

P. LADURE

J. LEGRAS

**Un exemple d'aide à la décision pour la recherche
de règles d'affectation et d'ordonnement
d'un système de ponts roulants**

Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle, tome 6, n° V3 (1972), p. 27-40

http://www.numdam.org/item?id=RO_1972__6_3_27_0

© AFCET, 1972, tous droits réservés.

L'accès aux archives de la revue « Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). 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/>

UN EXEMPLE D'AIDE A LA DECISION POUR LA RECHERCHE DE REGLES D'AFFECTATION ET D'ORDONNANCEMENT D'UN SYSTEME DE PONTS ROULANTS

par P. LADURE ⁽¹⁾ et J. LEGRAS ⁽²⁾

Résumé. — *Le but de ce travail était double :*

1) *Construire une maquette de simulation reflétant aussi fidèlement que possible un système de ponts roulants.*

2) *Étudier, à l'aide de l'ordinateur, une organisation de l'activité de ce système en respectant des règles de sécurité. Cette organisation comprend une phase d'affectation et une phase d'ordonnancement. Nous insistons, ici, sur la phase d'ordonnancement. A partir d'un certain nombre de règles de traitement formalisées et introduites dans des programmes nous générons d'une part un ordonnancement et d'autre part des documents permettant d'améliorer la solution fournie par un « dialogue » entre l'utilisateur et l'ordinateur.*

Nous nous proposons d'étudier la gestion d'un ensemble de ponts roulants qui effectuent des tâches de manutention dans une halle. Le travail réalisé par l'ensemble des ponts est un travail limité à trois types d'opérations élémentaires distinctes :

- 1) Levage d'une charge précédé ou non par un accrochage de cette charge.
- 2) Transfert de cette charge d'un point à un autre.
- 3) Descente d'une charge suivie ou non du décrochage de cette charge.

Ce système est formé de quatre ponts roulants utilisant le même chemin de roulement et ne pouvant donc pas se croiser. Ce chemin de roulement couvre toute la longueur de la halle de sorte que chaque pont peut accéder à tout point de la halle : deux ponts peuvent avoir besoin de se trouver au même endroit, au même moment. La proximité de deux ponts est une source d'incidents. Citons, en outre, quelques caractéristiques du système de production qui sont importantes pour la suite.

(1) Informaticien au Laboratoire de Psychologie du travail de l'EPHE.
(2) Professeur à l'Université de Nancy I.

Ainsi, lorsqu'un pont a entrepris la réalisation d'une tâche, il ne peut l'interrompre pour en faire une autre : il doit l'achever. Les zones de travail de la majorité des tâches du processus de production sont assez regroupées, mais certaines tâches nécessitent un déplacement assez long du pont chargé de leur réalisation. Ces déplacements, en plus du fait qu'ils peuvent être sources d'incidents, interdisent tout travail par un autre pont aux endroits situés sur le parcours. Le déclenchement de ces tâches doit donc être fixé de façon à gêner le moins possible la réalisation des autres tâches « compactes ». L'activité réalisée par les ponts dans cette unité de production était formée de 62 tâches décomposées en plus de 310 composantes.

Dans une première phase, nous avons élaboré une maquette ⁽¹⁾ de simulation qui reflète aussi fidèlement que possible l'activité du système-ponts. Pour présenter ces résultats, nous utilisons une représentation graphique sur laquelle nous plaçons en abscisse la longueur de la halle dans laquelle circulent les ponts et en ordonnée le temps. Nous avons connaissance, ainsi, à intervalle de temps régulier, de l'« état » des ponts.

— Cet état ε_{ti} , à l'instant t_i , peut être caractérisé par le n -uplet suivant. A chaque pont j ($j = 1, \dots, 4$), nous associons :

- 1) Sa position P_j dans l'aciérie.
- 2) Son activité A_j fonction elle-même de
 - 2.1) la tâche ta_j que le pont réalise
 - 2.2) l'état e_j du pont.

— Nous obtenons la formule suivante

$$\varepsilon_{ti} = \varepsilon_{ti}(P_1, A_1(ta_1, e_1), P_2, A_2(ta_2, e_2), \dots, P_4, A_4(ta_4, e_4)).$$

— Pour définir la position P_j du pont, l'abscisse d'un point dans la halle est une variable que l'on discrétisera pour des raisons de commodité. Prenant comme unité la largeur d'un pont appelée « Case », nous repèrerons la position d'un pont par le numéro de la case au-dessus de laquelle il se trouve. Pour définir l'activité A_j , nous avons décomposé le processus en tâches et les tâches en composantes. Les états possibles d'un pont sont au nombre de trois :

- libre et inactif quand le pont, au repos, n'effectue pas de tâche,
- coupé et actif quand le pont réalise effectivement une tâche,
- occupé et inactif quand le pont, occupé à la réalisation d'une tâche, est obligé d'interrompre *momentanément* sa tâche (par exemple, pour laisser un autre pont effectuer sa propre tâche).

La représentation graphique de l'état du système-pont consiste à positionner sur la ligne, correspondante à l'instant t_i , les n -uplets représentatifs des rensei-

(1) Nous appelons « maquette » le programme qui permet la modélisation de l'activité du système-pont sans passer par la modélisation mathématique.

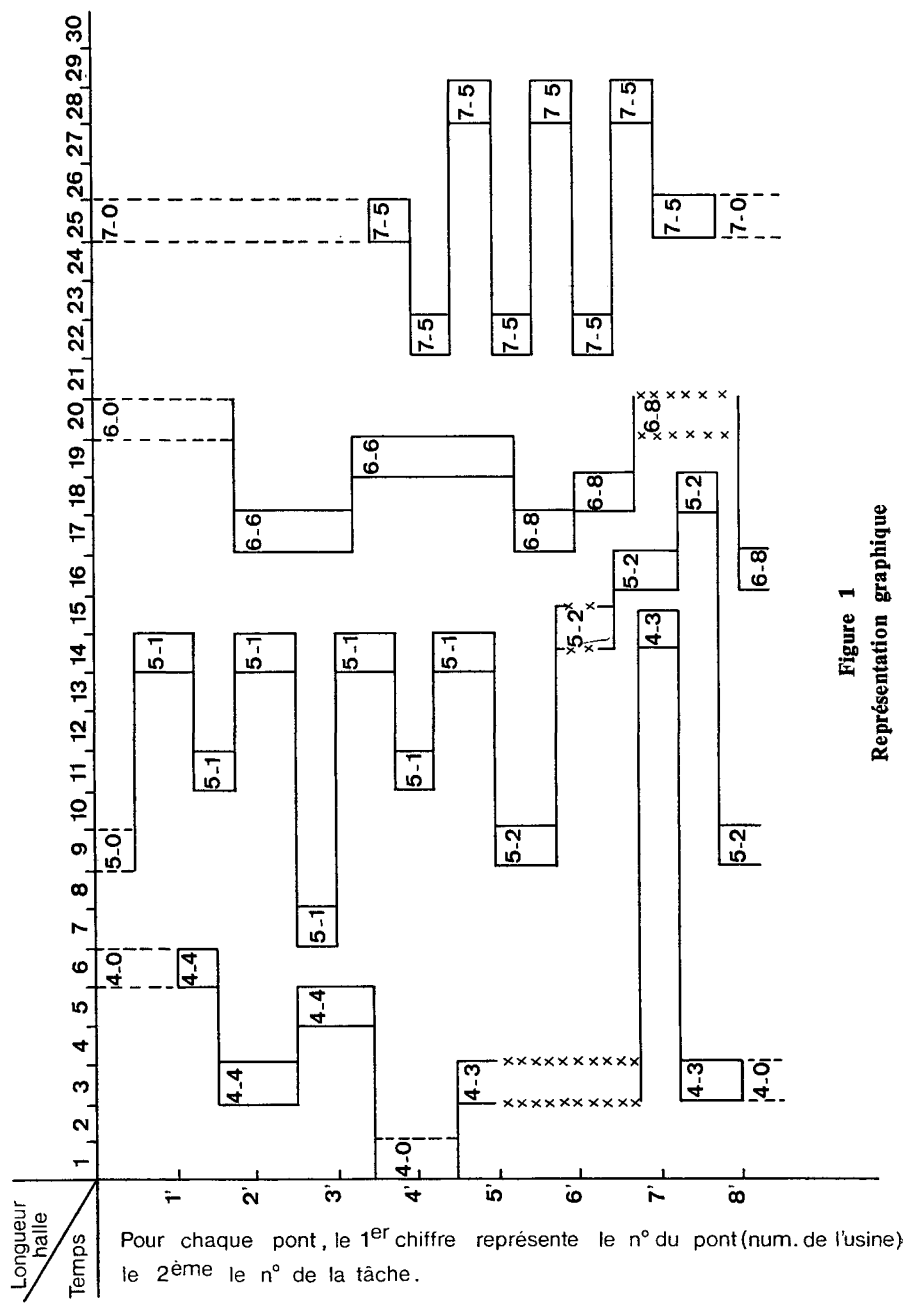


Figure 1
Représentation graphique

nements attachés à chaque pont. Cette représentation graphique faite manuellement demande un temps d'établissement très long et une première phase de notre travail consiste à automatiser cette représentation. Nous avons, pour cela, rédigé un programme en FORTRAN IV 10070 qui fournit la reproduction graphique de tout processus observé. Cette représentation indique, outre le processus proprement dit, les déplacements à vide des ponts et leurs attentes, alors qu'ils sont chargés. Il est tout-à-fait exact que si le déplacement des ponts prenait un temps notable d'exécution, il faudrait le représenter graphiquement par un trait oblique. Or, ce n'est pas le cas dans notre exemple. Par souci d'exactitude, nous avons cependant, lors des observations, intégré ces temps négligeables dans la durée des composantes. « Le temps de déplacement des ponts est négligeable par rapport aux durées des opérations : un déplacement est donc matérialisé par une ligne horizontale dans la figure 1 ».

Nous donnons, ci-dessus, un exemple de représentation graphique (fig. 1). Cette représentation permet à l'utilisateur de détecter des zones dangereuses : zones de rapprochement des ponts, zones dans lesquelles un pont se déplace rapidement, balayant un large secteur de l'aciérie, etc...

Moyennant l'introduction de certaines règles, nous pouvons demander à la maquette de simuler, tester la validité et au besoin « aménager » tout processus dont on lui donne les éléments. Dans le cas où le processus simulé est effectivement réalisable, la maquette en engendre la représentation graphique. Dans le cas contraire, la maquette signale le conflit existant entre les ponts (par exemple, deux ponts doivent travailler au même endroit, simultanément). Elle en fournit une solution (par exemple, en donnant la priorité à un pont et en déplaçant l'autre). On obtient les « corrections » nécessaires par une analyse (du conflit) à l'aide de trois critères (priorité des ponts, durée prévisible du conflit, position géographique des ponts), suivie de modifications obtenues à partir de trois règles (déplacer un pont, déplacer un pont et le faire attendre, retarder le déclenchement d'une tâche).

Pour organiser le plan de travail des ponts, il faudra déterminer pour chaque tâche :

- 1° L'instant de déclenchement de cette tâche (problème d'ordonnancement).
- 2° Le pont qui doit effectuer cette tâche (problème d'affectation).

Cette organisation peut être faite par des moyens élémentaires, mais si on recherche une bonne organisation assurant une sécurité satisfaisante et une gestion optimisée des ponts, le travail de préparation est considérable. Nous nous sommes proposés d'alléger ce travail en ayant recours à l'ordinateur.

Nous avons, tout d'abord, mis au point une méthode d'affectation dont l'objectif était double.

1. Améliorer le niveau de sécurité en évitant l'emploi simultané de deux ponts trop voisins.
2. Répartir équitablement le travail entre les quatre ponts.

Nous avons limité notre étude au cas simplifié où il n'y a aucune contrainte d'ordre entre les tâches et où toute tâche est réalisable par tout pont. La méthode consiste, à partir de l'ensemble E des tâches à effectuer, à diviser la halle en autant de zones que de ponts de telle sorte que les charges de travail des ponts soient équivalentes — chacune de ces zones devient le secteur préférentiel du pont correspondant. Nous n'insistons pas sur cette méthode et examinerons plus en détail la méthode d'ordonnancement.

Ici, encore, nous nous proposons de conserver un niveau de sécurité satisfaisant tout en cherchant à réaliser toutes les tâches dans le minimum de temps.

La méthode, développée sous le nom de « Méthode des Contours » est basée sur la schématisation suivante :

Si une tâche nécessite un pont entre les instants t_1 et t_2 , et si les abscisses extrêmes du chemin balayé par le pont sont x_1 et x_2 , nous associons à cette tâche les rectangles ($t_1 < t < t_2$, $x_1 < x < x_2$) quelque soient les déplacements réels du pont (figures en hachuré sur fig. 2).

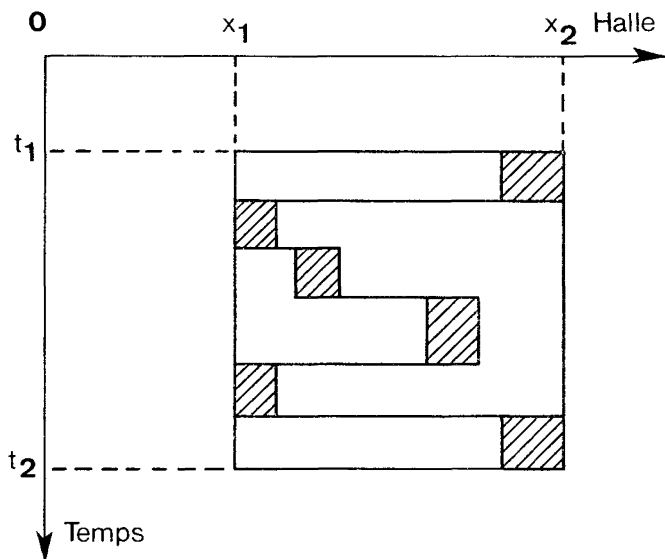


Figure 2

Une conséquence est que si 2 tâches sont représentées par des rectangles disjoints, nous sommes certains qu'il n'y aura aucune « perturbation » dans leurs réalisations, ni aucune proximité dangereuse des ponts (Règle n° 1 de sécurité).

Nous sommes donc conduits au problème suivant :

Des données sont constituées d'un certain nombre de rectangles, de dimensions connues, qui peuvent être déplacés parallèlement à t (choix de t_1 et donc

de t_2), mais uniquement sur des parallèles à $Ot(x_1$ et x_2 fixés). Il faut placer ces rectangles en respectant impérativement deux règles :

- le non recouvrement de 2 rectangles,
- le nombre de rectangles coupés par une parallèle quelconque à l'axe des x ne doit pas être supérieure à 4 (pour un système de 4 ponts).

On peut chercher à profiter de la génération automatique d'une solution de ce problème pour l'étendre; par exemple en libérant certains ponts pour un entretien ou une réparation. Mais dans le problème que nous traitons ici nous nous préoccupons seulement de réaliser l'ordonnancement dans le minimum de temps.

La méthode que nous avons développée, utilise des techniques mises au point à l'occasion de recherches expérimentales (ordonnancement manuel où chaque rectangle était matérialisé par un « carton »). Nous allons l'exposer sur un exemple et mettre en évidence, au fur et à mesure de leur emploi, les règles de traitement que nous avons adoptées. La formalisation de ces règles se trouve dans la référence III, chap. V. L'ensemble des tâches que nous avons à ordonnancer est formé de la liste de toutes les tâches. Chacune de ces tâches est munie :

- a) d'un numéro d'ordre (dans la liste),
- b) des 2 bornes extrêmes de leurs zones de travail (*i e* deux n^{os} de cases),
- c) de sa durée.

Les éléments *b* et *c* donnent, rappelons-le, pour chaque rectangle les deux dimensions du rectangle représentant la tâche.

Ensemble de Base

Pour un ensemble *E* de tâches donné, toutes les cases de la halle n'ont pas la même durée d'occupation. Pour chaque case cette durée d'occupation est égale à la somme des durées des tâches dont la zone de travail contient cette case. Nous nommerons *case de base* la case la plus occupée, qui, dans notre exemple (cf. fig. 3), est la case n^o 13. Les tâches qui « utilisent » cette case sont les tâches n^{os} 2, 5, 8, 9, 10, 12, 13, 14. Il ne peut y avoir de simultanéité dans la réalisation de deux (ou plus) de ces tâches sous peine de provoquer des chevauchements de rectangles. Le sous-ensemble de tâches, ainsi formé, sera dit *ensemble de base*. Une condition nécessaire pour que l'ensemble *E* soit réalisé dans le minimum de temps est que la case de base soit occupée sans discontinuité. L'ensemble de base est donc un *ensemble critique* pour lequel tout décalage introduit dans le déclenchement de ces tâches risque de rallonger d'autant la durée minimum du processus. La durée d'occupation de cette case de base est une limite inférieure en dessous de laquelle la durée du processus ne peut aller sans enfreindre la règle de sécurité. Nous chercherons une solution approchant d'aussi près que possible cette borne inférieure.

L'ensemble de base dans notre exemple possède 8 éléments. Le choix d'un ordonnancement parmi les $8!$ façons de placer ces rectangles de base doit tenir compte des 6 rectangles restants (n° 1, 3, 4, 6, 7, 11). Ces rectangles

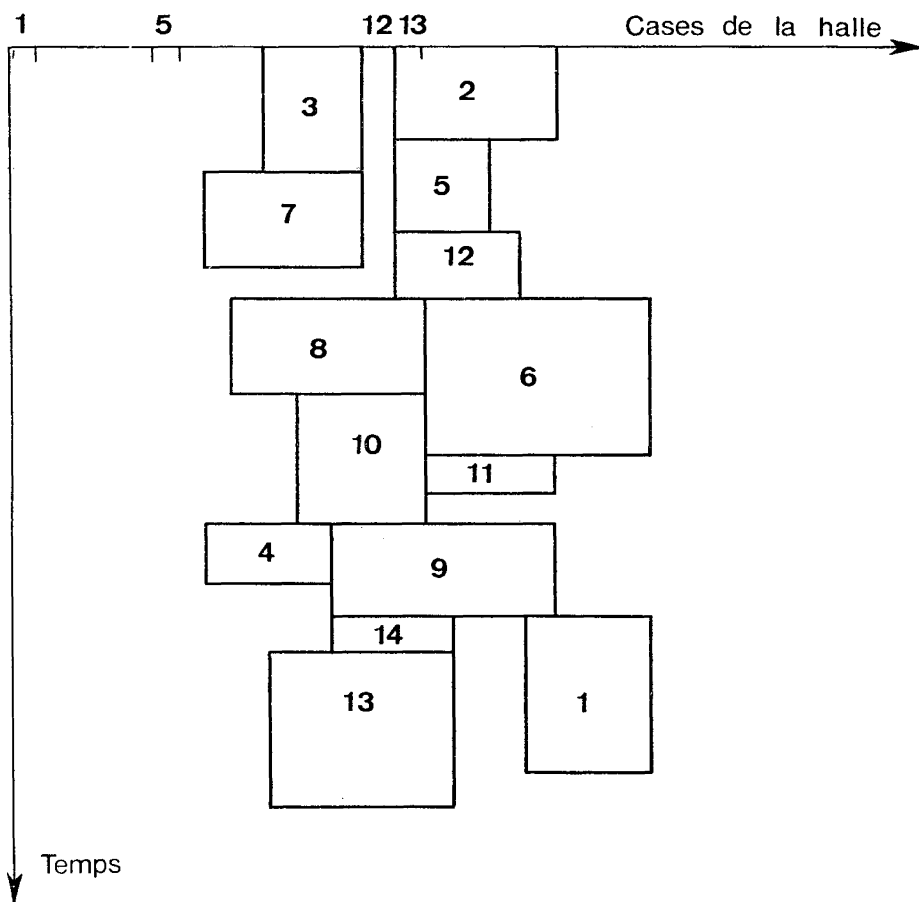


Figure 3

n'occupent pas la case de base. Situés soit à gauche, soit à droite de cette case, on les répartit ainsi en un sous-ensemble gauche (rect. n°s 3, 4, 7) et un sous-ensemble droit (n°s 1, 6, 11).

Sous-ensembles complémentaires. Relations de succession immédiate. Relation de voisinage possible

Nous cherchons à regrouper les rectangles qui peuvent être placés simultanément. Nous procédons par étapes successives. Chacune de ces étapes est

n° novembre 1972, V-3.

divisée en deux sous-étapes (Étude à droite. Étude à gauche). Le travail pour chacune de ces étapes est le suivant :

α 1. Nous cherchons parmi les rectangles restant à gauche ceux qui sont les plus proches de la case de base. Ce sont les rectangles n^{os} 3 et 7. Ils forment le sous-ensemble A . On crée entre eux une relation de succession immédiate (S.I.).

α 2. Parmi les rectangles de l'ensemble de base, nous cherchons ceux qui peuvent être placés en même temps que ceux de l'ensemble A . Ce sont les rectangles n^{os} 2, 5, 12. Ils forment l'ensemble B . On crée, également entre eux des relations S.I.

α 3. Les ensembles A et B forment un *couple de sous-ensembles complémentaires*. On crée entre leurs éléments une *relation de voisinage possible*. Tout élément de A pouvant avoir lieu en même temps que tout élément de B .

α 4. L'ordre adopté pour ranger les rectangles à l'intérieur de ces sous-ensembles A et B est l'ordre de numérotation (donc dépendant de la structure de la liste d'entrée). C'est la *règle de rangement n° 1 (R1)*.

β). Nous cherchons, alors, parmi les rectangles restant à droite ceux qui sont les plus proches de la case de base. Ce sont les rectangles 6 et 11. A nouveau création d'une relation S.I. sur cet ensemble A' ainsi formé.

— Puis recherche similaire pour l'ensemble B' — 8 et 10 sont les rectangles concernés.

— On utilise à nouveau les points α 3 et α 4 vus pour l'étude à gauche.

Règle de rangement n° 2. Rectangles communs

Il peut arriver qu'un rectangle appartienne aux ensembles B et B' lors d'une même étape. Cela signifie qu'il sert aux positionnements simultanés à gauche et à droite. Ainsi le rectangle 14 forme d'une part avec le n° 9 l'ensemble B , et d'autre part avec le rectangle n° 13 l'ensemble B' . Les ensembles A et A' sont chacun réduit à 1 élément (respectivement n° 4 et n° 1). Dans ce cas on range d'abord les rectangles appartenant à B et non à B' suivant la règle R1. Puis on range les rectangles appartenant à B et à B' suivant la règle R1 et enfin les rectangles appartenant à B' et non à B .

Création d'une Matrice de Relations inter-Tâches

Nous venons de montrer qu'il était commode d'introduire entre certaines tâches des relations indiquant s'il était possible de les placer côte à côte ou s'il était obligatoire de les placer l'une immédiatement après l'autre (respectivement voisinage possible et succession immédiate). Nous schématisons toutes ces relations dans une matrice dont l'ensemble des lignes et l'ensemble des colonnes représente chacun l'ensemble des tâches. L'élément situé à l'intersection de la $i^{\text{ème}}$ ligne et de la $j^{\text{ème}}$ colonne indique d'une part le type de la

relation existant entre la $i^{\text{ème}}$ et la $j^{\text{ème}}$ tâche et d'autre part le n° de l'étape au cours de laquelle cette relation aura été créée. Une procédure d'exploration systématique de cette matrice permet de générer un processus. Le processus ne met en œuvre qu'un certain nombre des relations figurant dans la matrice. C'est donc à partir de cette matrice que l'on obtient le graphique (fig. 3) représentant le processus. Signalons que l'on peut compléter les renseignements figurant dans cette matrice par exemple en indiquant lors du « voisinage possible » si i est à gauche de j ou inversement.

Possibilité d'intervention de l'utilisateur

La méthode proposée n'est pas une méthode d'optimisation : Dans un problème difficile, où il est certain qu'il n'y a pas unicité de la solution, où les critères d'optimisation sont eux-mêmes mal définis, nous demandons, simplement à l'ordinateur de faciliter le travail du responsable tout en laissant à ce dernier non seulement le choix de la solution mais également la possibilité d'intervenir dans le travail de l'ordinateur. Présentons deux types d'interventions possibles de l'utilisateur.

I.1. Modification de la structure de la liste des tâches

Sur l'exemple précédent nous voulons ajouter une tâche supplémentaire (n° 15) dont le placement sur la figure 4 entraîne une augmentation de la durée du processus de Δt . Cependant si on échange les n°s des tâches 3 et 7 et les n°s des tâches 8 et 10, on libère ainsi un emplacement suffisant pour placer la tâche n° 15 sans augmenter la durée du processus initial. Nous voyons ainsi l'importance de la numérotation adoptée et son incidence sur la durée du processus.

La possibilité donnée à l'utilisateur de modifier l'ordre des tâches, lui permet donc d'obtenir diverses configurations et, éventuellement d'obtenir des résultats plus intéressants que ceux obtenus dans un premier traitement. *L'utilisateur peut donc introduire dans le traitement automatisé, les modifications qui lui sont suggérées à la vue des résultats d'une étude précédente.* L'ordinateur intervient comme outil d'aide à la décision.

Indication sur le programme

Cette méthode a fait l'objet d'un programme rédigé en FORTRAN IV et mis au point sur le CII 10070 de l'Institut Universitaire de Calcul Automatique de Nancy.

La mise en œuvre de ce programme, composée de la lecture du programme (sur cartes), de la lecture des données (sur cartes également), de la compilation, de l'exécution et de la sortie des résultats sur imprimante, nécessite moins de cinq minutes. Le total des cartes lues est d'environ 1 500 cartes. L'occupation du programme complet est d'environ 40 K mots mémoire.

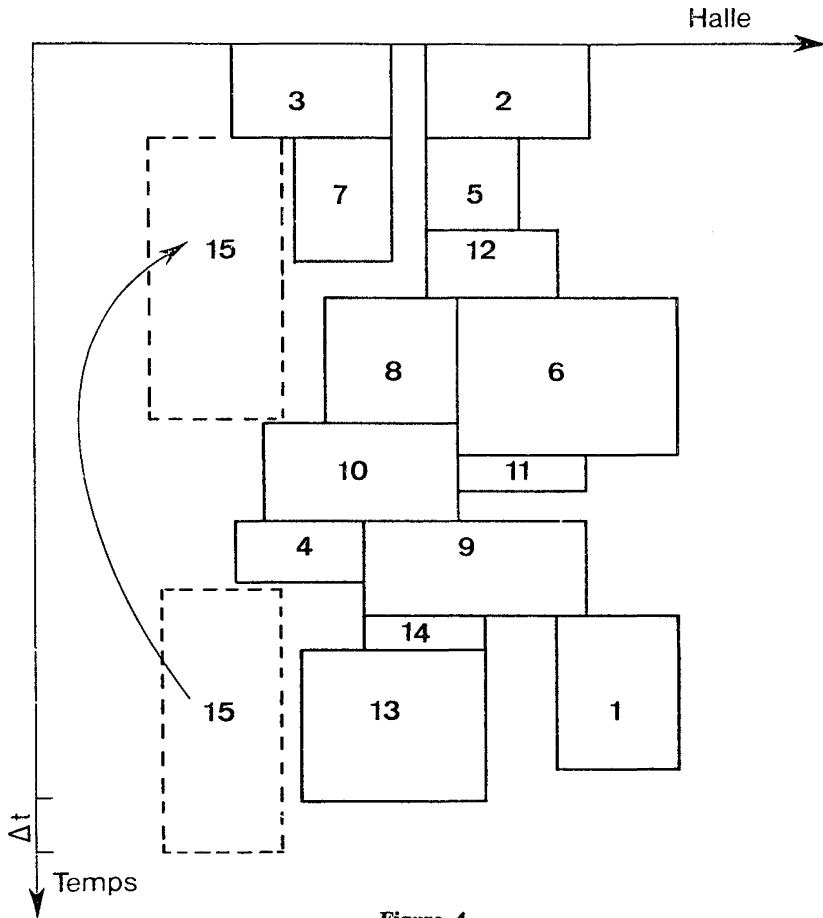


Figure 4

Nous joignons à ces quelques indications un exemple de résultat obtenu sur imprimante. Le nombre de tâches de cet exemple est de 34, (cf. pages 38 et 39, fig. 5).

I.2. Implantation de tâches prioritaires

Il a paru intéressant, dans un souci d'extension du champ d'application de la méthode, de prévoir le cas des tâches prioritaires. Par tâches prioritaires, nous entendons les tâches qui doivent avoir lieu à un moment donné, fixé à l'avance (le rectangle correspondant est bloqué dans le temps). On dispose donc de deux types de tâches :

a) les tâches libres (celles sur lesquelles porte la méthode des contours) sans aucune contrainte entre elles et pouvant avoir lieu à tout moment;

b) les tâches prioritaires dont la date d'exécution, et, bien sûr, le lieu sont fixés.

Pour réaliser un ordonnancement sur un ensemble de tâches composé de tâches libres et de tâches prioritaires, le principe adopté est le suivant :

On construit, par la méthode des contours, un processus constitué des seules tâches libres, puis on insère les tâches prioritaires à leurs places fixées par avance; ou l'endroit « prioritaire » est libre et il n'y a qu'à placer la tâche prioritaire correspondante, ou l'endroit « prioritaire » est occupé par une ou plusieurs tâches libres; on place, alors, la tâche prioritaire et on décale les tâches libres à une date ultérieure déterminée de façon à ce que la perte de temps soit la plus faible possible. Il est à noter que le décalage de ces tâches est effectué en essayant de conserver au maximum l'ordre établi (par la méthode des contours) de l'ensemble des tâches libres. Il est évident que le retard engendré par l'introduction de ces tâches prioritaires est fonction de leur endroit de déroulement (une tâche prioritaire se déroulant dans une zone peu occupée peut n'influer en rien la durée du processus) et de leur nombre (si les tâches prioritaires représentaient la grande majorité des tâches, on ne procéderait pas de cette façon).

CONCLUSION

La méthode présentée de façon très succincte est évidemment susceptible d'amélioration et d'extension (cf. référence III, chapitre V).

En résumé, nous avons voulu montrer :

1° comment certaines règles simples permettaient de fournir une solution assez satisfaisante au problème posé;

2° comment la représentation graphique des résultats obtenus sur ordinateur et les suggestions que peut apporter une telle visualisation des résultats, permettent dans un dialogue entre l'utilisateur et l'ordinateur, de modifier, éventuellement d'améliorer, les processus engendrés automatiquement.

L'ordinateur, dans cette optique, n'apparaît pas comme imposant une décision, par exemple à la suite de calcul d'optimisation, mais comme outil *d'aide à la décision* dégageant le responsable d'un travail automatisable, mais lui laissant en « dialogue », la possibilité d'invention par des modifications du traitement automatique et le choix entre diverses solutions proposées.

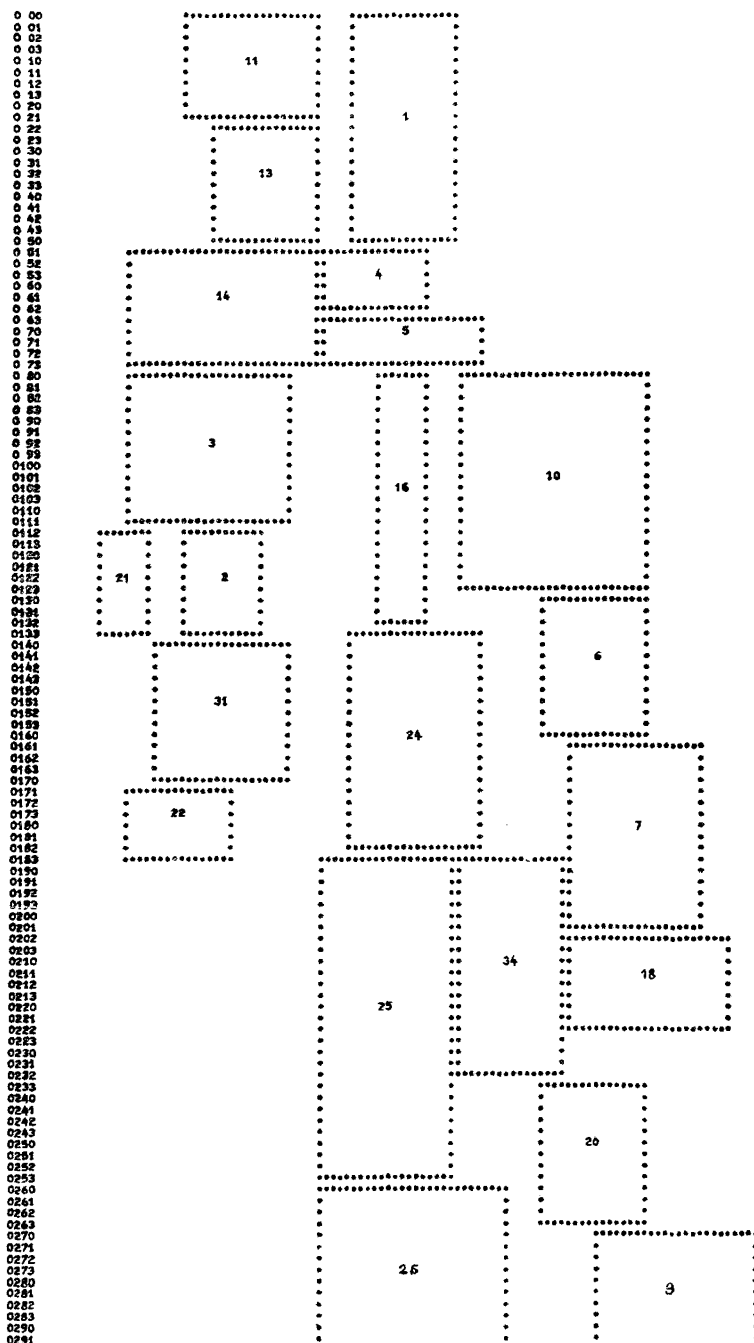


Figure 5

Un exemple de document obtenu par le programme décrit

0292
0293
0300
0301
0302
0303
0310
0311
0312
0313
0320
0321
0322
0323
0330
0331
0332
0333
0340
0341
0342
0343
0350
0351
0352
0353
0360
0361
0362
0363
0370
0371
0372
0373
0380
0381
0382
0383
0390
0391
0392
0393
0400
0401
0402
0403
0410
0411
0412
0413
0420
0421
0422
0423
0430
0431
0432
0433
0440
0441
0442
0443
0450
0451
0452
0453
0460
0461
0462
0463
0470
0471
0472
0473
0480
0481
0482
0483
0490
0491
0492
0493
0500
0501
0502
0503
0510
0511
0512
0513
0520
0521
0522
0523
0530
0531
0532
0533
0540
0541
0542
0543
0550
0551
0552
0553
0560
0561
0562
0563
0570
0571
0572
0573
0580
0581
0582
0583
0590
0591
0592

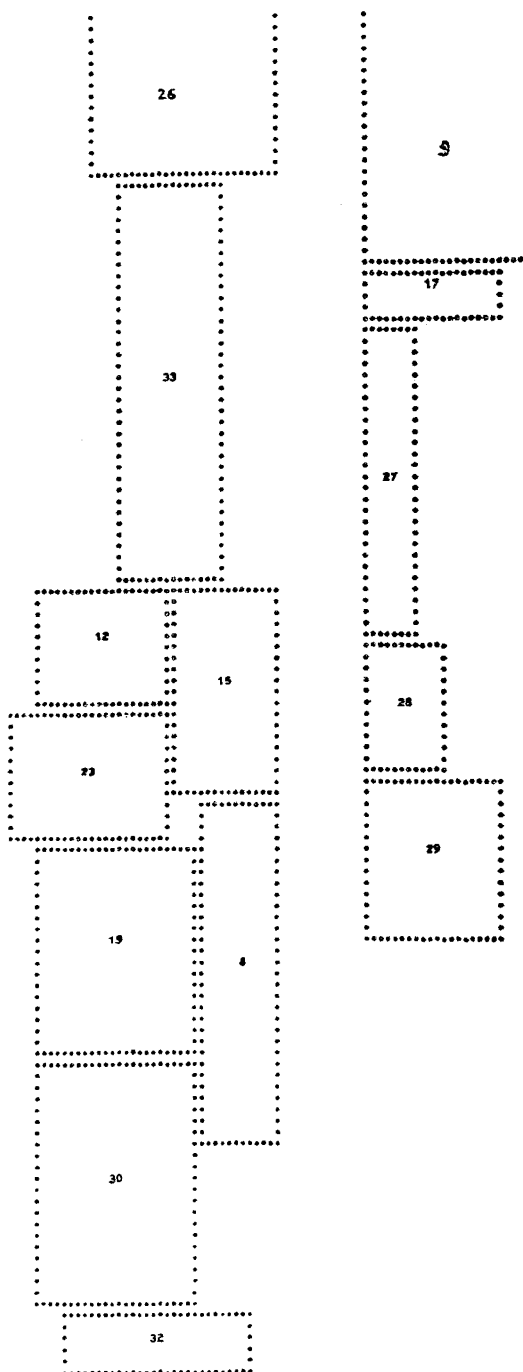


Figure 5

BIBLIOGRAPHIE

Restreinte car portant sur un sujet où aucune étude, à notre connaissance, n'a été effectuée dans cette optique et où l'aspect aide à la décision « dialogue » ordinateur-utilisateur est assez neuf.

- [I] ARMOUR et BUFFA, *A Heuristic Algorithm and Simulation approach to the Relative Location of Facilities*. Management Science, janvier 1963, p. 294-309.
- [II] BERGE, *Graphes et Hypergraphes*, Dunod, 1970.
- [III] LADURE, *Recherche de règles d'Affectation et d'ordonnancement pour un système de Ponts Roulants*. Thèse de spécialité, U.E.R Sciences Mathématiques, Nancy, 1972.
- [IV] PROTH, *Recherche de règles d'ordonnancement par traitement sur ordinateur*. Thèse de spécialité, U.E.R Sciences Mathématiques, Nancy, 1971.