Algorithm of decision-making aids To optimize the Scheduling of the tasks of maintenance in real time

Aissam Aboussalim [1], Mohamed Medioui [2], Ouafae Bennis [3]

[1,2] Industrial data processing department, Ensa Agadir, Laboratory of the Industrial engineering and data-processing
email: aboussalimaissam@hotmail.fr

[3] *Institut PRISME, Orleans University,*
email : ouafae.bennis@univ-orleans.fr

Abstract - The approach presented in this article is a contribution to research relating to the optimization of the scheduling of the spots of maintenance.
This work answers the following objectives: Decision in real-time(planning and scheduling), Allowance/assignment of resources, to ensure a minimum availability required, to minimize times delay while inserting the dynamic tasks `corrective maintenance_

Keywords: tasks, preventive maintenance, scheduling, optimization, resource, cost

1. INTRODUCTION

The problem of scheduling consists in organizing in time the realization of the tasks, taking into account temporal constraints and constraints relating to the use and the availability of necessary resources. The optimization of the scheduling of the tasks of maintenance became an essential field of research. It contributes to help the companies to show a profit better with their systems of production.
We are interested more particularly in the dynamic strategies of maintenance for which decisions (to maintain, repair, replace, inspect) are taken on the basis of information of monitoring on line
The approach presented in this work is intended to assign resources to tasks to reduce, on a H horizon, the costs of delays or deviations, proposes an algorithm to sort, insert dynamic tasks in real-time, providing an affin user interface prioritize tasks in real time and allow the maximum use lowest lost cost resources.

2. POSITION OF THE PROBLEM

We position in the case of the preventive maintenance, the dates of intervention and the durations of tasks are calculated on the basis of statistical analysis of the breakdowns. A team of maintenance must have a certain qualification level to be able to maintain the equipment and to carry out the task in her optimal time.[1].
On the level of the organization of the system, we adopt the following structure: a center of competence, which evaluates human resources and the class by assessment of competence for one given period.

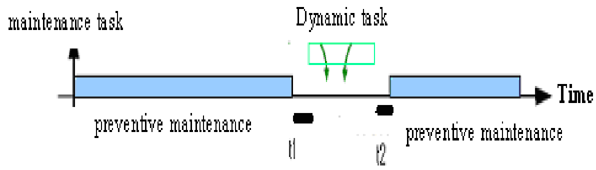


FIG1: Insert of the dynamic spot in the elementary window

Function to be optimized
The cost of the activities of maintenance is the sum of the costs of all the tasks and the costs lost between spots which one can call window. The lost cost is the product of the time of displacement and the one hour cost of displacement:

$C_{k\ lost} = [T_{K+1} - T_K] \times C_{hour}$

T_K : Execution time of the task K

The total function is written like:

$$F_{OG} = \sqrt{C_{Ti}}_{\,i} + \sqrt{C_{k\ perdu}}_{\,k}, \quad [2]$$

$$F_{opti\_tâches} = \sum_i C_{Ti} = \min \sum_{l=1} W_i\theta_i + h_i\theta_{i-} + C_{i0} \quad [2]$$

This function decides for each task to deviate positively or negatively respectively compared to its date which had as soon as possible or its date which had at the latest, that is to say to start in time_
Each task I is characterized by one noted operational duration Ti
I: indicate the negative deviation of task I compared to its which had date as soon as possible or also the advance.
i+: indicate the positive deviation of task I compared to its date which had at the latest or also the delay.
A Wi late penalty, a penalty in advance hi, Ci0 the minimal cost of preventive maintenance
The cost of the activities of maintenance is the sum of the costs of the tasks added to the costs lost between spots k+1et K

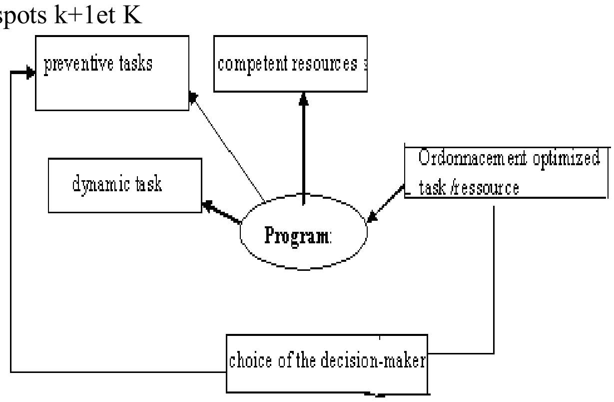


fig2: template assignment dynamic scheduling a new and task

3. SOLUTION SUGGESTED

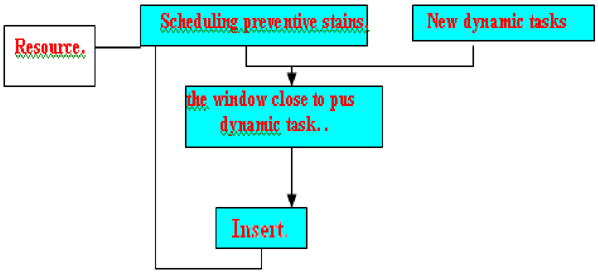


Fig3 schema of principle optmezation ordonnancement stains

The solution suggested receives in entry a list *L* containing the tasks of maintenance with their characteristics (*laughed, di, Ci*). *laughed:* go back beginning to task (*laughed*), *di:* go back fine to task *Ci* is the cost of the task: [3]

Algorithm

To affect Task (Ti) // Ti : task i
begin
if (P=∅) then // P : Piled of the task
Pile ← Ti // first of it piled up
else go_ end_ Piled (Piled)
P ← Ti // Ti is added at the end of the task
End if
End
Affect_Piled_Resource (Ri)
// Ri : resource i
begin
If (R〈〉 ∅)
then go _End _Resource (R)
// R :Piled up des resources
R ← Ri
Endif
End
Select_Resource (R)
// R : resources Piled. A resource is characterized by a note and an available variable
Begin
Ri ← first _resource
While (Ri.disponible = False)
Ri ← Resource_next
End while
If (Ri .disponible = true) then
Affect_Resource_task (Ri , P)
else Write "not available resource".
End if
End
Affect_Resource_task (Ri, piled P)
//Ri : The most competent resource available
Variable P_gT : task
// P_gT the greatest task
Variable h ,pos,N
//h counter of piled P // N number on task
// pos the position is task of piled P. di et ri are the task settings (first and end)

Sort _tasks ( )
h ← 2
pos ← 1
P_gT ← T_1. di - T_1. ri
begin
While (h Ħ N)
if ((T_h. d_h ˉTh. r_h) 〉 P_gT) then
P_gT ← T_h
Pos ← h
end if
h ← h+1
end while
Ri .disponible ← False
end
// Available is a setting that is true if the resource is not taken, otherwise itˇs false
Lost_cost_Calculus ( ) // Downtime
Begin
Somme=0
for i=1 to N do
Somme=somme + T_{i+1}.d-T_i.r
end for
// print « coût perdu = », somme*Ci0
Lost_cost_Calculus ← somme*Ci0
end
F_before ← Lost_cost_Calculus ( )
After optimization // inserting dynamic task
Inser_task_D (Td ; P)
begin
Var X, T : task
Var : Duree _TD, pos: integer
Duree _Td ← Td.di -Td. Ri
T ← first _ task (Piled)
Pos ← 1
i ← 1
While (T < > end_ piled)
X ← Task _ next (Piled)
Min ← X.ri ˉ T.di
if (Min > = Durée _ Td) then
Pos ← i
T ← X
X ← Task _next (piled)
if (Min > = (X. ri ˉ T. di)) and (X. ri ˉ T. di) > Duree_TD)
Then Min ← X.ri ˉ T.di
T ← X
Pos ← i
Endif
Endif
End while // Scroll forward the task or insert
T ← first Task (piled)
for i = 1 to pos
T ← Task_ next (piled)
end for
Z ← T. next
T. next ← Td
Td. next ← Z .next
Lost_cost_Calculus // Downtime // After optimization

```
begin
Somme=0
for i=1 to N do
Somme=somme + T(i+1).d-T(i).r
end for
//print  lost cost =  somme*Ci0
 Lost_cost_Calculus       ←       somme*Ci0
 End
 F_after  ←   Lost_cost_Calculus ( )
 //after  inserting dynamic task
 // Tg ←      F_before -  F_after
```

SIMULATIONS RESULTS

The algorithm of tasks assignment was programmed by JAVA, , and the provision of the tasks resources by ms-Poject. To evaluate this program, we considered a whole of 10 tasks of preventive maintenance to schedule according to the resource and execution time. Initially 3 dynamic tasks were inserted, then 9 tasks.

Spots before modeling

The tasks are sorted by order ascending of the execution time .la the longest task is. Assigned to the most qualified resource .le program calculates the cost of each task and the cost lost between tasks

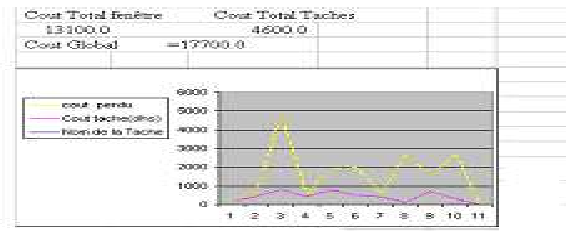
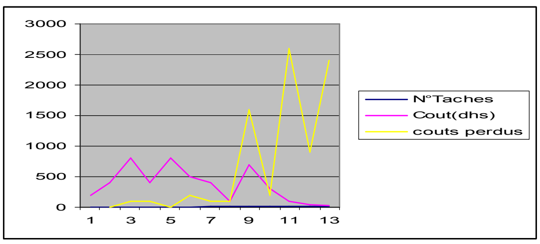


FIG4 Spots before modeling

Spots after modeling (optimization) insertion of 3 dynamic tasks:

According to the results, one notices that the most qualified resource is assigned to the preventive spot longest and it is this resource which is reallocated with the dynamic task longest. One notices according to the curve that the cost lost after optimization was attenuated of **37%.**



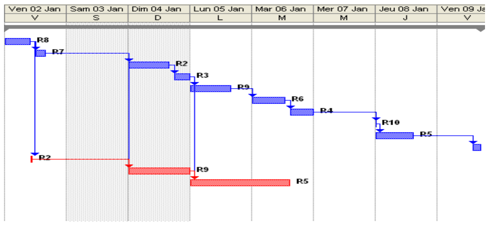
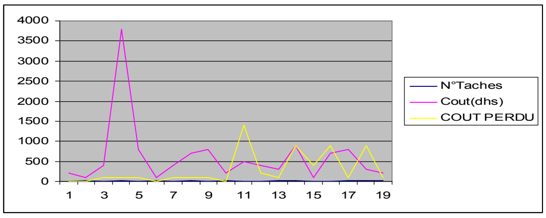


FIG5 Spots after modeling (optimization), insertion of 3 dynamic tasks

Spots after modeling (optimization) insertion of 9 dynamic tasks:

According to the results, one notices that the most qualified resource is affected with the preventive spot longest and it is this resource which is reallocated with the dynamic spot longest. One notices according to the curve that the cost lost after optimization was attenuated of 54%



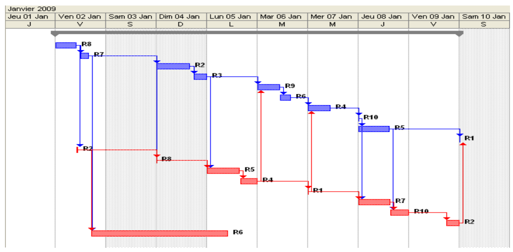


FIG6 Spots after modeling (optimization) , insertion of 9 dynamic tasks

4. CONCLUSION

This work deals with problem of optimization of the scheduling of the tasks and of resource allocation, which aims is the reduction of the lost cost. We presented an algorithm which can schedule N preventive tasks of maintenance, (n-1) dynamic tasks. This method showed good performances of optimization of the cost lost while inserting current task more and more. By comparing the 3courbes one notes that the lost cost falls while inserting dynamic spots more and more

5. REFERENCES

.[1] Djalal HEDJAZI 1, Abdelmadjid ZIDANI 2, Samra DEY3, Zouleikha TEJANI 4Algorithme d'arrête gauche adapté pour L'ordonnancement des tâches de maintenance dans un Contexte de maintenance distribuée « Conférence on Computer Integrated Manufacturing CIP'2007 »

[2] Baptiste P., Néron E., et Sourd F., (2004) «Modèles et Algorithmes en Ordonnancement», Groupe GOThA

[3] M. Rebai1, A. Kondo2 et I.Kacem3
Minimisation du coût de maintenance préventive sur une machine