ALGORITHM OF THE MAINTENANCE ACTIVITIES OPTIMIZATION WITH AN APPROXIMATE SOLUTION TO THE PROBLEM *OF SCHEDULING PREVENTIVE TASK*

Aissam Aboussalim [1],
Mohamed Mediouni [2],
[1,2] Industrial data processing department, Ensa Agadir, Laboratory of the Industrial engineering and data-processing
Email: aboussalimaissam@hotmail.fr

Abstract - Maintenance is a factor of competitivity since it affects the production, quality and cost. It is essential for the increasing development of the company and its assets.
The approach presented in this paper is a contribution to research related to the scheduling of maintenance tasks, to the allocation of resources and to the optimization of maintenance costs. This work proposes an algorithm based on: the real-time decision, the planning of maintenance tasks, the allocation of tasks to the appropriate resources under the constraints of the tasks duration and skills, and at the same time minimizing the cost of maintenance service.
The criterion for maintenance optimization is evaluated by the sum of advances / delays of preventive maintenance cost by including other tasks, dynamic or delays, in elementary windows

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

1. Introduction

The planning problem consists of organizing, in time, the tasks, taking into account the time constraints and the availability of resources. This programming is done to optimize certain criteria such as minimization of project dates, the dates for completing tasks, use of resources under constraints, of time, task priority, and availability of work material competences.To reduce maintenance costs, calibra's work [1] dealt with problems of reliability allocation. Stadje and Zuckerman [2] proposed a method based on queues for the repairing allocation of the broken machines. The works of Bennour et al [3] address the typical problem of optimal resource planning and maintenance activities . Using stochastic models of maintenance, Derman [4], Frostig [5], Koole [6] and Seshadri [7], propose a method based on the queue for the allocation of repair; Ivanov [8] proposed a solution to the problem of dynamic scheduling in the context of a multi-system remote maintenance skills. The same author [9] proposed a method for solving the problem of scheduling and allocation in remote maintenance systems based on stochastic descent algorithms and kangaroo. These results show that this method solves the problem large enough in polynomial time. The common context of these works is the integration of maintenance policies in the planning of maintenance tasks and production.
The target of the presented approach in this work is to present an algorithm that ensures the integration of the research technique of time spaces between preventive tasks and to insert new tasks or corrective delays, and allocate resources to tasks in order to minimize the cost of delays, on a horizon H. This paper is organized as follows: in the second section, we introduce the position of the problem. In the third section we give the solution suggested in section 4 and report the results of simulation and interpretation of the curves. Finally we conclude with an appendix that gives some prospects of the accomplished work.

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:

$$F_{OG} = \sum_i C_{Ti} + \sum_k C_{k\ perdu}, \quad [2]$$

$$F_{opti\_tasks} = \sum_i C_i = \min(\sum_{l=1} W_i\, i+ + h_i\, i- + C_{i0})$$

[2]

$$F_{opti\_OG} = \boldsymbol{min} \sum C_{Ti} + \boldsymbol{min} \sum C_{k\ lost}$$

$$F_{OG}\text{-optimize} = \min(\sum_{l=1} W_i\, i+ + h_i\, i- + C_{i0}) + \sum_K \sum_j^{n-1} ([T_{K+1} - T_K] - T_{dj}) \times C_{hour}$$

$$F_{OG}\text{-optimize} = \min(\sum_{l=1} W_i\, i+ + h_i\, i- + C_{i0}) + \sum_K \sum_j^{n-1} (F_K - T_{dj}) \times C_{hour}$$

$$C_{lost} = \sum_K [T_{K+1} - T_K] \times C_{hour} >= C'_{lost\text{-optimize}} = \sum_K \sum_j^{n-1} ([T_{K+1} - T_K] - T_{dj}) \times C_{hour}$$

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

Figure 3 : Diagram off principle optimization Scheduling task

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  PgT : task
 // PgT  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
PgT ← T1. di - T1. ri
begin
While (h ≤ N)
 if ((Th. dh –Th. rh) 〉 PgT) then
 PgT ← Th
 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 it's false
Lost_cost_Calculus ( )    // Downtime
Begin
Somme=0
for i=1 to N do
Somme=somme + Ti+1.d-Ti.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– T.di
 if (Min > = Durée _ Td) then
 Pos ← i
 T ← X
  X ← Task _next (piled)
  if  (Min > = (X. ri – T. di)) and
  (X. ri – T. di) > Duree_TD)
  Then Min ← X.ri – 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 + Ti+1.d-Ti.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
```

4. SIMULATIONS RESULTS

The algorithm of the tasks assignment was programmed by JAVA and the provision of the tasks resources by ms-Project. To evaluate this program, we considered a number of 10 tasks of preventive maintenance to schedule according to the resource and execution time. Then at the first time we inserted 3 dynamic tasks and in the second time 9 dynamic tasks.

a. Tasks before modeling

The tasks are sorted by ascending order of the execution time. The longest task is assigned to the most qualified resource. The program calculates the cost of each task and the lost cost between tasks, and posts the total costs of the windows called lost cost and the total costs of the tasks.

See (figure 4) (program java and execution)

b. Tasks after modeling(optimization) insertion of 3dynamic tasks:

According to the program execution, it is noticed that each dynamic task is inserted in a window to which its execution time is closest to the time of the current task. Moreover, the most qualified resource is assigned to the longest preventive task and it is this resource which is reallocated with the longest dynamic task, thus the program calculates the cost of each task and the lost cost between tasks and posts the total costs of the windows called lost cost and the total costs of the tasks . One notices according to the curve that the lost cost after optimization was attenuated of 37%.

See (figure 5) (program java and execution)

c. Tasks after modeling (optimization) insertion of 9 dynamic tasks:

According to the program execution, it is noticed that each dynamic task is inserted in a window to which its execution time is closest to the time of the current task. Furthermore, the most qualified resource is assigned to the longest preventive task and it is this resource which is reallocated with the longest dynamic task, thus the program calculates the cost of each task and the lost cost between tasks and posts the total costs of the windows called lost cost and the total costs of the tasks . We notice according to the curve that the lost cost after optimization was attenuated of 54%.
See (figure 6) (program java and execution)

d. Interpretation of the curves

By comparing the 3 curves (4, 5 and 6) it is noted that the lost cost falls while inserting a more dynamic tasks.
The lost cost after optimization (9 dynamic tasks) =6100DHS< lost Cost after optimization (3 dynamic tasks) =8300DHS < Cost lost before optimization =13100 DHS
Curve 4: cost of the preventive tasks and lost cost between tasks.
Curve 5: The lost cost is almost null until the preventive task N°7 starts to increase which means that it is necessary to program dynamic tasks starting from this task.
Curve 6: The lost cost is almost null until the preventive task N°10 which explains why the program inserted dynamic tasks into the lower part of this task.

5. CONCLUSION

This work deals with problem of scheduling optimization of the tasks and resource allocation, which aims to reduce the lost cost. We presented an algorithm which can schedule N preventive tasks of maintenance, (n-1) dynamic tasks. This method shows good performances of optimization of the lost cost while inserting more current tasks.
In our work, we projected the use of a more realistic . The methodology of scheduling and assignment of human resources developed here must be improved too. It is essential to integrate an explicit taking into consideration of the availability of the resources through their assignment planning by considering not only a single process as in the framework of this article, but the whole of the operational processes of a workshop. Then create a simulator which makes it possible to compare the performances of the algorithms of scheduling and to evaluate the impact of different simulation parameters.

Figures

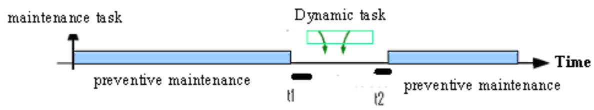

FIG1: Insertion of the dynamic spot in the elementary window

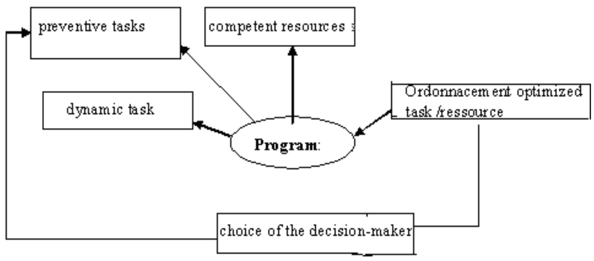

fiG2: model of dynamic assignment and scheduling of a new task

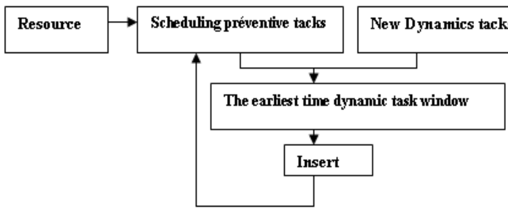

Figure 3: Schematic Diagram tasks scheduling optimization

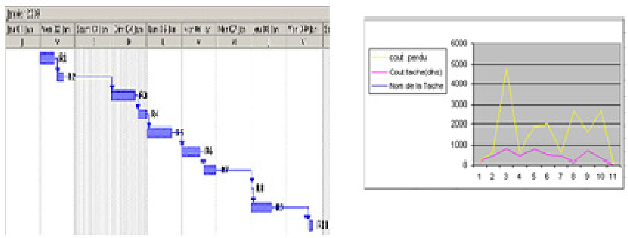

Figure 4 tasks before modeling

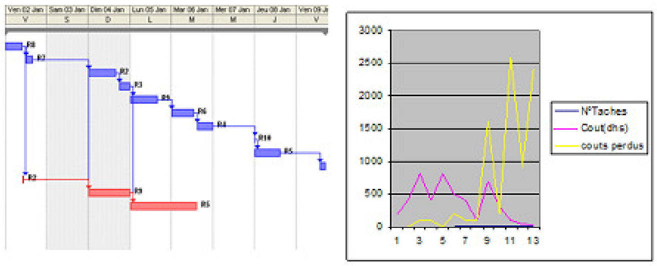

Figure 5:Insert 3 Dynamic tasks (after modeling)

References:

[1] R.Calibra, G. Pulcini, and Mr. Rapone, Optimal reliability and maintainability allowance in manufacturing systems, Quality and Reliability International Engineering 183-188(1995)

[2] W. Stadje and D. Zuckerman, Optimal repair policies with general degree of repair in two maintenance model, Operations Research Letters 11, 77–80. (1992)

[3] Bennour Contribution to the modeling and the human resource allocation in processes, Thesis of doctorate, University of Montpellier 2, (2004).

[4] C. Derman, G.J. Lieberman S.M., and Ross, One the optimal assignment off servers and has repairman,Newspaper off Applied Probability 17 703-715. (1980),

[5] E. Frostig, Optimal policies for machine repairman (1996)

[6] G. Koole, Optimal repairman assignment in two symmetric maintenance models, European Journalof Operational Research 82 295-301 (1995),

[7] A. Ivanov, C. Varnier, N. Zerhouni, Ordonnancement dynamique des tâches de maintenance dans un système de télémaintenance, CIP'03, Alger (Algérie), Octobre 2003.

[8] A. Ivanov, C. Varnier, N. Zerhouni, Ordonnancement des activités de maintenance dans un contexte distribué

[9] Djalal HEDJAZI 1, Abdelmadjid ZIDANI 2, Samra DEY3, Zouleikha TEJANI Algorithme of stops left adapted for the scheduling of the tasks of maintenance in a Context distributed maintenance "Conference one Computer Integrated Manufacturing (2007)

[10] EH .Aghezzaf ,A.jamali,D.Ait Kadi An integrated production and preventive maintenance planning model .European journal of operational Research (2007).

[11] Joseph Y-T. Leung, ***Handbook*** *of Scheduling: Algorithms, Models, and Performance Analysis*, Chapman & Hall/CRC Computer & Information Science Series, 2004.

[12] **S. Seshadri**, Determination off aggregate preventive maintenance programs using production schedules, Computers and Industrial Engineering 14 193-200 (1988),