OPTIMIZATION

Numerical methods in chemical engineering

Edwin Zondervan

OVERVIEW

- In this lecture we get introduced to constrained and unconstrained optimization.
- We will use the simplex method to solve linear programming problems (LP)
- We will use the Lagrange multiplier method to solve nonlinear programming problems (NLP's)
- And we will briefly discuss optimal control, using Pontryagin's principle.
- Lastly we will play a little with another optimization platform (AMPL)

WHAT IS OPTIMIZATION?

- Optimization is minimization or maximization of an objective function (also called a performance index or goal function) that may be subject to certain constraints:

min $f(x)$ = max $-f(x)$

$$\min\ f(x)$$

Goal function

$$s.t.$$

$$g(x) = 0$$

Equality constraints

$$h(x) \geq 0$$

Inequality constraints

(10-1)

OPTIMIZATION SPECTRUM

MATHEMATICAL PROGRAMMING

Problem	Method	Solvers
LP	**Simplex method** Barrier methods	Linprog (Matlab) CPLEX (GAMS, AIMMS, AMPL, OPB)
NLP QP	**Lagrange multiplier method** Successive linear programming Quadratic programming	Fminsearch/fmincon (Matlab) MINOS (GAMS, AMPL) CONOPT (GAMS)
MIP MILP MINLP MIQP	**Branch and bound** Dynamic programming Generalized Benders Decomposition Outer Approximation method Disjunctive programming	Bintprog (Matlab) DICOPT (GAMS) BARON (GAMS)

META HEURISTICS

Neural networks, fuzzy modeling, genetic algorithms, expert systems, etc.

ADVANCED TOPICS

Constraint programming, stochastic programming, multi-objective programming, etc.

FACTORS OF CONCERN

- Continuity of the functions
- Convexity of the functions
- Global versus local optima
- Constrained versus unconstrained optima

LINEAR PROGRAMMING

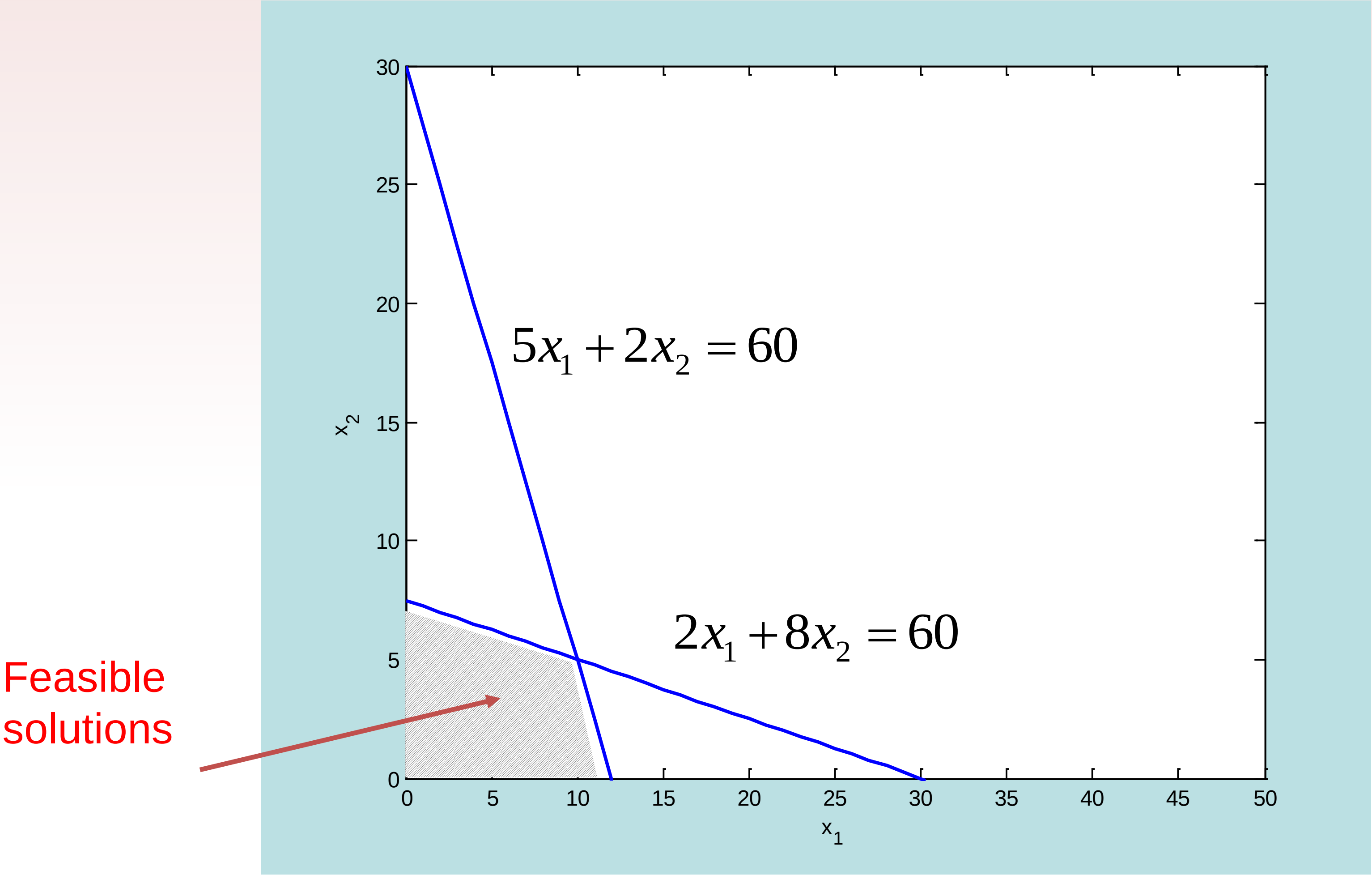
- In linear programming the objective function and the constraints are **linear functions**!
- For example:

$$
\begin{aligned}
&\max z = f(x_1, x_2) = 40x_1 + 88x_2 \\
&s.t. \\
&2x_1 + 8x_2 \leq 60 \\
&5x_1 + 2x_2 \leq 60 \\
&x_1 \geq 0 \\
&x_2 \geq 0
\end{aligned}
\qquad \textbf{(10-2)}
$$

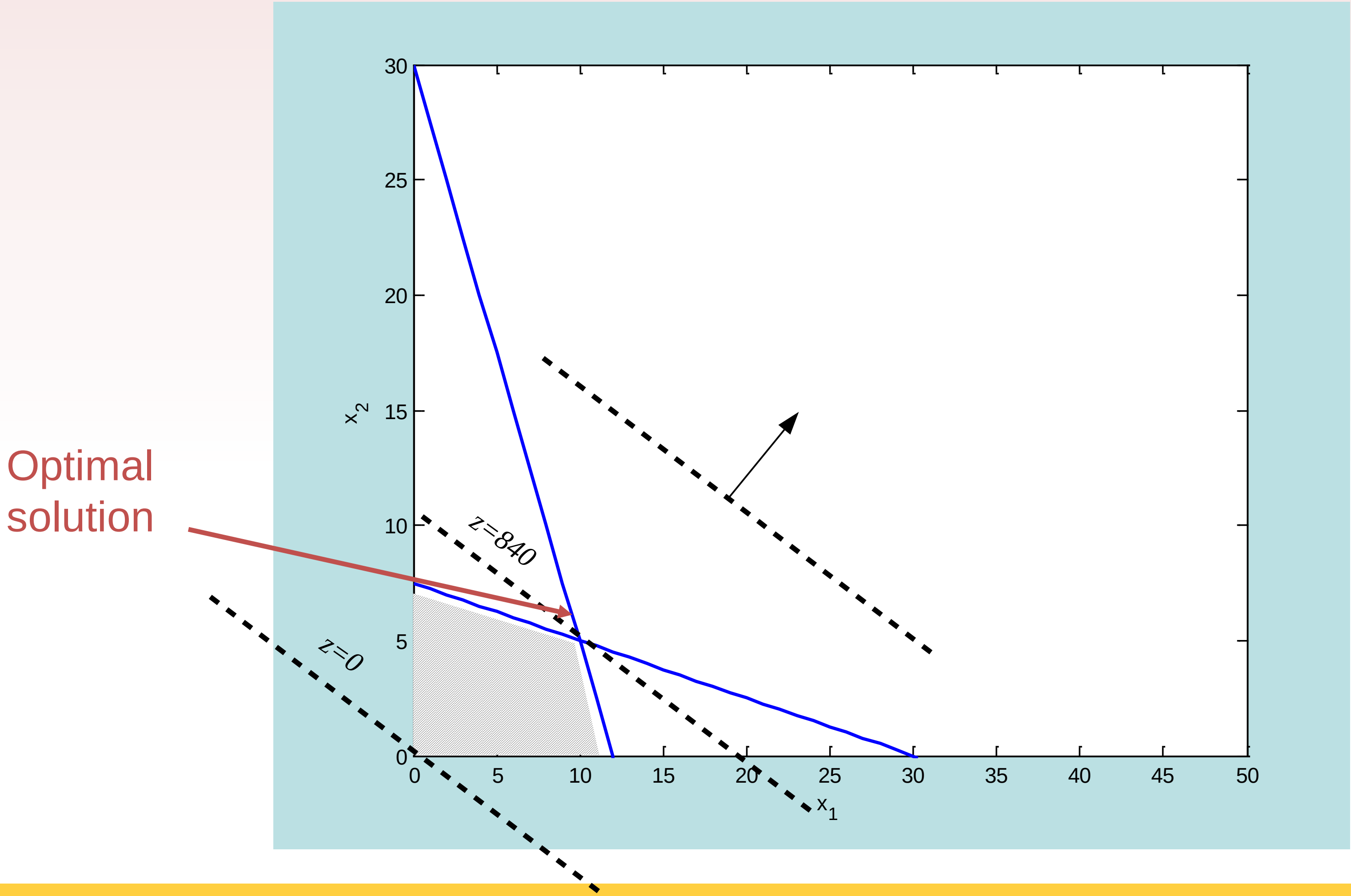
If the constraints are satisfied, but the objective function is not maximized/minimized we speak of a **feasible solution**.

If also the objective function is maximized/minimized, we speak of an **optimal solution**!

PLOTTING THE CONSTRAINTS



PLOTTING THE OBJECTIVE FUNCTION



NORMAL FORM OF AN LP PROBLEM

NORMAL FORM OF THE LP PROBLEM

$$\begin{aligned}
&\max z = f(x_1, x_2) = 40x_1 + 80x_2 \\
&s.t. \\
&2x_1 + 8x_2 \le 60 \\
&5x_1 + 2x_2 \le 60 \\
&x_1 \ge 0 \\
&x_2 \ge 0
\end{aligned}$$

$$\begin{aligned}
&\max f(x) = 40x_1 + 88x_2 \\
&s.t. \\
&2x_1 + 8x_2 + x_3 = 60 \\
&5x_1 + 2x_2 + x_4 = 60 \\
&x_i \ge 0 \{i = 1,\dots,4\}
\end{aligned} \tag{10-3}$$

x_3 and x_4 are called **slack variables**, they are non auxiliary variables introduced for the purpose of converting inequalities in to equalities

THE SIMPLEX METHOD

- We can formulate our earlier example to the normal form and consider it as the following **augmented matrix**:

$$
\begin{array}{cc}
 & \begin{array}{cccccc} \;z\; & \;x_1\; & \;x_2\; & \;x_3\; & \;x_4\; & \;b\; \end{array} \\
T_0 = & \left[\begin{array}{c:c:c:c:c:c}
1 & -40 & -88 & 0 & 0 & 0 \\ \hdashline
0 & 2 & 8 & 1 & 0 & 60 \\ \hdashline
0 & 5 & 2 & 0 & 1 & 60
\end{array}\right]
\end{array}
\qquad \textbf{(10-4)}
$$

This matrix is called the (initial) simplex table

Each simplex table has two kinds of variables, the **basic variables** (columns having only one nonzero entry) and the **nonbasic variables**

THE SIMPLEX METHOD

$$
T_0 = \begin{array}{c}
\begin{array}{cccccc} z & x_1 & x_2 & x_3 & x_4 & b \end{array} \\
\left[\begin{array}{c|c|c|c|c|c}
1 & -40 & -88 & 0 & 0 & 0 \\ \hline
0 & 2 & 8 & 1 & 0 & 60 \\ \hline
0 & 5 & 2 & 0 & 1 & 60
\end{array}\right]
\end{array}
$$

- Every simplex table has a **feasible solution**. It can be obtained by setting the nonbasic variables to zero: $x_1 = 0$, $x_2 = 0$, $x_3 = 60/1$, $x_4 = 60/1$, $z = 0$

THE OPTIMAL SOLUTION?

- The optimal solution is now obtained stepwise by pivoting in such way that z reaches a maximum.
- The big question is, how to choose your pivot equation …

STEP 1: SELECTION OF THE PIVOT COLUMN

- Select as the column of the pivot, the first column with a negative entry in Row 1. In our example, that's column 2 (-40)

$$
\begin{array}{c}
\begin{array}{cccccc} z & x_1 & x_2 & x_3 & x_4 & b \end{array} \\
T_0 = \left[\begin{array}{c|c|c|c|c|c}
1 & -40 & -88 & 0 & 0 & 0 \\
\hline
0 & 2 & 8 & 1 & 0 & 60 \\
\hline
0 & 5 & 2 & 0 & 1 & 60
\end{array}\right]
\end{array}
\qquad \textbf{(10-5)}
$$

STEP 2: SELECTION OF THE PIVOT ROW

- Divide the right sides by the corresponding column entries of the selected pivot column. In our example that is 60/2 = 30 and 60/5 = 12.

$$
\begin{array}{c c}
 & \begin{array}{cccccc} z & x_1 & x_2 & x_3 & x_4 & b \end{array} \\
T_0 = & \left[\begin{array}{c:c:c:c:c:c} 1 & -40 & -88 & 0 & 0 & 0 \\ \hdashline 0 & 2 & 8 & 1 & 0 & 60 \\ 0 & 5 & 2 & 0 & 1 & 60 \end{array}\right] \begin{array}{l} \\ \\ \text{Pivot eqn.} \end{array}
\end{array}
\qquad \textbf{(10-6)}
$$

- Take as the pivot equation the equation that gives the smalles quotient, so 60/5

STEP 3: ELIMINATION BY ROW OPERATIONS

$$
\begin{array}{cc}
 & \begin{array}{cccccc} z & x_1 & x_2 & x_3 & x_4 & b \end{array} \\
T_1 = & \left[\begin{array}{c:c:c:c:c:c} 1 & 0 & -72 & 0 & 8 & 480 \\ \hdashline 0 & 0 & 7.2 & 1 & -0.4 & 36 \\ \hdashline 0 & 5 & 2 & 0 & 1 & 60 \end{array}\right]
\end{array}
\begin{array}{l} \text{Row 1 + 8*Row 3} \\ \text{Row 2 + 0.4*Row 3} \\ \textbf{(10-7)} \end{array}
$$

- The basic variables are now x_1, x_3 and the nonbasic variables are x_2, x_4. Setting the nonbasic variables to zero will give a new feasible solution: $x_1 = 60/5$, $x_2 = 0$, $x_3 = 36/1$, $x_4 = 0$, $z = 480$

THE SIMPLEX METHOD

- We moved from z = 0 to z = 480. The reason for the increase is because we eliminated a negative term from the eqation, so: elimination should only be applied to negative entries in Row 1, but no others.
- Although we found a feasible solution, we did not find the optimal solution yet (the entry of -72 in our simplex table) → so we repeat step 1 to 3.

THE SECOND ITERATION

- Step 1: select column 3
- Step 2: 36/7.2 = 5 and 60/2 = 30 → select 7.2 as the pivot
- Elimination by row operations:

$$T_2 = \begin{array}{c} \left[\begin{array}{cccccc} 1 & 0 & 0 & 10 & 4 & 840 \\ 0 & 0 & 7.2 & 1 & -0.4 & 36 \\ 0 & 5 & 0 & -1/36 & 1/0.9 & 50 \end{array}\right] \end{array} \begin{array}{l} \text{Row 1 + 10*Row 2} \\ \textbf{(10-8)} \\ \text{Row 3 – (2/7.2)*Row 2} \end{array}$$

- The basic feasible solution: x_1= 50/5, x_2 = 36/7.2, x_3 = 0, x_4 = 0, z =840 (no more negative entries: so this solution is also the **optimal solution**)

USING MATLAB FOR LP PROBLEMS

- We are going to solve the following LP problem:

$$\min\ f(x) = -5x_1 - 4x_2 - 6x_3$$

$$s.t.$$

$$x_1 - x_2 + x_3 \leq 20$$

$$3x_1 + 2x_2 + 4x_3 \leq 42$$

$$3x_1 + 2x_2 \leq 30$$

$$0 \leq x_1, 0 \leq x_2, 0 \leq x_3$$

(10-9)

Using the function LINPROG:

```
f = [-5; -4; -6]
A = [1 -1 1 3 2 4 3 2 0];
b = [20; 42; 30];
lb = zeros(3,1);
[x,fval,exitflag,output,lambda]
   = linprog(f,A,b,[],[],lb);
```

Gives:

```
x = 0.00 15.00 3.00
lambda.ineqlin = 0 1.50 0.50

lambda.lower = 1.00 0 0
```

NONLINEAR PROGRAMMING

- In nonlinear programming the objective function and the constraints are **nonlinear functions**!
- For example:

$$\min f(x) = 5x_1^2 + 3x_2^2$$

$$s.t.$$

$$g(x) = 2x_1 + x_2 - 5$$

(10-10)

LAGRANGE MULTIPLIER METHOD

- Consider the general problem:

$$\begin{aligned}&\min\ f(x)\\&s.t.\\&g(x)=0\end{aligned} \qquad \textbf{(10-11)}$$

- A Lagrangian function can be defined as:

$$L(x,v) = f(x) + vg(x) \qquad \textbf{(10-12)}$$

- To find the optimum, differentiate L with respect to x and v and set the equations to zero:

$$\frac{\partial L}{\partial x} = \frac{\partial f}{\partial x} + v\frac{\partial g}{\partial x} = 0,\ \ g(x) = 0 \qquad \textbf{(10-13)}$$

BACK TO THE EXAMPLE

$$\min\ f(x) = 5x_1^2 + 3x_2^2$$

$$s.t.$$

(10-14)

$$g(x) = 2x_1 + x_2 - 5$$

$$L = 5x_1^2 + 3x_2^2 + v(2x_1 + x_2 - 5)$$

$$\left.\begin{aligned} \frac{\partial L}{\partial x_1} &= 10x_1 + 2v = 0 \\ \frac{\partial L}{\partial x_2} &= 6x_2 + v = 0 \\ \frac{\partial L}{\partial v} &= g(x) = 2x_1 + x_2 - 5 = 0 \end{aligned}\right\} v = 150/17, x_1 = 30/17,\ x_2 = 25/17$$

(10-15)

LMM FOR NLPS WITH INEQUALITY CONSTRAINTS

- When the problem has the following shape:

$$
\begin{aligned}
&\min f(x) \\
&s.t. \\
&h_j(x) = 0 \quad \{j = 1, \dots, m\} \\
&g_i(x) \geq 0 \quad \{i = m+1, \dots, p\}
\end{aligned}
\qquad \textbf{(10-16)}
$$

- The Lagrangian function is defined as:

$$L(x,u,v) = f(x) + \sum_{j=1}^{m} v_j h_j(x) + \sum_{j=m+1}^{p} u_j g_j(x) \qquad \textbf{(10-17)}$$

$$\nabla f(x) + \sum_{j=1}^{m} v_j \nabla h_j(x) + \sum_{j=m+1}^{p} u_j \nabla g_j(x) = 0$$

(10-18)

This condition, known as the **Karush-Kuhn-Tucker condition for optimality** should be satisfied.

USING MATLAB FOR NLP PROBLEMS

- We are going to solve the following NLP problem:

$$\min f(x) = -x_1 x_2 x_3$$

$$s.t.$$

$$0 \le x_1 + 2x_2 + 2x_3 \le 72$$

(10-19)

Using the function FMINCON:

```
function f = myfun(x) f = -x(1) * x(2) * x(3);

A=[-1 -2 -2; 1 2 2]; b = [0 72];
x0 = [10; 10; 10];

solution [x,fval] = fmincon(@myfun,x0,A,b)
```

Gives:

```
x = 24.00 12.00 12.00
```

SOME TIPS FOR SOLVING NLPPS

- Avoid nonlinearity if possible
- Better nonlinearities in the objective function than in the constraints
- Better inequalities than equalities
- Supply good starting guesses to a solver
- Don’t blame the solver if you don’t find a solution, take a critical look at the problem formulation

OPTIMAL CONTROL PROBLEMS

- In an Optimal control problem, or **dynamic optimization** problem an objective function is maximized/minimized by finding optimal **trajectories** for the control variables.

$$\max P(t_f) = f(\mathbf{x}, \mathbf{u}, t_f)$$

$$\dot{\mathbf{x}} = g(\mathbf{x}, \mathbf{u}, t)$$

Find the values for **u**(t) that maximize $P(t_f)$

(10-20)

PONTRYAGIN'S PRINCIPLE

- Step 1: Define the Hamiltonian:

$$H = \lambda_1 \dot{x}_1 + \lambda_2 \dot{x}_2 + ..$$

(10-22)

- Step 2: Choose the adjoint variables such that:

$$\frac{d\lambda}{dt} = -\frac{\partial H}{\partial x}$$

(10-23)

- The optimum for u(t) can be found by minimizing the Hamiltonian:

$$\frac{\partial H}{\partial u} = 0$$

(10-24)

http://www.sjsu.edu/faculty/watkins/pontryag.htm

VECTOR PARAMETERIZATION

- A practical approach is by assuming a function for the control variables, e.g. a simple polynomial function:

$$u(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_p t^p \qquad \textbf{(10-25)}$$

- And subsequently determine the values for the parameters $a_0, \ldots, a_p$, that maximizes/minimizes the objective function.

MIP PROBLEMS -B&B METHOD

- An example

$$
\begin{aligned}
&\max z = 8x_1 + 11x_2 + 6x_3 + 4x_4 \\
&s.t. \\
&5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14 \\
&x_j \in \{0,1\},\ j = 1,\ldots,4
\end{aligned}
$$

- Solving the relaxed problem (binary variables are treated as they were continuous:

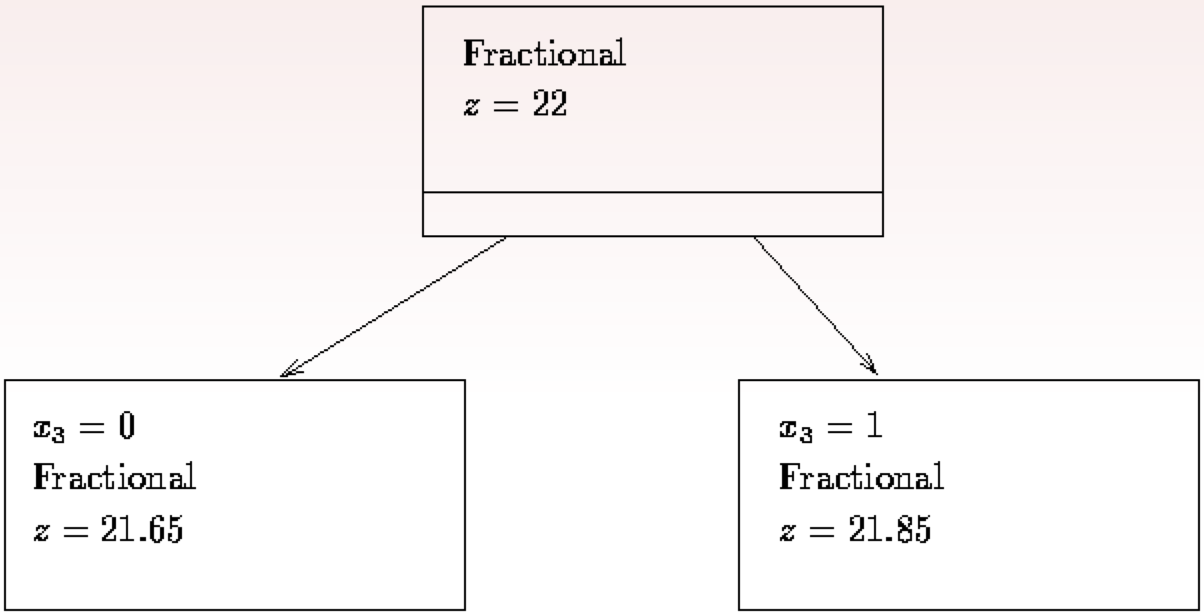
$$x_1 = 1, x_2 = 1, x_3 = 0.5, x_4 = 0$$

z=22

BRANCHING

- We want x3 to be an integer, so we branch on x3:
  - In one case we add a constraint x3=0
  - In another we add the constraint x3=1

SOLVE RELAXATIONS

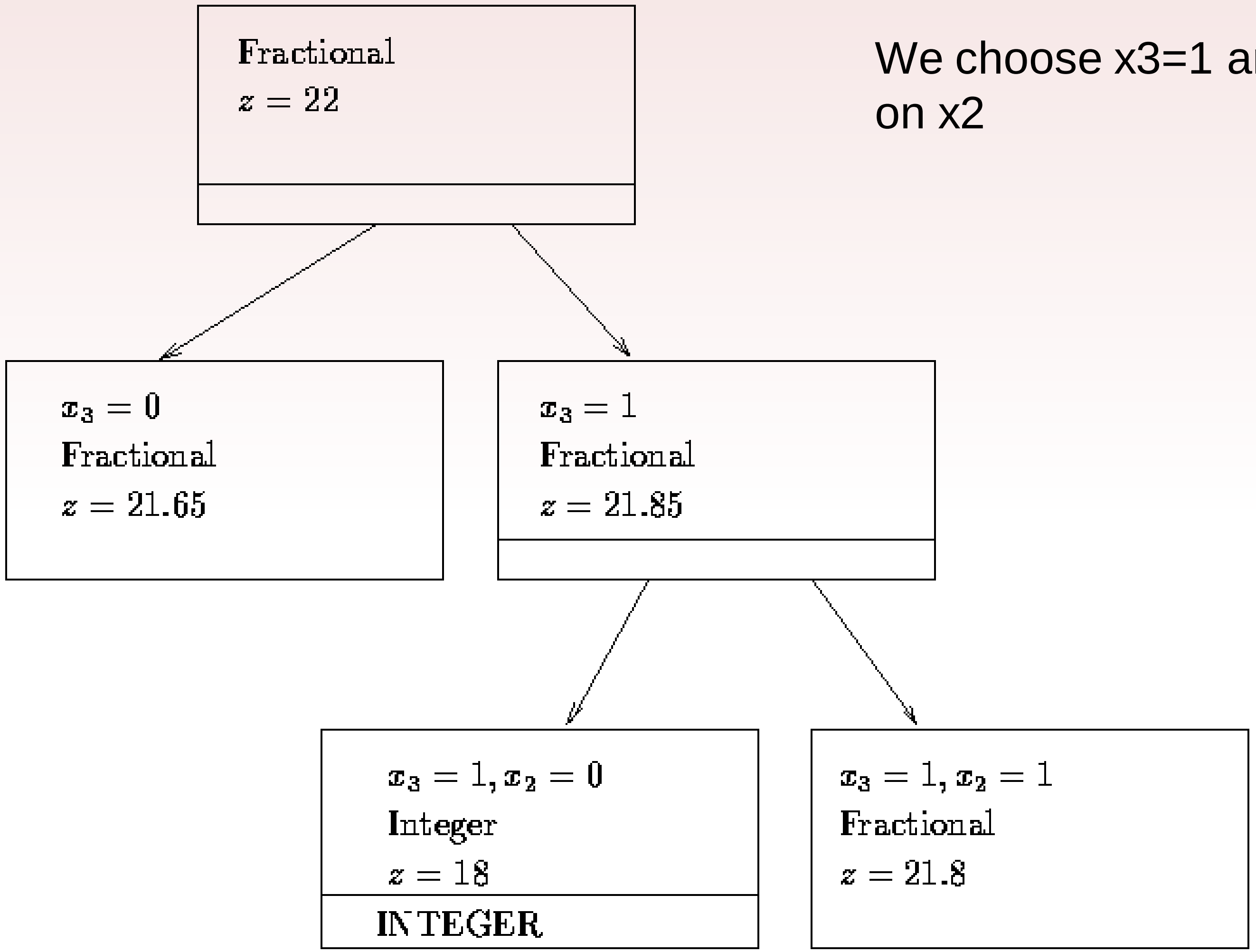


- $x_3 = 0$: objective 21.65, $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 0.667$;

  $x_3 = 1$: objective 21.85, $x_1 = 1$, $x_2 = 0.714$, $x_3 = 1$, $x_4 = 0$.

SELECT ACTIVE SUBPROBLEM

We choose x3=1 and branch on x2

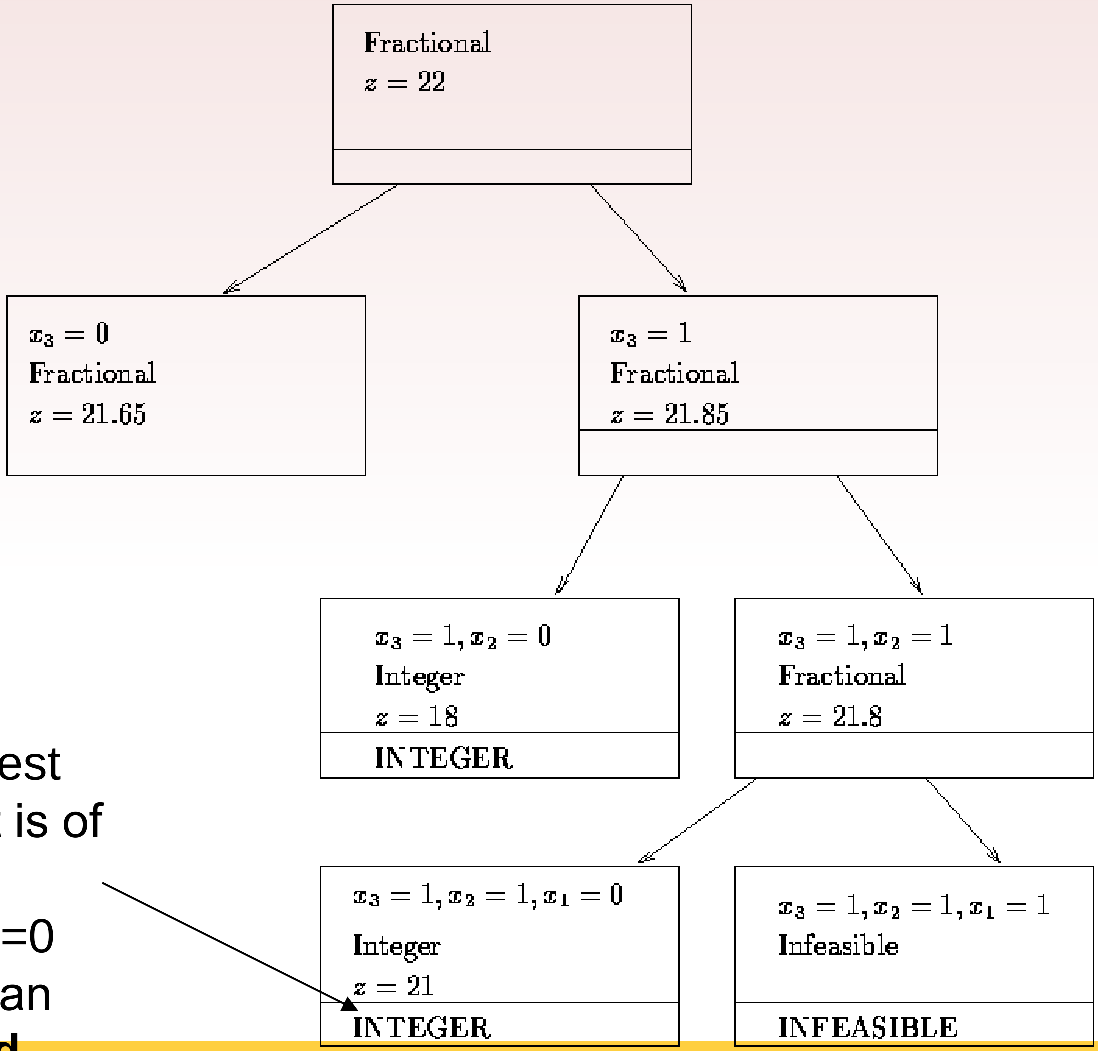


SOLVE RELAXATION

Integer solution!

- $x_3 = 1$, $x_2 = 0$: objective 18, $x_1 = 1$, $x_2 = 0$, $x_3 = 1$, $x_4 = 1$;
- $x_3 = 1$, $x_2 = 1$: objective 21.8, $x_1 = 0.6$, $x_2 = 1$, $x_3 = 1$, $x_4 = 0$.

OTHER ACTIVE PROBLEMS



This is the best solution … it is of no need to branch at x3=0 … this line can be **fathomed**

SOLVE RELAXATION

- $x_3 = 1$, $x_2 = 1$, $x_1 = 0$: objective 21, $x_1 = 0$, $x_2 = 1$, $x_3 = 1$, $x_4 = 1$;
- $x_3 = 1$, $x_2 = 1$, $x_1 = 1$: infeasible.

CONDITIONS FOR B&B

$$
\begin{aligned}
&\min z = f(\mathbf{x}) + c^T \mathbf{y} \\
&s.t. \\
&h(\mathbf{x}) = 0 \\
&g(\mathbf{x}) + \mathbf{My} \leq 0 \\
&\mathbf{x} \in X, \mathbf{y} \in Y
\end{aligned}
$$

- Objective term f(x) has to be convex
- Each component in h(x) is linear
- Each component in g(x) is convex over X
- X is convex
- Y is determined by linear constraints

MULTI PURPOSE OPTIMIZERS

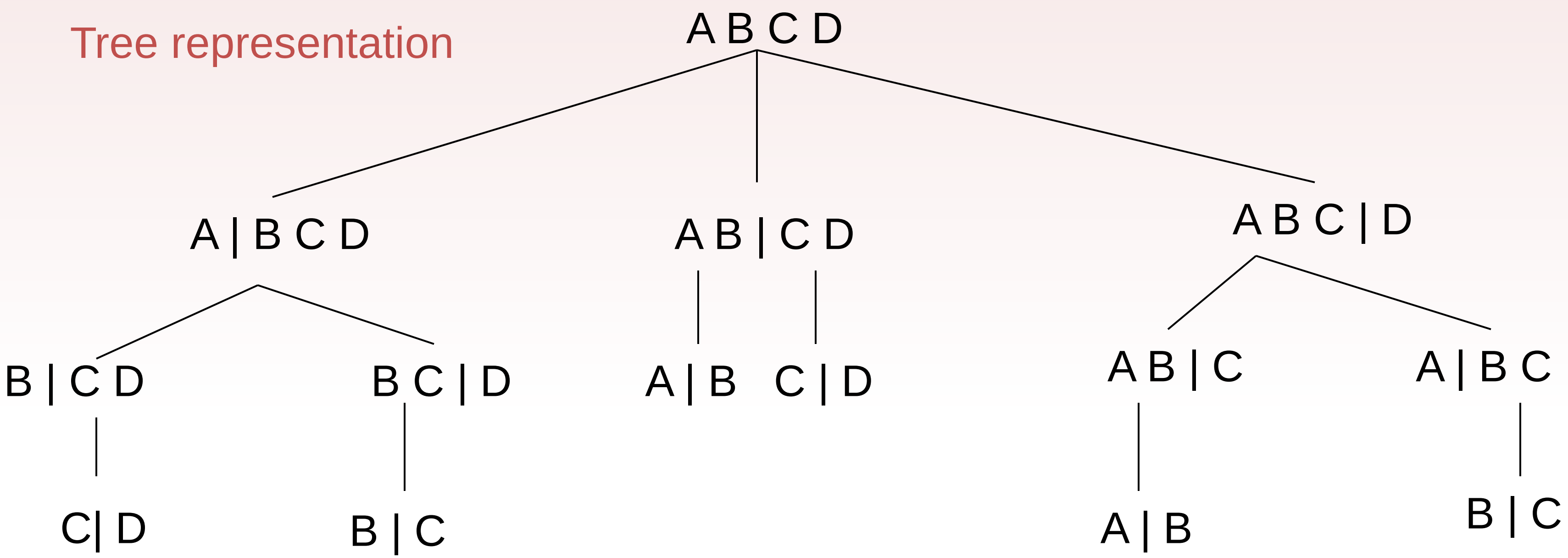
- Besides MATLAB there are alternative solvers, specifically for optimization there are powerful platforms available:
  - AMPL (Algebraic Modeling Programming Language) www.ampl.com
  - GAMS (General algebraic modeling system) www.gams.com
  - Xpress MP http://www.dashoptimization.com/

EXAMPLES

- Process synthesis and design
- Process operation
- Logistic processes (scheduling and planning)

PROCESS SYNTHESIS

Tree representation



Sharp split separation of a multi-component mixture, each route has certain costs asociated with it … which route to take→ MI(N)LP problem

HEAT EXCHANGER NETWORKS

The cost C_{ij} of assigning stream i to exchanger j is as follows:

	Exchangers			
Streams	1	2	3	4
A	94	1	54	68
B	74	10	88	82
C	73	88	8	76
D	11	74	81	21

$$\min Z = \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}x_{ij}$$

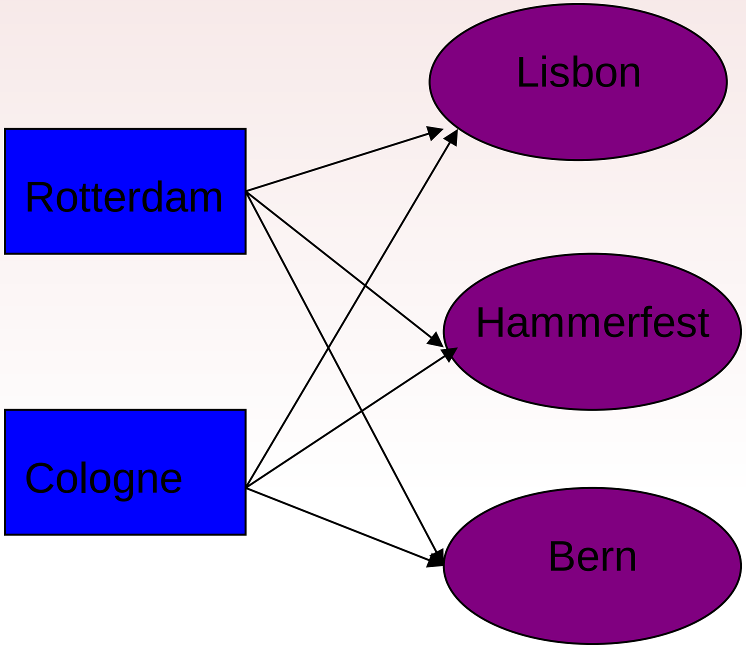
$$s.t \quad \sum_{i=1}^{n} x_{ij} = 1 \qquad j = 1,..n$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad i = 1,..n$$

$$x_{ij} = 0,1 \quad i = 1,..n \quad j = 1,..n$$

This is a MILP problem

LET'S EVALUATE AN EXAMPLE IN AMPL



We have two plants and three customers (markets) where we have to ship products to. As you can see there are several ways we can get the products to the markets

The objective is to do it the most efficient way, such that the supply meets demand.

FORMULATION OF THE PROBLEM

$$\min J = \sum_i \sum_j c_{ij} x_{ij}$$

$$s.t.$$

$$\sum_i x_{ij} \leq a_i$$

$$\sum_j x_{ij} \leq b_j$$

(10-26)

a_i	supply of commodity at plant i
b_j	demand for commodity at market j
c_{ij}	costs per unit shipment per plant i and market j
x_{ij}	volume of commodity to ship from plant i to market j (x_{ij}>0)

DATA FOR OUR PROBLEM

Shipping distance	Lisbon	Hammerfest	Bern	Plant supplies
Rotterdam	2300	3240	850	550
Cologne	2360	3230	580	400
Market demands	335	290	265	

CODING THE PROBLEM IN AMPL

- You should prepare a model file TRANSP1.MOD:

```
# References:
#
set ORIG;     # origins
set DEST;     # destinations
param supply{ORIG};                 # amounts available at origin
param demand{DEST};                 # amounts required at destinations
   check: sum {i in ORIG} supply[i] >= sum {j in DEST} demand[j];
param Unitcost >=0;                 # shipping cost per case per 1000 miles
param distance{ORIG,DEST} >= 0;    # shipping distances
param cost{i in ORIG,j in DEST} := Unitcost * distance[i,j]/1000;
var   NoUnits{ORIG,DEST} >= 0;     # units to be shipped
minimize total_cost:
   sum {i in ORIG, j in DEST} cost[i,j] * NoUnits[i,j];
subject to Supply {i in ORIG}:
   sum {j in DEST} NoUnits[i,j] <= supply[i];
subject to Demand {j in DEST}:
   sum {i in ORIG} NoUnits[i,j] = demand[j];
```

CODING THE PROBLEM IN AMPL

- And a data file (TRANSP1.DAT)

```
param   Unitcost := 95;

param:  ORIG:            supply:=   # defines set "ORIG" and param
   "supply"
        ROTTERDAM        550
        COLOGNE          400 ;

param:  DEST:            demand:=   # defines "DEST" and "demand"
   LISBON                335
   HAMMERFEST   290
   BERN         265 ;

param           distance:
        LISBON           HAMMERFEST     BERN :=
ROTTERDAM       2300            3240            850
COLOGNE         2360            3230            580 ;
```

RUNNING THE MODEL

```
option solver minos;
solve;
display NoUnits, NoUnits.rc > C:\EXAMPLES\TRANSP1.OUT;
display total_cost >> C:\EXAMPLES\TRANSP1.OUT;
close C:\EXAMPLES\TRANSP1.OUT;
```

Will give the following output:

```
:                        NoUnits  NoUnits.rc     :=
COLOGNE   BERN              265        0
COLOGNE   HAMMERFEST        135        0
COLOGNE   LISBON              0        6.65
ROTTERDAM BERN                0       24.7
ROTTERDAM HAMMERFEST        155        0
ROTTERDAM LISBON            335        0
;

total_cost = 176933
```

SUMMARY

- Optimization is minimization or maximization of an objective function. The optimization variables can be constrained by equality or inequality constraints.
- We approached LP problems with a simplex algorithm (and linprog in MATLAB)
- We approached NLP problems with the Lagrange multiplier method
- And we noted that dynamic optimization problems can be encountered with Pontryagin's principle or (more practical) with vector parameterization.
- We ended with a logistic optimization problem solved with AMPL