A SCHEDULING MODEL FOR REPETITIVE ACTIVITIES

Mohammed Al Joma[1] and Jean-Claude Mangin[2]

Laboratoire d'Optimisation de la Conception et Ingénierie de l'Environnement(LOCIE), Ecole Supérieure d'InGEnieurs de Chambéry (ESIGEC), Université de Savoie, 73376 Le Bourget du Lac, France

Construction management engineers are often faced with projects containing multiple units wherein activities repeat from unit to another. The repetitive activities, which can be seen clearly in multi-storey buildings, require schedules that ensure the uninterrupted usage of resources for an activity in one unit to a similar activity in the next unit. The repetitive and non-repetitive scheduling model described in this paper recognizes the constraints between the activities and includes also an additional resource continuity constraint to ensure continuous resource usage. It's applied to both vertical and horizontal projects containing either discrete or continuous activities. The scenarios stocked in the model will generate automatically the majority of these constraints and therefore the validation point between two successive activities is located and checked. For each activity the model validates the constraints with the precedent activities. The decision of the early time date of the sequence activity is taken when all the constraints are satisfied. The planning of the whole project can finally be obtained.

Keywords: planning scheduling, productivity, project duration, repetitive activity.

INTRODUCTION

Construction management engineers are often faced with projects that contain several or similar units, such as stories in high-rise buildings. These multi-unit projects are identified by their repetitive activities in which individual identical activities are repeated consecutively from unit to another. Scheduling the activities of projects should allow the work continuity to enable the repetitive activities to finish work on one unit and then move to the next without dead time. These activities advance into two directions: the horizontal direction within the same floor and the vertical direction in the superposed floors. Repetitive activities have two types; the upward activities, which follow the construction logic, from lower storey to the higher one, and the opposite behaviour, which is the downward activity, which proceeds from higher to lower storey. Two types of constraints between activities control their sequences, the succession constraints, which are applied to activities having cross zones and the imposed constraints described by imposed delays between activities.

Construction activities have frequently a repetitive nature; they are under the microscope of research since the end of Sixties. The numerous techniques available for planning construction processes are based on graphical, networking or analytical techniques. Some of them were based on linear scheduling methods as Line Of

[1] Mohammed.Al-Joma@univ-savoie.fr
[2] Jean-Claude.Mangin@univ-savoie.fr

Al-Joma, M and Mangin, J-C (2004) A scheduling model for repetitive activities. *In:* Khosrowshahi, F (Ed.), *20th Annual ARCOM Conference*, 1-3 September 2004, Heriot Watt University. Association of Researchers in Construction Management, Vol. 2, 1197-208.

Balance (LOB) (O'Brien 1975, Carr and Meyer 1974; Halpin and Woodhead 1979)or on both CPM and PERT methods, which have also seen the light in the same time (J-C Mangin 1979). Some other works continue later using the LOB method (Lutz and Hijazi 1993, Mends & Heineck 1998) or using CPM and LOB methods (Harris 1998).

In spite of these various methods which have been developed with their own particular targets, all of them have one shared goal: to schedule the project and validate the constraints between activities (Thabet and Beliveau 1994) and draw up their progress in time. The model proposed here is able to construct a complete schedule as the LOB method but this model is based on a graphical representation close to a bar chart representation GANTT which is often used, needed and well understood by practitioners.

OBJECTIVE

The first stage of this research is to develop, with a purpose of investigating the use of these methods, a general and simple model, which responds to the needs of repetitive and non-repetitive activities. Firstly, the model was developed with the purpose of making a planning for a project using a list of stocked in cases. They consider all possible constraints in order to calculate the project duration. Secondly, the model takes into account different types of repetitive activities that have either upward or downward behaviour as well as non-repetitive activities. Finally, it describes and validates imposed and succession constraints that could join the activities in multi-storey buildings projects. The model is able to construct a project scheduling by satisfying all the constraints and by calculating the start time of each activity.

ACTIVITY PATTERNS

The different types of activities are listed and stocked in the model. Two main types are considered: repetitive and non-repetitive activities. A non-repetitive activity is an individual or unique activity, which is done in one unit of work. It is the basic element of repetitive activity types. Repetitive works in multi-storey buildings form chains of activities. Each activity chain poses a spatial orientation of workflow in the vertical direction. The direction of flow is either upward or downward. Most construction works are scheduled in an upward direction to follow the erection of the building. Some of these activities, however, are scheduled in a downward direction for some reasons such as safety or to prevent damage of the realised work. Finishing works like external cladding, clearing, and cleaning, may be typical examples of such types of activities. An other type that should be presented is the intermittent form of repetitive activities. For some reasons, the activity can be divided into some intermittent activities. These blocks of activities are connected with specific constraints. These activities are turned to the forms presented before and thus, the model can identify all the activity forms in order to validate the constraints.

SCHEDULING OF MULTI-STOREY ACTIVITIES

Repetitive activities such as multi-storey buildings allow working with rhythmic planning. A same resource is used by several activities. This is why the optimum use of resource is a very important need. In order to maintain the work teams continuity, repetitive units must be scheduled to enable timely movements of teams from one unit to the next, avoiding work team delay time. This is known as work continuity constraint. This constraint always works with other constraints such as unit's constraints and activity's constraints.

Logic unit constraints control the logic sequence of the activities in these units. These constraints are defined by precedence links between the activities on each unit, as established in the typical unit logic network. Based on such constraints, an activity can not be scheduled until all its preceding activities are scheduled. These constraints are a result of many factors, such as technological dependencies between activities, weather constraints, imposed dates, safety constraints, and so on. An example of such constraint is the technological constraint, which needs the completion of the prior storey columns and beams before finishing the slab for this storey. The second type of logic constraint is the teams work continuity. The activities have a logic sequence unit, which is not far from the team's work continuity. This constraint need the promptly movement of the team's works without interruption from one unit to the next in order to minimize its idle time.

TEAMSWORK CONTINUITY AND WORKFLOWRATE

The contractors need to solve teamwork continuity difficulties. Teamwork must move from one unit to another without dead time or interruption in order to avoid several arrivals to the building site. That supposes a simple adaptation of workflow rate. For two repetitive activities A and B presented in (Figure 1-a), activity B is interrupted

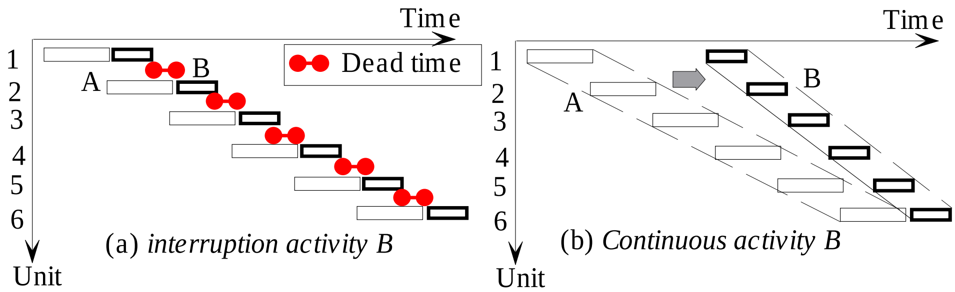

Figure 1 : Teamwork Continuity

several times because it has a lower rate of realisation than activity A. In order to avoid these dead time, individual activities of B should be delayed (Figure 1-b). The activity duration is given by equation (1):

$$d_A = \frac{\mathrm{Q_{Ar}}}{\mathrm{U_A} \times \mathrm{QR_{Ar}}} \qquad (1)$$

d_A: duration of individual activity A; $\mathrm{Q_{Ar}}$: quantity of activity A realised with resource r; $\mathrm{QR_{Ar}}$: quantity of resource needed for activity A; $\mathrm{U_A}$: unit rate of activity A. The workflow rate for an activity is given by the equation (2):

$$C_A = \frac{d_A}{q_A} \qquad (2)$$

C_A: workflow rate of the activity A; q_A : number of teams for activity A.

The workflow rate can be used in order to adjust the rhythm of activity duration. In fact, it depends on the number of teams or resource needed for an activity. These resources determine the activity duration. For example, increasing the team's number of an activity reduces its duration, which is given by equation (3):

$$D_A = |j - i| \times C_A + d_A \qquad (3)$$

D_A: repetitive activity duration, i: start unit of A; j: finish unit of A, (Figure 2). Manipulating the workflow rate minimizes or cancels a time lag between two activities (Figure 2,b,c,d).

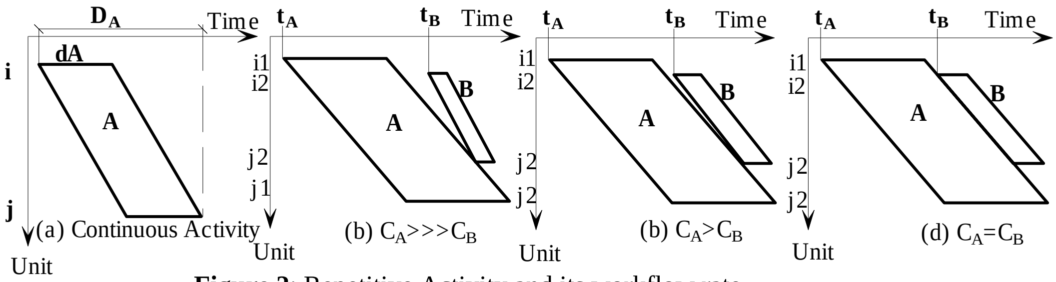

Figure 2: Repetitive Activity and its workflow rate

The additional teams hold out two ways of scheduling: (1) altering teams; (2) or non-altering teams (Figure 3). In an altering teams scheduling, a team finishes his work in

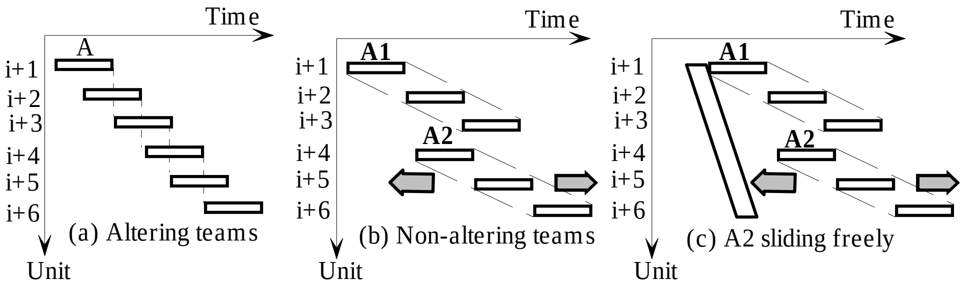

Figure 3: Altering and non-altering teams

unit number i and moves to unit $i+q$, where q is the number of teams (Figure 3-a, two teams affected for the activity A). In some other scheduling, the teams are affected to finish a block of units. For example, the first team finishes the units (i+1,i+2,i+3), the second one for the (i+4, i+5, i+6), etc (Figure 3-b). In the second type of scheduling, the activities A1 and A2 are slided freely along the time with the constraint of the precedent activities (Figure 3-c).

VALIDATION OF CONSTRAINTS

Succession and imposed constraints should be satisfied in order to make a correct planning. A succession constraint is related with the sequence of activities which is linked to the work teams availability and the possibility of delaying the next activity in order to avoid finding two activities in the same unit at the same time. An imposed constraint is a constraint between two particular individual activities in different units. As a result, the constraints are checked in 5 cases in this model. They are the heart of our model.

VALIDATION OF SUCCESSION CONSTRAINTS

The satisfaction of a succession constraint for an activity consist in checking its starting date that must be equal or greater than the completion dates of all preceding activities on every typical units. But, the validation of the activity doesn't need, in fact, to be made for every unit. The validation will be done for a particular unit, named validation point or critical unit. In fact, the validation of the constraint for this unit will insure the constraint validation for all other units.

The validation point is determined by the two activities to be checked, their continuity, their duration, their workflow direction and the units where they will be done. The validation of succession constraints will be done by five cases, which are adopted in the model. The cases from one to four present the determination of validation point when the two activities are repetitive. On the other hand, the fifth case is used when one of the two activities is individual.

In order to calculate the next activity starting date, the constraint between two successive activities A and B is given by the following equation (4) to be checked:

$$t_B - t_A \geq \delta_{AB} \qquad (4)$$

t_A : The starting date of activity A; t_B : The starting date of activity B; δ_{AB} : The minimum time lag between the two starting times of A and B.

Case 1:
The first case validates a succession constraint between two activities upwards directed. Four sub-cases are detected according to activities positions (Figure 4)

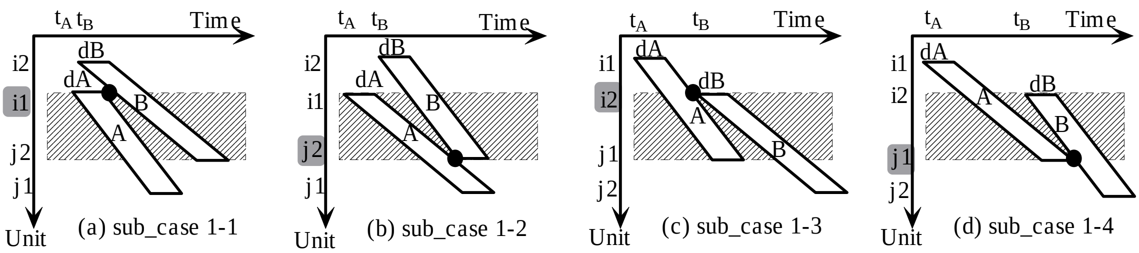

Figure 4: Validation succession constraint between two upward activities

$i1, j1$: start and finish units of activity A; $i2, j2$: start and finish units of activity B; $j1 > i1$: describes an upward activity (type 1); $j1 < i1$: describes an downward activity (type –1); $j1 = i1$: describes an individual activity (type 0).

Table 1: Sub-cases for upwards activities

	(Sub-case 1-1)		**(Sub-case 1-2)**	
Activity	A	B	A	B
Type	1	1	1	1
Workflow rate	$C_A \leq C_B$		$C_A \geq C_B$	
Units Condition	$i_1 \geq i_2$		$j_1 \geq j_2$	
Critical unit	i_1		j_2	
Equation	$\delta_{AB} = d_A - (i_1 - i_2)C_B$ (5)		$\delta_{AB} = (j_2 - i_1)C_A + d_A - (j_2 - i_2)C_B$ (6)	
	(Sub-case 1-3)		**(Sub-case 1-4)**	
Activity	A	B	A	B
Type	1	1	1	1
Workflow rate	$C_A \leq C_B$		$C_A \geq C_B$	
Units Condition	$i_1 < i_2$		$j_1 < j_2$	
Critical unit	i_2		j_1	
Equation	$\delta_{AB} = (i_2 - i_1)C_A + d_A$ (7)		$\delta_{AB} = (j_1 - i_1)C_A + d_A - (j_1 - i_2)C_B$ (8)	

Case 2:

The second case appears when the two activities have different types. The activity A is upwards and the B is downwards. The workflow rate has no influence on the determination of the validation point. This case can be decomposed into two sub cases. (Figure 5).

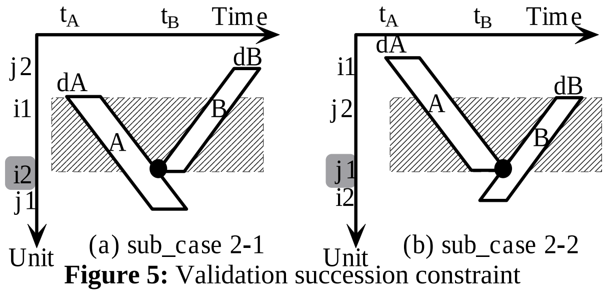

Figure 5: Validation succession constraint between upwards and downwards activities

Table 2: Sub-cases for upwards and downwards activities

	(Sub-case 2-1)		(Sub-case 2-2)	
Activity	A	B	A	B
Type	1	-1	1	-1
Workflow rate	$C_A \geq C_B$ ou $C_A \leq C_B$		$C_A \geq C_B$ ou $C_A \leq C_B$	
Units Condition	$j_1 \geq i_2$		$j_1 < i_2$	
Critical unit	i_2		j_1	
Equation	$\delta_{AB} = (i_2 - i_1)C_A + d_A$ (9)		$\delta_{AB} = (j_1 - i_1)C_A + d_A - (i_2 - j_1)C_B$ (10)	

Case 3:

It is the same as case 2; it is reserved for two activities having different types, but activity A is upwards directed and activity B is downwards directed. The workflow rate has no influence . Two sub cases are detected according to activities positions (Figure 6).

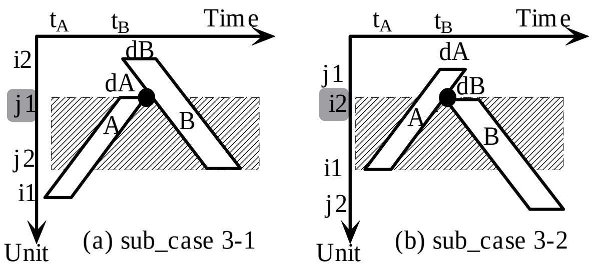

Figure 6: Validation succession constraint between downwards and upwards activities

Table 3: Sub-cases for downwards and upwards activities

	(Sub-case 3-1)		(Sub-case 3-2	
Activity	A	B	A	B
Type	-1	1	-1	1
Workflow rate	$C_A \geq C_B$ ou $C_A \leq C_B$		$C_A \geq C_B$ ou $C_A \leq C_B$	
Units Condition	$j_1 \geq i_2$		$j_1 < i_2$	
Critical unit	j_1		i_2	
Equation	$\delta_{AB} = (i_1 - j_1)C_A + d_A - (j_1 - i_2)C_B$ (11)		$\delta_{AB} = (i_1 - i_2)C_A + d_A$ (12)	

Case 4:

The two activities have a downwards type. The workflow rate and the relative positions between activities have an influence on the determination of starting date. Four cases are established(Figure 7).

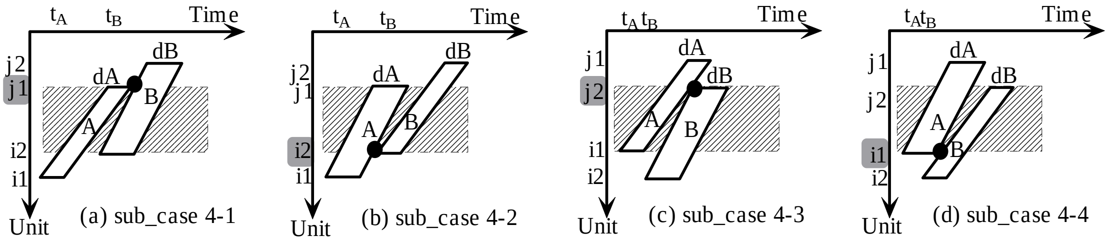

Figure 7: Validation succession constraint between two downward activities

Table 4: Sub-cases for downwards activities

	(Sub-case 4-1)		(Sub-case 4-2)	
Activity	A	B	A	B
Type	-1	-1	-1	-1
Workflow rate	$C_A \geq C_B$		$C_A \leq C_B$	
Units Condition	$j_1 \geq j_2$		$i_1 \geq i_2$	
Critical unit	j_1		i_2	
Equation	$\delta_{AB} = (i_1 - j_1)C_A + d_A - (i_2 - j_1)C_B$ (13)		$\delta_{AB} = (i_1 - i_2)C_A + d_A$ (14)	
	(Sub-case 4-3)		**(Sub-case 4-4)**	
Activity	A	B	A	B
Type	-1	-1	-1	-1
Workflow rate	$C_A \geq C_B$		$C_A \leq C_B$	
Units Condition	$j_1 < j_2$		$i_1 < i_2$	
Critical unit	j_2		i_1	
Equation	$\delta_{AB} = (i_1 - j_2)C_A + d_A - (i_2 - j_2)C_B$ (15)		$\delta_{AB} = d_A - (i_2 - i_1)C_B$ (16)	

Case 5:

The last case of succession constraints appears when one of the two activities is not repetitive. One of the activities is individual and the other one can be either upwards or downwards oriented, or can be individual. Since one of the activities is individual, it is realised in a unique unit and the critical unit will be this unit. It should be mentioned that the workflow rate has non-influence in this case. Five sub-cases can explain this case (Figure 8)

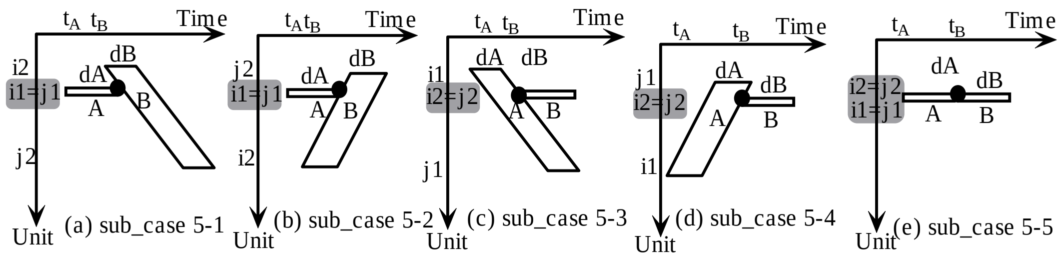

Figure 8: constraint between individuals and upwards or downwards activities

Table 5: Sub-cases for upwards or downwards and individuals activities

	(Sub-case 5-1)		(Sub-case 5-2)	
Activity	A	B	A	B
Type	0	1	0	1
Workflow rate	$C_A \leq C_B$ ou $C_A \geq C_B$		$C_A \leq C_B$ ou $C_A \geq C_B$	
Units Condition	$i_2 \leq i_1 \leq j_2 ; i_1 = j_1$		$i_2 \leq i_1 \leq j_2 ; i_1 = j_1$	
Critical unit	i_1		i_1	

Equation	$\delta_{AB} = d_A - (i_1 - i_2)C_B$ (17)	$\delta_{AB} = d_A - (i_2 - i_1)C_B$ (18)

	(Sub-case 5-3)		**(Sub-case 5-4)**	
Activity	A	B	A	B
Type	1	0	-1	0
Workflow rate	$C_A \leq C_B$ ou $C_A \geq C_B$		$C_A \leq C_B$ ou $C_A \geq C_B$	
Units Condition	$i_1 < i_2 < j_1; i_2 = j_2$		$i_1 < i_2 < j_1; i_2 = j_2$	
Critical unit	i_2		i_2	
Equation	$\delta_{AB} = (i_2 - i_1)C_A + d_A$ (19)		$\delta_{AB} = (i_1 - i_2)C_A + d_A$ (20)	

	(Sub-case 5-5)	
Activity	A	B
Type	0	0
Workflow rate	$C_A \leq C_B$ ou $C_A \geq C_B$	
Units Condition	$i_1 = i_2 = j_1 = j_2$	
Critical unit	i_1	
Equation	$\delta_{AB} = d_A$ (21)	

VALIDATION OF THE IMPOSED CONSTRAINTS

Similarly to the succession constraints validation, the model can validate the imposed constraints. For each activity, the validation point is located according to the constraint that links two units together and is checked in result. This constraint represents the delay between two individual activities; it can be seen in four kinds: finish-finish (ff), finish-start (fs), start-finish (sf) and start-start (ss).

The constraint joins the unit z in activity A with the unit y of activity B. The equations needed in order to validate the constraint and therefore the starting date of the following activity B are:

Finish-start (fs): $t_B - t_A \geq (z-1)C_A + d_A + fs - (y-1)C_B$ (22)

Start-finish (sf): $t_B - t_A \geq (z-1)C_A + sf - (y-1)C_B - d_B$ (23)

Start-start (ss): $t_B - t_A \geq (z-1)C_A + ss - (y-1)C_B$ (24)

Finish-finish (ff): $t_B - t_A \geq (z-1)C_A + d_A + ff - (y-1)C_B - d_B$ (25)

These equations are used to validate the imposing constraints whatever the types of the two activities are. $z = 1$ or $y = 1$ when the activity unit is the start one and $z = |j_1 - i_1| + 1$ or $y = |j_2 - i_2| + 1$ when it is the last one. Figure 9 presents the different types of activities that are connected with imposed constraints.

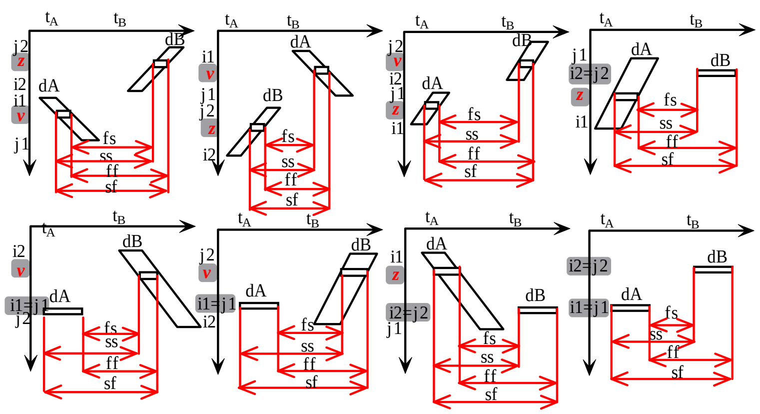

Figure 9: The imposing constraint between the activities

SCHEDULING CALCULATION

In order to have the start time of a given activity, all the constraints with the precedent activities must be satisfied. The validation of these constraints includes the both types of constraints and two possible starting times of the concerned activity are calculated. The next step consists in selecting the maximum starting time between them. The following example explains the validation of constraints and the activity start time calculation (Figure 10). For a multi-storey building, the activity A is divided

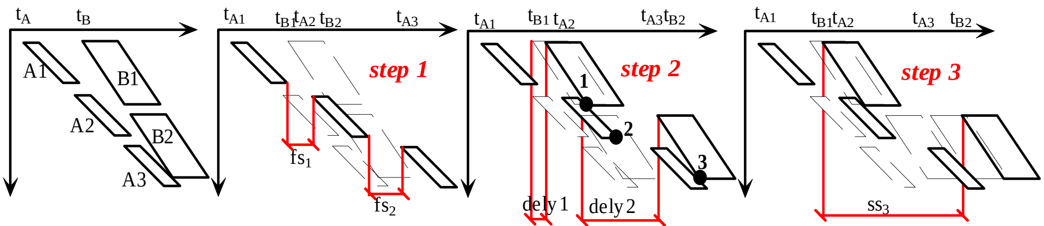

Figure 10: Validation the constraint by delaying the activities

into three parts (A1, A2, A3). These activities are now considered as three activities linked together with constraints to be validated. In consequence, their starting times should be calculated. They are connected to each other by constraints: fs_1 between A1 and A2 and fs_2 between A2 and A3.

At the first step, in order to validate fs_1, which is the first imposed constraint, the activity A2 should be delayed from its initial position. For the constraint fs_2, the second imposed constraint, activity A3 is also delayed and the activity start times are:

$$t_{A2} = t_{A1} + |j_{A1} - i_{A1}| C_{A1} + d_{A1} + fs_1 \tag{26}$$

$$t_{A3} = t_{A2} + |j_{A2} - i_{A2}| C_{A3} + d_{A2} + fs_2 \tag{27}$$

The second step consists in validating the succession constraints between the activities A and B. The activity B1 should be delayed one time to validate the point number 1 by sub-case 1-2, but B2 should be delayed two times: (1) to validate the validation point 2 with A2 by sub-case 1-4, and (2) to validate the validation point 3, sub-case 1-2, with A3. Activities start times will be:

$$t_{B1} = t_{A2} + (j_{B1} - i_{A2})C_{A2} + d_{A2} - (j_{B1} - i_{B1})C_{B1} \tag{28}$$

$$(t_{B2})_1 = t_{A2} + (j_{A2} - i_{A2})C_{A2} + d_{A2} - (j_{A2} - i_{B2})C_{B2} \tag{29}$$

$$(t_{B2})_2 = t_{A3} + (j_{B2} - i_{A3})C_{A3} + d_{A3} - (j_{B2} - i_{B2})C_{B2} \tag{30}$$

But the two parts of B are linked together with ss constraint. The third step needs the delaying of B2 to validate this constraint. $(t_{B2})_3 = t_{B1} + dd$ (31)

After comparing the three values of t_{B2} the maximum is picked up.

CONCLUSION

The repetitive and non-repetitive scheduling model has been developed in order to satisfy all requirements for a schedule with repetitive and non-repetitive activities in multi-storey buildings. This model can be used for construction activities as well as finishing activities that have several forms. The sequence of activities is controlled by the scenarios, which are defined and stocked in a knowledge base of the model in order to use it later.

Succession constraints, which describe relations between activities partially or totally done in the same units. Imposing constraints are used to link two activities in different or same units, are checked and validated.

Although the manipulation of the workflow rate can minimize the project duration, the use of a method of optimization, like genetic algorithms, is necessary for project scheduling optimization.

REFERENCES

Igusti Agung Adnyana Putera, S (2002) Génération et évaluation des activités de construction des bâtiments Prise en compte des travaux des tâches répétitives : Optimisation de la planification par les algorithmes génétiques, Diplôme de Docteur, LOCIE, Université de Savoie, France.

James D Lutz, Adib Hijazi (1993) Planning repetitive construction: Current practice. Construction Management and Economics, 11, 99-110.

Jean-Claude Mangin, S (1979) Prise en compte des travaux des tâches répétitives dans l’ordonnancement des travaux de bâtiment, le grade de docteur d’état es-sciences, l’Institut National des Sciences Appliquées de Lyon et l’Université Claude-Brenard Lyon I, France.

Johnston D W (1981) Linear scheduling method for highway construction. Journal of the Construction Division, 2, 241–261.

Khaled El_Rayes (2001) Object-Oriented Model For Repetitive Construction Scheduling. Journal of Construction and Management, 127(3), 199-205.

Khaled El_Rayes, Osama Moselhi (1998) Resource-driven scheduling of repetitive activities. Construction Management and Economics, 16, 433-446.

Ricardo Mendes Jr & Luiz Fernando M Heineck (1998) Preplanning Method for Multi-Storey Building Construction using Line of Balance. In: Guarujá, Brazil, 6th Annual Conference of the International Group for Lean Construction, 13-15 August 1998.

Robert B Harris, Photios G Ioannou (1998), Scheduling Projects With Repeating Activities. Journal of Construction Engineering and Management, 124(4), 269-278.

Russell A D, Caselton W F (1988), Extensions to linear scheduling optimization. Journal of Construction Engineering. and Management, 1, 36–52.

Walid Y Thabet, Yvan J Beliveau (1994) HVLS: Horizontal and Vertical Logic Scheduling for Multi-Storey Projects. Journal of Construction Engineering and Management, 120(4), 875-892.

O’Brien, J J (1975) VPM Scheduling for high-rise building. Journal of the Construction Division, 4, 895-905.

Carr R I, Meyer W L (1974) Planning construction of repetitive building units. Journal of the Construction Division, 3, 403–412.