FACTA UNIVERSITATIS
Series: **Mechanical Engineering** Vol. 4, N° 1, 2006, pp. 83 - 92

SCHEDULING OF BATCH OPERATIONS – MODEL BASED OPTIMIZATION APPROACH

UDC 519.863:65.012.122

Branislav Stanković, Vukman Bakić

Laboratory for Thermal Engineering and Energy, Institute of Nuclear Sciences Vinca
PO Box 522, 11001 Belgrade
bakicv@vin.bg.ac.yu

Abstract. *The coordinated use of computers throughout the entire spectrum of manufacturing and business operations has been growing during the 1990s and is expected to continue during the 21st century. With the continued increases in computing power and advances in telecommunications, the use of optimization has expanded as well, including planning and scheduling. General access to a common database and enterprise information are provided to managers, engineers and operators so that optimum decisions can be made and executed in a timely and efficient manner. One of the richest aspects of life is our ability to shape our own destiny through the choices that we make. We have to decide, over time, how best to allocate a scarce resource (our time) amongst a number of competing demands. The outcome of each decision that we take is uncertain and affects our situation and the options that will be available to us in the future. Consciously or otherwise, we try to make choices with the aim of achieving certain goals or in order to maximize some measure of "utility" or "pleasure". Similar resource allocation problems are found in a large number of industrial, financial and computing settings. Here the parameters tend to be easier to quantify, in terms of money or time for example. An optimal strategy for allocating the resource is then deemed to be one that maximizes or minimizes some measure of performance. It is problem of this nature that the study of scheduling seeks to address.*

Key Words: *Scheduling, Optimization, Model, Time, Batch Operations*

1. BATCH PROCESS FEATURES

For years, continuous operations have been the most prevalent mode in chemical processing. In recent years, however, there has been a renewed interest in batch processes for a variety of reasons. The most appealing feature of batch processes is their flexibility in producing multiple products in a single plant through sharing of process equipment. The batch operations are economically desirable, especially when small amounts of

Received June 19, 2006

complex, high-value-added chemicals are produced or when a large number of products are made using similar production paths.

The manufacture of all chemical products involves three key elements: a process or recipe which describes the set of chemical and physical steps required to make product, a plant comprising a set of equipment within which these steps are executed and a market which defines the amounts, timing and qualities of the product required. A distinguishing feature of continuous operations is the one-to-one correspondence between the recipe steps and the plant equipment items: the flowsheet is the physical realization of the recipe and its structure remains fixed in time. In batch plants, the structure of the recipe and the plant equipment network structure are in general distinct. Moreover, the equipment configuration may change each time that a different product is made. Thus, in the batch case there exists an additional engineering decision level: the assignment of recipe steps to equipment items over specific intervals of time. These assignments decisions are inherently discrete in nature, introducing a combinatorial aspect to operational and design problems which are not normally present in the continuous process case. Such problems can be modeled mathematically in an optimization framework.

2. THE SCHEDULING PROBLEM

A key problem which arises in batch operations is the scheduling of the plant to meet specified product requirements. Specifically, given the mode of operation, the product orders, the product recipes, the number and capacity of the various types of existing equipment, the list of equipment types allowed for assignment to each task, any limitations on shared resources (such as utilities or manpower) and any operating or safety restrictions, the scheduling problem is to determine the order in which tasks use equipment and resources and the detailed timing of the execution of all tasks so as to optimize plant performance.

The scheduling problem involves three closely linked elements:

- Assignment of units and resources to tasks
- Sequencing of the tasks assigned to specific units
- Determination of the start and stop times for the execution of all tasks.

For instance, given two reactors (U1 and U2) and six product batches (A-F) which need to be processed, the assignment step might involve allocating A, B and C to U1 and D, E and F to U2 [1]. The sequencing step would involve determining the processing order on each unit (e.g. first B, then C, and then A on U1), while the timing step would assign specific start and stop times for each batch on each unit. The above problem elements are shared by scheduling problems arising in a wide range of applications – ranging from machine shops to transportation systems to class room allocation. The assignment component of the problem involves binary decisions (assign U1 to task A or not) as does the sequencing component (position A first in the sequence or not). The timing component can be a discrete decision problem or not depending upon whether time is treated as a continuum or divided into individual time quanta. It is the binary decisions which provide the challenge to scheduling problem solution. Indeed, the theoretical worst case analysis of computational complexity has shown that even the conceptually simplest forms of scheduling problems (those involving only sequencing considerations such as the sequencing of products on a single process with set-up costs which are dependent on the product processing order) can exhibit exponential growth in computational effort with increasing problem size.

The staged nature of a processing network, consisting of a number of units in series, allows four different storage operations:

- Unlimited (infinite number) intermediate storage (UIS)
- Finite (specified number) intermediate storage (FIS)
- No intermediate storage (NIS)
- Zero wait or no wait (ZW or NW)

The interstage storage capacity is measured in terms of the number of units, not the physical size of storage, since it is usually assumed that each storage unit can temporarily hold any product batch [2]. In both the NIS and ZW modes, there is no storage between stages. While a batch, after its completion in a processing unit, may be held in it temporarily in the NIS, UIS or FIS modes, it must be transferred to the downstreams unit immediately in the ZW mode. In the situations where unstable intermediates are produced and must be immediately processed in succeeding steps, the ZW mode of operation is used.

In a batch/semicontinuous plant, the general short-term scheduling problem is characterized by:

- A set of N products or product batches to be produced
- A set of M available processing units
- A sequence for each product in which operations are to be performed
- A set of fixed processing times for each product from each equipment item
- A matrix of fixed transfer times for each product from each equipment item
- A matrix of fixed (possibly sequence-dependent) setup times or costs between every pair of products in each equipment item
- Constraints on the production order for some products (precedence constraints)
- A suitable performance or cost criterion to be optimized
- The nature of the intermediate storage between processing stages
- The structure of processing network

The solution of the scheduling problem is critically affected by the performance criterion, the intermediate storage and the structure of the network. In terms of performance criteria, different objective functions can be used. Minimum total time required to produce all products or makespan is one of the most studied objective functions.

To sum up, a single, universal solution approach to all scheduling problems does not exist and it is highly unlikely that one will ever be found.

3. SOLUTION ALGORITHMS – MODEL-BASED OPTIMIZATION

The categories of solution algorithms which have been advanced for the solution of scheduling problems include: rule-based dispatching methods, randomized search methods, artificial intelligence related methods, simulation approaches and model-based optimization methods.

Model-based optimization employs a mathematical model of the application as the basis for conducting a systematic search of the solution domain using numerical and logical methods, such as linear programming. The advantage of a model is that it offers a rigorous measure of the quality. However, model formulation may require considerable expertise and the optimization process can be quite computationally intensive.

Short-term scheduling (as the model-based optimization technique) involves sequencing and scheduling the production of N products across M processing units to optimize a suitable performance criterion [3]. This problem can be looked upon as a combination of two interlinked subproblems: sequencing and timetable. Sequencing involves determining the order in which the products are to be processed to obtain the best possible schedule. The timetable of production for a given sequence involves determining the starting and finishing times of each product on all processing units.

An interesting feature of most scheduling problems is that, while they are deceptively simple, they are very hard to solve. Determination of the production sequence and the timetable is a combinatorial optimization problem. The number of candidate solutions grows exponentially as the size of the problem increases. For example, in a flowshop, the number of candidate permutation schedules is N! where N is the number of products.

A detailed schedule for a given production sequence is carried out by recurrence relations. Recurrence relations are a set of expressions that can be used recursively to generate the start and finish times of each product in all processing and storage units. For a simplified UIS flowshop, the relations are quite simple:

$$\begin{aligned} &C_{ij} = Max\,[\, C_{(i-1)j} \,,\, C_{i(j-1)} \,] + t_{k_i j} \\ &C_{i0} = 0, \quad i = 1, N \\ &C_{0j} = 0, \quad j = 1, M \end{aligned} \tag{1}$$

where C_{ij} is the completion time of the i-th job (task) in the sequence on the j-th processor (unit) and $t_{k_i j}$ is the processing time of product k_i on unit j.

Here, the completion time refers to the time at which a product finishes processing on a unit. The schedule obtained from the application of the equations (1) to the sequence 1-2-3-4 in the problem of Table 1 is shown in Fig. 1.

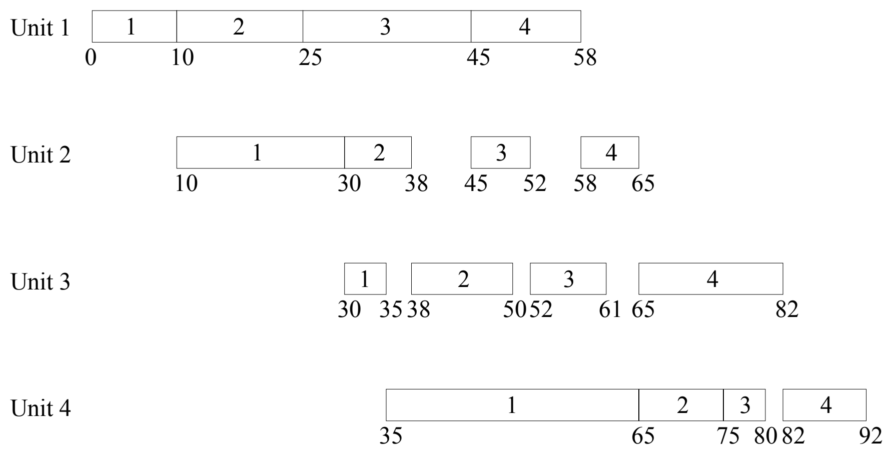

Fig. 1. Unlimited intermediate storage schedule

The recurrence relations indicate that the completion time of a job on a unit is its processing time plus the time at which processing can start. This implicitly assumes that the transfer time of the job from one unit to another is negligible. Applying these equations recursively, the completion times for the entire sequence of jobs on all processing units can be calculated with an amount of computational effort proprotional to MxN. As the complexity of the flowshop increases, the recurrence relations become more complex to formulate.

Table 1. Processing Times

Products	Units			
	1	2	3	4
1	10	20	5	30
2	15	8	12	10
3	20	7	9	5
4	13	7	17	10

3.1. MILP Approach

An MILP (mixed integer linear programming) is an optimization problem with linear objective function and constraints [4]. It differs from an LP (linear programming) in that some of its variables are integer or binary (0-1). In general, the formulation for a simple flowshop problem can be easily inferred from its basic recurrence relations. Binary variables are defined as: $X_{ij} = 1$ if product i is in position j in the sequence, or $X_{ij} = 0$ otherwise. Makespan is used as the performance criterion. Using C_{ij}'s as the completion times, the objective function is: Minimize C_{NM}.

The first set of constraints ensures that a product is assigned to only one position in the processing sequence:

$$\sum_{i=1}^{N} X_{ij} = 1 \qquad i = 1, 2, 3, 4 \tag{2}$$

The second set of constraints ensures that a product is assigned to only one processing unit:

$$\sum_{j=1}^{M} X_{ij} = 1 \qquad j = 1, 2, 3, 4 \tag{3}$$

The remaining constraints, which come directly from the UIS recurrence relations, ensure that the completion times of a product sequence are correctly calculated.

Since the Max functions in the recurrence relations are discontinuous, we must replace them by multiple, continuous relations. For unit 1 (j = 1):

$$C_{i1} - C_{(i-1)1} - \sum_{k=1}^{N} t_{k1} X_{ki} \geq 0 \qquad i = 1, 2, 3, 4 \tag{4}$$

For units 2 to 4 (j = 2, 3, 4):

$$C_{ij} - C_{(i-1)j} - \sum_{k=1}^{N} t_{kj} X_{ki} \geq 0 \qquad i = 1, 2, 3, 4 \tag{5}$$

$$C_{ij} - C_{i(j-1)} - \sum_{k=1}^{N} t_{kj} X_{ki} \geq 0 \qquad j = 1, 2, 3, 4 \tag{6}$$

Note that since only one of X_{ki}'s for each i will be one, the term $\sum_{k=1}^{N} t_{kj} X_{ki}$ in the above constraints will pick up the processing time of an appropriate product.

3.2. Multiproduct Batch Process with Finite Interstage Storage (FIS)

Problem description. An M-unit serial multiproduct batch plant consists of M batch units in series. As shown in Fig. 2, z_j ($z_j \geq 0$, $j = 1, M-1$) storage units are present between batch units j and (j+1).

Determination of completion times. We begin by presenting recurrence relations for computing completion times and the makespan of a given product sequence. A product sequence is characterized by a permutation of integers: $k_1 - k_2 - k_3 - ... - k_N$, where product k_i is in the i – th position in the sequence. Let C_{ij} denote the time at which the i – th product in the sequence leaves unit j. Note that C_{ij} is not necessarily equal to the time at which product k_i finishes processing on unit j. For example, when the downstream unit is busy and the storage is full, product k_i must be held temporarily in unit j and C_{ij} would then be the time at which product k_i actually leaves unit j.

To compute C_{ij}, consider a scenario in which product k_i has just finished processing on unit j. The transfer of product k_i out of unit j is dictated by two events. One is the situation in which either the downstream unit is free or a storage unit is free between units j and (j+1). In either case, product k_i can leave unit j immediately upon completion. The other situation is the one in which the downstream unit is busy and no storage unit is available.

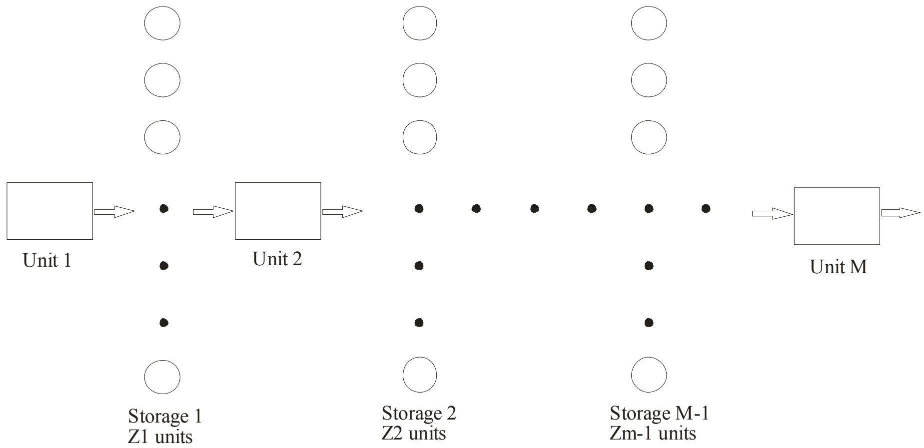

Fig. 2. FIS system configuration

In the first case, the completion time of product k_i on unit j is simply the time at which unit j starts processing product k_i plus its processing time, $t_{k_i j}$. But unit j cannot start processing product k_i until it has processed the previous product, namely product $k_{(i-1)}$, or until product k_i has been processed by the upstream unit, namely, unit (j–1).

So we have:

$$C_{ij} = Max\,[C_{(i-1)j}, C_{i(j-1)}] + t_{k_i j} \tag{7}$$

In the second case, product k_i must be held temporarily in unit j until a storage unit becomes available. Since no storage unit is currently available and the downstream unit is busy, products $k_{(i-1)}$, $k_{(i-2)}$, ... ,$k_{(i-z_j)}$, must be occupying the storage facility and product $k_{(i-z_j-1)}$ must be in unit (j + 1).

The time at which a storage unit becomes available is the time at which product $k_{(i-z_j)}$ is discharged to the downstream unit, namely, unit $(j+1)$. But this discharge is possible only when the downstream unit becomes available, i.e., product $k_{(i-z_j-1)}$ leaves unit $(j+1)$. Therefore:

$$C_{ij} = C_{(i-z_j-1)(j+1)} \tag{8}$$

From equations (7) and (8), we obtain the following proposition.

Proposition 1. For an M-unit serial FIS system, completion times C_{ij} are given by:

$$\begin{aligned} &C_{ij} = Max\,[C_{(i-1)j}, C_{i(j-1)}, C_{(i-z_j-1)(j+1)} - t_{k_i j}] + t_{k_i j} \\ &i = 1, 2, \dots, N \qquad j = 1, 2, \dots, M \end{aligned} \tag{9}$$

where $C_{ij} = 0$, if $i \le 0$ or $j \le 0$ or $j > M$.

Note that the above recurrence relations are also applicable to the UIS and NIS policies, since they are in fact special cases of the FIS policy. In the case of UIS, $z_j \ge N-1$; thus, the term $C_{(i-z_j-1)(j+1)} - t_{k_i j}$ in the max function is negative and can be omitted altogether. In the case of NIS, although the recurrence relations can easily be obtained by setting z_j equal to zero, a closer examination indicates that one of the three terms in the max function of equation (9) is redundant. Setting $z_j = 0$ in equation (9) yields:

$$C_{ij} = Max\,[C_{(i-1)j}, C_{i(j-1)}, C_{(i-1)(j+1)} - t_{k_i j}] + t_{k_i j} \tag{10}$$

Substituting $j = j-1$ in the above equation gives:

$$C_{i(j-1)} = Max\,[C_{(i-1)(j-1)}, C_{i(j-2)}, C_{(i-1)j} - t_{k_i (j-1)}] + t_{k_i (j-1)} \tag{11}$$

But this implies that $C_{i(j-1)} \ge C_{(i-1)j}$. Consequently, the term $C_{(i-1)j}$ in equation (10) is redundant and can be omitted. Thus, for the NIS policy:

$$C_{ij} = Max\,[C_{i(j-1)}, C_{(i-1)(j+1)} - t_{k_i j}] + t_{k_i j} \tag{12}$$

In terms of complexity, the recurrence relations are easy to code. The fact that the completion times can be easily and cheaply computed is critical, because most good heuristics require a large number of makespan calculations. These recurrence relations form the basis behind the optimal MILP formulation of the scheduling problem.

Optimal MILP formulation. Bearing in mind the first and second set of constraints mentioned above, the remaining constraints come directly from Proposition 1 and they ensure that completion times of a product sequence are correctly calculated. Since the max functions in the recurrence relations are discontinuous, we replace them by multiple, continuous relations as follows:

For unit 1 $(j = 1)$

$$\begin{aligned} &C_{i1} - C_{(i-1)1} - \sum_{k=1}^{N} t_{k1} X_{ki} \ge 0 \\ &C_{i1} - C_{(i-z_j-1)2} \ge 0 \end{aligned} \tag{13}$$

For units 2 to $M-1$ $(j = 2, M-1)$:

$$C_{ij} - C_{(i-1)j} - \sum_{k=1}^{N} t_{kj} X_{ki} \geq 0$$
$$C_{ij} - C_{i(j-1)} - \sum_{k=1}^{N} t_{kj} X_{ki} \geq 0 \tag{14}$$
$$C_{ij} - C_{(i-z_j-1)(j+1)} \geq 0$$

For unit M:

$$C_{iM} - C_{(i-1)M} - \sum_{k=1}^{N} t_{kM} X_{ki} \geq 0$$
$$C_{iM} - C_{i(M-1)} - \sum_{k=1}^{N} t_{kM} X_{ki} \geq 0 \tag{15}$$

where $i = 1, 2, 3, \dots, N$.

Note that since only one of the X_{ki}'s for each i will be one, the term $\sum_{k=1}^{N} t_{kj} X_{ki}$ in the above constraints will pick up the processing time of an appropriate product. Note also that for any value of i, if one or no term in a *Max* function of a recurrence relation is nonzero (recall that $C_{ij} = 0$ for $i \leq 0$ or $j \leq 0$), then we have an equality constraint instead of an inequality constraint for that recurrence relation and the zero terms do not constitute any constraints.

As an example, consider the processing times in Table 2. By use of the MILP formulation on appropriate optimization package, the products are scheduled in UIS, FIS and NIS systems. In the case of the FIS system, a storage configuration of $z_1 = 0$, $z_2 = 0$ and $z_3 = 1$ is assumed. Optimal sequences of 5-1-2-6-4-3 (makespan 107), 5-1-4-6-2-3 (makespan 107), 5-1-4-6-2-3 (makespan 107) and 5-6-1-4-2-3 (makespan 111) are obtained respectively for the UIS, FIS and NIS systems and the corresponding Gantt chart of UIS system is shown in Fig.3.

Note that the sequences obtained for the three systems are different, which shows that the storage has a significant impact on the optimal solutions obtained.

Optimal formulation with ZW blocks. While storage plays an important role in batch processing by reducing idle times of processing units, some product batches cannot be stored at all between some stages in the processing sequence. These batches are usually unstable intermediates which must be processed by the next processing unit immediately upon completion on the current processing unit. For such processing stages, the ZW policy is most appropriate.

Table 2. Processing times

Products	Units			
	1	2	3	4
1	10	20	5	30
2	15	8	12	10
3	20	7	9	5
4	14	6	15	10
5	6	11	5	15
6	13	7	17	10

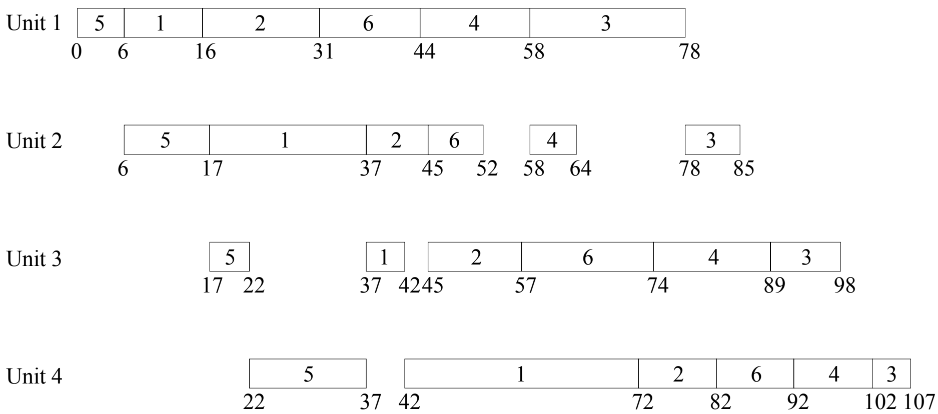

Fig. 3. UIS schedule of the example of Table 2

Although the ZW policy is not incorporated into the recurrence relations presented earlier, the optimal MILP formulation can be easily extended to include ZW blocks embedded in the multiproduct process [5]. First, we redefine completion time C_{ij} to be the time at which the i-th product finishes processing on unit j, rather than the time at which it leaves the unit. We will denote this new completion time by C'_{ij}.

Consider a scenario in which the (i-1)-th product, rather than i-th product as before, has just finished processing on unit j; we wish to determine the earliest time at which the i-th product can start processing on unit j. Once this start time is determined, C_{ij} is simply the start time plus $t_{k,j}$, by our new definition of the completion time. The start time of the i-th product on unit j is determined by either the time at which it finishes processing on the previous unit, namely $C'_{i(j-1)}$, or the time at which the (i-1)-th product leaves unit j.

The transfer of product $k_{(i-1)}$ out of unit j is in turn governed by two events. In the first event, a storage unit or the downstream unit is available to receive product $k_{(i-1)}$, in which case the product can leave unit j immediately upon completion. This time is simply given by $C'_{i(j-1)}$.

In the second event, product $k_{(i-1)}$ must wait in unit j, since no free storage is available and the downstream unit is busy. Clearly, products $k_{(i-2)}$, $k_{(i-3)}$, ..., $k_{(i-1-z_j)}$ must be occupying the storage and product $k_{(i-2-z_j)}$ must be in unit (j + 1). The time at which a storage unit becomes free is the time at which product ($i - 1 - z_j$) is discharged from storage and starts processing on unit (j + 1). This start time of product $k_{(i-1-z_j)}$ on unit (j + 1) is simply $C_{(i-1-z_j)} - t_{k_{(i-1-z_j)}(j+1)}$. From the preceding discussion, we get the following proposition.

Proposition 2. The completion times C'_{ij} for a serial FIS flowshop are :

$$C'_{ij} = Max\,[C'_{i(j-1)}, C'_{(i-1)j}, C'_{(i-1-z_j)(j+1)} - t_{k_{(i-1-z_j)}(j+1)}] + t_{k_i j} \tag{16}$$

where $C'_{ij} = 0$ if $i \le 0$ or $j \le 0$ or $j > M$.

Again, the third and the second term in the Max function of equation (16) can be omitted for the UIS and the NIS policy, respectively.

Note that C_{ij} is always equal to C'_{ij} unless product k_i must be held temporarily in unit j upon its completion. Since the last product in the preceding sequence never has to wait in the last unit due to the assumption of unlimited storage for finished products, makespans computed from both sets of recurrence relations are always the same, ie, $C_{NM} = C'_{NM}$. Hence, the two sets of recurrence relations are equivalent as far as makespan computation is concerned.

4. CONCLUSION

Production scheduling is of immense importance in noncontinuous processes of the chemical process industry. Application of the scheduling methodology can significantly improve the productivity and cost-effectiveness of batch processes. The determination of completion times for a given product sequence in serial multiproduct noncontinuous plants is examined thoroughly. The categories of solution algorithms which have been advanced for the solution of scheduling problems include: rule-based (heuristics) dispatching methods, randomized search methods, artificial intelligence related methods, simulation approaches and model-based optimization methods. The advantage of a model-based optimization approach is that it offers a rigorous measure of the quality and the feasibility of any solution that is obtained.

REFERENCES

1. Henning, G.P., Cerda, J., 1996, *A knowledge-based approach to production scheduling for batch processes*, Computers and Chemical Engineering, 20, pp. 1295-1300.
2. Shah, N., Pantelides, C.C., Sargent, R.W.H., 1993, *A general algorithm for short-term scheduling of batch operations – II. Computational issues*, Computers and Chemical Engineering, 17, pp. 229-244.
3. Birewar, D.B., Grossmann, I.E., 1989, *Simultaneous production planning and scheduling in multiproduct batch plants*, Industrial Engineering and Chemical Research, 29, pp. 570-580.
4. Sahinidis, N.V., Grossmann, I.E., 1991, *Reformulation of multiperiod MILP models for planning and scheduling of chemical processes*, Computers and Chemical Engineering, 15, pp. 255-272.
5. Wiede, W.Jr., Reklaitis, G.V., 1987, *Determination of completion times for serial multiproduct processes – 2. Multiunit finite intermediate storage system*, Computers and Chemical Engineering, 11, pp. 345-356.

VREMENSKI RASPORED ŠARŽNIH OPERACIJA – PRILAZ ZASNOVAN NA MATEMATIČKOM MODELU I OPTIMIZACIJI

Branislav Stanković, Vukman Bakić

Koordinisana upotreba kompjutera kroz celokupni spektrum proizvodnih i poslovnih operacija rasla je tokom devedesetih godina i očekuje se da će se taj trend nastaviti tokom dvadeset prvog veka. Sa neprekidnim povećanjem mogućnosti kompjutera i dostignućima u oblasti telekomunikacija, upotreba optimizacije se širila, uključujući planiranje i određivanje vremenskog rasporeda. Opšti pristup zajedničkim bazama podataka i informacijama dostupan je menadžerima, inženjerima i operaterima, tako da se optimalne odluke mogu doneti i izvršiti na pravovremen i efikasan način. Jedan od najznačajnijih aspekata života je naša sposobnost da usmeravamo našu sudbinu kroz izbore koje pravimo. Mi moramo da odlučimo tokom vremena kako najbolje da raspodelimo oskudno sredstvo (naše vreme) između puno konkurentnih zahteva. Ishod svake odluke koju donesemo je neizvestan i utiče na naš položaj i mogućnosti koje će nam biti dostupne u budućnosti. Svesno ili drugačije, mi pokušavamo da izaberemo sa namerom postizanja izvesnih ciljeva ili da bismo maksimizirali neku meru "korisnog" ili "lepog". Slične probleme raspodele sredstava nalazimo u velikom broju u industrijskom, finansijskom i kompjuterskom okruženju. U ovim slučajevima parametri se lakše mogu predstaviti brojem, u vidu novca ili vremena na primer. Optimalnom strategijom raspodele sredstava smatra se ona koja maksimizira ili minimizira neku meru učinka. Problemom ove prirode bavi se proučavanje vremenskog rasporeda.

Ključne reči: *vremenski raspored, optimizacija, matematički model, vreme, šaržne operacije.*