Optimization of Sticky Separation in Waste Paper Processing

Armin Fügenschuh
Helmut Schmidt University /
University of the Federal Armed Forces Hamburg

joint work with:

Christine Hayn
Friedrich-Alexander University Erlangen-Nuremberg
Dennis Michaels
Technical University Dortmund

with further contributions from:

Mirjam Dür, Björn Geißler, Alexander Martin, Antonio Morsi, Samuel Schabel, Stefan Vigerske & Klaus Villforth

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

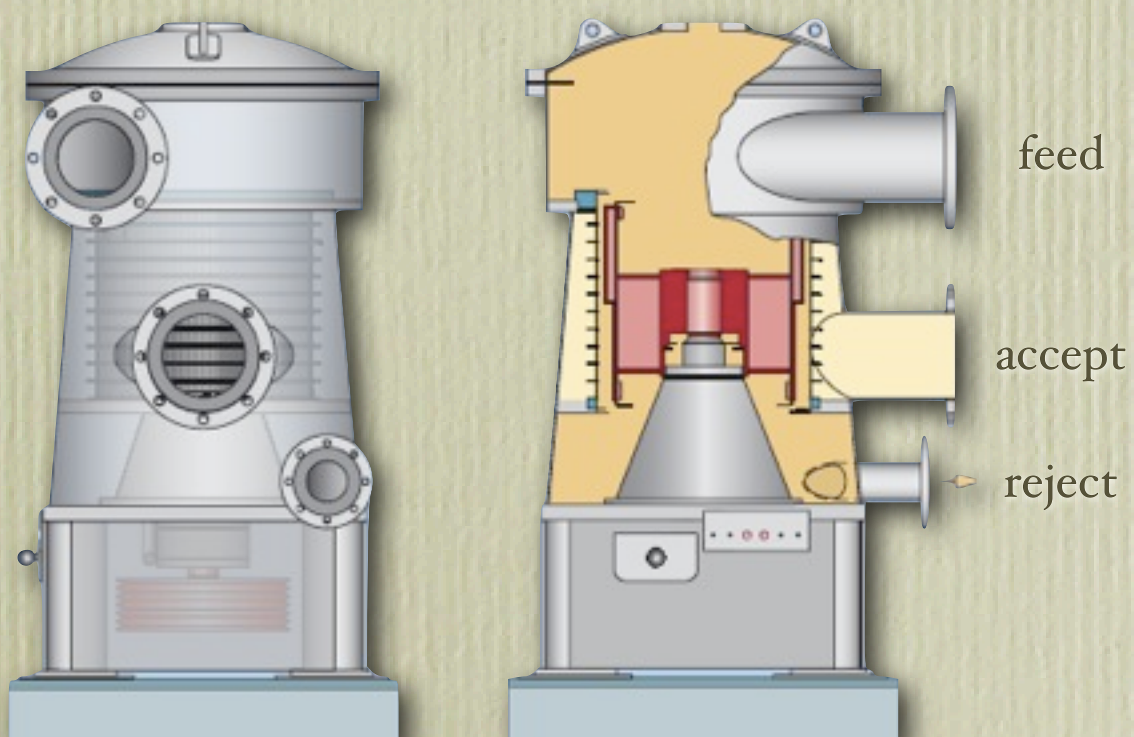
(see Valkama, 2007)

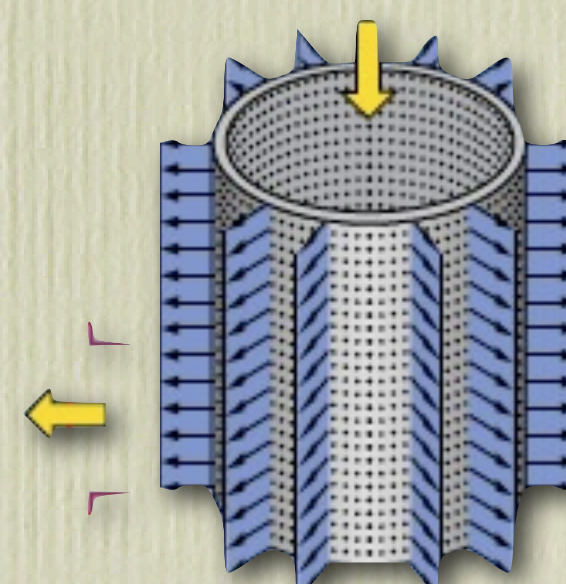
- Recycling fibres from waste paper consists of several steps:
  - Manual removal of contaminent materials.
  - Hackle paper into small pieces.
  - Resolve pieces in water and obtain pulp.
  - **Clean the pulp from** paper clips, plastic materials, and **stickies**.
  - De-ink the pulp.
  - The recovered 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. €.



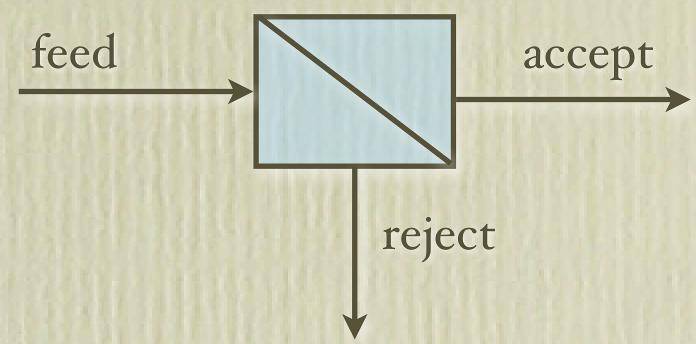
Sticky Sorting in Practice

- 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.



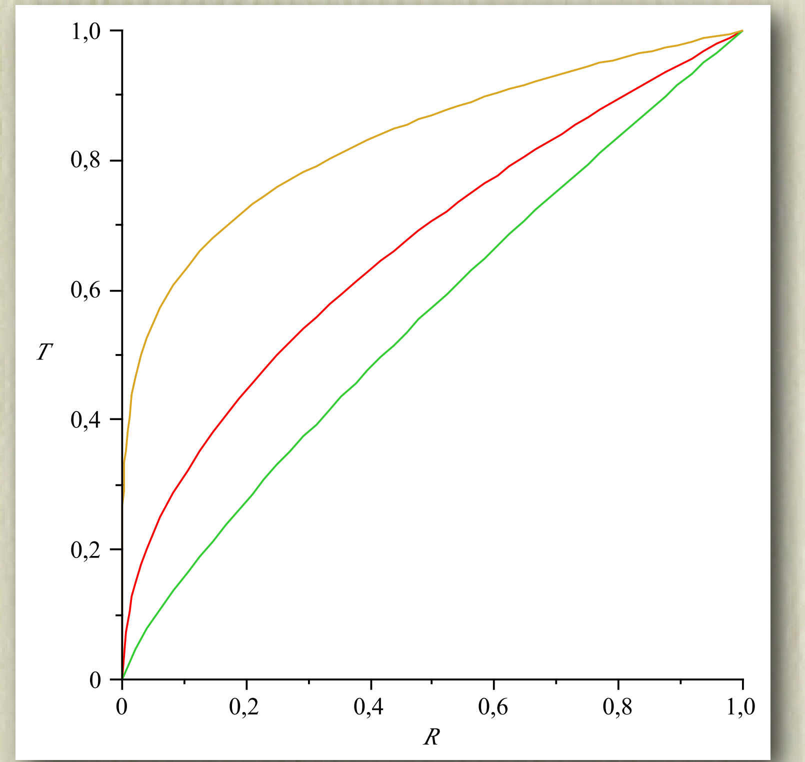


The Plug Flow Model



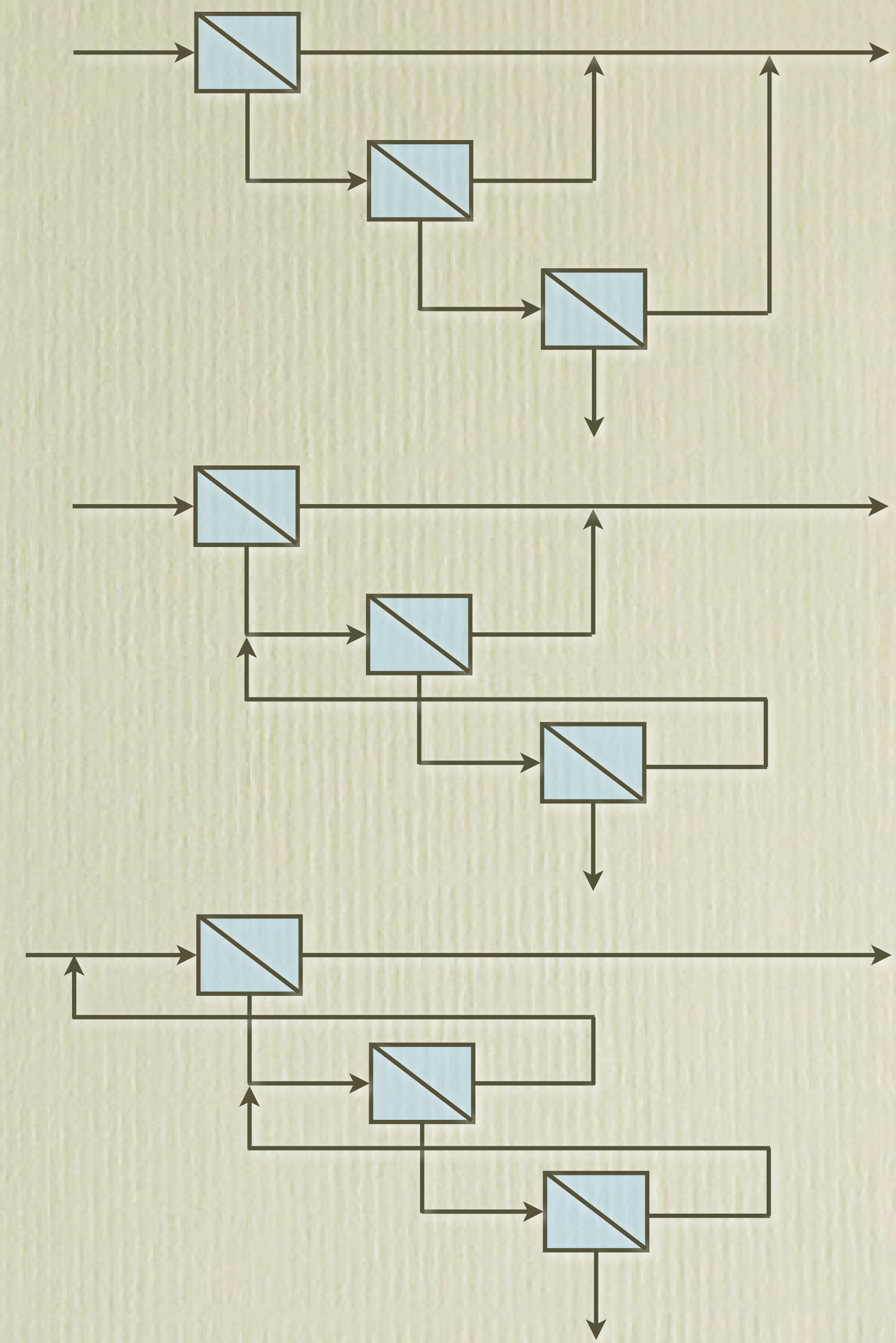
- Each sorter has one inflow feed, and two outflows, accept and reject.
- Mass is conserved: $m^{in} = m^{acc} + m^{rej}$.
- Several components (~12) are in the pulp flow; we restrict here to two, fibers and stickies.
- The separation efficiency for component k is $T_k = \dfrac{m_k^{rej}}{m_k^{in}}$.

  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

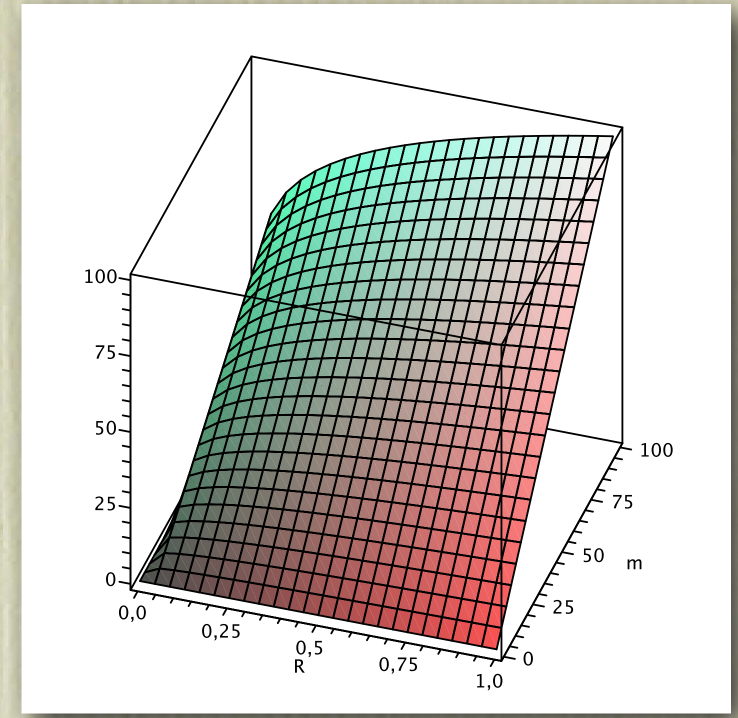
- There are many ways to connect the sorters.

#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.

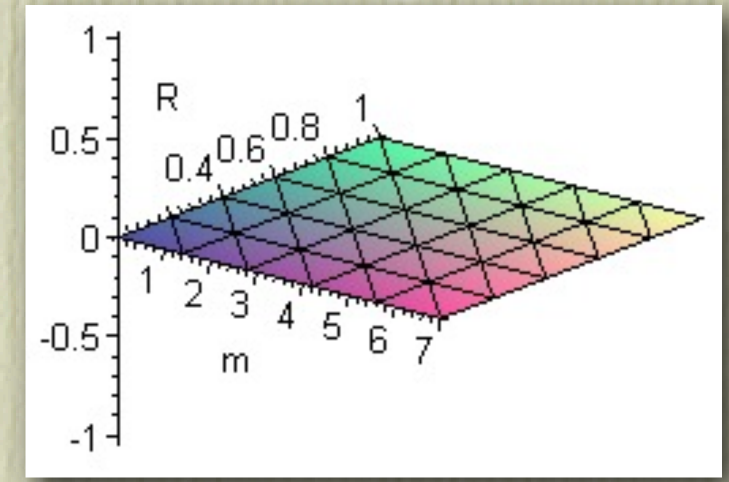
Solving the Model

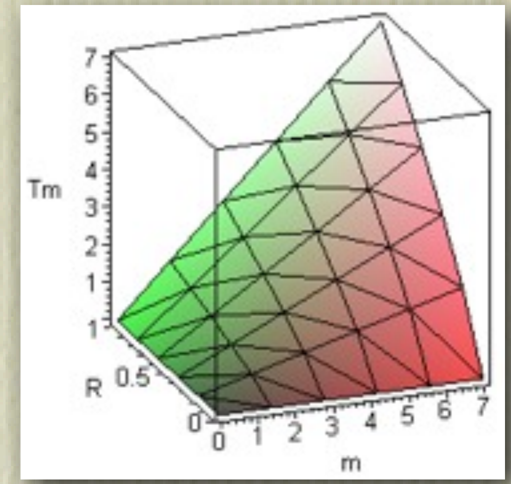
- We want to obtain global optimal solutions.
- Linear programming based branch-and-bound methods can find global optimal solutions.
- But: the model is nonlinear, nonconvex.
- Problematic constraints are $m_{s,k}^{rej} = R_s^{\beta_{s,k}} \cdot m_{s,k}^{in}$, i.e., plug flow.
- To apply them here, the nonlinear constraints have to be approximated by piecewise linear ones.
- We implemented several different approaches.



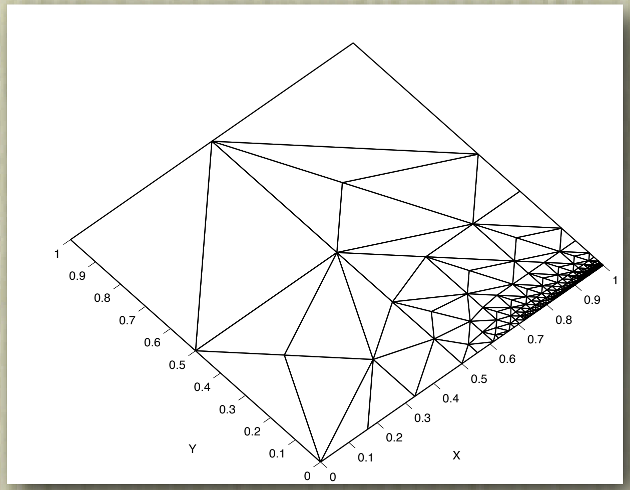
2d Approximation

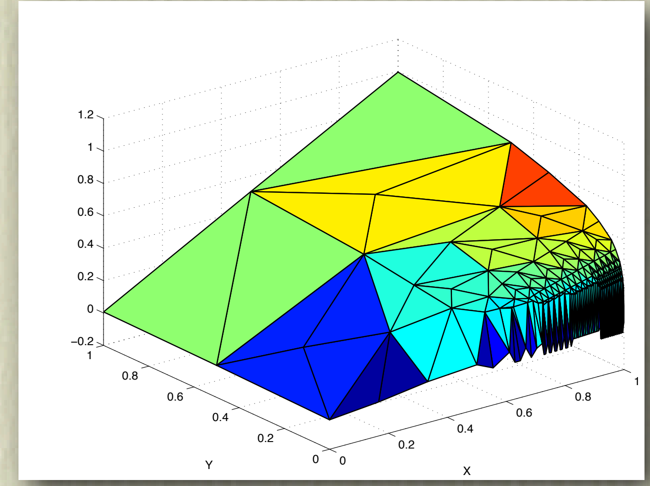
- Approximation by triangulation.
- Equidistant (regular):





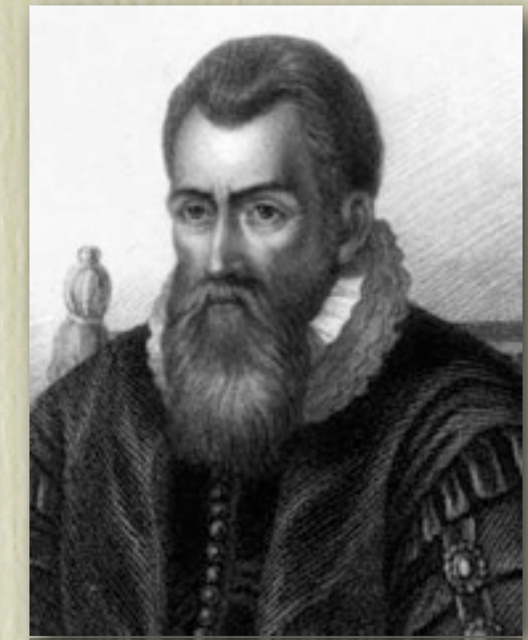
- Irregular:





- Inclusion into the MIP model by higher-dimensional generalization of incremental method (Wilson, 1998) or special ordered sets (Moritz, 2006).

Transformation from 2d to 1d



- We transform the 2d nonlinear function into several 1d functions.
- Using an idea of John Napier (1614)... logarithms!
- From $m_{s,k}^{rej} = R_s^{\beta_{s,k}} \cdot m_{s,k}^{in}$ we thus obtain

$$\log(m_{s,k}^{rej}) = \log(R_s^{\beta_{s,k}} \cdot m_{s,k}^{in}),$$

$$\log(m_{s,k}^{rej}) = \log(R_s) \cdot \beta_{s,k} + \log(m_{s,k}^{in}).$$

- Introduce new variables $\tilde{m}_{s,k}^{rej}, \tilde{R}_s, \tilde{m}_{s,k}^{in}$ and replace the nonlinear constraint $m_{s,k}^{rej} = R_s^{\beta_{s,k}} \cdot m_{s,k}^{in}$ by the following constraint system:

$$\tilde{m}_{s,k}^{rej} = \log(m_{s,k}^{rej}),$$

$$\tilde{R}_s = \log(R_s),$$

$$\tilde{m}_{s,k}^{in} = \log(m_{s,k}^{in}),$$

$$\tilde{m}_{s,k}^{rej} = \beta_{s,k} \cdot \tilde{R}_s + \tilde{m}_{s,k}^{in}.$$

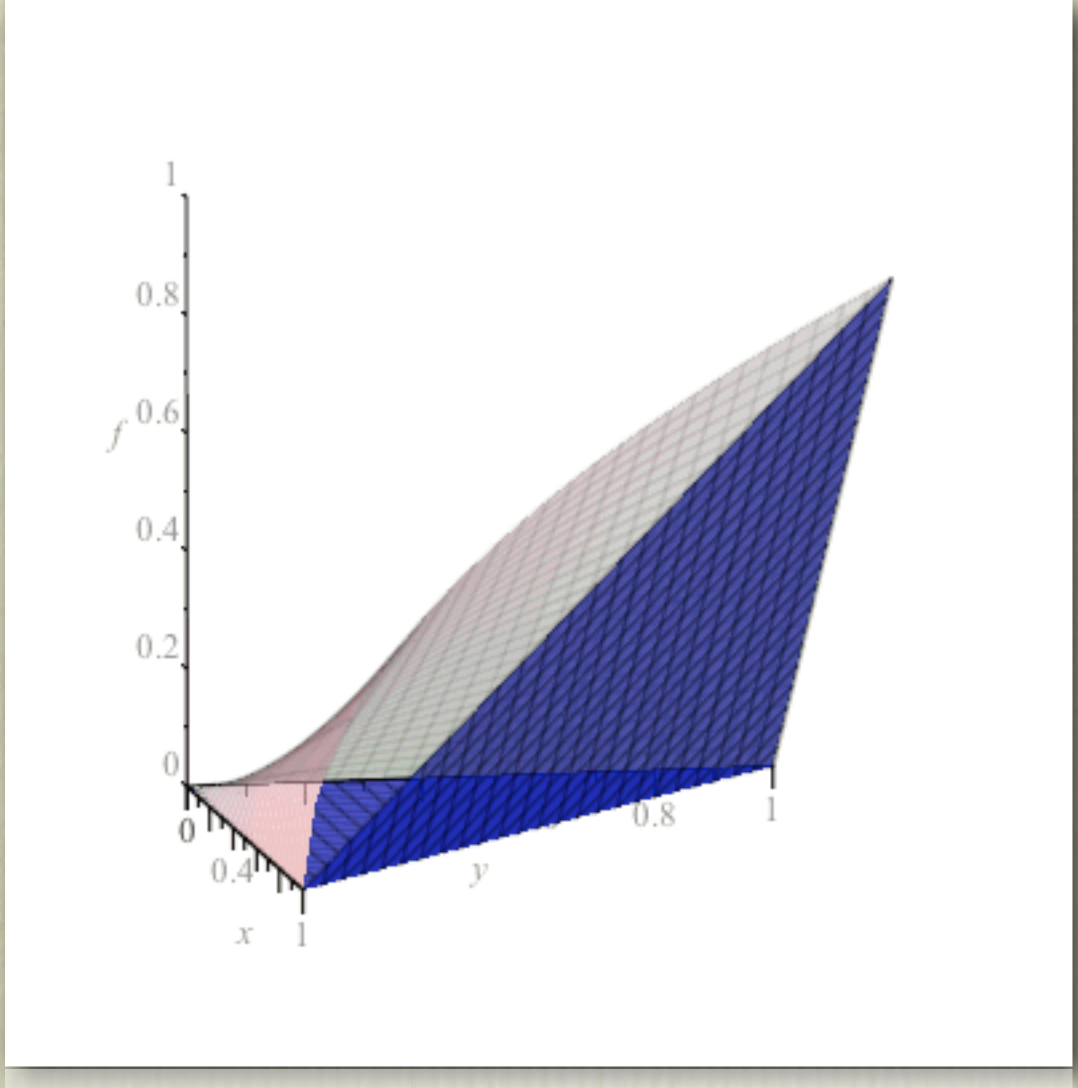
Adaptive Linear Approximation

- The previous 1d and 2d methods have some disadvantages in common:
  - The number of location of interpolants is determined beforehand.
  - The locations are selected might be selected in a nonequidistant way (i.e., more interpolants where the function is „interesting“, less where it is „boring“), but it is not related to the location of the optimal solution.
  - No further adaptation takes place, once the MILP solution process has been started.
- Solution:
  - Start with a coarse linear approximation of the nonlinear function.
  - Refine it during the branch-and-bound solution process by spatial branching and refined linear approximations.

The Convex Envelope

- Bivariate function $f(x,y) := xy^\beta$ with $\beta \in]0,1[$ on $[l_x,u_x]\times[l_y,u_y]$.
- **Convex** envelope (Tawarmalani, Sahinidis 2001, 2002; Benson 2004):

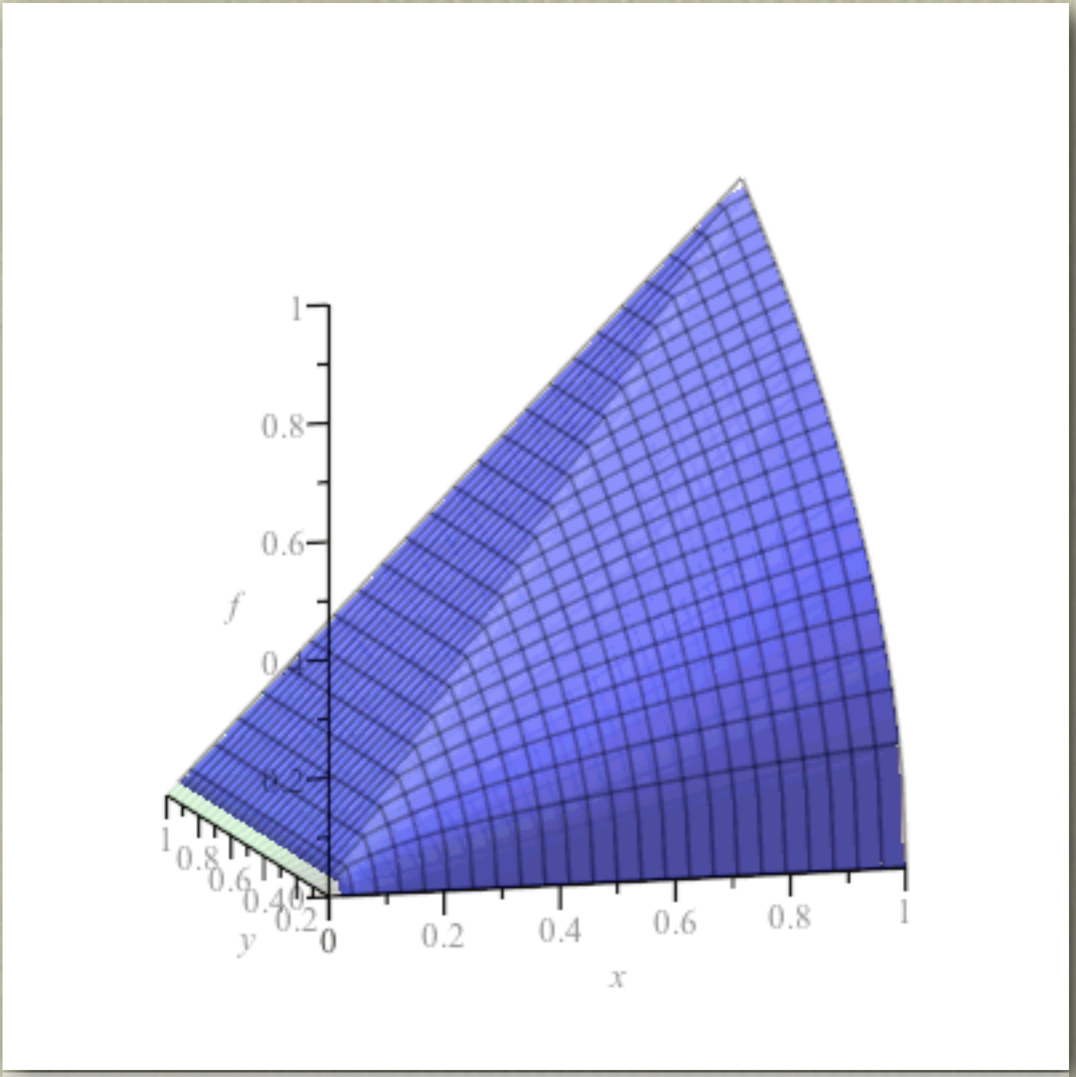
$$\mathrm{vex}_{[l,u]}f(x,y) = \max\Big\{(l_y)^\beta x + l_x\frac{(u_y)^\beta-(l_y)^\beta}{u_y-l_y}y - l_x\frac{(u_y)^\beta-(l_y)^\beta}{u_y-l_y}l_y,$$
$$(u_y)^\beta x + u_x\frac{(u_y)^\beta-(l_y)^\beta}{u_y-l_y}y - u_x\frac{(u_y)^\beta-(l_y)^\beta}{u_y-l_y}u_y\Big\}.$$



The Concave Envelope

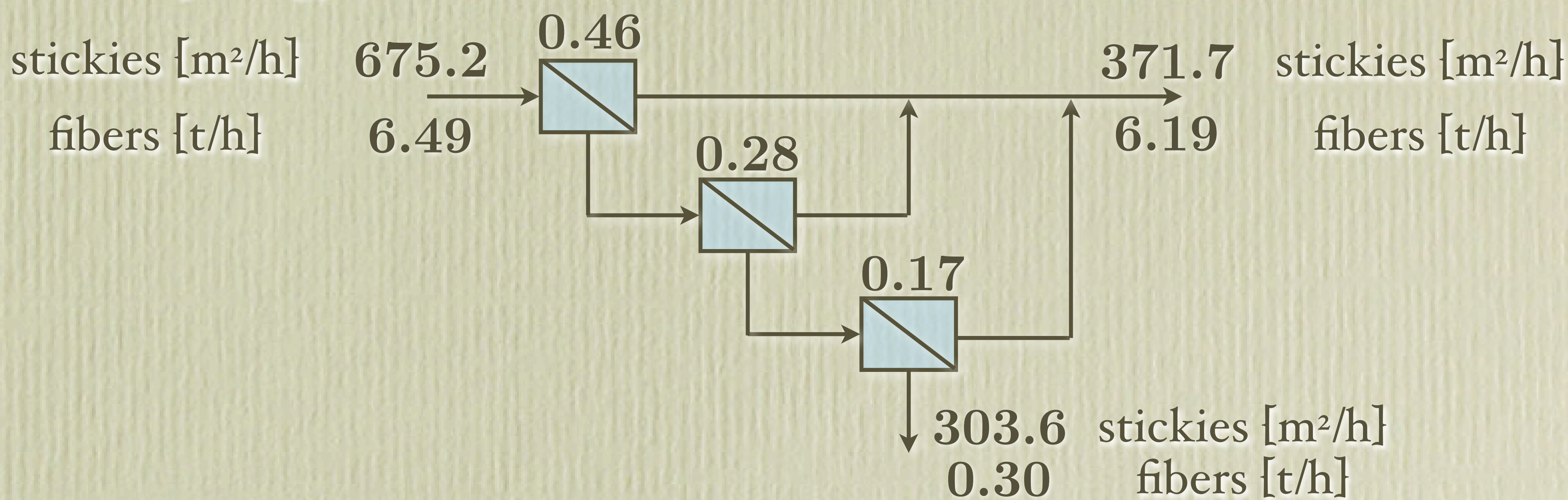
- Bivariate function $f(x,y) := xy^{\beta}$ with $\beta \in]0,1[$ on $[l_x,u_x] \times [l_y,u_y]$.
- **Concave** envelope (Tawarmalani, Sahinidis 2001, 2002; Jach, Michaels, Weismantel 2008; Khagavira, Sahinidis 2013):

$$\mathrm{cave}_{[l,u]} f(x,y) = \begin{cases} x \cdot y^{\beta}, & \text{if } x \in \{l_x, u_x\}, \\ \lambda l_x (r^\star)^{\beta} + \lambda u_x \left(\frac{y}{1-\lambda} - \frac{\lambda}{1-\lambda}\, r^\star\right)^{\beta}, & \text{if } l_x < x < u_x. \end{cases}$$

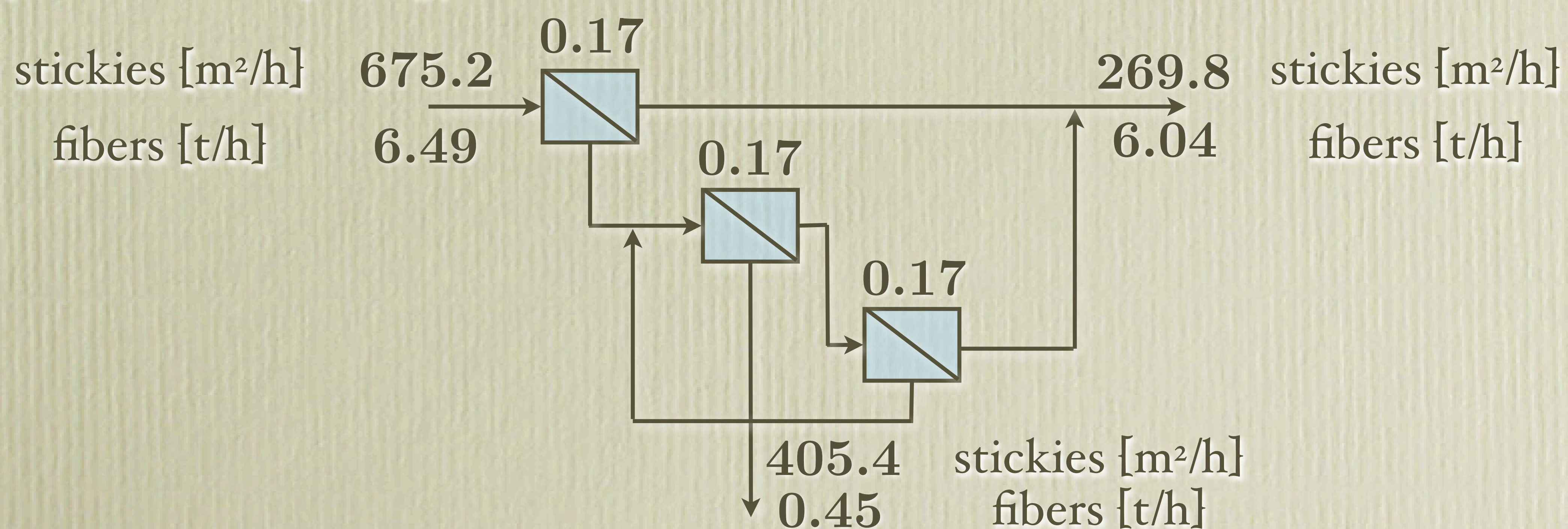


Computational Results

- Objective function: $\sum_{k \in K} (c_k^{acc} \cdot m_k^{acc} + c_k^{rej} \cdot m_k^{rej}) \to \max .$
- Given topology: $obj = 2368.67$



- Optimized topology: $obj = 2519.11$



Runtime Comparison

- SCIP 3.0.1 & SoPlex 1.7.1 on Intel Core i7 CPU @ 2.93 GHz, 16 GB.
- CPU times for different linearization methods:
  - 2d (best of: convex, incremental, sos2, logarithmical)
    - 37 sec (epsilon = 0.1)
    - 15357 sec (epsilon = 0.01)
    - - (epsilon = 0.001)
  - 1d (best of: convex, incremental, sos2, logarithmical)
    - 16 sec (eps = 0.1)
    - 89 sec (eps = 0.01)
    - 278 sec (eps = 0.001)
  - Envelope-cuts & spatial branching
    - 5 sec (eps = 0.0001)

Conclusions

- There are many ways to handle nonlinear functions within a branch-and-cut framework.
- If CPU time matters:
  - **Never** use additional binary variables!
  - **Never** work with a fixed approximation of the nonlinear functions!
  - **Always** compute concave/convex over/underestimators, and adaptive refinements during the branch-and-bound search!

- Further informations:
  F., Hayn, Michaels.
  Mixed-Integer Linear Methods for Layout-Optimization of Screening Systems in Recovered Paper Production.
  Optimization and Engineering, 2014.
  DOI: 10.1007/s11081-014-9249-7