Chapter 2

Mixed Integer Nonlinear Programming Applications

Josef Kallrath
BASF-AG, ZX/ZC-C13, D-67056 Ludwigshafen, Germany
e-mail*: josef.kallrath@basf-ag.de*

2.1 Introduction

In this contribution we apply different approaches to solve four rather different MINLP problems: special extensions to time-indexed formulations of production planning problems, a production planning problem in BASF's petrochemical division, a site analysis of one of BASF's bigger sites, and a process design problem. The first problem is related to a useful nonlinear extension of production planning problems based on time-indexed formulations. The second problem[1] leads to a mixed-integer nonlinear model for describing a petrochemical network including several steam crackers and plants located at two different sites. The third problem, a network design problem, leads to mixed integer nonlinear programming problem dominated by pooling problems (KW97, Section 11.1.2) as is true for the second problem. The pooling problem refers to the intrinsic nonlinear problem of forc-

[1]**Acknowledgments**: Thanks is directed to Christian Timpe (Section 3) and Norbert Vormbrock (Section 4 and 5) who provided material or joined the author in the project work. Eventually, this article benefited greatly from suggestions and comments provided by Anna Schreieck and Beate Brockmüller.

ing the same (unknown) fractional composition of a multi-component streams emerging from a pool, *e.g.*, a tank or a splitter in a petrochemical network. The fourth problem is concerned with a process design problem in which some process parameters and the topology of a system of chemical reactors are the degrees of freedom to optimize total production, selectivity, energy, and costs. All numercial experiments and production runs have been carried out on a 166 MHz Pentium processor.

The covered models lead to mixed integer nonlinear programming (MINLP), *i.e.*, to optimization problems of the form

$$\min \left\{ f(\mathbf{x},\mathbf{y}) \;\middle|\; \begin{array}{ll} \mathbf{g}(\mathbf{x},\mathbf{y}) = 0 & \mathbf{x} \in \mathbb{R}^{n_c} \\ \mathbf{h}(\mathbf{x},\mathbf{y}) \geq 0 & , \quad \mathbf{y} \in \mathbb{Z}^{n_d} \end{array} \right\} , \qquad (2.1.1)$$

with n_c continuous and n_d discrete variables, nonlinear objective function $f(\mathbf{x},\mathbf{y})$ and constraints $\mathbf{g}(\mathbf{x},\mathbf{y})$ and $\mathbf{h}(\mathbf{x},\mathbf{y})$. Problems such as (2.1.1) are very difficult to solve. They belong to the class of $\mathcal{NP}$-complete problems. An overview of algorithms capable of solving such problems is given in Leyffer (1993), Floudas (1995) or Kallrath and Wilson (1997; hereafter KW97, Section 11.4).

Many mixed integer linear programming (MILP) problems are combinatorial optimization problems for which the Branch&Bound algorithm (Nemhauser and Wolsey, 1988) based on linear programming (LP) relaxation proves sufficiently efficient. The algorithm is deterministic but in the worst case we see complexity growing exponentially in the problem size. Nonlinear programming (NLP) problems force us to distinguish between local and global optima. Algorithms to solve NLP problems [see, for instance, Bazaraa *et al.* (1993)] have their roots in calculus and depend on the concept of convergence. Except in special cases it is not possible to prove that an NLP algorithm converges to the global optimum. So we should keep in mind that a) in non-convex NLP or MINLP problems we cannot strictly prove optimality or provide safe bounds, and b) that if nonlinear equations are present the NLP or MINLP is immediately non-convex.

Unfortunately, MINLP problems combine the difficulties of both its subclasses: MILP and NLP. Even worse, in addition they have properties absent in NLP or MILP. While for convex NLP problems a local minimum is identical to the global minimum, we find that this result does not hold for MINLP problems. The best we can do in non-convex MINLP is to provide a safe bound or relative optimality with respect to a certain local optimum of the continuous problem.

2.2 Nonlinear Extensions to Production Network Models

This section describes an extension which can be added to any planning model based on time-indexed formulations with at most one setup-change per period. Here we apply this approach to a special case and extend the production planning problem and its model (M1, for brevity) described in KW97 (Section 10.4). As already discussed in KW97 (Chapter 6) certain nonlinear terms [absolute value terms, products of binary variables] can be expressed by linear relations involving additional binary variables. The key idea used in the current production planning problem is to replace products of continuous variables and binaries, theta-functions and absolute value terms by linear relations involving additional binary variables.

2.2.1 Batch Constraints Across Periods

The motivation for the model approach developed in Section 2.2.2 has its root in batch or campaign production in the chemical process industry. Batch production operates in integer multiples of batches where a batch is the smallest unit to be produced, *e.g.*, 200 tons. Several batches following each other immediately establish a campaign. Some typical batch restrictions group batches into campaigns, or consider that only campaigns of a minimal size can be produced. The batch reactors can be, for example, operated in different modes producing several products in each mode with different free or fixed recipes leading to a general mode-product relation (KW97, pp.153-155, 320-324): in a certain mode several products are produced (with different daily production capacity rates), and vice-versa, a product can be produced in different modes. Daily production can be less then the capacity rates. Within a fixed planning horizon, T, a certain product can be produced in several campaigns. In the context of time-indexed formulation where variables p_{pt} describe the production [*e.g.*, in tons] of a product p in period (time-interval) t it is not easy to model such batch restrictions if the batch or minimal campaign size is larger than the capacity per period. Assume that production is performed in batches of 200 tons, and that our time intervals have a length of ten days with a daily production rate of 10 tons/day. The minimum time to produce the batch would cover 20 days, or exactly two time intervals. A plan looking like $p_{p4} = 45$ tons, $p_{p5} = 100$ tons, and $p_{p6} = 55$ tons covers three periods (the first and third only partial) to produce exactly 200 tons, and thus provides more degrees of freedom. Brockmüller and Wolsey (1995) solved the problem for a special case (production equals the capacity rates). Their approach

uses explicitly the feature that production equals the capacity rates in order to compute a priori the number of periods to produce a campaign of specified minimal size. If daily production can take any value between zero and the capacity rate, or if a product is produced, for example, according to general mode-product relation, then this a priori information is not available. Our approach does not depend on this a priori information and can be used for more general cases.

2.2.2 Formulation of Batch Constraints

Our goal is to compute the amount, p^C_{rpnt}, of product $p \in \mathcal{P}$ produced for a certain campaign n in period $t \in \mathcal{T}$. The mathematical model is, for a certain site, unit, or reactor $r \in \mathcal{R}$, based on some binary state variables δ^P_{rpt} indicating whether product p is produced on r in period t, and binary start-up variables δ^S_{rpt} indicating whether the production of p is started in period t on r. Let P^-_{rpt} and P^+_{rpt} be bounds on p_{rpt} if $p_{rpt} > 0$. We may choose the upper bound P^+_{rpt} for p_{rpt}, *e.g.*, as the length of the period (in days) times the daily production capacity, and the lower conditional bound $P^-_{rpt} = 0.8P^+_{rpt}$.

Let us, at first, connect δ^P_{rpt} to the production variables p_{rpt} starting with the inequalities

$$p_{rpt} \leq P^+_{rpt}\delta^P_{rpt} \quad , \quad \forall\{rpt\} \quad . \tag{2.2.1}$$

If Δ_{rp} tells us whether product p is produced at the beginning of the first period, and $\sum_p \Delta_{rp} = 1$, then for the first period we have

$$P^-_{rp1}\delta^P_{rp1} - P^+_{rp1}(1-\Delta_{rp}) - P^+_{rp1}\left(1-\delta^P_{rp2}\right) \leq p_{rp1}, \quad \forall\{rp\} \quad , \tag{2.2.2}$$

and for all other period (except the last one) $\mathcal{T}_1 := \{2, \ldots, T-1\}$

$$P^-_{rpt}\delta^P_{rpt} - P^+_{rpt}\delta^S_{rpt} - P^+_{rpt}\left(1-\delta^P_{rpt+1}\right) \leq p_{rpt}, \quad \forall\{rpt \in \mathcal{T}_1\} \quad . \tag{2.2.3}$$

The inequalities (2.2.1) to (2.2.3) hold the *positivity conditions* ($\delta^P_{rpt} = 0 \Leftrightarrow p_{rpt} = 0$) and ($\delta^P_{rpt} = 1 \Leftrightarrow P^-_{rpt} \leq p_{rpt} \leq P^+_{rpt}$) for all inner periods of a campaign. The second and third term on the left-hand side of (2.2.3) ensure that the positivity conditions is not applied to the first and last period of campaigns.

Now we need to relate the start-up variables to the state variables. This part depends on the problem considered. A formulation, valid for any continuous variable (*e.g.*, the production variable p_{rpt} or the variable m^D_{rmt} denoting the time spent in mode m both used in M1) subject to constraints cross periods, and the conditions that we can produce only one product per time and that at most two products can be produced during one period (*i.e.*, at most one setup-change

per period), needs to represent the following set of implications for δ^S_{rpt}:

$$\begin{array}{c|cc} \delta^P_{rpt-1} \backslash \delta^P_{rpt} & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 0 & \mu_{rpt} \end{array} \quad , \tag{2.2.4}$$

with

$$\mu_{rpt} := \left\{ \begin{array}{l} 1, \text{ if any other production } p' \neq p \text{ is started in period } t-1 \\ 0, \text{ if no other production started in period } t-1 \end{array} \right. \tag{2.2.5}$$

These rules, for $\delta^P_{rpk} + \delta^P_{rpk-1} \neq 2$ are enforced by

$$\delta^S_{rpt} = \delta^P_{rpt} \quad , \quad \forall\{rp\} \quad , \quad t = 1 \quad , \tag{2.2.6}$$

for the first period, and for all other periods $\mathcal{T}_T := \{2, \ldots, T\}$ by

$$\delta^S_{rpt} \leq \delta^P_{rpt} \quad , \quad \delta^S_{rpt} \geq \delta^P_{rpt} - \delta^P_{rpt-1} \quad , \quad \forall\{rpt \in \mathcal{T}_T\} \quad . \tag{2.2.7}$$

The case $\delta^P_{rpk-1} = \delta^P_{rpt} = 1$ is properly described by additional inequalities

$$\delta^S_{rpt} \geq -2 + \sum_{p' \neq p} \delta^S_{rp't-1} + \delta^P_{rpt} + \delta^P_{rpt-1} \quad , \quad \forall\{rpt \in \mathcal{T}_T\} \quad , \tag{2.2.8}$$

and

$$\delta^S_{rpt} \leq 2 + \sum_{p' \neq p} \delta^S_{rp't-1} - \delta^P_{rpt} - \delta^P_{rpt-1} \quad , \quad \forall\{rpt \in \mathcal{T}_T\} \quad . \tag{2.2.9}$$

If, in a general mode-product relation several products can be produced simultaneously, we may want that the case $\delta^P_{rpt-1} = \delta^P_{rpt} = 1$ also leads to $\delta^S_{rpt} = 0$; this can easily be realized by neglecting the second term on the right-hand sides of (2.2.8) and (2.2.9). Alternatively, we may require that $\mu = 1$ if any other more complicated rule than the above (2.2.6) is fulfilled.

Let us from now on assume that δ^P_{rpt} and δ^S_{rpt} are available. The production of product p may start in several time periods, *i.e.*, we have several product-p-campaigns within the planning horizon T. Therefore we introduce continuous variables, $c_{rpt} \geq 0$, counting the number of start-ups and related to the start-up variables δ^S_{rpt} by

$$c_{rp1} = \delta^S_{rp1} \quad , \quad \forall\{rp\} \quad ; \quad c_{rpt} = c_{rpt-1} + \delta^S_{rpt} \quad , \quad \forall\{rpt \in \mathcal{T}_T\} \quad . \tag{2.2.10}$$

Now we introduce continuous variables ν_{rptn} indicating whether a certain campaign, $n \in \mathbb{N}_0$, could be active ($c_{rpt} = n$) or not, *i.e.*,

whether c_{rpt} is equal to a certain fixed integer $n \in \mathbb{N}_0$, or not. ν_{rptn} represents the nonlinear function

$$\nu_{rptn} = 1 - \theta\left(|c_{rpt} - n|\right) \quad , \quad \theta(x) := \begin{cases} 1, \text{ if } x > 0 \\ 0, \text{ if } x = 0 \end{cases} \quad . \tag{2.2.11}$$

Let us assume that at most $N^+_{rp} \in \mathbb{N}_0$ campaigns of product p can be produced within the planning horizon; a typical value in the current planning problem is $N^+_{rp} = 6$. A special case is $N^+_{rp} = 1$ enforcing that a product can be produced in only one campaign.

The relation (2.2.11) is enforced by

$$1 = \sum_{n=0}^{N^+_{rp}} \nu_{rptn} \quad \text{and} \quad \sum_{n=0}^{N^+_{rp}} n\nu_{rptn} = c_{rpt} \quad , \quad \forall\{rpt\} \quad , \tag{2.2.12}$$

i.e., one campaign has to be chosen in any case (possibly, the "0" campaign), and if campaign n is selected then $c_{rpt} = n$. The sets

$$S_{rpt} := \left\{ \nu_{rptn} \,\middle|\, 0 \le n \le N^+_{rp} \right\} \quad , \quad \forall\{rpt\} \tag{2.2.13}$$

form a special ordered set of type 1. We use the second equation of (2.2.12) as the reference row for efficient branching.

The total amount, p^C_{rpn}, of product p produced within campaign n is given by

$$p^C_{rpn} = \sum_{t=1}^{N^T} p^C_{rptn} \quad , \quad \forall\{rp\} \quad , \quad \forall n \in \mathcal{N}_1 := 1, \ldots, N^+_{rp} \quad , \tag{2.2.14}$$

where p^C_{rptn} is the amount of product p produced for campaign n in period t, *i.e.*,

$$p^C_{rptn} = p_{rpt}\nu_{rptn} \quad , \quad \forall\{rptn \in \mathcal{N}_1\} \quad . \tag{2.2.15}$$

Applying the formalism described in Section 2.2.3 with $K = 1$ we replace (2.2.15) by

$$\begin{aligned} p^C_{rptn} &\le P^+_{rpt}\nu_{rptn} \quad , \quad p^C_{rptn} \le p_{rpt} \quad , \\ p^C_{rptn} &\ge p_{rpt} - P^+_{rpt} + P^+_{rpt}\nu_{rptn} \quad , \quad \forall\{rptn\} \quad . \end{aligned} \tag{2.2.16}$$

With the formalism at hand described above we reached our goal: the computation of the amount, p^C_{rpn}, of product p produced for campaign n. p^C_{rpn} may be now subject to specific batch constraints, *e.g.*, a campaign may just consist of one single batch of fixed batch size B_{rp},

$$p^C_{rpn} = B_{rp} \quad , \quad \forall\{rpn \in \mathcal{N}_1\} \quad . \tag{2.2.17}$$

Alternatively, campaigns may be built up by a discrete number of batches following each other immediately, *i.e.*,

$$p^C_{rpn} = B_{rp}\beta_{rpn} \quad , \quad \forall\{rpn \in \mathcal{N}_1\} \quad , \tag{2.2.18}$$

where the integer variable β_{rpn} indicates the number of batches of size B_{rp} within campaign n. Finally, p^C_{rpn}, may behave like a semi-continuous variables, *i.e.*,

$$p^C_{rpn} = 0 \quad \text{or} \quad C^-_{rp} \leq p^C_{rpn} \leq C^+_{rp} \quad , \quad \forall\{rpn \in \mathcal{N}_1\} \quad , \tag{2.2.19}$$

where C^-_{rp} and C^+_{rp} are lower and upper bounds if production takes place.

2.2.3 Modeling Product Terms Including One Continuous & Several Binary Variables

To model products like $x\Pi^K_{k=1}\delta_k$, where δ_k are binary variables and x is any kind of non-negative variable, let us assume that X^+ is a valid upper bound on x. The product $\Pi^K_{k=1}\delta_k$ is exactly represented by the variable y subject to the inequalities

$$\forall k : y \leq X^+\delta_k \quad , \quad y \leq x \quad , \quad y \geq x - X^+\left(K - \sum_{k=1}^{K}\delta_k\right) \quad . \tag{2.2.20}$$

The first inequality of (2.2.20) has the implications ($\delta_k = 0 \Rightarrow y = 0$) and ($y > 0 \Rightarrow \sum^K_{k=1}\delta_k = K$), while the second and third inequality give us ($\sum^K_{k=1}\delta_k = K \Rightarrow y = x$) and ($y = 0 \Rightarrow \sum^K_{k=1}\delta_k < K$). Note that if we want to know the product $y = x\Pi^K_{k=1}\delta_k$ explicitly we do not need to introduce an extra variable.

2.2.4 Implementation and Results

If we want to add the batch constraints in Section 2.2.2 to the production planning model M1[2], it is not strictly necessary to use (2.2.2)-(2.2.9) to compute δ^P_{rpt} and δ^S_{rpt}. Alternatively, we can derive δ^P_{rpt} and δ^S_{rpt} from the mode state variables α_{rmt} and start-up variables β_{rmt} used in the model M1 by Kallrath *et al.* (1994) and KW97 (pp.320-324). If $\mathcal{P}$ is the union of disjunctive sets $\mathcal{P}_m$ of products produced in mode m and I_{rmp} indicates whether product p can be produced in

[2]Although the problem instance specified by the data leads to a one-to-one relation between modes and products, the coupling $p_{rpt} \leq \sum_m R_{mp}m^D_{rmt}$ with production rates R_{mp} holds for any mode-product relation. Note that in M1 $r \leftrightarrow i$, and $t \leftrightarrow k$.

mode m on reactor r, (in the current case we have even $\sum_p I_{rmp} = 1$, *i.e.*, exactly one product per mode) we just have

$$\delta^P_{rpt} = \sum_{m|I_{rmp}=1} \alpha_{rmt} \quad , \quad \delta^S_{rpt} = \sum_{m|I_{rmp}=1} \beta_{rmt} \quad , \quad \forall\{rpt\} \quad . \tag{2.2.21}$$

This *special approach* based on (2.2.21) is, however, only exactly identical with the more *general approach* based on (2.2.6)-(2.2.9) if $P^-_{rpt} = 0$.

For the model M1 and a typical reference scenario (S_1) covering 12 to 36 production time periods we have used both approaches indicated by indices s and g to derive production plans maximizing total sales. The scenarios S_2 use (2.2.19) to model campaigns whose minimum size is 300 tons. The scenarios S_3 include 49 partial integer variables and use (2.2.18) to enforce that campaigns are built up by discrete batches of 100 tons each. Finally, in scenario S_m we require that if a certain mode is chosen the plant has to stay in that mode for at least 3 days. In this case, the variables m^D_{imk} used in KW97 (pp.320-324), expressing how much time the plant at site i spends in mode m in period k, play the role of p_{rpt} used above; the length of the period (10 to 30 days) is a useful upper bound on m^D_{imk}. Using Dash's MILP-solver `XPRESS-MP` 10.05 (Ashford and Daniel, 1987, 1991), we got the following results (including the number of continuous, binary and semi-continuous variables, constraints, integer solution, number of nodes n_n, running time τ, and gap Δ in percent) when we applied the formalism to all possible reactor(site)-product(mode)-time combinations:

	P^-_{rpt}	n_c	b	$s-c$	c	IP	n_n	τ	Δ
S_1	—	12397	2973	1608	8441	1	440	8^{m}	1.9
S_1	—					2	960	$+6^{\mathrm{m}}$	1.4
S_1	—					3	1721	$+8^{\mathrm{m}}$	1.0
S_{2s}	—	14833	2973	1650	13997	1	786	52^{m}	19.4
S_{2g}	1	15217	2973	1650	14033	1	858	59^{m}	28.8
S_{2g}	$0.8P^+_{rpt}$	15217	2973	1650	14033	2	3860	$5^{\mathrm{h}}59^{\mathrm{m}}$	31.9
S_{3s}	—	15687	2973	1608	13855	1	632	58^{m}	3.9
S_{3g}	1	16215	2973	1608	15681	1	907	$1^{\mathrm{h}}09^{\mathrm{m}}$	16.5
S_{3g}	$0.8P^+_{rpt}$	16215	2973	1608	15681	3	18272	$39^{\mathrm{h}}02^{\mathrm{m}}$	4.6
S_m	—	15333	2973	1650	15681	1	511	28^{m}	4.6
S_m	—					3	1943	$+40^{\mathrm{m}}$	1.8
S_m	—					4	3972	$+2^{\mathrm{h}}44^{\mathrm{m}}$	1.5

The use of special ordered sets of type 1 for the variables ν_{rptn} is essential; the model contains 192 sets and 1080 set members. In previous versions when these variables were declared as binary variables computing times were much larger. In the S_3 runs (multiple batches), the variables β_{rpn} were declared as partial integers (integer below 10, continuous above 10). Although the variables δ^S_{prt} become binary automatically, it is advantageous to declare them as binary variable explicitly because that enables us to prioritize them and to

improve branching. The use of directives in the model was crucial. In the general approach scenarios n_n, τ, and the quality of the solution indicated by Δ depended critically on P^-_{rpt}. Note that the run for $P^-_{rpt} = 0.8P^+_{rpt}$, (*i.e.*, high utilization rates of the plant system), shows the third integer solution found and required much more computing time.

The benefit achieved by the extended model features is qualitative because it leads to an improved representation of the real world process. The production plans do not suffer any longer from the time-indexed formulation and look more stable avoiding small campaigns and many setup-changes. In practical planning runs it is sufficient to use the formalism only for a few products or modes, and sometimes only for one site or reactor. Thus, the Pentium 166 MHz computing time reduces to less than 15 minutes and becomes similar to the one of the reference scenario S_1.

2.3 Production Planning in a Petrochemical Production Network

This section describes the mathematical model of a petrochemical production network. The network includes 5 plants [2 steamcrackers, 3 units to extract certain fractions] with about 30 products (or streams) in Ludwigshafen, and 7 plants with about 30 products in Antwerp. The crackers can be operated in continuously varying modes defined by the cracking severity. The modes of operation are modeled through interpolation between three predefined values (sharp, medium, mild). For a fixed mode of operation incoming and outgoing flows are linearly coupled by yield coefficients ($[t/t]$). Plants are subject to capacity restrictions (upper and lower bounds in $[t/h]$); for some plants there also exist lower or upper bounds for the relative weight of products in blends (cuts). In every time period only limited amounts of raw materials are available. Streams from intern or extern sources are treated similarly. Utilities are also treated as products.

Every raw material or finished product can be transferred between sites; utilities, representing different types of energy (*e.g.*, heating gas, electricity), cannot. It is possible to define different modes of transportation with different prices for every material. There exist minimal transport limits, below which transport is unreasonable. The solver can choose whether to transport nothing or an amount between the minimal limit, M_p^{TU} (t/h), and maximal transport capacity, M_p^{TL} (t/h).

The objective function is to maximize the net operating margin. Model output is the full feed stock information for all plants, the net

operating margin for both networks and shadow prices. The multi-period model can be used for monthly and 6 months planning.

2.3.1 Foundations of the Mathematical Model

2.3.1.1 Indices, Index sets, Sets and Projectors

We use the following indices and set of indices:

c	$\in \{1,2,3\}$	$= \mathcal{C} \subset \mathcal{N}$	crackers	$\lvert\mathcal{C}\rvert$	$=$	3
d	$\in \{d_1,\ldots,d_{N^D}\}$	$= \mathcal{D}$	markets	$\lvert\mathcal{D}\rvert$	$\sim$	5
m	$\in \{m_1,\ldots,m_{N^M}\}$	$= \mathcal{M}$	cracker modes	$\lvert\mathcal{M}\rvert$	$=$	3
n	$\in \{n_1,\ldots,n_{N^N}\}$	$= \mathcal{N}$	nodes	$\lvert\mathcal{N}\rvert$	$\sim$	20
o	$\in \{o_1,\ldots,o_{N^o}\}$	$= \mathcal{O}$	processes	$\lvert\mathcal{O}\rvert$	$\sim$	80
p	$\in \{p_1,\ldots,p_{N^P}\}$	$= \mathcal{P}$	products	$\lvert\mathcal{P}\rvert$	$\sim$	40
r	$\in \{r_1,\ldots,r_{N^R}\}$	$= \mathcal{R}$	resources	$\lvert\mathcal{R}\rvert$	$\sim$	30
s	$\in \{s_1,\ldots,s_{N^S}\}$	$= \mathcal{S}$	sites	$\lvert\mathcal{S}\rvert$	$=$	2
t	$\in \{1,\ldots,N^T\}$	$= \mathcal{T}$	time periods	$\lvert\mathcal{T}\rvert$	$=$	6

Processes, *e.g.*, cracking of LPG in a certain mode of operation or production of Benzol through extraction from the blend BTX, are represented by nodes (called *submodels* in the PIMS[3] nomenclature). Because additional equations are defined in the crackers, special indices are defined for them. We use the term *products* for all streams considered in the planning model.

Different modes of operation are possible only in the crackers. Therefore the indices for the modes are not used and instead three different processes are defined. They correspond to the three cracker modes mentioned above. Resources stand for capacity restrictions. It is possible to share resources among processes. Usually resources are identical to certain minimal or maximal node flow constraints. We do not introduce an additional index for transport because we use only one.

For simplification of description we introduce the following sets:

$\mathcal{B}$	blends
$\mathcal{L}$	stored streams
$\mathcal{O}_{mc}$	processes belonging to mode m in cracker c
$\mathcal{P}_{\mathrm{F}}$	finished products
$\mathcal{P}_{\mathrm{T}}$	products transported between sites
$\mathcal{P}_{\mathrm{R}}$	raw materials purchased
$\mathcal{P}_{\mathrm{P}}$	finished products purchased

.

Additionally we define the following projectors:

[3]The model is formulated and solved by PIMS (Module PPIMSXX and XPIMS) by Bechtel Corp. (Houston, US); now ASPEN Tech.

$\mathcal{P}^P_{p_b}$ the blended products $p \in \mathcal{P}$ for every blend (cut) $p_b \in \mathcal{B}$.
$\mathcal{P}^O_n$ the corresponding processes $o \in \mathcal{O}$ for every node $n \in \mathcal{N}$.
$\mathcal{P}^O_b$ the corresponding processes $o \in \mathcal{O}$ for $b \in \mathcal{B}$,
$\mathcal{P}^N_b$ the corresponding nodes $n \in \mathcal{N}$ for $b \in \mathcal{B}$,
$\mathcal{P}^O_s$ the processes at site $s \in \mathcal{S}$.

Finally, we define the length of periods. For every period its length is given by F_t (h/month). This increases model flexibility, because it provides an easy way to change from monthly to weekly planning. Additionally different lengths of months can be incorporated.

2.3.1.2 Variables

We use the following set of variables:

b_{pp_bts}	Blending of product p [t/h] into product p_b in period t
e_{pts}	Amount purchased [t/h] of product p at site s in period t
f_{ot}	Flow [t/h] of process o in period t; process flow variable
i_{pts}	Stock [t] of product p ($p \in \mathcal{L}$) at the end of period t
q^C_{pts}	Collector for pool p in period t at site s
$q^D_{p_bnts}$	Flow of pool p_b into node n in period t
r_{pp_bts}	Recursion error for the amount of product p in product p_b
s^j_{its}	Variable for interpolation of severity ($i, j \in \{1, 2, 3\}$)
$t_{ps_is_jt}$	Transport of product p from site s_i to site s_j in period t
v_{ptsd}	Sold [t/h] product p at site s in period t on market d
$\lambda_{ps_is_jt}$	Binary variable for transport of product p from s_i to s_j

2.3.2 The Mathematical Model

The model consists of a network of nodes connected by appropriate balance equations defining and representing the topology. The *flow balances*

$$e_{pts} + t_{ps_js_it} - t_{ps_is_jt} + \sum_{o \in \mathcal{P}^O_{s_i}} Y_{op} f_{ot} - \sum_d v_{ptsd} \begin{matrix} \leq \\ = \end{matrix} 0 \quad , \quad \forall \{pts_i\} \tag{2.3.1}$$

typically include the sum of purchases, transport, consumption and production, and sells; inventories will be considered later ignoring the inventories for the moment. Consumption and production are represented by positive und negative values of yield coefficients Y_{op} coupled to the process flow variables f_{ot}. Y_{op} with o identifying a process defines how much of product p is consumed ($Y_{op} > 0$) or generated ($Y_{op} < 0$). Necessarily $\sum_p Y_{op} = 0$. For convenience, yield coefficients are normalized to $\sum_p |Y_{op}| = 2$, so that the sum of the positive and negative entries is 1.

A similar balance relation must hold for the utilities. Both crackers use heating gas, in this case a mixture of CH_4 and H_2, and electricity to heat up some feed stock products. Consumption of utility p in process o is given by H_{op} which leads us to the relations

$$e_{pts} + \sum_{o \in \mathcal{P}_s^O} H_{op} f_{ot} \stackrel{\leq}{=} 0 \quad , \quad \forall \{pts\} \quad , \tag{2.3.2}$$

between the purchase variables, e_{pts}, and the process flow variables, f_{ot}. For numerical advantages the equalities (2.3.2) are relaxed and equivalently replaced by inequalities [for further reasoning and validation see KW97 (p.281)].

For some products p, inventories with initial stock S_{ps}^A have to be considered, *i.e.*, at the end of the first period the inventory is the sum of the opening inventory and the incoming flow minus the outgoing flow:

$$i_{pts} = S_{ps}^A + F_t f_{pts}^N \quad , \quad \forall p \in \mathcal{L} \quad , \quad t = 1 \quad , \quad \forall s_i \tag{2.3.3}$$

with the number, F_t, of hours in period t and

$$f_{pts}^N := e_{pts} + t_{ps_j s_i} - t_{ps_i s_j} + \sum_{o \in \mathcal{P}_{s_i}^O} Y_{op} f_{ot} - \sum_d v_{ptsd} \quad . \tag{2.3.4}$$

In the inventory balance equations applied to the end of the periods $t \in \mathcal{T}_2 = \{2, \ldots, N^T - 1\}$

$$i_{pts} = i_{pt-1m} + F_t f_{pts}^N \quad , \quad \forall p \in \mathcal{L} \quad , \quad t \in \mathcal{T}_2 \quad , \quad \forall s_i \tag{2.3.5}$$

the initial inventory is replaced by the inventory of the previous period. In the last period the target inventory S_p^E (at present, the implementation assumes $S_{ps}^A = S_{ps}^E = 0$) is used, so we get

$$i_{pts} = i_{pt-1m} + F_t f_{pts}^N = S_{ps}^E \quad , \quad \forall p \in \mathcal{L}, \quad t = N^T, \quad \forall s_i \quad . \tag{2.3.6}$$

Modelling of Blends and Pooling: Blends (blended products), or cuts are flows, which consist of several components, but behave as one stream topologically. A cut is characterized by its components and its composition which is variable to a certain amount. Cuts are treated as products and their components as the cut's properties. Cuts can be components of other cuts. Two different types of cuts are used described by their own set of variables, although mathematically they are similar. The first type, products leaving the submodels, are in fact blends and separated into their components later, the second type are products pooled in blending nodes.

Equations for submodel nodes need to connect at first the pool collector variable $q^C_{p_b ts}$ (this is the total mass of the produced blend) to the yield coefficients and flow variables

$$q^C_{p_b ts} = \sum_{p \in \mathcal{P}^P_{p_b}} \sum_{o \in \mathcal{P}^O_s} Y_{op} f_{ot} \quad , \quad \forall p_b \in \mathcal{B} \quad , \quad \forall \{ts\} \quad . \tag{2.3.7}$$

For every component the error vectors $r_{pp_b ts}$ denotes the deviation of the total amount, $\sum_{o \in \mathcal{P}^O_s} Y_{op} f_{ot}$, of the component p in the blend from the value computed as the product of $q^C_{p_b ts}$ and the assumed value, $F^C_{pp_b ts}$, for its its relative fraction. Thus we get [compare (2.3.8) to eq. (11.1.21) in KW97, p.371]

$$t^{CC}_{pp_b ts} := F^C_{pp_b ts} q^C_{p_b ts} + r_{pp_b ts} = \sum_{o \in \mathcal{P}^O_s} Y_{op} f_{ot}, \qquad \begin{matrix} \forall p_b \in \mathcal{B} \\ \forall p \in \mathcal{P}^P_{p_b} \\ \forall \{ts\} \end{matrix} \quad . \tag{2.3.8}$$

The quantity $t^{CC}_{pp_b ts}$ describes the linear approximation of the amount of p in the blend p_b. The error vector is used in subsequent processes involving the blend. The guessed quantity $F^D_{p_b nt}$ in (2.3.9) specifies which fraction of the error vector flow is distributed to the node belonging to the processes:

$$F^C_{pp_b ts} q^D_{p_b nt} + F^D_{p_b nt} r_{pp_b ts} = \sum_{o \in \mathcal{P}^O_n} Y_{op} f_{ot}, \quad \forall \{st\}, \qquad \begin{matrix} \forall n \in \mathcal{P}^N_s \\ \forall p_b \in \mathcal{B} \\ \forall p \in \mathcal{P}^P_{p_b} \end{matrix} \quad . \tag{2.3.9}$$

Equations for the blending nodes again connect the pool collector variable $q^C_{p_b ts}$ with the flow-in variables. For blends consisting only of base products, and for blends containing other blends we have two different formulae, namely

$$\begin{aligned} \sum_{p \in \mathcal{P}^P_{p_b}} b_{pp_b ts} \quad &= \quad q^C_{p_b ts} \quad \& \quad \sum_{p \in \mathcal{P}^P_{p_b}} F^C_{pp_b ts} b_{pp_b ts} + F^D_{p_b nt} r_{pp_b ts} = q^C_{p_b ts} \\ \forall p_b \quad &\in \quad \mathcal{B} \quad , \quad \forall \{ts\} \quad . \end{aligned} \tag{2.3.10}$$

The equations for the error vectors $r_{pp_b ts}$ are:

$$\begin{aligned} b_{pp_b ts} \quad &= \quad t^{CC}_{pp_b ts} \quad \& \quad F^C_{pp_b ts} b_{pp_b ts} + F^D_{p_b nt} r_{pp_b ts} = t^{CC}_{pp_b ts} \\ \forall p_b \quad &\in \quad \mathcal{B} \quad , \quad \forall p \in \mathcal{P}^P_{p_b} \quad , \quad \forall \{ts\} \end{aligned} \tag{2.3.11}$$

In the case of a blend flowing into a submodel we again have again (2.3.9).

In *distributive recursion*, an equivalent technique to sequential linear programming, after every solution of the linear problem the guesses for concentrations and distributions are exchanged and a new iteration $(k+1)$ is started, until convergence is achieved:

$$F^{C,(k+1)}_{pp_bts} = F^{C,(k)}_{pp_bts} + r^{(k)}_{pp_bts} \Big/ q^{C,(k)}_{p_bts} \quad , \quad \begin{array}{l} \forall p_b \in \mathcal{B} \\ \forall p \in \mathcal{P}^S_{p_b} \\ \forall \{ts\} \end{array} \quad , \tag{2.3.12}$$

$$F^{D,(k+1)}_{pnt} = q^{D,(k)}_{p_bnt} \Big/ q^{C,(k)}_{p_bts} \quad , \quad \begin{array}{l} \forall p_b \in \mathcal{B} \\ \forall \{ts\} \\ \forall n \in \mathcal{P}^N_s \end{array} \quad . \tag{2.3.13}$$

Initial guesses $F^C_{pp_bts}$ for concentrations and $F^D_{p_bnt}$ for distributions are used to start the iteration.

Concerning *blends (cuts)* we need to restrict the composition of some cuts. The numbers $F^{EL}_{pp_bt}$ und $F^{EU}_{pp_bt}$ represent lower and upper bounds on the relative fractions of components. The quantities $F^{C,(k+1)}_{pp_bts}$ and $F^{D,(k+1)}_{pnt}$ must observe these bounds.

Capacity restrictions apply to all processes consuming R_{or} (units/h) of available capacity resources. Additionally lower and upper bounds D^{LR}_r and D^{UR}_r on the total capacity consumption are needed. The sum of all processes using the same resource is then limited by:

$$D^{LR}_r \le \sum_o R_{or} f_{ot} \le D^{UR}_r \quad , \quad \forall r \quad . \tag{2.3.14}$$

To model the *interpolation of cracker modes*, cracking severity variables s_{cmt} for every operating mode of the crackers are introduced $(0 \le s_{cmt} \le 1)$. They describe the percentage at which the cracker works in the corresponding mode (sharp or medium or mild). The process variables, f_{ot}, are coupled to the severity variables by

$$\sum_{o \in \mathcal{O}_{mc}} f_{ot} \le R_c s_{cmt} \quad , \quad \forall \{cmt\} \quad ; \quad \sum_{m=1}^{3} s_{cmt} = 1 \quad , \quad \forall \{ct\} \quad . \tag{2.3.15}$$

If the resource R_c is the total capacity of the cracker, and if the crackers operate at full capacity the interpolation is exact if we proceed as follows: in the case of only three modes the cracking severity variables are connected to binary variables

$$\mu_{ct} := \begin{cases} 1, & \text{in period } t \text{ cracker } c \text{ operates} \\ & \text{between mode 1 and 2} \\ 0, & \ldots \text{ between mode 2 and 3} \end{cases} \quad , \quad \forall c \in \{1,2\} \quad , \tag{2.3.16}$$

i.e., we interpolate only between two neighboring modes (sharp, medium or medium, mild):

$$s_{c1t} \leq \mu_{ct} \quad , \quad s_{c3t} \leq 1 - \mu_{ct} \quad , \quad \forall\{ct\} \quad . \tag{2.3.17}$$

This formulation enables us to compute the cracking severity ε_{ct} by

$$\varepsilon_{ct} = 0.5 s_{c1t} + 0.55 s_{c2t} + 0.6 s_{c3t} \quad , \quad \forall\{ct\} \quad . \tag{2.3.18}$$

where the reference values 0.5, 0.55 und 0.6 represent the three modes.

2.3.2.1 Bounds

The *cracking severities* ε_{ct} are bounded by S_c^{LS} and S_c^{US}. *Initial inventories*, S_{ps}^A, (t) for the first period in and target closing inventories, S_{ps}^E, (t) for the last period are assumed to be zero. *Inventory capacities* (t) are defined by S_{ps}^C and have to be observed by the inventory variables i_{pts}. *Availability Restrictions and Purchase* All materials are bought on the same market, but at possibly different prices and bounds for the sites. The purchased streams e_{pts} are subject to availability restrictions, *i.e.*, to upper and lower bounds (t/h) E_{pts}^U and E_{pts}^L. *Sales restrictions* put bounds (t/h) V_{ptsd}^U and V_{ptsd}^L on the sold streams v_{ptsd}. Lower bounds represent given contracts with demand to be satisfied. There is the possibility to fulfill these obligations by buying products from extern sources instead of producing them. This type of purchase is treated separately.

Transport restrictions consider that a certain minimal amount has to be transported, if transport is to take place at all:

$$M_p^{TL} \lambda_{ps_i s_j t} \leq t_{ps_i s_j t} \leq M_p^{TU} \lambda_{ps_i s_j t}, \quad \forall\{pts_i\}, \quad \forall s_j \neq s_i \quad . \tag{2.3.19}$$

Unfortunately it is not possible to implement (2.3.19) into `PIMS`, because transport variables can (at present) not be connected to other variables. Therefore we have to define the restrictions in the local models, for example

$$M_p^{TL} \lambda_{ps_i s_j t} \leq b_{pp_b t s_j} \leq M_p^{TU} \lambda_{ps_i s_j t}, \quad \forall\{pts_i\}, \quad \forall s_j \neq s_i \tag{2.3.20}$$

for a product absorbed to 100% by a blend (otherwise the corresponding sum has to be used in the left-hand side), or

$$M_p^{TL} \lambda_{ps_i s_j t} \leq f_{ot} \leq M_p^{TU} \lambda_{ps_i s_j t}, \qquad \forall\{pts_i\} \quad , \quad \forall s_j \neq s_i \tag{2.3.21}$$

for a product absorbed to 100% by a certain process.

2.3.2.2 Objective Function

The objective function includes terms for revenue, product consumption, external purchase, transport and inventory. Revenues and purchases have to be multiplied with the factor, F_t, because the corresponding data is defined on an hourly basis. The costs for holding inventory is defined on a monthly basis. F_t^{PV} describes the effects of inflation and interest rates in the model. As a side-effect, some symmetries in the model are broken, which leads to better convergence. This effect is added to by the usage of inventory holding costs. The objective function is formulated as

$$z = \sum_{t \in \mathcal{T}} F_t^{PV} \left[F_t \left(y_t^E - y_t^R - y_t^P - y_t^T \right) - y_t^L \right] \tag{2.3.22}$$

total revenue z_t^E based on specific revenues E_{ptsd}^P (DM/t),

$$z_t^E := \sum_{p \in \mathcal{P}_\mathrm{F}} \sum_{s \in \mathcal{S}} \sum_{d \in \mathcal{D}} E_{ptsd}^E v_{ptsd} \quad , \tag{2.3.23}$$

total cost costs for raw materials, z_t^R, and products purchased externally

$$z_t^R := \sum_{p \in \mathcal{P}_\mathrm{R}} \sum_{s \in \mathcal{S}} C_{pts}^R e_{pts} \quad , \quad z_t^P := \sum_{p \in \mathcal{P}_\mathrm{T}} \sum_{s \in \mathcal{S}} C_{pts}^P e_{pts} \quad , \tag{2.3.24}$$

based on specific costs E_{ptsd}^R and C_{pts}^E (DM/t) for materials consumed and purchased externally, and transport and inventory costs

$$z_t^T := \sum_{p \in \mathcal{P}_\mathrm{P}} \sum_{s_i \in \mathcal{S}} \sum_{\substack{s_j \in \mathcal{S} \\ s_j > s_i}} C_{pzs_is_j}^T t_{pzs_is_j} \quad , \quad z_t^L := \sum_{p \in \mathcal{L}} \sum_{s \in \mathcal{S}} C_{pts}^L i_{pts} \quad . \tag{2.3.25}$$

The specific transport costs are given for every material specifically as $C_{pzs_is_j}^T$ (DM/t). The specific costs for inventories C_{pts}^L (DM/t) are calculated from the cost of the working capital tied up in the inventory. Prices for sold and purchased products are given on a monthly base (DM/t). They can be either "official" market prices or intern "computational" prices.

2.3.3 Solution Approach and Results

The model falls into the class of MINLP problems. The mathematical algorithm implemented in `PIMS` is similar to a first step of outer approximation [see Section 2.4.2.1]. At first, the NLP relaxation is solved

by *distributive recursion* which is equivalent to sequential linear programming but has better scaling [see, for instance, KW97 (pp.368)]. The next step is to fix the recursed terms (concentrations) and to solve the MILP problem. Finally, with the discrete variables fixed, an NLP problem is solved updating the recursed terms.

The matrix of the resulting linear programs (only Ludwigshafen) has about 1300 rows, 1600 columns and 15500 non-zeros. 30 of the variables are binary, and we have 269 nonlinear constraints. Antwerp adds another 800 rows and 1000 columns. Solution times are between 2 and 15 minutes. Once a solution has been found the recursed terms are stored and used as initial values in subsequent runs.

Typical questions analyzed are for example, "which product should be produced at which site?", "which raw materials should be purchased?", "under which circumstances is it advantageous to transfer products between the sites?", or "which effects do certain changes have on the global system?". Several case studies have been performed for the individual sites and for the 2-site-network. The savings are generally in the order of magnitude of 1%, which corresponds to several millions DM per year.

2.4 An Integrated Site Analysis

The purpose of the model is to design an integrated production network minimizing the costs for raw material (RM), investment and variable costs for re-processing units, and a cost penalty term for remaining impurities.

Three types of production processes (units) are considered: source processes producing RM not requesting it, sink processes only requiring RM not producing it and stream processes requesting and producing the RM. The flow rate of a process and its effect on the quality of RMs are known *a priori*.

Until now purchased RM of different qualities measured in terms of certain impurities has been used for all production processes requiring RM. It seems recommended to re-use impure RM for other processes and so to reduce the costs for purchasing RM. This may require new connections between units. Unfortunately, for most processes inlet specifications for the RM restrict the direct re-use of impure RM of other processes. In addition, some flow connections between processes are not allowed or not possible ("forbidden matches"). However, the inlet specifications may be satisfied by pooling RM streams of different quality. This has the additional advantage that only one pipeline is necessary instead of several so that investment costs for pipelines (depending on required capacity and the distance between the production processes to be connected) and costs for pumping RM into

pipelines are reduced. For each single process we know the amount and type of impurities it produces, *i.e.*, has to be added to the impurities already in the stream. To pool the RM streams no *investment or variable costs* have to be taken into account, except for the case of new pipelines to be built because pools are realized by joining different RM streams without any technical or financial expense.

An alternative approach is to process the quality of the RM when leaving a process. This requires local *re-processing units* (RPUs) for a (partial) improving of the RM quality causing investment costs for RPUs, as well as costs for operating the RPUs. The components extracted by the RPUs might be re-used or sold. Since for certain impurities we consider penalty costs when they leave the system money can be saved by reducing the amount of these impurities.

A central RPU already exists which improves the quality of RM before leaving the site; the RM is used by other sites but the more impurities remain leaving the site the lower the quality, and thus, the less valuable it is. Since we cannot easily convert quality into money we consider penalty costs for the impurities remaining in the RM. Therefore, it might be more promising of building small local RPUs.

The investment costs for new local RPUs depend on the required capacities (RM flow rates) but additionally on the type of impurity and on its concentration. The variable costs for the RPUs depend on the mass load (in kg/h) of the input impurities. For the use of the central RPU only variable costs have to be payed depending on the type and the total mass of the impurity.

The RPUs can be regarded as a certain type of stream process, where the total input and output of RM and impurities are the same. Unlike production stream processes RPUs split both RM stream and mass load of impurities, *i.e.*, two RM streams leave the RPU: a main RM stream (relatively high quality) and a small stream (very low quality). The ratio of the amount of RM of the main outlet stream to that of the input stream is prescribed for each RPU. Analogously the ratio ("extraction rate") of the mass load of an impurity in the low quality stream to those of the inlet mass load can be estimated as a constant or prescribed as a function of the inlet concentration for each RPU and each impurity. Similar to production processes, for technical reasons, there might exist inlet specifications limiting the concentration of a certain impurity. Since the raw material leaving the production network might be used for other purposes we also consider bounds for the concentration of impurities in the RM leaving the system. They can be understood as "outlet specification" (*i.e.*, maximum concentration of an impurity in the RM) of the central RPU.

2.4.1 The Mathematical Model

At first let us summarize the dimensions of the model and the indices:

	#	description		
	3	raw material qualities	s	sinks
N	~ 60	source processes	k	impurities
M	~ 7	impurities	m	re-processing units
L	≥ 1	RPUs	p	pools
P	~ 60	pools	s	connection capacity

(2.4.1)

We consider a total of $3+N+L+P \sim 125$ processes, and introduce, for convenience, the following sets of indices

$\mathcal{K}^P$	set of all pairs (i,j) of possible matches
$\mathcal{K}^N$	set of all pairs (i,j) of unexisting matches ($\mathcal{K}^N \subset \mathcal{K}^P$)
$\mathcal{P}^{SO}$	$i \in \mathcal{P}^{SO} \Leftrightarrow P_i$ is a source process
$\mathcal{P}^{SI}$	$j \in \mathcal{P}^{SI} \Leftrightarrow P_j$ is a sink process
$\mathcal{P}^{ST}$	$i \in \mathcal{P}^{ST} \Leftrightarrow P_i$ is a stream process
$\mathcal{P}^{T}$	$m \in \mathcal{P}^{T} \Leftrightarrow P_m$ is a re-processing unit
$\mathcal{P}^{P}$	$m \in \mathcal{P}^{P} \Leftrightarrow P_p$ is a pool
$\mathcal{P}_1$	$\mathcal{P}_1 := \mathcal{P}^{SO} \cup \mathcal{P}^{ST} \cup \mathcal{P}^{P} \cup \mathcal{P}^{T}$
$\mathcal{P}_2$	$\mathcal{P}_1 := \mathcal{P}^{SI} \cup \mathcal{P}^{ST} \cup \mathcal{P}^{P} \cup \mathcal{P}^{T}$

(2.4.2)

2.4.1.1 Variables

real variable	dim.	description
$x^A_{ij} \equiv x_{ij}$	t/h	output from process P_i going to process P_j (at waste RM RPUs: output (A), high quality)
$x^B_m = x^A_{im}\,(1-G_m)$	t/h	low quality output (B) from process P_i; for other processes than local RPUs $x^B_i = 0$
c^{in}_{ik} (c^{out}_{ik}):	ppm	input (output) concentration of impurity k in process P_i
$z^{in}_{jk} := \sum_{(i,j)\in K^P} c^{out}_{ik} x_{ij}$	t/h	input mass load of impurity k in process P_j summed over all processes P_i which send RM to it

In addition we use the binary variables μ_{ij}, ε_{sij} and ν_m indicating whether a connection exists from P_i to P_j, whether a pipeline from P_i to P_j is of capacity and whether process (RPU) P_m exists.

data	dim.	description
K_{ik}^{out}	$[ppm]$	specific outlet concentration of impurity k of process P_i
K_{jk}^{in}	$[ppm]$	inlet specification for impurity k of process P_j
X_i	$[t/h]$	inlet flux of process P_i, $X_i = 0$ for source processes P_i
Y_i	$[t/h]$	outlet flux of process P_i, $Y_i = X_i$ for all stream processes and $Y_i = 0$ for sink processes; for local RPUs it is $Y_i \leq X_i$ because the RM stream is splitted into a main stream (A) and a smaller low quality stream (B)
$F_{mk}(c_{mk}^{in})$	$[-]$	rate of extraction for a single impurity k removed in RPU P_m. This rate is a function of the input concentration c_{mk}^{in} and is of order 0.7....0.95.
G_m	$[-]$	splitting rate of the RM stream within RPU m.
C_{ij}^{PI}		investment costs for building a pipeline from P_i to P_j
$C_m^{TI}(x_{im}, z_{mk}^{in})$		investment costs for a local RPU P_m. These costs are a function of the RM flow rate once it has been decided which impurities have to be removed.
$C_{mk}^{TV}(z_{mk}^{in})$		variable costs for RPU P_m. They are a function of the product of extraction rate and mass load of impurity k, *i.e.*, $f_{mk}(c_{mk}^{in})z_{mk}$
C_i^{RM}		costs for purchased RM of quality i
C_k^{PEN}		penalty costs for impurities leaving the system
V_s^{PI}	$[t/h]$	capacity of pipeline is of type (size) s

2.4.1.2 Constraints

The production processes (sink and stream processes) require a constant amount of RM, and the production (source or stream) processes have a constant emission of waste RM. Therefore the sum of all inlet

and outlet fluxes is constant, and we have the *mass balances*

$$\sum_{i|(i,j)\in K^P} x_{ij} = X_j \quad , \quad \forall j \in \left\{\mathcal{P}^{SI} \cup \mathcal{P}^{ST}\right\} \quad , \tag{2.4.3}$$

and

$$\sum_{j|(i,j)\in K^P} x_{ij} = Y_i \quad , \quad \forall i \in \left\{\mathcal{P}^{SO} \cup \mathcal{P}^{ST}\right\} \quad . \tag{2.4.4}$$

Additionally, for stream and pool processes it is assumed that there is no loss of RM, *i.e.*,

$$X_i = Y_i \quad , \quad \forall i \in \left\{\mathcal{P}^{P} \cup \mathcal{P}^{ST}\right\} \quad . \tag{2.4.5}$$

In RPUs the RM stream is separated into a main stream (A) of relatively high quality and a stream (B) of low quality. Since no RM loss is assumed the RM flow of the RPU P_m can be modeled similar to those of stream processes:

$$\sum_{i|(i,m)\in K^P} x_{im} = \sum_{j'|(m,j')\in K^P} x_{m,j'} + x^B_m, \qquad \begin{array}{l} \forall i \in \left\{\mathcal{P}^{SO} \cup \mathcal{P}^{ST} \cup \mathcal{P}^{P}\right\} \\ \forall j \in \left\{\mathcal{P}^{SI} \cup \mathcal{P}^{ST} \cup \mathcal{P}^{P}\right\} \\ \forall m \in \mathcal{P}^T \end{array} \tag{2.4.6}$$

The splitting of the main RM stream into two streams within a RPU plant is prescribed by the factor G_m, the ratio of RM flow rate in stream (B) to the total input flow rate ($G_m = 1$ implies that all RM is kept in the main stream):

$$G_m = \frac{x^B_m}{\sum_{i|(i,m)\in K^P} x_{im}} \quad \Leftrightarrow \quad x^B_m = \sum_{i|(i,m)\in K^P} G_m x_{im} \quad , \quad \forall m \in \mathcal{P}^T \tag{2.4.7}$$

The pool P_j collecting different RM streams originating from processes P_i is described by

$$c^{in}_{jk} X_j = \sum_{i|(i,j)\in K^P} c^{out}_{ik} x_{ij} \quad , \qquad \begin{array}{l} \forall i \in \mathcal{P}_1 \\ \forall k, \forall j \in \mathcal{P}^P \end{array} \quad . \tag{2.4.8}$$

The concentration limits for certain impurities originating from the processes P_i and entering P_j yield the material balance with respect to the content of impurities:

$$\sum_{i|(i,j)\in K^P} c^{out}_{ik} x_{ij} =: z^{in}_{jk} \leq K^{in}_{jk} X_j \quad , \quad \forall j \in \mathcal{P}_2 \quad , \quad \forall k \quad . \tag{2.4.9}$$

Knowing the concentration of impurity k and the RM fluxes originating from process P_i and entering process P_j the pooled concentration

c_{jk}^{in} , *i.e.*, the inlet concentration of P_j, can be calculated by

$$c_{jk}^{in} \sum_{i|(i,j)\in K^P} x_{ij} = \sum_{i|(i,j)\in K^P} c_{ik}^{out} x_{ij} \tag{2.4.10}$$

$$\Leftrightarrow \quad c_{jk}^{in} X_j = z_{jk}^{in} \quad , \quad \forall j \in \mathcal{P}_2, \quad \forall k \tag{2.4.11}$$

The outlet mass load of impurities of a process consists of the impurities which already have been in the inlet RM stream and the impurities added by the production process. Their total concentration in the outlet stream can be calculated by

$$c_{jk}^{out} = c_{jk}^{in} + K_{ik}^{out} \quad , \quad \forall j \in \left\{\mathcal{P}^{SO} \cup \mathcal{P}^{ST}\right\}, \forall k \tag{2.4.12}$$

and the output mass load of stream processes (not valid for RPUs!) is

$$z_{ik}^{out} X_i = z_{ik}^{in} + K_{ik}^{out} X_i \quad . \tag{2.4.13}$$

The outlet concentration of a impurity k of a RPU P_j depends on the inlet concentration and the extraction rate for removing the substance. As already mentioned above the impurities can be divided in two groups with additional subgroups for which the removal in RPUs is very different. In general it can be described by the function $z_{jk}^{out} = H(z_{jk}^{in})$: The left and right formulae show typical relations representing the operation of the RPUs:

$$\begin{array}{lcl} z_{jk}^{out} = z_{jk}^{in}\left(1 - F_{jk}\left(z_{jk}^{in}\right)\right) & & \\ \text{or} & & \\ z_{j1}^{out} = z_{j1}^{in} - \sum_{k=2}^{6} z_{jk}^{in} F_{jk}(z_{jk}^{in}) & & \end{array} \qquad \begin{array}{lcl} z_{jk_1}^{out} = z_{jk_*}^{in} - z_{j,k_2}^{in} & & \\ z_{j,k_2}^{out} = 0 & & \\ z_{j,k_2}^{out} = z_{j,k_3}^{in} & & \end{array} \quad . \tag{2.4.14}$$

The *construction of pipelines* from i to j can be required by the inequalities

$$x_{ij} \leq \min_{ij} \{Y_i, X_j\} \mu_{ij} \quad , \quad x_{ij} \leq \sum_s V_{sij}^{PI} \varepsilon_{sij} \quad , \quad \forall (i,j) \in \mathcal{K}^N \quad , \tag{2.4.15}$$

where the second inequality describes the required type (capacity) of pipeline.

To be sure that only one pipeline is built between process P_i and P_j the sum over s over the binary variable ε_{sij} which describes the type (capacity) of pipeline is forced to be μ_{ij} which is 1 if the connection exists and which is 0 else:

$$\mu_{ij} = \sum_{s=1}^{S} \varepsilon_{sij} \quad , \quad \forall i \in \mathcal{P}_1 \quad , \quad \forall j \in \mathcal{P}_2 \quad . \tag{2.4.16}$$

In order to force the construction of a RPU P_m the inequalities

$$\sum_{i|(i,m)\in K^P} x_{im} \leq \left(\sum_{j'|(m,j')\in K^P} x_{m,j'} + x_m^B \right) \nu_m, \qquad \begin{array}{l} \forall i \in \mathcal{P}^{SO} \setminus \mathcal{P}^T \\ \forall m \in \mathcal{P}^T \\ \forall t \end{array} \tag{2.4.17}$$

have to be fulfilled.

2.4.1.3 Objective Function

The objective function sums over all investment and variable costs, over penalty costs to be payed and over the income which can be achieved by re-using impurities and it considers all possible matches between the processes. This sum is to be minimized:

$$\begin{aligned} Z \quad : \quad & = \min \sum_{\substack{j \\ (i,j)\in K^P}} \sum_{i} C_{ij}^{PI} \mu_{ij} \\ & + \sum_{\substack{j \\ (i,j)\in K^P}} \sum_{i} \left\{ C_j^{TI}(x_{ij}, z_{jk}^{in})\nu_j + \sum_k C_j^{TV} z_{jk}^{in} \right\} \\ & + C_{ij}^{RM} x_{ij} - \sum_{\substack{j \\ (i,j)\in K^P}} \sum_k S_k \left(z_{jk}^{in} - z_{jk}^{out} \right) \\ & + + \sum_k C_k^{PEN} z_{ctr,k}^{out} \quad . \end{aligned} \tag{2.4.18}$$

The first term represents the investment costs for pipelines from process i to j, the third the variable costs and investment costs for RPU j collecting streams from process i, and the fourth term the costs for RM streaming from process i to j, $i \in \{1, 2, 3\}$. The second-last term is the revenue from sold impurity k in streams originating from processes i and extracted in process j, the last term in (2.4.18) represents the costs for impurity k leaving the system after having passed the central RPU P_{ctr} .

2.4.2 Solution Approach and Results

2.4.2.1 Mathematical Solution - Outer Approximation

To solve this MINLP problem we use the *outer approximation* (OA) algorithm by Duran and Grossmann (1986). This algorithm generates a sequence of NLP sub-problems (produced by fixing the binary variables $\mathbf{y}^k$) and MILP Master problems. Algorithms based on OA describe the feasible region as the intersection of an infinite

collection of sets with a simpler structure, *e.g.*, polyhedra. In OA the Master problems are generated by "outer approximations" (linearizations, or Taylor series expansions) of the nonlinear constraints at *those* points which are the optimal solutions of the NLP subproblems. The key idea of the algorithm by Duran and Grossmann (1986) is to solve the MINLP with a much smaller set of points, *i.e.*, tangential planes. In convex MINLP problems, a superset of the feasible region is established. Thus, the OA Master problems (MILP problem in both discrete and continuous variables) produce a sequence of lower bounds monotonically increasing. The NLP sub-problems yield upper bounds for the original problem while the MILP Master problems yield additional combination of binary variables $\mathbf{y}^k$ for subsequent NLP sub-problems. Under convexity assumptions the Master problems generate a sequence of lower bounds increasing monotonically. The algorithm terminatestermination criterionintermination criterion if lower and upper bounds equal or cross each other. The OA algorithm has heuristic extensions for non-convex MINLP.

2.4.2.2 Software

To model and solve the MINLP the software package `GAMS` by GAMS Inc. (Washington) [see, *e.g.*, Broocke *et al.*, 1992)] with the `DICOPT`-algorithm (Viswanathan & Grossman, Carnegie Mellon University) using a nonlinear solver (`CONOPT` by ARKI Consulting & Development A/S, Denmark) in combination with a MILP-solver are used. `DICOPT`DICOPTDICOPT (Viswanathan and Grossmann, 1990) seems to be the only commercial software available for solving the MINLP problem (2.1.1) of realistic size. It uses OA with some extensions for non-convex problems. To initialize the algorithm the first linearization is derived from the solution of the continuous relaxation of the MINLP, *i.e.*, it is not required that the user provides any discrete initial point. The termination criterion is different from a pure "crossing bounds" method. In a non-convex model the algorithm terminates when the solutions of the NLP problems do not provide improved upper bounds.

2.4.2.3 Homotopy Method

The problem is solved in sequence of sub-models formulated in `GAMS`, exploiting the results of the previous one, *i.e.*, we use a homotopy method. A simple linear model provides initial values for two simple nonlinear submodels. Solution times are of the order of one or two hours.

2.4.2.4 Results

The model was well appreciated by the client for its high degree of reality, exact mass balances of raw material impurities, and the free pools. The model reproduced and confirmed earlier suggestions by engineer establishing a certain amount of trust, and finally, suggested further non-intuitive improvements with remarkable financial savings.

2.5 A Production Planning and Process Design Problem

This optimization problem is concerned with a production process of a certain product P involving a system of connected reactors. The problem is a typical process design problem leading to a mixed integer nonlinear model. Nonlinear terms are related to the exponential terms for the reaction kinetics and rational terms to describes the mass flow. The description of reaction kinetics is in parts based on nonlinear expression (interpolated and approximated functions describing density and viscosity). Discrete features are needed to count the number of reactors, the existence of connections, the length of reactor chains, and to select the size of reactors. The variables are the flow rates, fractions, and the number and size of reactors.

2.5.1 Mathematical Formulation of the Model

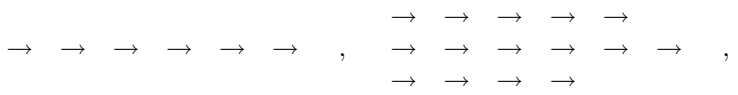
Throughout this model description the following set $r \in \mathcal{R} \cup \mathcal{T} := \{1, \ldots, N^R\} \cup \{p\text{-tank}\}$ of indices is used. Most of the variables are non-negative (continuous) flow variables

$$m_{pr} = M_p n_{pr} \quad , \quad p \in \mathcal{P} := \mathcal{L} \cup \mathcal{G} := \{A, B, C, P\} \cup \{G_1, G_2, G_3\} \tag{2.5.1}$$

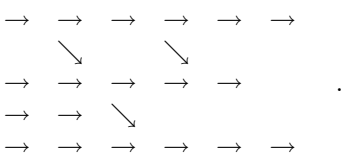
describing the total mass flow of product p or gas g into or out from node r, a reactor or the product tank. We distinguish between the liquids $\mathcal{L}$ and the gases $\mathcal{G}$ because they are subject to different topologies. While the variables m have the dimension tons/hour the variables n are in kmol/hour; they are coupled by the molecular masses M_p. Other variables are the temperature T_r and pressure p_r^P in reactor r, the stirring energy e_r used in reactor r, and as auxiliary variables, the weight fractions w_r^p.

The valuable product, P, is produced by a system of reactors r. The reactors are connected according to free or fixed pattern (single chain, parallel chains of different lengths, parallel chains with connec-

tions) as shown below



or parallel chains with connections



During the synthesis within a reactor r side-products are produced like A, B or G_1, and raw material remains. Liquid components are fed to one or several subsequent reactors or via filter to the product tank. For the gaseous components incidence tables control the flow between reactors. The gaseous outlet of some reactors can leaves to the incinerator or can be fed back to a reactor.

2.5.1.1 Mass Balances for the Reactors

The input side of reactors is described by

$$n^i_{pr} = I^S_{pr} n^S_{pr} + \sum_{s \in \mathcal{R} | I^I_{psr}=1} x_{psr} n^o_{ps} \quad , \quad \forall r \in \mathcal{R} \cup \mathcal{T}, \quad \forall p \in \mathcal{P} \quad (2.5.2)$$

For a fixed topology of reactors the (binary) incidence table I^S_{pr} describes whether reactor r is connected to a supply tank of product p; the variable n^S_{pr} describes the flow of product p from the supply tank to reactor r. If the topology is free we just set $I^S_{pr} = 1$ for all combinations. The second term describes the flow of product p from all possible reactors $r_s \in \mathcal{R}$ to reactor r. The fractions, $0 \leq x_{psr} \leq 1$, distribute the output flow from a certain reactor to other subsequent reactor. Conservation of total flow is enforced by

$$\sum_{d \in \mathcal{R} | I^I_{prd}=1} x_{prd} = 1 \quad , \quad \forall r \in \mathcal{R} \cup \mathcal{T} \quad , \quad \forall p \in \mathcal{P} \quad . \qquad (2.5.3)$$

For a fixed topology with either single or several unconnected parallel chains we have $x_{prr+1} = 1$, and $x_{prd} = 0$ for all other combinations. For all products p within a stream from s to r we enforce the pooling condition

$$x_{psr} = x_{p'sr} \quad , \qquad (2.5.4)$$

which expresses the conservation of composition. The synthesis of P needs a catalyst to be fed to the reactors. Since none of the catalyst flowing through the reactors is consumed we can approximately describe the flow of the catalyst by only one continuous variable, m^{CAT}. The mass of the catalyst is separated from the product outflow and re-used again in the reactors.

How many kmol/h of substance p will leave the reactor r is described by

$$n^o_{pr} = n^i_{pr} + \sum_{p' \in \mathcal{P} | p' \neq p \wedge S_{pp'} \neq 0} S_{pp'} \Delta n_{p'r} \quad , \quad \forall p \in \mathcal{P} \quad , \quad \forall r \quad . \tag{2.5.5}$$

with

$$\Delta n_{p'r} := n^o_{p'r} - n^i_{p'r} \quad , \quad \forall p \in \mathcal{P} \quad , \quad \forall r \quad . \tag{2.5.6}$$

From the throughputs, $\Delta n_{p'r}$ of products p' in reactor r, with known reaction scheme represented by the stoichiometric coefficients $S_{pp'}$, we can derive the production and loss terms of product p. The amount of product p' leaving reactor r is the amount which has gone into the reactor and the amount which is produced in the reactor. The amount $\Delta n_{p'r}$ produced depends on the reaction rates $r_{p'r}$ and the volume of the reactor V_r:

$$\Delta n_{p'r} = n^o_{p'r} - n^i_{p'r} = r_{p'r} V_r \quad , \quad \forall p' \in \mathcal{P} \quad , \quad \forall r \quad . \tag{2.5.7}$$

In the current case, C is consumed by the production of P and its by-product B, *i.e.*, S_{CP} and S_{CB} are non-zero. Product A flows from a certain reactor to all possible subsequent reactors, but it is also produced as a by-product of the P-synthesis, namely during producing B and G_1, *i.e.*, S_{AG_1} and S_{AB} are different from zero.

G_1 is produced in a undesired reaction of G_2 and G_3. So the amount leaving the reactor consists of the amount flowing in and the amount produced depending on reaction rates $r_{G_1 r}$ and reactor volume V. The gaseous basic chemicals G_2 and G_3 are consumed within the reaction. The consumption depends on the amount of P, B and G_1 produced, *i.e.*, we need to consider $S_{G_3 P}$, $S_{G_3 B}$ and $S_{G_3 G_1}$.

2.5.1.2 Reaction Rates and Weight Fractions

The amount of P, B and G_1 produced in reactor r depends on the reaction rates. These, in turn, are functions of temperature, density, concentration of catalyst and other parameters describing the chemical synthesis.

In the following the reaction rates are formulated for the substances which are synthesized or which decay. While for elementary

reactions the rates are of the form

$$r_{AB} = k_0 \, [A] \, [B] \, e^{-\frac{E}{kt}} \quad , \tag{2.5.8}$$

where $[A]$ and $[B]$ denote the concentrations of compounds A and B, the reaction rates in the current scheme are more complicated and depend especially on interpolated functions for density, viscosity etc. The reaction rates r_{Pr} and r_{Br} for the P and B synthesis are computed by

$$r_{Pr} = r_{Pr}^{(1)} - r_{G_1 r}^{(2)} \quad , \quad r_r^B = C_1 m^{CAT} f_5(c_r^l) \quad , \tag{2.5.9}$$

where c_r^l is an auxiliary variable involved in the interpolation of the reaction kinetics, $r_{Pr}^{(1)}$ is computed in formula (2.5.14), and C_1 is a constant. The rate $r_{G_2 r}^{(2)}$ describing the consumption of G_2 can be calculated as the sum of the rates

$$r_{G_2 r} = r_{Pr} + r_{Br} + r_{G_1 r} \quad , \tag{2.5.10}$$

and $f_n(x)$ using the catalyst data $P_1 = 25700$ and $P_2 = 400$ is defined as

$$f_n(x) := h_n(x) e^{-P_1 h_1(x)} \quad , \quad h_n(x) := \frac{x}{(1 + P_2 x)^n} \quad . \tag{2.5.11}$$

Since c_r^l is of the order of $5 \cdot 10^{-5}$, by substituting $x = s(y)$ and exploiting the partial fraction relation

$$x = 5 \cdot 10^{-5}(1 + y) \quad , \quad \frac{1+y}{\alpha + \beta y} = \frac{1}{\alpha} + \frac{\alpha - \beta}{\alpha} \frac{y}{\alpha + \beta y} \quad , \tag{2.5.12}$$

(2.5.11) can be replaced by the numerical more stable expression

$$f_n(x) = g_n(y) := \frac{5 \cdot 10^{-5} \, (1+y)}{(1.02 + 0.02y)^n} e^{-\frac{1.825}{1.02}} e^{-\frac{1.825}{1.02} \frac{y}{1.02 + 0.02y}} \quad . \tag{2.5.13}$$

Finally we calculate the rate, $r_{G_1 r} = r_{G_1 r}^{(1)} + r_{G_1 r}^{(2)}$, of the synthesis of G_1

$$\begin{pmatrix} r_{G_1 r}^{(1)} \\ r_{G_1 r}^{(2)} \\ r_{Pr}^{(1)} \end{pmatrix} = m^{CAT} \begin{pmatrix} C_2 c_r^l f_2(c_r^l) \\ C_3 c_r^l \, c_r^P e^{-n_{Cr}^i} \\ C_4 f_3(c_r^l) \end{pmatrix} \quad , \quad c_{Pr} = \frac{n_{Pr}^o}{V_r} \quad . \tag{2.5.14}$$

The concentration, c_{Pr}, of P in reactor r is directly available, c_r^l can only be calculated implicitly using the nonlinear Arrhenius equation

$$g(c_r^l) := c_r^a h_{-2}(c_r^l) + 2000 m^{CAT} f_0(c_r^l) - c_{G_2 r} c_r^a \left(1 + P_2 c_r^l\right)^2 = 0 \quad , \tag{2.5.15}$$

where c_r^a is the gas-fluid exchange coefficient in reactor r depending on the energy e_r,

$$c_r^a = C_5 e_r^{0.7} (1000\rho_r)^{0.27} (0.001\eta_r)^{-\frac{5}{6}} \quad . \tag{2.5.16}$$

The concentration, $c_{G_2 r}$, of G_2 at the phase boundary

$$c_{G_2 r} = \frac{\alpha_r}{22.4} p_{G_2 r} \quad , \tag{2.5.17}$$

depends on the gas solubility, α_r. Both α_r and the viscosity η_r in reactor r are polynomials of third order in two variables

$$\begin{array}{l} \alpha_r = C_6 + \left(\mathbf{v}^{\mathrm{T}} \mathsf{H}_1 \mathbf{v}\right) \\ \eta_r = C_7 + \left(\mathbf{v}^{\mathrm{T}} \mathsf{H}_2 \mathbf{v}\right) \end{array} \quad , \quad \mathbf{v}^{\mathrm{T}} = (w_{Pr}, w_{Cr}) \quad , \tag{2.5.18}$$

where H_1 and H_2, as well as H_3 and H_4 used below, are constant matrices of appropriate dimensions. The mean density ρ_r in reactor r is a polynomial of fourth order, *i.e.*,

$$\rho_r = C_8 + \left(\mathbf{u}^{\mathrm{T}} \mathsf{H}_3 \mathbf{u}\right) \left(\mathbf{u}^{\mathrm{T}} \mathsf{H}_4 \mathbf{u}\right) \quad , \quad \mathbf{u}^{\mathrm{T}} = (w_{Pr}, w_{Br}, w_{Cr}, T_r) \quad . \tag{2.5.19}$$

Finally, the weight fractions depend nonlinearly on the molecular masses according to

$$w_{pr} = m_{pr}^o \Big/ \sum_{p' \in \mathcal{P}_p} m_{p'r}^o \quad , \quad \begin{array}{c} \forall r \in \mathcal{R} \\ \forall p \in \mathcal{P} \\ \mathcal{P}_p := \{p' \in \mathcal{P} \mid p' \neq p\} \end{array} \quad . \tag{2.5.20}$$

2.5.1.3 Discrete Features of the Model

The dominant discrete feature is the requirement that the flow rates between reactors do not become arbitrarily small. This is guaranteed by binary variables δ_{sr} indicating that reactor s has a connection to reactor r and the constraints

$$C_{sr}^{\min} \delta_{sr} \leq x_{psr} \leq C_{sr}^{\max} \delta_{sr} \quad , \quad \forall p \in \mathcal{P}_{sr} \quad . \tag{2.5.21}$$

Three additional inequalities ensure that each reactor has at least one in-flowing and one out-flowing stream, and that the number of subsequent reactors fed by reactor r does not exceed a maximum number N^{SR}, *i.e.*,

$$\sum_{r' \in \mathcal{R}} \delta_{r'r} \geq 1 \quad , \quad 1 \leq \sum_{r' \in \mathcal{R}} \delta_{rr'} \leq N^{SR} \quad , \quad \forall r \in \mathcal{R} \quad . \tag{2.5.22}$$

A similar constraint is used to enforce that at least one reactor is connected to the filter. The logical constraint

$$\delta_{Arr'} = \delta_{Crr'} \tag{2.5.23}$$

is used to guarantee that the liquids A and C are not used solely.

Finally, we need to select the size of the reactors. The mathematics is almost identical to the selection of the size of pipelines in Section 2.4.1.2, and is not repeated here.

2.5.1.4 The Objective Function

We consider four alternative objective functions: The first one is to maximize the selectivity of P defined as reaction rate r_P of P over reaction rate, r_{G_2}, of G_2, the most expensive raw material. First experiments with this objective function showed that solutions are produced with high selectivity but only for low amount of P produced. Thus in this objective function scenario we require that a certain amount of P has to be produced. The second objective function maximizes the total mass of P. The third objective function minimizes the variable cost to produce P while guaranteeing that a certain minimal amount of P is produced. A fourth objective function minimizes the total consumption of stirring energy.

2.5.2 Solution Approach

The convergence of the problem depends critically on appropriate initial values required to solve the NLP problems. So, in the beginning we often experienced divergence. The numerics improved when variables and constraints were re-scaled. It became also necessary to apply bounds on some of the process variables (*e.g.*, temperature, pressure, etc.) to keep the values in physical realistic ranges. A special example of scaling is related to the quantity c_r^l used as an argument of (2.5.11) in many places. Finally, `GAMS` supported the computation of useful initial values by minimizing the violation of certain equations. Among the most difficult one is (2.5.15). In this case the minimization of violation variables $\mathbf{v}^+$ and $\mathbf{v}^-$ in the relaxation $\mathbf{g}(\mathbf{c}) = \mathbf{v}^+ - \mathbf{v}^-$ of $\mathbf{g}(\mathbf{c}) = 0$ provides good initial value in short time. For fixed topology, with appropriate initial values the NLP problems is solved in a few minutes. This approach is more typically for a production planning system. Current initial values are stored and re-used in next production planning runs in daily life. The real design problem and the full MINLP approach is only used by experts to investigate new situations.

2.5.3 Case Study Results

The following description of optimization results are based on the analyses of a typical reference scenario provided by the client. If not mentioned otherwise the feed back of products from the sixth reactor

to the first is set to zero. For fixed topology (a chain of six reactors) we get within less than one minute of CPU time:

- The total amount of stirring energy has been kept constant but the optimiser was free in distributing it to the six reactors. The input mass flow of the chemicals were not kept constant compared to the reference solution. The optimisation yielded a slightly larger consumption of G_3, G_2 and C and an increasing (from the very first reactor) need of stirring energy. The mass fraction of P and the produced mass of P were increased by about 1%.

- Here we wanted to minimize the total stirring energy while a minimum production of P was required in order to compare the results with the reference solution. The results show a slightly higher consumption of G_2, for G_3 vice versa, while the total energy need was about 8% lower than for the standard scenario.

- Distribution of the stirring energy with a constant amount of total energy but in contradiction to scenario E1 the input of chemicals was set to be the same as in the reference scenario. The results show a distribution of the stirring energy increasing from the first reactor whereas all other output values did not improve.

- Here we wanted to investigate the influence of the feedback of the products from the last to the first reactor. The results show that the amount of produced P increases if one requires a certain feedback (*e.g.*, 10% or 20%). Unfortunately, this gain gets lost because of the feedback so that the effective amount of P is reduced compared with the reference solution. The optimal solution is found for the case of no feedback.

The optimization model has been embedded into an attractive and easy to use user-interfaces. It helps the client in his daily production planning duties to adjust his plant immediately to current needs, *i.e.*, changes in costs, capacities fluctuations or to attributes of orders. The tool supports the design phase and helps to lay out cascades and connections of a system of reactors. Here, the client sees the benefit being able to compare variants proven optimal or of at least known quality. The new designs safe raw material, minimize waste material and increase the capacity of the reactor system. In the lay-out phase the tool support design and other changed constraints.

2.6 Conclusions

In this article mixed integer nonlinear optimization has been considered as an approach to solve complex production planning and design problems. The problems discussed are very demanding in terms of the mathematical modeling, and appropriate tuning of the algorithms. In all cases special heuristics had been constructed to provide reasonable initial values to the solver.

In the first case it was possible to substitute the nonlinear terms by equivalent linear terms involving binary variables. In the second and third problem dominated by pooling problems initial guesses for the fractional composition of the multi-component streams could be derived from a simplified linear model. A homotopy method is used in the third problem by solving a sequence of sub-models of increasing complexity (LP, NLP-1, NLP-2, MINLP) exploiting the results of the previous ones. The third problem is solved in sequence of sub-models formulated in `GAMS`, exploiting the results of the previous one, *i.e.*, we use a homotopy method. A simple linear model provides initial values for two simple nonlinear submodels. In the fourth model scaling was very important. Solving an auxiliary problem in which an artificial objective function measuring the violation of certain nonlinear constraints was minimized provided excellent initial guesses.

Future direction regarding the first problem will focus on special branching rules and cuts to improve the gap. For the current application this is not a problem because only a few products required constraints across period. The third and fourth problem helped to accumulate experience in solving MINLP problems. However, the lesson to be learned is that each MINLP problem is different from others and requires special treatment and techniques. One common features seems to be the problem of getting good initial values to start the solver, which, according to our experience, can be overcome by homotopy techniques.

The heterogenous approaches to solve the problems indicate that mixed integer nonlinear optimization is an area under continual development. It has proven itself as a useful technique to reduce costs and to support other objectives, and it certainly has much to offer for the future. MINLP is another example that mathematical methods and techniques can support human inventiveness and decisions. Especially, they can ensure that less intuitive solutions are not lost, and can provide a quantitative basis for decisions and allow to cope most successfully with complex problems.

References

[1] R. W. Ashford and R. C. Daniel. LP-MODEL XPRESS-LP's model builder. *Institute of Mathematics and its Application Journal of Mathematics in Management*, 1:163–176, 1987.

[2] R. W. Ashford and R. C. Daniel. Practical aspects of mathematical programming. In A. G. Munford and T. C. Bailey, editors, *Operational Research Tutorial Papers*, pages 105–122. Operational Research Society, Birmingham, 1991.

[3] M. S. Bazaraa and H. D. Sherali and C. M. Shetty. *Nonlinear Programming, Theory and Algorithms*. Wiley, New York, 1993.

[4] Beate Brockmüller and Laurence Wolsey. BASF Problem 1 modelling - Production Scheduling with Batch-Sizing and Storage Restrictions. Report of ESPRIT project PAMIPS, Task 1.4 8/1995, BASF-AG, ZX/ZC-C13, 1995.

[5] Anthony Brooke, David Kendrick, and Alexander Meeraus. *GAMS - A User's Guide (Release 2.25)*. Boyd & Fraser Publishing Company, Danvers, Massachusetts, 1992.

[6] Marco A. Duran and Ignacio E. Grossmann. An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programms. *Mathematical Programming*, 36:307–339, 1986.

[7] C. A. Floudas. *Nonlinear and Mixed Integer Optimization*. Oxford University Press, Oxford, UK, 1995.

[8] Josef Kallrath, Yves Pochet, and Serge Raucq. BASF Problem 1 modelling - Production Network Planning System. Report of ESPRIT project PAMIPS, Task 1.1 10/1994, BASF-AG, ZX/ZC-C13, 1994.

[9] Josef Kallrath and John M. Wilson. *Business Optimisation Using Mathematical Programming*. Macmillan, UK, 1997.

[10] S. Leyffer. *Deterministic Methods for Mixed Integer Nonlinear Programming*. Phd thesis, Department of Mathematics and Computer Science, University of Dundee, Dundee, Scotland, 1993.

[11] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley, New York, 1988.

[12] J. Viswanathan and Ignacio E. Grossmann. A Combined Penalty Function and Outer-Approximation Method for MINLP Optimization. *Comp. Chem. Eng.*, 14(7):769–782, 1990.