



**HAL**  
open science

# Algorithme d'aides à la décision pour Optimiser l'Ordonnancement des tâches de maintenance en temps-réel

Aissam Aboussalim, Mohamed Mediouni

► **To cite this version:**

Aissam Aboussalim, Mohamed Mediouni. Algorithme d'aides à la décision pour Optimiser l'Ordonnancement des tâches de maintenance en temps-réel. 2010, pp.X. inria-00493379v2

**HAL Id: inria-00493379**

**<https://inria.hal.science/inria-00493379v2>**

Submitted on 28 Aug 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Algorithme d'aides à la décision pour Optimiser l'Ordonnancement des tâches de maintenance en temps-réel.

Mr Aboussalim Aissam, Mr Mediouni Mohamed

LGII- Laboratoire du Génie Industriel et informatique Ensa Agadir

Email : [aboussalimaissam@hotmail.fr](mailto:aboussalimaissam@hotmail.fr)

## Résumé :

L'approche présentée dans cet article est une contribution à la recherche portant sur l'optimisation de l'ordonnancement des tâches.

Ce travail répond aux objectifs suivants : Décision en temps-réel (planification et ordonnancement), Allocation/affectation de ressources, Assurer une disponibilité minimum requise, Minimiser les temps retard en insérant les tâches dynamiques « maintenance corrective ».

**MOT-CLÉS :** tâches, ordonnancement, optimisation, ressource, coût

## 1. INTRODUCTION

Le problème d'ordonnancement consiste à organiser dans le temps la réalisation des tâches, compte tenu de contraintes temporelles et de contraintes portant sur l'utilisation et la disponibilité de ressources requises. L'optimisation de l'ordonnancement des tâches est devenue un domaine essentiel de la recherche. Il contribue à aider les entreprises à rentabiliser au mieux leurs systèmes de production.

Nous nous intéressons plus particulièrement aux stratégies d'insertion des tâches dynamiques pour lesquelles les décisions (entretenir, réparer, remplacer, inspecter, ...) sont prises sur la base d'informations de surveillance en ligne.

La modélisation des systèmes à composants multiples avec dépendances et sur les techniques d'optimisation multicritères. Dans la plupart des articles dédiés à ce type de problème, les auteurs supposent qu'il n'y a pas de temps mort entre deux tâches consécutives. De plus, les problèmes traités se limitent à l'ordonnancement de tâches sur une seule machine. Tout d'abord de présenter un algorithme qui garantit le respect des contraintes temporelles et des ressources et assure l'ordonnancement des tâches ainsi que l'affectation des ressources.

Afin de réduire les coûts de maintenance, les travaux de Calibra *et al.* [1] ont traité des problèmes d'allocation de fiabilité, L'amélioration des coûts de maintenance et de la disponibilité peut aussi être obtenue par la gestion optimale des ressources humaines, de même que Parmi les rares travaux qui abordent le problème typique de planification optimale des ressources et activités de maintenance, il y a notamment les travaux de Graves et Lee [2], Lee et Chen [3], Ce dernier groupe de travaux est plutôt fondé sur des approches de gestion de production. Ainsi Les travaux par Cervin d'ordonnancement et planification des tâches en temps réelles [4],

L'approche présentée dans ce travail a pour objectif d'affecter les ressources aux tâches de façon à minimiser, sur un horizon  $H$ , le coût des retards ou déviations, de proposer un algorithme permettant de trier, insérer les tâches dynamiques en temps réel, offrir une interface utilisateur afin d'ordonner les tâches en temps réel et permettre une utilisation maximale des ressources pour un coût perdu minimal.

Ce papier est organisé de la manière suivante, dans la deuxième section, on présente la position du problème,, la section 3, on donne la solution proposée, et dans la section 4 on présente les résultats expérimentaux et enfin conclusion.

## 2. Position du problème

Le problème d'ordonnancement de la maintenance ou de la production est NP-difficile. il est classé en 2 catégories, ordonnancement en ligne et ordonnancement hors-ligne. Un ordonnancement hors-ligne signifie que la séquence d'ordonnancement est prédéterminée à l'avance : dates de début d'exécution des tâches. En pratique, l'ordonnancement prend la forme d'un plan hors-ligne (ou statique), exécuté de façon répétitive. L'ordonnanceur s'appelle dans ce cas un *séquenceur*. Un ordonnancement en-ligne correspond au déroulement d'un algorithme qui tient compte des tâches effectivement présentes dans la file d'ordonnancement. lors de chaque décision d'ordonnancement

On doit minimiser les dates d'achèvement de la réalisation des tâches, optimiser l'utilisation des ressources sous des contraintes de durées. On plus il faut insérer des tâches dynamiques en temps réel. Pour modéliser ce problème par des algorithmes considérés comme NP difficile on a procédé à donner des heuristiques qui permettent d'obtenir des solutions approchées par rapport à la solution optimale

Au niveau de l'organisation du système, nous adoptons la structure suivante : un centre de compétence, qui évalue les ressources humaines et

les classe par bilan de compétence pour une période déterminée.

Ce modèle est une suite de l'article d'Algorithme d'arrête gauche [5] adapté pour l'ordonnancement des tâches de maintenance. Des fenêtres optimales permettent d'affecter la tâche courante dans la meilleure fenêtre élémentaire et de calculer à chaque fenêtre élémentaire le coût perdu. La valeur ajoutée de notre approche est d'affecter les ressources les plus compétentes aux premiers tâches et ceci pour achever les tâches dans les meilleurs délais et minimiser la somme pondérée des temps mort, en insérant les tâches dynamiques dans les fenêtres avoisines de leur durées, ainsi l'indisponibilité de chaque site et le coût perdu doit être minimal et de programmer un nombre illimité de taches

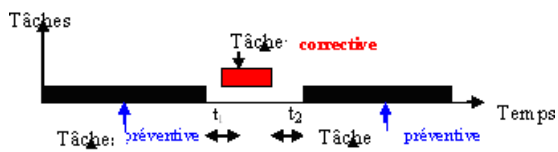


FIG1 : Insertion de la tâche dynamique dans la fenêtre élémentaire

Note but est d'insérer une tâche dynamique telle que  $\lim t1 \rightarrow 0$  et  $\lim t2 \rightarrow 0$

### a- Hypothèses

- Nous nous positionnons dans le cas des tâches préventives pour planifier les tâches sur un horizon H.
- Les équipes ont des niveaux de compétences différents classés par un service d'exams responsables des bilans de compétence.
- On procède à appliquer notre modèle sur une seule machine
- Les tâches ont des priorités égales.
- Les dates d'intervention et les durées de tâches sont calculées sur la base de l'analyse statistique des pannes.
- Chaque tâche achevée doit être supprimée.
- Les dates d'échéance de tâches sont réparties dans l'ordonnancement optimal, tous les temps libres entre les tâches seront rempli par des tâches courantes pour minimiser Le coûts global

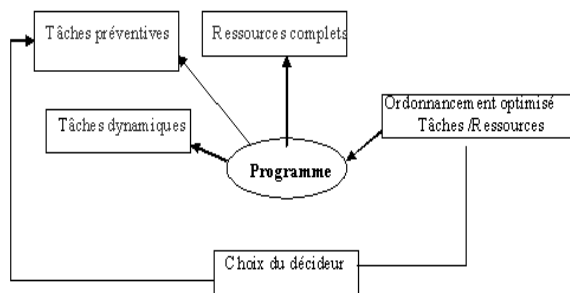


FIG2 : modèle d'affectation et d'ordonnancement dynamiques d'une nouvelle tâche

### b- Formulation des hypothèses

L'ordonnancement d'un ensemble de n tâches periodiques est faisable

$$\tau = \{\tau_1, \dots, \tau_i, \dots, \tau_n\}$$

Si le facteur d'utilisation

$$U = \sum_{i=1}^n C_i / P_i \quad [6]$$

Vérifie la relation suivante

$$U \leq n \cdot (\sqrt[n]{2} - 1) \quad [6]$$

une condition suffisante d'ordonnancabilité d'après l'algorithme RM [6]

$C_i$  : Durée d'exécution maximale ;  $P_i$  : période de la tâche

Le service est composé de m ressources humaines ( $i=1 \dots m$ ), chacune Caractérisée par un profil de compétence. Chaque ressource a donc un niveau de qualification correspondant à chaque type de tâche. C'est ce niveau de qualification qui fait que, d'une ressource à une autre, le temps de traitement ne sera pas le même.

Où  $C_{ik}$  est le taux de compétence de la ressource i dans la compétence requise pour le traitement de la tâche du type k. Il est donc possible de représenter cela par une matrice dans laquelle, pour chaque type de tâche et pour chaque ressource, on obtient le taux de Compétence correspondant.. Cette partie présente la formulation du problème d'affectation qui peut être développée dans le cas où l'on impose de satisfaire, pour un processus donné, la contrainte Performance\_Processus  $\leq$  OB (durée, coût). [7]  
OB : l'objectif de performance imposé au processus Elle peut être adaptée sans difficulté au cas où la contrainte définie est de type Performance\_Processus  $\geq$  OB (qualité). [7]

$$\begin{bmatrix} C_{1,1} & \dots & C_{1,o} \\ \vdots & \ddots & \vdots \\ C_{m,1} & \dots & C_{m,o} \end{bmatrix} \quad [8]$$

Le coût des activités est la somme des coûts de toutes les tâches et les coûts perdus entre taches qu'on peut appeler fenêtre. Le coût perdu est le produit du temps de déplacement et le coût d'une heure de déplacement :

$$C_k \text{ perdu} = [T_{K+1} - T_K] \times C_{\text{heure}}$$

$T_K$  : Temps d'exécution de la tâche k

La fonction globale s'écrit comme :

$$F_{OG} = \sum_i C_{Ti} + \sum_k C_k \text{ perdu},$$

$C_{Ti}$  : cout d'une tache i

$$F_{\text{opti\_taches}} = \sum_i C_{Ti} = \sum_{l=1} W_l \theta_l + h_l \theta_{l-} + C_{i0} \quad (9)$$

« Cette fonction décide pour chaque tâche de dévier positivement ou négativement respectivement par rapport à sa date due au plus tôt ou sa date due au plus tard, soit de co

Chaque tâche  $i$  se caractérise par une durée opératoire notée  $T_i$

$\theta_i^-$  :: désigne la déviation négative de la tâche  $i$  par rapport à sa date due au plus tôt ou également l'avance.

$\theta_i^+$  :: désigne la déviation positive de la tâche  $i$  par rapport à sa date due au plus tard ou également le retard.

Une pénalité de retard  $W_i$ , une pénalité d'avance  $h_i$ ,  $C_{i0}$  le coût minimal de maintenance préventive

$$F_{OG} = \sum_{l=1}^n W_l \theta_l + h_l \theta_l^- + C_{i0} + \min \sum_K [T_{K+1} - T_K] \times C_{heur}$$

Le coût des activités est la somme des coûts des tâches ajoutés aux coûts perdus entre tâches  $k+1$  et  $k$

### 3. Solution proposée

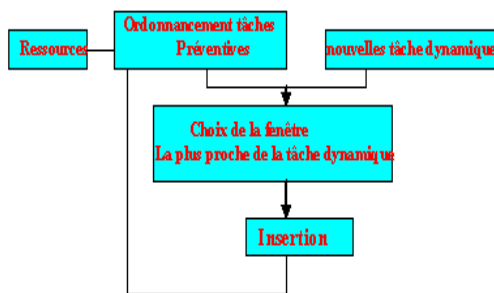


Fig3 Schéma de principe d'optimisation d'ordonnement des tâches

La solution proposée reçoit en entrée une liste  $L$  contenant des tâches avec leurs caractéristiques ( $r_i$ ,  $d_i$ ,  $c_i$ ).

$r_i$  : date début de tâche ( $r_i$ ).

$d_i$  : date fin de tâche

$c_i$  est le coût de la tâche :

- ✓ Optimisation d'une seule tâche afin d'être planifiée dans son intervalle optimal
- ✓ Tri des tâches par ordre croissant par date début et par temps d'exécution ( $d_i - r_i$ )
- ✓ Affectation des ressources les plus compétentes en temps d'exécution ( $d_i - r_i$ ) le plus grand.
- ✓ Calcul des coûts des tâches et de coûts perdu par la fonction FOG
- ✓ Calculer temps d'exécution de la tâche dynamique ( $d_j - r_j$ )
- ✓ Chercher et calculer le temps de toutes les fenêtres « temps perdu entre les tâches »
- ✓ Insertion des tâches dynamique dans la meilleure fenêtre dont son temps d'exécution est approximativement égale.
- ✓ affecter les ressources les plus compétentes disponibles aux premières tâches dynamiques les plus longues.

#### EXEMPLE D'APPLICATION

- Insertion des nouvelles tâches courantes non prévues ou en retards
- Robot d'exploration : Ce robot doit se déplacer dans un environnement non connu (zone radioactive, planète, épave sous la mer).

- Système de contrôle arien : ce système rassemble l'information sur l'état de chaque avion via un ou plusieurs radars actifs. Ce radar interroge chaque avion périodiquement. Le système traite les messages reçus de l'avion et stocke les informations dans une base de données. Ces informations sont prises en compte et traitées par des processeurs graphiques. Au même temps, un système de surveillance analyse de façon continue le scénario et alerte les opérateurs toutefois qu'il détecte une collision possible.

Tous ces exemples cités on doit avoir des temps libre entre tâches automatisées afin d'intervenir

### Algorithme

**Affectation des tâches ( $T_i$ )** //  $T_i$  : Tâche  $i$   
**Début**

Si ( $P = \emptyset$ ) alors //  $P$  : Pile des Tâches

$P \leftarrow T_i$  // début de la pile

Sinon Aller\_Fin\_Pile ( $P$ )  $P \leftarrow T_i$

//  $T_i$  est ajouté à la fin de la pile

**FinSi**

**FIN**

**Affecter\_Pile\_Ressource ( $R_i$ )** //  $R_i$  : ressource  $i$

**Début**

Si ( $R \neq \emptyset$ ) alors  $P \leftarrow R_i$

//  $R$  : Pile des ressources

sinon

Aller\_Fin\_Ressource ( $R$ )

$R \leftarrow R_i$

**FINSI**

**Fin**

**Tri\_Ressource ( $R$ )** //  $R$  : Pile des ressources, une ressource se caractérise par une note et une variable disponible

**Début**

$R_i \leftarrow$  Première\_ressource

Tant que ( $R_i$ .disponible = False)

$R_i \leftarrow$  Ressource\_suivante

**FinTque**

Si ( $R_i$ .disponible = vrai) alors

Affecter\_Ressource\_Tache ( $R_i, P$ )

**Sinon**

Ecrire « Ressource non Disponible »

**Fin Si**

**Fin**

**Affecter\_Ressource\_Tache ( $R_i$ , pile  $P$ )**

//  $R_i$  : Ressource la plus Compétente disponible  
 Variable  $P_g T$  : Tâche //  $P_g T$  la plus grande tâche

Variable  $h, pos, N$  //  $h$  compteur de la pile  $P$

//  $N$  nombre de tâches

//  $pos$  est la position de tâche dans la pile  $d_i$  et  $r_i$  sont les paramètres de la tâche ( $d_i$  et  $r_i$ )

```

Trier_tâche ( )
h ← 2
pos ← 1
PgT ← T1. di - T1. ri
DEBUT
Tant que (h ≤ N)
    Si ((Th. dh - Th. rh) > PgT) alors
        PgT ← Th
        Pos ← h
    Finsi
    h ← h+1
FinTque
Ri .disponible ← False

Fin
// Disponible est un paramètres qui vaut
true si la ressource n'est pas prise sinon false
Calcul_cout_perdu ( ) //Temps d'arrêt
Début
Somme=0
Pour i=1 à N faire
    Somme=somme + Ti+1.d-Ti.r
Fin Pour
// Afficher « coût perdu = », somme*Ci0
Calcul_cout_perdu ← somme*Ci0
Fin
Favant ← Calcul_cout_perdu( )
Après Optimisation //insertion des
tâches dynamiques
Inser_Tache_D (Td ; P)
Début
Var X, T : tâche
Var : Durée_TD, pos : ENTIER
Durée -Td ← Td.di -Td. ri
T ← première_tache (Pile)
Pos ← 1
i ← 1
Tant que (T < > Fin_pile)
    X ← Tache_Suivante (Pile)

```

```

Min ← X.ri- T.di
SI (Min ≥ Durée _ Td) alors
    Pos ← i
    T ← X
    X ← Tache_suivante (pile)
    SI (Min ≥ (X. ri - T. di)) et
        (X. ri - T. di) > Durée- TD)
        Min ← X.ri - T.di
        T ← X
        Pos ← i
    fsi
fsi
fin
// Se positionner avant la tâche ou il faut insérer
T ← premier Tache (Rle)
Pour i = 1 à pos
    T ← Tache_suivant (Rle)
Fpour
    Z ← T. suivant
    T. Suivant ← Td
    Td. suivant ← Z .suivant

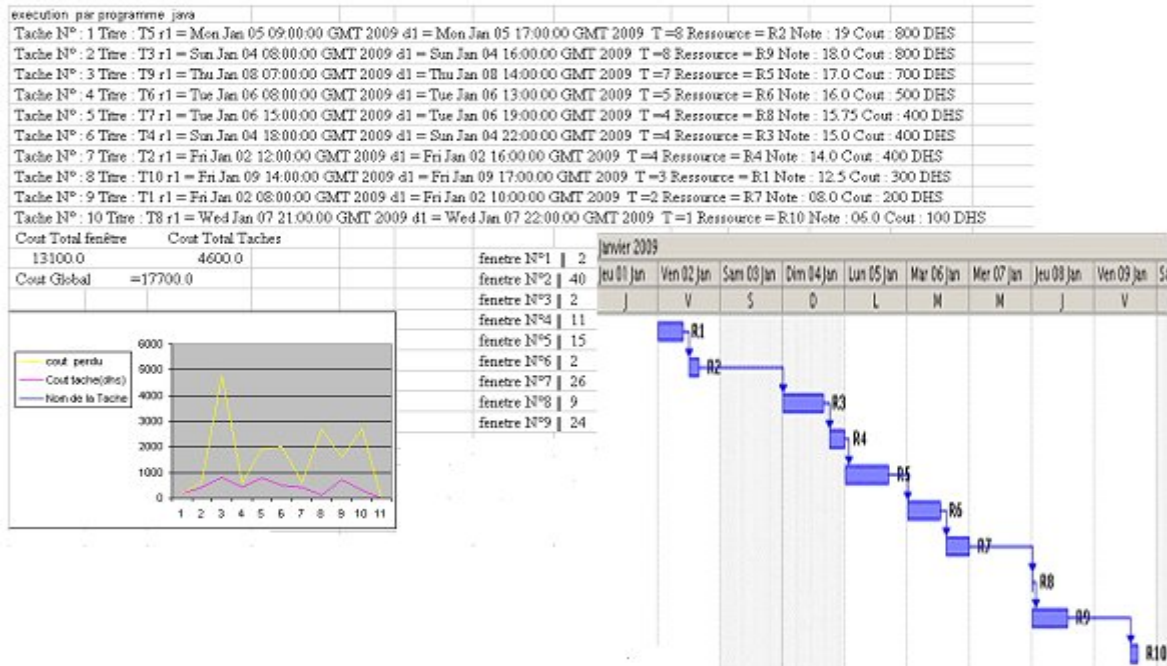
Calcul_cout_perdu ( )
//Temps d'arrêt //après optimisation
Début
Somme=0
Pour i=1 a N faire
    Somme=somme + Ti+1.d-Ti.r
Fin Pour
// Afficher « coût perdu = »,
somme*Ci0
Calcul_cout_perdu ← somme*Ci0
Fin
Fapres ← Calcul_cout_perdu( )
//après insertion des tâches dynamiques
// Tg ← Favant - Fapres

```

#### **4. RÉSULTATS**

L'algorithme d'affectation des tâches a été programmé par JAVA.et la disposition des tâches ressources par ms-Project. Pour évaluer ce programme, nous avons considéré un ensemble de 10 tâches préventives à ordonnancer en fonction du temps d'exécution et des ressources. Puis on a inséré dans un premier temps 3 tâches dynamiques et dans un deuxième temps 9 tâches dynamiques.

##### **a- Taches avant modélisation**



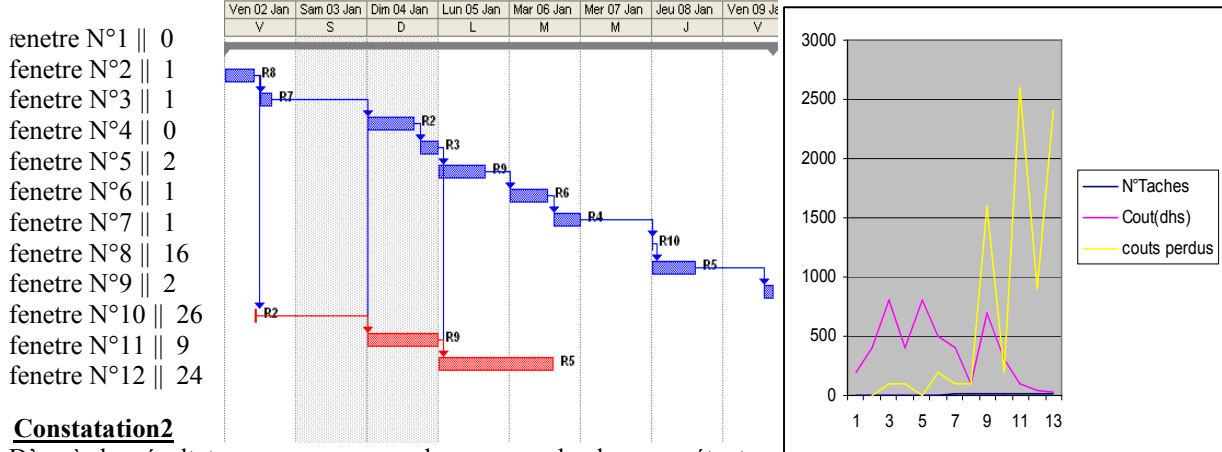
**Constatation1**

Les tâches sont triées par ordre croissant du temps d'exécution .la tâche la plus longue est affectée à la ressource la plus compétente .le programme calcule le coût de chaque tâche et le coût perdu entre tâches

**b- Taches après modélisation(optimisation) INSERTION DE 3TACHES DYNAMIQUES**

Tache N° :1 Titre : T1 r1 = Fri Jan 02 08:00:00 GMT 2009 d1 = Fri Jan 02 10:00:00 GMT 2009 T=2 Ressource = R7 Note : 08.0 Coût : 200 DHS  
Tache N° :2 Titre : TD1 r1 = Fri Jan 02 10:00:00 GMT 2009 d1 = Fri Jan 02 11:00:00 GMT 2009 T=1 Ressource = R2 Note : 19 Coût : 100 DHS  
Tache N° :3 Titre : T2 r1 = Fri Jan 02 12:00:00 GMT 2009 d1 = Fri Jan 02 16:00:00 GMT 2009 T=4 Ressource = R4 Note : 14.0 Coût : 400 DHS  
Tache N° :4 Titre : TD2 r1 = Fri Jan 02 17:00:00 GMT 2009 d1 = Sun Jan 04 08:00:00 GMT 2009 T=39 Ressource = R9 Note : 18.0 Coût : 3900  
Tache N° :5 Titre : T3 r1 = Sun Jan 04 08:00:00 GMT 2009 d1 = Sun Jan 04 16:00:00 GMT 2009 T=8 Ressource = R2 Note : 19 Coût : 800 DHS  
Tache N° :6 Titre : T4 r1 = Sun Jan 04 18:00:00 GMT 2009 d1 = Sun Jan 04 22:00:00 GMT 2009 T=4 Ressource = R3 Note : 15.0 Coût : 400 DHS  
Tache N° :7 Titre : TD3 r1 = Sun Jan 04 23:00:00 GMT 2009 d1 = Mon Jan 05 07:00:00 GMT 2009 T=8 Ressource = R5 Note : 17.0 Coût : 800 DHS  
Tache N° :8 Titre : T5 r1 = Mon Jan 05 08:00:00 GMT 2009 d1 = Mon Jan 05 16:00:00 GMT 2009 T=8 Ressource = R9 Note : 18.0 Coût : 800 DHS  
Tache N° :9 Titre : T6 r1 = Tue Jan 06 08:00:00 GMT 2009 d1 = Tue Jan 06 13:00:00 GMT 2009 T=5 Ressource = R6 Note : 16.0 Coût : 500 DHS  
Tache N° :10 Titre : T7 r1 = Tue Jan 06 15:00:00 GMT 2009 d1 = Tue Jan 06 19:00:00 GMT 2009 T=4 Ressource = R8 Note : 15.75 Coût : 400 DHS  
Tache N° :11 Titre : T8 r1 = Wed Jan 07 21:00:00 GMT 2009 d1 = Wed Jan 07 22:00:00 GMT 2009 T=1 Ressource = R10 Note : 06.0 Coût : 100 DHS  
Tache N° :12 Titre : T9 r1 = Thu Jan 08 07:00:00 GMT 2009 d1 = Thu Jan 08 14:00:00 GMT 2009 T=7 Ressource = R5 Note :17.0 Coût : 700 DHS  
Tache N° :13 Titre : T10 r1 = Fri Jan 09 14:00:00 GMT 2009 d1 = Fri Jan 09 17:00:00 GMT 2009 T=3 Ressource = R1 Note : 12.5 Coût : 300 DHS

Coût Total fenêtré      Coût Total Taches  
8300.0                    9000.0  
Coût Global            =17300.0



**Constatation2**

D'après les résultats, on remarque que la ressource la plus compétente est affectée à la tâche préventive la plus longue et c'est cette ressource qui se réaffecte à la tâche dynamique la plus longue. On remarque d'après la courbe que le coût perdu après optimisation a été atténué de 37%.

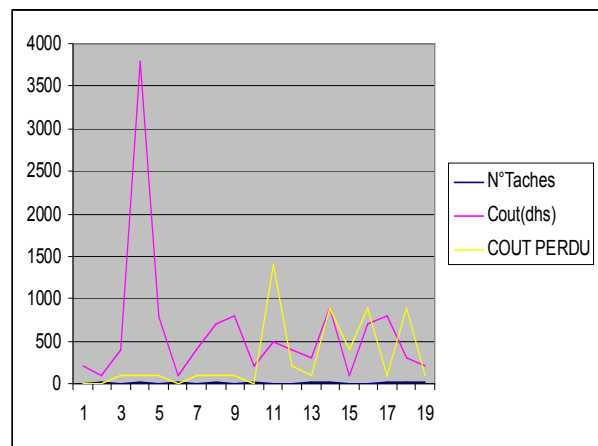
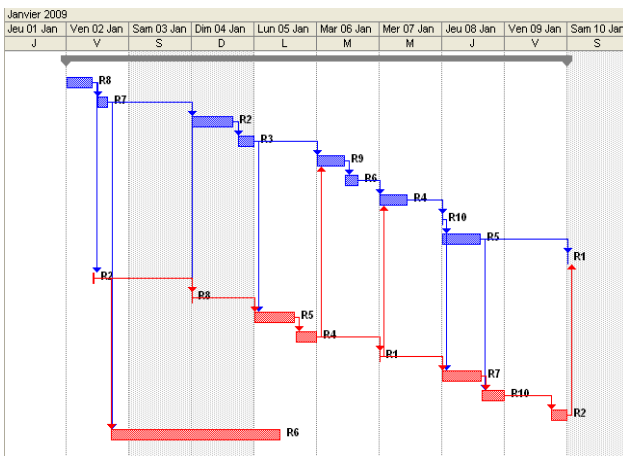
**c- Taches après modélisation(optimisation) INSERTION DE 9TACHES DYNAMIQUES**

Tache N° :1 Titre : T1 r1 = Fri Jan 02 08:00:00 GMT 2009 d1 = Fri Jan 02 10:00:00 GMT 2009 T=2 Ressource = R7 Note : 08.0 Coût : 200DHS  
Tache N° :2 Titre : TD1 r1 = Fri Jan 02 10:00:00 GMT 2009 d1 = Fri Jan 02 11:00:00 GMT 2009 T=1 Ressource = R1 Note : 12.5 Coût : 300 DHS  
Tache N° :3 Titre : T2 r1 = Fri Jan 02 12:00:00 GMT 2009 d1 = Fri Jan 02 16:00:00 GMT 2009 T=4 Ressource = R4 Note : 14.0 Coût : 400 DHS  
Tache N° :4 Titre : TD2 r1 = Fri Jan 02 17:00:00 GMT 2009 d1 = Sun Jan 04 08:00:00 GMT 2009 T=39 Ressource = R2 Note : 19 Coût : 3900 DHS  
Tache N° :5 Titre : T3 r1 = Sun Jan 04 08:00:00 GMT 2009 d1 = Sun Jan 04 16:00:00 GMT 2009 T=8 Ressource = R2 Note : 19 Coût : 800 DHS  
Tache N° :6 Titre : TD4 r1 = Sun Jan 04 16:00:00 GMT 2009 d1 = Sun Jan 04 17:00:00 GMT 2009 T=1 Ressource = R7 Note : 08.0 Coût : 200 DHS  
Tache N° :7 Titre : T4 r1 = Sun Jan 04 18:00:00 GMT 2009 d1 = Sun Jan 04 22:00:00 GMT 2009 T=4 Ressource = R3 Note : 15.0 Coût : 400 DHS  
Tache N° :8 Titre : TD3 r1 = Sun Jan 04 23:00:00 GMT 2009 d1 = Mon Jan 05 07:00:00 GMT 2009 T=8 Ressource = R5 Note : 17

Tache N° :9 Titre : T5 r1 = Mon Jan 05 08:00:00 GMT 2009 d1 = Mon Jan 05 16:00:00 GMT 2009 T=8 Ressource = R9 Note : 18.0 Cout : 800 DHS  
Tache N° :10 Titre : TD5 r1 = Mon Jan 05 16:00:00 GMT 2009 d1 = Mon Jan 05 18:00:00 GMT 2009 T=2 Ressource = R3 Note : 15.0 Cout : 200 DHS  
Tache N° :11 Titre : T6 r1 = Tue Jan 06 08:00:00 GMT 2009 d1 = Tue Jan 06 13:00:00 GMT 2009 T=5 Ressource = R6 Note : 16.0 Cout : 500 DHS  
Tache N° :12 Titre : T7 r1 = Tue Jan 06 15:00:00 GMT 2009 d1 = Tue Jan 06 19:00:00 GMT 2009 T=4 Ressource = R8 Note : 15.75 Cout : 400 DHS  
Tache N° :13 Titre : TD6 r1 = Tue Jan 06 20:00:00 GMT 2009 d1 = Tue Jan 06 23:00:00 GMT 2009 T=3 Ressource = R8 Note : 15.75 Cout : 300 DHS  
Tache N° :14 Titre : TD7 r1 = Wed Jan 07 08:00:00 GMT 2009 d1 = Wed Jan 07 17:00:00 GMT 2009 T=9 Ressource = R9 Note : 18.0 Cout : 900 DHS  
Tache N° :15 Titre : T8 r1 = Wed Jan 07 21:00:00 GMT 2009 d1 = Wed Jan 07 22:00:00 GMT 2009 T=1 Ressource = R10 Note : 06.0 Cout : 100 DHS  
Tache N° :16 Titre : T9 r1 = Thu Jan 08 07:00:00 GMT 2009 d1 = Thu Jan 08 14:00:00 GMT 2009 T=7 Ressource = R5 Note : 17.0 Cout : 700 DHS  
Tache N° :17 Titre : TD8 r1 = Thu Jan 08 15:00:00 GMT 2009 d1 = Thu Jan 08 23:00:00 GMT 2009 T=8 Ressource = R6 Note : 16.0 Cout : 800 DHS  
Tache N° :18 Titre : T10 r1 = Fri Jan 09 14:00:00 GMT 2009 d1 = Fri Jan 09 17:00:00 GMT 2009 T=3 Ressource = R1 Note : 12.5 Cout : 300 DHS  
Tache N° :19 Titre : TD9 r1 = Fri Jan 09 18:00:00 GMT 2009 d1 = Fri Jan 09 20:00:00 GMT 2009 T=2 Ressource = R4 Note : 14.0 Cout : 100 DHS

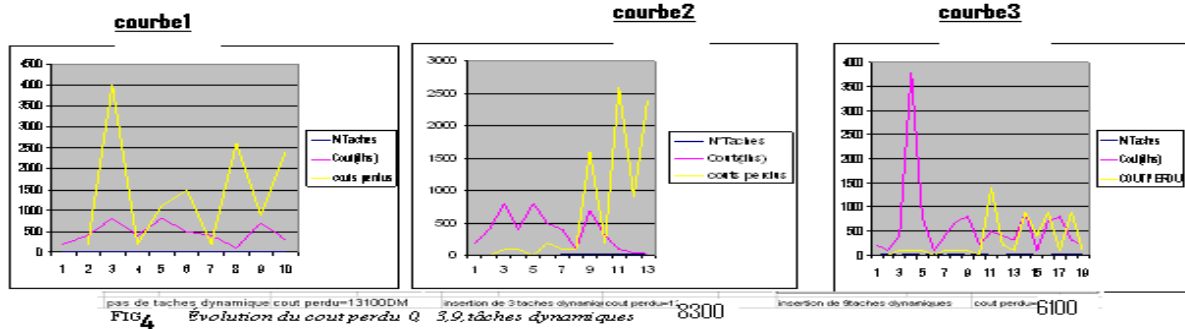
Cout Total fen?tre      Cout Total Taches  
6100.0                      10000.0  
Co?t Global                =16100.0

fenetre N°1    0	fenetre N°7    1	fenetre N°13    9
fenetre N°2    1	fenetre N°8    1	fenetre N°14    4
fenetre N°3    1	fenetre N°9    0	fenetre N°15    9
fenetre N°4    0	fenetre N°10    14	fenetre N°16    1
fenetre N°5    0	fenetre N°11    2	fenetre N°17    15
fenetre N°6    1	fenetre N°12    1	fenetre N°18    1



**Constatacion3**

D'après les résultats, on remarque que la ressource la plus compétente est affecté à la tache préventive la plus longue et c'est cette ressource qui se réaffecte a la tache dynamique la plus longue. On remarque d'après la courbe que le cout perdu après optimisation a été atténué de **54%**



**Interprétation des courbes**

En comparant les 3courbes on constate que le cout perdu chute en insérant de plus en plus de taches dynamiques  
Cout perdu après optimisation (9taches dynamiques) =6100DHS < Cout perdu après optimisation(3 taches dynamiques) =8300DHS < Cout perdu avant optimisation=13100 DHS

Courbe 1 : coût des taches préventives et cout perdu entre taches

Courbe 2 : Le coût perdu est quasiment nul jusqu' à la tâche préventive N°7 commence à augmenter ce qui explique il faut programmer des tâches dynamiques à partir de cette tâche

Courbe 3 : Le coût perdu est quasiment nul jusqu' à la tache préventive N°10 ce qui explique que le programme a inséré des taches dynamiques .au dessous de cette tâche

**4. CONCLUSION**

Ce travail traite un problème d'optimisation d'ordonnancement des tâches et d'affectation des ressources, qui a pour objectif la réduction du coût perdu. Nous avons présenté un algorithme qui peut ordonnancer n tâches

préventives, (n-1) tâches dynamiques. Cette méthode a montré de bons résultats d'optimisation du coût perdu en insérant de plus en plus de tâches courantes.

## REFERENCES

[1] R. Calibra, G. Pulcini, and M. Rapone, *Optimal reliability and maintainability allocation in manufacturing systems*, Quality and Reliability Engineering International **11** (1995), 183–188

[2] C.Y. Lee and Z.L. Chen, *Scheduling jobs and maintenance activities on parallel machines*, Naval Research Logistics **47** (2000), 145

[3] X. Qi, T. Chen, and F. Tu, *Scheduling the maintenance on a single machine*, Journal of the Operation Research Society **50** (1999), 10,1071

[4] A. Cervin "Improved Scheduling of Control Tasks" in Proceedings of the 11th Euromicro Conference on Real-Time Systems, York, England, June 1999.

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

[6] C. Liu and J. Layland, Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment, Journal of ACM, vol. 29, n° 1, 1973, pp 46-61

[7] Bennour M., Contribution à la modélisation et à l'affectation des ressources humaines dans les processus, Thèse de doctorat, Université de Montpellier 2, 2004.

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

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

## Annexes

N°Taches	Durée (H)	Début	Fin	Coût(dhs)	tom Ressourc	type taches	coûts perdus	fenetres
1	2	2/1/09 8:00	2/1/09 10:00	200	R8=15,75	preventive		
2	4	2/1/09 12:00	2/1/09 16:00	400	R7=8	preventive	200	fenetre N°1
3	8	4/1/09 8:00	4/1/09 16:00	800	R2=19	preventive	4000	fenetre N°2
4	4	4/1/09 18:00	4/1/09 22:00	400	R3=15	preventive	200	fenetre N°3
5	8	5/1/09 8:00	5/1/09 16:00	800	R9=18	preventive	1100	fenetre N°4
6	5	6/1/09 8:00	6/1/09 13:00	500	R6=16	preventive	1500	fenetre N°5
7	4	6/1/09 15:00	6/1/09 19:00	400	R4=14	preventive	200	fenetre N°6
8	1	7/1/09 21:00	7/1/09 22:00	100	R10=6	preventive	2600	fenetre N°7
9	7	8/1/09 7:00	8/1/09 14:00	700	R5=17	preventive	900	fenetre N°8
10	3	9/1/09 14:00	9/1/09 17:00	300	R1=12,5	preventive	2400	fenetre N°9
cout perdu (avant optimisation)=131*100=13100DHS								13100
		Tableau1			saisie des durees des taches preventives et calcul des fenetres			
N°Taches	Durée (H)	Début	Fin	Coût(dhs)	tom Ressourc	type taches	coûts perdus	fenetres
1	2	2/1/09 8:00	2/1/09 10:00	200	R8=15,75	preventive		
11	1	2/1/09 10:00	2/1/09 11:00	100	R2=19	dynamique	0	fenetre N°1    0
2	4	2/1/09 12:00	2/1/09 16:00	400	R7=8	preventive	100	fenetre N°2    1
3	8	4/1/09 8:00	4/1/09 16:00	800	R2=19	preventive	100	fenetre N°3    1
12	15	4/1/09 8:00	4/1/09 23:00	1500	R9=18	dynamique	0	fenetre N°4    0
4	4	4/1/09 18:00	4/1/09 22:00	400	R3=15	preventive	200	fenetre N°5    2
13	22	4/1/09 23:30	5/1/09 21:30	2200	R5=17	dynamique	100	fenetre N°6    1
5	8	5/1/09 8:00	5/1/09 16:00	800	R9=18	preventive	100	fenetre N°7    1
6	5	6/1/09 8:00	6/1/09 13:00	500	R6=16	preventive	1600	fenetre N°8    1
7	4	6/1/09 15:00	6/1/09 19:00	400	R4=14	preventive	200	fenetre N°9    2
8	1	7/1/09 21:00	7/1/09 22:00	100	R10=6	preventive	2600	fenetre N°10
9	7	8/1/09 7:00	8/1/09 14:00	700	R5=17	preventive	900	fenetre N°11
10	3	9/1/09 14:00	9/1/09 17:00	300	R1=12,5	preventive	2400	fenetre N°12
		Tableau2			insertion de 3 taches dynamiques			
							8300	
N°Taches	Durée (H)	Début	Fin	Coût(dhs)	tom Ressourc	fenetres	COÛT PERDU	fenetres
1	2	2/1/09 8:00	2/1/09 10:00	200	R8=15,75	0	0	
11	1	02/01/2009 10:00	02/01/2009 11:00	100	R9=18	0	0	fenetre N°1    0
2	4	2/1/09 12:00	2/1/09 16:00	400	R7=8	1	100	fenetre N°2
19	38	02/01/2009 17:00	04/01/2009 07:00	3800	R6=16	1	100	fenetre N°3
3	8	4/1/09 8:00	4/1/09 16:00	800	R2=19	0	0	fenetre N°4
12	1	04/01/2009 16:00	04/01/2009 17:00	100	R8=15,75	0	0	fenetre N°5
4	4	4/1/09 18:00	4/1/09 22:00	400	R3=15	1	100	fenetre N°6
13	7	04/01/2009 23:00	05/01/2009 07:00	700	R3=15	1	100	fenetre N°7
5	8	5/1/09 8:00	5/1/09 16:00	800	R9=18	1	100	fenetre N°8
14	2	05/01/2009 16:00	05/01/2009 18:00	200	R4=14	0	0	fenetre N°9
6	5	6/1/09 8:00	6/1/09 13:00	500	R6=16	14	1400	fenetre N°10
7	4	6/1/09 15:00	6/1/09 19:00	400	R4=14	2	200	fenetre N°11
15	3	06/01/2009 20:00	06/01/2009 23:00	300	R1=12,5	1	100	fenetre N°12
16	9	07/01/2009 08:00	07/01/2009 17:00	900	R7=8	9	900	fenetre N°13
8	1	7/1/09 21:00	7/1/09 22:00	100	R10=6	4	400	fenetre N°14
9	7	8/1/09 7:00	8/1/09 14:00	700	R9=18	9	900	fenetre N°15
17	8	08/01/2009 15:00	08/01/2009 23:00	800	R10=6	1	100	fenetre N°16
10	3	9/1/09 14:00	9/1/09 17:00	300	R1=12,5	15	1500	fenetre N°17
18	2	09/01/2009 18:00	09/01/2009 20:00	200	R2=19	1	100	fenetre N°18
		Tableau3			insertion de 9 taches dynamiques			
							6100	