Yugoslav Journal of Operations Research
8 (1998), Number 1, 9-23
Jovan Petrić memorial volume

ROUNDING STRATEGIES FOR MIXED INTEGER PROGRAMS ARISING FROM CHEMICAL PRODUCTION PLANNING*

Rainer E. BURKARD, Michael KOCHER, Rüdiger RUDOLF
Institute of Mathematics B, Technical University Graz,
Steyrergasse 30, A-8010 Graz, Austria

In memoriam Professor Jovan Petrić

Abstract: In this paper we consider problems which stem from production planning processes in the chemical industry. Many of these problems may be formulated as mixed integer linear programs. Since it is a big deal to obtain an optimal solution of this model in a reasonable amount of time, the design of fast and efficient heuristics is very important for practical purposes. In this paper we investigate heuristic approaches which consist of different rounding strategies based on an optimal solution of the corresponding linear relaxation of the integer program. Computational experiences on practical data are reported.

Keywords: Multi-product and multi-facility production planning, scheduling problems, heuristics, rounding strategies.

1. INTRODUCTION

Due to the internationalization of our markets and the increasing competition in all areas of industry, it becomes more and more important for any company to reduce costs and save resources. To circumvent this new situation simply by laying off employees seems not to be the right way, in most cases structural changes have to be

* This research has been supported by the Spezialforschungsbereich F 003 "Optimierung und Kontrolle", Projektbereich Diskrete Optimierung.

performed. One possibility is to optimize the production process by applying mathematical methods stemming from operations research.

In the following we concentrate on production planning problems from the chemical industry, since the production processes arising in this area incorporate most of the characteristics of general production planning problems in process industry (see Westenberger and Kallrath [5] for benchmark problems).

We model here the production process by a mixed integer linear program (MIP). This MIP is tried to be solved with the help of software packages, like CPLEX, MINOS etc. It turned out that this approach is not useful, since due to the large number of binary variables in the MIP formulation of the problem, the time for obtaining an optimal solution is quite far from being practical.

To obtain fast solutions for this sort of problems one may think of various heuristics which compute suboptimal solutions in a reasonable amount of time. One possibility is the following heuristical strategy: First relax all integer constraints and solve the corresponding pure linear program to optimality and secondly use the optimal solution of this linear program to obtain somehow a good integer solution for the original MIP. In this paper we pursue this idea and investigate several rounding strategies for obtaining promising integer solutions. Moreover, we discuss the advantages and arising difficulties.

The paper is organized as follows: Section 2 gives a general description of the production planning problem under investigation and its MIP formulation in particular. In Section 3 several rounding strategies are described and, finally, in Section 4 the computational results are reported.

2. PROBLEM DESCRIPTION AND MIP MODEL

The problem type under consideration arises in multi-product batch processing in the chemical industry and can roughly be described as follows: Processing starts with a set of raw materials from which a given set of final products is to be produced by a sequence of chemical processes. The production process can be represented by a network of multi-product facilities which are linked by convergent, divergent and cyclic material flows (a certain example of such a network is given in Appendix A). Typically, a given product can be produced on several facilities and a facility can be used to produce several products. The processing tasks are performed in batch-mode which means that there are minimum and maximum batch sizes for each task which depend on the involved chemical reactions and the capacity of the used reactors. It is assumed that the processing of a batch is carried out without interruption and that the material transfer times are neglectable.

The processing times only depend on the reactions performed and on the facility used, but not on the batch sizes. Furthermore, it has to be taken into account that there exist two types of intermediate products, (i) those which can be stored up to a given storage capacity and (ii) those which cannot be stored and must undergo further processing immediately (no-wait conditions). A further complication is added to the problem by the fact that for some processing tasks the ratios of all inputs and outputs are not fixed, but may vary within certain limits.

A feasible solution to the production planning problem described above can be represented by a production schedule which specifies

- the sequence of the batch processes at each facility,
- the sizes of the batches,
- the assignment of production tasks to facilities,
- the start and the finish time of each batch process, and
- the distribution of the materials and the inventory levels over time.

A production schedule is said to be feasible, if it satisfies the given requirements for the final products and takes the constraints on the batch sizes and the storage amounts into account. The optimization goals may differ. In this paper we only consider problems where the objective is to find a feasible production schedule with minimum makespan, i.e. the time when the required amount of all final products is available should be minimized. (Other objective functions could be modeled in a similar fashion.)

In a first step the entire time horizon is discretized into a number of smaller periods of the same length. The length of this period is determined as the largest common divisor of all batch processing times, such that each individual processing time of a batch may be expressed as an integer multiple of the period length and it may be assumed w.l.o.g. that each batch starts and ends at period boundaries.

After having determined the length of time-slots we next fix a time horizon $t_{\max}$, in which the required amount of final products can be produced and denote the set of feasible time-slots by $T = \{ 1,..., t_{\max} \}$. Let P be the set of products and $P_E \subset P$ the set of final products, F be the set of facilities used in the production process and R_{F_i} be the set of chemical processes (reactions) which are performed on the facility F_i. Following technological restrictions we distinguish mainly between two different types of reactions, namely reactions in which the ratio of the ingredients is fixed in advance (reactions of type I) and reactions for which these ratios may vary in between prespecified upper and lower bounds (reactions of type II). For each reaction r we denote the corresponding processing time which is independent of the actual batch size by τ_r, and by $m_{r,t}$ we denote the actual batch size of reaction r starting at time t. By introducing the variables $x_{i,t}$ for $i \in P$ and $1 \le t \le t_{\max}$ denoting the stock size of

product i at time t we are able to formulate the mass conservation via the subsequent stock balance constraint:

$$x_{i,t} = x_{i,t-1} + \sum_{r \in R_{P_i}} \bar{f}_r^i \cdot m_{r,t-\tau_r} - \sum_{r \in R_{P_i}^*} f_r^i \cdot m_{r,t} \qquad \forall i \in P, \quad \forall t \in T .$$

Here R_{P_i} denotes the set of reactions in which product i is input and by $R_{P_i}^*$ the set of reactions which produce product i as final product. The constants $\bar{f}_r^i$ denote the ratio of product i as ingredient to reaction r and f_r^i the ratio of product i as final product of reaction r. If r is a reaction of type II, a slight modification of the constraints above is necessary (see Appendix A for a short description). Additionally, the stocks must fulfill

$$x_{i,0} = L_i \qquad \text{and} \qquad x_{i,t} \le U_i \qquad \forall i \in P, \quad \forall t \in T$$

where L_i is the initial stock size of product i and U_i is the capacity of the stock for product i. By setting $U_i := 0$ one can model no-wait conditions for products which cannot be stored and have to undergo further processing steps immediately. Finally, to ensure the production of a certain amount B_i of each final product i, $i \in P_E$, we have the condition

$$B_i \le x_{i,t_{\max}} \qquad \forall i \in P_E .$$

The binary variables $s_{r,t}$ indicate whether reaction r starts at time t ($s_{r,t} = 1$) or not ($s_{r,t} = 0$). They are used in the following constraints which guarantee the exclusive assigning of one reaction to a facility at the same time,

$$\sum_{r \in R_{F_i}} \sum_{t'=t-\tau_r+1}^{t} s_{r,t'} \le 1 \qquad \forall i \in F, \quad \forall t \in T \tag{1}$$

and for constraints on the batch sizes:

$$A_{\min}^r \cdot s_{r,t} \le m_{r,t} \le A_{\max}^r \cdot s_{r,t} \qquad \forall r \in R, \quad \forall t \in T,$$

where $A_{\min}^r$ and $A_{\max}^r$ are given lower and upper bounds for the batch size of reaction r.

To formulate the objective function – minimizing the makespan – we introduce further binary variables f_t, indicating whether or not a production is

performed in period t. By replacing the right-hand side of constraint (1) by f_t and adding constraints of the form

$$f_{t+1} \le f_t \qquad \forall\, t \in T \setminus \{ t_{\max} \}$$

the minimization of the makespan leads to

$$\min \sum_{t \in T} f_t \,. \tag{2}$$

Since $s_{r,t}$ are already binary variables, the condition $f_t \in \{ 0, 1 \}$ may be relaxed to $0 \le f_t \le 1$ without changing the problem itself.

During the computational tests with the relaxation of MIP it turned out that better solutions are obtained whenever the original objective function (2) is replaced by

$$\min \sum_{t \in T} t^2 \cdot f_t \,. \tag{3}$$

Note that in the case that $s_{r,t}$ and f_t are binary variables, both objective functions (2) and (3) are equivalent, but in the case of continuous variables in the LP-relaxation the objective function (3) punishes reactions which are performed later in time. Moreover, the subsequent redundant constraints are added to the original mixed integer linear program:

$$\sum_{t'=t}^{t+\tau_r-1} s_{r,t'} \le \sum_{r \in S_r} \sum_{t'=t+\tau_r}^{t_{\max}} s_{r,t'} \qquad \forall\, r \in R, \quad \forall\, t \in T$$

meaning that whenever a reaction is performed on a facility at least one succeeding reaction has to be performed, too. The set S_r denotes all reactions succeeding reaction r.

Starting from the required amount of final products one may compute in a recursive way a lower bound on the required amount of each raw material and each intermediate product. If we denote this amount by B_i for each product i, the minimal number of batches which is necessary to produce the desired amount of product i is given by

$$V_i := \left\lceil \min_{r \in R_{P_i}} \left\{ \frac{B_i}{A^r_{\max}} \right\} \right\rceil \qquad \forall\, i \in P \,.$$

Thus for each product i we add the subsequent constraint to our model MIP:

$$\sum_{r \in R_{P_i}} \sum_{t=1}^{t_{\max}} s_{r,t} \geq V_i \qquad \forall i \in P .$$

Before discussing different rounding strategies for obtaining good integer solutions of the mixed integer linear program described above let us mention two facts on the number of binary variables. Although some of the binary variables may be set to zero already in advance (note that reactions may not start before all ingredients are available and that it makes no sense to start reactions which cannot finish on time) the number of binary variables is still very high. Note that the number of binary variables $s_{r,t}$ highly depends on one hand on the discretization of time and on the other hand on the chosen constant $t_{\max}$. Since the correct value of $t_{\max}$ is not known a priori (it is the optimal value of MIP), good estimations for this parameter are desirable and may be obtained by using fast heuristics.

3. ROUNDING STRATEGIES

Owing to the large number of binary variables in the mixed integer linear program MIP the computation of the optimal solution is too time consuming. Therefore we propose to relax the integrality constraints of the variables $s_{r,t}$ to $0 \leq s_{r,t} \leq 1$ and denote this relaxation of MIP as LP. In the remainder of this section we will describe several strategies how the optimal solution of the LP can be used to find good suboptimal solutions of the mixed integer program. Computational experiments and comparisons between all strategies are given in the next section.

3.1. Strategy 1

A very first and naive approach to obtain an integral solution is the following. After having solved the LP a fixed parameter p with $0 < p < 0.5$ is chosen and all continuous variables $s_{r,t}$ with $s_{r,t} < p$ are fixed to zero, whereas all variables $s_{r,t}$ with $s_{r,t} \leq 1-p$ are fixed to one. Variables whose value lies in between $[p, 1-p)$ remain unchanged. Then the LP is reoptimized. This procedure is repeated until either all variables $s_{r,t}$ are already binary or no variable is fixed during the rounding step. In this case all remaining variables $s_{r,t}$ are rounded in the usual way, i.e. $s_{r,t}$ is set to zero, if $s_{r,t} < 0.5$ and 1 otherwise.

Figure 1 gives a short description of Strategy 1 with $p = 0.2$.

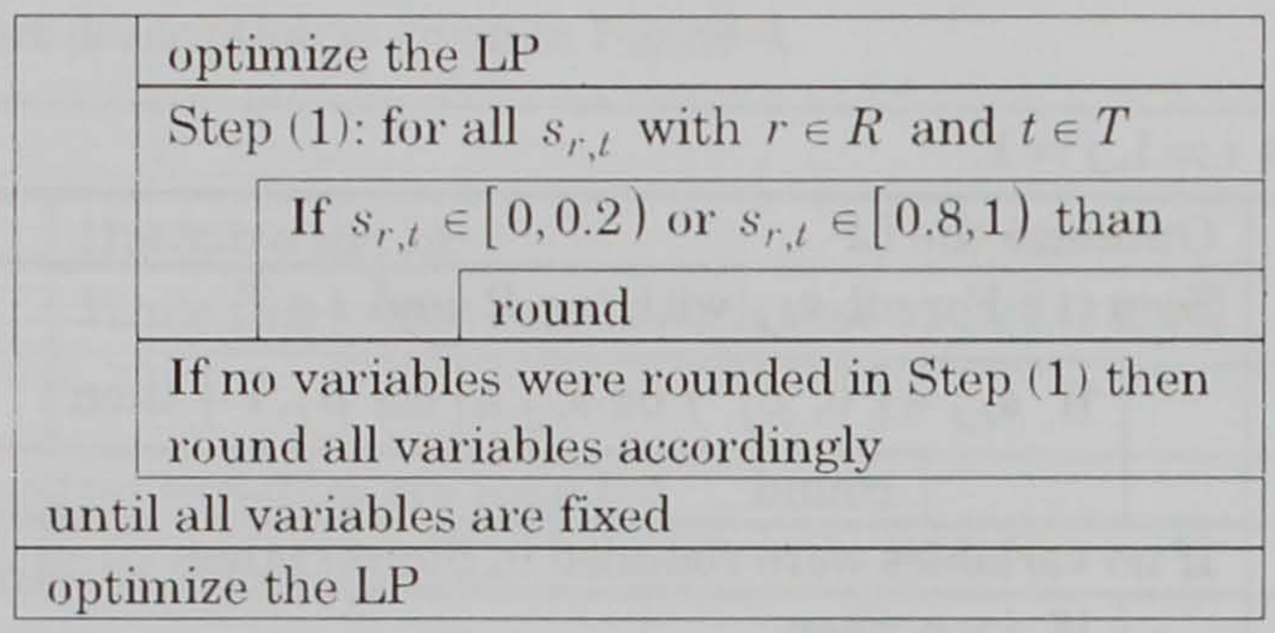


Figure 1: Schematic description of Strategy 1

3.2. Strategy 2

The subsequent rounding strategy is a modification of the pure approach described as Strategy 1. There are two modifications: First, instead of chosing one parameter p for an appropriate rounding, a set of n parameters p_1, p_2 up to p_n is chosen satisfying $0 < p_1 < p_2 < \cdots < p_n = 0.5$ and secondly, the entire time horizon is divided into several time periods T_1 up to T_k with $T_1 = \{ 1,\ldots,t_1 \}$, $T_2 = \{ t_1 + 1,\ldots,t_2 \}$ and $T_k = \{ t_{k-1} + 1,\ldots,t_{\max} \}$. Then the rounding scheme described in the first strategy is applied to each time period T_i starting from $i = 1$ to k while the actual rounding parameter p takes all values from p_1 up to p_n. Let us assume that i is the actual time period. First, p is fixed to p_1. As long as there exist variables $s_{r,t}$ with $r \in R$ and $t \in T_i$, such that either $s_{r,t} < p$ or $s_{r,t} \geq 1 - p$, we round accordingly and leave all parameters unchanged. Otherwise we increase the value of p to the next parameter on the list, say p_2, and repeat. If all variables in the actual time period are fixed (holds since $p_n = 0.5$), we proceed to the next time period T_{i+1} and start again with the rounding parameter p_1.

Note that by fixing $T_1 = \{ 1,\ldots,t_{\max} \}$ and $p_1 := p$ and $p_2 = 0.5$ we end up with the simple Strategy 1. The scheme of Strategy 2 is given in Figure 2.

3.3. Strategy 3

The next rounding strategy is a small modification of Strategy 2. Instead of rounding all variables of a certain time period T_i at the same time, only one variable $s_{r,t}$ with $s_{r,t} < p_j$ or $s_{r,t} \leq 1 - p_j$ is selected randomly and rounded in each iteration. If no such variable exists, proceed as in Strategy 2 (see Figure 3).

Set $i := 1, j := 1$.
  Optimize the LP
  Step (1): For all $s_{r,t}$ with $r \in R$ and $t \in T_i$
    If $s_{r,t} \in [\, 0, p_j \,)$ or $s_{r,t} \in [\, 1 - p_j, 1 \,]$ then
      round
  If no variables were rounded in Step (1) then
    If $j = n$ then
      Fix $i := i + 1$ and $j := 1$.
    otherwise
      Fix $j := j + 1$.
until all binary variables are fixed
optimize the LP

Figure 2: Schematic description of Strategy 2

Set $i := 1, j := 1$.
  Optimize the LP
  Step (1): Determine r and t such that $s_{r,t} \in [\, 0, p_j \,)$ or $s_{r,t} \in [\, 1 - p_j, 1 \,]$ and round this variable
  If no variable was rounded in Step (1) then
    If $j = n$ then
      Fix $i := i + 1$ and $j := 1$.
    otherwise
      Fix $j := j + 1$.
until all binary variables are fixed
optimize the LP

Figure 3: Schematic description of Strategy 3

3.4. Strategy 4

Another possibility in designing a rounding scheme based on the linear relaxation of MIP is the following adaptation of Strategy 1. First optimize the linear program and round all variables $s_{r,t} \geq 1 - p$ and then reoptimize the LP until no such rounding step can be performed any longer. Instead of rounding the remaining

variables as proposed in Strategy 1, here the original mixed integer linear program is optimized. A short description is given in Figure 4.

```
Optimize the LP
While there exist r and t such that s_{r,t} ∈ [ 0.8, 1 ]
    set s_{r,t} = 1 .
until no variables are rounded
optimize the MIP
```

Figure 4: Schematic description of Strategy 4

3.5. Strategy 5

This heuristic runs as follows: After every optimization step of the LP we choose the largest binary variable $s_{r,t}$ among all variables $s_{r,t} \in (0,1)$ which are not fixed already, fix it to 1 and put the name of the variable on a stack. This procedure is repeated until we either obtain a feasible integer solution or the linear program becomes infeasible. In this case we use the last name of the variable put on the stack and do the following: if this variable, say $s_{r,t}$ is 1, we set it to zero, put its name again on the stack and proceed by reoptimizing the LP. Otherwise, if $s_{r,t} = 0$, we free this variable again and take the next variable from top of the stack and repeat. This heuristic is performed until either the last variable put on stack is freed again – in this case no feasible integer solution exists – or a feasible integer solution is determined. Figure 5 contains a schematic description of this heuristic with backtracking.

```
Optimize the LP
If the solution is feasible then
    search for the largest s_{r,t} ∈ (0,1)
    set this variable to 1 and put it on the stack
otherwise
    until the last variable on the stack is set to 0
        set the bounds to 0 or 1, respectively
        delete this variable from the stack
    fix the last variable on the stack to 1
until all binary variables are fixed
```

Figure 5: Schematic description of Strategy 5

3.6. Strategy 6

In this strategy we combine some ideas of Strategy 2 and Strategy 5. Thus we divide again the entire time horizon into distinct periods $T_1, ..., T_n$ and define a rounding parameter p. From Strategy 5 we use the idea of branching along variables with backtracking with a slight modification. In this new strategy we do the following: After having solved the LP we fix all variables $s_{r,t}$ whose value is not smaller than p in the actual time period to one. All names of the variables are put on the stack at once. If no variable exists for rounding, we simply choose the largest variable $s_{r,t}$ contained in this time-period, fix it to one and put its name on the stack. If all binary variables are already fixed in a certain time period, we look at the next time period and repeat. At each time when the underlying LP becomes infeasible, we take the last names of variables put on the stack: if this is only one variable then we act as in Strategy 5, if more variables occur, we free all of them and fetch the next names from the stack. Finally, when working in the last time period and there are still some variables not yet fixed, we return to the first time period again and repeat until all variables are set to zero or one. For a description see Figure 6.

```
Set i := 1.
    Optimize the LP
    If the solution is feasible
        Search for the largest variable s_{r,t} ∈ (0,1), r ∈ R, t ∈ T_i
        If there exists no such variable
            Set i := i + 1.
            If i > n and some variables are continuous then
                Set i := 1.
        else
            if this variable is larger than p
                round all variables larger than p and
                put them on the stack at once
            else
                round this variable and put it on the stack
    else
        free elements that contain more than one variable and
        delete them from the stack
        while the last variable is set to 0
            set the bounds to 0 or 1, resp.
            delete it from the stack
            free and delete all elements with more than 1 var.
        set the last variable to 0
until all variables are fixed
```

Figure 6: Schematic description of Strategy 6

4. TEST PROBLEMS AND COMPUTATIONAL RESULTS

To evaluate the rounding strategies of the previous section we used the benchmark problems described in Westenberger and Kallrath [5]. (A short description of the processing network and all data is given in Appendix A.) We tested several problems which vary only in the amounts of their final products. In all test problems we used the same processing network and the parameters of all facilities as well as of all stocks remain unchanged. In all test problems three different final products are to be produced. Table 1 contains all different problems together with the considered values of $t_{\max}$ and the corresponding number of binary variables. Here Problem 24-13-10 means that 24 kg of final product P_1, 17 kg of final product P_2 and 13 kg of final product P_3 are required.

Table 1: Problem description with different $t_{\max}$ and number of binary variables

Problem	$t_{\max}$	bin. variables
12-7-7	24	432
24-13-10	34	612
	40	720
30-17-14	45	810
	50	900
60-0-0	50	900
45-25-20	65	1170
	75	1350
30-30-30	75	1350
90-50-40	110	1980
	120	2160

All strategies were implemented in C and the callable library of CPLEX was used for solving the arising linear programs as well as the mixed integer linear program in case of Strategy 4. All test runs were performed on a HP9000/J200 workstation with 512 MB memory capacity.

During the computational study it turned out that Strategies 1 to 3 are not suited for obtaining a good feasible solution of MIP. More precisely, the structure of the underlying processing network and the connection between several facilities and the capacities of various stocks seems to be too complicated for this kind of rounding strategies. In fact in almost all cases – by choosing various parameters – no feasible solution could be found at all. So a simple rounding scheme without backtracking cannot be considered as a useful heuristic for multi-product and multi-facility production planning problems.

The situation for Strategy 4 is a bit different. Although this method shows a good behaviour for the smaller problems (for the problems 24-13-10 and 12-7-7 even the optimal solutions have been computed within several minutes), for problems containing more binary variables this method seems to be inappropriate (e.g. problem 30-17-14 could not be solved within several hours).

Strategy 5 and Strategy 6 show the best behaviour of all strategies described in the previous section. Whereas Strategy 5 is only suitable for smaller problems (the running times increase drastically for larger problems), the heuristic based on Strategy 6 could find a feasible integer solution for almost all problems within 15 minutes. Only Problem 90-50-40 could not be solved in one hour. Table 2 shows the running times in CPU-seconds and the solution values obtained for the test problems. Some of the problems were also solved for different values of parameter $t_{\max}$. We obtained the best computational results both from the running times as well as from the objective value by setting $T_i = \{ i \}$, i.e. by fixing the length of each time period to 1, and by setting $p = 0.8$.

Table 2: Solution Values and CPU-times in seconds for all test problems

Problem	Solution Value	$t_{\max}$	CPU-time
12-7-7	20	24	6.89
24-13-10	33	34	30.61
	34	40	106.92
30-17-14	43	50	213.25
	44	45	111.37
60-0-0	44	50	58.12
45-25-20	56	75	452.78
	59	65	376.14
30-30-30	67	75	948.84
90-50-40	103	110	5233.82
	105	120	4212.33

We also compared our heuristics based on rounding strategies with other approaches known from the literature which have already been applied to these benchmark problems. Ahleff [1] and Rosenau [4] try to solve the MIP by means of standard software packages for mixed integer programming, but – not very surprisingly – it turned out that this approach is only suitable for very small problems even if preprocessing and other techniques are used to speed up the solution process. Blömer and Günther [2] propose several other LP-based heuristics to obtain near-optimum production schedules. But in contrast to our approaches they approximate the original MIP with another mixed integer linear program having a smaller number of binary variables by e.g. changing the processing times or concentrating to only one facility. Another approach described in Kreßmaier [3] works in a greedy fashion. When

comparing our results to the results obtained by these heuristics, it can be seen that the solutions we found are for almost all test problems the best solutions known so far.

We close this section by mentioning that the parameter t_{max} influences both the quality of the obtained solution value and the running times. Here two different situations can be observed. For the smaller test problems a larger value of t_{max} causes an increase in the running time but often leads to a better solution value. For the larger test problems the situation seems to be different. Here a decrease of t_{max} reduces the number of feasible solutions drastically and consequently the running time increases because more rounding and backtracking steps to find a feasible solution must be performed. This behaviour is illustrated in Table 3. The first three columns contain the problem description, the fourth column the number of variables $s_{r,t}$ which are already integral in the optimal solution of the initial LP relaxation, the fifth one the number of rounding steps performed by the heuristic and the last column the number of backtracking steps necessary to obtain a feasible solution.

Table 3: Solution values and problem size versus number of rounding and backtracking steps in Strategy 6

Problem	Solution Value	t_{max}	bin. variables in LP-solution	roundings	backtracks
24-13-10	33	34	566	20	5
30-17-14	43	50	825	46	7
90-50-40	103	110	1786	317	257
	105	120	1955	128	34

REFERENCES

[1] Ahleff, J., "Analyse von mathematischen Modellen der Reihenfolgeplanung in der Prozeß-industrie", Diploma thesis, RWTH Aachen, Germany, 1995 (in German).

[2] Blömer, F., and Günther, H.-O., "Scheduling of a multi-product batch process in the chemical industry", Technical Report 05/96, Department of Industrial Management, Technical University Berlin, 1996.

[3] Kreßmaier, U., "Heuristische Methoden zur Produktionsplanung auf einer chemischen Anlage", Diploma thesis, Technical University Graz, 1996 (in German).

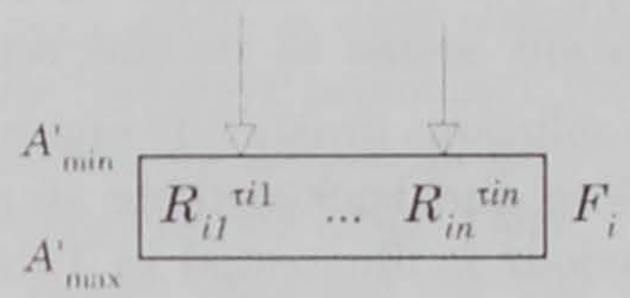
[4] Rosenau, B., "Bearbeitung eines Losgrößenproblems aus der chemischen Industrie mit Schnittebenenverfahren", Diploma thesis, Philipps University, Marburg, Germany, 1996 (in German).

[5] Westenberger, H., and Kallrath, J., "Formulation of a job shop problem in process industry", Working paper, Bayer AG, Leverkusen and BASF AG, Ludwigshafen, Germany, 1994.

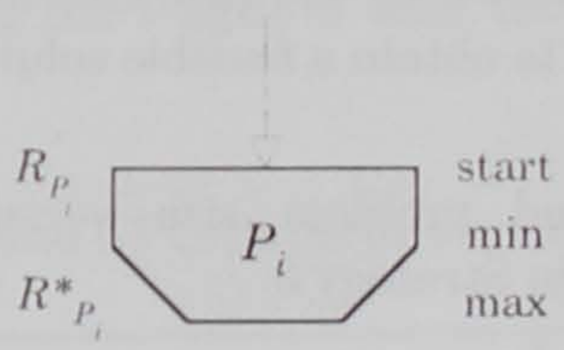
APPENDIX A: A PRODUCTION PROCESS

For the description of the production process, the following symbols are used:

Symbol for a facility F_i with n reactions and the bounds for the batch size $(A^r_{\min}, A^r_{\max})$.

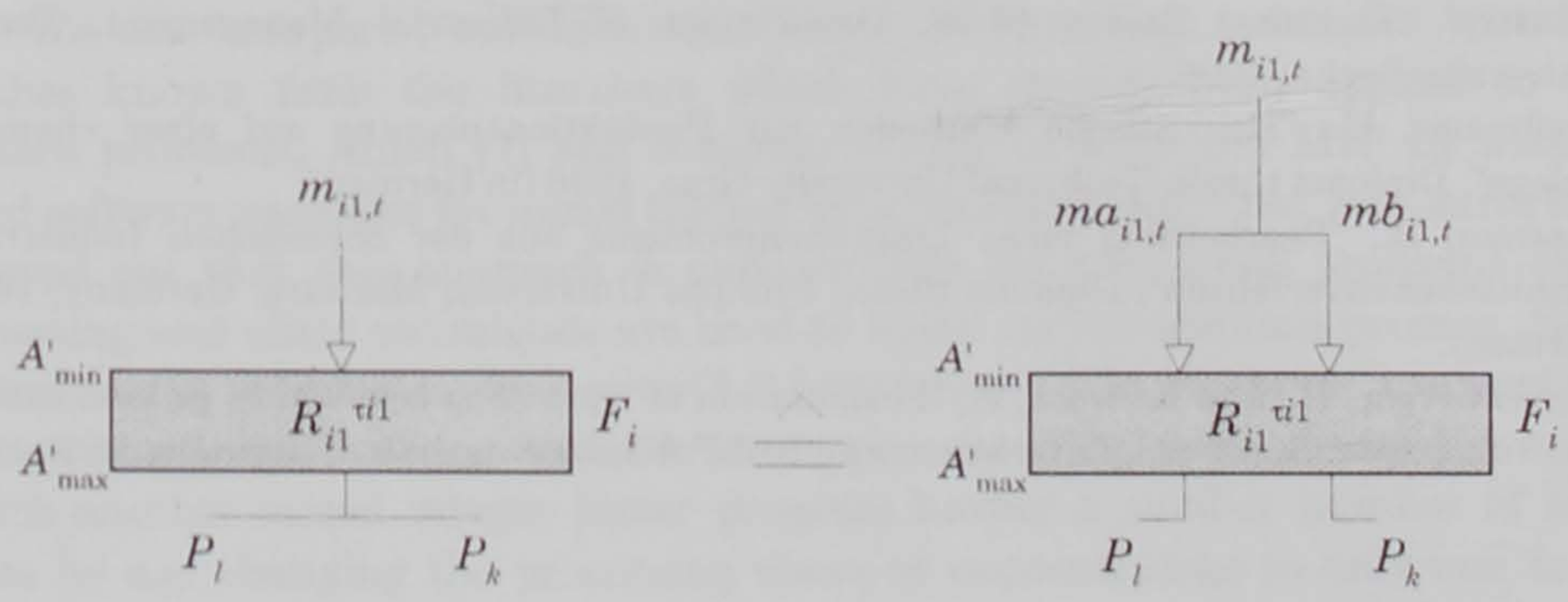


Symbol for the stock of the product P_i:



R_{P_i}	...	reactions that produce P_i
$R^*_{P_i}$	...	reactions that use P_i as input
start	...	initial stock size for product P_i at time $t = 0$
min	...	minimum inventory level for product P_i
max	...	maximum inventory level for product P_i

If R_i is a reaction of type II, the input variable has to be divided into two additional variables ma and mb to obtain a linear program. As a consequence, the amount of input for product P_l is defined by ma and the amount for P_k by mb, where the value of ma can vary in between specified upper and lower bounds.



Description of the production process: