Proceedings of the 8th IEEE Mediterranean Conference
on Control and Automation (MED 2000)
Rio, Patras, GREECE 17-19 July, 2000

A DECISION SUPPORT MODEL FOR TECHNICAL DATA AGGREGATION AND FLOW ASSESSMENT IN PRODUCTION NETWORKS

P. Baillet-Farthouat, F. Pereyrol and J.P. Bourrieres

LAP/GRAI University of Bordeaux/ENSMM, CNRS EP 2026 Fax: +33 5 56 84 66 44
Email : Farthouat@lap.u-bordeaux.fr

Abstract *A multi-level technical data model and a method to assess the throughput of aggregated production units is presented. A generic template guides the application of recursive aggregation operators which are assumed to be defined by a human expert. This approach aims to support the macro-evaluation of performances to be expected from networked complex manufacturing units.*

Key words :Global Manufacturing, Data Aggregation, Production Networks, Flow Assessment

1. Introduction

Flow optimization in manufacturing workshops -both at the design an operating phases-, is based on the search for bottlenecks within a network of machines interconnected by part transport routes. Operating a distributed set of networked resources can actually be seen as a much more general problem to be instantiated at different levels of resource integration : cells, workshops, plants and even a virtual production unit resulting from the interconnection of enterprises within an industrial partnership (Extended Enterprise concept)[1]. The allocation of tasks to resources can itself be seen as a generic problem. Its formulation requires a set of more or less aggregated data (product/process data, resource availability, routing capacity, etc…) depending on the level considered [2], [3]. This paper presents a multi-level data model to support the assessment of distributed production systems flows using bottom-up aggregation of technical data on products, processes and resources. It is assumed that the choice of aggregation mechanisms is made by a human expert depending on the application case, using the generic framework presented here.

The results are based on some restrictive conditions. The workstations are non cumulative, so that each of them can perform only a single type of task at once. The physical structure of the routing network as well as the logical structure of the process (precedence constraints between tasks) are depicted by graphs with some topological particularities. Nevertheless, these hypothesis will be verified in most of the flow shops.

The generic formulation of the multi-level technical data model is first developed in Section 2. The flow capacity of the resources and of the transport channels are then introduced in Section 3. A method to calculate the capacity of an aggregated resource – outer view of networked resources- is then presented in detail in Section 4 to 6.

The data model is based on [4]:
– a set of tasks
– a set of resources
– a graph of task interrelation
– a graph of resource interrelation

These entities are recursive, which means that :
– each task (respectively resource) can be considered as the reduced view of an interrelation graph between more detailed tasks (respectively resource). This top-down refinement of technical data can be performed until reaching the lowest level of description of tasks and resources , *i.e* operations and workstations.
– each task (respectively resource) can be considered as a component of a larger process (respectively network). This bottom-up aggregation mechanism can be *ad lib* iterated until reaching a relevant level of representation.

2. Model entities and notations

2.1. Data levels

The notations used in the paper make a distinction between the *level of encapsulation* and the *level of detail.*

v-level of encapsulation

A task or a resource seen as whole at level v is called a v-level entity *(*$v\in$N*)*. At the lowest level $v=0$ are to be found the workstations and the operations.

k-level of detail

A v-level entity can be seen at more or less deep level of details through iterative break up. Let k *(*$k\in$ N*)* denote the level of representation of the entity, assumed that $k\leq v$. The representation depth of a v-level entity is then v-k.

When a v-level entity is seen *externally*, then $v=k$. If term “ level ” is used without further precision, then it is assumed that $v=k$. Most of the time, the *internal* view of an entity will be given with representation depth v-$k=1$.

More generally, the following notations are introduced (note that tasks and resources are in italics, whereas sets of tasks and sets of resources are plain) :

$T_i^{v,k}$	v-level task i seen at level k
$T_i^{v,v}$	external view of v-level task i
$\mathrm{T}_i^{v,v-1}=\{\ T_j^{v-1,v-1}\ \}$	set of v-1 level tasks within v-level task i
$\mathrm{G}_i^{v,v-1}=$ card $\mathrm{T}_i^{v,v-1}$	
$R_i^{v,k}$	v-level resource i seen at level k
$R_i^{v,v}$	external view of v-level resource i
$\mathrm{R}_i^{v,v-1}=\{\ R_j^{v-1,v-1}\ \}$	set of v-1 level resources within v-level resource i
$\mathrm{H}_i^{v,v-1}=$ card $\mathrm{R}_i^{v,v-1}$	

2.2. Grouping model entities

The data aggregation process first consists in defining the subsets of entities to be encapsulated. Set of v-level tasks $\mathcal{T}^{v,v}$ is *partitioned* into subsets $\mathcal{T}_i^{v+1,v}$ as follows :

$$\bigcup_{i=1..G^{v+1,v+1}} \mathcal{T}_i^{v+1,v} = \mathcal{T}^{v,v} \qquad (1)$$

$$\bigcap_{i=1..G^{v+1,v+1}} \mathcal{T}_i^{v+1,v} = \varnothing \qquad (2)$$

and set of v-level resources $\mathcal{R}^{v,v}$ is partitioned into subsets $\mathcal{R}_i^{v+1,v}$ as :

$$\bigcup_{i=1..H^{v+1,v+1}} \mathcal{R}_i^{v+1,v} = \mathcal{R}^{v,v} \qquad (3)$$

$$\bigcap_{i=1..H^{v+1,v+1}} \mathcal{R}_i^{v+1,v} = \varnothing \qquad (4)$$

Grouping tasks and resources iteratively leads to a multi-level task and resource composition tree as shown on Figures 1 and 2.

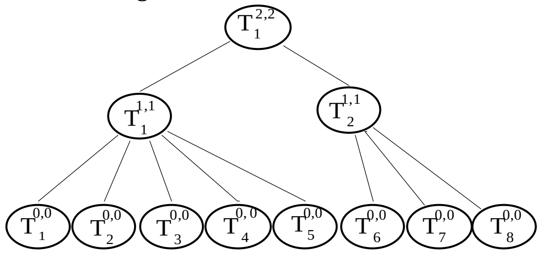


Figure 1 : Task grouping tree

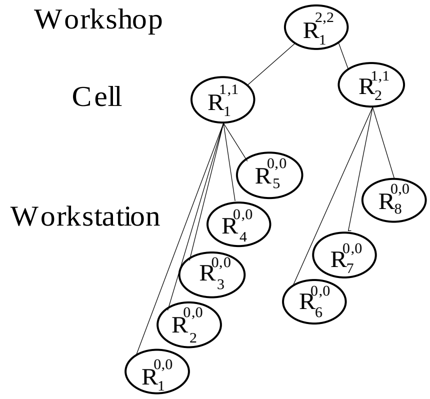
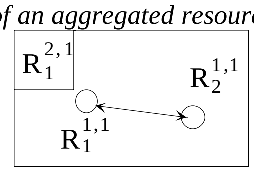


Figure 2 : Resource grouping tree

2.3. Resource and task breakup

Let $R_p^{v+1,v+1}$ be a resource at level v+1. The internal view $R_p^{v+1,v}$ of this aggregated resource is a network interconnecting a set of v-level resources. It can be represented by a break up graph as shown on Figure 3.

Break up of an aggregated resource (depth 1)



Break up of an aggregated resource (depth 2)

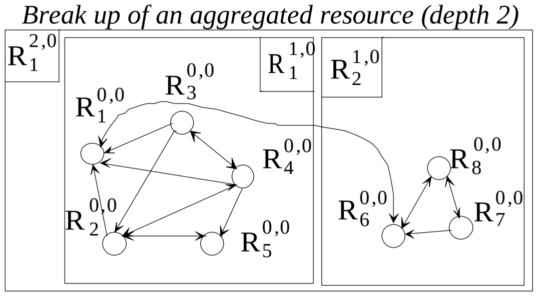


Figure 3 : Aggregated resource break up

Let $T_p^{v+1,v+1}$ be a task at level v+1. The internal view $T_p^{v+1,v}$ of this task $T_p^{v+1,v+1}$ is a manufacturing process involving a set of v-level tasks. It can be represented by a break up graph as shown on Figure 4.

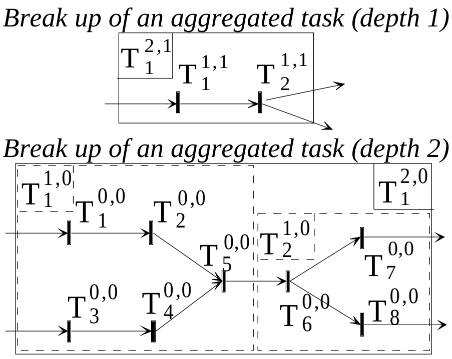


Figure 4 : Aggregated task break up

3. Resource and transport capacity

3.1. *Resource performance*

Whatever the level v of a resource is, its production capacity can be defined in terms of maximum flow, i.e. the mean number of tasks the resource is able to achieve per time unit. More precisely, the flow is considered for each type of task, assumed that the resource is dedicated to one single type of task at once. The capacity of $R_j^{v,v}$ resource with regard to $T_i^{v,v}$ task is a non negative real number noted $\alpha_p^{v+1,v}(T_i^{v,v}/R_j^{v,v})$. Considering the set of internal resources within $R_p^{v+1,v+1}$ leads to introduce the flow matrix :

$$A_p^{v+1,v} = [\,\alpha_p^{v+1,v}(T_i^{v,v}/R_j^{v,v})] \qquad (5)$$
$$i = 1, \ldots, G_p^{v+1,v} \qquad j = 1, \ldots, H_p^{v+1,v}$$

Note that resources capacities are supposed to be available at level 0 (workstations capacity). The main purpose of this paper is to assess the flow capacity of aggregated resources at any level v>0 through iterative bottom-up aggregation..

3.2. *Transport performance*

With resource $R_p^{v+1,v+1}$ can then be associated the flow matrix :

$$B_p^{v+1,v} = [\,\beta_p^{v+1,v}(R_i^{v,v}/R_j^{v,v})] \qquad (6)$$
$$i,j = 1, \ldots, H_p^{v+1,v}$$

with $\beta_p^{v+1,v}(R_i^{v,v}/R_j^{v,v})$ the maximum flow ensured by the transport channel from $R_i^{v,v}$ to $R_j^{v,v}$ within the interconnection network between the *v*-level resources. It is assumed that square matrix $B_p^{v+1,v}$ has no circuit.

4. Flow aggregation :method

Knowing the resource and routing capacities at level *v*, the issue is here to evaluate the throughput of an aggregated resource at level *v+1*, agreed that the tasks are themselves aggregated. Basically, different *aggregation paths* [5] can be followed. In this paper, the flow will be aggregated first *along the resource axis, i.e.* the capacity of a *v+1* level with regard to *v*-level tasks, then *along the task axis*, *i.e.* the capacity of a *v+1* level resource with regard to *v+1* level tasks. The method and corresponding notations are first introduced, then applied in detail in Section 5 and 6.

4.1. *Generic notations*

Task- transition matrix

Let $T_i^{v,v} \rightarrow T_j^{v,v}$ denote the task sequence "$T_i^{v,v}$ then $T_j^{v,v}$" and :

$$\gamma_p^{v+1,v}(T_i^{v,v}/T_j^{v,v})$$

the number of times the sequence $T_i^{v,v} \rightarrow T_j^{v,v}$ is performed by a given resource $R_p^{v+1,v}$ and by time unit. The flow capacity of the network with regard to the different possible task sequences is depicted by the following matrix :

$$C_p^{v+1,v} = [\gamma_p^{v+1,v}(T_i^{v,v}/T_j^{v,v})] \qquad (7)$$
$$i, j = 1, \ldots, G_p^{v+1,v}$$

It is assumed that square matrix $C_p^{v+1,v}$ has no circuit.

Flow aggregation along resource axis

Let finally introduce the flow capacity of resource $R_p^{v+1,v+1}$ with regard to *v*-level task $T_i^{v,v}$ by the line-matrix :

$$D_p^{v+1,v} = [\delta_p^{v+1,v}(T_i^{v,v}/R_p^{v+1,v+1})] \qquad (8)$$
$$i = 1, \ldots, G_p^{v+1,v}$$

4.2. *Recursive aggregation*

Flow aggregation is a bottom-up recursive process starting at level *0*. Given the technical data at level *0* (resource and transport flow capacities), matrices $A_p^{1,0}$ and $B_p^{1,0}$ can be valuated for each resource *p*. Then the flow capacity is aggregated along the resource axis, matrices $C_p^{1,0}$ and $D_p^{1,0}$ being calculated as shown in section 5. Finally, the capacity is aggregated along the task axis, which provides matrices $A_p^{2,1}$ and $B_p^{2,1}$ at the upper level. At this point, one step of aggregation is completed and the process can be recursively iterated, as shown on Figure 5.

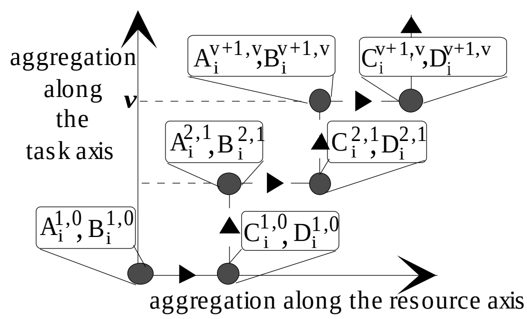


Figure 5 : Iterative aggregation path

5. Flow aggregation along resource axis

Matrices $A_i^{0,0}$ and $B_i^{0,0}$ are the technical data at the shop-floor level and assumed to be given. The next developments refer to the example of task and resource break up shown on Figures 1 and 2.

5.1. *Shopfloor level data*

Matrices $A_1^{1,0}$, $B_1^{1,0}$ $A_2^{1,0}$, $B_2^{1,0}$ are given :

$A_1^{1,0}$	$T_1^{0,0} \dots T_5^{0,0} \dots T_8^{0,0}$
$\begin{array}{c} R_1^{0,0} \\ \dots \\ R_3^{0,0} \\ \dots \\ R_5^{0,0} \end{array}$	$\begin{pmatrix} 0 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 3 & 0 & 0 & 0 & 0 \\ 0 & 3 & 4 & 0 & 0 & 0 & 0 & 0 \\ 7 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \end{pmatrix}$

$B_1^{1,0}$	$R_1^{0,0} \dots R_3^{0,0} \dots R_5^{0,0}$
$\begin{array}{c} R_1^{0,0} \\ \dots \\ R_3^{0,0} \\ \dots \\ R_5^{0,0} \end{array}$	$\begin{pmatrix} 24 & 0 & 0 & 0 & 0 \\ 6 & 15 & 0 & 3 & 5 \\ 7 & 5 & 30 & 2 & 0 \\ 5 & 8 & 4 & 10 & 7 \\ 0 & 6 & 0 & 0 & 20 \end{pmatrix}$

$A_2^{1,0}$	$T_1^{1,0} \dots T_5^{0,0} \dots T_8^{0,0}$
$\begin{array}{c} R_6^{0,0} \\ R_7^{0,0} \\ R_8^{0,0} \end{array}$	$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \end{pmatrix}$

$B_2^{1,0}$	$R_6^{0,0} \dots R_8^{0,0}$
$\begin{array}{c} R_6^{0,0} \\ R_7^{0,0} \\ R_8^{0,0} \end{array}$	$\begin{pmatrix} 10 & 7 & 5 \\ 5 & 20 & 6 \\ 4 & 6 & 16 \end{pmatrix}$

5.2. *Calculus of C matrices*

There are as many matrices $C_i^{1,0}$ as macro resources $R_i^{1,1}$ at level *1*.

To compute $C_i^{1,0}$, it is necessary to assess the flow capacity of network $R_i^{1,0}$ with regard to any sequence of two consecutive tasks $T_a^{0,0} \rightarrow T_b^{0,0}$.

Depending on the topology of the resource network (see Figure 2) -assumed to be free of circuit-, some typical cases can be found. An example of calculation is hereafter given for each case.

Case 1 (serial)

$$R_4^{0,0} \longrightarrow R_5^{0,0}$$

Let, as an example, consecutive task $T_3^{0,0}$ then $T_1^{0,0}$ to be carried out. The flow capacity of path $R_4^{0,0} \rightarrow R_5^{0,0}$ is then :

$$\gamma_1^{1,0}(T_3^{0,0};\, T_1^{0,0}) = Min\,\{\alpha_1^{1,0}(T_3^{0,0}/R_4^{0,0}),\ \alpha_1^{1,0}(T_1^{0,0}/R_5^{0,0}),\ \beta_1^{1,0}\,(R_4^{0,0}/R_5^{0,0})\} = Min\,\{4,7,7\}=4$$

Case 2 (split):

$$R_3^{0,0} \longrightarrow R_2^{0,0}, \quad R_3^{0,0} \longrightarrow R_4^{0,0}$$

Consecutive tasks $T_4^{0,0} \rightarrow T_2^{0,0}$ can be carried out with the following flow capacity :

$$\gamma_1^{1,0}\,(T_4^{0,0};\, T_2^{0,0}) = Min\{(Min(\alpha_1^{1,0}\,(T_2^{0,0}/R_2^{0,0}),\ \beta_1^{1,0}(R_3^{0,0}/R_2^{0,0})) + Min(\,\alpha_1^{1,0}\,(T_2^{0,0}/R_4^{0,0}),\ \beta_1^{1,0}(R_3^{0,0}/R_4^{0,0}))),\ \alpha_1^{1,0}(T_4^{0,0}/R_3^{0,0})\} = Min\{(Min(2,5)+ Min(3,2)),\, 3\} = 3$$

Case 3 (fuse) :

$$R_3^{0,0} \longrightarrow R_1^{0,0}, \quad R_4^{0,0} \longrightarrow R_1^{0,0}$$

Consecutive tasks $T_3^{0,0} \rightarrow T_5^{0,0}$

$$\gamma_1^{1,0}(T_3^{0,0};\, T_5^{0,0}) = Min\{(Min(\alpha_1^{1,0}\,(T_3^{0,0}/R_3^{0,0}),\ \beta_1^{1,0}(R_3^{0,0}/R_1^{0,0})) + Min(\,\alpha_1^{1,0}\,(T_3^{0,0}/R_4^{0,0}),\ \beta_1^{1,0}(R_4^{0,0}/R_1^{0,0}))),\ \alpha_1^{1,0}(T_5^{0,0}/R_1^{0,0})\} = Min\{(Min(5,7)+ Min(4,5)),\, 6\} = 6$$

Case 4 (parallel)

$$R_2^{0,0} \longrightarrow R_4^{0,0}, \quad R_4^{0,0} \longrightarrow R_3^{0,0}$$

Consecutive tasks $T_2^{0,0} \rightarrow T_3^{0,0}$

$$\gamma_1^{1,0}(T_2^{0,0};\, T_3^{0,0}) = \{Min(\alpha_1^{1,0}(T_2^{0,0}/R_2^{0,0}),\ \alpha_1^{1,0}(T_3^{0,0}/R_4^{0,0}),\ \beta_1^{1,0}\,(R_2^{0,0}/R_4^{0,0})) + Min(\alpha_1^{1,0}(T_2^{0,0}/R_4^{0,0}),\ \alpha_1^{1,0}(T_3^{0,0}/R_3^{0,0}),\ \beta_1^{1,0}(R_4^{0,0}/R_3^{0,0}))\} = Min(2,5,3)+ Min(3,5,4) = 5$$

Each entry of $C_1^{1,0}$ and $C_2^{1,0}$ is calculated as explained before, which leads finally to :

$C_1^{1,0}$	$T_1^{0,0} \dots T_5^{0,0} \dots T_8^{0,0}$
$\begin{array}{c} T_1^{0,0} \\ \dots \\ T_3^{0,0} \\ \dots \\ T_5^{0,0} \\ \dots \\ \\ T_8^{0,0} \end{array}$	$\begin{pmatrix} 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 3 & 5 & 5 & 0 & 0 & 0 \\ 4 & 5 & 0 & 5 & 6 & 0 & 0 & 0 \\ 2 & 3 & 3 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

$C_2^{1,0}$	$T_1^{0,0} \dots T_5^{0,0} \dots T_8^{0,0}$
$\begin{array}{c} T_1^{0,0} \\ \dots \\ \\ \\ \\ T_6^{0,0} \\ \dots \\ T_8^{0,0} \end{array}$	$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 4 \\ 0 & 0 & 0 & 0 & 0 & 5 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 4 & 4 & 0 \end{pmatrix}$

5.3. *Calculus of D matrices*

The flow capacity of aggregated resource $R_j^{1,1}$ with regard to task $T_i^{0,0}$ is a function with the following form :

$$\delta_1^{1,0}(T_i^{0,0}/R_j^{1,1}) = F\{\alpha_1^{1,0}(T_i^{0,0}/R_1^{0,0}),\alpha_1^{1,0}(T_i^{0,0}/R_2^{0,0}),\ \dots,\alpha_1^{1,0}\,(T_i^{0,0}/R_p^{0,0})\}$$

with $p = \text{card } R_j^{1,0}$ (9)

Function F is to be defined *ad lib* by the analyst on the basis of Min, Max, average etc… aggregation operators. Note that Max function provides an optimistic assessment of the capacity, whereas Min function underestimates the capability of the physical system.

Assuming here that operator Max is applied to the example previously presented :

$$\delta_1^{1,0}(T_2^{0,0}/R_1^{1,1}) = Max\{\alpha_1^{1,0}(T_2^{0,0}/R_1^{0,0}),\ \alpha_1^{1,0}(T_2^{0,0}/R_2^{0,0}),\ \alpha_1^{1,0}(T_2^{0,0}/$$

$R_3^{0,0}), \alpha_1^{1,0}(T_2^{0,0}/ R_4^{0,0})\ \alpha_1^{1,0}(T_2^{0,0}/ R_5^{0,0})\}=$
$Max(0,2,0,3,0)=3$

Each entry of $D_1^{1,0}$ and $D_2^{1,0}$ is likewise calculated, which yields to :

$D_1^{1,0}$	$T_1^{0,0} \ldots T_5^{0,0} \ldots T_8^{0,0}$	$D_2^{1,0}$	$T_1^{0,0} \ldots T_5^{0,0} \ldots T_8^{0,0}$
$R_1^{1,1}$	$(7\ 3\ 5\ 3\ 6\ 0\ 0\ 0)$	$R_2^{1,1}$	$(0\ 0\ 0\ 0\ 0\ 8\ 5\ 4)$

6. Flow aggregation along task axis

Starting from matrices $\boldsymbol{A}^{1,0}$, $\boldsymbol{B}^{1,0}$, $\boldsymbol{C}^{1,0}$, $\boldsymbol{D}^{1,0}$ stated at level *0*, the purpose is here to compute matrices $\boldsymbol{A}^{2,1}$, $\boldsymbol{B}^{2,1}$ at level *1*, in other words to calculate the throughput of *1*-level resources with regard to *1*-level tasks. One complete step of aggregation will then be done (see Figure 5).

6.1. Calculus of A matrices

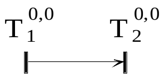
An aggregated task at level *v* is composed of a set of tasks at level *v-1* interrelated according to a specific graph (see Figure 3). The question is here to find $\alpha_p^{2,1}(T_i^{1,1}/R_j^{1,1})$, the flow capacity of aggregated resource $R_j^{1,1}$ with regard to aggregated task $T_i^{1,1}$.

Different cases are to be considered depending on the topology of the task break up graph. It is assumed that the task break up graph is an event graph free of circuit, without loss of applicability for most of manufacturing processes.

6.1.1 Elementary schemes

Case 1 (sequence)

Let $T_x^{1,1}$ be the aggregation of the following sequence :

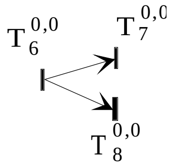


the flow capacity of $R_1^{1,1}$ with regard to $T_x^{1,1}$ is then :

$$\alpha_1^{2,1}(T_x^{1,1} / R_1^{1,1}) = \gamma_1^{1,0}(T_1^{0,0} ; T_2^{0,0}) = 2$$

Case 2 (disassembly):

$T_2^{1,1}$ being the aggregation of :



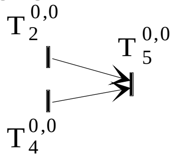
then :

$\alpha_1^{2,1}(T_2^{1,1}/R_1^{1,1})= Min\{(\gamma_2^{1,0}(T_6^{0,0} ; T_7^{0,0}) + \gamma_2^{1,0}(T_6^{0,0} ; T_8^{0,0})) ; \delta_2^{1,0}(T_6^{0,0}/ R_2^{1,1})\}$

$\alpha_1^{2,1}(T_2^{1,1}/R_1^{1,1})= Min\{(5+4) , 8\} = 8$

Case 3 (assembly) :

$T_Z^{1,1}$ being the aggregation of :



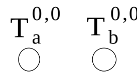
then :

$\alpha_1^{2,1}(T_Z^{1,1}/R_1^{1,1})=$
$Min\{(\gamma_2^{1,0}(T_2^{0,0} ; T_5^{0,0})+ \gamma_2^{1,0}(T_4^{0,0} ; T_5^{0,0})) ; \delta_2^{1,0}(T_5^{0,0}/ R_1^{1,1})\}$

$\alpha_1^{2,1}(T_Z^{1,1}/R_1^{1,1})= Min\{(5+3) , 6\} = 6$

Case 4 (independent)

It might also be that an aggregated $T_q^{1,1}$ is composed of independent tasks :



Tasks $T_a^{0,0}$ and $T_b^{0,0}$ are then performed concurrently.

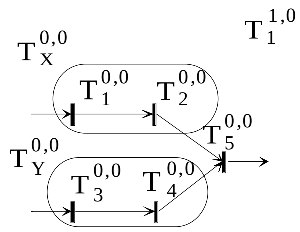
Then :

$\alpha_s^{2,1}(T_q^{1,1}/R_s^{1,1})$
$= Min\{\delta_s^{1,0}(T_a^{0,0}/ R_s^{1,1}), \delta_s^{1,0}(T_b^{0,0}/ R_s^{1,1})\}$

6.1.2 Generalization

The task break up graph is supposedly an event graph, which means that the manufacturing process is deterministic. The task break up graph is split into elementary schemes as identified in § 6.1.1. Then the aggregated flow capacity is calculated recursively.

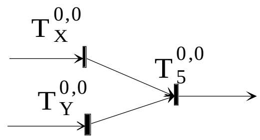
Let the following sub-graph, a part of $T_1^{1,0}$, be considered.



Here it is shown how to compute the throughput of $R_1^{1,1}$ with regard to $T_1^{1,1}$. Elementary schemes $T_x^{0,0}$ and $T_y^{0,0}$ can be identified, then according to §6.1.1, *case 1*:

$$\alpha_1^{2,1}(T_x^{1,1} / R_1^{1,1}) = 2$$
$$\alpha_1^{2,1}(T_y^{1,1} / R_1^{1,1}) = 5$$

so that the new situation is :



According to §6.1.1, *case 3*, it is clear that :

$$\alpha_1^{2,1}(T_1^{1,1}/R_1^{1,1}) =$$
$$Min\{(Min(\gamma^{1,0}\,(T_2^{0,0},T_5^{0,0}),\ \alpha_1^{2,1}(T_x^{1,1}/R_1^{1,1}))$$
$$+ Min\ (\gamma_1^{1,0}(T_4^{0,0},T_5^{0,0}),\ \alpha_1^{2,1}(T_y^{1,1}/R_1^{1,1})))\ ;\ \delta_1^{1,0}(T_5^{0,0}/R_1^{1,1})\}$$

$$\alpha_1^{2,1}(T_1^{1,1}/R_1^{1,1}) = Min\{(Min(5,2)+Min(3,5),\ 6\} = 5$$

Using this method for the complete example leads to the identification of matrix $A_1^{2,1}$:

$$\begin{array}{c|c} \boldsymbol{A_1^{2,1}} & T_1^{1,1}\ T_2^{1,1} \\ \hline \begin{array}{c} R_1^{1,1} \\ R_2^{1,1} \end{array} & \begin{pmatrix} 5 & 0 \\ 0 & 8 \end{pmatrix} \end{array}$$

6.2. Calculus of B matrices

Once the capacity of each aggregated resource with regard to the different aggregated tasks has been identified, the purpose is to take into account the capacity of transport channels between the aggregated resources.

Let first matrix $B_i^{v+1,v-1}$ be considered and depicted by blocks.

The blocks situated on the diagonal of $B_i^{v+1,v-1}$ are :

$$B_j^{v,v-1} \quad j = 1, \dots H_i^{v+1,v}$$

Note that the entries of $B_j^{v,v-1}$ are the capacities of the routes within $R_j^{v,v}$.

The entries of the blocks not situated on the diagonal of $B_i^{v+1,v-1}$ are the capacities of the routes between resources that belong to different *v*-level resources within $B_i^{v+1,v}$.

In the case shown on Figure 3, matrix $B_1^{2,0}$ is *:*

$$\begin{array}{c|c} B_1^{2,0} & R_1^{0,0}R_2^{0,0}R_3^{0,0}R_4^{0,0}R_5^{0,0}R_6^{0,0}R_7^{0,0}R_8^{0,0} \\ \hline \begin{array}{c} R_1^{0,0} \\ R_2^{0,0} \\ R_3^{0,0} \\ R_4^{0,0} \\ R_5^{0,0} \\ R_6^{0,0} \\ R_7^{0,0} \\ R_8^{0,0} \end{array} & \left(\begin{array}{ccccc|ccc} 24 & 0 & 0 & 0 & 0 & 8 & 0 & 0 \\ 6 & 15 & 0 & 3 & 5 & 0 & 0 & 0 \\ 7 & 5 & 30 & 2 & 0 & 0 & 0 & 0 \\ 5 & 8 & 4 & 10 & 7 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 20 & 0 & 0 & 0 \\ \hline 8 & 0 & 0 & 0 & 0 & 10 & 7 & 5 \\ 0 & 0 & 0 & 0 & 0 & 5 & 20 & 6 \\ 0 & 0 & 0 & 0 & 0 & 4 & 6 & 16 \end{array}\right) \end{array}$$

Each block has now to be reduced to a scalar, according to an exact (Ford-Fulkerson algorithm)[6] or any approximate operator of aggregation, *ad lib.* selected by the expert.

Having for instance selected the *average*-operator, matrix $B_1^{2,0}$ is as follows :

$$\begin{array}{c|c} \boldsymbol{B_1^{2,1}} & R_1^{1,1}\ \ R_2^{1,1} \\ \hline \begin{array}{c} R_1^{1,1} \\ R_2^{1,1} \end{array} & \begin{pmatrix} 6{,}28 & 8 \\ 8 & 8{,}7 \end{pmatrix} \end{array}$$

At this point, one complete step of aggregation has been made and the process can be re-iterated if necessary.

7. Conclusion

The aggregation framework presented in this paper is generic and supports the calculation of the capacity of complex organizations. The relevance of the aggregation operator chosen to reduce information from a level to another is let to the appreciation and experience of the user. Rather than to search for generic aggregation operators supposed to be valid in any application case, which seems nonrealistic, the approach merely provides a methodology and a technical data model to facilitate the macro-analysis of complex productive organizations, whatever the aggregation operators are.

References

[1] Jagdev H.S., Browne J., “ The extended Enterprise - a context for manufacturing, Production Planning and Control ”, Vol . 9, n°3, pp 216, 229, 1998.

[2] S.B. Gershwin, “ Hierarchical flow control : a framework for scheduling and planning discrete events manufacturing systems ”, Proc. of the IEEE, Vol 77, n°1, January 1989

[3] S.Axaster, “ On the feasibility of aggregated production plan ”, Operation Research, vol.34, n°5, September-October 1986.

[4] D. Chen, P. Farthouat, F. Pereyrol, J.P. Bourrieres “A generic model for control of hierarchical multi-level production systems” Proc. of the IFAC workshop on Manufacturing systems : modelling, management and control, MIM'97 pp. 227-232, Feb. 3-5 1997, Vienna

[5] M.Zolghadri, J.P.Bourrieres “ A Data Aggregation Framework for Multi-level Production System control” IEEE Int.Conf. on Systems, Man, and Cybernetics Intelligent Systems For Humans In A Cyberworld October 11-14, 1998, San Diego

[6] Ford L.R., Fulkerson D. R. Maximal flow through a network, Canad. J. of Math., p. 399, 1956.