Prof. Bożena Skołud
M.Sc. Aleksandra Zientek
Department of Engineering Processes Automation and Integrated Manufacturing Systems, Silesian University of Technology, Gliwice, Poland

AN APPROACH TO TIME MANAGEMENT

The objective of this paper is to answer the question: How to exploit resources to complete all projects in expected time in a multi-project environment? A solution to maximise the number of projects, which the company is able to implement concurrently is proposed by combining the Theory of Constraints and conditions guaranteeing project due dates with constraint-based scheduling.

ZARZĄDZANIE CZASEM ZORIENTOWANE NA PROJEKT

Zagadnieniem poruszanym w artykule jest udzielenie odpowiedzi na pytanie: jak wykorzystać zasoby danego przedsiębiorstwa realizującego wiele projektów tak, aby realizacja wszystkich projektów była możliwa w założonym czasie. Maksymalizacja liczby projektów, które dana organizacja może wykonać współbieżnie jest zaproponowana przez połączenie filozofii Teorii Ograniczeń, warunków zapewniających ukończenie projektu na czas oraz harmonogramowania w oparciu o ograniczenia.

1. INTRODUCTION

Complexity of manufacturing processes causes that new approaches and concepts are needed. One of them called *innovative production* [17] is based on the integration of close engineers' collaboration. In the production management context (in which one of the participants is customer) the innovative production means that production planning based on make-to-order and the fact that both product lead time and product price are subjects of negotiation with the customer [6]. Moreover products quality and data of delivery price is not a subject of negotiation. Consequently, the allocation of activities to resources and theirs capacities in order to keep the time regime is one of the most difficult tasks. In other words, the main task is to schedule activities in a given system determined by the resources availability in time. Manufacturing shift in relation between customer and producer has influenced [2]:

- drastic reduction of time to market,
- quality and *life cycle* performance prevail,
- from *design of the customer* (60s) to *design by the customer* (2000),
- mass customisation.

These changes need forming an appropriate tool supporting decision-making in the course of production flow planning. Moreover, the considered computational complexity of such tasks requires more effective approaches.

Time and capacity management may refer to the acceptance of single or small production order in the multi-project conditions. In this case decision depends on constraints of the customer (expected realisation time, delivery batch size) and producer (available machine capacity, buffers space, etc).

Generally speaking, knowing the goal of the system and its constraints, the following steps of TOC should be executed:

1. Identification of the system constraint (the weakest link in the chain). Constraint is anything limiting the performance of the system, slowing or preventing it from continuing to move toward its goal.
2. Exploit the constraint.
3. Subordinate every aspects of the system to this decision.
4. Increase the capacity of the constraint.
5. Identify a new constraint which appears during increasing in the capacity of already identified and exploited constraints.

In the context of the project management one of the most important constraint is the project duration. Any delay in project execution causes that the return of investment is retarded.

The most important decision is connected with answering following questions:

- When each activity should start? (time placement decisions),
- On which resource(s) each activity should be executed? (resource allocation decisions).

A variety of constraints (like: an activity duration, release and due dates, precedence constraints, transfer and set-up times, resource availabilities, and resource sharing) may influence scheduling. In addition, *relaxable preference* constraints characterise the quality of scheduling decisions. These preferences are related to due dates, productivity, frequency of tool changes, inventories level, overtime, etc. However from the Theory of Constraints (TOC) point of view, only one or two constraints influence both the system behaviour and success of the project execution. These conditions are: due time and resource utilisation.

2. PROJECT MANAGEMENT

The inability to deal with real life project scheduling problems such as late completion, over spending, cutting specifications of a project calls for permanent analysis and the Theory of Constraints application. Critical Chain (CC) is proposed by TOC for single project scheduling and formed by extending Critical Path (CP) of activities using scarce resources. The aim is to focus on critical areas, by identifying CC, and to insert time buffers in the appropriate places (moments) in the project network.

For a new project it is more practical to make use of existing resources rather than hiring or subcontracting others. In this situation there are big advantages of considering resource dependencies between projects. Sometimes resource are dedicated to specified projects, but usually occurs that projects overlap (when one resource is required for several projects in the same time). By planning projects together, decisions might affect time estimations of other projects. Thanks to that events are better coordinated and less protection from buffers is needed. The more globally scheduling is done, the smaller (in total) buffers need to be, the shorter lead times, and the more competitive the company is.

If several activities of a project have to be executed on the frequently limited resources, it is necessary to take into account dependencies that might arise between activities that require the utilisation of the same resource(s). If that is the case, these activities must be carried out sequentially, rather than in parallel. In the multi-project situation the CC scheduling should be extended of drum buffer scheduling, which is the constraint of the multi-project environment and which limits a greater number of projects. The constraint resource becomes the drum for multi-projects scheduling and it establishes the rhythm for all projects. The drum schedule determines a sequence of all projects utilising the drum resource. If the drum completes a project early subsequent projects are also executed earlier otherwise they are delayed. For that reason, projects in a multi-project environment require time buffers to protect the drum resource(s) and to ensure that project(s) running on it never starve the constraint.

3. DIFFERENCE BETWEEN SINGLE AND MULTI-PROJECT SCHEDULING

The aim of TOC approach is to relocate safety time buffers to strategic positions. The application of TOC assumes that almost all activities (time estimates) in a project can be reduced by up to 50%, but the safety time buffer called Project Buffer (PB) has to be added at the end of the whole project [14].

The decision about buffers time allocation is caused by multitask approach, and the student syndrome, which means that each task starts as late as it is possible. One should remember that for project manager it is not important the duration of each single project stage but the total project duration.

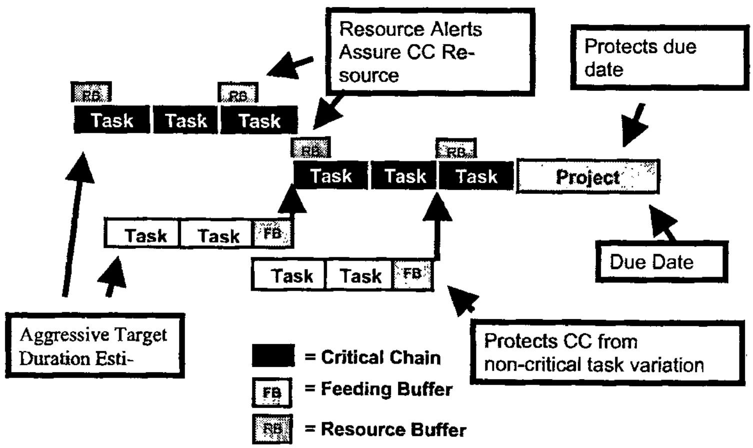

Fig.1. Time buffers: Project Buffer (PB) and Feeding Buffers (FBs) and Resource Buffers.

A great number of companies deal with numerous, interdependent projects running at the same time. In that situation the strategic resource used by a great number of projects can determine the overall throughput of the company. Planning of a multi-projects environment requires pointing out some considerations [10]:

- The market is a leverage point. The biggest inter-project impact is on the capacity of the strategic resource. Projects must be subordinated to market requirements, and therefore CC scheduling in order to obtain earlier and more reliable completion times is required.
- A resource may be considered a company-wide strategic leverage point. If there is such a resource, attention has to be focussed on it, because it contain the overall throughput of the company.
- A non-strategic resource can become a constraint. Multi projects often need to be planned together. Both strategic resources and CC are important. There are many reasons on account of which scheduling more than one project at a time can be difficult.

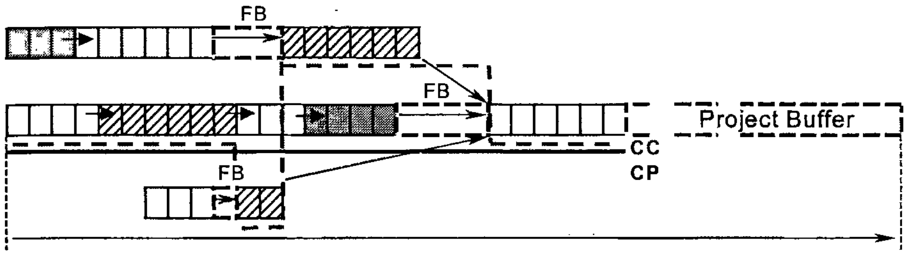

Fig. 2 A diagram showing the difference between CP and CC.

In Figure 2 duration of activities was reduced by 50%. Different resources are illustrated by different texture or shade. FBs are added at the end of non-critical sub-paths.

4. APPROACHES TO THE NEW PROJECT ALLOCATION

Activities of a new project should be allocated to a company resource constraint across all running projects prioritised before creating the drum schedule. The drum schedule determines the system capacity. Individual CC schedules determine the duration, earliest time, and relative times for each project executed on the drum. The new project priority should be determined before introducing a new project into the current running system. If a company prefers first-in, first-out priority rule the new project calling for execution gains the lowest priority. If the new project is important for a company it may gain a higher priority than other ongoing projects. The CC of the new project must be prepared to determine when the drum resource(s) is necessary for its activities. Next new project activities have to be scheduled together with ongoing projects in the proper sequence. On the basis of the new drum schedule the start time of the new project is determined by backward scheduling. The rest of the project is scheduled forward.

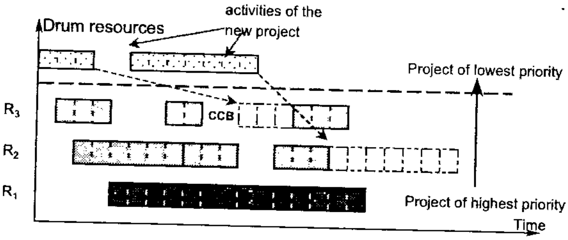

Fig.3. A drum schedule accommodates all four projects.

Let's assume that three resources (R_1, R_2, R_3) belong to the drum (see Figure 3). It is established that the capacity of the drum can not be exceeded. Three ongoing projects are running on three available resources. Activities of the new project are waiting for an idle resource in the drum schedule. The method is to schedule projects activities of the lower priority later in time. Critical Chain Buffer (CCB) ensures that the constraint resources belonging to the drum are available for the new project when they are needed. CCB is placed between the use of the constraint resource in the prior ongoing project and the first its use in the first activity of the next project. Moreover Drum Buffer (DB) ensuring that the drum resource(s) has projects to work on when they are required.

If the new project is of a higher priority than any of the running projects, the schedule of already accepted projects of lower priority might be changed. The difference between this example and the one presented before is that the new, 5-th project is of a higher priority than the ongoing one which was scheduled before (dashed contour in Fig. 4a). The project of a lower priority should be put above the new one (dashed arrows in Fig 4a). Then once again both projects should be scheduled in the proper sequence in the drum schedule starting from the project of a higher priority rule.

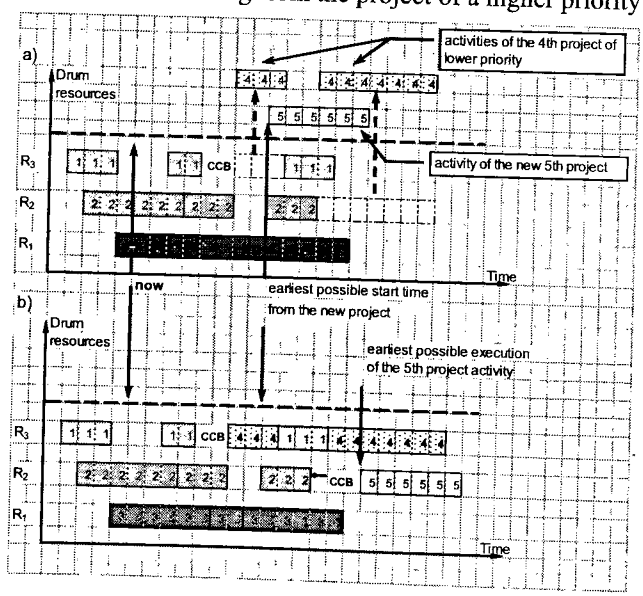

Fig. 4. Introducing a new project of higher priority

The new project influencing the schedule of the drum resource and causes delay of the second activity of 4-th project is shown in Figure 4b.

5. CONSTRAINTS PROGRAMMING

Computing a schedule that respects the constraints and objectives of a given scheduling problem is *combinatorial*: many alternatives need to be explored; many decisions need to be made and undone before a feasible schedule can be found. In fact, in most cases scheduling problems belong to the NP-complete class. It means that polynomially bound algorithm for solving them optimally does not exist yet. As a result, scheduling problems are often solved by means of *heuristics*: solution procedures that focus on finding a feasible schedule of "good" (as opposed to optimal) quality within an acceptable amount of time.

Declarative character of constraint programming (CP) supports the natural modelling of real-life combinatorial problems via specialised constraints. In the traditional operation research approach *constraints* binding all the variables and solving methods are used to global reasoning about the problem. In the constraint programming the problem is modelled by using a set of constrains binding small sets of variables, which are much smaller than the set of all variables in the problem. The aim of constraint propagation is to achieve some level of consistency in the network of constraints and variables by removing inconsistent values form variables domain. Constraint-based scheduling is an approach for solving scheduling problems using constraints satisfaction technique and can be modelled as constraint satisfaction problem using a set of variables, their domains, and constraints describing feasible combination of variables values.

5.1. ALGORITHM

Decision-making, based on scheduling problems, belong to the area of combinatorial optimisation problems and can be naturally described as constraint satisfaction problems. To model the problem as CSP the problem objects should be mapped into variables and constraints. One of the traditional modelling approaches uses three variables identifying the position of the activity in time: start time, end time, and processing time (duration). Let *a* be an activity, we denote these variables *start(a)*, *end(a)*, and *p(a)* [1]. We expect the domains for these variables to be discrete (e.g., natural numbers) where the release time and the deadline of the activity make natural bounds for them (and time windows make the domains even more restricted).

In paper [15] two conditions assuring possibility of a new project execution using available resources was presented. Fulfilment of the first condition guarantees that there is a time window on the r-th resource when the r-th resource is idle. The second formula says that a time interval is at least equal duration of the new project on that resource. The main purpose was to check the possibility of a new project implementation without comparing the chart representing system resource occupancy and the new project chart. Each resource belonging to the drum should be only analysed in the range of the time window of the new activity achieved from its CC. These time windows ($V_{r,t}^{a}$) define boundaries of domains on each resource belonging to the drum.

$V_{r,t}^{a}$ – time window on the r-th resource for the a-th activity at the j_i moment,
$V_{r,t}^{a} = [s^a, s^a+1 \ldots, b^a]$

start(a) – s^a – forward scheduling start time for the a-th activity of the new project,
end(a) – b^a – backward scheduling end time for the a-th activity of the new project,
p(a) – processing activity a + its CCB

$$j_i = \begin{cases} 0 \text{ - drum resource not occupied} \\ 1 \text{ - drum resource occupied by already running process(es)} \end{cases}$$

where i = start(a), start(a)+1,...,end(a)

1st condition that should be fulfilled: $j_i, j_{i+1}, \ldots, j_{p(a)} \neq 1$

2nd condition: $\overset{m+p(a)}{\underset{i=m}{\forall}} j_i = 0$

5.2. ILLUSTRATIVE EXAMPLE

Let us consider an organisation of 2 resources available in the drum schedule (M_1, M_2). Some projects are already running on the drum. The activity of a new project (placed above the dashed horizontal line in Figure 6) is waiting for its acceptance on these resources. Dashed vertical lines represent the time window for the new activity. Under the line occupancy of projects already running in the system is shown.

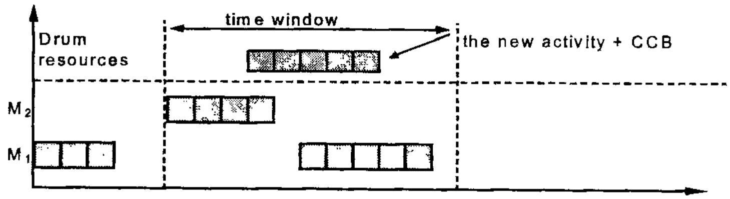

Fig.5. Illustrative example

start(a)= 6 (j_6) end(a)=16 (j_{16}) p(a)=5

$dM_1 = [j_6, j_7, j_8, \ldots, j_{16}]$ (V_1^a)
$dM_2 = [j_6, j_7, j_8, \ldots, j_{16}]$ (V_2^a)
where dM_1, dM_2 - domains of M_1 and M_2;

Time units representing occupancy of resources belonging to the drum are marked by 0 or 1, where 0 means that the r-th resource is idle; 1 that it is occupied. That is why of resources domains values in our case will be as follow:
$dM_1 = [0,0,0,0,0,1,1,1,1,1,0]$ (V_1^a)
$dM_2 = [1,1,1,1,0,0,0,0,0,0,0]$ (V_2^a)

1st condition: $j_6, j_7, j_8, \ldots, j_{16} \neq 1$
That condition constraints domains of M_1 and M_2 to:
$dM_1' = [j_6, j_7, j_8, j_9, j_{10}, j_{16}]$ (V_1^a)

$dM_2' = [j_{10}, j_{11}, j_{12}, j_{13}, j_{14}, j_{15}, j_{16}]$ (V_2^a)

Fulfillment of 1[st] and 2[nd] condition is presented in the *tree* form in which to find the possible solution of the problem the backtracking method based on the systematically inquisition of the possible solutions was applied. Through the procedure, sets of possible solutions are rejected before even examined so their number is getting smaller. For this reason the examination and produce of the solutions follows a model of non-cycle graph considered as a *tree*. The root of the tree represents the set of all the solutions. Nodes in lower levels represent smaller sets of isolated solutions. The *tree* is created during the examination of the solutions so that no rejected solutions are created. When a node is rejected (no), the whole sub-tree is rejected, and we backtrack to the ancestor of the node. Conceptually, a backtracking algorithm does a depth-first search of a tree of possible (partial) solutions. In our example faster new project execution was possible on the M_1 resource.

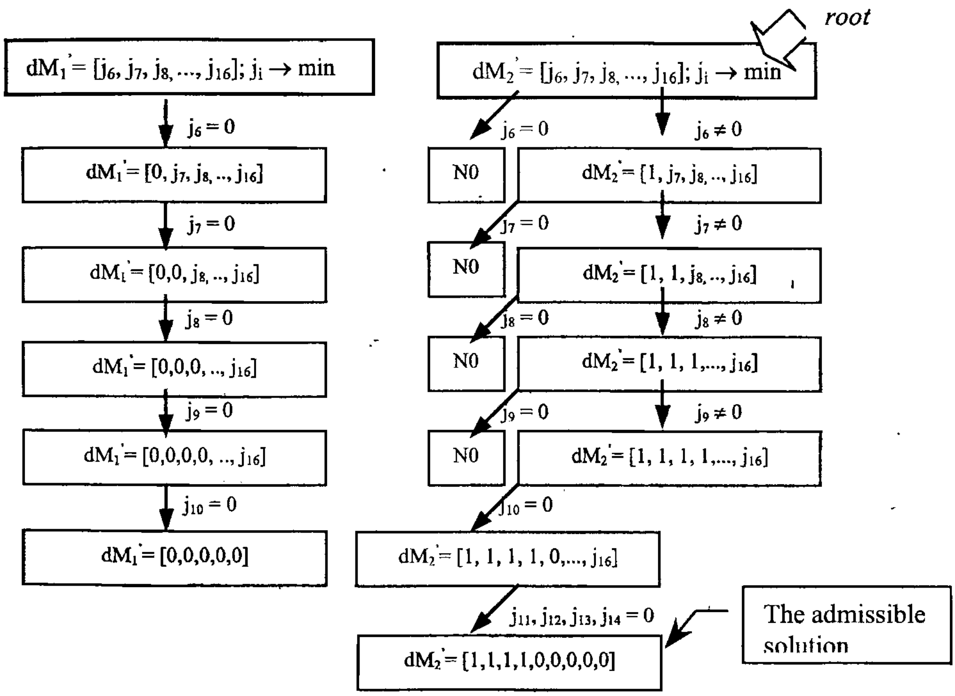

Fig.6. Examination of the solutions in the tree form

The solution was first achieved on M_1 resource: j_i of $M_1 < j_i$ of $M_2 \Rightarrow j_i$ of M_1

CONCLUSION

The existence of many disparities in the requirements of scheduling problems implies that a general-purpose software package efficiently solving all scheduling problems does not exist. Moreover, currently available scheduling software packages tend to be rigidly designed and implemented for a particular type of scheduling problem, if not for a particular instance of that problem, and cannot be easily used for others: they are based on a pre-configured and inflexible scheduling model that does not extend gracefully to other scheduling situations. That is why attempts of constraint-based scheduling for solving scheduling problems are taken. Its major advantages of that kind of scheduling over the existing approaches are: clarity and generality of the models. Moreover, it provides generic solution techniques of constraint satisfaction that can be further tuned for scheduling problems by using special filtering algorithms and scheduling strategies. Constraint-based scheduling is an efficient tool for solving real-life scheduling problems.

REFERENCES

1. Barták R.: Constraint-Based Scheduling: An Introduction for Newcomers
2. von Brussel H. Manufuture worshop, white paper, 1st July 2004, Dortmund, Germany
3. Dobrzańska, A.: Production Planning in Manufacturing Assembly Systems in Conditions of Resource Constraints (in Polish), PhD Thesis, Silesian University of Technology, Faculty of Mechanical Engineering, Gliwice, 2003
4. Goldratt E.M., Cox J.F.: 1987, The Goal, Coton on Hudson, NY; North River Press
5. Goldratt E.M. Cox J.F.: Łańcuch krytyczny, Werbel, Warszawa 2000
6. Hendry L.C., Kingsman B.G., Production planning systems and their applicability to make to order companies. European Journal of Operational Research, No.40, 1989, pp.1-15
7. Krenczyk D.: Algorithms of the multiassortment repetitive production flow synchronisation (in Polish), PhD Thesis, Silesian University of Technology, Faculty of Mechanical Engineering, Gliwice, 2002
8. Lowa A., Maroto C, Tormos P.: A multicriteria heuristic method to improve resource allocation in multi-project scheduling. European Journal of Operational Research. 127, 2000, pp.408-424
9. NcMullen T.B., Newbold R.C.: Project Management in the Fast Line. Applying the Theory of Constraints, St.Luice Press, 2000
10. Rand G.K.: Critical Chain: The theory of constraints applied to project management, International Journal of Project Management 18, 2000
11. Schonsleben P.: Integral Logistic s Management, St. Lucie Press, London, 2001
12. Simonis H.: Constraint Logic Programming, Cosytec S.A., 11/12/1997, France
13. Skołud B.: Planning of the multiassortment repetitive production (in Polish) Zeszyty Naukowe Politechniki Śląskiej, s. Mechanik, No.136, Gliwice, 2000
14. Skołud B., Zientek A.: Project Management – TOC Approach, Automation 2003, pp 177-184
15. Skołud B., Zientek A.: Constraints identification in multi-project scheduling, AIMETCH 2004 Conference, 129-130
16. http://swz.of.pl
17. http//www.volkswagen-stiftung.de