Nonlinear Mixed-Integer Programming
- the MILP perspective

CO@Work Berlin

Armin Fügenschuh

08.10.2009

name of the author

Berlin Mathematical School

berlin

DFG Research Center MATHEON
Mathematics for key technologies

ZIB

The Beginning of Linear Programming
(G. Dantzig, Linear Programming, Oper Res 50, 42-47, 2002)

- George Dantzig's talk on „Programming in a Linear Structure" (meeting of the Econometric Society, Univ. of Wisconsin in Madison), 1948.
- Harold Hotelling objected „...but we all know the world is nonlinear".
- John von Neumann defended the flustered young Dantzig, saying that „if one has an application that satiesfied the axioms of the model, then it can be used, otherwise not."
- Hotelling was right: The world **is** highly nonlinear.
- But: Systems of linear **<u>in</u>**equalities allow an approximation of most kinds of nonlinear relations encountered in practical applications.

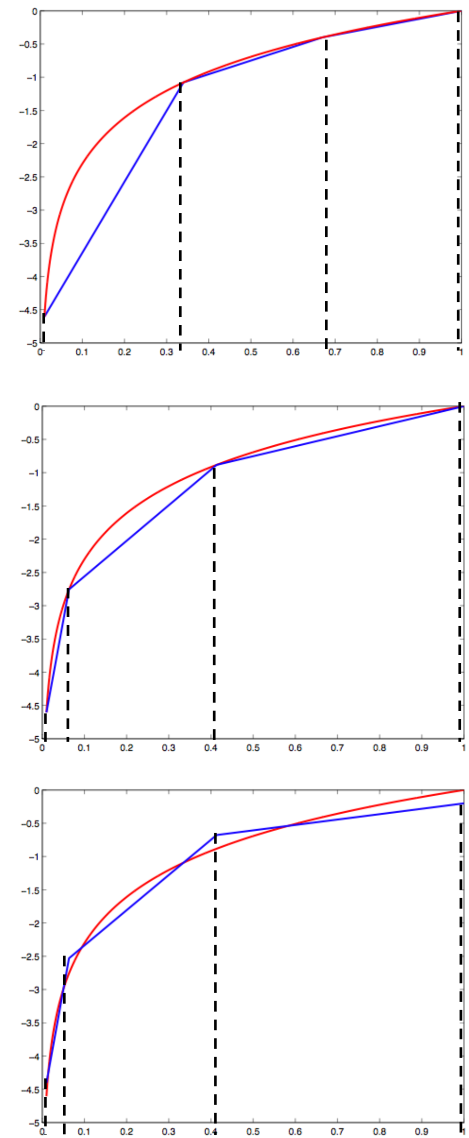

George Dantzig (1913-2005)


Harold Hotelling (1895-1973)


John von Neumann (1903-1957)

Piecewise Linear (Affine) Approximation

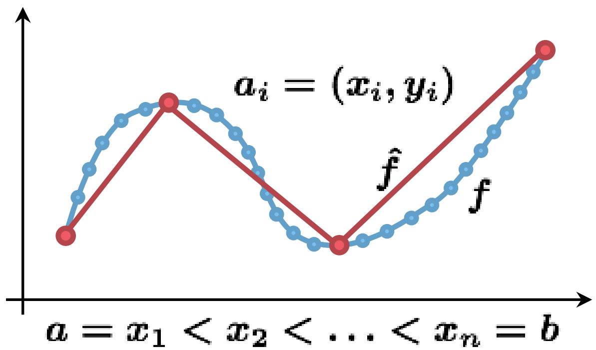
- In one dimension, a nonlinear function can be approximated by a sequence of piecewise linear functions (linear splines).
- The question is where to place the interpolation nodes for a given nonlinear function.
  - Equidistant interpolation.
  - Adaptive interpolation.
  - Adaptive approximation.
- This is a classical well-studied problem in numerical analysis, see for instance the textbook:
  - C. de Boor, A practical guide to splines, Springer, 1978.
- To apply these methods the function has to be „nice“ (i.e., twice continuous differentiable)

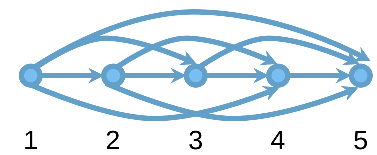


Piecewise Linear Approximation and Shortest Paths

(H. Imai, M. Iri, An optimal algorithm for approximating a piecewise linear function, J Inf Proc 9, 159-162, 1986)

- Assume a continuous function $f : [a, b] \to \mathbb{R}$ is given by a huge table of measurements, and linear interpolation between two measurements.
- We want to replace it by an approximation $\hat{f} : [a, b] \to \mathbb{R}$ using a smaller table.

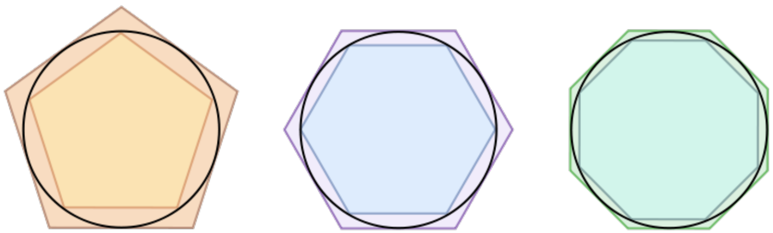




- Weighted digraph $D = (V, A, c)$ with $V = \{1, \ldots, n\}$ and $A = \{(i, j) : i < j\}$
- Select parameters $\alpha, \beta > 0$
  - α : cost per segment
  - β : cost for approximation error
- Define total cost per segment as $c_{i,j} := \alpha + \beta \left(\sum_{k=i}^{j} (f(x_k) - \hat{f}(x_k))^2 \right)$
- A shortest path from a to b in D is the desired approximation.

Piecewise Linear Approximation - An Ancient Idea

- The perhaps first mathematician who used piecewise linear approximation was Archimedes of Syracuse (287 BC-212 BC).
- He considered regular n-gons for an approximative computation of π.



source: wikipedia

- With n = 96 Archimedes achieved the bounds

$$3.1408450 \approx 3\tfrac{10}{71} < \pi < 3\tfrac{10}{70} \approx 3.1428571$$

- Liu Hui (220-280): 192-gon and 3072-gon.
- Zu Chong-Zhi (429-500): $3*2^{12}$ = 12288-gon.
- Jamshid Masud Al-Kashi (1380-1429): $3*2^{28}$-gon.
- 1596 Ludolph van Ceulen computed the first 35 digits of π using Archimedes method on a 2^{62}-gon. It took him 30 years of his life.

Archimedes, 1620
Domenico Fetti (1589-1624)

Ludolph van Ceulen
(1540-1610)

The Incremental Method

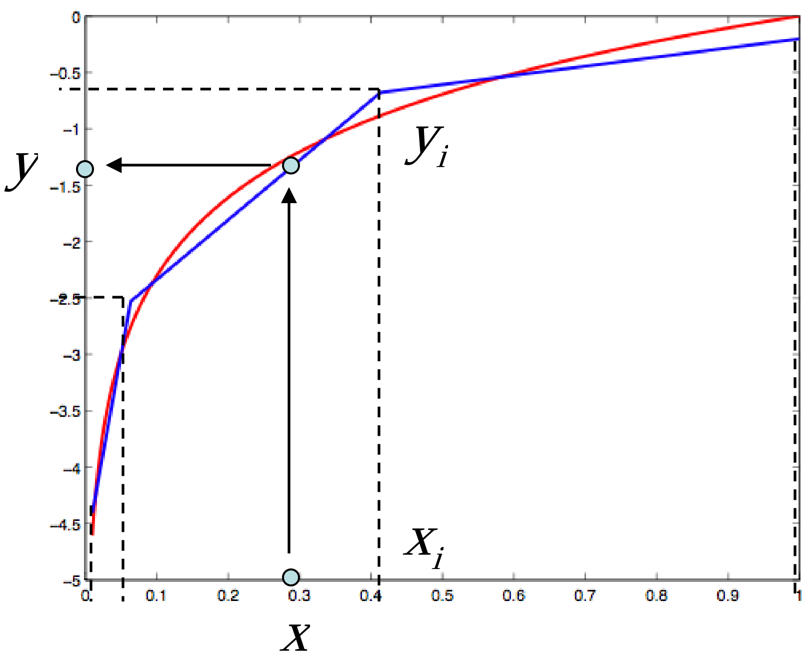
(H. Markowitz, A. Manne, On the solution of discrete programming problems, Econometrica 25, 84-110, 1957)

- Idea: „Filling“ of intervals
- Variables:
  - $w_i \in \{0,1\}$
  - $\delta_i \in [0,1]$
- Constraints:

$$w_i \geq \delta_i \geq w_{i+1}$$

$$x = x_0 + \sum_{i \in I} (x_i - x_{i-1})\delta_i$$

$$y = y_0 + \sum_{i \in I} (y_i - y_{i-1})\delta_i$$



$\delta_1 = 1$ $\delta_2 = 0.7$ $\delta_3 = 0$

$w_1 = 1$ $w_2 = 1$ $w_3 = 0$

The Convex Combination Method

(G. Dantzig, On the significance of solving linear programming problems with some integer variables, Econometrica 28, 30-44, 1960)

- Idea: Selection of exactly one interval
- Variables:
  - $w_i \in \{0,1\}$
  - $\lambda_i \in [0,1]$
- Constraints:

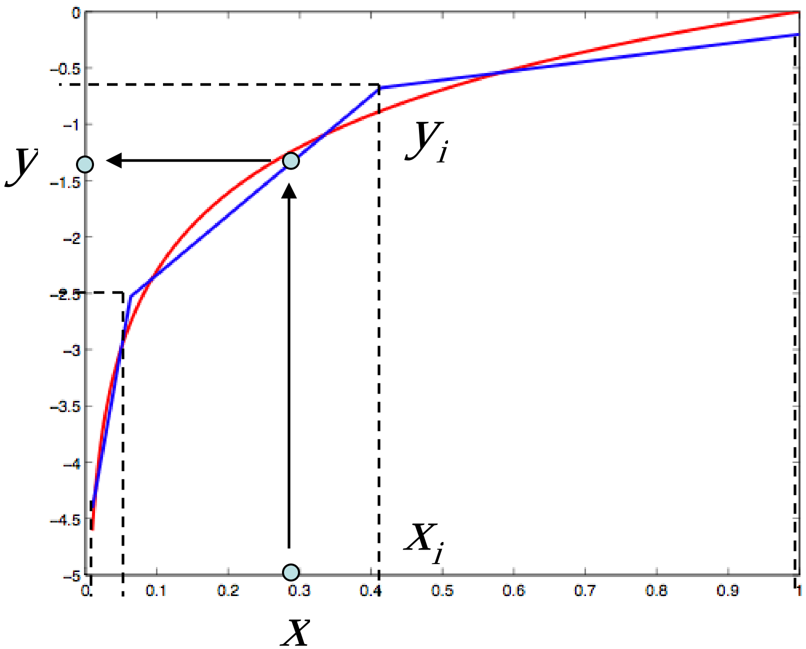
$$\sum_{i \in I} w_i = 1$$

$$\lambda_0 + \sum_{i \in I} \lambda_i = 1$$

$$w_i \leq \lambda_{i-1} + \lambda_i$$

$$x = x_0 \lambda_0 + \sum_{i \in I} x_i \lambda_i$$

$$y = y_0 \lambda_0 + \sum_{i \in I} y_i \lambda_i$$



$\lambda_0 = 0$ $\lambda_3 = 0$

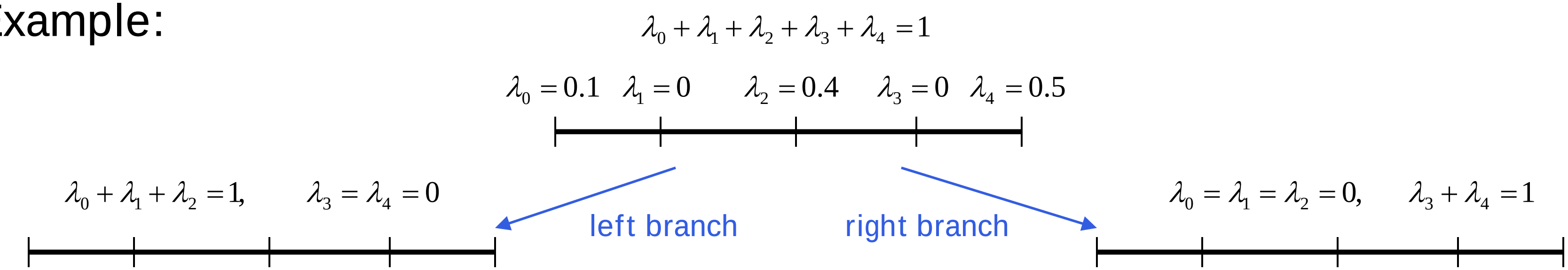
$\lambda_1 = 0.3$ $\lambda_2 = 0.7$

$w_1 = 0$ $w_2 = 1$ $w_3 = 0$

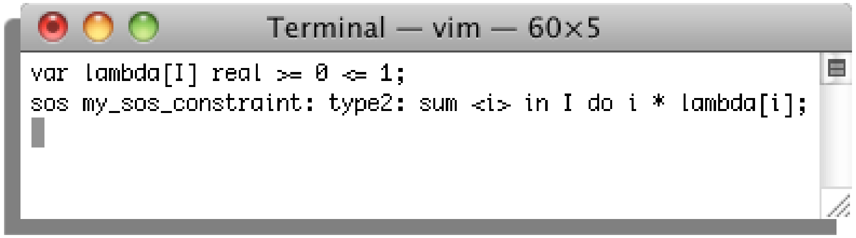
Special Ordered Sets (SOS2)

(E.M.L. Beale, J.J.H. Forrest, Global Optimization Using Special Ordered Sets, Math Prog 10, 52-69, 1976)

- Binary variables in the convex combination model were only introduced to model a logical relation:
  - At most two lambda variables are nonzero.
  - Nonzero lambda variables must be adjacent.
- Beale and Tomlin (1970) introduce SOS2 to handle this constraint implicitly in a branch-and-bound framework, without an explicit use of binary variables and constraints.
- Example:



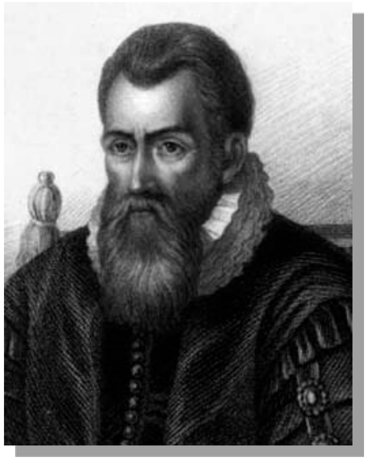
- Modern modeling languages / MILP solvers offer SOS2 as a built-in feature.
- Example (Zimpl):




```
var lambda[I] real >= 0 <= 1;
sos my_sos_constraint: type2: sum <i> in I do i * lambda[i];
```

Dealing with Functions of Several Variables

- Sometimes multivariate functions can be reduced to a sequence of monovariate functions.
- Example: $z = x^a \cdot y^b$.
- Here we can use an idea of John Napier from 1614: „*Mirifici logarithmorum canonis constructio*“.
- Assume $x, y > 0$. Then:


John Napier (1550-1617)

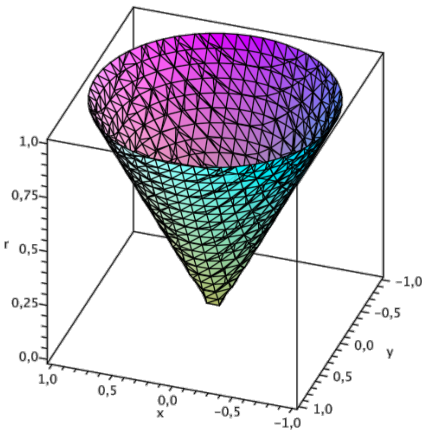
$$z = x^a \cdot y^b$$
$$\Leftrightarrow \quad \ln z = \ln(x^a \cdot y^b) = a \ln x + b \ln y$$

- Instead of one bivariate function we have to approximate three monovariate functions (by one of the methods discussed before):

$$\tilde{z} = \ln z$$
$$\tilde{x} = \ln x$$
$$\tilde{y} = \ln y$$
$$\tilde{z} = a\tilde{x} + b\tilde{y}$$

- Some multivariate functions have a special structure that can be exploited for highly efficient approximations.
- A quadratic cone (also second order cone, coll. „ice-cream cone“) is the set described by

$$\sqrt{x_1^2 + x_2^2 + \ldots + x_{n-1}^2} \leq x_n$$

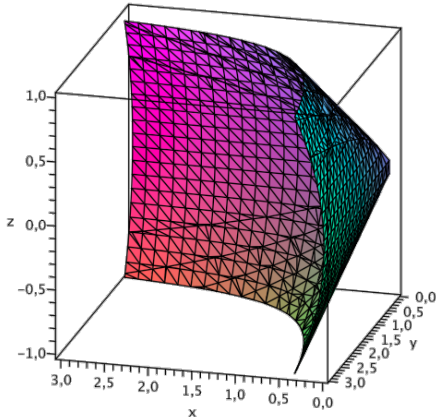


- Many other constraints can be transformed into a SOC:

$$a \cdot b \geq c \quad \Leftrightarrow \quad \sqrt{x^2 + y^2} \leq z$$

where

$$x := \tfrac{1}{2}(a - b), \quad y := \sqrt{c}, \quad z := \tfrac{1}{2}(a + b)$$



- Further examples include the cone

$$z^T z \leq xy, \quad x, y \geq 0$$

Linear Approximations of Second Order Cones

(A. Ben-Tal, A. Nemirovski, On polyhedral approximations of the second-order cone, Math Oper Res 26, 193-205, 1998 & F. Glineur, Polyhedral approximation of the second-order cone: computational experiments, Tech Rep, 2000)

- Pure SOCPs can be solved by nonlinear (interior point) methods.
- When also integer constraints are present, linear approximations of the SOC should be considered.
- Clearly one can approximate the SOC in the original space by linear cones (i.e., the unit disc by an n-gon).
- The error is $\varepsilon = \cos(\frac{\pi}{n})^{-1} - 1$.
- For an accuracy of 10^{-4} we would need a 223-gon.
- A much better approximation is the following:
  - Variables: $\alpha_i, \beta_i \in]-\infty, \infty[$
  - Constraints: $\sqrt{x_1^2 + x_2^2} \leq x_3 \quad <\approx>$

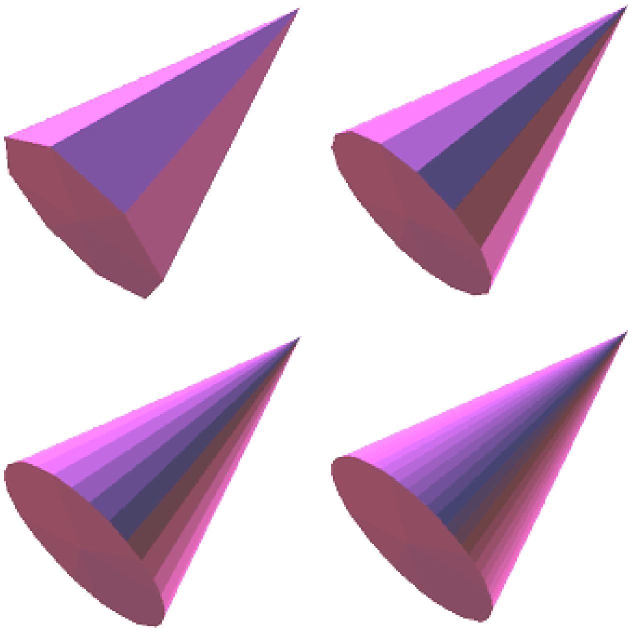
$$\alpha_0 = x_1$$

$$\beta_0 = x_2$$

$$\alpha_{i+1} = \cos(\tfrac{\pi}{2^i})\alpha_i + \sin(\tfrac{\pi}{2^i})\beta_i$$

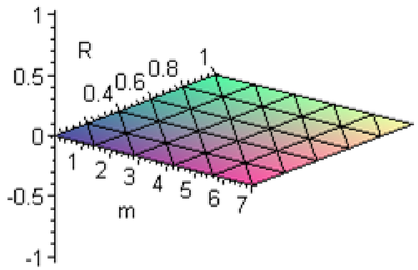
$$\beta_{i+1} \geq \left|\sin(\tfrac{\pi}{2^i})\alpha_i - \cos(\tfrac{\pi}{2^i})\beta_i\right|$$

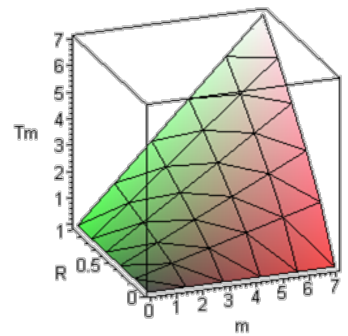
$$\cos(\tfrac{\pi}{2^I})\alpha_I + \sin(\tfrac{\pi}{2^I})\beta_I = x_3$$



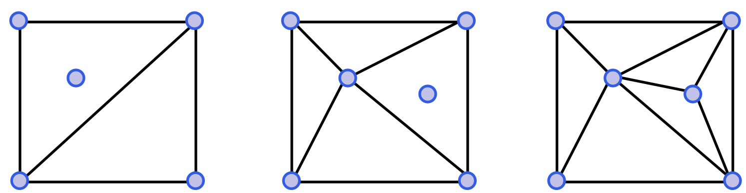
- Accuracy is $\varepsilon = \cos(\frac{\pi}{2^n})^{-1} - 1$.

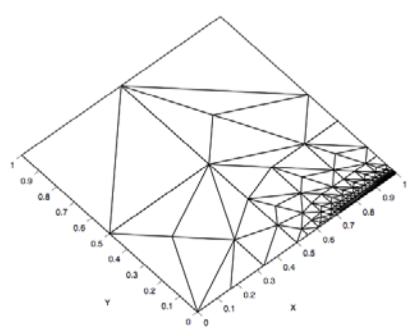
- If the function does not fit in that catagories, one can apply a higher dimensional analogon of 1-d piecewise linear approximation.
- Prerequisite: Introduce a triangulation (in 2-d) or, in general, a decomposition of the domain in simplices.
    - Equidistant:

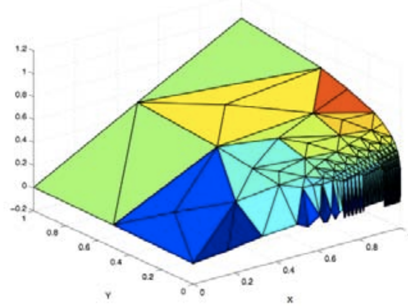




    - Adaptive:
        - Delauney triangulation of the domain.
        - Find the point with maximum error.
        - Introduce a new node there.
        - Compute a refined Delauney triangulation.







The Convex Combination Method in Higher Dimensions

- Idea: Selection of exactly one ~~interval~~ triangle
- Variables:
  - $w_i \in \{0,1\}$
  - $\lambda_i \in [0,1]$
- Constraints:

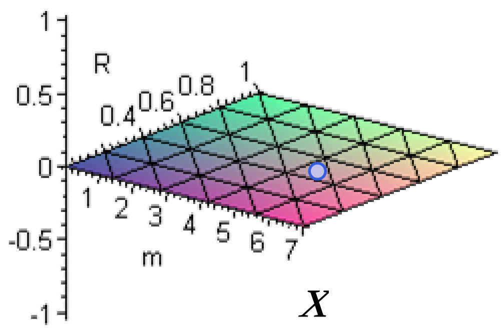
$$\sum_{i \in I} w_i = 1$$

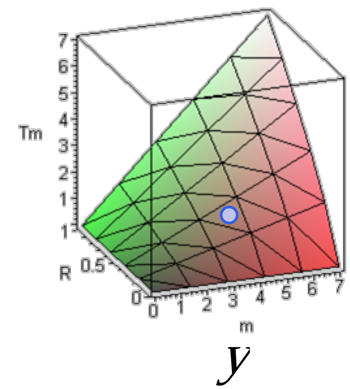
$$\lambda_0 + \sum_{i \in I} \lambda_i = 1$$

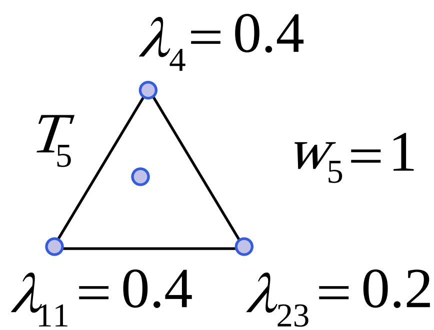
~~$w_i \le \lambda_{i-1} + \lambda_i$~~ $\quad w_i \le \sum_{j \in T_i} \lambda_j$

$$x = x_0 \lambda_0 + \sum_{i \in I} x_i \lambda_i$$

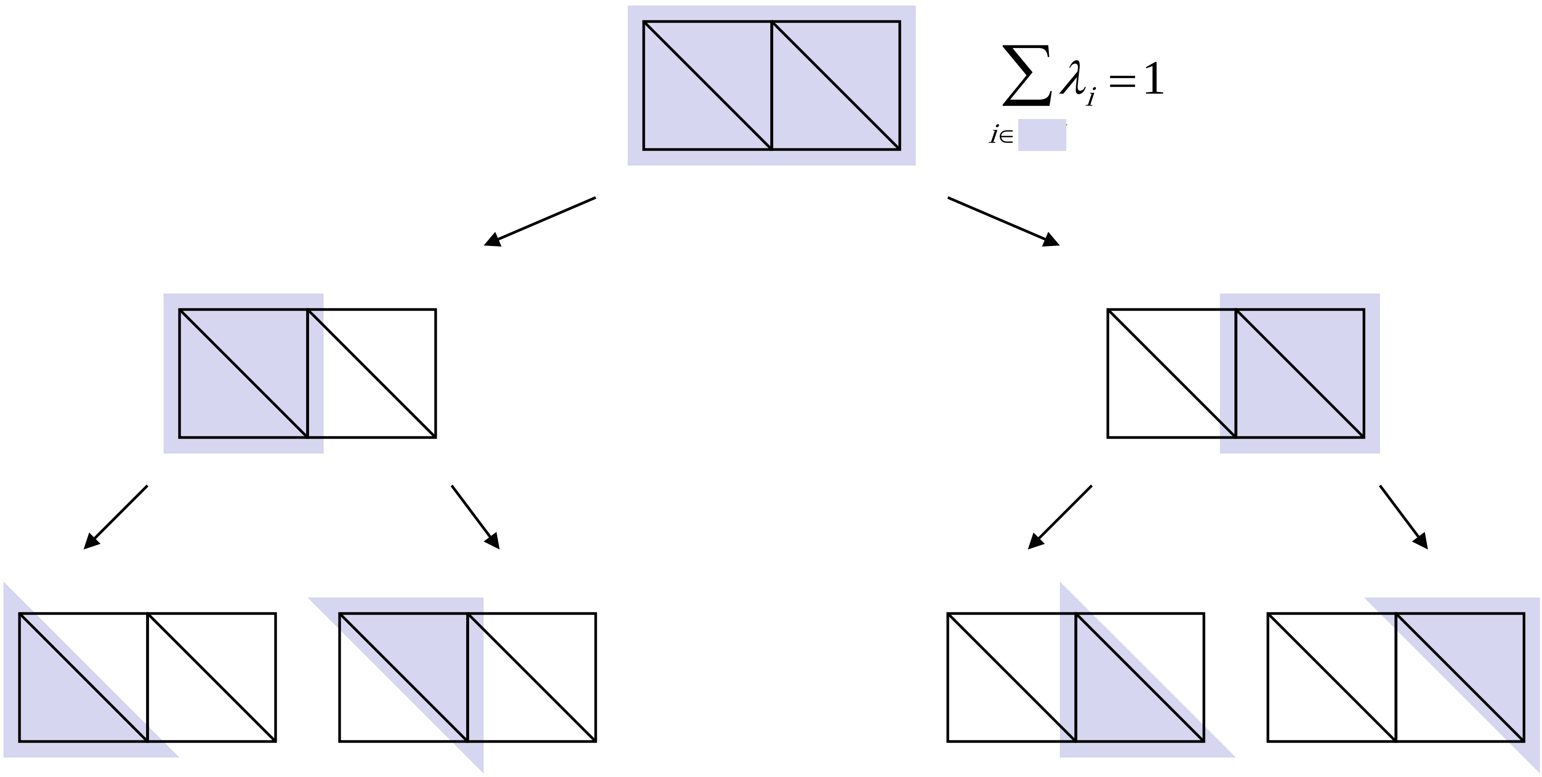
$$y = y_0 \lambda_0 + \sum_{i \in I} y_i \lambda_i$$







- Similar to the 1-d case it is possible to remove the auxiliary binary variables and handle the SOS property within the branching.



The Incremental Method in Higher Dimensions

(D. Wilson, Polyhedral Methods for piecewise-linear functions, PhD Thesis, 1998)

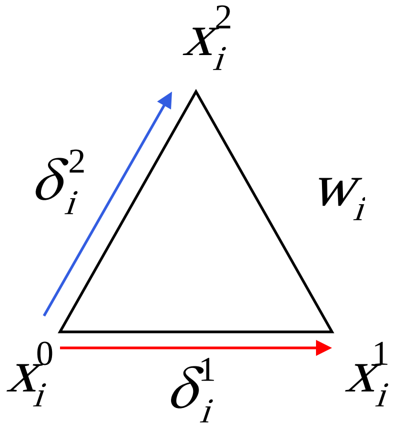
- Idea: „Filling“ of triangles
- Variables:
  - $w_i \in \{0,1\}$
  - $\delta_i^1, \delta_i^2 \in [0,1]$
- Constraints:

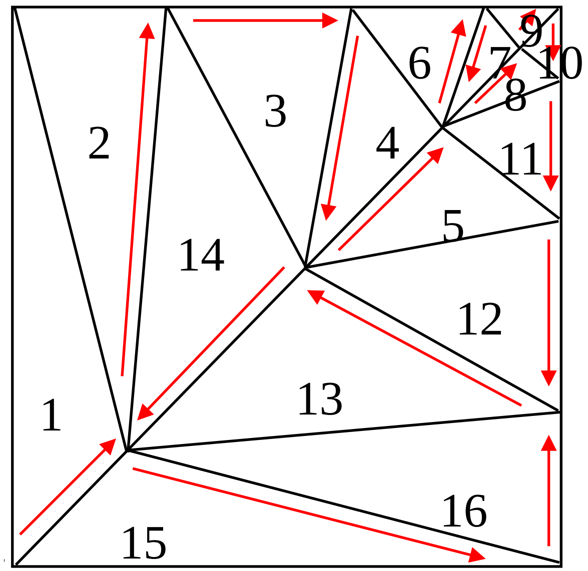
$$\delta_i^1 \geq w_i \geq \delta_{i+1}^1 + \delta_{i+1}^2$$

$$x = x_0^0 + \sum_{i \in I} \left[(x_i^1 - x_i^0)\delta_i^1 + (x_i^2 - x_i^0)\delta_i^2 \right]$$

$$y = y_0^0 + \sum_{i \in I} \left[(y_i^1 - y_i^0)\delta_i^1 + (y_i^2 - y_i^0)\delta_i^2 \right]$$

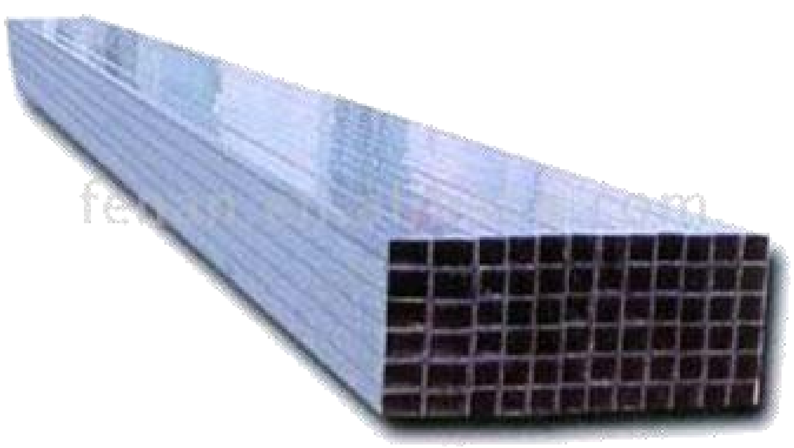
- Prerequisite: Ordering of the main direction





A Sheet Metal Design Task

- Design of square-tube conduits
  - Several square-shaped channels, surrounded by metal
  - Given cross section areas
  - Further engineering constraints
  - Design goals
    - <u>Minimal material usage</u>
    - Minimal deflection
    - Minimal torsion

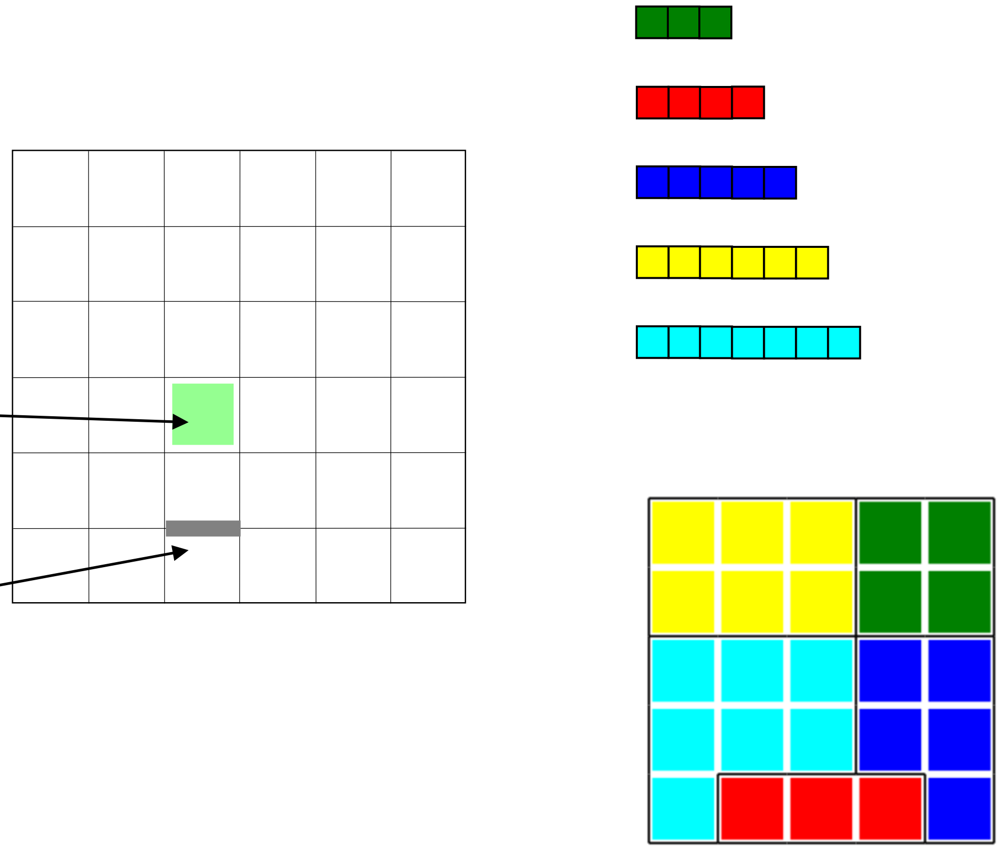


(A. Fügenschuh, M. Fügenschuh, Integer Linear Programming Models for Topology Optimization in Sheet Metal Design, Math Meth Oper Res 68, 313-331, 2008)

- Discretize the design envelope.
- Place channels on pixels and metal on the boundaries.

$$\delta_i^t \in \{0,1\}$$

$$\mu_{i,j} \in \{0,1\}$$

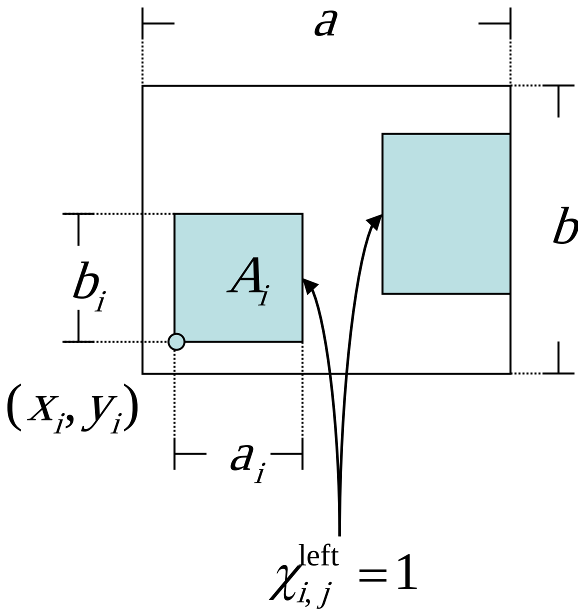


An Nonlinear Nonconvex Model

(A. Fügenschuh, W. Hess, L. Schewe, A. Martin, S. Ulbrich, Verfeinerte Modelle zur Topologie- und Geometrie-Optimierung von Blechprofilen mit Kammern, 2nd Proc SFB 666, 17-28, 2008)

- Given areas $a_i \cdot b_i = A_i$
- Given total area $a \cdot b = \sum_i A_i$
- Packing without overlapping

$$x_i + a_i \le x_j + M \cdot (1 - \chi_{i,j}^{\text{left}})$$
$$x_j + a_j \le x_i + M \cdot (1 - \chi_{i,j}^{\text{right}})$$
$$y_i + b_i \le y_j + M \cdot (1 - \chi_{i,j}^{\text{below}})$$
$$y_j + b_j \le y_i + M \cdot (1 - \chi_{i,j}^{\text{above}})$$
$$\chi_{i,j}^{\text{left}} + \chi_{i,j}^{\text{right}} + \chi_{i,j}^{\text{below}} + \chi_{i,j}^{\text{above}} \ge 1$$



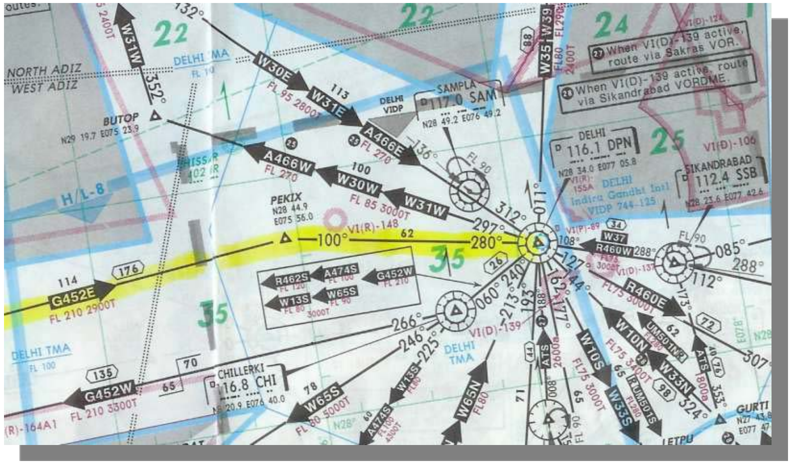
- Boundary conditions $x_i + a_i \le a, \quad y_i + b_i \le b$
- Minimize total border length $\min a + b + \sum_i (a_i + b_i)$

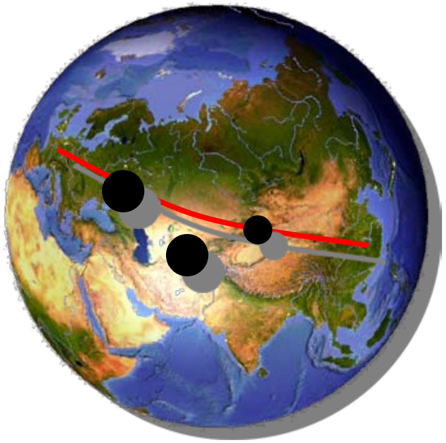
Exercise

- Try the various linearization techniques for this problem
- Basic model: template.zpl
- Contains everything but width*height = Area
- Data set: conduit6.dat
- Interpolation of log function: log-approx <lb> <ub> <error>
- Compute sin/cos values for SOC: trigonometrics <n>
- Solve problem with SCIP
- Write solution to file (wr sol my.sol)
- Visualisation tool: makesvg <my.sol>

Free Flight Routes

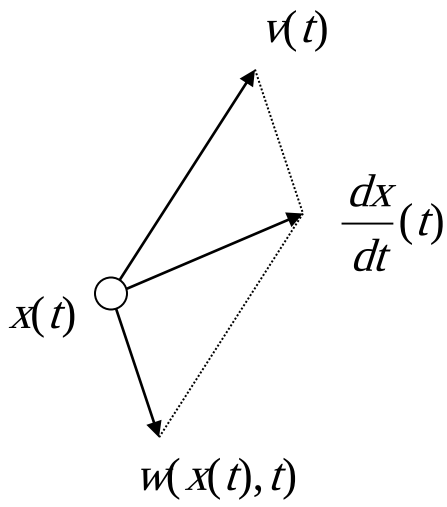
- Past/present: air traffic network (ATN)
- Future: free flight
- Main obstacles:
  - Restricted airspaces
  - Weather (wind, temp.)
  - Overflight costs
  - ETOPS
  - …





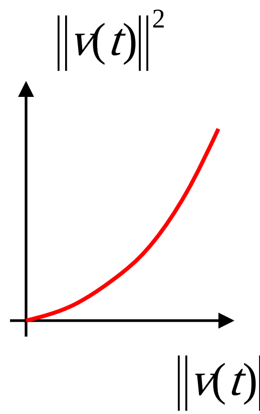
- Equations of motion

$$\frac{dx_1}{dt}(t) = v_1(t) + w_1(x_1(t), x_2(t), t)$$
$$\frac{dx_2}{dt}(t) = v_2(t) + w_2(x_1(t), x_2(t), t)$$



- Minimize fuel consumption:

$$\min \int_{t=0}^{T} \|v(t)\|^2 \, dt$$

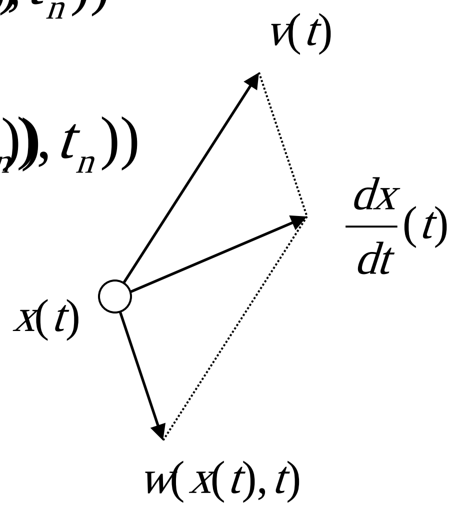


t	time
$x(t)$	position of aircraft at time t
$v(t)$	aircraft velocity at time t
$w(x(t),t)$	wind velocity at time t in $x(t)$

- Equations of motion

$$\frac{x_1(t+\Delta t) - x_1(t)}{\Delta t} \approx \frac{dx_1}{dt}(t) \qquad x_1(t_{n+1}) = x_1(t_n) + \Delta t\,(v_1(t_n) + w_1(x_1(t_n), x_2(t_n), t_n))$$

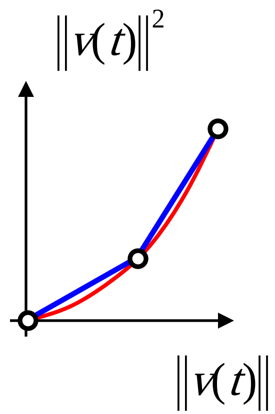
$$\frac{x_2(t+\Delta t) - x_2(t)}{\Delta t} \approx \frac{dx_2}{dt}(t) \qquad x_2(t_{n+1}) = x_2(t_n) + \Delta t\,(v_2(t_n) + w_2(x_1(t_n), x_2(t_n), t_n))$$



- Minimize fuel consumption:

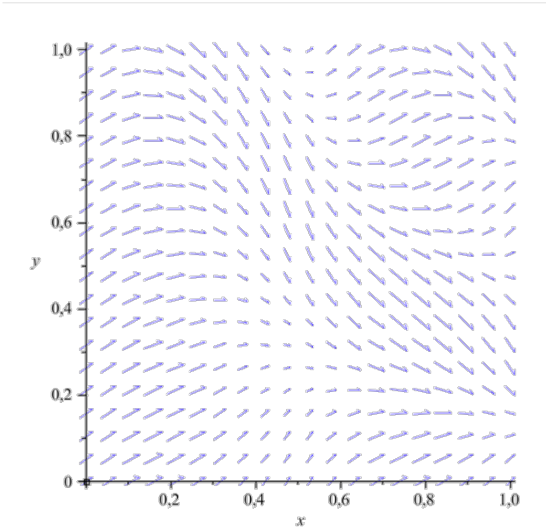
$$\min \sum_{n=0}^{N} \|v(t_n)\|^2 \Delta t \qquad \min \int_0^T \|v(t)\|^2\,dt$$

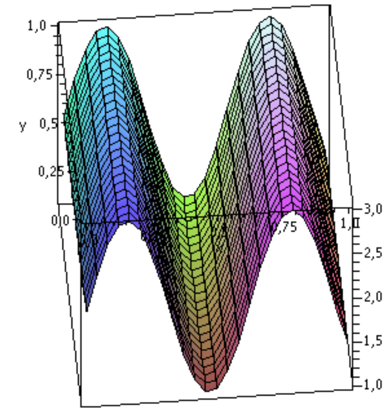
t	time
$x(t)$	position of aircraft at time t
$v(t)$	aircraft velocity at time t
$w(x(t),t)$	wind velocity at time t in $x(t)$

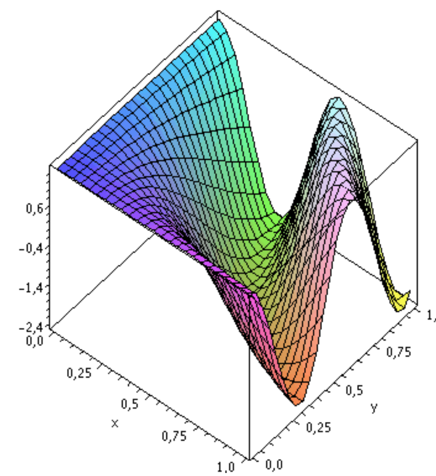


Wind

- 2-d vectorfield
- 2 nonlinear functions in 2-d

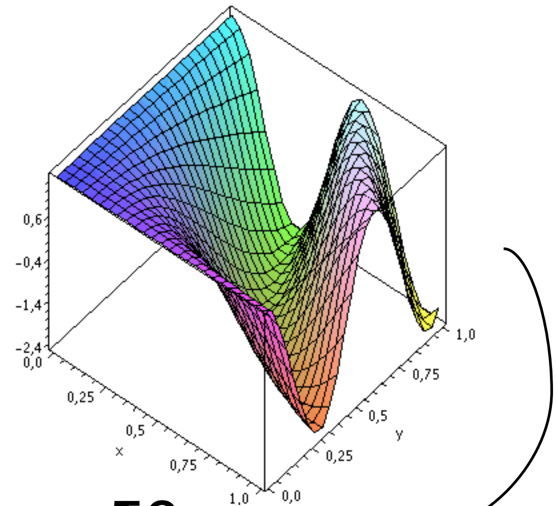


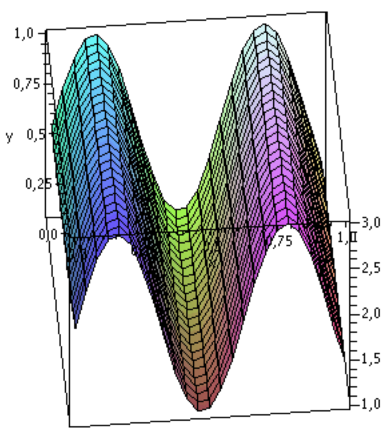




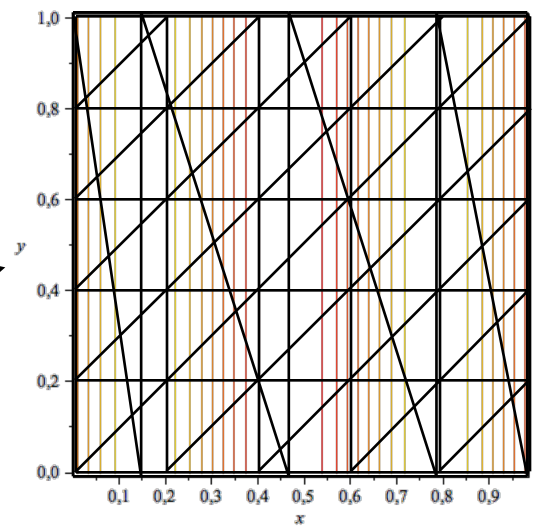
Piecewise Linear Approximation of a 2-D Nonlinear Function

- Contour lines
- Interpolation (equidistant)
- Interpolation (adaptive)

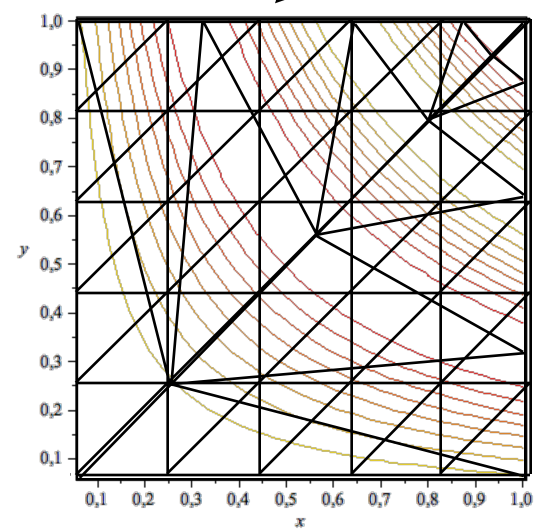




80

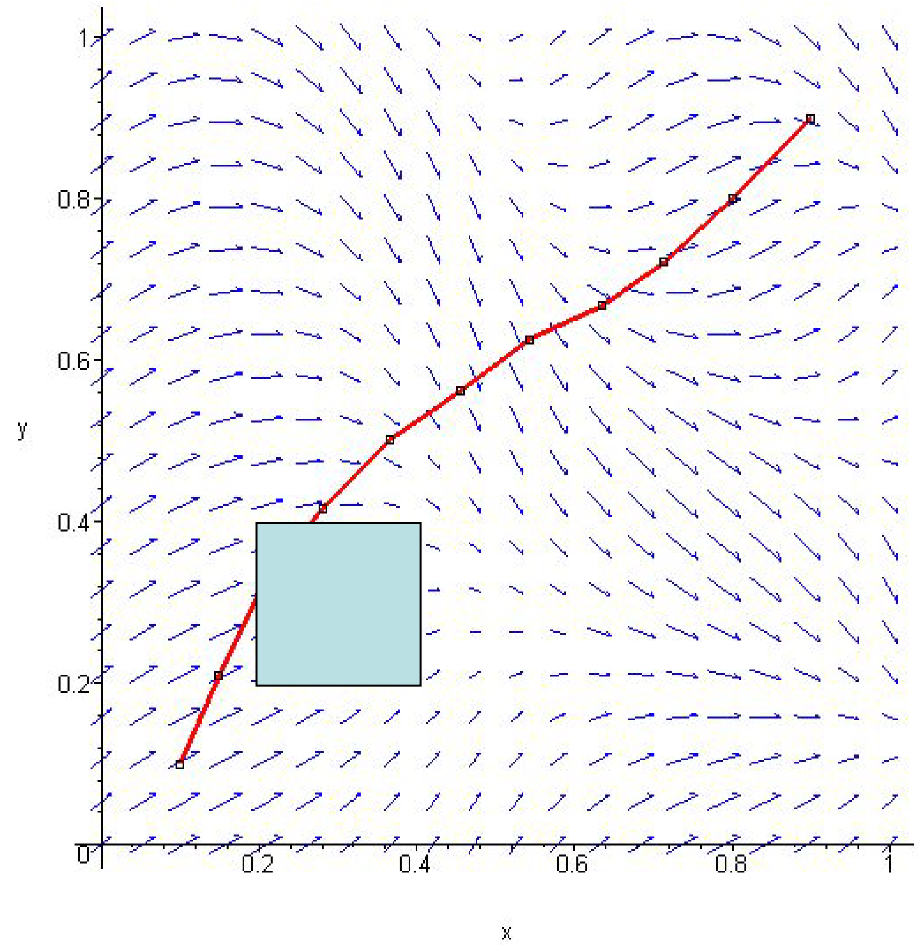


56



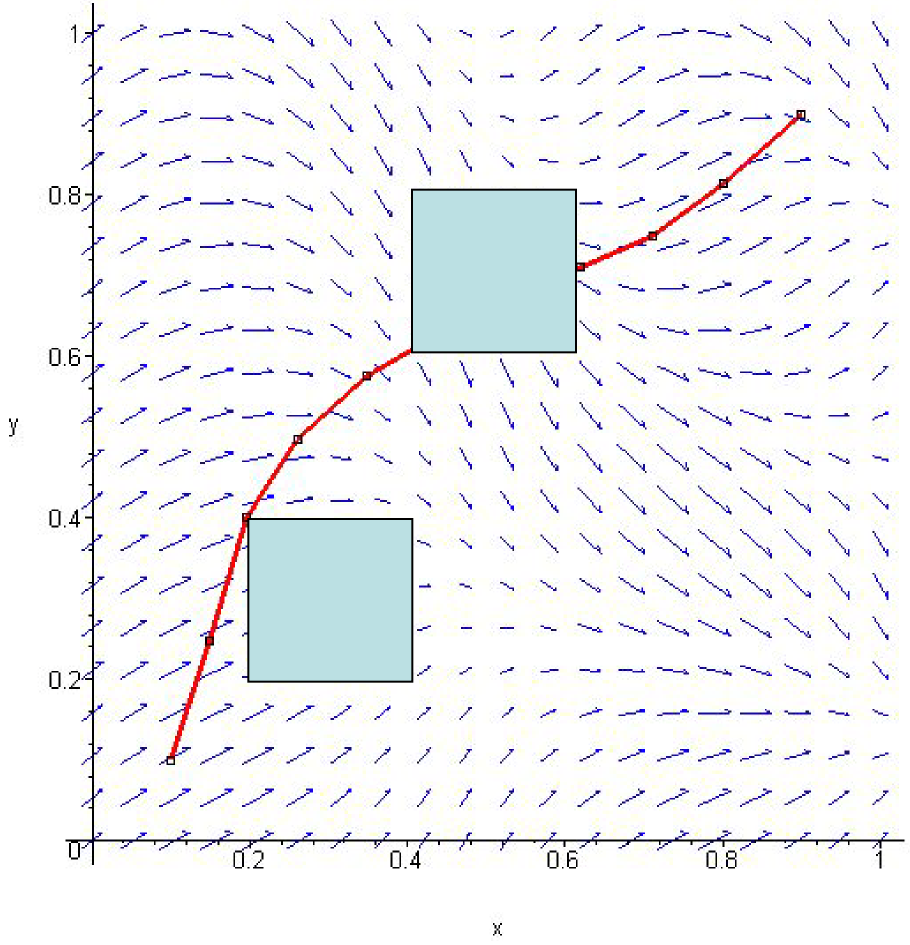
Free Flight: Results

- Incremental method in 2-d (CPlex10)
  - Equidist. 1: 891 sec.
  - Equidist. 2: 40 sec.
  - Adaptive: 5 sec.
- Restricted air-space
  - Blue area forbidden from $t = 0$ to $t = 5$



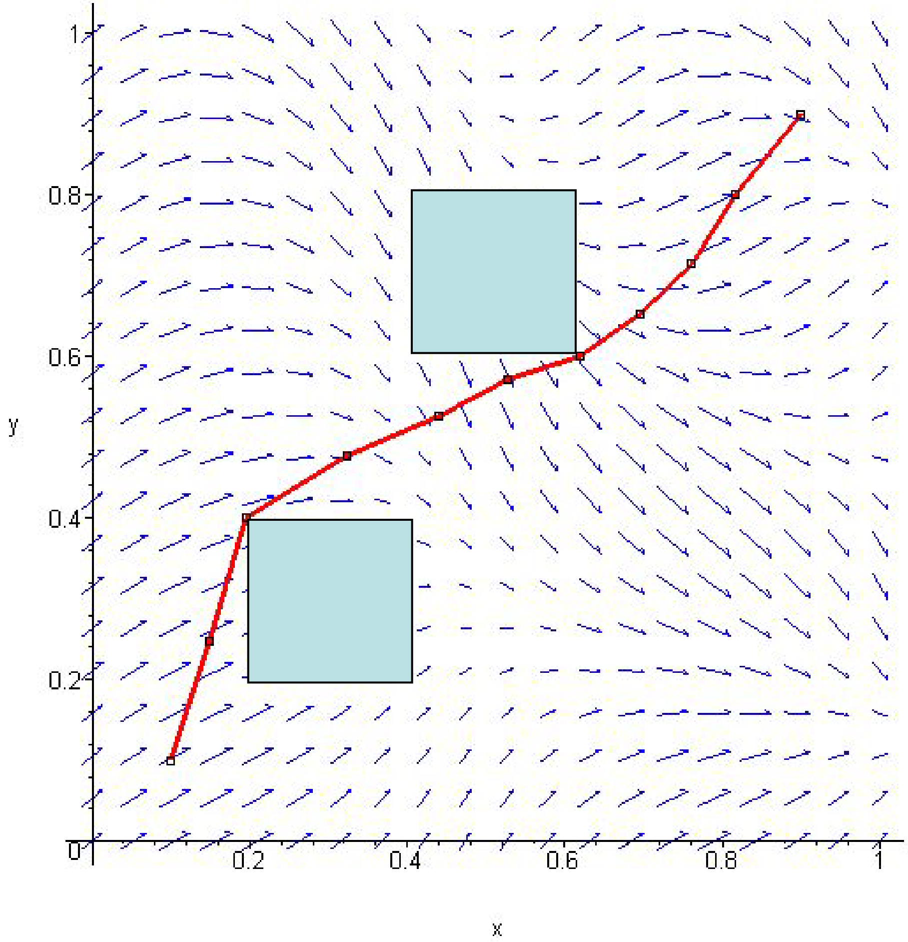
Free Flight: Restricted Airspaces

- Incremental method in 2-d (CPlex10)
  - Equidist. 1: 6600 sec.
  - Equidist. 2: 238 sec.
  - Adaptive: 78 sec.

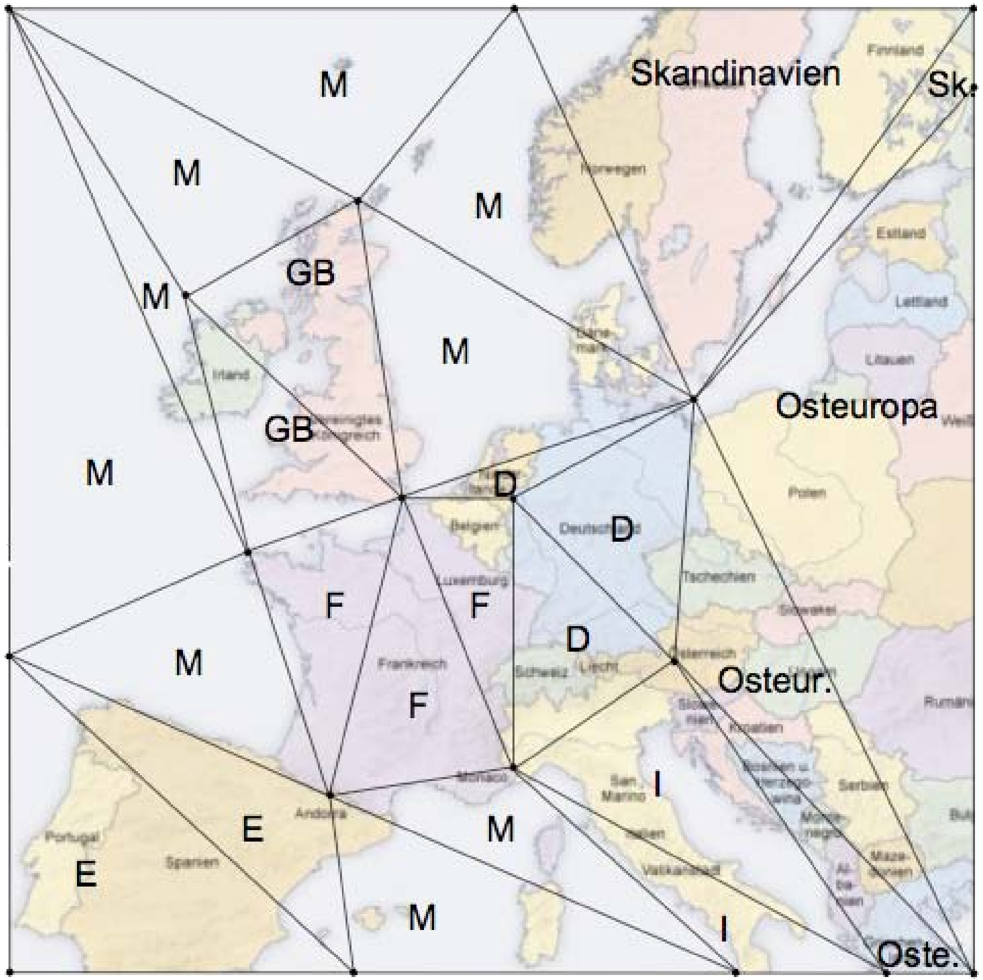


Free Flight: Resticted Airspaces

- Incremental method in 2-d (CPlex10)
  - Equidist. 1: 36579 sec.
  - Equidist. 2: 900 sec.
  - Adaptive: 618 sec.



Free Flight: Overflight Costs



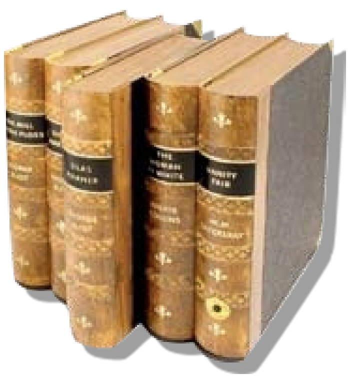
Free Flight: Overflight Costs

Free Flight: Overflight Costs

Some Facts about Paper and Recycling

(sources: Valkama, 2007 & Wikipedia)

- Newspapers, journals, books, packing material, hygienic articles,... are all made of paper and carton.
- Per year Germany consumes 21 million tons of paper and carton. That is, every person consumes ~250kg paper/year.
- Paper is one of the best-recycled products: 15.5 millions tons are reused.
- An increasing rate of today 67% of the fibers come from these sources.

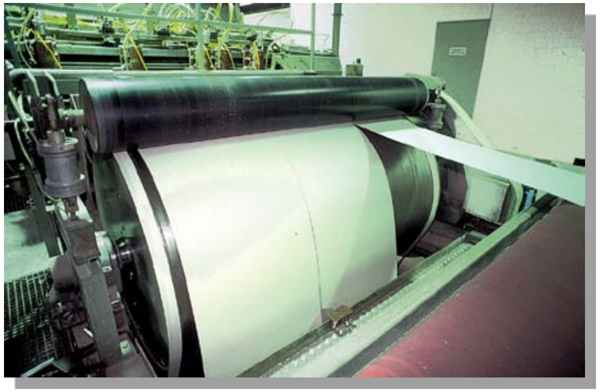


Steps in the Recovered Paper Production

(Valkama, 2007)

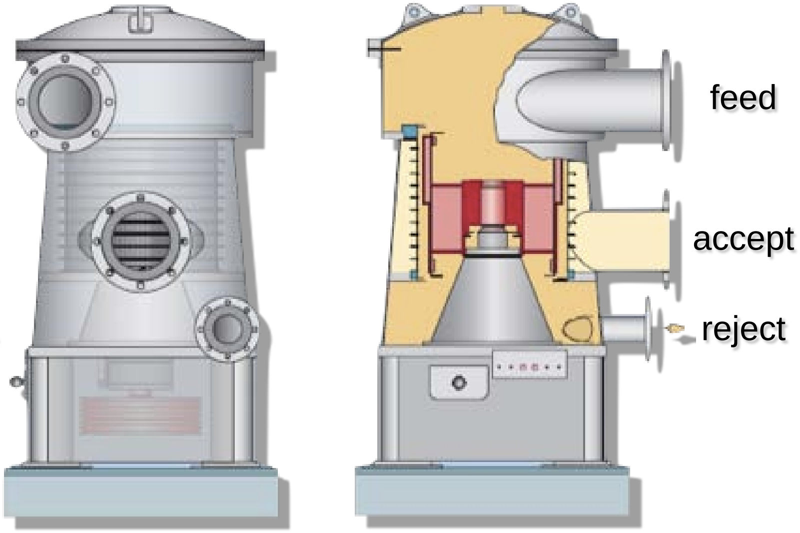
- Recycling fibres from waste paper consists of several steps:
  - Manual removal of contaminent materials.
  - Hackle paper into small pieces.
  - Resolve pieces in water and obtain pulp.
  - **Clean the pulp from** paper clips, plastic materials, and **stickies**.
  - De-ink the pulp.
  - The recovered paper suspension (fibres) is layed on grids and dried.
  - New paper rolls can now be produced.
- Too many stickies reduce the quality of the recovered paper, and can even break the rolls during production.
- Estimated production loss due to stickies: 265 mill. €.

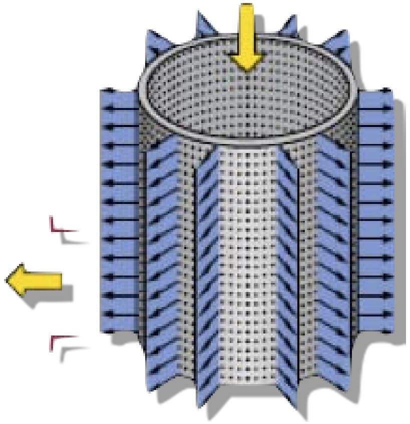




Sticky Sorting in Practice

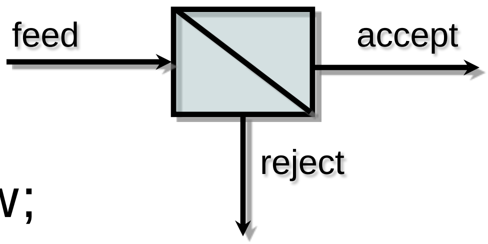
- Sorters (screeners, separators) come in various types and sizes.
- Differences:
  - Capacity (amount of pulp per time).
  - Sieves (size, slot type and width).
  - Max. admissible operating pressure.



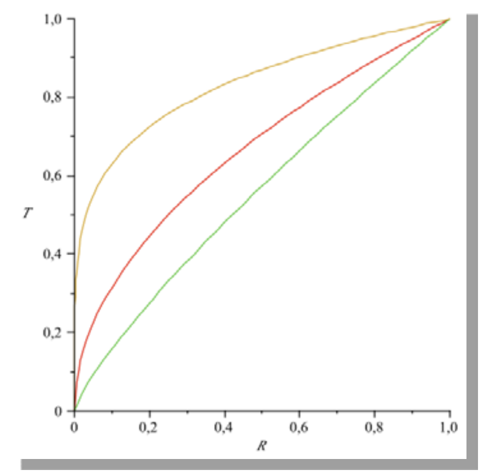


The Plug Flow Model

- Each sorter has one inflow feed, and two outflows, accept and reject.

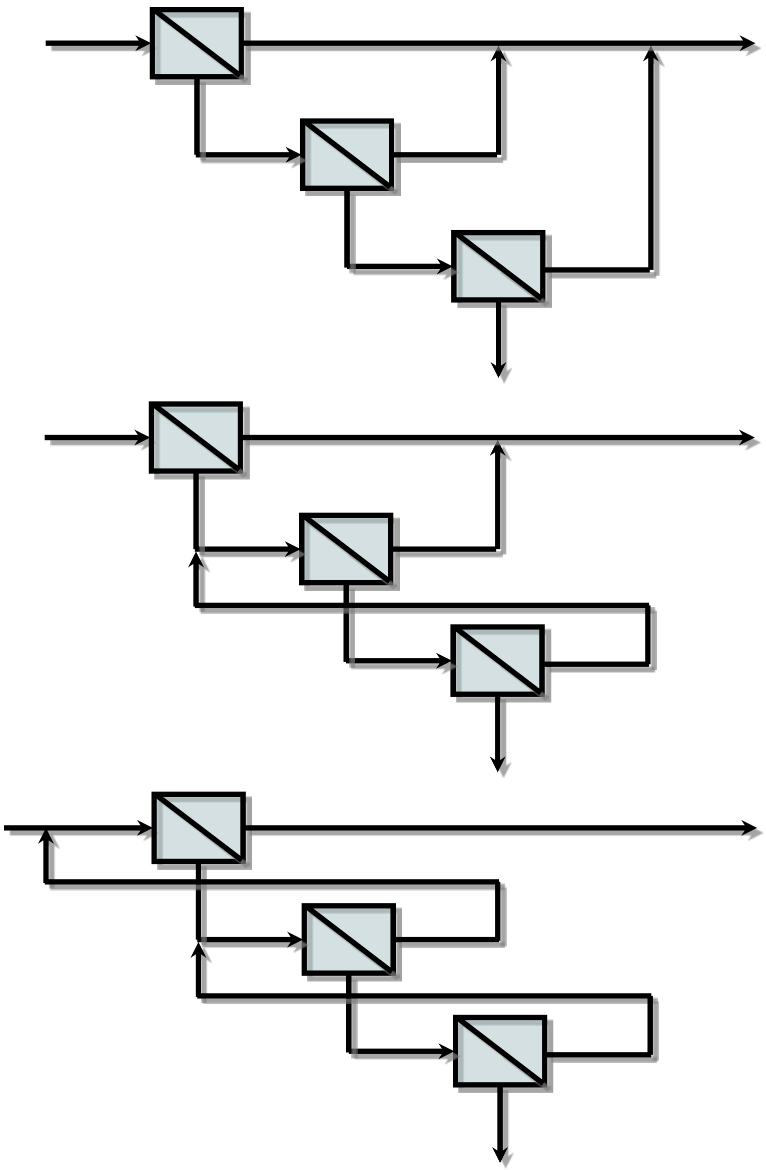


- Mass is conserved: $m^{in} = m^{acc} + m^{rej}$.
- Several components (~12) are in the pulp flow; we restrict here to two, fibers and stickies.
- The separation efficiency for component k is $T_k = \dfrac{m_k^{rej}}{m_k^{in}}$.
- The total mass reject loss (the reject rate) is $R = \dfrac{m^{rej}}{m^{in}}$.
- Kubát and Steenberg developed in the 1950‘s the plug flow model. According to their model the coupling $T_k = R^{\beta_k}$ holds for each k. Parameters β_k depend on the sorter and the component. They are obtained by measurements.



From Single Sorters to Systems of Sorters

- The sticky-sorter facility consists of 3-5 sorters and pipelines.
- Several examples of such systems are known:
  - feed forward,
  - partial cascade, and
  - full cascade.
- The pulp flow is sent through pipelines from one sorter to the next.
- The amount per commodity in the total inflow is known.
- The system has a total accept and a total reject.
- Goal: maximize stickies in total reject and fibers in total accept.



A Nonlinear Mathematical Model (NLP)

- Sets: pipes P, sorters S, components K .
- Parameters
  - Component $k \in K$ inflow mass: $m_k^{in} \geq 0$.
  - Pipe from accept/reject of sorter s_1 to inflow of s_2? $p_{s_1 s_2}^{acc}, p_{s_1 s_2}^{rej} \in \{0,1\}$.
  - Gain/loss per unit of k in total accept/reject: $c_k^{acc}, c_k^{rej} \in \mathbb{R}$.
  - Sorter's beta parameter vector: $\beta_{s,k} \in]0,1[$.
- Variables
  - Mass flow of k into/out of sorter s: $m_{s,k}^{in}, m_{s,k}^{acc}, m_{s,k}^{rej} \geq 0$.
  - Mass flow to total accept/reject: $m_k^{acc}, m_k^{rej} \geq 0$.
  - Reject rate of sorter s: $R_s \in [l_s, u_s]$.
- Constraints
  - Mass conservation: $m_{s,k}^{in} = m_{s,k}^{rej} + m_{s,k}^{acc}$.
  - Plug flow: $m_{s,k}^{rej} = R_s^{\beta_{s,k}} \cdot m_{s,k}^{in}$.
  - Network topology: $m_{s_2,k}^{in} = \sum_{s_1:(s_1 s_2)\in P} (p_{s_1 s_2}^{acc} \cdot m_{s_1,k}^{acc} + p_{s_1 s_2}^{rej} \cdot m_{s_2,k}^{rej})$.
- Objective: $\sum_{k\in K} (c_k^{acc} \cdot m_k^{acc} + c_k^{rej} \cdot m_k^{rej}) \rightarrow \max$.

Including the Topology

- There are many ways to connect the sorters.

#sorters	#topologies
1	1
2	8
3	318
4	26,688
5	3,750,240

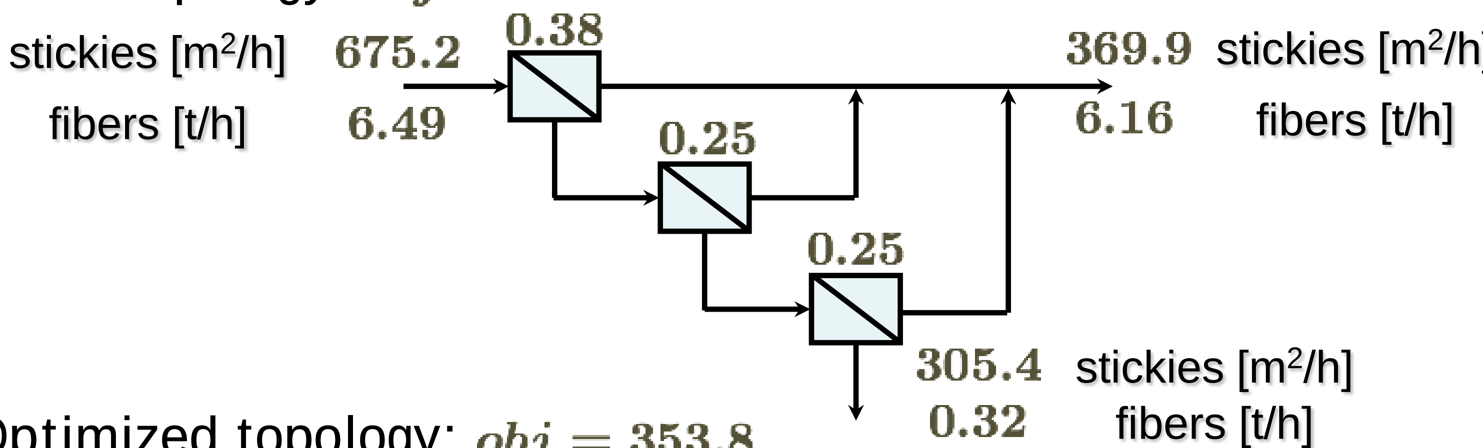
- Topological decisions can be taken into the model.
- Instead of parameters $p^{rej}_{s_1 s_2}$ and $p^{acc}_{s_1 s_2}$ we introduce a binary variables.
- Expressions $p^{acc}_{s_1 s_2} \cdot m^{acc}_{s_1,k}$ and $p^{rej}_{s_1 s_2} \cdot m^{rej}_{s_2,k}$ then are also nonlinear.
- They have to be linearized again.
- See also Floudas (1987, 1995), Nath, Motard (1981), Nishida, Stephanopoulos, Westerberg (1981), Friedler, Tarjan, Huang, Fan (1993), Grossmann, Caballero, Yeomans (1999), and many more.

Linearizing the Topology Constraints

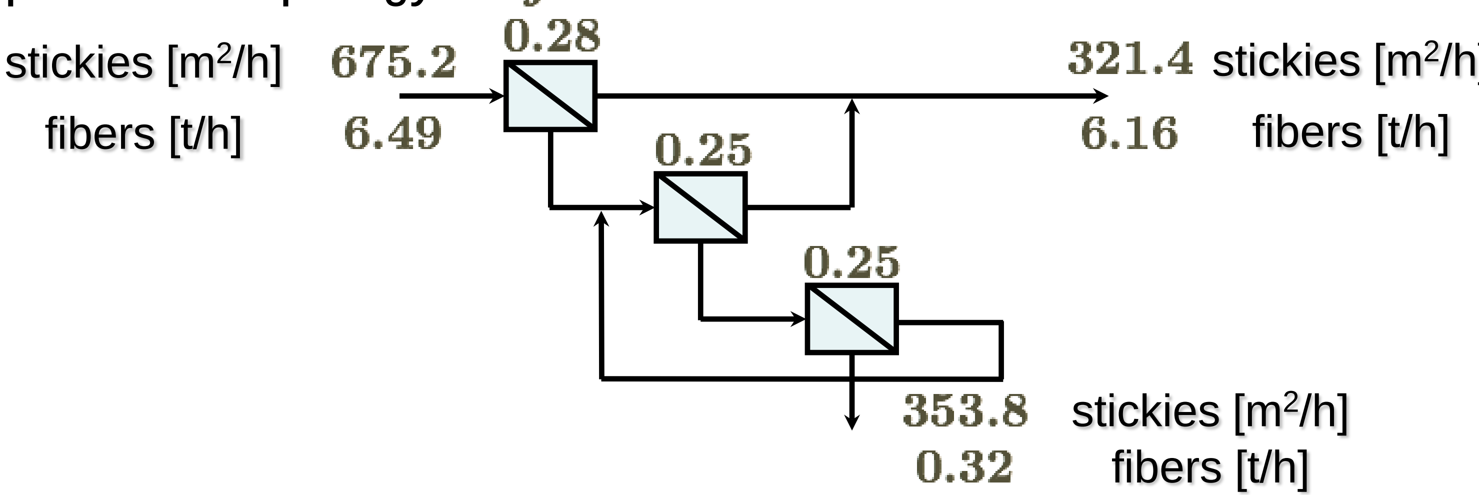
- Remember: $m^{in}_{s_2,k} = \sum_{s_1:(s_1 s_2)\in P} (p^{acc}_{s_1 s_2} \cdot m^{acc}_{s_1,k} + p^{rej}_{s_1 s_2} \cdot m^{rej}_{s_2,k})$.
- Introduce new variables $\mu^{acc}_{s_1 s_2,k}, \mu^{rej}_{s_1 s_2,k} \geq 0$ for potential mass flowing from accept/reject of sorter s_1 to the input of sorter s_2 .
- Constraints
  - Mass flow only in pipes (M a sufficiently large constant):
    $$\mu^{acc}_{s_1 s_2,k} \leq M \cdot p^{acc}_{s_1 s_2}, \quad \mu^{rej}_{s_1 s_2,k} \leq M \cdot p^{rej}_{s_1 s_2}.$$
  - Exactly one pipe is selected:
    $$\sum_{s_2:(s_1 s_2)\in P} p^{acc}_{s_1 s_2} = \sum_{s_2:(s_1 s_2)\in P} p^{rej}_{s_1 s_2} = 1.$$
  - Coupling of mass and potential mass:
    $$m^{in}_{s_2,k} = \sum_{s_1:(s_1 s_2)\in P} (\mu^{acc}_{s_1 s_2,k} + \mu^{rej}_{s_1 s_2,k}),$$
    $$m^{acc}_{s_1,k} = \sum_{s_2:(s_1 s_2)\in P} \mu^{acc}_{s_1 s_2,k},$$
    $$m^{rej}_{s_1,k} = \sum_{s_2:(s_1 s_2)\in P} \mu^{rej}_{s_1 s_2,k}.$$

Computational Results

- Objective: maximize stickies within reject.
- Additional constraint: fiber-loss (i.e., fibers in reject) at most 5%.
- Given topology: $obj = 305.4$



- Optimized topology: $obj = 353.8$



Zur Anzeige wird der QuickTime™
Dekompressor „H.264“
benötigt.

Thank you for your attention!

Questions?