

## TENDANCES ACTUELLES EN PROGRAMMATION AUTOMATIQUE

par G. TIBAUX,  
*Université de Liège.*

---

Le nombre et la qualité des services que peut rendre une machine à traiter l'information dépendent, pour une large mesure, de sa commodité d'emploi. C'est la raison majeure du développement considérable pris, en l'espace de dix ans, par la programmation automatique. L'ampleur du sujet ne nous permettrait guère d'en faire ici une étude exhaustive; nous nous bornerons donc à dégager quelques aspects caractéristiques illustrant l'évolution actuelle de ces procédés.

Nous en rappellerons brièvement les buts et les méthodes principales.

Le plus souvent, les langages-machine sont très analytiques, conventionnels, asservis à la structure technologique, et requièrent un gros travail de codification. L'objectif premier de la programmation automatique est de permettre la formulation des problèmes dans un langage plus synthétique, plus naturel, plus commode. Un programme approprié rendra la machine capable d'utiliser ce nouveau langage, soit directement, soit en transformant le programme d'origine en un programme machine, prêt pour l'exécution.

Deux procédés fondamentaux sont donc applicables, conjointement ou indépendamment, au traitement du langage d'origine. D'une part, le programme interprétatif, à chaque étape de la résolution du problème, tire directement du programme original les informations nécessaires à la poursuite du travail; il y a donc dans ce cas simultanéité entre la décodification du langage et l'exécution proprement dite du programme. D'autre part, le programme de traduction, préalablement à tout calcul, transforme le programme rédigé en langage source en un programme équivalent exprimé dans un autre langage: le langage objet. Ce dernier peut être, soit le langage machine, soit un langage intermédiaire destiné à une traduction ou une interprétation ultérieures.

L'exécution peut parfois suivre directement la traduction, mais il s'agit, de toute manière, de deux processus indépendants.

Les programmes de traduction peuvent se présenter sous diverses formes. Le programme de conversion réalise un simple changement de code, une modification de représentation sans altérer la structure du langage; par exemple, le passage du décimal au binaire, de la forme symbolique à l'écriture numérique des ordres. L'assembleur est capable d'incorporer au programme objet des sous-programmes ou des macro-instructions extraits de bibliothèque, et d'établir entre les diverses parties les liens nécessités par l'organisation logique du problème. Le compilateur effectue, outre les assemblages et transcodages, le passage d'un langage source à un langage objet ayant une structure logique profondément différente. De plus, du fait que l'utilisateur d'un compilateur peut presque tout ignorer de la machine objet, un tel programme est souvent intégré dans un système de programmation qui non seulement permet la préparation automatique des programmes, mais facilite encore leur mise au point, leur correction, leur enchaînement.

Le développement de ces diverses méthodes de traitement a conduit à l'élaboration de nombreux langages-sources. Au stade le plus élémentaire, les langages destinés aux seuls programmes de conversion ne sont souvent que des écritures plus plaisantes, plus mnémotechniques du langage machine : ils sont intimement liés à la structure du matériel. Au contraire, la souplesse du compilateur et du programme interprétatif permettent de l'affranchir des contingences technologiques; la commodité d'expression du processus à automatiser devient le but primordial. On passe ainsi des « computer oriented languages » (COL) aux « problem oriented languages » (POL), dans lesquels la variété des travaux effectués par les calculatrices a apporté de multiples spécialisations. Ainsi, pour les calculs numériques, on trouve des langages algébriques, tels Fortran ou Math-matic, pour les applications administratives et comptables, des systèmes comme Flow-matic, Comtran, Fact. La variété des structures de machine, les divers niveaux possibles entre le langage machine et l'expression la plus synthétique d'un problème, les particularités des divers problèmes, ont fait qu'actuellement plusieurs centaines de langages différents sont utilisés pour la programmation.

A partir d'un certain degré de perfectionnement, les langages sont indépendants des machines; ils peuvent donc jouer le rôle de moyens de communication. Il suffit en effet de disposer des programmes de traitement appropriés pour que, sans aucune modification, un programme écrit dans un tel langage puisse être directement utilisé sur plusieurs matériels différents. Les perspectives d'échanges ainsi offertes aux programmeurs étaient certes

des plus alléchantes; cependant, pour assumer cette importante fonction avec toute l'efficacité souhaitable, il fallait autre chose qu'une Tour de Babel où presque chaque utilisateur avait un langage, un système particulier de programmation. De cette nécessité est né l'Algol.

Le besoin d'un langage universel commun fut ressenti simultanément, en 1957, par l'ACM (Association for Computing Machinery) (U.S.A.) et la GAMM (Gesellschaft für angewandte Mathematik und Mechanik) (Allemagne). Un groupe de travail commun à ces deux associations publia, début 1959, un rapport préliminaire. L'intérêt suscité par ce document conduisit à un élargissement du groupe initial, et c'est une commission internationale qui élaborera la première forme définitive de ce langage algorithmique commun. En mai 1960, l'Algol 60 fut livré aux méditations des programmeurs.

Moins d'un an après, divers groupes, notamment des universités, avaient produit les premiers compilateurs destinés à ce langage. Les résultats d'exploitation encouragèrent les chercheurs à se pencher sur le problème et, à l'heure actuelle, de nombreux centres de calcul et la majorité des constructeurs disposent ou disposeront très prochainement de compilateurs Algol pour leur matériel. Simultanément, les échanges de programmes ont démarré, entre autres par la publication des algorithmes dans les Communications de l'ACM. L'usage de l'Algol se généralise, tandis que les anciens langages algébriques sont plutôt en régression.

Bien sûr, il ne suffit pas d'être le produit d'une commission internationale pour remporter un tel succès. Avant d'examiner sommairement les principales qualités qui ont motivé cette adoption universelle, rappelons que l'Algol 60 est avant tout conçu pour l'expression des algorithmes, des procédés de calcul. Il peut évidemment servir à décrire d'autres types de processus, mais ne constitue pas une panacée. Il n'est pas sans défauts, mais présente, sur les autres systèmes du même genre, des avantages certains.

Avant d'être une méthode de programmation sur machine, l'Algol est un moyen de communication, de publication et d'échange de programmes. Ceci exige que la syntaxe et la sémantique du langage soient parfaitement définies, indépendamment de toute compilation particulière. Ce but a été atteint, malgré quelques imperfections, et l'on peut dire que l'Algol met fin à l'époque où la seule définition complète d'un langage était le programmeur compilateur. Ce langage est d'autre part beaucoup plus indépendant de toute structure de calculatrice que la plupart de ses semblables; l'idée même de machine en est absente, au moins jusqu'à un certain point.

Les conventions d'écriture de l'Algol sont parmi les moins restrictives et les plus naturelles qui soient. On en a notamment éliminé toute limitation, que ce soit dans le symbolisme des variables, l'échafaudage des formules arithmétiques, les formes indicielles, les expressions d'organisation logique. Une telle généralité a été rendue possible, sans rien perdre de la rigueur des définitions, par l'emploi d'un formalisme récursif. En effet, non seulement l'Algol peut servir à exprimer un programme contenant des processus récursifs, mais en outre la définition même des éléments du langage est donnée de manière récursive.

On saisira plus facilement sur un exemple la nature et la portée de cette méthode. En Algol, les règles de formation des noms symboliques destinés à représenter les variables sont énoncées de la manière suivante :

$$\begin{aligned} \langle \text{identificateur} \rangle ::= & \langle \text{lettre} \rangle \mid \langle \text{identificateur} \rangle \langle \text{lettre} \rangle \\ & \mid \langle \text{identificateur} \rangle \langle \text{chiffre} \rangle . \end{aligned}$$

Le signe  $::=$  marque l'équivalence, et le signe  $\mid$  remplace la conjonction ou. Cette règle signifie qu'un identificateur peut être formé soit par une seule lettre, soit par un autre identificateur suivi d'une lettre ou d'un chiffre. Un identificateur formé de  $N$  symboles est donc obtenu à partir d'un autre identificateur comprenant  $N - 1$  symboles, sauf pour  $N = 1$ . Dès lors, il est possible d'utiliser, pour la définition d'un terme, ce terme lui-même. L'exemple choisi est bien sûr élémentaire, et l'on trouve, dans le rapport Algol, des échafaudages de définitions bien plus complexes, mais analogues, notamment pour les expressions arithmétiques dont certains éléments peuvent être d'autres expressions arithmétiques.

Le procédé a une puissance comparable à celle de la méthode de démonstration par récurrence en mathématiques théoriques. Il n'est d'ailleurs pas limité à la seule description formelle d'un langage, mais se transpose directement à la programmation. De même qu'une définition récursive se reporte à elle-même, un sous-programme sera appelé récursif s'il est capable de s'appeler lui-même sans aucune perte d'information. Il y a là un principe qui transcende le simple bouclage, et ce n'est certes pas le moindre mérite de l'Algol que d'avoir largement contribué à répandre ces notions.

Et cependant, la naissance de l'Algol 60 n'a pas mis un terme à l'élaboration et au perfectionnement des langages. Au contraire, les principes et les méthodes mis en lumière par le rapport Algol ont fréquemment servi de bases et d'outils aux créateurs de nouveaux systèmes. C'est qu'en effet, quoi qu'en disent ses adeptes les plus fervents, l'Algol 60 n'est pas capable de

tout exprimer. Et même, dans les cas où un processus est programmable dans ce langage, il est parfois souhaitable de disposer d'un outil spécialisé, mieux adapté, plus commode. Ainsi, alors que nombre de problèmes scientifiques se trouvent à l'aise dans une méthode qui fait fi de toute contingence technologique, les problèmes administratifs ou comptables, par leurs liaisons plus étroites avec le matériel, s'en accommodent plus difficilement. Les spécialistes savent fort bien que le mode de traitement d'un fichier peut être très différent suivant que l'on dispose de bandes magnétiques ou de la seule carte perforée.

Un groupe composé principalement de représentants des constructeurs américains de calculatrices, sous la direction du Département de la Défense de ce pays, a entrepris, depuis 1959, l'élaboration d'un langage destiné à jouer, pour ce genre de problèmes, un rôle identique à celui de l'Algol en calcul scientifique : c'est le COBOL (Common Business Oriented Language). Dans ce système, un programme destiné à traiter l'information est composé de trois parties. La division « procédure » décrit la succession des opérations traitement à effectuer. La division « données » expose la structure des divers groupes d'informations mis en jeu, leur répartition en fichiers, libellés, etc. La division « environnement » détaille la manière dont sont matérialisées, dans l'équipement utilisé, les différentes unités logiques impliquées dans le processus.

Si l'on désire faire passer un tel programme, écrit pour une machine déterminée, sur un autre matériel, il devrait suffire, en principe, de modifier la division « environnement » du programme de manière à l'adapter aux éléments physiques de la nouvelle machine. C'est malheureusement loin d'être toujours le cas ; souvent, il faut réviser la description des données, et parfois même apporter des changements dans la division « procédure ». Le langage lui-même comporte d'autre part des inconvénients d'écriture qui en alourdissent le maniement. Aussi Cobol est-il loin d'avoir suscité l'enthousiasme des utilisateurs de calculatrices. Après avoir été accueilli très fraîchement, il a néanmoins remporté certains succès ; mais d'autres langages adaptés aux mêmes types de problèmes gardent la faveur d'un nombre important de programmeurs. Malgré le travail constant des auteurs du Cobol pour perfectionner leur œuvre, il semble bien qu'un langage satisfaisant à toutes les exigences des travaux administratifs et comptables ne sera pas disponible avant un certain temps.

Autres applications, autres langages. Les nouveaux domaines où s'implante le traitement automatique des informations ont conduit à l'écriture de programmes décrivant des processus totalement différents du calcul scien-

tifique ou des problèmes de gestion. On a donc entrepris de développer des langages, très éloignés de ceux que nous avons cités plus haut, dont les opérations fondamentales et la structure logique sont spécialement adaptés à ces nouveaux problèmes. De nombreux travaux ont porté, notamment, sur des langages destinés à la documentation automatique, à la traduction automatique des langues — nous y trouvons entre autres le COMIT mis au point par l'équipe du MIT — ou aux techniques de simulation digitale, qu'elles soient appliquées aux processus industriels ou à la perception et l'intelligence artificielles.

Il faut encore mentionner ici des études plus théoriques, abordant par des méthodes proches de la logique formelle la synthèse de nouveaux langages ou l'analyse des langages actuels et de leurs propriétés. Les praticiens de la programmation ont parfois tendance à hausser les épaules devant des travaux de ce genre. Certes, on y rencontre bon nombre de voies sans issue, de publications sans lendemain, de spéculations sans utilité pratique. Toutefois, certaines de ces recherches ont été à l'origine de progrès importants tant dans la structure des langages que dans leur mise en œuvre sur machine. Il ne nous est pas possible, dans cet exposé, d'examiner ces questions de façon plus détaillée, mais nous croyons devoir attirer sur ce point l'attention des programmeurs. Si ces travaux doivent être abordés avec une certaine prudence, il serait dommage de vouloir les ignorer.

La multiplicité des langages et leur évolution rapide posent d'une façon très pressante le problème de la construction des programmes de traitement. En effet, si la programmation automatique avait permis de réduire considérablement la main-d'œuvre nécessaire pour préparer la résolution d'un problème sur machine, l'expansion même du procédé risquait de provoquer une nouvelle pénurie de spécialistes. De nombreux chercheurs se sont donc attachés à simplifier et à perfectionner les méthodes de rédaction des programmes interprétatifs et des compilateurs. Cette étude n'a pas été vaine. En effet, les premiers compilateurs destinés à des langages assez synthétiques sont apparus comme de véritables monstres, ayant requis un effort énorme : 30 années-homme, ou même plus (si toutefois ces chiffres n'ont pas été exagérés dans un but publicitaire). Mais, à l'heure actuelle, de nombreux programmes de traitement sont l'œuvre d'un seul homme et représentent moins d'un an de travail.

Parmi les procédés qui ont permis de réduire le volume des compilateurs et d'en accroître la souplesse, il faut mentionner en premier lieu l'emploi généralisé du stack, ou mémoire intermédiaire ayant une structure de pile, en sorte que le premier symbole que l'on retire de cette mémoire est

toujours le dernier que l'on y a placé. Ceci facilite entre autres le réordonnement des expressions comprenant plusieurs niveaux opératoires. Un recours intensif à la consultation de tables permet de rendre la structure logique de la traduction indépendante du langage objet. Les programmes de traitement de listes, non seulement évitent tout gaspillage de mémoire pendant la compilation, mais simplifient également les manipulations de tables. Enfin, surtout pour l'Algol, l'utilisation de sous-programmes fermés indépendants et récursifs, dont chacun peut appeler n'importe quel autre et en particulier lui-même, résout de façon élégante les problèmes posés par l'emboîtement des ordres.

Pour les petites machines qui ne disposent que d'une mémoire de l'ordre de 8.000 à 16.000 mots, sans mémoires auxiliaires à grande capacité, et ont le plus souvent un langage très analytique, les programmes interprétatifs restent très en vogue et sont fréquemment préférés aux compilateurs. En effet, l'insuffisance de la mémoire conduit non seulement à accroître le temps de compilation, mais aussi à demander, en cours de traduction, des manipulations qui deviennent vite sources de fautes. Au contraire, le caractère analytique du langage objet rend le temps de décodification par programme interprétatif très petit vis-à-vis du temps d'exécution des sous-programmes opératoires. Si le traitement d'un langage machine simplifié reste le domaine d'élection du système interprétatif, de telles méthodes ont également été appliquées, avec un rendement très honorable, à des langages synthétiques indépendants de toute structure de machine, notamment à l'Algol par Van der Poel.

Il convient encore de noter, parmi les études les plus récentes, une tendance qui vise à gagner un nouveau degré dans l'automatisation de la programmation par la mise au point de programmes générateurs de compilateurs; citons notamment, dans les travaux présentés au congrès IFIP 62, l'étude théorique de Paul et le système Tool d'Opler. Il s'agit donc de pouvoir produire, sans intervention humaine, un programme destiné à une machine déterminée et rendant cette machine capable de traduire un langage source donné en un langage objet également fixé. Le générateur idéal devrait être capable d'effectuer ce travail pour une machine et des langages choisis de façon quelconque et après avoir simplement reçu, sous une forme appropriée, une description de ces divers éléments. Si une telle perfection n'est pas encore atteinte, des résultats pratiques très encourageants ont déjà été obtenus.

En même temps que les méthodes de programmation, les machines elles aussi évoluent. Leur perfectionnement se traduit non seulement par une

amélioration des performances technologiques, mais également par une modification de leur organisation logique. Ainsi, à l'heure actuelle, la mode est à la simultanéité. La plupart des constructeurs ont introduit dans leur matériel la possibilité de fonctionnement en parallèle de plusieurs organes, et l'on rencontre fréquemment des dispositifs permettant, soit d'exécuter simultanément plusieurs programmes, soit tout au moins de satisfaire à certaines priorités. La programmation automatique doit s'adapter à cette évolution de manière à tirer des perfectionnements du matériel tout le rendement souhaitable.

Quelques essais ont donc été entrepris en vue d'élaborer des langages qui autoriseraient la description de processus comportant, soit des séquences simultanées, soit des fonctions prioritaires. Mais les efforts les plus nombreux portent sur l'adaptation des procédés de mise en œuvre des langages actuels en vue de tirer parti des nouvelles caractéristiques des machines. Par exemple, la plupart des compilateurs assurent, là où il est possible, le fonctionnement en parallèle des organes périphériques. Pourtant le caractère séquentiel de la majorité des langages dont on dispose ne se prête guère à l'introduction, lors de leur traitement, d'une simultanéité très poussée à l'intérieur d'un même programme. On s'est alors tourné vers l'exécution simultanée de plusieurs programmes différents. Cette dernière méthode pose, entre autres, un délicat problème de protection d'un programme contre toute perturbation provoquée par un autre programme et demande une affectation dynamique des mémoires qui n'est plus uniquement basée sur des motifs d'économie, mais est conditionnée à tout moment par l'état global des divers programmes en cours d'exécution. Des solutions ont déjà été proposées; elles exploitent, soit de nouveaux circuits, soit de nouvelles techniques de programmation.

Ceci n'est qu'un aspect particulier du développement parallèle des calculatrices, d'une part, des procédés de programmation automatique, d'autre part. Outil de travail dans l'exploitation des machines à traiter l'information, la programmation automatique est naturellement conditionnée par toute modification apportée à ces machines.

Mais l'influence inverse, quoique moins évidente, n'en est pas moins réelle : les recherches en programmation automatique ont plus d'une fois provoqué une évolution de la structure logique des machines. Cela provient de ce que les opérations de l'esprit impliquées dans l'élaboration d'un système de programmation ou l'établissement du schéma fonctionnel d'une machine sont formellement identiques. Nous voudrions simplement illustrer par un exemple cette réaction des langages sur les machines.

La notation dite polonaise fut introduite en 1929 par G. Lukasiewicz dans le but d'éliminer les parenthèses dans les expressions en algèbre logique, mais elle peut aussi être utilisée en algèbre ordinaire. Elle consiste simplement à placer le symbole d'une opération avant les deux opérandes au lieu de l'insérer entre ces opérandes. Ainsi :

$$\begin{array}{lll}
 a + b & \text{s'écrira} & + a b \\
 a \times b & & \times a b \\
 a + b + c = (a + b) + c & & + + a b c \\
 (a + b) \times c & & \times + a b c
 \end{array}$$

dans ces deux derniers cas, le premier opérande correspondant au signe de tête est constitué par l'ensemble  $+ a b$  équivalant à  $a + b$ .

Cette notation fut utilisée par certains comme langage simplifié de programmation, après en avoir cependant inversé la forme : le symbole opératoire venait cette fois après les opérandes. En notation polonaise inverse, les exemples précédents deviennent respectivement :

$$\begin{array}{l}
 a b + \\
 a b \times \\
 a b + c + \\
 a b + c \times
 \end{array}$$

De même, si l'on considère le calcul d'une fonction comme une opération élémentaire à un opérande,

$$\log [(a + b)/(c + d)] \quad \text{devient} \quad a b + c d + / \log$$

L'avantage d'un tel mode de programmation réside dans le fait que les opérations sont notées exactement dans l'ordre de leur exécution sur machine. Dans le dernier exemple, la machine doit successivement, pour calculer la valeur de l'expression :

- 1 - appeler l'opérande  $a$ ;
- 2 - appeler l'opérande  $b$ ;
- 3 - en faire la somme et conserver le résultat;
- 4 - appeler l'opérande  $c$ ;
- 5 - appeler l'opérande  $d$ ;
- 6 - en faire la somme et conserver le résultat;

- 7 - calculer le quotient des deux résultats précédents;
- 8 - prendre le logarithme de ce quotient.

Des machines ont été construites en vue d'utiliser comme langage machine des représentations de ce genre : la K D F 9 de l'English Electric et la Burroughs B 5.000. L'unité arithmétique de ces machines comporte une mémoire intermédiaire ayant une structure de stack. L'ordre d'appel d'une quantité consiste à extraire ce nombre de la mémoire générale et à le placer au sommet de la pile.

Cet exemple est bien sûr un cas extrême et encore rare où toute la logique d'une machine est conditionnée par les caractéristiques d'un langage. Cependant, si l'on examine de près les matières actuellement sur le marché, on trouve nombre de détails, de petits perfectionnements logiques, ou même technologiques, qui ne s'expliquent que par le souci de faciliter la mise en œuvre des procédés de programmation automatique.

Nous ne voudrions pas terminer sans dire quelques mots d'un problème qui nous est particulièrement cher, car le Bureau des Langages du Centre de Calcul de l'Université de Liège s'est depuis quelque temps déjà attelé à sa résolution : il s'agit de la traduction automatique des programmes d'un langage machine dans un autre. Il arrive fréquemment, à la suite d'échange de programmes par exemple, que l'on dispose d'un programme en langage machine prêt à l'emploi pour une machine déterminée, et que l'on doit exécuter les calculs sur une machine d'un autre type. Or, la reconversion manuelle, comprenant d'abord l'analyse du programme d'origine et ensuite l'écriture d'un programme équivalent destiné à la machine dont on dispose, demande souvent un très gros travail au programmeur humain.

Le premier procédé de programmation automatique utilisé pour résoudre ce problème est l'interprétation, c'est-à-dire l'écriture d'un programme simulant sur la deuxième machine le fonctionnement de la première.

Toutefois, cette méthode, qui donne de bons résultats dans le traitement d'un langage de programmation simplifié, conduit à des pertes de temps considérables lorsqu'on veut l'appliquer à un langage machine. Si le calcul à exécuter est quelque peu répétitif, le rendement tombe à un niveau tel que la simulation doit être abandonnée.

La mise au point de programmes traducteurs effectuant sans intervention humaine la transposition de programmes d'un langage machine dans un autre langage machine, ou même dans un langage symbolique qui servirait d'intermédiaire et que l'on pourrait ultérieurement compiler, s'avère un

travail d'une grande ampleur, hérissé de nombreuses difficultés. L'un des principaux obstacles réside dans le fait que le passage d'un langage symbolique au langage machine implique habituellement une dégradation de l'information contenue dans le programme, alors que le passage inverse se propose de restituer cette information dans sa forme initiale. Evidemment, nous excluons le cas trivial où l'on désire passer d'une machine à une autre ayant une structure presque identique.

Des expériences récentes de traduction entre deux langages machines appartenant à des matériels de structures logiques profondément différentes ont donné des résultats encourageants. Les programmes traducteurs sont énormes; ils sont asservis à un cas particulier et aucune méthode générale ne peut encore s'en dégager. Il apparaît cependant clairement que le problème peut être résolu, et l'on doit s'attendre à des applications pratiques de ce procédé dans les prochaines années.

Ainsi, la programmation automatique, qui était partie de la simplification de l'écriture des programmes, gouverne déjà la plupart des communications entre machines. Elle a même réagi, par la mise au point de nouveaux langages, sur les communications entre hommes.

---

PUBLICATIONS REÇUES

ONTVANGEN PUBLICATIES

- 1) *Indagationes Mathematicae* - Koninklijke Nederlandse Akademie van Wetenschappen (Vol. XXIII, fasc. 1 à 5 - Vol. XXIV, fasc. 1 à 5 - Vol. XXV, fasc. 2).
- 2) *Revue IBN* (Institut Belge de Normalisation), n<sup>os</sup> 11 et 12, novembre-décembre 1963.  
*BIN Revue* (Belgisch Instituut voor Normalisatie), n<sup>os</sup> 11 en 12, november-december 1963.
- 3) *Annales de Sciences Economiques Appliquées* (Université Catholique de Louvain), n<sup>o</sup> 5, décembre 1963.

**A propos de la 3<sup>me</sup> Conférence Internationale  
de la Fédération Internationale de Recherche Opérationnelle (IFORS)**

La troisième Conférence Internationale de l'IFORS, qui s'est tenue à Oslo du 1<sup>er</sup> au 5 juillet 1963, a été incontestablement une réussite, du point de vue de l'information scientifique et des contacts entre chercheurs, deux objets fort corrélés, comme on le sait. L'organisation avait été conçue de manière à favoriser ces contacts : lunches en commun, logement du plus grand nombre de congressistes à la Cité Universitaire et — à proximité des salles de conférences — une salle de réunion spacieuse et confortable, spécialement destinée aux conversations et au travail « hors cadre ». Nous avons personnellement eu l'occasion de nouer ainsi des relations avec des collègues étrangers, relations qui continuent à porter leurs fruits par l'échange de travaux et par des projets de rencontres d'équipes de chercheurs, mais aussi de faire la connaissance ou plus ample connaissance de « chercheurs opérationnels » belges, et d'apprendre l'objet de leurs travaux... Même l'excursion en bateau a été un catalyseur de rencontres et d'entretiens. C'est une loi générale d'ailleurs que ce type d'excursions est, à cet égard, de grande efficacité.

Les congressistes n'ont eu qu'à se louer de l'amabilité des organisateurs et de leurs collaborateurs. Parmi les responsables de l'organisation, citons le si affable Monsieur Kreweras, dont ni le dévouement ni l'urbanité n'ont jamais été pris en défaut.

---

Vu l'abondance des communications, témoignage de la vitalité de la Recherche Opérationnelle, les « sessions » se tenaient parallèlement, à raison de trois ou quatre par demi-journée. Une seule parmi ces trois ou quatre sessions bénéficiait de la traduction.

Citons, parmi les centres d'intérêt, la programmation mathématique, les problèmes de trafic (y compris les problèmes d'attente et d'occupation), les critères de décision, la simulation, des problèmes de production et de gestion, les méthodes de prévision, les investissements à long terme, l'organisation de la recherche, des problèmes de mesure, la méthode d'input-output à l'intérieur des firmes, des problèmes d'achat et de vente, la santé, l'agriculture et la foresterie, et des applications militaires.

La programmation mathématique et le trafic ont fait l'objet, respectivement, de deux et de trois sessions, s'imposant ainsi comme des questions à l'ordre du jour.

C'est à propos de certaines communications relevant de ces deux catégories de problèmes que nous donnons ci-dessous quelques informations

fragmentaires. L'une ou l'autre de ces communications relèvent en fait des deux catégories.

*La programmation mathématique.*

Si la programmation non linéaire retient de plus en plus, et par la force des choses, l'attention des chercheurs, la programmation linéaire suscite encore bien des problèmes de réalisation pratique, de simplification et, partant, d'expérimentation sur machine, les évaluations s'exprimant, à cet égard, en termes de « performances ».

On retiendra notamment l'exposé, par P. Broise et P. Huard, de l'Electricité de France, de résultats d'expériences relatifs à la méthode de décomposition de Dantzig et Wolfe, celle-ci s'avérant réellement, dans bien des cas, comme très efficace et économique, un compte rendu, par L. Cutler et Ph. Wolfe, de la Rand Corporation, de plusieurs expériences sur diverses variantes de la méthode simpliciale, et la description, par A. Le Garff et Y. Malgrange, de la Compagnie des Machines Bull, d'un algorithme de résolution des programmes linéaires à valeurs entières, inspiré de méthodes booléennes, et permettant d'éviter certains inconvénients opératoires d'autres méthodes, description assortie, elle aussi, de résultats d'expériences.

A. Charnes, W.W. Cooper et K. Kortanek, de la Northwestern University et du Carnegie Institute of Technology, ont présenté un nouveau type de modèle, « la programmation semi-infinie », impliquant une dualité aussi précise et fine que celle de la programmation linéaire, et qui contient, en particulier, tous les modèles connus de programmation convexe. Ce modèle semble offrir un moyen de résolution effective de plusieurs problèmes non linéaires.

J.B. Rosen et J.C. Ornea, de la Compagnie Shell, ont décrit une méthode itérative de résolution de problèmes non linéaires se présentant notamment dans la mise au point de systèmes de fabrication et de transport. Elle est basée sur la décomposition des problèmes non linéaires indépendants relativement petits.

J.C.G. Boot et H. Theil, de l'Institut d'Econométrie de la Nederlandse Economische School, ont traité du problème de la maximisation d'une fonction quadratique définie, dans le cas où quelques-unes, ou toutes les variables, sont astreintes à prendre des valeurs entières. Ils ont décrit une solution simple, qui peut être considérée comme efficace lorsque le nombre des variables n'est pas trop grand.

Le Professeur R. Pallu de la Barrière, de l'Université de Caen, a présenté très clairement, quoique au pied levé, des recherches de D.C. Carton, de l'Electricité de France, sur l'adaptation de l'algorithme de Howard à la

résolution de processus de décision markoviens non stationnaires périodiques, tels qu'on en rencontre dans certains phénomènes saisonniers. Il a proposé aussi sa propre solution de pareils problèmes.

W.S. Jewell, de l'Université de Californie, à Berkeley, a montré comment on peut étendre assez considérablement le champ d'application de ce même algorithme de Howard, ainsi que de la méthode de programmation dynamique de Bellman, considérée dans son utilisation à la recherche de politiques transitoires optimales pour des processus de décisions discontinus markoviens. On peut en fait substituer aux chaînes de Markov des processus de renouvellement markoviens.

B. Roy, de la Société d'Economie et de Mathématique appliquée, de Paris, a signalé les possibilités qu'offre en programmation mathématique la méthode de description segmentée, qu'il avait présentée déjà, avec M. Simonard, dès 1961, dans la Revue Française de Recherche Opérationnelle, mais dont il a, depuis, approfondi l'étude.

Citons enfin l'excellence des interventions de J. Abadie, de l'Electricité de France, qui, dans la discussion de la communication de Roy, a rapproché la méthode présentée d'une méthode due à Fourier, et qui, à propos de l'exposé de Boot, a signalé l'intérêt éventuel, pour la question traitée, du célèbre lemme de la théorie des nombres de Minkowski, sur les systèmes d'inéquations diophantiennes linéaires.

#### *Les problèmes de trafic.*

Des sujets divers ont été traités, certains d'ordre essentiellement théorique ou méthodologique, d'autres consistant en l'étude de cas.

Arrêtons-nous d'abord aux premiers.

L. Kosten, de la Technische Hogeschool de Delft, a traité de problèmes de décision continus markoviens, tels qu'il s'en présente à propos de problèmes de réservation de places dans les compagnies de transport aérien. Il a présenté une méthode de recherche d'une politique optimale, qui diffère de la méthode de Howard.

P. Noar, de l'Institut de Technologie d'Israël et de l'Université de la Caroline du Nord, a présenté, dans la théorie des processus d'attente, des modèles très généraux faisant intervenir notamment des priorités et des interruptions de service. Il a énoncé, à leur sujet, plusieurs résultats d'intérêt économique.

J. Teghem a exposé les résultats sur des problèmes d'attente et d'occupation, dus à des chercheurs du Centre d'Etudes de Recherche Opérationnelle, de Bruxelles, et de l'Université de Bruxelles, résultats portant sur des problèmes de priorité, l'amélioration technique de méthodes de recherche de probabi-

lités d'état transitoires, la recherche de probabilités d'état stationnaires pour des services en série, une application de la méthode de la chaîne de Markov incluse et la recherche de distributions de temps d'attente et d'occupation par les méthodes de Lindley, de Takacs et de Benes, la dernière se caractérisant par la très grande généralité des hypothèses relatives aux lois des arrivées et des durées de service, mais se confinant encore, actuellement, au cas d'un seul guichet.

Venons-en maintenant à des études de cas que l'on peut assimiler, vu leur aspect particulier, à ce genre de recherches.

Un membre de la Société d'Economie et de Mathématique appliquée, de Paris, a décrit, au nom de R. Loue, les différents facteurs ayant dû être pris en considération dans le problème de la construction éventuelle d'un pont entre l'île de Ré et le continent. Il a exposé notamment les méthodes qu'on a utilisées pour l'évaluation de l'intérêt national et de l'intérêt local de l'ouvrage, ainsi que le mode d'établissement des perspectives de trafic liées aux diverses solutions techniques.

R.M. Oliver, de l'Université de Californie, à Berkeley, a traité, avec résultats numériques et graphiques à l'appui, d'un problème de réduction des délais de transmission d'une lettre entre deux bureaux de poste.

T. Rallis, de l'Université Technique du Danemark, a montré comment ont été appliquées à un problème de capacité de trafic de la gare centrale de Copenhague des formules d'Erlang et d'Arne Jensen sur l'indisponibilité et les durées d'attente dans la théorie des communications téléphoniques.

J. J. Agard, du Groupe de Recherche Opérationnelle d'Air France, a présenté une méthode de détermination par simulacre des courbes de charge relatives à la réalisation d'un programme de vol.

Miss E.M.L. Beale, de C.E.I.R (UK) Ltd., a décrit deux problèmes de transport ayant été résolus grâce à des méthodes mises au point par Land et Doig pour le problème général de programmation en nombres entiers.

J.C. Holl a décrit, au nom d'une équipe de la Société d'Economie et de Mathématique appliquées, de Paris, la simulation des manœuvres de remorquage des navires dans un port.

J.P. Jeanniot et Miss A. Bodnarchuk, des Trans Canada Air Lines, ont présenté un système automatisé de calcul des frais de main-d'œuvre, dans lequel a été incorporé un modèle de recrutement optimum basé sur des principes de programmation dynamique.

Y. Rapp, de la Telefonaktiebolaget L.M. Ericsson, à Stockholm, a développé quelques principes de planification des réseaux téléphoniques à plusieurs secteurs, avec le souci de les présenter sous une forme adaptée à l'utilisation de calculateurs électroniques.

Par ailleurs, relevons encore quelques communications qui nous ont paru marquantes en envisageant surtout le point de vue économique.

En *session plénière*, la conférence de R. Frisch a traité du sujet : « Un système de mise en œuvre d'un plan économique national sans fixation des quantités détaillées par une autorité centrale ». Le célèbre économiste a décrit, en y insérant certaines remarques bien personnelles et souvent caustiques, la méthode d'établissement des programmes nationaux ou supranationaux. Il a notamment mis en garde contre certaines erreurs fréquentes, comme celle qui consiste à fixer les buts à atteindre avant l'analyse préalable de l'économie, ou encore celle de construire des plans pour des périodes de temps qui ne se recouvrent pas.

#### *Les investissements.*

Les problèmes d'investissement sont parmi ceux qui sont les plus étudiés, surtout par les firmes qui utilisent des installations fixes importantes. Une session très suivie s'est tenue sur ce thème.

B. Barberi, directeur de l'Institut de Statistique de Rome, a énoncé quelques remarques d'ordre méthodologique sur les problèmes d'investissement à long terme, notamment sur les relations entre les aspects micro et macroéconomiques des investissements et les problèmes de prévision.

P. Lhermitte et F. Bessière, d'Electricité de France, offraient une communication sur les possibilités de la programmation non linéaire appliquée au choix des investissements. Partant du programme linéaire déjà utilisé par l'E.D.F. (224 contraintes, 253 inconnues), les auteurs ont cru que les améliorations ne devaient pas porter sur une extension du modèle existant mais devaient se développer suivant une voie différente : une fonction économique non linéaire. Sans pouvoir être quadratique, cette fonction peut être de différents types en relation avec le mode de résolution choisi. La dimension du programme non linéaire homologue du linéaire antérieur est encourageante : 48 contraintes et 69 inconnues.

MM. J. Carteron et J. Carpentier, d'Electricité de France, ont traité le problème de la gestion optimale d'un ensemble de production d'énergie électrique, Avec un équipement donné il s'agit de déterminer des règles de gestion qui rendent minimum l'espérance mathématique de la somme des dépenses thermiques et des pénalités de défaillance, la valeur de la pénalité de défaillance étant laissée sous forme de paramètre. Dans ce problème, la programmation dynamique s'indique naturellement, mais dans la pratique, les auteurs ont trouvé son application impossible en raison de la dimension. Ils se sont donc orientés vers l'emploi de la simulation, tout en poursuivant les recherches dans le sens de la simplification des programmes dynamiques : les

solutions de ces problèmes simplifiés donneraient des indications pour la simulation.

M. Algan, de la firme DIVO de Francfort, a présenté une communication, préparée avec la collaboration de MM. J. Ceron et P. Bertier, de la SEMA à Paris, traitant d'une méthode pratique de détermination d'un plan optimum d'investissement. Les auteurs rappellent les méthodes classiques pour décider du choix des investissements, en y introduisant la considération de la source du financement.

J. Melèse, de l'AUROC, a exposé une méthode pratique de mise au point d'un programme d'équipement, qui met en relief les variantes intéressantes par rapport à différents critères, dont la flexibilité. Le problème du choix du plan s'en trouve ainsi réduit.

Quant à R. Solt, d'Albert E. Reed & Co, Ltd, il a décrit un cas d'évaluation d'investissements à long terme. Compte tenu de prévisions de demande, de prix et d'autres variables, on a déterminé les différentes possibilités d'investissements (32). Le problème a été d'évaluer ces possibilités, en fonction de critères de profit, dans un éventail de situations (12) correspondant à la réalisation d'événements aléatoires.

#### *Problèmes commerciaux.*

R.L. Ackoff, du Case Institute of Technology, a émis quelques observations sur la recherche opérationnelle dans les problèmes de « marketing » aux Etats-Unis. Se référant à de nombreux cas traités par son groupe, il a dénoncé certaines idées erronées qui ont cours sur l'équivalence des produits, les habitudes d'achat, les frais de vente, les effets de la promotion de ventes et les prévisions, tout en soulignant la rentabilité de l'application des méthodes de la recherche opérationnelle dans ces domaines.

G. Maarek a présenté une communication préparée avec M. R. Descamps, à la SEMA, sur la façon d'établir les politiques de tarification et de distribution (affectation des consommateurs et des négociants aux différents bassins) de charbons industriels par Charbonnages de France.

J. Ferrier, de la Marine française, a proposé une formule simple adaptée à la gestion automatique des approvisionnements. Cette formule à trois paramètres est basée sur l'hypothèse que les demandes individuelles se présentent suivant une loi de Poisson et portent sur des quantités (grappes) répondant à une distribution quelconque. Une mise à l'épreuve rétrospective de cette formule a abouti à des résultats probants.

Une expérience de prédiction à court terme a été relatée par E. Kay et J. S. Hampton. Le groupe d'étude sur la prévision de la British O. R. Society a été amené à comparer la méthode de lissage exponentiel à celle des

moyennes mobiles sur des séries temporelles réelles et artificielles. Il apparaît que, du point de vue des erreurs de prédiction, les deux méthodes ne diffèrent pas de manière significative, mais si l'on tient compte de la diminution du coût relatif, le lissage exponentiel est en pratique supérieur.

M. Sakaguchi, de l'Université des Electro-Communications de Chôfu, Tokyo, a exposé les solutions mathématiques de problèmes d'enchères fermées.

En ce qui concerne d'autres sessions intéressantes, il faut mentionner celle sur les *plans de production* (« scheduling ») où des communications traitaient de la régulation de la production d'acier, du planning de fabrication de nombreux articles sur des équipements communs, de la succession de modèles sur une chaîne d'assemblage, de programmation mathématique dans l'industrie minière.

En plus des sessions consacrées à des exposés, deux demi-journées étaient prévues pour des réunions en groupes d'étude. On peut signaler spécialement le groupe d'étude des *méthodes de prévision* qui a rassemblé un grand nombre de participants sous la présidence de M. Schalkwijk. Les exposés succincts et les discussions ont porté sur les techniques statistiques de prévision à court terme, des méthodes de prévision à long terme et des exemples pratiques.

On peut se rendre compte que dans l'ensemble ce congrès fut très profitable à la diffusion des idées neuves dans les différents domaines abordés.

Du point de vue de l'organisation, on comprend que l'abondance des communications nécessite la subdivision des thèmes et amène à une formule de sessions parallèles. Malheureusement, toute classification étant arbitraire, certaines communications intéressant un même participant peuvent être présentées dans des sessions différentes tenues simultanément. Cette situation a été aggravée dans ce cas par le fait que l'horaire de chaque session n'était pas connu d'avance et excluait ainsi un certain « panachage ».

Certains ont pu regretter que la traduction simultanée n'était pas assurée pour toutes les sessions, mais il faut reconnaître que cela a surtout gêné les participants de langue anglaise.

Quant à la publication préalable des résumés des communications on ne peut que s'en féliciter.

Nous ne pouvons terminer sans souligner l'impression de force créatrice qui se dégage de certaines équipes de chercheurs étrangers, telles que celle de l'Electricité de France, soutenues par des organismes privés ou l'Etat aux vues larges. Il est des endroits de par le monde, où l'on a compris que la recherche scientifique est payante.

J. TEGHEM.

F. JUCKLER.