Le Travail Humain
tome 34, n° 1/1971
pages 157 à 176

LES ALGORITHMES EN PSYCHOLOGIE ET EN PÉDAGOGIE DÉFINITION ET INTÉRÊTS (1)

par P. Vermersch (2)

Laboratoire de Psychologie du Travail de l'E.P.H.E., 41, rue Gay-Lussac, 75-Paris (5e)
Equipe de Recherche Associée au C.N.R.S.

RÉSUMÉ

Cet article présente et définit le concept d'algorithme dans une première partie. Il retrace d'abord la filiation de ce concept dans l'histoire des mathématiques, aboutissant à la formalisation actuelle. Un exemple est développé, sous la forme de la « Machine de Turing », qui va servir de base de comparaison avec les définitions que l'on rencontre dans les sciences humaines.

Ces différents niveaux de définition : algorithme absolu, algorithme aboutissant, consigne de type algorithmique, sont mis en relation avec les propriétés des différents systèmes servant de support à leur application. Puis on définit les distinctions que fait Landa entre algorithme de reconnaissance et algorithme de résolution, en essayant de se placer dans le cadre théorique de la psychologie soviétique.

Enfin, on examine les divers modes de représentation des algorithmes.

La deuxième partie a pour but d'examiner l'intérêt de la notion d'algorithme dans les sciences humaines.

Les arguments développés portent d'abord sur l'observation, en pédagogie et en psychologie du travail par exemple, de comportements algorithmiques spontanés.

Les travaux de Vergnaud permettent alors de développer les implications théoriques et d'établir une distinction entre processus algorithmiques et conduite algorithmique.

Une autre perspective est alors présentée dans le prolongement de la théorie du reflet.

Enfin, on replace cette notion dans les courants de pensée pédagogiques visant à introduire l'apprentissage de méthodes générales de pensée.

On conclut sur les limites du concept d'algorithme.

Le concept d'algorithme est indissociable du développement des mathématiques. Il se rencontre actuellement dans de nombreux domaines, qu'ils soient dérivés plus ou moins directement des mathématiques comme la recherche opérationnelle ou les techniques de programmation, ou bien qu'au contraire ces domaines cherchent à s'en rapprocher pour mettre à profit la rigueur que peuvent offrir des outils formels, comme c'est le cas des différentes branches de la psychologie.

Une des premières questions qui doit alors être soulevée est de savoir si précisément cet éventail assez large d'utilisation recouvre bien la même définition du concept d'algorithme.

L'éclaircissement de ce point nous paraît essentiel à la compréhension de la place qu'occupe ce concept dans les sciences humaines. La plus grande partie de cet article essentiellement théorique sera donc consacrée à une pré-

(1) Cet article a été rédigé dans le cadre d'une recherche financée par la Délégation générale à la Recherche Scientifique et Technique, Action concertée : Enseignement programmé, contrat n° 68.01.160.

(2) Attaché de Recherche au C.N.R.S.

sentation et à une définition des algorithmes. L'étude des propriétés des algorithmes à différents niveaux de définition nous conduira à essayer de mettre en évidence plusieurs éléments de réponse à la question : « Quel est l'intérêt et la place des algorithmes dans les sciences humaines ? » La description des applications déjà réalisées donnera matière à un autre article.

A) PROBLÈMES DE DÉFINITION

1. Algorithme et développement des mathématiques

Trahtenbrot (1963) a retracé l'évolution de la notion d'algorithme en mathématique.

Dans cette discipline, à un niveau de définition intuitif, un algorithme est synonyme de procédé de calcul, de méthode de résolution. Les opérations arithmétiques les plus élémentaires sont des algorithmes ; quand on a appris l'algorithme de l'addition, on sait faire toutes les additions.

Historiquement, une proposition sur l'existence d'un algorithme, pour un problème donné, était accompagnée de sa description complète ; le caractère algorithmique pouvait immédiatement être mis en évidence par la simple mise en œuvre des règles fournies. Cela peut permettre d'expliquer pourquoi le concept de procédé de calcul a mis tant de temps à s'individualiser, à se séparer d'une description matérielle qui en cachait la généralité.

La raison de l'individualisation du concept par rapport à un support matériel réel se trouve dans le développement des préoccupations des mathématiciens. Ce développement a été lié en grande partie à la recherche de procédés de résolution de plus en plus puissants, c'est-à-dire de techniques permettant de traiter sous une forme unique des classes de problèmes de plus en plus larges.

Trahtenbrot donne l'exemple de l'algorithme de l'extraction de la racine carrée ; on peut aussi essayer de construire l'algorithme pour l'extraction d'une racine d'ordre arbitraire d'un nombre donné quelconque ; on peut encore formuler le problème de façon plus générale : car extraire la racine de $n^{\text{ième}}$ ordre d'un nombre veut dire résoudre l'équation : $x^n - a = 0$.

On est conduit ainsi au problème qu'a soulevé Leibnitz, la construction d'un procédé de résolution pour un problème mathématique arbitraire.

Cependant, un certain nombre de problèmes n'ont pu être résolus de façon générale : il a été possible de construire des solutions pour des cas particuliers, mais pas *une* solution pour le cas général.

Devant l'impossibilité de résoudre certains problèmes mathématiques, les mathématiciens ont posé le problème de la démonstration mathématique de cette impossibilité. D'où simultanément plusieurs nécessités :

— d'une part, le concept de procédés de résolution s'est dégagé, abstraction faite d'un problème précis ou d'une solution précise ;
— d'autre part, la notion intuitive d'algorithme devait être reformulée, de manière à en faire un objet mathématique rigoureux mais commode, permettant un calcul formel.

De nombreux auteurs ont construit un tel objet, que ce soit « la *recursive enumerability* de Kleene (1936), la calculabilité de Godel (1934), la définition combinatoire de Rosse (1935), la *definability* de Church (1936), la *computability* de Turing (1936) et de Post (1936), la théorie des algorithmes de Markow (1954) » (Lorenzen, 1969).

Il est remarquable que tous les auteurs soient d'accord pour admettre l'équivalence de ces constructions formelles.

Nous avons jusqu'à présent uniquement parlé de procédés de calcul ; il serait plus correct de parler d'une formalisation de la « calculabilité » (Gross et Lentin, 1967). La calculabilité est alors une notion totalement abstraite, qui ne tient compte ni de problèmes précis ni des contraintes spatio-temporelles nécessaires à la réalisation effective d'un calcul donné.

Il nous semble particulièrement illustratif de montrer que l'on peut formaliser la notion de « calculabilité » sous la forme d'une machine à calculer idéale. C'est l'idée (développée ici de façon tout à fait intuitive et non pas mathématique) qui correspond à la construction d'une machine de Turing.

2. Un exemple de formalisation : la machine de Turing

Notre intention n'est pas de faire une présentation technique de la formalisation connue sous le nom de machine de Turing. Nous essaierons seulement d'en développer les aspects essentiels avec les conséquences qui en découlent, afin de donner une référence suffisamment claire pour établir des comparaisons avec la définition d'un algorithme telle qu'on la trouve en psychologie et en pédagogie.

L'idée de base de cette formalisation est de construire une machine à calculer — au moins (et surtout) sur le papier — telle que toute solution puisse être recodée sous la forme d'un programme de cette machine. C'est d'ailleurs ce programme (ou schéma fonctionnel) qui définit une machine de Turing donnée.

Nous suivrons Gross et Lentin (1967) pour décrire une machine de Turing, elle se compose :

— d'une *unité centrale* qui peut prendre un certain nombre d'états internes (q_1, q_2, q_3), en nombres finis ;
— d'une *bande* sur laquelle sont inscrites au départ les données à traiter et sur laquelle viennent s'inscrire les calculs intermédiaires.

La bande est divisée en cases, dans lesquelles figure un seul symbole de l'alphabet extérieur S_0, S_i, S_n. Elle est illimitée ;

— d'une tête de lecture écriture qui assure la communication entre l'unité centrale et la sortie ; elle n'opère que sur une seule case à la fois et elle peut : remplacer le symbole lu par un nouveau symbole, déplacer éventuellement la bande à sa gauche ou à sa droite.

Cette machine comprend des opérations présentant le maximum de simplicité. L'opération d'addition qui est élémentaire dans une machine à calculer réelle est ici composée de plusieurs opérations (voir Trahtenbrot, 1963).

Cela exclut donc tout souci de réalisme. Les principes de construction d'une telle machine ne tiennent pas compte des contraintes spatio-temporelles qui seraient liées à la réalisation effective d'un calcul. Nous retrouvons bien ici les implications de la notion abstraite de calculabilité. Le système des opérations élémentaires est extrêmement limité, un commandement prescrit seulement le remplacement d'un signe S_i, gardé dans la case vue par un autre sentation et à une définition des algorithmes. L'étude des propriétés des algorithmes à différents niveaux de définition nous conduira à essayer de mettre en évidence plusieurs éléments de réponse à la question : « Quel est l'intérêt et la place des algorithmes dans les sciences humaines ? » La description des applications déjà réalisées donnera matière à un autre article.

A) PROBLÈMES DE DÉFINITION

1. Algorithme et développement des mathématiques

Trahtenbrot (1963) a retracé l'évolution de la notion d'algorithme en mathématique.

Dans cette discipline, à un niveau de définition intuitif, un algorithme est synonyme de procédé de calcul, de méthode de résolution. Les opérations arithmétiques les plus élémentaires sont des algorithmes ; quand on a appris l'algorithme de l'addition, on sait faire toutes les additions.

Historiquement, une proposition sur l'existence d'un algorithme, pour un problème donné, était accompagnée de sa description complète ; le caractère algorithmique pouvait immédiatement être mis en évidence par la simple mise en œuvre des règles fournies. Cela peut permettre d'expliquer pourquoi le concept de procédé de calcul a mis tant de temps à s'individualiser, à se séparer d'une description matérielle qui en cachait la généralité.

La raison de l'individualisation du concept par rapport à un support matériel réel se trouve dans le développement des préoccupations des mathématiciens. Ce développement a été lié en grande partie à la recherche de procédés de résolution de plus en plus puissants, c'est-à-dire de techniques permettant de traiter sous une forme unique des classes de problèmes de plus en plus larges.

Trahtenbrot donne l'exemple de l'algorithme de l'extraction de la racine carrée ; on peut aussi essayer de construire l'algorithme pour l'extraction d'une racine d'ordre arbitraire d'un nombre donné quelconque ; on peut encore formuler le problème de façon plus générale : car extraire la racine de n^ième^ ordre d'un nombre veut dire résoudre l'équation : $x^n - a = 0$.

On est conduit ainsi au problème qu'a soulevé Leibniz, la construction d'un procédé de résolution pour un problème mathématique arbitraire.

Cependant, un certain nombre de problèmes n'ont pu être résolus de façon générale : il a été possible de construire des solutions pour des cas particuliers, mais pas *une* solution pour le cas général.

Devant l'impossibilité de résoudre certains problèmes mathématiques, les mathématiciens ont posé le problème de la démonstration mathématique de cette impossibilité. D'où simultanément plusieurs nécessités :

— d'une part, le concept de procédés de résolution s'est dégagé, abstraction faite d'un problème précis ou d'une solution précise ;
— d'autre part, la notion intuitive d'algorithme devait être reformulée, de manière à en faire un objet mathématique rigoureux mais commode, permettant un calcul formel.

De nombreux auteurs ont construit un tel objet, que ce soit « la *recursive enumerability* de Kleene (1936), la calculabilité de Godel (1934), la définition combinatoire de Rosse (1935), la *definability* de Church (1936), la *computability* de Turing (1936) et de Post (1936), la théorie des algorithmes de Markow (1954) » (Lorenzen, 1969).

Il est remarquable que tous les auteurs soient d'accord pour admettre l'équivalence de ces constructions formelles.

Nous avons jusqu'à présent uniquement parlé de procédés de calcul ; il serait plus correct de parler d'une formalisation de la « calculabilité » (Gross et Lentin, 1967). La calculabilité est alors une notion totalement abstraite, qui ne tient compte ni de problèmes précis ni des contraintes spatio-temporelles nécessaires à la réalisation effective d'un calcul donné.

Il nous semble particulièrement illustratif de montrer que l'on peut formaliser la notion de « calculabilité » sous la forme d'une machine à calculer idéale. C'est l'idée (développée ici de façon tout à fait intuitive et non pas mathématique) qui correspond à la construction d'une machine de Turing.

2. Un exemple de formalisation : la machine de Turing

Notre intention n'est pas de faire une présentation technique de la formalisation connue sous le nom de machine de Turing. Nous essaierons seulement d'en développer les aspects essentiels avec les conséquences qui en découlent, afin de donner une référence suffisamment claire pour établir des comparaisons avec la définition d'un algorithme telle qu'on la trouve en psychologie et en pédagogie.

L'idée de base de cette formalisation est de construire une machine à calculer — au moins (et surtout) sur le papier — telle que toute solution puisse être recodée sous la forme d'un programme de cette machine. C'est d'ailleurs ce programme (ou schéma fonctionnel) qui définit une machine de Turing donnée.

Nous suivrons Gross et Lentin (1967) pour décrire une machine de Turing, elle se compose :

— d'une *unité centrale* qui peut prendre un certain nombre d'états internes (q_1, q_2, q_3), en nombres finis ;
— d'une *bande* sur laquelle sont inscrites au départ les données à traiter et sur laquelle viennent s'inscrire les calculs intermédiaires.

La bande est divisée en cases, dans lesquelles figure un seul symbole de l'alphabet extérieur S_0, S_i, S_n. Elle est illimitée ;

— d'une tête de lecture écriture qui assure la communication entre l'unité centrale et la sortie ; elle n'opère que sur une seule case à la fois et elle peut : remplacer le symbole lu par un nouveau symbole, déplacer éventuellement la bande à sa gauche ou à sa droite.

Cette machine comprend des opérations présentant le maximum de simplicité. L'opération d'addition qui est élémentaire dans une machine à calculer réelle est ici composée de plusieurs opérations (voir Trahtenbrot, 1963).

Cela exclut donc tout souci de réalisme. Les principes de construction d'une telle machine ne tiennent pas compte des contraintes spatio-temporelles qui seraient liées à la réalisation effective d'un calcul. Nous retrouvons bien ici les implications de la notion abstraite de calculabilité. Le système des opérations élémentaires est extrêmement limité, un commandement prescrit seulement le remplacement d'un signe S_i, gardé dans la case vue par un autre signe quelconque S_j. C'est une machine qui ne comporte que trois adresses, droite, gauche, la même place.

Aussi une machine de Turing est entièrement décrite quand on a dressé le tableau qui a tout couple S_j, q_i, c'est-à-dire à un symbole déterminé lu sur la bande et un état intérieur donné fait correspondre un triplet de sortie S_i, P, q_i (où P signifie position ou adresse).

Ce tableau est aussi appelé *schème fonctionnel* d'une machine de Turing.

On admettra de dire qu'un problème (une fonction) est soluble (calculable ou plus précisément Turing calculable), quand il existe une machine de Turing qui résout ce problème.

Cette formalisation suppose simultanément qu'une machine de Turing ne peut travailler que sur un univers complètement défini.

Pour que la machine puisse effectivement travailler, il faut que l'information initiale contenue dans chaque case du ruban corresponde à un nombre fini de signes $S_1, S_2, \ldots, S_k$ qui vont constituer l'alphabet extérieur de la machine.

Autrement dit, la machine ne travaille que sur un univers limité et standardisé. Si dans le cadre de systèmes formels, on peut ramener tout problème à une comparaison de *mots* composés de signes définis, dès que l'on quitte ce cadre très strict, le problème de codage des énoncés sous forme standard se pose aussitôt. Cependant, à ces restrictions près (de taille il est vrai), il est admis que la machine de Turing lit un signe quelconque s_i faisant partie de son alphabet extérieur sans aucun risque d'erreur. Autrement dit, là encore, il s'agit bien d'un objet mathématique qui fait abstraction de tout problème de réalisation effective, par exemple ici de l'identification réelle d'un signe.

Si nous reprenons les caractéristiques essentielles de l'objet mathématique : « La machine de Turing » correspond à la notion intuitive d' « algorithme » ; nous avons, *d'une part, un système d'opérations élémentaires* présentant le maximum de simplicité et en nombre limité, *d'autre part, une définition complète* d'une telle machine par son schème fonctionnel, *enfin*, l'existence d'un *univers standard et limité* sur lequel travaille la machine et qui correspond à son alphabet extérieur.

Toutes ces caractéristiques ont comme point commun de faire totalement abstraction des contraintes que soulève obligatoirement toute réalisation matérielle, en particulier les contraintes introduites par les calculateurs modernes — même les plus puissants — et celles qui définissent l'opérateur humain.

3. Niveaux de définition

La formalisation présentée peut maintenant servir de référence pour étudier les définitions des algorithmes que l'on trouve en sciences humaines.

Landa est un des premiers chercheurs qui a utilisé ce concept. Rappelons ici que cet auteur soviétique représente une des tendances majeures de la recherche de l'application des algorithmes en pédagogie. On trouve, dans ses très nombreux articles, une définition de la notion d'algorithme assez générale : « On appelle algorithme une suite d'opérations, tellement rigoureuse qu'elle permet de résoudre tous les problèmes d'une classe donnée », ou encore : « Un algorithme est une suite d'opérations définies en nombre fini qui permet de résoudre une classe de problèmes donnée » (1966). A sa suite, de nombreux

auteurs ont utilisé des définitions équivalentes (Gagé, Chapiro, Edwards, Ziener, Talysina...).

L'article de Birukov et Landa (1969) représente un pas important dans l'analyse méthodologique du concept d'algorithme. Ces auteurs distinguent plusieurs niveaux de définition : les « algorithmes absolus » au sens de la théorie mathématique, les « algorithmes aboutissants », les « consignes de type algorithmique », que nous allons considérer plus en détail.

Au niveau de la formalisation, la définition des algorithmes suppose selon ces auteurs que l'on fasse abstraction de toute réalisation possible, *abstraction* de tout problème posé par l'identification, *abstraction* de toute possibilité d'erreur, ce qui correspond à ce que nous avons développé : *ces abstractions ont des conséquences trop fortes* puisqu'au niveau de la réalisation des gros calculateurs par exemple on ne peut pas faire abstraction de la quantité de papier ou du temps nécessaire à l'exécution d'un calcul, c'est-à-dire des contraintes spatio-temporelles.

Gross et Lentin (1967) précisent que « certaines recherches mathématiques ont été entreprises en vue de dégager une notion plus réaliste de la calculabilité ; mais de toute façon, les théories mathématiques traitant de la calculabilité ne constituent pas une approche au calcul *pratique* » (p. 45). Birukov et Landa (1969) introduisent ici la notion d'*algorithme aboutissant*, c'est-à-dire d'algorithme permettant d'arriver à un résultat compte tenu, par exemple, des contraintes des calculateurs actuels. La notion d'algorithme aboutissant peut paraître paradoxale au premier abord puisqu'une des propriétés des algorithmes est l'effectivité, c'est-à-dire précisément la propriété d'être une solution certaine et *a priori* d'une classe de problème donnée.

Cependant, le paradoxe disparaît aussitôt si l'on se situe dans la hiérarchie des niveaux de définition des algorithmes qui est présentée ici, car dans un cas on se situe au niveau de la théorie mathématique, l'effectivité est une propriété des algorithmes absolus. Le fait qu'on puisse introduire la précision d' « algorithme aboutissant » revient bien à réintroduire les contraintes particulières d'un système donné. Si le système qui doit effectuer le calcul est un opérateur humain, pour qu'il y ait algorithme aboutissant il faut tenir compte des contraintes caractérisant cet opérateur.

Prenons par exemple le cas des opérations élémentaires : dans une machine, une opération donnée correspond à un mécanisme physique, cette opération existe et est définie dans sa mise en œuvre par construction (par exemple l'opération + dans une machine à calculer). Ce qui était opération élémentaire dans le cas de la machine de Turing est remplacé par une opération élémentaire beaucoup plus élaborée, dans une machine à calculer quelconque ; à plus forte raison quand on s'adresse à un système aussi perfectionné que l'homme, l'opération élémentaire peut être encore beaucoup plus riche et plus complexe. Elle peut faire intervenir des indications portant sur la présence d'un contexte, sa signification... Mais ces opérations élémentaires doivent, dans leur définition, tenir compte chez l'homme de la variabilité très grande des aptitudes, de la nécessité ou non d'un apprentissage préalable, du degré d'assimilation de cet apprentissage.

On arrive ainsi à la conclusion que pour être sûr qu'un algorithme donné est un algorithme aboutissant quand le système qui l'exécute est un opérateur humain, il faut résoudre le problème préalable de la définition de l'opération élémentaire pour une classe d'opérateurs ayant des caractéristiques données. On est là hors de la théorie mathématique et plutôt dans le champ de l'expérimentation empirique. Dans ce cas-là, on aura donc une « consigne de type algorithmique » selon Birukov et Landa.

Ce que nous développons à propos de l'opération élémentaire — c'est un des problèmes essentiels à l'application des algorithmes à la psychologie — pourrait être repris de la même façon à propos de l'alphabet extérieur, c'est-à-dire à propos de l'identification des objets sur lesquels s'applique l'algorithme.

On pourrait objecter à l'analyse faite par ces auteurs le fait qu'ils mélangent les propriétés des algorithmes et les propriétés des systèmes qui vont appliquer ces algorithmes. Les auteurs répondent à cela en montrant que, une fois résolus les différents problèmes permettant de caractériser un algorithme aboutissant, ces définitions sont tout à fait semblables et les propriétés doivent être les mêmes. Par exemple, la théorie des schèmes logiques (1) (Ianov) s'appliquera à tous ces niveaux.

3.1. Propriétés des algorithmes et propriétés des systèmes supports

Un algorithme est *défini*, c'est-à-dire qu'il est entièrement décrit aussi bien dans les différentes étapes élémentaires que dans l'enchaînement de ces étapes, c'est là une propriété qui est valable quel que soit le niveau de définition auquel on se réfère, encore que cette propriété ne soit pas compréhensible sans référence à un système quelconque donné, car le *degré de définition* est extrêmement relatif à ce système.

De plus, un algorithme est un ensemble d'instructions de taille finie (Gross et Lentin, 1967), même s'il comporte un très grand nombre d'instructions élémentaires. Il est *effectif*. Effectif est ici pris au sens d'une efficacité *a priori* (Vergnaud, 1968). Enfin, il est plus ou moins *puissant*, c'est-à-dire qu'il permet de résoudre une classe de problèmes plus ou moins large.

Ces propriétés décrivent bien les algorithmes, quel que soit le niveau de définition auquel on se situe. Elles laissent cependant échapper un aspect important.

Au niveau de définition qui a été caractérisé par le terme « algorithme absolu » (la théorie mathématique de la calculabilité), il est possible de savoir si un problème donné est soluble sans avoir à écrire effectivement la solution complète ; le cadre de cette théorie permet par un « calcul » de déterminer s'il existe une solution (si c'est décidable) ; encore faut-il que la classe de problème traitée soit elle-même exprimable dans le cadre d'un système formel. C'est là peut-être l'aspect le plus fort de la théorie mathématique ; mais cette propriété n'est vraie que dans le contexte de systèmes formels, d'êtres mathématiques totalement idéaux. Elle peut être encore vérifiée dans le cadre des calculateurs, mais nous avons vu que l'outil mathématique est trop puissant, et ne tient pas compte des contraintes, ne serait-ce que spatio-temporelles. Il reste cependant la possibilité de faire « tourner le programme à la main » comme disent les programmeurs.

Mais quand il s'agit de consignes de type algorithmique, les contraintes impliquées par le système (l'opérateur humain) sont telles qu'aucune de ces propriétés n'est vérifiée sans une expérimentation préalable. La psychologie

(1) Schème logique, au sens où dans les systèmes formels on parle d'axiomes ou de *schémas d'axiomes*, indépendamment de toutes caractérisations de variables.

ne peut alors dissocier les propriétés des algorithmes des limitations auxquelles ils doivent se plier pour être vérifiées. Pour l'opérateur humain, ces problèmes sont difficiles (sinon pour le moment impossibles) à poser dans le cadre de systèmes formels permettant de savoir s'ils sont « décidables » ou non (s'il existe un algorithme ou non) ; on perd là un aspect qui est la motivation majeure de la construction des algorithmes. Si, de plus, on perd aussi la possibilité de « faire tourner le programme à la main » puisque la définition des « opérations élémentaires » comme de l' « alphabet extérieur » relève d'une expérimentation, alors le principal intérêt des propriétés des algorithmes doit être trouvé dans leurs possibilités d'adaptation aux caractéristiques des individus.

Nous avons voulu montrer ici que s'intéresser seulement aux propriétés des algorithmes sans vouloir prendre en compte simultanément les propriétés des systèmes qui doivent leur servir de support conduit, d'une part, à ignorer que ces propriétés ne se retrouvent pas identiquement à tous les *niveaux*, d'autre part, à ne pas pouvoir poser clairement le problème de l'intérêt de cette notion pour les sciences humaines, compte tenu précisément de ces modifications limitatives.

Nous essaierons de poser plus complètement ce problème, en conclusion, après avoir présenté d'autres problèmes de définition posés par des psychologues.

4. Modes de représentation des algorithmes

Les divers modes de description, de représentation, de présentation des algorithmes nous semblent importants pour préciser la place et le contenu de la notion d'algorithme en psychologie et en pédagogie. En effet, l'algorithme une fois construit, se pose le problème de sa transmission. Quand nous parlons de transmission, cela peut être de plusieurs points de vue. Il y a un aspect proprement pédagogique, soulevé par Landa (1966) : un algorithme peut être donné tout fait, ou bien on peut essayer de guider l'enfant pour qu'il le découvre et jusqu'à un certain point le construire ; il s'agit donc là de modalités de l'acte pédagogique. Mais un autre aspect existe qui n'est pas complètement indépendant du premier et qui se pose dans un champ plus vaste d'application : c'est celui du support devant assurer l'enseignement ou la communication à un utilisateur éventuel (élève, travailleur...). Nous aborderons ce problème du type de support et, plus particulièrement : 1° les différentes notations symboliques permettant d'écrire un algorithme ; 2° les différentes représentations graphiques utilisées.

En effet, que ce soit dans le cadre de la psychologie du travail ou dans celui plus délicat encore de la pédagogie, le mode de présentation peut être un des aspects contraignants qui conditionne pour un opérateur humain la réalisation d'un algorithme aboutissant au sens de Landa (1969).

Chapiro (1967) donne une description précise d'un algorithme :

La suite des étapes élémentaires dans un algorithme est composée de :

1° Les *opérateurs*, qui représentent les *actions élémentaires* de remaniement des informations (par exemple, exécuter tel calcul, lire telle information) ;

2° Les *conditions logiques*, qui caractérisent la situation au moment du choix de l'un ou de plusieurs opérateurs possibles (par exemple, ce sont toutes les informations du type si... alors).

On retrouve dans les langages machines permettant d'écrire des programmes qui sont par définition algorithmiques une structure équivalente : des instructions du type *Go to*, *Stop*, *Read*, et des instructions conditionnelles du type IF permettant de passer à telle ou telle opération suivante. La description donnée par Chapiro permet de comprendre comment on passe à des modes de présentation des algorithmes assez techniques comme la notation des logiciens et cybernéticiens soviétiques (Liapunov, Chestopal) que l'on trouve utilisée dans les articles de Landa, Chapiro, Pouchkine, Klix, Kelbert...

Les conditions logiques sont représentées par des lettres minuscules : p, q, r, s..., elles peuvent prendre plusieurs valeurs suivant qu'elles sont vérifiées ou non.

Les opérateurs sont représentés par des lettres majuscules A, B, C...

La notation se lit de gauche à droite, après chaque condition logique se trouve une flèche verticale dirigée vers le haut, indexée d'un chiffre ; si la condition est vérifiée, on continue la lecture vers l'opérateur ou la condition logique suivante ; si elle n'est pas vérifiée, cette flèche renvoie à une flèche dirigée vers le bas ↓ portant le même index.

Par exemple, si l'objet U est défini par des caractères a, b, c suivant le schéma suivant $U \longleftrightarrow a \,\&\, (b \vee c)$, c'est-à-dire on reconnaîtra qu'il s'agit de l'objet U, s'il présente le caractère a (et) le caractère b (ou) le caractère c.

Définissons les opérateurs :

Soit A l'action de vérifier que le caractère a est présent
— B — — b —
— C — — c —
U = Conclusions : il s'agit de l'objet U
W = Conclusions : il ne s'agit pas de l'objet U

Définissons les conditions logiques :

Soit p = 1 (ou vrai) si a est présent, 0 sinon
— q — b —
— r — c —

On peut alors écrire l'algorithme sous la forme Liapunov-Chestopal :

$$A\,p\overset{1}{\uparrow}B\,q\overset{2}{\uparrow}U\underset{2}{\downarrow}C\,r\overset{1}{\uparrow}U\underset{1}{\downarrow}W$$

En lisant de gauche à droite, on voit que si p n'est pas vérifié, on passe directement à W (dans une conjonction $(A \wedge O)$; il faut que les deux membres soient vérifiés, sinon on vérifie q ; s'il est vérifié, alors c'est l'objet U :

$$A\,p\overset{1}{\uparrow}B\,q\overset{2}{\uparrow}U$$

S'il n'est pas vérifié, alors on passe à l'opérateur $\downarrow_2$ C (une disjonction (V) est vraie si l'un de ses membres est vrai, ou les deux).

Si r = 1, alors, on lit à droite U ; sinon passe à $\downarrow_1$ W (dans une conjonction).

La notation de Liapunov-Chestopal semble d'un usage assez commode pour les différents auteurs cités. Cependant, dans un contexte pédagogique, cette notation nécessite l'apprentissage d'un symbolisme supplémentaire et peut cacher par son caractère conventionnel l'appréhension directe de l'articulation du processus algorithmique. Chapiro précise qu'il ne l'utilise que dans les grandes classes. Landa soulève aussi ce problème, il renonce à utiliser cette notation dans le cadre scolaire.

La plupart des auteurs aboutissent à un mode de représentation plus figuratif. Chapiro, par exemple, transcrit directement la notation de Liapunov-Chestopal sous forme d'un graphe.

Si l'on reprend l'exemple développé précédemment avec la formule de Liapunov-Chestopal, le graphe se présente ainsi (fig. 1) ou bien encore, sous forme de schémas utilisant des éléments électroniques (fig. 2) :

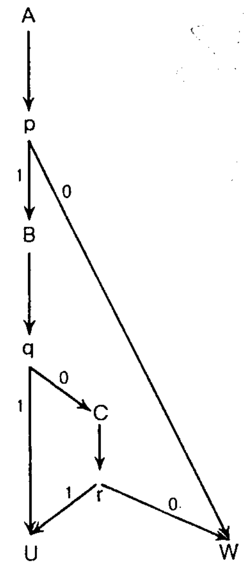

Fig. 1
Graphe représentant la formule (1)

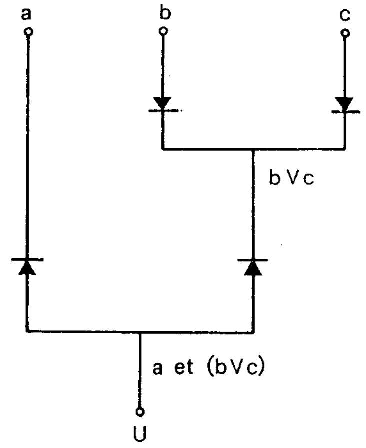

Fig. 2
Circuit électronique correspondant à l'identification de l'objet U

Landa, pour les petites classes, utilise une forme simplement verbale en esquissant la forme d'un arbre. Par exemple :

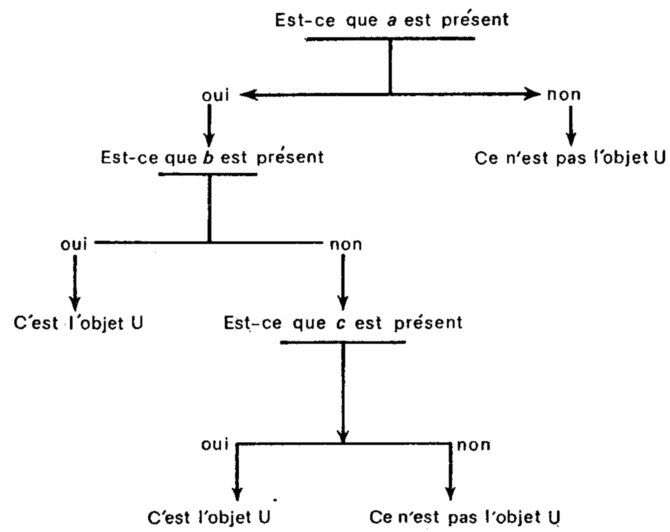

Fig. 3. — Arbre logique utilisé par Landa dans les petites classes

Les techniques de représentation sous forme de schémas sont très variées et couvertes par une littérature abondante. L'article de Gane (1967) en présente plusieurs illustrations, il utilise de nombreux exemples tirés de la psychologie du travail et empruntés en majeure partie au projet anglais de Wason et Jones (1965).

Pour terminer avec ce point, nous voudrions soulever un problème de vocabulaire et de définition. Certains auteurs comme Kelbert (1964) utilisent des organigrammes en précisant explicitement qu'il s'agit d'un mode de représentation des algorithmes. Le mot *organigramme* a fait fortune ces dernières années au point d'être extrêmement *lâche* dans sa définition ; on le trouve le plus étroitement lié à l'écriture des programmes machines. Dans ces cas, il est synonyme de modes de représentations graphiques particuliers d'un algorithme.

Cependant, certains auteurs emploient la notion d'organigramme sans utiliser (dans la terminologie) la notion d'algorithme. Par exemple, Bisseret, Leplat et Enard se réfèrent à la définition proposée par Rosensthiel : « Ce type de schéma est une image topologique de l'organisation des opérations... l'organigramme logique dont nous parlons ici diffère particulièrement des organigrammes et sociogrammes habituels dont le rôle est de mettre en évidence des relations intemporelles ; cet effort de géométrisation vise, en définitive, à mettre en évidence le support logique invariant sur lequel se déroulent les opérations séquentielles. » Pour Rosensthiel un organigramme logique est un « graphe, connexe, orienté, comportant une entrée et au moins une sortie ». Pris dans ce sens, on peut considérer que le terme anglais de *Flow-chart* (Gane, 1967), diagramme continu (Pohl) sont équivalents.

Mais alors un organigramme, au sens défini par Rosensthiel, décrit nécessairement un algorithme.

D'où plusieurs conséquences :

— Cette définition apporte une justification à la prise en compte de travaux ne comportant pas explicitement le mot algorithme, alors que la démarche est celle que nous avons décrite. Par exemple, certaines études de psychologie du travail (Leplat, Bisseret, Enard).

— Elle permet de distinguer clairement un processus algorithmique (la suite des actions réelles qui doivent être effectuées), l'algorithme (la description de ces actions), le mode de représentation de l'algorithme (notation logique, schéma, organigramme).

B) PLACE DES ALGORITHMES EN PSYCHOLOGIE

En essayant de cerner les différents problèmes de définition posés par l'introduction des algorithmes en psychologie et en pédagogie, dans la limite des textes que nous avons pu consulter, nous avons vu qu'une des propriétés essentielles de la théorie des algorithmes, la possibilité d'un calcul *a priori*, n'était plus vérifiée quand on passait aux consignes de type algorithme, que, de plus, l'on perdait aussi les propriétés des calculateurs, c'est-à-dire la possibilité de faire « tourner le programme à la main » ; et qu'enfin, certains auteurs sans s'embarrasser de la notion d'algorithme avaient développé des travaux expérimentaux fructueux, et notre intervention n'a fait apparemment qu'éclaircir le champ notionnel sur le problème de définition.

On peut alors, à juste titre, se demander quel est l'intérêt de l'introduction de la notion d'algorithme dans les sciences humaines.

1. Algorithme spontané

L'intérêt qu'un psychologue peut porter à un concept nouveau ne peut guère trouver de meilleure justification que de se fonder sur le sujet lui-même. Ce sont donc des arguments à partir de l'observation du *comportement spontané* du sujet qui seront d'abord développés.

Si l'on se place d'abord dans le monde scolaire, les auteurs comme Landa ou Gentilhomme montrent que l'enfant, spontanément, essaye de construire des systèmes de règles, de trouver des algorithmes pour résoudre des problèmes de ponctuation ou d'accord. Mais est-il besoin dans ce contexte de faire appel aux auteurs ? L'attitude chez les enfants d'essais de construction de recette ou de systématisation partielle de procédés de solution est d'observation courante pour les enseignants. On peut en conclure, comme le fait Landa, que si les enfants essaient de construire des algorithmes, ces derniers sont souvent faux ou incomplets aux yeux de l'observateur, et qu'il est justifié pour ces problèmes de construire des algorithmes complets et rationnels (1). Cependant, si la démarche du pédagogue qui consiste à remplacer le tâtonnement (déjà organisé) d'un enfant par une analyse et une construction systématique est justifiée, le statut exact de la démarche de l'enfant qui fonde cet argument est encore ambigu.

Le monde du travail présente bien des analogies avec les exemples déjà cités. On peut voir comme, par exemple, chez les contrôleurs de la navigation aérienne des « stratégies systématiques » sont élaborées spontanément et comment l'analyse *a priori* du psychologue retombe ensuite sur ces mêmes stratégies, démontrant leurs caractères algorithmiques (Leplat, Bisseret ou J. Pailhous (1968) sur les chauffeurs de taxis...).

Il est fréquent dans le monde du travail d'observer l'existence de telles constructions systématiques spontanées pour l'accomplissement d'une tâche. Si l'intérêt du psychologue se fonde encore une fois sur l'existence de comportements spontanés de type algorithmique, on garde jusqu'à un certain point l'ambiguïté du statut de ces comportements spontanés.

Cependant, ces exemples montrent la valeur d'une attitude méthodologique, qui oriente systématiquement l'analyse du travail vers la recherche d'éléments permettant de définir des comportements algorithmiques quand cela est possible et rationnel. Dans ce cadre, où la complexité des problèmes est très élevée, seule une analyse complète du travail permettra de construire un algo-

(1) Rationnel est pris dans le sens que lui donne Landa (1961, par exemple) : il peut être possible de construire un algorithme pour une classe de problèmes donnée, mais les caractéristiques de l'algorithme construit peuvent faire qu'il est beaucoup plus commode d'utiliser une solution non algorithmique. Landa donne comme exemple le cas de problèmes de géométrie où l'algorithme demandé, pour être suivi, l'identification de tous les traits pertinents de la figure, cet algorithme n'est pas rationnel parce que :

1° Il exige de passer par le démembrement exhaustif beaucoup trop long ; et

2° Souvent, dans les problèmes de géométrie, la principale difficulté est d'identifier les traits pertinents, et le problème est alors déjà résolu.

rithme ; il devient alors très difficile au sujet de prendre en compte toutes les données ; ce dernier procède plutôt par heuristique (1).

Vergnaud dans sa thèse *La réponse instrumentale* (2) *comme solution de problèmes* (1968) présente des expériences à support spatial (séries de verrous emboîtés, ou autres s'inspirant du taquin...) effectuées sur une population d'enfants de 3 à 9 ans. Ces expériences montrent que spontanément le sujet n'a pas un comportement par essais au hasard, mais qu'il a dans des tâches finalisées (et algorithmisables) une conduite sinon algorithmique, tout au moins réglée.

Précisons ce dernier point :

Le sujet n'aboutit pas toujours à l'algorithme que l'on peut définir, *a priori*, par l'analyse logique de la tâche à effectuer. Il peut aboutir à la construction de solutions qui ne couvrent pas tous les cas possibles, donc incomplètes pour l'observateur, mais qui, de son point de vue, sont algorithmiques (3). Ces solutions peuvent être décrites comme étant heuristiques, au sens où il manque l'effectivité.

Ces remarques peuvent faire mieux comprendre l'intérêt et les limites des réponses spontanées du sujet, si l'on veut y voir un argument pour fonder l'utilité du concept d'algorithme.

Vergnaud définit un *algorithme pour le sujet* comme étant une *conduite* :

— finalisée : elle vise explicitement la résolution d'une classe de problèmes déterminée ;
— effective : le sujet sait que la solution est certaine ;
— réglée, l'algorithme est complètement défini et le sujet peut le décrire sous forme de règle.

Il y a alors plusieurs statuts possibles de l'algorithme.

Une procédure de résolution peut être algorithmique pour un observateur ou pour une machine, mais pas pour le sujet qui l'effectue dans la mesure où on ne lui a pas enseigné avec la suite des opérations élémentaires constituant l'algorithme, le lien de nécessité entre la succession de ces opérations et le but qui est visé. Le sujet peut donc mener à bien une procédure algorithmique, mais cela ne sera pas pour lui une conduite algorithmique. C'est le cas — selon Vergnaud (1968) — avec les algorithmes appris.

Cet auteur établit alors une dichotomie entre algorithme appris et donc non motivé (4), et algorithme spontané et donc motivé. Le statut des algorithmes spontanés est donc différent, il s'agit d'une conduite motivée et fondée sur une analyse de la situation. « C'est là un point important de la signification psychologique des algorithmes, car si un algorithme spontané est toujours motivé, sa valeur n'est pas seulement descriptive mais théorique : l'existence d'un algorithme spontané implique chez le sujet une représentation cognitive

(1) Heuristique est pris ici au sens de procédure finalisée, réglée mais à laquelle il manque l'effectivité uniquement.

(2) Réponse instrumentale est définie de façon large comme toute réponse visant (et atteignant éventuellement) la solution d'un problème.

(3) Précisons qu'un « algorithme faux ou incomplet » est contradictoire dans les termes du point de vue mathématique, mais du point de vue du sujet, il a le sens d'une conduite finalisée, déterminée, et que le sujet *croit* effective.

(4) Motivé au sens où la nécessité n'est pas perçue.

de la situation », Vergnaud (1968, p. 20). L'importance théorique des implications que suppose l'existence d'algorithmes spontanés amène Vergnaud à dire : « Qu'il n'y a guère de sens à parler d'algorithmes pour d'autres situations que les situations nécessaires, c'est-à-dire les situations dans lesquelles toutes les relations et transformations en jeu sont accessibles au sujet... » (p. 123). Dans les situations où les relations et transformations ne sont pas observables par le sujet, Vergnaud préconise lorsqu'on peut mettre en évidence des conduites systématiques de parler de règle, de procédure, de stratégie (algorithmique).

Nous n'avons pas épuisé les différents aspects du travail de Vergnaud, mais il était important de montrer — à notre avis — que l'approche algorithmique en psychologie est fondée au niveau descriptif sur l'existence de conduites spontanées de type algorithmique.

Chez le sujet, nous avons essayé d'écarter les ambiguïtés de cette formulation ; de plus, au niveau théorique, l'approche algorithmique trouve des prolongements cognitifs dans la conception d'un modèle du sujet qui permet de rendre compte des conduites les plus élaborées.

Vergnaud défend d'ailleurs un modèle algorithmique du sujet dont les implications sont extrêmement riches, mais qui mérite un développement que nous n'avons pas la place de lui donner ici.

2. Les algorithmes et la psychologie soviétique

On peut justifier également l'intérêt des psychologues pour les algorithmes, en se situant dans un des cadres d'origine de la théorie des algorithmes appliquée aux sciences humaines, nous voulons parler de la psychologie soviétique. Pour cela, nous essaierons de justifier une typologie des algorithmes, à partir des thèses principales des théories psychologiques soviétiques.

Landa, dès ses premiers articles (par exemple Landa, 1961) a introduit à côté de la définition assez large des algorithmes une typologie sur laquelle il insiste beaucoup : il distingue deux catégories d'algorithmes, les algorithmes de *résolution*, et les algorithmes de *reconnaissance* ou d'*identification*.

Cette distinction peut de prime abord être considérée comme artificielle puisqu'elle semble se fonder uniquement sur une distinction de but, alors qu'il s'agit toujours d'algorithme et que seul le problème à résoudre est changé. En fait, de nombreux arguments justifient l'existence de cette typologie. Tout d'abord, rappelons que Landa doit être replacé dans un cadre pédagogique dont la finalité n'est pas uniquement la transmission à tout prix des seuls procédés de résolution, mais aussi la transmission de la compréhension de leur logique et des conditions de leurs utilisations. On peut opposer à cette attitude la mise au point d'un algorithme pour une machine où le seul but visé est d'obtenir la résolution dans des conditions optimales.

Le premier argument que développe Landa est que « nous avons établi des algorithmes de résolution, mais il n'a pas été répondu à la question de savoir par quel moyen nous avons appris que ce problème donné appartient justement au type donné, supposant cette question résolue. Dans la résolution de problèmes, la difficulté essentielle consiste non pas à réaliser les opérations prévues par l'algorithme, mais à déterminer par l'analyse le type dont relève le problème donné et de reconnaître les conditions qui permettent d'appliquer l'algorithme » (Landa, 1956). Cet argument est repris par de nombreux pédagogues soviétiques et allemands (cf. Talyzina, 1968); on trouvera, en français, à ce sujet, un article de Gentilhomme (1968) qui reprend ce problème en s'appuyant sur Landa.

Si l'application d'un algorithme suppose reconnues au préalable les conditions de son application, on se trouve devant un type de problème qui peut être résolu algorithmiquement selon Landa ; cet auteur donne des indications pour l'établissement d'algorithmes traitant ce type de problème.

C'est là la source d'un *deuxième argument* justifiant sa typologie. En effet, l'établissement d'un algorithme de résolution repose essentiellement sur la prise en compte de la logique propre du problème. Par contre, l'établissement d'un algorithme de reconnaissance « représente une tâche spéciale ». La structure de ses opérations ne dépend en aucune façon du contenu concret des objets sur lesquels ils s'appliquent, mais du schéma des relations qui lient entre eux les différents caractères qui définissent un objet (Landa, 1961). Le schéma des relations peut être présenté par une suite de caractères liés par les opérateurs logiques (conjonction ou disjonction) ; l'algorithme de reconnaissance est alors défini par cette suite.

Ces deux arguments ont une valeur indéniable; cependant, ils répondraient de façon limitée à l'objection que nous faisions sur la validité de la typologie définie par Landa s'ils n'étaient appuyés par un troisième argument plus psychologique.

La portée de cet argument pour être appréciée doit être resituée dans sa logique propre : la psychologie soviétique et plus particulièrement la théorie du reflet (développée par Leontiev, Vigotski, Rubinstein) et la théorie de formation des actions par étapes de Galperine, qui prend racine dans la théorie du reflet. Cet effort de remise en place dans le contexte de la psychologie soviétique n'est pas uniquement fondé sur l'existence d'un lien logique apparent, mais aussi sur l'existence dans les textes (Landa, Chapiro, Talysina) de références explicites et précises à ces différentes théories.

Notre objet n'est cependant pas de faire une présentation complète de ces aspects théoriques mais d'apporter les éléments d'informations nécessaires.

Dans ce cadre théorique, d'une façon générale, la vie psychique est conçue en tant qu'activité, « qui n'est compréhensible que d'après son rôle dans l'activité matérielle de l'individu » (Galperine, 1967). L'individu apparaît en tant que principe actif et non comme réceptacle de la vie psychique. Le psychisme n'est pas un tableau du monde mais une activité, c'est-à-dire un système d'action. Selon Talyzina, chaque action est composée d'unités plus petites : les opérations accomplies dans un certain ordre et selon une règle déterminée. Ce qui fonde la théorie du reflet, c'est qu'il y a une correspondance entre l'activité psychique et l'activité pratique de l'homme : l'activité psychique peut être définie comme l'activité pratique extérieure transformée « comme un processus de transition du reflété au reflet » (Leontiev). Pour Vigotski, les processus psychiques « se forment par enracinement de l'extérieur vers l'intérieur ». Dès lors, ce sont les actions extérieures, matérielles (ou matérialisées) qui sont primaires par rapport aux actions psychiques.

Landa (1966) s'exprime avec une grande clarté sur ce point : « Ainsi il est possible d'examiner les règles décrivant les propriétés objectives, les relations et les liaisons des objets de la même façon que la description des images correspondantes des objets et des actions produites avec eux, ainsi que des liaisons entre elles de ces images et de ces actions, c'est-à-dire comme des règles selon lesquelles doit se dérouler l'activité psychique de l'homme. En d'autres termes, on peut les examiner aussi comme des règles de l'activité psychique exigées. Ainsi si la lettre *a* dans une règle considérée comme la description des phénomènes grammaticaux représente un indice déterminé de l'objet grammatical, alors cette même lettre dans la règle considérée comme la description de la structure des processus intellectuels peut s'interpréter comme la désignation de l'image de cet indice ou de l'opération intellectuelle correspondante. La viabilité de cette double interprétation des règles découle du fait que les images de l'homme et ses actions doivent se trouver en correspondance avec les propriétés objectives et la structure des phénomènes qu'ils reflètent et transforment.

D'après ces premières propositions assez générales de la théorie du reflet, la description du monde matériel et des relations entre les objets est par là même une description des opérations psychiques. De plus, cette correspondance existe et peut exister parce qu'il y a reflet, c'est-à-dire qu'il y a un mécanisme de passage de l'extérieur vers l'intérieur.

Ces premières propositions permettent de rendre compte du rôle que peuvent jouer les algorithmes dans l'apprentissage. En effet, ce mécanisme de passage du reflété vers le reflet, de l'extérieur vers l'intérieur, tend à privilégier la notion du modèle, au sens de modèle susceptible de servir de base pour la formation du reflet. Selon Talyzina (1968), l'on ne peut « diriger la formation des images qu'au moyen des actions à l'aide desquelles elles se forment ». Définir ces opérations consiste donc à s'intéresser à la suite des étapes intermédiaires qui permet de résoudre un problème donné.

C'est ainsi que Landa présente son propre travail : « Notre travail, depuis dix ans, vise à élaborer des procédés qui permettent à la fois de découvrir les structures internes des mécanismes de la pensée, et de décrire celles-ci sous la forme d'algorithmes. Leur représentation sous forme de modèles servira précisément de fil directeur au cours de l'enseignement, ils devront fournir des exemples de ce qui devra être développé chez l'élève, tout en indiquant les structures et les modèles internes à former chez ce dernier » (Landa, 1966).

Nous avons justifié l'intérêt des algorithmes dans le contexte général de la théorie du reflet. Les étapes mêmes du passage du reflété au reflet ont fait l'objet d'une théorie de la formation par étape des concepts (Galperine, 1966). Si l'on schématise, on peut dire qu'il y a successivement une forme matérielle (forme de départ), l'objet de l'action est donné à l'élève sous la forme d'objets réels ou de modèles, schèmes ou dessins, puis la forme extérieure verbale, enfin la forme intellectuelle. Dans le cadre d'un apprentissage, un algorithme va donc représenter la forme de départ sous une forme matérialisée.

Cependant, l'action psychique a trois fonctions (Talyzina, 1968) : une fonction d'orientation, une fonction de réalisation, une fonction de contrôle. Mais la fonction d'orientation est privilégiée de plusieurs points de vue.

L'orientation de l'action est liée à la façon dont l'homme reflète l'ensemble des conditions objectives indispensables à l'exécution d'une action donnée. En préalable, à toute action, il faut avoir reconnu que c'est cette action-là qui s'applique. Galperine appelle cet ensemble de conditions la base d'orientation (BO). La BO conditionne toutes les qualités des actions futures (la psychologie soviétique distingue la forme de l'action, le degré de généralisation, le degré de développement, l'automatisation, etc.). Une fois admis le schéma de la théorie du reflet et le rôle privilégié des algorithmes comme première étape ou étape de l'action matérialisée, la distinction entre algorithme de reconnaissance et algorithme de résolution devient importante, car elle se fonde sur la distinction des différentes fonctions de l'action psychique (n'oublions pas, d'autre part, que précisément toute la vie psychique est conçue comme système d'action). D'autre part, en relation avec ce mécanisme de transition de l'extérieur vers l'intérieur, du reflété vers le reflet, la BO doit être apprise en même temps que l'action qui y correspond elle-même (un algorithme de résolution par exemple).

Talyzina distingue par exemple trois niveaux de BO :

— composition incomplète de la BO, l'action formée est très sensible aux changements de situation ;
— conditions nécessaires pour la réalisation mais sous forme concrète ;
— conditions sous forme généralisée.

La définition de ces trois niveaux indique la relation entre la BO et l'action formée, ce qui est d'un grand intérêt pour la pédagogie.

Tout cela peut rendre compte et de la distinction entre algorithme de reconnaissance et algorithme d'identification, et de la motivation pédagogique profonde à mettre cette BO sous forme d'algorithme quand cela est possible et rationnel. Regrettons cependant que cet ensemble théorique ne soit pas (à notre connaissance) soutenu par un ensemble d'études expérimentales convaincantes.

Nous avons voulu montrer comment le concept d'algorithme prenait place naturellement dans le contexte théorique de la psychologie soviétique sans préjuger de la valeur de ces théories. Deux aspects étroitement liés nous ont paru importants : la notion d'algorithme en tant que modèle pour la formation des opérations intellectuelles correspondantes, la distinction entre algorithme de reconnaissance et algorithme de résolution.

3. Les méthodes générales de pensée

Si la conception des algorithmes telle que la développe Landa s'articule assez bien avec les thèmes de la psychologie soviétique, cet auteur se resitue aussi dans le cadre de conceptions pédagogiques générales que l'on trouve également dans les pays anglo-saxons.

Ces conceptions peuvent se regrouper autour du titre *Méthodes générales de pensée.* Landa a publié, en 1959, un article intitulé : « De la formation chez les élèves d'une méthode générale d'activité intellectuelle leur permettant de résoudre les problèmes. » L'idée est qu'il existe des « méthodes générales » de pensée, entendons qu'il existe des règles qui peuvent servir de base pour résoudre les problèmes. Un auteur comme Polya, qui est d'ailleurs abondamment cité, a écrit *How to solve it* qui traite de ce sujet, mais il est loin d'être isolé et s'inscrit dans un véritable courant de pensée.

Un tel courant a encore d'ailleurs un statut plus philosophique que psychologique en France, mais en Union soviétique il semble avoir une influence non négligeable.

On comprend que les algorithmes s'inscrivent alors naturellement dans ce courant ; leur établissement représente des systématisations (pas encore un algorithme) de la recherche et de la construction d'une solution. A la limite, ils sont pour certains pédagogues le témoignage de la puissance du rationnel, nous voulons dire de la possibilité de résoudre n'importe quel problème par une démarche analytique simple.

Pour Landa, il ne s'agit pas seulement de faire apprendre aux enfants les algorithmes que le pédagogue a pu mettre au point pour différents cours (grammaire, mathématiques...), mais surtout de leur faire acquérir des « méthodes rationnelles de pensée » transférables à d'autres apprentissages (Landa, 1961). On doit alors apprendre simultanément aux enfants à construire ces algorithmes, c'est-à-dire analyser un problème en ses différents éléments et construire une procédure logique qui permette, quand cela est possible, de résoudre ce problème. Il devient clair qu'en ce sens les algorithmes apparaissent dans ce cadre pédagogique comme la base d'un apprentissage de « méthodes de pensée ».

Ces idées, quoique d'un grand intérêt, ont encore été peu exploitées dans un cadre expérimental — à notre connaissance —, dans cette mesure, elles ont encore une connotation assez philosophique.

Cependant, si nous nous tournons vers des motivations plus techniques (sinon épistémologiques) de l'introduction des algorithmes en psychologie, ce concept présente l'intérêt d'avoir un rôle coordinateur. Certaines recherches que nous avons citées plus haut ont pu étudier les procédures algorithmiques avant la lettre, mais la mise en place d'une définition claire de l'étude de ces procédure peut avoir valeur de méthode. En particulier, dans le domaine de la psychologie du travail ou encore dans la préparation d'un cours programmé, ce concept peut systématiser utilement l'approche des problèmes soulevés.

Et si nous sommes quand même loin des « méthodes générales de la pensée » que nous citions précédemment, ce concept a certainement une place comme instrument de formation à certains métiers de la psychologie.

CONCLUSION

Nous avons essayé de présenter et de définir les algorithmes et leur mode d'introduction dans les sciences humaines. S'il s'agit de la notion la plus comportementale des mathématiques, et la plus propre, semble-t-il, à l'usage des psychologues — un algorithme n'est-il pas « une suite d'actes à faire pour atteindre un certain but, et donc une conduite » (Vergnaud, 1968) — et si nous avons vu nombre d'aspects intéressants de cette notion, en particulier l'introduction d'implication théorique, qui permettent de sortir du niveau descriptif, il n'empêche qu'il n'y a là ni solution miracle, ni panacée universelle à l'usage des psychologues ou des pédagogues.

Les motivations existent à plusieurs niveaux pour faire usage de cette notion, mais sa mise en œuvre ne relève pas encore (loin de là) de la recette.

Il n'existe pas d'algorithme permettant de résoudre ce problème. Nous ne pouvons l'illustrer clairement dans cet article par le choix d'exemples d'applications dans différents domaines. Cet aspect-là fera l'objet d'un second article qui permettra de poser de façon plus précise les limites actuelles auxquelles se heurtent les travaux expérimentaux.

BIBLIOGRAPHIE

ALEXIEV, N. G. (1963). — Pravomesen algoritmitcheskii podkhod kanalizov prochessov oboutcheniia. (L'approche algorithmique est-elle juste ?) — *Voprosi psychologii*, **3**.

BELOPOLSKAYA, A. R. (1963). — Opit primenenia obouchaoutchikh algoritmov Expérimentation d'adaptation des algorithmes d'enseignement) — *Vestn. Vyss. Skoly*, **6**.

BIRUKOV, B. H. et LANDA, L. N. (1969). — *Analyses méthodologiques du concept d'algorithme en psychologie et en pédagogie, en liaison avec les problèmes d'enseignement*, trad. Unesco, 1970, à paraître.

CHAPIRO, J. H. (1967). — Ob algoritmizatsii processa forminovaniia poniatit (Algorithmisation du processus de formation des concepts) — *Voprosi Psychologii*, **2**.

EDWARDS, K. F. (1966). — Program learning in the Soviet block countries — in *Aspects of educational technology*, The proceeding of the Loughborough PI Conference, Avril 1966, Dereick Unwin ed., Methoen.

EDWARDS, K. S. (1967). — Algorithms and the teaching of grammar, *Audio-visual language journal*, **5**.

ENARD, C. (1966). — *Essai d'enseignement programmé appliqué à l'apprentissage d'éléments d'opérations aériennes*. Rapport C.E.R.P., CO 1060 R 11.

ENARD, C. (1968). — Relations entre l'analyse du contenu et l'analyse des opérations — *Travail Humain*, **31**, 1-2, 25-46.

GAGE, C. P. et MORABIN, I. S. et LEWIS, B. N. (1966). — *Algorithms for decision making in Aspects of Educational Technology*, The proceeding of the Loughborough PI Conference, avril 1966, Dereick Unwin ed., Methoen.

GALPERINE, P. I. (1966). — Sur la formation par étapes des actions et des concepts — in *Recherches psychologiques en U.R.S.S.*, Editions du Progrès, Moscou.

GALPERINE, P. I. et TALYZINA, N. F. (1961). — Formation of elementary geometrical concepts and their dependance on directed participation by pupils — *in* O'CONNOR, *Recent Soviet Psychologii*, Pergamon Press, Londres.

GANE, HORABIN and LEWIS (1967). — Algorithms for decision making in UNWIN, D. and LEEDHAM, J. (ed.), *Aspects of Educational technology*, Methuen and Co., 481-502.

GENTILHOMME, Y. (1969). — Optimisation des algorithmes d'enseignement — *La pédagogie cybernétique*, **VII**, 4, 13-31.

GROSS, M. et LENTIN, A. (1967). — *Notions sur les grammaires formelles*. Paris, Gauthiers-Villars.

GUNTHER, K. (1966). — Nature et fonction de la théorie des schémas modèles — *L'enseignement des langues étrangères*, **12**, 10-16 et 63-64.

IANOV, I. V. (1965). — The logical schemes of algorithms — in *Problems of Cybernetics*, I, 82-140, Pergamon Press.

JOSSE (1968). — Pédagogie cybernétique — in *Intersystèmes*, **13**.

LANDA, L. N. (1959). — De la formation chez les élèves d'une méthode générale d'activité intellectuelle leur permettant de résoudre les problèmes — *Vop. Psych.*, **3**.

LANDA, L. N. (1962). — L'enseignement aux élèves des écoles des méthodes de pensée rationnelle et le problème des algorithmes — *Vop. Psych.*

LANDA, L. N. (1962). — L'application du mode de pensée cybernétique à la théorie de l'apprentissage — *Vop. Filosofii*, **16**, 9, 75-87.

LANDA, L. N. (1963). — Méthodes mathématiques de construction et évaluation des algorithmes de reconnaissance, Communications 1 et 2, *Izvestia APN RSFSR*, fasc. 129.

LANDA, L. N. (1963). — *Simulation logique des processus de la pensée, comme méthode de recherche sur ces processus*. Travaux du II[e] Congrès de la Société Soviétique de Psychologie, fasc. 5, Moscou, Ed. APN RSFSR.

LANDA, L. N. (1966). — *Algorithmes dans l'enseignement*. Moscou, Editions du Progrès.

LANDA, L. N. (1966). — Recherches sur l'application de logique mathématique et de la théorie de l'information à quelques problèmes d'enseignement — *Vop. Psych.*, **2**.

LANDA, L. N. (1966). — *Diagnostic et enseignement programmé*. IV[e] Conférence Internationale sur l'Enseignement programmé et les machines à enseigner, Munich, Klett & Oldendourg.

LANDA, L. N. (1969). — Nekotorie outoch neniia povoprosov ob algorimakhy obouchenii. (Quelques précisions au sujet des algorithmes d'enseignement) — *Vop. Psych.*, **2**, 139-141, Réponse aux critiques formulées par Volovitch dans la même revue.

LANDA, L. N. et SURIKINUJ, L. N. (1969). — Développement de la pensée au moyen de l'enseignement programmé. Formation de processus algorithmiques et heuristiques. V[e] Conférence U.R.S.S. sur l'emploi de moyens techniques et sur l'enseignement programmé. *Symposium*, n° 2, Moscou, juin 1969.

LEONTIEV, A. N. (1966). — Le concept du reflet : son importance pour la psychologie scientifique — *Bull. de Psych.*, **254**, **XX**, 5, 236-242.

LEONTIEV, A. N. (1962). — Le problème du biologique et du social dans la mentalité de l'homme — *Bull. de Psych.*, **201**, XV, 7-8, 297-306.

LEPLAT, J. et BISSERET, A. (1965). — Analyse des processus de traitement de l'information chez le contrôleur de la navigation aérienne — *Bull. du C.E.R.P.*, XIV, 1-2, 51-67.

LIAPOUNOV, A. A. (1958). — *Questions générales de cybernétique, Problèmes de cybernétique* (en russe).

LIAPOUNOV, A. A. et CHESTOPAL (1957). — Description algorithmique des processus de contrôle — *Mathematitchekoe Provecheme*, **2**.

PAILHOUS, J. (1965). — Algorithme de déplacement chez les chauffeurs de taxis — in *Actes du XVI[e] Congrès de Psychologie appliquée*, Amsterdam, Swett & Zlinger, 553-557.

POHL, L. (1965-1966). — La signification des schémas modèles de pensée logique et son application à la rationalisation de l'enseignement des langues étrangères — *Cizi Jazyki ve skole*, **3**, 101-108 et **4**, 149-155.

ROSENSTIEHL, P. et GHOVILA-HOURI, A. (1960). — *Les choix économiques. Décisions séquentielles et simulation*, Paris, Dunod, 360 pages.

TALYZINA, M. F. (1968). — *Principes théoriques de l'enseignement programmé*. Moscou, éd. « Zmanje » (traduit par l'Unesco).

TRAHTENBROT, A. (1963). — *Algorithmes et machines à calculer*. Paris, Dunod.

VERGNAUD, G. (1968). — *Pour un modèle algorithmique du sujet*. Doc. ronéotypé.

VERGNAUD, G. (1968). — *La réponse instrumentale comme solution de problèmes : contribution*. Thèse de 3[e] cycle, Paris.

WASON, P. C. and JONES, S. S. (1963). — *The logical tree project. Mimeographic report*. London, Dpt of Psychology, University College.