# Alternatives to MILP for scheduling of batch operations

*Jaroslav Požívil*[1] *and M. Ždánský*

***Alternatívy k MILP pre rozvrhovanie dávkových banských procesov***

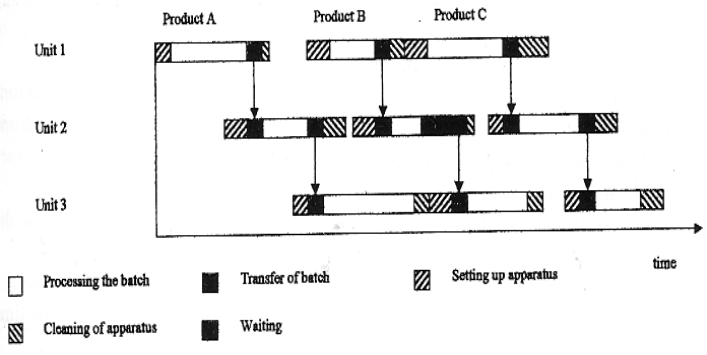
*Cieľom príspevku je navrhnúť možnosť, nahradenia optimalizačných metód na báze zmiešaného celočíselného programovania použitím približných metód rozhodovania (heuristika) v oblasti plánovania a riadenia banských procesov. Výsledkom tohto počítačom podporaovaného plánovania sú detailné výrobné rozvrhy vytvorené podľa požiadaviek ako sú vysoká efektivita výroby, alebo znížená redukcia odpadov. Tieto problémy patria medzi zložité, NP-úplné problémy, čiže ich riešenie je v súčasnosti podľa nášho názoru pre reálne problémy efektívnejšie s pomocou heuristík. Heuristiky uvedené v príspevku sú: simulované žíhanie, tabu search a genetické algoritmy. Tabu search(metóda zakázaného prehľadávania), aj keď sa jedná o heuristiku, je v štandardnej verzii deterministicky stachastická. V prispevku sú zhrnuté hlavné výhody heuristík v porovnaní s MILP, predovšetkým ich rýchlosť a jednoduchosť a teda aj nižšie nároky na výpočtovú techniku a software, ako aj kvalita poskytovaných výsledkov. Článok uvádza stručný popis riešených problémov a základy matematického popisu týchto problémov, zhrnuté sú aj rôzne ciele optimalizácie reprezentované rôznymi cieľovými kritériami. Príspevok ďalej popisuje jednotlivé heuristiky, ich princípy, ich vlastnosti a ich možnosti, požiadavky ktoré musí úloha splňovať, aby bolo možné algoritmus použiť. Prekaždú z uvedených heuristík uvádzame aj slovný popis jednotlivých častí algoritmu. Článok uvádza jednotlivé výsledky porovnania výkonov týchto heuristík oproti MILP, uvedené sú aj výsledky aplikácie týchto algoritmov na iné podobné problémy v chemickom priemysle. Vzhľadom k zámerom tohto príspevku text obsahuje aj odkazy na ďalšiu literatúru zaoberajúcu sa touto problematikou.*

## Introduction

Oplimization of many mining operations is not a simple lask. The complexity of such operations, as well as other factors, make this optimization very difficult but in a today's highly competitive world these problems must be solved. The current state of computer technology, i.e. its relatively low costs and high computing power, allows for a wide use of modern software tools in such applications, offering efficient allernatives to older methods. The aim of this paper is to present some of the akorithms that can be. and sometimes are. used for the optimization of various problems in the mining industry,

Our work focuses on productions where several different products are being manufactured using a similar technology (e. g. sorted coal). The production is, usually divided into several campaigns. during which only onr or a few of the whole range of products are manufactured. This brings the needí to determine the sequence of products so as to satisfy contractual obligations. Additionally. an exact sdiedule must be determined for all Operations during campaigns, i. e. available apparatures must be assigned to tasks and timing of all operations must be found. Solving these problems is called Production Planning and Scheduling (PPS). The production planning means sequencing the production in order to fulfill contractual obligations. The scheduling means the determining of timing all operations for all units (pieces oť equipment). Results of Ihis work are often used in ihe form of Gantt charts (Fig. l shows a sirnple example of such chart)

Fig. 1 Example of a Gantt chart




The focus of this paper is on miscella-neous batch plants (plus planta containing continuous and semicontinuous operations that can be treated as batch operations for the purpose of optimi-zation) that appear in mining industry, Mixed Integer (Non)Linear Programming (MI(N)LP) is often used when dealing with this kind of optimization problems but its features leave a lot to be desired. Therefore. three alternative algorithms are decscribed in this paper

[1] *Jaroslav Požívil,,*Department of Computing and Control Engineering, Fac ultyof Chemical Engineering, Prague Institute of Chemical Technology, Technická 1905, 166 28 Praha 6, Czech Republic, pozivilj@vscht.cz
(Recenzovaná a revidovaná verzia dodaná 24.1.2001)

## Formulation of the scheduling problem

Scheduling problems are often described using: formulations resulting in the MILP problem (Kondili e; al., 1993)

Batch (and discrete) processes are described using the State-Task Network representation (STN). This representation has two types of nodes; the state nodes, representing feeds. intermediate and final products and the task nodes, representing the processing operations. Two rules followed in its construction are:

1. A task has as many input (output) states as different types of input (output).
2. Two or more streams entering the same state must be of the same quality. Possible mixing of different streams should form a separate task.

It should be noted that STNs are not necesserily connected graphs. A number of parameters are associated with nodes defining the STN and with the available equipment items.

A task i is defined by:

$S_i$ : the set of states which the task i;

$\overline{S}_i$ : the set of states which the task i produces as its outputs;

$\rho_{is}$ : the proportion of input of task i from the state $s \in S_i$; $\sum_{s \in S_i} \rho_{is} = 1$;

$\overline{\rho_{is}}$ : the proportion of output of task i from the state $s \in \overline{S_i}$; $\sum_{s \in \overline{S_i}} \overline{\rho_{is}} = 1$;

$P_{is}$ : the processing time for the output of task i to state $s \in \overline{S_i}$ ;

$P_i$ : the completion time for task i, $p_i \equiv \max_{s \in \overline{S_i}} P_{is}$;

$K_i$ :the set of units capable of performing the task i.

The state s is defined by:

$T_s$ .: the set of tasks receiving a material from the state s;

$\overline{T_s}$ : the set of tasks producing a material in the state s;

$C_s$ : the maximum -.forage capacity dedicated to the state s.

The unit j (may perform one or more tasks) is characterized by:

$I_i$: the set of tasks that can be performed by the unit j;

$V_{ij}^{max}$: the maximum capacity of unit i when used for performing the task i;

$V_{ij}^{min}$: the minimum capacity of unit j when used tor performing the task i.

In this paper, the following additional assumptions are made: no task may be interrupted, the processing times of the tasks are fixed; the material is transferred instantaneously from states to tasks and from tasks to states, all data are deterministic and fixed over the time horizon of interest.

The mathematical formulation is based on the discrete time representation. The time horizon of operations is divided into intervals of equal duration. Events of any type can occur only at the interval boundaries. If some input data change during the task, or any other similar changes occur, the interval length can be reduced to ensure that all such change fit into the grid set by discrete time.

Based on the above STN and the time representation, following variables are introduced to describe the tasks and states:

1. $W_{ijt}$= 1 if the unit j starts processing the task i at the beginning of the time period t: otherwise it is zero.
2. $B_{ijt}$= the amount or material which starts undergoing the task i in the unit j at the beginning of the time period t,
3. $S_{st}$ = the amount of material stored in the states. at the beginning of the time period t.

Now we need to translate system limitations into explicit mathematical constraints. In this paper, the following contraints are taken into consideration

***Allocation constraints***

At any given time, an idle piece of equipment can only start one task at most. This leads to the mathematical constraint:

$$\sum_{i \in I_j} W_{ijt} \leq 1$$

Futhermore, no other task can be started until the current one is finished. This can be expressed as follows:

$$\sum_{i' \in I_j} \sum_{t' \in t}^{t+p_i-1} W_{j'jt'} - 1 \leq M(1 - W_{ijt}) \forall j, t, i \in I_j, \quad (1)$$

where M is a sufficiently large positive number.

***Capacity limitations***

The amount of material that starts undergoing the task i in the unit j at the time 1 is liminted by the maximum and minimum capacities of that unit:

$$W_{ijt} V_{ij}^{\min} \leq B_{ijt} \leq W_{ijt} V_{ij}^{\max} \forall i, t, j \in K_i \quad (2)$$

The amount of material stored in a state s must not exceed the maximum storage capacity for this slate at any time:

$$0 \leq S_{st} \leq C_s \forall s, t. \quad (3)$$

***Material balances***

Material balances are expressed mathematically as:

$$S_{st} = S_{s,t-1} + \sum_{i \in \overline{T}_s} \overline{\rho}_{is} \sum_{j \in K_i} B_{i,j,t-p_{is}} - \sum_{i \in \overline{T}_s} \rho_{is} \sum_{j \in K_i} B_{ijt} \forall s, t. \quad (4)$$

It is possible to add many other constraints. Examples found in literature include product deliveries and raw material receipts during the horizon, the temporary unavailability of equipment, the limited availability of utilities alto manpower, the sequence-dependent maintenance and the frequency dependent maintenance.

Finally, the scheduling problem for a batch processing system can be stated as:

| | |
|---|---|
| *Given* | the STN ot a batch process and all the information and additional data associated with it and a time horizon of interest. |
| *Required* | sequencing of products; assignment and timing of operation, for each unit (i.e. which task, if any. the unit performs at any time during the time horizon) and the flow of material through the network. |
| *So as so optimize* | a given objective criterion. |

Objective criterion in question can be defined as a combination of several requirements. Such requirements typically include:

- satisfying contractual obligations (product types, quantities, times of delivery).
- maximization of profit,
- minimization of completion time,
- maximal and/or uniform utilization of equipment.
- minimization of changeover costs».
- optimizing schedule of obtaining feedstocks.
- minimizing cost of storage of material (raw materials, intermediate states, products).

Examples presented later in this paper usually use objective criterion in the form of minimization of completion lime.

What makes this problem so difficult to solve is the fact that it is the NP-complete problem. This means that no deterministic algorithm offering the computation time as a polynomial function of problem dimension exists and although the solution can be obtained in the polynomial time it is only possible with the use of heuristics. Additional information on theory of NP-completeness can be found the in literature (Garey e. al *1979)*

### Drawbacks of MILP approach

While the solutions obtained using MILP can he guaranteed to be optimum (however, they are not, necessarily unique), several problems discourage its use.

The main difficulty is the size of resulting problems. For realistic problems, the MILP problem may involve thousands (even tens of thousands) of variables.

Furthermore, even smaller problems involve hundreds of variables and require substantial computation times. While many algorithm improvements and modifications, together with a faster computer hardware, appeared in recent years. MILP-based solvers remain somewhat unsuitable for practical use. Using the MILP for solving the scheduling problems requires a modern (and expensive) software solver, running on a fast computer. Such tools require an expert to use them, which means that a company in question does have to hire one (or have somebody trained). Many examples of computation time requirements can be found in literature (Stluka, 1998): e.g. the optimization of rnultiproduct plant *(6* apparatuses. 9 products) using the MILP took 8 hours and 45 minutes (Barbosa-Póvoa et al.. 1994), campaign planning (6 apparatures, 9 operations. 2 campaigns. 3 products) took 6 hours IS minutes (Papageorgiou et al., 1993).

The resulting schedule can also prove to be unusable for the real problem because some of the real-world limitations are left out. or human errors caused by difficult and complex mathematics involved can lead to the erroneous formulation and consequently to the bad solution.

### Alternative stochastic methods

Three methods presented in this paper are often used as replacements for MILP. These are: the simulated annealing, genetic algorithmus and the tabu search(tabu search in its basic form is not in fact a heuristic algorithm, but it's usually grouped together with them). It can be said that all these methods search the solution space while gradually improving the objective value of the best solution found. The principle of this can be illustrated by the graph showing the objective function value during the optimization of some random problem using tabu search algorithm as shown in Fig.2.

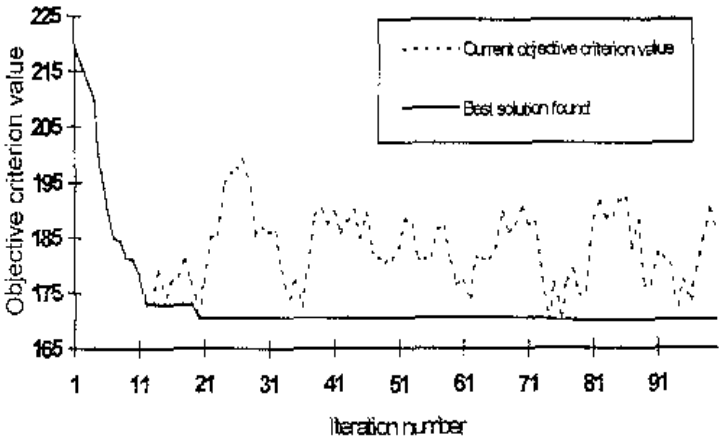



*Fig.2. Example of an optimisation run.*

All these stochastic methods try to search only a part of the solution space, and this leads to great time-savings, especially in the cases of larger problems. Table 1 illustrates this, using the data obtained in the flowshop optimization studies (Stluka. 1998). by comparison with the simple enumeration method (i.e. testing all possible solutions"). The principles of stochastic methods do not guarantee finding the global optimum, but they do find good sub-optimum solutions.

| | 10 products | | 20 products | |
|---|---|---|---|---|
| | Solutions generated | Solution space searched [%] | Solutions generated | Solution space searched [%] |
| Enumeration | 3 628 800 | 100 | $2.43*10^{18}$ | 100 |
| Simulated annealing | 70 000 | 1.93 | 110 000 | $4.52*10^{-12}$ |
| Tabu search | 24 578 | 0.68 | 151 715 | $6.24*10^{-12}$ |
| Genetic algorithm | 63 705 | 1.76 | 88 596 | $3.64*10^{-12}$ |

*Tab.1. Effectiveness of the optimization methods*

***Simulated annealing***

Simulated annealing algorithmus are based on an analogy with the way metals cool and anneal. During a slow cooling of heated metal, atoms begin to align themselves into crystals. During the cooling process transitions are accepted between low and high energy levels. The probability distribution describing this is the Bolmnar.n distribution. FinalK. a pure crystal is formed, corresponding to the minimum energ; le\cl allowed for the system. If a metal is cooled quickly and the substance is allowed to deviate from the equilibrium, the state of minimum energy is not reached and the system ends in a metaslablc. locally optimal structure. The optimisation algorithm tries to simulate this physical process. Let $E_1$, be the objective function value of the initial solution and $T_i$ the initial temperature. The algorithm consists of two loops. In the inner loop, a new solution $E_{2:}$ is generated and i; $E_2 < E_1$. the new solution is made the current solution and ihe process is repealed. In the case of $E_{2:} > E_{1,}$ the new solution is accepted/rejected randomly with the probability based on the Boltzmann distribution:

$$P(\Delta E) = \exp\left(\frac{E^i - E^j}{K_B T}\right), \qquad (5)$$

where $K_s$ is the Boltzmann constant and T is the current temperature. If ihe solution is accepted, it becomes the current solution and the process continues; if the solution is rejected, another solution is generated from the current one. In the outer loop, the temperature is decreased. Two methods for deciding when to decrease the temperature arc widely used. The first method decreases the temperature as soon as the solution with lower objective function value is found. Second method decreases the temperature after reaching the limit of L rearrangements.

To apply the Simulated annealing algorithm, the following problem-dependent issues must be resolved:

a) an objective function to be minimized,
b) denning possible solutions based on given constraints,
*c)* method of generating random solutions from a current solution.

Using the theory of Markov chains, it was shown (hat the simulated annealing algorithm is guaranteed to converge to the set of optimum solutions, given an infinite number of iterations. However, this asymptotic behavior can be approximated in the polynomial lime Iterations are stopped when a defined termination criterion (pre-set temperature $T_f$, is often used) is met, and such approximation algorithms are known to give near-optimum solutions.

Additional information about simulated annealing algorithms, e.g. defining the cooling schedule, can be

found in some of the works listed in references (Cardoso et al.. 1997; Ku et al., 1991).

***Tabu search***

The basic tabu search heuristics starts from an initial solution; at each step a move to a neighboring solution is chosen in such a way that hopefully objective function value improves. This is close to a local improvement technique except for the fact that a move to a solution worse than the current solution may be accepted. The algorithm tries to take steps to assure that the method does not re-enter a solution previously generated which serve as a way to avoid becoming trapped in local optimum. This is accomplished by creating a list of tabu moves: these are moves which are discouraged at the current iteration. A move remains restricted only during a limited number or iterations. Some details on this algorithm as well as on its modifications are summarized below.

For practical purposes, a correct selling of a tabu list of length L is of highest importance. Setting this value too small can lead to the algorithm cycling through a sequence of moves. Using too long tabu list means that promising moves are often rejected, and the computation time increases;. The optimum tabu list length depends or. the dimension, objective function and the exact type of the problem.

Selecting a next move from a neighborhood can be done using two different approaches. The steepest descent technique, common in local optimization algorithms, searches a whole neighborhood and then selects a move with the best objective function value. The alternate method is fastest descent technique, which means using the first move leading to the objective function value improvement. Advantages of this technique are evident during the beginning steps of the algorithm, but the computation time requirements later increase again as ways for quick improvements disappear.

Inclusion of some form of long-term memory is an interesting modification of the basic tabu search algorilhm. The transformation usage frequency $\omega(t)$ is used as a replacement for the short-term memory. This function is used as a penalty component of an objective function when heuristics searches neighborhood for ihe next move. This modified objective function can be expressed as: $f(x)+\alpha\omega(t)$. where $\alpha$ is the weight to be set empirically;. The main problem of long-term memory algorithm is the need to store usage frequencies of all iranstormations; their number increases quadratically with the increase of n. Choosing ihe correct value of the weight $\alpha$ is very important, because for a value of $\alpha=0$ the heuristics degrades to a simple downhill local optimization, stopping in a first local optimum found.

The tabu search heuristics shows very good results and is somewhat faster compared !o the Simulated annealing algorithms; its mam drawback is iis sensitivity to parameter settings. For an additional information, ses some of the works mentioned in ihe references ('Glover el al.. 1997; Stluka. 1998)

***Genetic algorithms***

Genetic algorithms are a group of optimization methods usable for a wide variety of problems. Genetic algorithms are based on principles of genetics and on survival of the finest. This technique examines simultaneously a set of possible solutions. The genetic algorithm can be used for solving optimization problems; provided that (a) solutions to the problem can be expressed in the form of a string of characters: *(b)* a ".fitness", which in some way quantifies the quality of the solution, can be defined for any feasible solution; and (c) strings in which "..part" of a good solution is present are rewarded by the allocation of a higher fitness than strings chosen entirely at random.

The string is the basic structure processed by the genetic algorithm. Strings are composed of a sequence of characters of finite length from some alphabet (numeral and binary strings are common). Strings of the current population are then manipulated to generate a new population for the next step. This is done by the use of the adaptive plan. The first set of valid strings is generated randomly: the algorithm stops when the pre-set number of steps is reached.

The adaptive plan, whose purpose is to produce a new, improved population of solution using information provided by the current generation, consists of the fitness proportionate reproduction, strins crossover, and mutation operators. The probability that a particular siring will participate in the reproduction process is set proportional to its fitness, where the fitness is usually a slightly modified objective function value. High fitnes.-, strings will have a higher expected number of offspring. The crossover is the most important operator of a genetic-based technique. Different crossover operators have been proposed for a wide variety of problems. Basically, when two strings are selected for a crossover, some of the characters they contain are semi-randomly exchanged. The crossover method must be chosen so as to guarantee that valid strings will be created. While the crossover exploits already existing knowledge, a new knowledge is introduced into the system by the mutation. Basically, it is the occasional and random alteration of some characters in a string.

The performance of genetic algorithms is often worse than that of the Simulated annealing/Tabu search but still remains sufficiently good: the algorithm is easy to implement and versatile, A further information about genetic algorithms can be found in the works listed in the references (Androulakis, 1991: Cartwright, 1993).

## Computational results

An example of a scheduling of batch flowshop campaign is shown here (Stluka, 1998). Compared to MILP all of (he abovementioned methods offer better performance and sufficiently good solutions. Results shown in Table 2 compare the results obtained using heuristic methods with the results obtained using MILP when solving a model problem (Kim. 1996)

The problem in question is that of optimizing a production of 8 different products in a flowshop consisting of 4 apparatuses. The data considered in this problem are: the processing times, the transfer times, the set-up times, the storage restrictions (Zero Wait policy). The objective criterion used is the completion time. The information in the processing times matrix indicates the length of time it takes to process certain product in certain apparatus. The transfer times matrix contains the information about times ior transferring products between apparatuses. The set-up time matrix contains times for the preparation of piece of equipment for the operation based on the type of production just finished.

All of the heuristic algorithms presented in this work have been written as a Turbo Pascal source code, and

| Method | MINLP | Tabu search | Genetic algorithm | Simulated annealing |
|---|---|---|---|---|
| Hardware | IBM RS/6000 workstation | PC, Intel Pentium 150 MHz processor | | |
| Software | DICOPT++ | compiled Turbo Pascal source code | | |
| Best solution | 195 h | 195 h | 195 h | 195 h |
| Mean percentage deviation | --- | 0 | 0.051 | 0 |
| Computation time | 204 s | 1.4 s | 4.2 s | 2.1 s |

Another example of industrial application of simulations of batch operations for scheduling is the work (Pozivil, 1999). The production of precipitated catalysts is a typical example of a batch plant where tens ol different products (catalysts) are being produced by similar technologies on different apparatuses. The technology includes typical batch and semi-continuous operations such as dissolution, precipitation, filtration and washing, drying, molding, and calcination. The production is planned based on customers' demands so as to satisfy contractual obligations regarding types of catalysts, quantities, and times of delivery. A computer program that was created makes creating production schedule easier and is able to optimize the sequence of produce when such optimization is possible. Another advantage of this, tabu-search based program is that it is able to react to both changes in demands and to unplanned pauses in the manufacture caused by technical problems.

Many other examples of the results shown by stochastic methods can be found in literature, e.g. in some of the works (e.g. Androulakis ct al., 1991; Cardoso, 1997: Stluka, 1998) listed in the references.

## Conclusions

The Production Planning and Scheduling is an important part of mining operations management, especially in cases of multiproduct campaigns. The combinatorial problems in question are the NP-complete ones, and therefore compulation Limes rapidly (exponentially, factonally) increase with the increase in the problem dimension. In cases of realistic problems, the compulation limes exceed several hours when using MILP solvers, and therefore better methods hawe to be found.

Three different optimization methods are described in this paper. The Simulated annealing, the tabu search and the genetic algorithms are presented as alternatives lo MILP for scheduling of batch operations. Compared lo MILP solvers, all these methods display much a better performance while yielding optimum or good near-optimum solutions. Alternative optimization methods described in this paper have been successfully used in a wide variety of industrial applications.

***Acknowledgements:***
*Contribution of P.Stluka to the work presented in this paper is gratefully acknowledged.*

## References

ANDROULAKIS. I. P., VENKATASUBRAMANIAN, V.: A Genetic Algorithmic Framework for Process Design and Optimization. *Computers chem. Engng.,* 15 (1991). 217-228.

BARBOSA-POVOA. A. P.. MACCHIETO. S.: Detailed design of multipurpose batch plants. *Ind. Chem. Engng,*

IS (1994), 1013-1042.
CARDOSO. M. F- SALCEDO, R. L.. FEYO DE AZEVEDO. S.. BARBOSA, D.: A Simulated Annealing Approach lo the Solution of MINLP Problems. *Computers chem. Engng.*. 21 (1997), 1349-1364.
CARTWRIGHT, H. M., LONG, R. A.: Simultaneous Optimization of Chemical Flow shop Sequencing and Topology Using Genetic Algorithms. *Ind. Eng. Chem. Res,..* 32 (1993). 2606-2713. GAREY, M. R..
JOHNSON. D. S.: Computers and Intractability (A Guide to the theory of NT-completeness), u; *H*, *Freeman &*, *Co.,* San Francisco (1979),
GLOVER, F., LACUNA. M.: Tabu search. *Kluwer Academic Publishers,* (1997),
KIM. M., JUNG. J.H., LEE. I.E.: Optimal scheduling of multiproduct batch processes for various intermediate storage policies. *Ind. Chem. Engng, 35* (1996), 4058-4066. KONDILI. E., PANTELIDES, *C.* C., SARGENT, R.W. H.: A general algorithm for short-term scheduling of batch operations - MILP formulation. *Computers chem. Engng.* 17 (1993), 211-227.
KU. H. M.. KARIML I. A.: An Evaluation of Simulated Annealing tor Batch Process Scheduling. *Ind. Eng. Chem. Res.*. 30(1991). 163-169
PAPAGEORGIOU. L. G., PANTELIDES. C. C.: Optimal campaign planning/scheduling ot multipurpose batch/semicontinuous plants. 2. A mathematical approach. *Ind. Eng. Chem.* Res.. 35 (1996), 510-529.
POZIVIL, J.: Rozvrhovani vyroby srazenych katalyzatoru (Scheduling of production of precipitated catalysts). *Proc. of the 12th'. Conif. Process Comfol PC '99, 2. M.* 201.
STLUKA. P.: Využití prvků uměle inceligence při rozvrhování vsádkových vyrob (.Use of artificial intelligence for schedulmg of batch plants). *PhD thesis on VŠCHT Praha (*1998).