

Etude du problème d'ordonnancement d'Open-shop par les règles de priorités au niveau des postes : approche par simulation

N Ghazi¹, M Bakalem² & M. Tadjine¹,

1. Laboratoire de Commande des Processus (LCP), 2. Département Génie Industriel, Ecole Nationale Polytechnique d'Alger.
E-mail(s) : ghazi_nawel@yahoo.fr, Mohammed.bakalem@gmail.com, tadjine@yahoo.fr

Résumé- Dans ce présent travail, nous traitons un problème d'ordonnancement du type Open Shop dans un environnement d'atelier flexible. Le problème étudié est $Om| \tau_j | F$. Notre objectif est la réduction du temps total de séjours des pièces dans un système cyclique conçu autour d'un convoyeur en boucle, ceci est fait par génération de plusieurs règles de priorité au niveau des postes associé à des plans non linéaires non prédéterminés. Les temps de transfert entre les stations sont considérés. Du fait de la complexité NP-difficile au sens fort du problème Open Shop considéré, la nature dynamique des règles d'ordonnancement testées et la typologie circulaire de la cellule étudiée (rendant le problème « plus combinatoire » par rapport aux lignes d'assemblage linéaires conventionnelles) nous avons opté pour une approche purement expérimentale moyennant la simulation.

I. INTRODUCTION ET PROBLEMATIQUE

Une étude sur l'apport d'un arrangement circulaire aux performances des lignes d'assemblage linéaires a été effectuée par Finke et Dupont [1]. Ils sont arrivés à la conclusion que les performances d'un système d'assemblage cyclique sont conditionnées par le robot goulot et que son pilotage par des plans non linéaires (Job Shop) n'apportait aucune amélioration significative aux performances du système, mais que l'intérêt est dans la manipulation des plans d'assemblage ouverts d'où l'acquisition d'une flexibilité dans l'ordre de passage des tâches (l'*open-shop*) où chaque ordre des opérations d'assemblage est possible. Ils ont montré que l'apport d'une combinaison plans non linéaire-pilotage FIFO à la performance du système est insignifiant (inférieur à 5%).

En partant de ce constat, on se propose d'étudier le comportement d'un tel système géré par des plans non linéaires et non prédéterminés, combinés à un ordonnancement judicieux. Dans ce cas, en plus de la détermination de l'ordre de passage des travaux sur chaque machine (ce qui constitue un problème d'ordonnancement) nous devons aussi déterminer l'ordre de passage sur les machines pour chaque travail c'est-à-dire le routage. L'objectif est de trouver différentes stratégies d'ordonnancement et d'analyser leurs impacts sur les performances du système.

Le problème d'ordonnancement que nous nous proposons d'étudier est du type Open Shop ou à cheminements libre et notre fonction objectif est la minimisation du temps total de séjours avec considération des temps de transport, il relève de

la classe : $Om| \tau_j | F$ selon la classification de Graham & al. (1979).

Les problèmes Open Shop sont déjà NP-difficiles dans le cas général et l'ajout de ressources, en plus des ressources de transformation (les machines), rend le problème plus difficile [2]. Même si la préemption est autorisée, ce problème est déjà NP-difficile. Dans notre cas, la préemption est non autorisée (ressources disjonctives). La typologie circulaire rend le problème « plus combinatoire » par rapport aux lignes d'assemblage linéaires conventionnelles. De plus, les temps de transport sont considérés (considération des ressources de transport) ce qui fait de notre problème un problème NP-difficile au sens fort. Du fait de cette complexité, nous avons opté pour une approche empirique de prise en charge du problème considéré $O|\tau_j|F$. C'est-à-dire que notre étude est établie sur un programme expérimental pré-généré. Pour ce faire nous avons opté pour un outil d'aide à la décision à savoir : la simulation.

La mise en œuvre du processus de simulation est réalisée en dix étapes tel c'est défini par PRITSKER [3]. Ces étapes constituent un processus itératif qui, très souvent, n'est pas rigoureusement séquentiel [4].

II. PRESENTATION DE LA CELLULE D'ASSEMBLAGE

La cellule d'assemblage cyclique étudiée est détaillée dans la figure 1. Elle est constituée de :

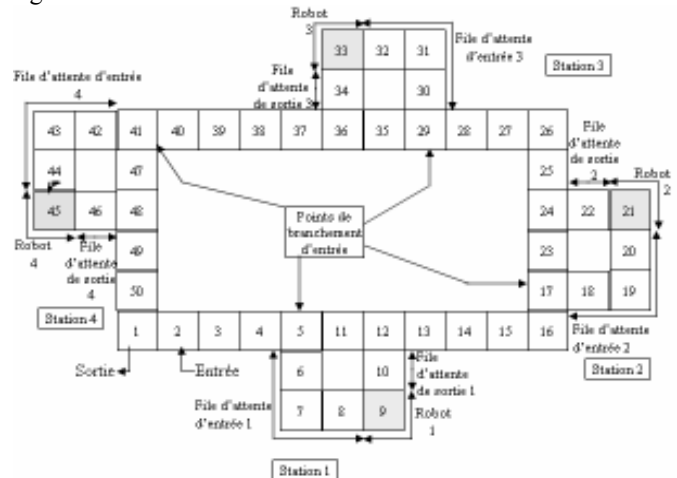


Fig. 1. Modélisation du système cyclique d'assemblage

Les stations de travail sont arrangées dans un ordre circulaire autour d'un convoyeur en boucle non bloquant à vitesse régulière. Les pièces à assembler sont toujours montées sur des palettes. Ces palettes sont associées aux pièces à assembler tout au long de leur présence dans le système et elles ne sont déchargées qu'au moment de la sortie des pièces assemblées (produit fini) du système. Ces palettes circulent en permanence sur le convoyeur. Lorsqu'une palette arrive à un point de branchement d'une station, un dispositif de lecture vérifie si la prochaine opération à effectuer est dans la station considérée, si c'est le cas, la palette pénètre dans la file d'attente d'entrée si sa capacité n'est pas saturée. Dans les autres cas, elle continue à circuler. À la fin de l'assemblage d'un produit fini ou semi-fini, la palette correspondante est déchargée au niveau de la sortie du système et tout de suite réintégré à l'entrée du système pour une réutilisation. Le flux des palettes à travers le système est représenté sous forme d'un organigramme dans la figure 2. Dans cette configuration, la priorité est au convoyeur c'est à dire que les palettes ne quittent pas les boucles périphériques si les sorties des stations sur le convoyeur sont occupées. La capacité des files d'attente d'entrée est d'au moins de deux unités. La capacité des files d'attente de sortie est d'au moins d'une unité pour éviter tout blocage [1].

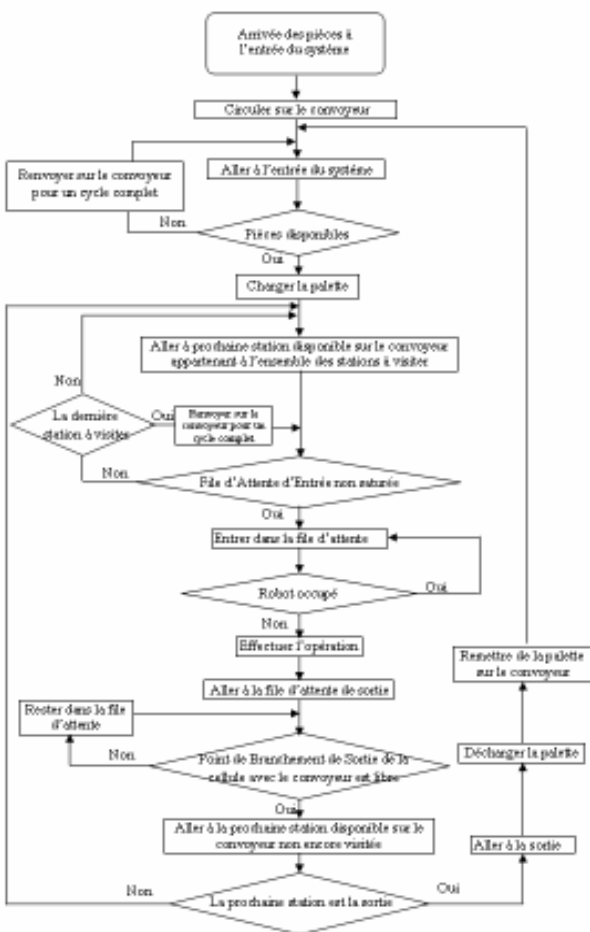


Fig. 2. Le cheminement d'une palette à travers le système cyclique

III. LA TRANSCRIPTION INFORMATIQUE DU MODELE PAR ARENA 7.1

Suivant la description précédente du système étudié, nous avons conçu un programme sous Rockwell ARENA 7.1. Dans le but de modéliser à tout instant, un nombre limité de palettes dans la Cellule (par la suite limitation du nombre de pièces), nous avons préféré l'utilisation du Data module **Transporters** qui nous permettra d'imposer le nombre d'unités transporteur circulant simultanément dans la cellule. Chacune des unités transporteur modélise une palette.

Les différentes conditions sont présentées par des Blocks **Branch** aux entrées et sorties des stations.

A. Modélisation de l'entrée du système : création des pièces et leurs entrées

L'entrée de la cellule est conçue tel que montré dans la figure 3.

Nous avons utilisé quatre combinaisons des modules **Create-Assign** afin de créer quatre différents types de pièces. Ils représentent nos quatre travaux considérés dans cette étude : *Pièce 1, Pièce 2, Pièce 3 et Pièce 4*.

Grâce au module **Assign**, nous avons affecté différents attributs aux entités créées.

- Le premier attribut *Piece Index* représente l'index de l'entité créée.
- Le second attribut *TimeIn* représente le moment d'entrée dans le système.
- Le troisième attribut *NBE* représente le nombre d'opérations (étapes) d'assemblage effectuées. C'est une variable initialisée à zéro et qui sera ensuite incrémentée à chaque étape d'assemblage d'une pièce c'est-à-dire à chaque sortie d'un module **Process**.
- les quatre attributs, suivant *DO1, DO2, DO3, et DO4* représentent les différentes durées opératoires assignées à l'entité. Ces durées sont associées aux robots des stations portant le même indice.
- un huitième attribut *DO* représente la somme des durées opératoires associées à une entité donnée et qui sera mis à jour à chaque étape d'assemblage (soustraction de la durée effectuée).
- le dernier attribut associe une image propre à l'entité

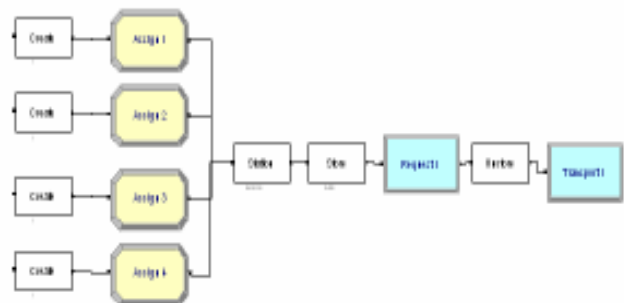


Fig. 3. Entrée des pièces dans la cellule

Une fois les entités créées, nous les stockons dans une file d'attente d'entrée nommée FAE par le module *Store* où elles y restent jusqu'à ce qu'une palette libre Arrive (*Request*). Dès que la palette est arrivée l'entité est directement mise en mouvement vers la première station (*Unstore-Transport*). Lorsque une entité est associée à une palette (transporteur), ARENA leur attribue un seul et unique indice c'est-à-dire qu'ils sont confondus.

B. Modélisations d'une Station

La cellule est composée de quatre stations, chacune contenant une file d'attente d'entrée, un robot, une file d'attente de sortie. Une station est modélisée en trois étapes : l'entrée de la station (figure 4); le Robot ainsi que les deux files d'attente d'entrée et de sortie (figure 5) et la sortie de la station (figure 6).

1. Modélisation de l'entrée d'une Station

Afin de suivre la logique de comportement d'une organisation d'atelier Open Shop, une série de tests d'entrée est effectuée à chaque entrée de station dans le but de vérifier si l'entité doit être traitée dans cette station ou pas (le premier module *Branch*).

Selon la station *i*, nous vérifions si l'attribut durée opératoire associé *DOi* est nul ou pas par les conditions *IF DO1 > 0*, *IF DO2 > 0*, *IF DO3 > 0* et *IF DO4 > 0*. Si la condition est vérifiée alors nous examinons le nombre de pièces en attente dans la file d'attente d'entrée de la station *i* par la condition : *IF NQ(FAEi) < 4* (*Branch 1* : 1ère condition). Dans ce cas où la capacité de la file d'attente (FAEi) est de quatre, si la condition est vérifiée alors la pièce est mise dans la file d'attente et traitée dès que le robot est libre, si non elle est renvoyée vers les autres postes où elle n'a pas été traitée (*Branch 1* : 2ème, 3ème et 4ème conditions). Dans le cas où cette opération est la dernière dans le processus d'assemblage de la pièce considérée, alors elle est automatiquement renvoyée sur le convoyeur pour y faire un cycle complet et revenir à nouveau dans cette station.

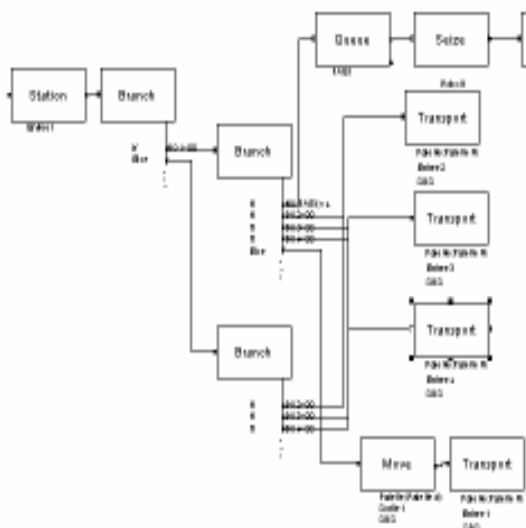


Fig. 4. Modélisation de l'entrée d'une station

2. Modélisation de la Station d'Assemblage

Une fois l'entité admise dans la file d'attente, elle y reste jusqu'à ce que le robot soit libéré. Selon la règle de priorité appliquée au niveau de la file d'attente, la pièce prioritaire est traitée (*Queue, Seize, Move, Station, Process*). Dès que le traitement de la pièce est fini, elle est mise dans une file d'attente de sortie FASi (*Store*) et la durée opératoire effectuée est tout de suite assignée à zéro pour marquer le passage de la pièce dans cette station (*Assign* : *DOi=0*).

3. Modélisation de la Sortie d'une Station

Une fois l'entité est dans la file d'attente de sortie, elle n'est sort que si le point de branchement entre la sortie de la station et le convoyeur n'est pas occupé sachant que la priorité revient aux pièces sur convoyeur (*Branch*). Si ce point de branchement est libre alors la pièce est tout de suite remise sur le convoyeur central si non elle reste dans la file d'attente de sortie de la station (*Move, Station*). La prochaine station est choisie selon les postes déjà visités (*Branch*: les trois premières conditions), si c'est la dernière alors la prochaine station est la sortie du système (*Branch*: la dernière condition).

C. Modélisation de la Sortie de la Cellule d'Assemblage

La sortie est représentée dans la figure 7.

A la fin de l'assemblage d'une pièce, elle est tout de suite orientée vers la sortie du système. La sortie du produit est modélisée par la destruction de l'entité représentative (*Dispose*). A la sortie de chaque entité, la variable *NumberOut* est incrémentée de 1 par le block *Assign* (*NumberOut==NumberOut+1*), ainsi la simulation est arrêtée dès que cette variable atteint la taille d'une commande. Dans le cas d'une commande de 120 pièces, la condition d'arrêt de la simulation, portée au niveau *Run/Setup/Replication Parameters/Terminating Condition*, est *NumberOut == 120*.



Fig. 5. Modélisation de la station (FAE, Robot, FAS)

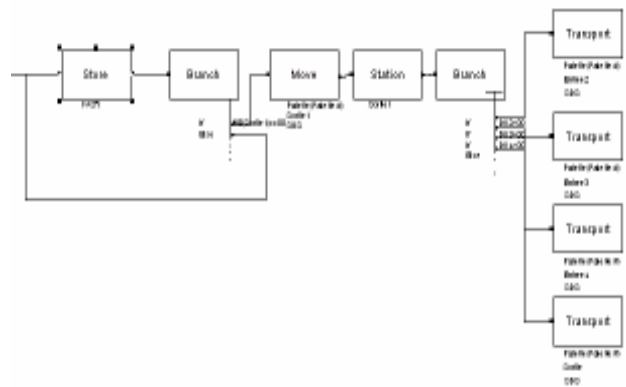


Fig. 6. Modélisation de la sortie d'une station

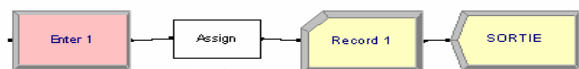


Fig. 7. Sortie de la cellule d'assemblage

IV. LA PLANIFICATION STRATEGIQUE ET TACTIQUE DE LA SIMULATION

Après les étapes de modélisation et de transcription informatique, cette étape est, sans doute, la plus critique et la plus délicate du processus de simulation. Selon l'objectif, arrêté, de l'étude, l'approche n'est pas toujours simple à mettre en oeuvre car les paramètres à considérer lors de l'étude de la cellule d'assemblage sont très nombreux et chacun d'eux peut avoir son influence sur la performance globale du système. Ceci constitue une situation combinatoire qui pourrait « exploser » exponentiellement. A titre d'exemple, nous pouvons étudier l'influence des paramètres suivants : La longueur du convoyeur central, le nombre de stations, la capacité des files d'attente d'entrée, le nombre de palettes, le nombre de tâches, la tailles des commandes, la fréquence d'entrée des pièces dans la cellule, ... (chaque paramètre pouvant faire l'objet de variations diverses). Dans la pratique, il n'existe pas de méthodes directes permettant de définir de façon optimale les types et le nombre d'expériences à mener en vue de bien cerner un problème ou un système étudié. Toutefois, une planification judicieuse des expériences à mener pourrait permettre une couverture suffisante du problème. Ceci peut se faire par recensement des paramètres à varier d'une part, et la détermination de l'intervalle et le pas de variation pour chaque paramètre d'autre part.

Pour ce faire, nous avons commencé par arrêter une configuration de base de la cellule, en définissant : la dimension du convoyeur, le nombre de stations, et le nombre de palettes. Conformément aux hypothèses arrêtées lors de la formulation, la dimension du convoyeur sera un multiple du nombre de palettes.

Pour chaque expérience, nous avons considéré :

- ☞ les durées opératoires sur les robots qui sont fonction des types de pièces traitées
- ☞ les dimensions des Files d'Attente d'Entrée (FAEs)¹ (possibilité de réduction des temps de transfert sur le convoyeur central en augmentant les capacités des files d'attente, et par conséquent de réduction des temps de séjours) ;
- ☞ les règles de priorité (possibilité de réduction des temps d'attente des pièces par un ordonnancement plus efficace au niveau des stations).

Nous testerons, également, des règles de priorité, pour en déduire des conclusions générales, et choisir une stratégie d'application de ces règles propre au contexte de l'application. Nous avons choisi les règles de priorité minimisant les temps de séjours des pièces dans la cellule ; c'est-à-dire minimisant

$$\sum_{i=1}^n (c_i - r_i) \text{ où } c_i \text{ est la date fin de l'opération } i, r_i \text{ est la date}$$

d'arrivée de l'opération i et n est le nombre d'opérations à ordonnancer [5]. Les règles choisies sont :

La SPT (Shortest Processing Time), la LPT (Longest Processing Time), la MWKR (Most Work Remaining First), et la LWKR (Least Work Remaining First).

REMARQUE : les deux règles la LWKR et la MWKR sont appliquées en deux étapes :

- ☞ MWKR1 et LWKR1 selon la durée d'exécution totale restante,
- ☞ MWKR2 et LWKR2 selon le nombre d'étapes d'assemblage restantes.

La configuration arrêtée pour l'expérimentation est une cellule composée de quatre stations, dont le nombre de palettes est de 20 et la longueur du convoyeur est de 500 palettes.

Dans un souci de maintien de l'équilibre entre les stations (Line balancing), les durées opératoires ont été choisies de sorte que les différences entre les durées des travaux (jobs) ne dépassent pas les 25 %.

Finalement le problème considéré est $O4|\tau_j|F$, tel que τ_j représente les temps de transport entre les stations et F le temps de séjour total (Total Flow Time).

V. ANALYSE ET INTERPRETATION DES RESULTATS

Conformément au plan d'expérimentation arrêté précédemment, nous avons mis en oeuvre les différentes configurations et relevé les résultats que nous avons analysés par la suite. Nous avons étudié la performance globale induite par l'application des règles d'ordonnancement. Pour ce faire, nous avons étudié les temps de séjour dans la cellule d'un ensemble de pièces correspondant à une commande c'est à dire la minimisation du C_{max}

Le nombre de pièces arrêté est de 120 (dans notre cas, c'est le nombre maximal de pièces autorisé par la version académique d'Arena 7.1). Pendant cette phase d'analyse, pour chaque expérience, nous calculerons les taux d'amélioration entre les résultats obtenus en FIFO et ceux obtenus par les autres règles. Une amélioration correspond à une réduction du temps de séjour d'un ensemble de pièces correspondant à une commande. Les taux d'amélioration des temps de séjours d'une commande dans la cellule $\tau_{TSLR\grave{e}gle}$ sont calculés comme suit :

$$\tau_{TSLR\grave{e}gle} = \frac{(T_{SLFIFO} - T_{SLR\grave{e}gle})}{T_{SLFIFO}} * 100 \quad (1)$$

tels que : $T_{TSLFIFO}$ est le temps de séjours d'une commande sous un pilotage FIFO ;

$T_{TSLR\grave{e}gle}$ est le temps de séjours d'une commande sous un pilotage de la Règle utilisée.

Conformément à la figure 9, les taux d'amélioration des temps de séjours d'une commande (un ensemble de pièces) dans la cellule sont insignifiants. La moyenne des taux de réduction ne dépassent pas 0.5% avec un écart type de l'ordre de 2%, principalement obtenu dans les cas dont la capacité des FAEs est de 5 et 10 pour des ordonnancements de LWKR2 et de MWKR1. La moyenne des taux d'amélioration dans les cas ordonnancés par les règles SPT, LPT, LWKR1 et MWKR2 est négative, ce qui signifie qu'il y a plus de détériorations que d'améliorations. Pour les six règles d'ordonnancement, les taux

¹ Par la suite les Files d'Attente d'Entrée dans les stations sont nommées FAEs.

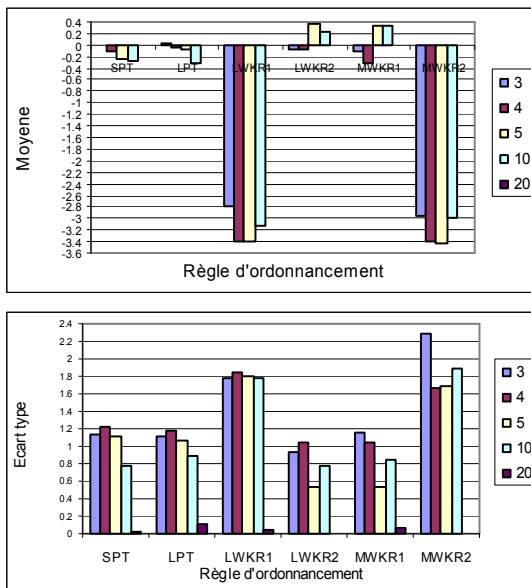


Fig. 9. Moyenne des taux de séjours d'une commande dans la cellule

d'amélioration les plus faibles sont obtenus dans les configurations dont la capacité des FAEs est de 20 (souvent nuls). Nous déduisons de ce qui précède que pour chacune des règles, l'impact des capacités des files d'attente sur le Cmax est quasiment inexistant.

Dans les cas dont la capacité des FAEs est de 20, nous retrouvons un comportement Flow Shop car toutes les palettes passent par les stations selon l'ordre de disposition des stations (capacité d'absorption d'une FAEi toutes les palettes à la fois). Les règles LRWK2 et MRWK2 perdent leurs sens, car dans un poste d'assemblage, le nombre d'étapes d'assemblage effectuées est le même pour toutes les pièces. Les pièces sont d'ailleurs traitées en FIFO. Les deux règles SPT et LWKR1 sont quasiment identiques car les deux favorisent les durées opératoires minimales de même pour la LPT et la MWKR1, qui favorisent les durées opératoires maximales.

VI. CONCLUSION

L'ordonnement d'un atelier open-shop avec prise en compte des ressources et des temps de transfert est un problème NP-difficile au sens fort. Dans le cadre de cette étude, nous avons examiné l'impact d'une combinaison plans non linéaires non prédéterminés avec des règles de priorité, sachant qu'il a été prouvé que la règle FIFO n'améliorait pas la performance du système. Pour ce faire, nous avons opté pour une démarche empirique basée sur la simulation.

Après une multitude de tests, nous avons constaté qu'aucune amélioration n'est réellement apportée à la performance globale de la cellule d'assemblage. Les règles d'ordonnement n'apportent aucune amélioration significative au temps d'assemblage d'une commande Cmax (la moyenne des taux d'amélioration pour les cinq expériences ne dépassent pas 0.4% avec un écart type de l'ordre de 2% obtenu dans les cas dont la capacité des FAE est de 5 et 10 par des ordonnancement LWKR2 et la MWKR1).

L'étude des temps de séjour moyens et des paramètres influents sur la performance de la cellule fera l'objet de futurs travaux.

VII. REFERENCES

- [1] G. Finke, L. Dupont, "Combinatorics of Cyclic Assembly Systems," *IEEE Systems, Man and Cybernetics Conference Proceedings*, Grenoble, France, 1993.
- [2] C. Gueret, "Problèmes d'ordonnement sans contraintes de précédence," Thèse de doctorat, Université de Technologie de Compiègne, France, 1997.
- [3] A.A.B. Pritsker, "Introduction to simulation and SLAM II," Halsted Press, New York, NY, 3rd Edition, 1986.
- [4] M. Bakalem, "Modélisation et simulation orientées objet des systèmes manufacturiers," Thèse de doctorat, ESIA, Savoie, France, 1996.
- [5] C. Chu et J.M. Proth, "L'Ordonnement et ses applications," Edition Masson, 1996.