AGGREGATED MODELS TECHNIQUE FOR INTEGRATING PLANNING AND SCHEDULING OF PRODUCTION TASKS

Ştefan Dumbravă

"Gheorghe Asachi" Technical University of Iasi-Romania, Department of Automatic Control and Applied Informatics
Prof.dr.doc. Dimitrie Mangeron Street, No. 27, 700050, Iasi, Romania

Corresponding author: Ştefan Dumbravă, sdumbrav@ac.tuiasi.ro

Abstract: Effective schedule of production operations that conducts to optimized realistic and feasible solutions is a complex task that has been intensively studied due to its economic importance. Starting from the graph of the detailed problem, a method of optimization the task schedule is emphasized. In order to obtain a minimum time for the project execution together the fulfilment of resource constraints, the detailed model is aggregated, optimized from the point of view of the graph height and cardinality and finally the solution is disaggregated. The method is applied in the case study of a Flexible Manufacturing System for an assembly project that is used for educational purposes and the results are emphasized.
Key words: scheduling algorithms, flexible manufacturing system, execution times, graphs.

1. INTRODUCTION

Production control varies greatly from one company/plant to the other. There are situations where effective production control is possible and there are situations where the manufacturing challenge is so large that it is almost impossible to generate any kind of feasible plan. Due to the market fluctuations, a good strategy is to work with small size inventories and a flexible control system that enables quickly react to the market changing demand, (Kenneth *et al*., 2004). For this purpose, the development of mathematical concepts or various tools and aids for the production control solving problems has been a continuous research preoccupation. Techniques for planning the minimum safety stock, forecast of production volume, a better scheduling in the detailed timetables of the jobs or the computation of the lead time for quoting orders are some examples of research topics in this field. The goal of all these techniques is to improve the production efficiency within modern manufacturing concepts like MRP/CRP systems. Presently many companies use material/capacity requirements planning systems for medium and long term production planning. In spite of MRP benefits, because of the assumptions of infinite resource capacity, the difficulty to model resource constraints, fixed lead times and the heuristic estimation of lead times based on history, the resulting production plans are sometimes unrealistic or unfeasible, (Vollmann *et al*., 1997). The drawbacks of these methods may be accomplished by using complex models, able to characterize different aspects of the manufacturing planning problem. One of the key problems of the production management is that of finding the optimum configuration and exploitation of the manufacturing facilities. It has been admitted that the solution of the problem depends on the considered time horizon. Consequently, it has been divided into three hierarchical planning levels, Fig. 1.

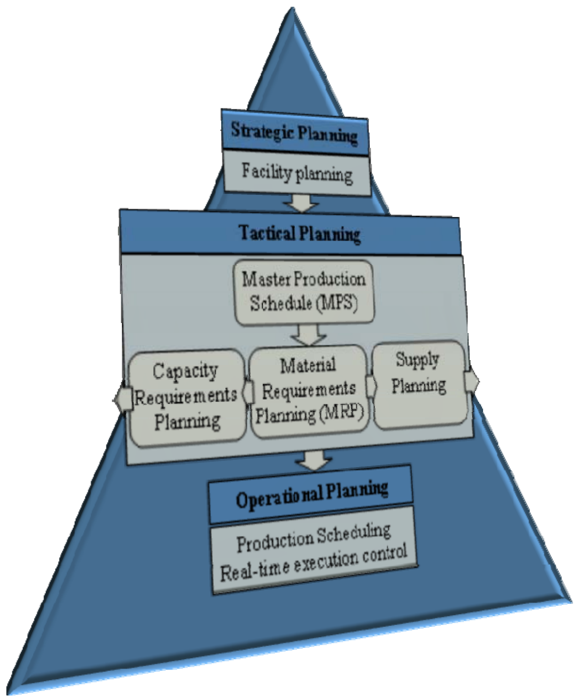

Fig. 1. Levels of the planning hierarchy

The levels of decision making are called strategic (or long-term), tactical (medium-term), and operational (short-term), (Kovacs, 2005). The level with the longest time horizon is the strategic one. At this level decision about final products, production facilities, capital, resources and politics are taken. In medium term planning the decision taken imply the resources and to some degree, policies and facilities.

Operational level is the shortest time horizon and the decisions regard the way in which available resources are used as efficient as possible in order to meet certain goals, like cost minimization and filling customer orders. Production scheduling on the operational level develop the first segments from production plan in detailed resource assignments and operation sequences, (Kovacs, 2005). The last module in the planning hierarchy is the real-time execution control. It ensures a feedback about the status of the shop-floor tasks flow compared to the planning.
The main objective of the paper is to emphasize an aggregate modelling method in solving the production planning problem that enables the solution to be further developed into detailed feasible schedules.

2. METHOD DESCRIPTION

An integrated approach of the production hierarchical planning levels, namely production planning and production scheduling, is proposed. It enables based on an aggregate formulation of the production planning, a feasible detailed schedule of the activities and tasks. Starting from the detailed problem, its decomposed solution is obtained by reformulating the aggregate problem from the detailed one, obtaining the aggregate solution and finally decomposing it. The steps followed in applying the method are emphasized in fig. 2.

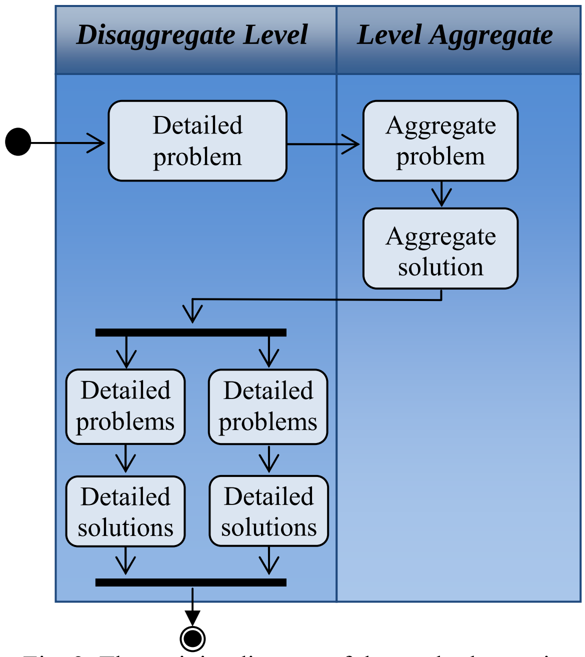

Fig. 2. The activity diagram of the method steps in planning problem solving

The solution of the detailed problem is obtained in three steps:

- The detailed model of the initial planning problem is aggregated, replacing the variables and the set of their constrains in the detailed model with one aggregate variable with an aggregate constrain;
- The aggregate model is solved by an adequate algorithm;
- The solution of the aggregate problem will be decomposed into the detailed solution during disaggregation

The aggregation/disaggregation procedure must maintain the temporal requirements and those of capacity. The aggregation model is a less complex one that permits to find easier an optimal feasible solution.

2.1 Modelling and representation of the Production Planning Problem

The detailed problem is modelled using the classical resource-constraint project scheduling problem *(RCPSP)* model, (Brucker, 2004). Its formulation considers the set of projects P that must be overcome during an apriory known time horizon. Each of the projects $p_i \in P$ has defined the earliest start time t_{s,p_i} and the final execution time t_{f,p_i}. Each of the projects p_i contains a set of tasks T_{p_i}. Each task $t_i \in T_{p_i}$ lasts a fix time d_i and requires a portion of the renewable cumulative resource $r_{t_i} \in R$. The resource capacity is denoted by $q(r_{t_i})$. The tasks belonging to the same project may be connected by precedence constraints $t_i \rightarrow t_j$, meaning that the execution of task t_j can start only after the execution of task t_i is finished. The *RCPSP* model of the detailed production planning problem is a tree whose nodes are the tasks the project and its edges represent the precedence relations between tasks. Running the project starts from the leaves and finishes with the root. A sample project is presented in fig. 3.

2.2 Model aggregation

The aggregation procedure is based on the tree partitioning in connected sub-trees composed of combined tasks that belong to an aggregate activity. The throughput of this operation on the project model is a new tree having in the nodes the activities (composed of the tasks) of the project. The model is an extension of the *RCPSP* one, called resource-constrained project scheduling problem with variable-intensity activities *(RCPSVP)*, (Kovacs et. *al.*, 2004; Rogers et. *al.*, 1991) based on the following formalism. The aggregate problem comprises a set of projects P, a set of activities A that build the projects, a set of continuously renewable and divisible resources R and an acyclic graph G, which describes the end-to-start precedence constraints. The time horizon of the project is assumed known and it is devised into discrete time units Δ of the aggregate planning. The length of Δ is chosen as an integer sub multiply of the project time horizon. In every time unit τ of the discrete time unit Δ, a part of activity A_i is executed. This is denoted by $x_\tau^{A_i}$ and is called the intensity of the activity A_i in time unit τ. Every activity may require the simultaneous use of some resources, proportional with their intensity. The total work of activity A_i using resource r is denoted by:

$$\sigma_r^{A_i} = \sum_{t \in A_i : r_t = r} d_t \qquad (1)$$

During the time unit τ, the A_i activity uses $\sigma_r^{A_i} x_\tau^{A_i}$ units of resource r. The solution of the aggregate problem consists in the calculation of the intensities for every activity in each time unit τ so that the precedence and temporal constraints to be fulfilled while the maximum capacity of the resources is not exceeded and the total cost is minimized. The partition of the project tree is called the aggregate project model. If two tasks connected by a precedence constraint of the project tree are inserted into the same activity then the constraint is omitted, while it is maintained if the tasks belong to two different activities. The activities precedence graph is also a tree. The activity A_i for which $\sum_{t \in A_i} d(t) = \Delta$ is called complete, while those for which $\sum_{t \in A_i} d(t) > \Delta$ are called broken activities. In order to characterise the aggregate model there are introduced the following properties:

- $\omega(A_i)$ is called the weight of activity A_i and represent an estimation of the required time necessary to execute all the tasks included in that activity.
- The cardinality of the aggregate model is denoted by $c(P)$.
- The height of the P tree is denoted by $h(P)$ and it is given by the longest path from the leave to the root.

The aggregate model will diminish the computational complexity of the planning problem but two long activities may affect the feasibility of the short-term schedules, (Toye, et. al., 1990). A good compromise is related to the assignment of the activities weight equal to the discrete unit of time that results from the discretization of the project time horizon. However, the model aggregation is not a relaxation of the detailed model since it introduces new constraints resulting from the time horizon aggregation in discrete time units. Thus, two coupled activities might execute in different discrete time units. Starting from a detailed model there are more than one solution for obtaining the aggregate model. A minimal height representation conducts to smaller execution times and an increase of parallelism in activities. Based on the above remarks, an aggregate optimal model of a project is defined as follows.

- An activity A_i is a connected component of the project tree if the estimated required time of its execution, expressed by $\omega(A_i)$, fits into an aggregate time unit.

The optimal aggregate model P of a tree T means finding a partition so that both $h(P)$ and $c(P)$ are minimal.

2.3 Optimal form of the aggregate model

Given the tree project $T = (V, E, r)$, where V, E and r are the vertices, edges and the root of the tree then $P = \{sT_1, ..., sT_q\}$ is a partitioning if and only if each component sT_i is a sub tree (connected sub graph) of T, the sT_i components are disjointed and the union of the vertex-sets $V(sT_i)$ of the sT_i equals V.

The root component of P is the one containing r, and will be denoted by $RC(P)$. A component weight function $\omega : sT \to R_+$ on the sub-trees of T and a real positive constant W are defined.

The partitioning $P = \{sT_1, ..., sT_q\}$ of T is admissible if and only if $\omega(sT_i) \le W$, for every $sT_i \in P$. We assume that $\omega(\{v\}) \le W$ for each $v \in V$, which implies that the trees have always admissible partitions. Furthermore, we introduce the notation of $r\omega(P) = \omega(RC(P))$ for the weight of the root component of P.

The function ω is said to be monotonous if for two sub trees sT_1 and sT_2 with $sT_1 \subseteq sT_2$ then $\omega(sT_1) \le \omega(sT_2)$.

It is denoted by $S(v)$ the set of the sons of vertice v and $T(u)$ a subtree of T rooted at $u \in S(v)$. The level of a vertice $u \in T(u)$ is defined as the height of an optimal partitioning of a sub tree $T(u)$ with $u \in S(v)$. Considering P_v and P_u the partitioning of $T(v)$ and $T(u)$ respectively, the maximum height of a partition P_u with $u \in S(v)$ is $h_{\max} = \max_{u \in S(v)} (h(P_u))$ and $K = \{u \mid u \in S(v) \wedge h(P_u) = h_{\max}\}$. Based on the above notations the following partitions are defined:

$$P_{1v} = comb(\{P_u \mid u \in S(v)\}, K) \qquad (2)$$

$$P_{2v} = comb(\{P_u \mid u \in S(v)\}, \Phi) \qquad (3)$$

The algorithm runs into two steps. In the first step the initialization $P_v = \{v\}$ is done. The second step is an iterative one, during which one vertice v with processed sons is chosen. The optimal partitioning is found based on the optimal partitioning of the sub trees $T(u)$. This step is repeated until the root r is found. The comb operator is applied to the optimal partitioning P_u with $u \in S(v)$, obtaining the partitions P_{1v} and P_{2v} having the heights $h(P_{1v}) = h_{\max}$ and

$h(P_{2v}) = h_{\max} + 1$. If P_{1v} is admissible then the algorithm assigns $P_v = P_{1v}$. Else the equality $P_v = P_{2v}$ takes place. P_{2v} is always admissible because it consists of sub trees belonging to an admissible partitioning P_u. In order to decide if P_{1v} is admissible $J = \{v\} \cup \{x \mid x \in T(u) \wedge u \in K\}$. If $\omega(J) \le W$ then P_{1v} it is admissible and $P_v = P_{1v}$ is the partitioning of minimal height.

2.4 Disaggregation

Disaggregation of the production plan involves ordering each task of the detailed model into one aggregate time unit (Kovacs, *et. al.,* 2005). The disaggregation of the production plan is complete with solving the detailed scheduling problems corresponding to the aggregate time units. The representation of the programming problem based on constraints is formalised as follows. A sequence of operations based on constraints $\prod$ is defined by the $\{X, D, C, O\}$, where $X = \{x_i\}$ is a finite set of variables, each of them x_i may take values in its domain D_i taking into account a defined set of constraints C. The solution represents a sequence S of the variables x_i so that $\forall x_i \in X : x_i = v_i^S \in D_i$, and the constraints are fulfilled $\forall c \in C : c(v_{i_1}^S, ..., v_{i_N}^S) = true$. O - is the last component of the quadruple, denotes an objective function that assigns a real number to a solution *S*. The *RCPS* model in this formalism is composed of the variables representing the start moments of the tasks denoted by $start_t$, where the tasks $t_i \in T$. The task duration d_t is assumed to be an integer number. Consequently, the initial domain of the variables is a set of integer numbers. These domains are further restricted by the constraints that may be of temporal, precedence and resource capacity type. The temporal constraints are conditioning the start and the end of the tasks, and belong to only one variable. The precedence constraints are related to more than one task and impose the order of the tasks execution. The resource capacity constraint ensures that the total capacity required does not exceed the available one. A resource capacity constraint is responsible of a resource r and all the tasks that need that resource. An optimisation problem involves the minimisation of function O, meaning the minimisation of the processing time of all the tasks.

3. JOB SCHEDULE IN THE FLEXIBLE MANUFACTURING SYSTEM

For the purpose of the paper, a real world educational Flexible Manufacturing System (FMS) developed within the faculty department was considered. The layout of the system, composed of two flexible manufacturing cells, is shown in Fig. 3. The first manufacturing cell has the task of automatically machining of the raw parts being developed around the robot #1 which manipulates the parts from the raw part storages to the milling machine and from the milling machine to the part buffer, or to the conveyor. The second cell is composed of robot #2, the visual inspection station and the finished part storage. The two cells are exchanging parts using a circular conveyor with pallets.

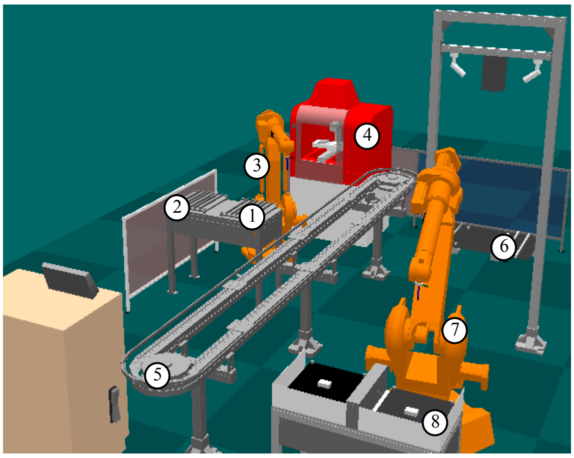

Fig. 3. A 3D view of the FMS layout, [2]
1 - The part storage M1; 2 - The processed part buffer; 3 - ABB robot#1 (IRB 1400); 4 - The CNC Milling machine; 5 - The Flex Link conveyor; 6 – Assembly and visual inspection computer station; 7 - ABB Robot#2 (IRB 2400); 8 - The part storage M2

The main objective of the system is to produce a mixed type of parts, belonging to a group with similar features and to deposit them on the indicated pallets of finished parts in the programmed positions. The finished parts are inspected by the visual inspection station within the on-line quality system.

The manufacturing project consists of assembly operations of a final product which are scheduled using the aggregation/ disaggregation method. The resources of the system are defined as follows: r_1- robot#1, r_2 - the CNC milling machine, r_3- the conveyor, r_4 - robot#2, r_5 - Visual inspection computer station and assembly centre. The tasks of the assembly project are denoted by t_i, $i = 1,12$: t_1- positioning of the robot#1 at the raw part storage M1, t_2 - a raw part is picked and it is carried to the CNC milling machine by robot#1; t_3 - the CNC milling machine is setup; t_4 - CNC milling machine is processing the part of type A; t_5 - The part A is unloaded from the milling machine and is carried to the conveyor; t_6 - the conveyor is carrying the part A to the assembly centre; t_7 - robot#2 takes the part from the conveyor and places it to the visual inspection system; t_8 - robot#2 is positioning at storage M2; t_9 - a part B is carried from the storage M2 to the visual inspection and assembly station by robot#2; t_{10} - robot#2 fulfils the assembly task; t_{11} - the assembly is inspected by the visual inspection system; t_{12} - the assembly part is carried to the storage M3 by robot #2. The detailed tree of the manufacturing project is built, taken into consideration the sequence of the tasks, fig. 4, with the model parameters given in table 1.

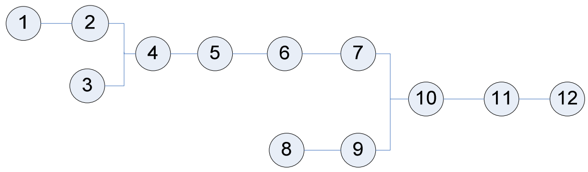

Fig 4. The project tree

Table 1. The parameters of the detailed project model

t_i	$d(t)$	$r_i(t)$
1	2	r_1
2	2	r_1
3	1	r_2
4	7	r_2
5	1	r_1
6	2	r_3
7	2	r_4
8	3	r_4
9	2	r_4
10	8	r_4
11	11	r_5
12	4	r_4

Based on the detailed graph a partition that is not optimised into activities is obtained as in fig. 5.

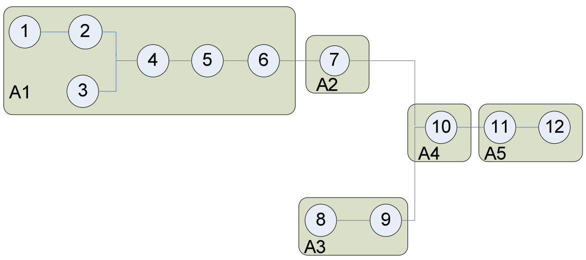

Fig. 5. The project partitioning without optimisation

In the second step of the algorithm, the aggregation model of minimal height is generated. In the first step it is obtained the $P_v = \{t_1, t_3, t_8\}$ partition represented in fig. 6.

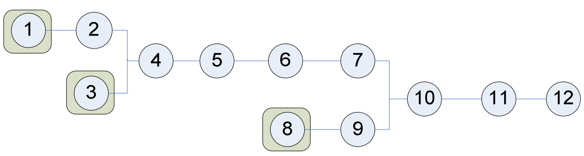

Fig. 6. The partition obtained after the first step of the algorithm

In the second iterative step, an unprocessed task with predecessor tasks processed is chosen. There are obtained the partitions $P_{u1} = \{\{t_1, t_2, t_3, t_4, t_5, t_6\}, \{t_7\}\}$ and $P_{u2} = \{\{t_8, t_9\}\}$, where $\{u_1, u_2\} \in S(v), v = t_{10}$ $u_1 = t_7$ and $u_2 = t_8$.

Furthermore, the activities $A_1 = \{t_1, t_2, t_3, t_4, t_5, t_6\}$, $A_2 = \{t_7\}$ and $A_3 = \{t_8, t_9\}$ are built in accordance to the constraint $\omega(A_i) \leq \Delta$. On the obtained partitions the comb operator is applied. There are chosen $K_1 = t_7$, $K_2 = t_9$ which conducts to the partitions, fig. 7.

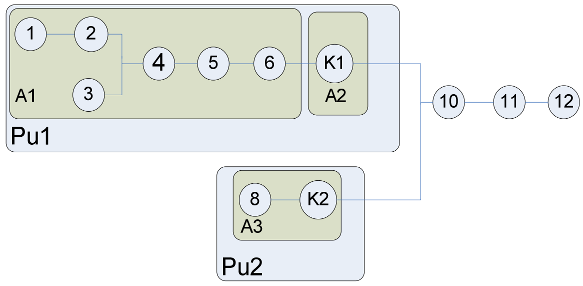

Fig. 7. The partition obtained after the comb operator was applied

The admissibility of this partition it is verified, checking the condition $\sum_{i=7}^{10} \omega(t_i) = 15 \leq \Delta$ with $\Delta = 15$.

In the next step the final optimal partition is obtained as: $P_v = \{\{t_1, t_2, t_3, t_4, t_5, t_6\}, \{t_7, t_8, t_9, t_{10}\}, \{t_{11}, t_{12}\}\}$.

It results an optimal model composed of 3 aggregated activities: $A_1 = \{t_1, t_2, t_3, t_4, t_5, t_6\}$, $A_2 = \{t_7, t_8, t_9, t_{10}\}$, $A_3 = \{t_{11}, t_{12}\}$ with minimal height $h(P_v) = 2$ and cardinality $c(P_v) = 5$, fig. 8.

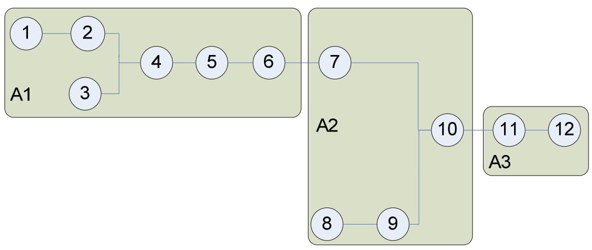

Fig. 8. The aggregated model of minimum height

Table 2. The minimal height aggregated model parameters

	$\sigma^A_{r_1}$	$\sigma^A_{r_2}$	$\sigma^A_{r_3}$	$\sigma^A_{r_4}$	$\sigma^A_{r_5}$	σ^A_r
A_1	5	8	2			15
A_2				15		15
A_3				4	11	15

4. JOB SCHEDULE GENERATION

The disaggregated schedule of the obtained model consists in transforming the 3 activities into three timetables, each of them representing the task schedule on resources during every time unit W and establishing for every task the start time and the finish time. The parameters of the disaggregated model are given in table 3. The timetables are built using constraint-based solver and are represented in figs. 9-11.

Table 3. The parameters of the disaggregated model

t_i	$d(t)$	$r_i(t)$	$t_{s,p}$	$t_{f,p}$
1	2	r_1	0	2
2	2	r_1	2	4
3	1	r_2	0	1
4	7	r_2	4	11
5	1	r_1	11	12
6	2	r_3	12	14
7	2	r_4	0	2
8	3	r_4	2	5
9	2	r_4	5	7
10	8	r_4	7	15
11	11	r_5	0	11
12	4	r_4	11	15

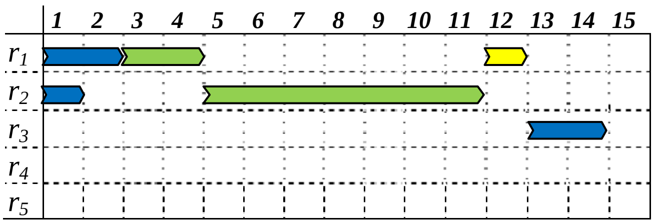

Fig. 9. The corresponding A_1 activity timetable of the aggregated model

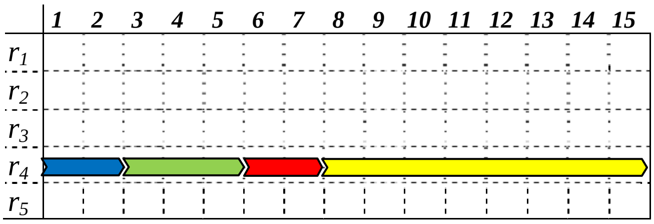

Fig. 10. The corresponding A_2 activity timetable of the aggregated model

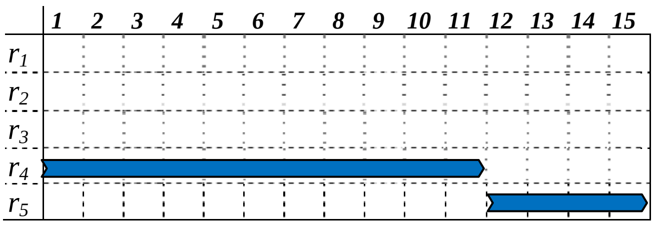

Fig. 11. The corresponding A_3 activity timetable of the aggregated model

5. CONCLUSIONS

At the operational level in production planning hierarchy, the production scheduling of tasks is a complex operation that must conduct to realistic and feasible timetable of operations. Consequently, a model of the system that enables to emphasize the shop constraints and the finite capacity of resources together the sequence constraints is required. Due to the complexity of such models an optimized solution is difficult to be obtained. A method to solve the problem is the aggregation of operation that maintains the initial constraints and conduct to simpler models. The optimization of the aggregated models refers to the height of the model graph and its cardinality. Algorithmically, the optimized aggregated model is obtained and the timetable of the operation is detailed by disaggregation of the optimized model. In this paper, the main notation together the steps of the algorithm are presented considering a case study. The main objective of the paper is to emphasize the aggregate/disaggregate method applied in the case of an educational Flexible Manufacturing System. Starting from an assembly project the activities, the task and the resource occupation are finally obtained.

6. REFERENCES

1. Brucker, A. Drexl, R. Mohring, K. Neumann & E. Pesch (2004). *Resource-constrained project scheduling: Notation, classification, models, and methods. European Journal of Operational Research, 112(1): 3–41,* ISSN: 0377-2217.

2. Kenneth N. McKay & Vincent C.S. Wiers (2004). *Practical Production Control a survival guide for planners and schedulers,* Co-published with APICS, pp.27-45, ISBN: 1-932159-30-4, Printed and bound in the USA.

3. Kovacs (2005). *Novel Models and Algorithms for Integrated Production Planning and Scheduling,* Ph. D. Thesis, Budapest University of Technology and Economics.

4. Kovacs & T. Kis (2004). *Partitioning of trees for minimizing height and cardinality. Information Processing Letters, 89(4):181–185,* ISSN: 0020-0190.

5. Markus, J. Vancza, T. Kis & A. Kovacs. (2003.). *Project scheduling approach to production planning. CIRP Annals – Manufacturing Technology, 52(1):359–362,* ISSN: 0007-8506.

6. Rogers, R.D. Plante, R.T. Wong & J.R. Evans. (1991.). *Aggregation and disaggregation techniques and methodology in optimization. Operations Research, 39(4):553–582,* ISSN: 0030-364X.

7. Toye, C. A. (1990). *Let's Update Capacity Requirements Planning Logic*. Proceedings of the American Production and Inventory Control Conference, Atlanta, Georgia: American Production and Inventory Control Society.

8. T. E. Vollmann, Berry W.L., Jacobs F.R. & Whybark D.C. (2004). *Manufacturing Planning and Control Systems, 5th ed.* McGraw-Hill, ISBN 10: 007144033X, New York.

Received: December 20, 2010 / Accepted: May 30, 2011 / Paper published online: June 10, 2011