Algorithme d'arrête gauche adapté pour l'ordonnancement des tâches de maintenance dans un contexte de maintenance distribuée

Djalal HEDJAZI [1], Abdelmadjid ZIDANI [2], Samra DEY [3], Zouleikha TEJANI [4]
Département Informatique, Université de Batna
[1] djalal05@yahoo.fr, [2] zidani@free.fr, [3] etude1@free.fr, [4] etude2@free.fr

***Abstract*-Today, more that the industrial enterprises ever, long for to take advantage of the TIC, especially in the context of maintenance in order to reduce the time of inactivity of the installation. Our interest is centered therefore around the modeling of the concept of telemaintenance that is based on a system of resources distributed. In this article, we present our approach of resolution of the problem of scheduling and affectation of the maintenance tasks, inspired of an algorithm used in the synthesis of the circuits integrated for the allowance of the registers to the variables and constants.**

Keywords-Scheduling, Allocation, industrial telemaintenance

I. INTRODUCTION

Le maintien en état de fonctionnement de l'outil de production a toujours été la préoccupation majeure de tous les gestionnaires dans un monde industriel où les notions de réactivité, de coûts et de qualité ont de plus en plus d'importance, et où il est vital de pouvoir s'appuyer sur un système de production performant à tout instant.

Dans ce contexte de juste à temps, la disponibilité des équipements, au moment voulu, est une condition nécessaire au bon déroulement de la production.

Souvent les problèmes de maintenance des installations industrielles sont réglés au second plan face aux impératifs de la production. De nos jours, avec les avancées technologiques dans le domaine de l'information et de la communication (TIC), la notion d'entreprise locale est passée vers celle d'entreprise étendue. Dispersée géographiquement les entreprises sont connectées par le biais des flux d'informations qui circulent entre ses composantes via les réseaux informatiques.

Partant de ce constat, nous se placerons dans le cas de la maintenance externalisée, c'est à dire que la fonction de la maintenance est effectuée soit par une entreprise extérieure qui peut assurer ce travail pour plusieurs entreprises, soit par un service indépendant de l'entreprise qui peut travailler pour plusieurs sites de l'entreprise étendue. L'externalisation de la fonction maintenance dans de nombreuses entreprises conduit à la mise en place de plus en plus fréquente de système de télémaintenance [1, 14, 15]. Ce dernier est conçu comme un système permettant de rapatrier et de traiter toutes les informations techniques de la maintenance et du processus de production. La télémaintenance vise donc bien à exploiter et gérer à distance des machines du parc industriel en entreprise. Elle se base sur un système de ressources distribuées, dans lequel les ressources sont mutualisées. En fonction des opérations de maintenance et du niveau de compétence requis, les équipes de maintenance ne sont pas toujours disponibles instantanément, puisque toutes les compétences ne sont pas disponibles partout. La maitrise de l'ordonnancement est d'un intérêt capital pour les entreprises, sans cesse confrontées à des impératifs de productivité, de flexibilité et réactivité [7].

Dans ce contexte, le problème que nous abordons est un problème d'ordonnancement et d'affectation de ressources dans lequel les tâches sont les opérations de maintenance préventive systématique et les ressources sont les équipes de maintenance (locale ou distribuée). Les interventions de maintenance préventive ont un rôle prépondérant, puisqu'elles permettent au système de production de fonctionner de façon nominale. En effet, le coût engendré par une panne (impliquant une maintenance corrective, un arrêt non programmé de la production, des retards conséquents de livraison, etc.) est largement supérieur à celui d'un arrêt prévu de la production. Les besoins du service ordonnancement sont donc quelque peu antagonistes : pour satisfaire dans les délais les clients, il doit utiliser de façon optimale l'ensemble des installations. Mais il doit également prévoir les interventions de maintenance.

Partant de ces constatations, nous avons proposé une solution au problème d'ordonnancement et d'affectation des tâches de maintenance. Elle est inspirée d'un algorithme dit d'arrête gauche [8] utilisé pour faire l'allocation des ressources (registres aux différentes variables) dans la synthèse des circuits intégrés [9, 10 et 11]. L'adaptation de cet algorithme pour notre problème se faite par l'intégration de la technique de recherche des fenêtres optimales [5] proposé pour résoudre le problème d'ordonnancement dynamique des tâches de maintenance dans un système de télémaintenance.

L'objectif de cet article est donc de présenter en détail notre approche pour résoudre le problème. En effet, chaque solution à ce problème tire à résoudre deux sous problèmes: Chercher pour chaque équipe la meilleure séquence de tâches et chercher pour chaque tâche la meilleure date d'exécution. La section 2 passe en revue le problème d'ordonnancement et d'affectation des tâches de maintenance La section 3 donne des définitions préliminaires. La section 4 présente notre approche de

résolution du problème. La section 5 discute les différents résultats obtenus par la simulation de la solution proposée. Finalement, nous concluons en donnant quelques perspectives du travail accompli.

II. ORDONNANCEMENT ET AFFECTATION DES TACHES DE MAINTENANCE

La résolution d'un problème d'ordonnancement consiste à placer dans le temps des activités ou tâches, compte tenu de contraintes temporelles (délais, contraintes d'enchaînement, etc.) et de contraintes portant sur l'utilisation et la disponibilité des ressources requises par les tâches [7, 2].

Dans la plupart des travaux 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. Abdul-Razaq [3] propose une méthode de résolution utilisant un algorithme de séparation et évaluation dans lequel les bornes de l'évaluation sont obtenues par une procédure de programmation dynamique relaxée. Leurs résultats prouvent que le problème consomme un temps de calcul important en dépassant 25 tâches. Plus tard, Ibaraki [16], à amélioré la procédure ; ils à proposé la méthode SSDP (Successive Sublimation Dynamic Programming) pour résoudre ce problème. Les expériences ont montré, que cette méthode peut résoudre les problèmes avec 35 tâches.

Gagne [4], à proposé une heuristique, basée sur un algorithme de colonie de fourmis. Le problème, qu'il traite est le problème avec les temps de réglages, dépendants de la séquence et la fonction à minimiser est le retard total. Il trouve, que l'algorithme est assez efficace du point de vue de la qualité de solutions, ainsi que de point de vue du temps de calcul. De plus ils proposent une comparaison de l'algorithme avec d'autres heuristiques.

Fry [17], décrit une procédure de séparation et d'évaluation pour la minimisation la somme moyenne de l'avance et du retard. Il remarque que l'algorithme peut résoudre les problèmes avec 20 tâches.

Ivanov [6], à proposé une méthode de résolution du problème d'ordonnancement et d'affectation dans les systèmes de télémaintenance, basée sur des algorithmes de descente stochastique et de méthode Kangourou. ses résultats montrent, que la méthode permet de résoudre les problèmes assez larges en temps polynomial. Un peut plus tard, le même auteur [5], à proposé une solution pour le problème d'ordonnancement dynamique dans le contexte d'un système de télémaintenance multi-compétences. L'algorithme proposé permet alors d'insérer des tâches de maintenance correctives avec un minimum de changement du planning initial.

Le problème, que nous abordons dans cet article, est un peu similaire par rapport aux deux derniers. Nous le traitons avec plusieurs équipes de maintenance de même compétence, chacune d'elles possède ses propres ressources (humaines ou matérielles). Le problème est NP-complet. Afin de le résoudre, nous avons proposé une nouvelle solution basée sur deux algorithmes utilisés dans deux domaines différents, le premier dit algorithme d'arrête gauche [8], appliqué pour résoudre le problème d'allocation de ressources (allocation des registres aux différentes variables et constantes) dans un système de synthèse de haut niveau [9, 10 et 11] (synthèse des circuits intégrés). Le second algorithme est celui de la fenêtre optimale [5].

III. DEFINITIONS PRELIMINAIRES

A. *Tâche de maintenance*

Chaque tâche de maintenance est caractérisée par sa durée pi, sa date de début au plus tôt ri et sa date de fin au plus tard di [7].

Le coût d'une tâche de maintenance est représenté sur la Fig.1.

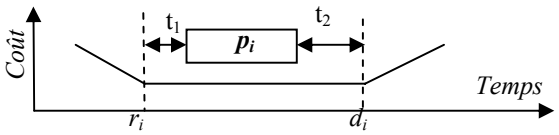

Fig. 1. Modélisation du coût d'une tâche de maintenance.

La période de temps entre ri et di est appelée fenêtre d'optimalité de la tâche pendant laquelle le coût d'exécution de la tâche est minimal. Si on se place devant la stratégie de maintenance préventive systématique, L'explication de la courbe est double. D'un coté, si on intervient avant la date ri, le coût de la maintenance augmente. Cela signifie que l'on réalise trop souvent les tâches de maintenance préventive. De l'autre coté, si on dépasse la date de fin au plus tard, le risque que l'équipement concerné tombe en panne augmente, ce qui nécessite des tâches de maintenance corrective. Cette dernière engendre des coûts supplémentaires liés à l'arrêt de la production.

B. *Fenêtre élémentaire et fenêtre maximale*

La fenêtre élémentaire entre deux tâches Ti et Ti+1 représente l'espace entre ces tâches (Fig. 2.).

$$F_e^{[i][i+1]} = [r^{\pi}_{[i]} + p_{[i]};\ d^{\pi}_{[i+1]} - p_{[i+1]}] \quad (1)$$

Une fenêtre maximale *Fmt* pour une date *tmps*, est la fenêtre la plus longue parmi les fenêtres élémentaires, contenant la date *tmps*.

$$F_m^{[t_i;t_{i+1}]} = max\ (F_e^{[i][i+1]})\ \forall i,i+1 \in \pi,\ t \in F_e^{[i][i+1]} \quad (2)$$

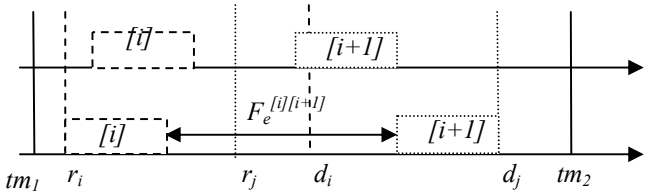

Fig. 2. Fenêtre élémentaire et fenêtre maximale

C. *Meilleure fenêtre élémentaire d'une tâche*

Une fenêtre élémentaire est dite meilleure pour une tache T_i est celle qui peut contenir la tâche T_i avec un coût global minimal.

IV. NOTRE APPROCHE

A. *Hypothèses*

Pour formuler le problème, nous avons adopté les hypothèses suivantes :

- Nous supposons, qu'il faut surveiller des équipements sur plusieurs sites de production.
- Nous nous positionnons dans le cas de la maintenance préventive systématique.
- Les équipes de maintenance ont le même niveau de compétence.
- Chaque équipe de maintenance possède ses propres moyens de réparation (outils de maintenance, etc.)
- Les dates d'intervention et les durées de tâches sont calculées sur la base de l'analyse statistique des pannes.
- La gestion des interventions est assurée par le centre de compétence [1, 12, 13]

B. *Objectifs*

Le coût des activités de maintenance est la somme des coûts de toutes les tâches :

$$COUT\_GLOBAL = \Sigma\, Cout_i \quad (3)$$

Dans notre approche, nous avons adopté la fonction objective permettant la minimisation de la somme des retards des tâches.

$$\Sigma\, Cout_i = \Sigma\, Retard_i \quad (4)$$

$Retard_i$ d'une tâche T_i représente l'intervalle du temps qui sépare la date de fin au plus tard d_i et la date d'exécution de la tâche re_i (date début d'exécution de la tâche qui est obtenue par l'ordonnancement)

Notre fonction objectif est donc exprimée par :

$$F=Min\ COUT\_GLOBAL \quad (5)$$

C. *Solution proposée*

La solution proposée reçoit en entrée une liste L contenant les tâches de maintenance avec leurs caractéristiques (r_i, d_i, p_i). Elle effectue en premier lieu un prétraitement sur la liste L qui peut être soit ***Trie****(L, Croissant)* qui représente le trie en ordre croissant par rapport à début de tâche (r_i). Si deux tâches en le même r_i, le tri se fait en ordre croissant par rapport à fin de Tâche (d_i), ou ***Trie****(L, Décroissant)* qui représente le trie de la liste ***L*** en ordre croissant par rapport à début de tâche (r_i). Si deux tâches en le même r_i, le tri se fait en ordre décroissant par rapport à fin de Tâche (d_i). Cette première étape nous crie deux versions de la solution. La première avec ***Trie****(L, Croissant)* et la seconde avec ***Trie****(L, Décroissant).*

En deuxième lieu, la solution fait plusieurs passes sur les éléments de cette liste, jusqu'à ce que toutes les tâches soient assignées aux équipes de maintenance. Une équipe peut intervenir pour exécuter plusieurs tâches, à condition que les dates de celles-ci ne chevauchent pas.

La première passe sert à affecter les tâches qui ne se chevauchent pas aux différentes équipes de maintenance. Les autres passes permettent d'affecter les tâches qui ne sont pas encore affectées. Pour ce faire, nous avons appliqué deux méthodes. Ceci nous donne deux autres versions de la solution. La première, place les tâches non affectées à la fin des plannings des équipes d'une manière à chercher pour chaque tâche la meilleure équipe (celle qui minimise le coût global). La seconde, fait la recherche du meilleur emplacement qui est associé à la meilleure fenêtre élémentaire (voir section III. C.).

L'objectif visé par le présent travail, est donc de choisir parmi les quatre versions de la solution proposé celle qui donne le meilleur ordonnancement (avec un coût minimal).

```
Trie(L, Croissant) ou Trie(L, Décroissant) ;
Equipe_Index ← 0 ;Cout_globale← 0 ;
Tant que (L ≠∅) et (Equipe_Index<nbre_Equipe)) Faire
   Equipe_Index← Equipe_Index+1 ;
   Curr_tache ← First(L) ;
   Last ← 0 ;
  Tant que (Curr_tache ≠ NULL) Faire
    Si (Debut(Curr_tache) >= Last) Alors
       Affecter_Tache(Equipe[Equipe_Index],Curr_tache) ;
       Last ← Fin(Curr_tache) ;
       Temp_tache ← Curr_tache ;
       Curr_tache ← Next(Curr_tache) ;
       Delete(Temp_tache) ;
    Sinon
       Curr_tache ← Next(Curr_tache) ;
    Fin Si
  Fin tant que
Fin tant que

Si (L≠∅) Alors
     Curr_tache ← First(L);
     Tantque (Curr_tache≠NULL) Faire
          Rechercher_Toutes_Fenetres(Curr_tache) ;
          Rechercher_Min_Fenetre(Curr_tache) ;
          Affecter_Tache_Fenetre(Curr_tache) ;
          Temp_tache ← Curr_tache ;
          Curr_tache ← Next(Curr_tache) ;
          Delete(Temp_tache) ;
     Fin tant que
Fin Si

Si (L≠∅) Alors
     Curr_tache ← First(L);
     Tant que(Curr_tache≠NULL) Faire
          Rechercher_Toutes_Fenetres(Curr_tache) ;
          Rechercher_Toutes_Pertcout(Curr_tache ,Fenetre) ;
          Rechercher_Min_Pertcout(Curr_tache,Fenetre) ;
          Affecter_Tache_Fenetre(Curr_tache) ;
          cout_globale ← Cout_globale +MINpertcout ;
          Temp_tache ← Curr_tache ;
          Curr_tache ← Next(Curr_tache) ;
          Delete(Temp_tache) ;
     Fin tant que
Fin Si
```

- Algorithme proposé -

- ***Recherche_Toutes_Fenetres (Curr_tache)*** : permet de chercher toutes les fenêtres élémentaires de la tache courante.
- ***Rechercher_Min_Fenetre(Curr_tache)*** : permet de donner la fenêtre la plus petite parmi les fenêtres élémentaires.
- ***Affecter_Tache_Fenetre (Curr_tache)*** : permet d'affecter la tache courant dans la meilleure fenêtre élémentaire.
- ***Rechercher_Toutes_Pertcout(Curr_tache, Fenetre)*** : permet de calculer a chaque fenêtre élémentaire le coût perdu.
- ***Rechercher_Min_Pertcout(Curr_tache, Fenetre)*** : permet d'obtenir la fenêtre qui donne le plus petit coût perdu.

V. RÉSULTATS

L'algorithme a été programmé en JAVA. Il a été testé sur un exemple avec quatre équipes de maintenance, plusieurs itérations (100 itérations) dont chacune génère N tâches. Les paramètres (p_i, r_i et d_i) de ces dernières sont générés aléatoirement.

Concernant le nombre de tâches (N) générées pour chaque itération, nous avons testé notre solution avec plusieurs valeurs de N (N = 10, 20, 40, 60).

Les résultats des tests sont donnés sur les figures (Fig. 3. Jusqu'à Fig. 6.). Elles montrent que l'application de la méthode de recherche de meilleure fenêtre augmente d'une manière considérable les performances de la solution à partir de (N >= 20).

D'après les résultats, nous constatons aussi que le trie des tâches (***Trie***(*L, Croissant)* ou ***Trie***(*L, Décroissant)*) n'influe pas beaucoup sur les performances de la solution. Dans certains cas l'algorithme donne des bonnes résultats avec le trie croissant (***Trie***(*L, Croissant)*) et dans d'autres cas les bonnes résultats sont donnés par l'algorithme intégrant le trie décroissant (***Trie***(*L, Décroissant)*).

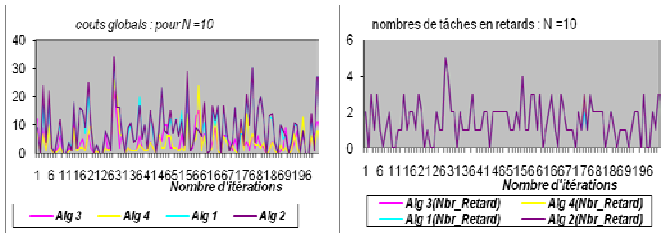

Fig. 3. Courbes des couts globales et nombre de tâches en retards pour (N=10 tâches)

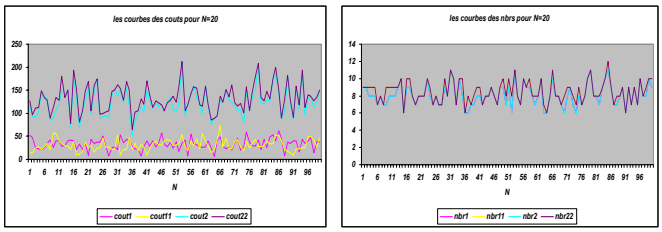

Fig. 4. Courbes des couts globales et nombre de tâches en retards pour (N=20 tâches)

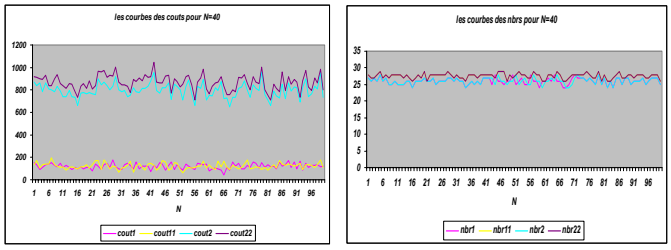

Fig. 5. Courbes des couts globales et nombre de tâches en retards pour (N=40 tâches)

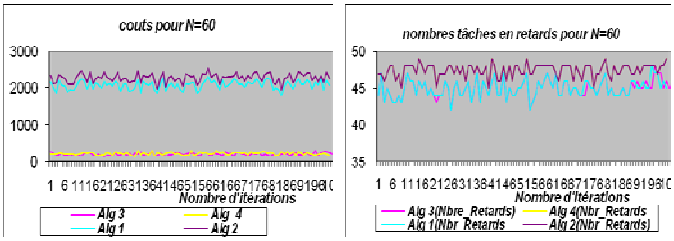

Fig. 6. Courbes des couts globales et nombre de tâches en retards pour (N=60 tâches)

Interprétation des couleurs des courbes

Alg 1 : Solution avec la méthode de trie croissant (***Trie****(L, Croissant)*) et sans application de la méthode de recherche des fenêtres élémentaires

Alg 2 : Solution avec la méthode de trie croissant (***Trie****(L, Décroissant)*) et sans application de la méthode de recherche des fenêtres élémentaires

Alg 3 : Solution avec la méthode de trie croissant (***Trie****(L, Croissant)*) et avec application de la méthode de recherche des fenêtres élémentaires

Alg 4 : Solution avec la méthode de trie croissant (***Trie****(L, Décroissant)*) et avec application de la méthode de recherche des fenêtres élémentaires

VI. CONCLUSION ET PERSPECTIVES

Dans ce papier, nous avons proposé une solution permettant de résoudre le problème d'ordonnancement et d'affectation des tâches de maintenance. Cette solution est inspirée d'une technique dite d'arrête gauche utilisée pour l'allocation des ressources dans la synthèse des circuits intégrés [9, 10, 11] (synthèse de haut niveau). Les résultats de la programmation et tests de la solution avec les quatre variantes proposées ont été discutés.

Finalement comme perspectives, nous proposons de comparer la meilleure solution parmi celles qu'on à proposé dans ce papier avec quelques solutions proposées par la communauté scientifique dans ce domaine.

ACKNOWLEDGMENT

M. Djalal HEDJAZI prépare une thèse de doctorat à l'université de Batna (Algérie). Titulaire d'un Magister en informatique, son sujet de doctorat concerne la conception d'environnements virtuels dédiés à la télémaintenance industrielle coopérative sur Internet.

M. Abdelmadjid ZIDANI est maître de conférences à l'université de Batna (Algérie) où il est responsable de l'école doctorale du pôle est. Ses domaines d'intérêt portent sur les systèmes coopératifs et multimédias et sur les interfaces homme-machine.

M^{elle}. Samra DEY et M^{elle}. Zouleikha TEJANI préparent leurs ingéniorats en informatique à l'université de Batna (Algérie). Leur sujet d'ingéniorat concerne la simulation d'une technique d'ordonnancement des tâches de maintenance

REFERENCES

[1] Kafel H., D'Amours S. et Ait-Kadi D., *The Concept of Distributed Maintenance*, 29th International conferences of Computers & Industrial Engineering, November 1-3, Montreal Canada, 2001.

[2] P. Lopez and F. Roubellat, editors. *Ordonnancement de la production*. Hermès Science Publications, Paris, 2001.

[3] Abdul-Razaq T., Potts C., 1988, *Dynamic programming state-space relaxation for single-machine scheduling*, J.Opnl. Res. Soc., vol. 39, pp. 141-152.

[4] C. Gagne, W.L. Price, M. Gravel, *Scehduling a single machine with sequence dependent setup time using ant colony optimisation*, Faculté des sciences de l'administration, l'Université Laval, 2001.

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

[6] A. Ivanov, C. Varnier, N. Zerhouni, *Ordonnancement des activités de maintenance dans un contexte distribué*, MOSIM'03, Toulouse (France), Avril 2003.

[7] P. Esquiro, *L'ordonnancement*, ed. Economica, Janvier 1999

[8] C. J. Tseng et D. P. Seiwiorek, *"Automated Synthesis of Data Paths on Digital Systems"*, IEEE Transaction on computer-aided design of integrated circuits and systems, vol.CAD-5, N° 3, pp. 379-395, 1986

[9] D. Hedjazi, M. Benmouhammed, *CS03 : Outil de Synthèse de Haut Niveau pour Contrôleurs Dédiés*, CIP'2003, Alger, Algérie, octobre 2003.

[10] D. Hedjazi, M. Benmouhammed, *Intégration d'un simulateur comportemental au CS03 : Outil de Synthèse de Haut Niveau pour Contrôleurs Dédiés*, CIIA'2005, Bordj Bou Airiridj, Algérie, Novembre 2005.

[11] D. Hedjazi, M. Benmohammed, *Simulateur comportemental pour contrôleurs dédiés*, CIIE'06, Oran, Algérie, Mai 2006.

[12] D. Hedjazi, A. Zidani, M. Boussedjra, *Proposition d'un protocole de gestion d'un groupe d'experts pour la télémaintenance industrielle coopérative*, journée d'étude sur la maintenance industrielle (JOSIM07), Oran, Algérie, mars 2007.

[13] D.Hedjazi, A. Zidani, M. Boussedjra, L. Bouzgou, D. Ghellab, *Maintenance industrielle assistée par les TIC : Gestion de l'information,* Séminaire sur les techniques et le management de la maintenance STMM07, Alger, Algérie, mai 2007.

[14] C. Kolski, P. Millot, *Decision aid criteria to integrate a telemaintenance tool into the maintenance man-machine system*, In: Advances in Industrial Ergonomics and Safety IV, Taylor & Francis, London, 1992, pp. 51-58.

[15] C. Kolski, P. Millot, *Problems in telemaintenance and decision aid criteria for telemaintenance system design*, in International Journal of Industrial Ergonomics, 1993.

[16] Ibaraki T., Nakamura Y., *A dynamic programming method for single machine scheduling*, European J. of Opnl. Res., 1994,vol. 76, , pp. 72-84.

[17] Fry T., Darby-Dowman K., Armstrong R., *Single machine scheduling to minimize mean absolute lateness*, Computers and Operations Research, 1989.