

J. LEGRAS

J. M. PROTH

**Recherches de règles d'ordonnancement par
expérimentation sur ordinateur**

*Revue française d'automatique, d'informatique et de recherche
opérationnelle. Recherche opérationnelle, tome 5, n° V3 (1971),
p. 65-75.*

http://www.numdam.org/item?id=RO_1971__5_3_65_0

© AFCET, 1971, tous droits réservés.

L'accès aux archives de la revue « Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/legal.php>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

RECHERCHES DE REGLES D'ORDONNANCEMENT PAR EXPERIMENTATION SUR ORDINATEUR

par J. LEGRAS ⁽¹⁾ et J. M. PROTH ⁽²⁾

Résumé. — *Le but du travail que nous nous sommes proposé était double :*

- 1) *Construire une maquette reflétant aussi fidèlement que possible un ensemble d'ateliers.*
- 2) *Étudier, par simulation à partir de cette maquette, l'évolution de la fabrication en fonction des règles d'ordonnancement adoptées.*

Ces règles étaient de deux types : des règles paramétriques qui permettaient, en faisant varier les paramètres, de donner une importance plus ou moins grande aux facteurs déterminant les priorités accordées aux produits, et des règles non paramétriques (priorité accordée aux produits les plus urgents, par exemple).

Notre but était donc de procurer des informations permettant de décider d'une politique de lancement.

I. INTRODUCTION

Le problème que nous allons esquisser trouve son origine dans l'industrie sidérurgique (aciéries). La fabrication y est faite uniquement sur commandes et consiste, partant d'une certaine quantité de matière première, à lui faire subir une suite de transformations qui la conduisent jusqu'à l'état de produit fini. (Nous appellerons ces transformations opérations subies par la matière.)

La fabrication qui nous intéresse peut donc être qualifiée de linéaire, contrairement à la fabrication d'un appareil que nous appellerions convergente, car elle exige l'assemblage de pièces différentes fabriquées séparément ou encore par opposition à certaines fabrications qui se différencient en cours d'élaboration et que nous appellerions divergentes.

Notons cependant une particularité de la fabrication envisagée : la matière première provient des hauts fourneaux, et ceux-ci libèrent nécessairement des quantités de matière première multiples de 5 tonnes, si bien que l'on sera conduit à stocker l'excédent en cours d'élaboration, le stockage se faisant de préférence aux niveaux d'élaboration les moins élevés.

(1) Professeur à la faculté des Sciences de Nancy, directeur de l'Institut Universitaire de Calcul Automatique.

(2) Assistant à l'Institut des Sciences de l'Ingénieur de Nancy.

Le lancement d'une fabrication nécessitera donc d'abord la recherche de la matière première éventuellement disponible en stocks, d'où un fractionnement éventuel des commandes dès leur lancement.

En définitive, une commande est caractérisée par son délai, son poids, la qualité d'acier nécessaire, les différentes opérations à subir ainsi que le temps moyen nécessaire à chaque opération. Par ailleurs, chaque machine disponible peut effectuer plusieurs opérations différentes, à condition toutefois d'en changer l'outillage (exemple : rouleaux des trains de laminage).

Chaque machine sera donc caractérisée par les opérations qu'elle peut effectuer, les temps de préparation et le nombre d'ouvriers nécessaires aux différentes préparations et à l'exécution des différentes opérations.

Enfin, et pour chaque commande, on devra connaître les chutes moyennes en cours de fabrication ainsi que le poids unitaire et la forme de la matière première après chaque opération.

Nous sommes donc en présence d'un système constitué de machines, de commandes, de stocks et de matières en cours d'élaboration. Ce système évolue au cours du temps en fonction des décisions prises et des nouvelles commandes qui se présentent. Nous avons cherché, dans un premier temps, à préciser les questions qu'il y a lieu de se poser pour « définir » ce système.

II. POSITION DU PROBLEME

Connaissant l'état ξ_0 du système au temps t_0 , et en admettant que les commandes qui se présentent au cours du temps échappent à notre contrôle, il est facile de voir que l'état du système évolue au cours du temps en fonction des décisions prises. Ces décisions interviennent dès qu'une machine se trouve libérée, et répondent essentiellement aux questions suivantes :

- a) Utilisera-t-on les machines libres?
- b) Si oui, quelle opération va-t-on leur faire effectuer?
- c) Au profit de quelle commande?

Ces décisions étant prises au temps t_0 , le système évoluera jusqu'au temps t_1 , instant où la première machine occupée au temps t_0 , ou qui a fait l'objet d'une décision d'utilisation au temps t_0 , se trouve libérée. Il sera alors nécessaire de prendre de nouvelles décisions et le cycle recommencera.

Si bien que l'évolution du système peut se schématiser comme suit (D_i représente l'ensemble des décisions prises au temps t_i) :

$$\xi_0 \xrightarrow{D_0} \xi_1 \xrightarrow{D_1} \xi_2 \xrightarrow{D_2} \dots \xrightarrow{D_{i-1}} \xi_i \xrightarrow{D_i} \xi_{i+1} \xrightarrow{D_{i+1}}$$

Deux questions essentielles et intimement liées se posent à ce niveau :

- 1) Comment caractériser l'état du système.
- 2) Comment décider si un état est ou n'est pas favorable?

Disons de suite qu'il n'y a pas de réponse catégorique à la seconde question. L'expérience montre en effet que l'amélioration de certains aspects de l'état du système va de pair avec une évolution défavorable d'autres aspects. Par exemple, une amélioration des délais de fabrication (évolution favorable) est observée en même temps qu'une augmentation des en-cours (évolution défavorable).

Le problème qui se pose n'est donc pas un problème d'optimisation, et notre but est de permettre à l'utilisateur de tester les décisions prises.

Nous avons choisi, pour caractériser l'état du système, un certain nombre de fonctions que l'on peut diviser en deux groupes :

a) les fonctions qui caractérisent l'état du système à l'instant considéré : pour les commandes, les fonctions indiquant si elles sont sur carnet ou en cours, en attente ou en cours de transformation, quel est leur niveau d'élaboration, leur délai actualisé, pour les machines, les fonctions indiquant quel est leur état actuel (arrêt, préparation ou fonctionnement), éventuellement les temps de préparation ou de fonctionnement, etc.;

b) les fonctions qui caractérisent l'histoire de l'état du système : pour toutes les commandes arrivées en fin de fabrication au cours de l'expérience, la moyenne, pondérée par les poids, des différences (temps de fin de fabrication, délais), la répartition pondérée par les poids, des temps de fin de fabrication autour des délais, l'évolution au cours du temps des en-cours, des stocks, des ouvriers employés, des temps moyens d'immobilisation des machines.

Les listes qui précèdent, et en particulier les fonctions présentées dans b), ne sont pas exhaustives et sont établies en coopération étroite avec l'utilisateur. Un commercial cherchera à satisfaire, en moyenne, les délais, alors que le fabricant attachera plus d'importance à une diminution du temps moyen d'immobilisation de machines.

Nous retiendrons donc deux idées essentielles, l'état du système n'est caractérisé que partiellement, et la caractérisation qui en est faite doit répondre aux besoins de l'utilisateur. D'autre part, il est hors de question d'optimiser l'évolution des états du système.

Notre but sera donc, dans un premier temps, de donner à l'utilisateur un instrument qui lui permette d'essayer, de tester ses décisions. Pour cela nous avons construit une maquette dont nous allons parler maintenant.

III. MAQUETTE

La maquette que nous avons cherché à élaborer est un programme* qui construit une image dynamique du système étudié. Notre but a été d'exposer la conception et la description de ce programme de manière claire et facilement accessible. Pour cela nous avons utilisé une présentation par *niveaux* (de complexité) qui répond à un certain nombre de règles simples.

1) Le niveau I est la liste des différents modules de traitement rencontrés. Dans notre cas, il se présente de la manière suivante

Niveau I. Page 1

- 1) Sortie des renseignements nécessaires aux prises de décision à l'instant t .
- 2) Enregistrement des décisions.
- 3) Recherche de Δt .
- 4) Calcul des fonctions caractérisant l'état à l'instant $t + \Delta t$.
- 5) Enregistrement à l'instant $t + \Delta t$ des commandes passées entre t et $t + \Delta t$.

2) Le niveau II reprend ces différents libellés pour les préciser en divisant le traitement en un certain nombre de sous-modules.

3) Le niveau III précise à son tour ces différents sous-modules etc. Les précisions peuvent ainsi se poursuivre jusqu'au niveau ultime de l'instruction.

Chaque module ou sous-module est soit une « phrase » simple (ci-dessus), soit une phrase plus complexe, par exemple :

- 1) *Pour* (précision des différentes applications du traitement)
 - 2) module 1
 - 3) module 2

Ce type de phrase correspond, en programmation, à la notion de boucle.

On trouvera également des tables de décisions ainsi que des « phrases » du type :

- 1) Si (condition I)
 - 2) Module 1
 - 3) Module 2
- Sinon
 - 4) Module 3
 - 5) Module 4
 - 6) Module 5

* Programme : au sens d'un ensemble de traitement sur ordinateur. Le lecteur intéressé trouvera tous détails dans « Recherche de règles d'ordonnement par traitement sur ordinateur », thèse de Spécialités, Faculté des Sciences de Nancy, janvier 1971, Proth.

Une telle maquette permet d'abord à l'utilisateur de se rendre compte des conséquences de certaines décisions (introduites au fur à mesure du déroulement du programme), c'est donc un outil qui permet d'expérimenter sur ordinateur des règles et méthodes de gestions.

Dans un deuxième temps, une fois les décisions fixées, la maquette permet de prévoir le déroulement des opérations et d'assurer les opérations administratives nécessaires à leur bon déroulement telles que les transferts d'informations aux intéressés ou la gestion des stocks d'approvisionnement en pièces.

Rappelons enfin qu'une telle maquette peut être utilisée sur les méthodes classiques de simulation pour introduire l'aléatoire.

IV. TESTS DE REGLES DE DECISION

Au lieu d'introduire « manuellement » les décisions prises, on peut s'intéresser en vue d'une éventuelle automatisation de la gestion à la recherche de règles définissant toutes les décisions.

Une seconde partie de ce travail a donc été d'utiliser la maquette précédente pour tester des règles de gestion, soit très élémentaires (premier type), soit des règles plus élaborées, suggérées par l'expérience et assouplies par la présence de paramètres dont on se propose d'étudier l'influence.

Citons deux règles élémentaires qui ont été testées : priorité accordée aux commandes les plus urgentes et priorité accordée aux commandes les plus importantes en poids.

Par ailleurs, voici un exemple de règle à paramètres que nous avons testée et qui s'est révélée être d'une grande souplesse. Les niveaux de priorité accordés aux commandes étaient obtenus par le calcul de l'expression.

$$D_{P_s, O_k} = l(qC_{P_s}^{t_i} + (1 - q)K_{O_k}^{t_i})$$

où $C_{P_s}^{t_i}$ mesure l'urgence de la commande P_s à l'instant t_i ,
et $K_{O_k}^{t_i}$ mesure la charge de l'opération O_k à l'instant t_i .

En faisant varier le paramètre q il était possible de modifier l'importance relative de l'urgence de la commande considérée et de la charge de l'opération concernée. D'autre part, une augmentation de l permettrait de donner priorité aux commandes en cours sur celles dont le lancement n'avait pas encore été décidé.

Les résultats obtenus à l'aide des règles précédentes nous ont permis de suivre l'évolution au cours du temps de certaines fonctions importantes et nous présentons quelques uns des résultats obtenus.

1) Temps moyens d'immobilisation des machines

Nous avons appelé « Temps moyen d'immobilisation des machines » le rapport :

$$\frac{\text{Temps total de l'immobilisation}}{\text{Période recouverte par la simulation.}}$$

La courbe en pointillés nous donne l'évolution de ce temps dans le cas où la priorité est donnée aux commandes enregistrées en premier. La courbe en trait plein nous permet de suivre l'évolution de ce temps moyen lorsque la priorité est donnée aux commandes les plus importantes.

Des graphes analogues ont été tracés dans le cas de la règle à paramètres, et pour différentes valeurs de ces paramètres.

Nous en présentons quelques-uns ci-dessous.

Le problème du temps d'immobilisation était important dans le problème envisagé : pour passer d'une opération à une autre, les machines demandaient un temps d'immobilisation non négligeable dont il fallait tenir compte.

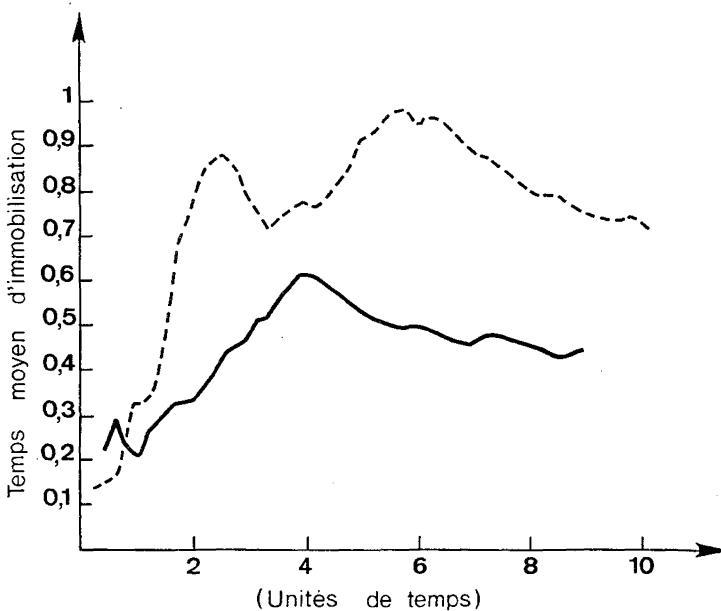


Figure 1

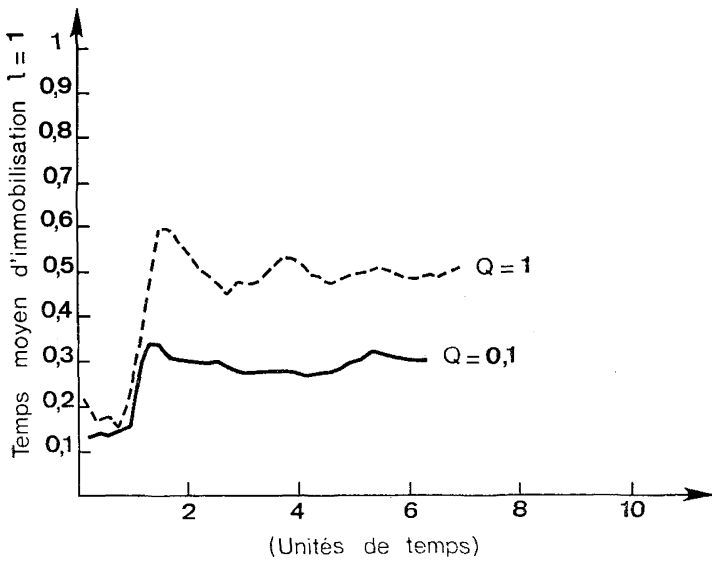


Figure 2

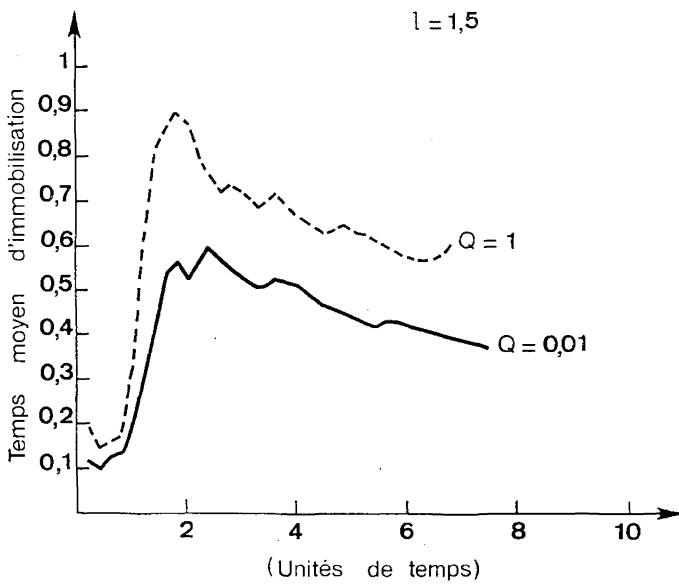


Figure 3

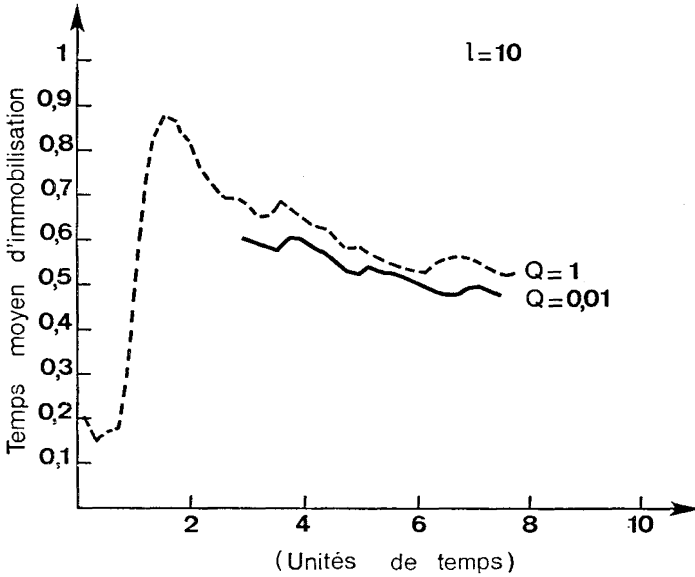


Figure 4

2) Poids des en-cours

L'évolution du poids des en-cours dans le temps a également retenu notre attention.

Nous avons donc noté ce poids tout au long du déroulement de la simulation.

Le premier graphe ci-dessous nous donne, avec les mêmes conventions de tracés que précédemment, l'évolution de cette fonction pour les deux règles non paramétriques.

Des tracés semblables ont été obtenus à l'aide de la règle paramétrique et pour différentes valeurs des paramètres. Nous en donnons quelques exemples.

Nous avons étudié de la même manière l'évolution du nombre d'ouvriers employés au cours du temps, l'écart (pondéré par le poids de la commande) entre le délai et le temps de fin de fabrication, l'évolution des stocks, etc...

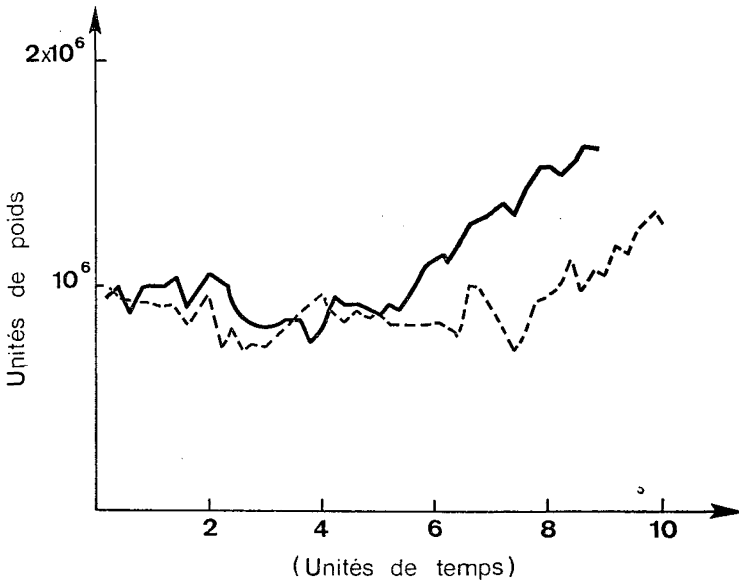


Figure 5

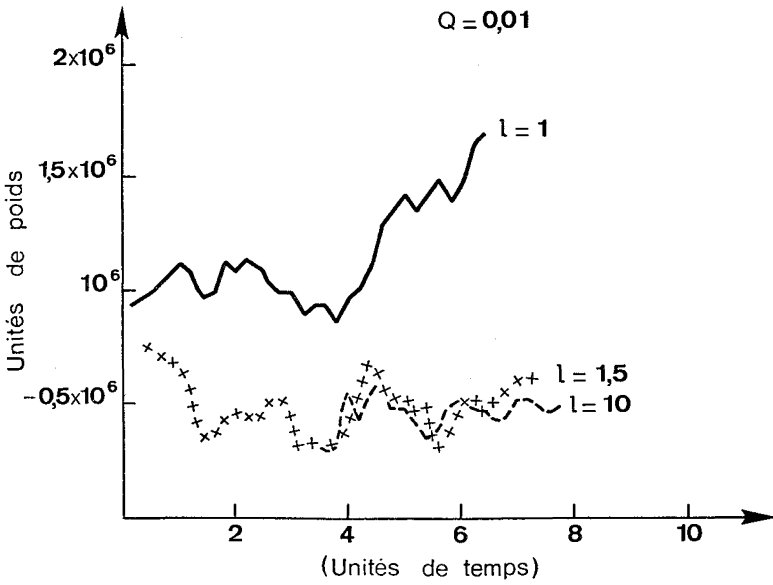


Figure 6

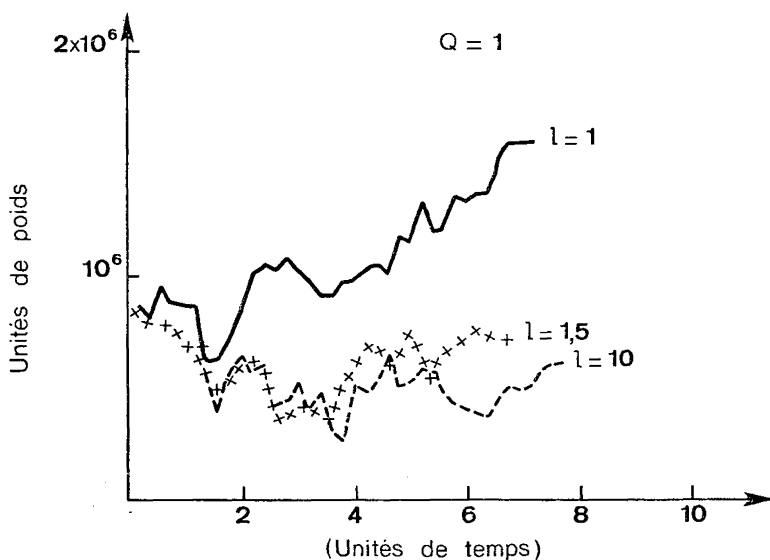


Figure 7

L'observation de ces différents graphes permet de se rendre compte de l'influence des politiques envisagées sur les différentes fonctions — donc de choisir une politique avec un risque moindre.

V. CONCLUSIONS

Nous retiendrons donc qu'une maquette permet à l'utilisateur de tester, à bas prix, certaines des décisions qu'il compte prendre. Par ailleurs nous pensons qu'il est souhaitable de développer des règles de décisions du type paramétrique qui permettent de se rendre compte de l'importance relative des différents facteurs utilisés pour les décisions. De plus, il faut se rendre compte que de tels problèmes ne conduisent pas à une « optimisation » mais à un choix raisonné de méthodes ou de règles de gestion.

Rappelons enfin que la grosse difficulté reste encore, au niveau de l'entreprise, la saisie des données : il est très difficile, bien souvent, non seulement de connaître les valeurs des paramètres, mais encore de recenser les paramètres indispensables à la construction de la maquette.

BIBLIOGRAPHIE

Des problèmes analogues ont été étudiés par

- [1] F. RUSSO, Thèse publiée au Massachusetts Institute of Technology 1965.

On aura également intérêt à consulter au sujet des langages de simulation :

- [2] R. EHRMANN, *Les langages de simulation*, Dunod, Paris, 1970.
- [3] J. ANDRÉ, *Introduction aux principaux concepts nouveaux de Simula 67*, E.S.D. control Data actes du Séminaire de Programmation de l'Université de Toulouse. Pub. de l'Univ. de Toulouse, 1971.
- [4] J. M. PROTH, *Recherche de règles d'ordonnancement par traitement sur ordinateur*, Thèse de spécialité, Institut Universitaire de Calcul Automatique. Nancy, janvier 1971.