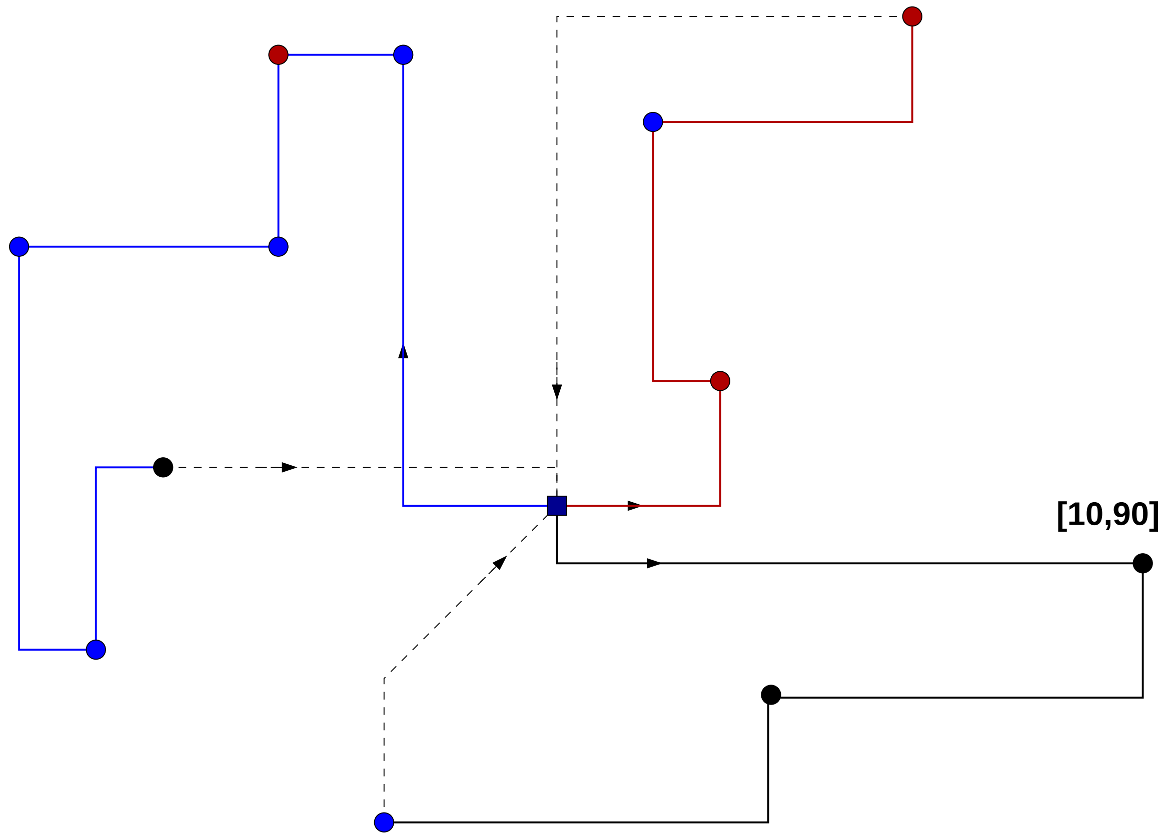
1) Each node/question has a notion of how much additional information it will add by providing a human answer – this is the node cost.

2) The similarity of each question to other questions: there is no point in answering the same question 20 times, so it’s great to have a diverse set of questions to ask humans to answer.

Vehicle routing application

Context: Food distribution company in North America trying to improve delivery to customers within desired time windows, while minimizing travel costs.

VRPTW with driver preferences

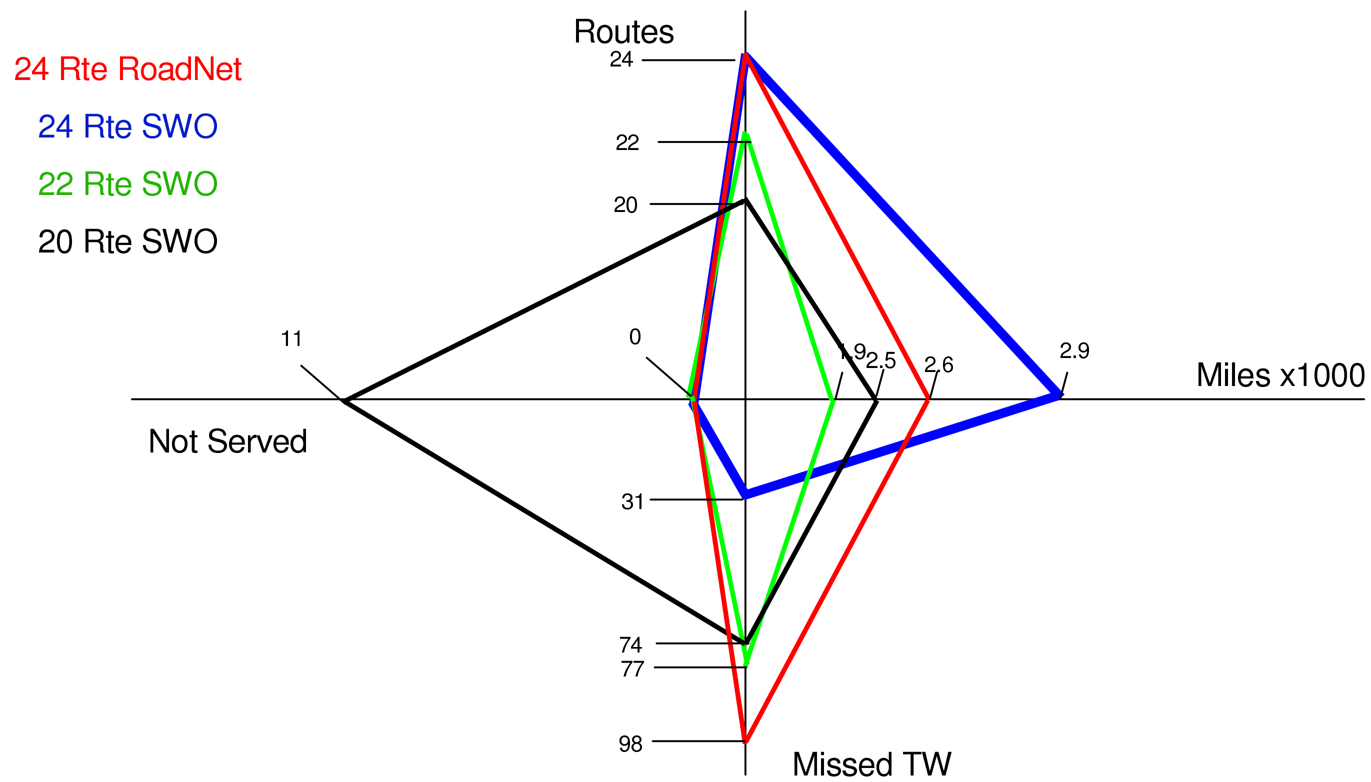


Customers have preferred drivers; penalize for delivery by non-preferred driver.

◇ 200-300 customers, 20-30 routes per shift, 3-6 shifts per day
◇ Create preference relationships between ≈ 200 drivers and 1000 customers

(joint work with O. Günlük, G. Sorkin)

Graphic Route Comparison



Comparison of route characteristics – Changing Input Parameters and Penalties directly impacts optimizer solution.

Conclusions

▷ Many real-life optimization problems can be modeled as instances of NP-hard problems. However, as the data and problem sizes are restricted, such problems can often be solved with customized techniques.

▷ Linear-integer programming is the most widely used optimization tool in practical applications, but some important problems (e.g., portfolio optimization) are modeled as nonlinear (quadratic) integer programs.

▷ Linear constraints are more common in combinatorial problems, whereas nonlinear constraints are more common in systems where the physics is important.