Pensée informatique : approche didactique de l'identification de motifs

Marielle Léonard, Yann Secq, Yvan Peter, Cédric Fluckiger

Pensée informatique : approche didactique de l'identification de motifs

Marielle LÉONARD[12], Yann SECQ[1], Yvan PETER[1], Cédric FLUCKIGER[2]

[1] Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, France
[2] Univ. Lille, ULR 4354 - CIREL - Centre Interuniversitaire de Recherche en Éducation de Lille, F-59000 Lille, France
prenom.nom@univ-lille.fr

Abstract. Cet article concerne l'identification de motifs, capacité relevant de de la pensée informatique, considérée comme essentielle lors de la résolution de problèmes de programmation de type itératif. L'approche didactique adoptée consiste à affiner la définition du concept de motif dans ce contexte en distinguant motif visuel et motif algorithmique. Une analyse quantitative d'une centaine de problèmes posés lors du concours de programmation Algorea (200 000 participants) permet de caractériser et de catégoriser un certain nombre de difficultés relatives à cette activité d'identification de motifs.

Mots clés: pensée informatique, motif, identification de motifs, boucle, didactique de l'informatique, analyse quantitative, large échelle

1 Introduction

La réintroduction de l'informatique dans les programmes scolaires de nombreux pays a provoqué un regain d'intérêt pour les travaux en didactique dans cette discipline. En France, des contenus d'informatique sont présents dans les programmes scolaires de l'école obligatoire depuis 2016. Ainsi, la programmation informatique apparaît dans le programme de mathématiques de cycle 3[1] (classes de CM1, CM2 et 6ème, de 9 à 12 ans) et prescrit l'activité "*Programmer les déplacements d'un robot ou ceux d'un personnage sur un écran en utilisant un logiciel de programmation.*". Pour le cycle 4 (classes de 5ème, 4ème et 3ème, de 12 à 15 ans), l'attendu de fin de cycle mentionné dans le programme[2] est "*Écrire, mettre au point et exécuter un programme simple.*". Mais qu'entend-on par "*un programme simple*" ?

[1] Programme du cycle 3 en vigueur à la rentrée 2020, mathématiques, section espace et géométrie

[2] Programme du cycle 4 en vigueur à la rentrée 2020, mathématiques, thème E – Algorithmique et programmation

Pour ce cycle, la boucle figure parmi les connaissances visées avec la séquence d'instructions, les instructions conditionnelles et les variables informatiques. En première approche, on pourrait donc considérer qu'un programme comportant une seule boucle, sans imbrication, instructions conditionnelles ni gestion de variables, est un programme simple.

Lors d'études précédentes [11] [8], nous avons mis en place des scénarios pédagogiques pour explorer comment des élèves de l'école primaire appréhendent des problèmes de programmation qui présentent une structure itérative, c'est-à-dire dont la solution implique l'utilisation d'une boucle. Les résultats obtenus lors de ces études de cas nous ont amenés à considérer l'identification de motifs, c'est-à-dire de redondances, comme essentielle dans le traitement de ce type de problèmes. En particulier, nous avons repéré un palier de difficulté de manière récurrente : le passage d'une à plusieurs instructions dans le corps de la boucle.

Dans cet article, nous approfondissons l'étude de l'identification de motifs lors de la résolution de problèmes de programmation de type itératif. Nous nous questionnons sur les paramètres qui rendent la résolution de ce type de problèmes plus ou moins difficile. Nous nous demandons aussi si les difficultés identifiées évoluent avec l'âge des élèves. À cette fin, nous mobilisons des éléments de la théorie des champs conceptuels de Vergnaud [17] pour conduire une analyse à priori, puis nous menons une analyse statistique à large échelle à partir des taux de réussite à 101 problèmes de type itératif issus des éditions 2018 à 2021 du concours national de programmation Algorea organisé par l'association France-ioi. Nous concluons en proposant de nouvelles pistes d'études pour affiner notre compréhension de l'apprentissage des bases de la programmation informatique.

2 Identification de motifs

Le terme "*motif*" est un terme polysémique employé dans de nombreux contextes, dont plusieurs domaines artistiques : "*petit élément caractéristique d'une composition musicale, qui en assure l'unité*", "*dessin, ornement, le plus souvent répété, sur un support quelconque*".

En informatique, le terme "*motif*" est associé au mot anglais "*pattern*". D'une part, il est présent dans des travaux menés autour des "*design patterns*" dans le champ du génie logiciel [5]. D'autre part, "*looking for patterns*" [21], "*pattern recognition*" [7], "*identifying and making use of patterns*" [4] l'expression variant suivant les auteurs, désigne une habileté identifiée comme faisant partie de la pensée informatique. Certains travaux mentionnent plus spécifiquement la notion de boucle sur laquelle nous nous concentrons dans cet article. Dans le cadre de description des compétences liées à la pensée informatique défini par Gouws et al. [6] à partir d'une revue de littérature, une catégorie s'intitule "*Patterns and Algorithms*", catégorie dans laquelle la notion de boucle est prise en exemple. Rich et al. [12] définissent des trajectoires d'apprentissage, comprenant des objectifs et des exemples d'activités associées, dont l'une porte sur les structures itératives. Les auteurs mentionnent l'importance de la perception de la redondance car intimement liée à l'initiation à la notion de boucle, mais

aucune analyse de l'activité d'identification de motifs n'est proposée dans ces travaux. Pourtant, cette activité a déjà été repérée comme source de difficultés pour les élèves [1].

En revanche, en didactique des mathématiques, certains travaux abordent cette question d'identification de motifs. Ainsi, pour Collins & Laski [3], un motif est une séquence avec une régularité réplicable, qui peut varier selon une ou plusieurs dimensions. Liljedahl [9] propose de distinguer deux catégories de motifs : les *repeating patterns* et les *number/growing patterns* (Fig. 1). La première correspond à une structure cyclique générée par la répétition d'une unité discernable. Cette définition est reprise dans plusieurs travaux [10][18][20]. La seconde correspond à un motif paramétré par une ou plusieurs informations.

Fig. 1.Catégorisation de motifs en didactique des mathématiques

En nous inspirant des définitions précédentes, nous définissons un « motif » dans notre contexte comme **une entité repérable au sein d'un ensemble car répétée à l'identique ou avec des variations prédictibles**.

Liljedahl [9] répertorie différentes tâches en lien avec le concept de motif : copier une suite de motifs, continuer une suite de motifs, retrouver des éléments manquants dans une suite de motifs, transférer une suite de motifs d'une représentation vers une autre, identifier un motif. En s'appuyant sur des expérimentations menées auprès de jeunes enfants de 3 à 6 ans, Warren & al. [18] construisent une séquence pédagogique puis établissent une progression dans la difficulté de ces tâches [19][20]. Dans cette progression, l'identification de motifs est la tâche la plus difficile et elle est celle qui révèle la compréhension de la structure de la suite de motifs [19]. En effet, la stratégie de correspondance terme à terme, qui consiste à traiter les éléments du motif un par un sans considérer celui-ci dans sa globalité, est systématiquement mise en échec lors de cette activité [3].

Dans notre contexte, nous nous intéressons à l'activité d'identification de motifs dans le champ de la didactique de l'informatique, en rapport avec la confrontation à des problèmes de programmation de type itératif. Nous considérons que la distinction proposée par Liljedahl [9] est un début de caractérisation des formes de complexité d'abstraction des motifs, notamment la transition de motifs directement observables (visuels) vers des motifs non observables (changements d'état de l'environnement, voire similarité de processus dans le cadre de *design patterns*) et nous proposons d'étudier plus en détails les caractéristiques des motifs à identifier.

3 Concept de classe de situation

Pour caractériser et catégoriser les motifs à identifier lors de problèmes de programmation de type itératif, nous nous appuyons sur le concept de classe de situation développé par Vergnaud [16] au sein de la théorie des champs conceptuels. Celle-ci s'inscrit dans une approche constructiviste et cognitiviste de l'apprentissage. Elle vise à comprendre la conceptualisation, en particulier dans le cas des activités cognitives complexes, dont la programmation informatique fait partie. L'unité d'analyse est le couple sujet / situation au sens de tâche. L'hypothèse de Vergnaud est que toute action finalisée repose sur une "*conceptualisation-en-acte*", c'est-à-dire que les actions du sujet rendent compte d'une activité cognitive restant le plus souvent implicite, y compris pour le sujet lui-même. En didactique de l'informatique, la théorie des champs conceptuels a été mobilisée par Rogalski [13][14] pour étudier "l'alphabétisation informatique" au lycée et plus récemment par Spach [15] pour analyser des situations de robotique pédagogique. Dans notre contexte, nous nous plaçons dans ce cadre d'analyse pour étudier des situations où le but du sujet est de concevoir un programme informatique pour résoudre un problème de type itératif.

Vergnaud invite à analyser les situations auxquelles le sujet est confronté en les regroupant en *classes*. Cette catégorisation peut être envisagée du point de vue de l'expert, par une analyse des caractéristiques des situations, et du point de vue du sujet, en étudiant la manière dont celui-ci traite les situations auxquelles il est confronté. L'expert s'appuie sur l'identification de *variables de situation* visant à différencier des situations proches. Le changement de valeur d'une variable de situation peut affecter ou non la structure du traitement de la situation par le sujet. Si c'est le cas, cela permet de définir deux classes de situation distinctes.

Vergnaud insiste aussi sur la progressivité de la conceptualisation, qu'il convient de considérer sur un temps long. Dans une étude sur les structures additives, Vergnaud & Durand [17] ont demandé à 28 élèves de chaque niveau du CP au CM2 de résoudre des problèmes additifs dont la réponse est strictement la même numériquement, mais pour lesquels la formulation du problème induit un raisonnement différent. Ils ont ainsi identifié des classes de situation qui correspondent à des paliers de difficulté dans la résolution de ces problèmes additifs. Leurs résultats montrent également un effet de l'âge sur la capacité des élèves à résoudre ces problèmes.

Dans cet article, nous proposons d'affiner la définition du concept de motif et de caractériser certaines difficultés relatives à l'activité d'identification de motifs lors du traitement de situations de type itératif dans un contexte de programmation. Nous nous appuyons sur les travaux réalisés autour de ce concept de motif et nous mobilisons le concept de *classe de situation* pour catégoriser ces problèmes de type itératif. Nous nous inspirons aussi de l'étude de Vergnaud & Durand (1976) que nous avons transposée dans notre contexte. La section suivante détaille la méthodologie et le cadre expérimental qui a été mis en œuvre pour réaliser cette étude.

4 Méthodologie et cadre expérimental

Nous travaillons à partir de 101 situations de type itératif qui sont issues des éditions 2018 à 2021 du concours national de programmation Algorea. Toutes ces situations consistent à programmer des actions et des déplacements d'un robot virtuel sur une grille en langage *Scratch*. Leur point commun est que la séquence d'actions à faire exécuter au robot virtuel comporte de la redondance, qu'il est nécessaire d'identifier pour résoudre le problème. La solution de référence implique donc une boucle ou plusieurs boucles en séquence (mais elle ne nécessite pas de boucles imbriquées). Pour l'étude de ces situations, nous envisageons les deux points de vue indiqués par Vergnaud [16]. D'une part, nous réalisons une analyse du point de vue de l'expert, appelée aussi analyse à priori, de ces 101 situations. D'autre part, nous analysons l'activité des sujets confrontés à ces situations lors de leur participation au concours Algorea, à travers les taux de réussite relevés pour ces problèmes.

4.1 Analyse à priori : motif visuel et motif algorithmique

Lorsqu'une situation de programmation d'un robot virtuel sur une grille est de type itératif, deux motifs distincts sont à considérer lors du traitement de la situation. Parmi les concepts piliers de l'informatique [2], le premier motif est lié au concept de données. Dans notre contexte, il s'agit d'un **motif visuel**, qui est observable sur la grille. Il est constitué de cases adjacentes, contenant ou non un élément saillant visuellement (case marquée, ou contenant un objet). Le second motif est lié aux concepts d'algorithme et de machine. Il est constitué d'actions à faire exécuter les unes après les autres par la machine, actions qui sont induites à la fois à partir du motif repéré dans les données et à partir des spécificités de cette machine. C'est la répétition d'une suite d'actions dans le même ordre chronologique qui détermine ce second motif, que nous appelons **motif algorithmique**. Celui-ci n'est observable que lors de l'exécution effective des actions. Dans notre contexte, le motif algorithmique dépend du motif visuel et du système d'orientation du robot virtuel. Dans un programme conçu en langage *Scratch*, il correspond à la séquence de blocs dans le corps de la boucle.

Résoudre un problème de programmation de type itératif dans notre contexte requiert donc d'identifier le motif visuel sur la grille, de mettre en correspondance ce motif visuel avec les actions à faire réaliser au robot virtuel sur cette même grille, puis d'exprimer ce motif algorithmique avec le langage de programmation *Scratch*.

Pour chacun de ces motifs, visuel puis algorithmique, nous identifions plusieurs paramètres ou caractéristiques, qui correspondent à des *variables de situation* au sens de Vergnaud [16]. Pour le motif visuel, nous considérons le nombre de cases qu'il occupe sur la grille, la présence d'éléments saillants visuellement au sein des motifs visuels et la présence d'éléments de décor sur la grille. Pour le motif algorithmique, nous retenons le nombre d'actions constituant le motif et la présence d'actions ne faisant pas partie de la suite de motifs (correspondant aux instructions hors de la boucle). Comme variable de situation, nous étudions aussi le degré de correspondance entre le motif visuel et le motif algorithmique.

4.2 Cadre expérimental

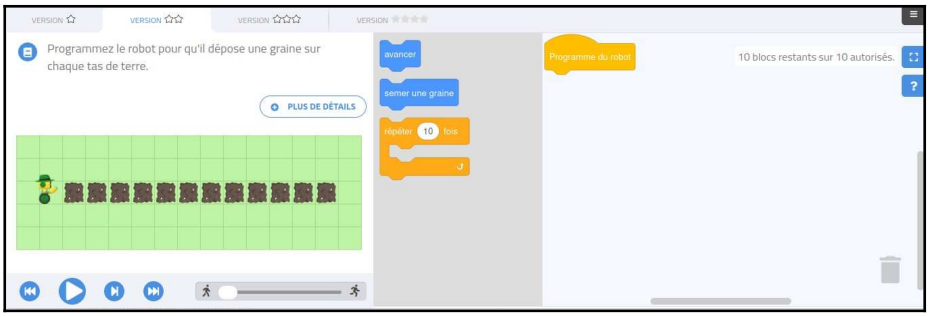

Fig. 2. Environnement de programmation du concours Algorea (situation 1, où la case délimite le motif visuel)

Les situations de déplacement de robot virtuel que nous étudions sont issues du concours de programmation en ligne Algorea, dont l'environnement de programmation est représenté sur la figure 2.

Cet environnement est adapté pour notre étude sur l'identification de motifs. D'une part, lors des problèmes de type itératif, le bloc *répéter* est le seul bloc de structure de contrôle disponible. Le sujet induit rapidement qu'il se trouve dans la situation où il a besoin d'utiliser ce bloc *répéter*. D'autre part, le nombre de blocs autorisé pour concevoir un programme est limité, ce qui contraint l'utilisation de ce bloc *répéter*. En revanche, le nombre d'essais n'est pas limité, ce qui permet le tâtonnement.

Suivant les tours (trois par an), le concours Algorea implique jusqu'à plus de 200 000 participants du CM1 à la terminale (de 9 à 18 ans). Dans le cadre de cette étude, nous nous intéressons seulement aux résultats individuels en langage *Scratch* pour les élèves du CM1 à la 3ème (de 9 à 15 ans), ce qui représente entre 6 000 et 75 000 participants suivant les tours, répartis sur les 6 niveaux étudiés, avec une surreprésentation des élèves de collège. Ainsi, nous ne maîtrisons pas la taille de l'échantillon étudié qui varie suivant les tours, mais reste conséquente. De plus, la situation est totalement écologique, la passation du concours ayant lieu en milieu scolaire ou à la maison. Toutefois, nous considérons que cette taille conséquente d'échantillon compense les variations de contexte de passation.

5 Résultats et analyse

Pour chaque situation, nous disposons du taux de réussite par niveau de classe. De manière préliminaire à l'étude sur l'identification de motifs, nous procédons à quelques analyses d'ordre plus général. Nous vérifions d'une part la robustesse de nos données concernant les taux de réussite. Lorsque l'on considère tous les niveaux de classe ensemble, un test d'indépendance de *khi2* nous permet de vérifier que tous les écarts de taux de réussite entre deux situations sont statistiquement significatifs avec

une *p-value* inférieure à 0,01. Pour un niveau de classe particulier, un écart de taux de réussite de 5 unités de pourcentage entre deux situations est significatif pour le collège (*p-value*<0,02). Seules quelques situations pour le niveau élémentaire dont l'effectif est plus réduit amènent à des écarts de taux de réussite de 5 unités de pourcentage moins robustes statistiquement.

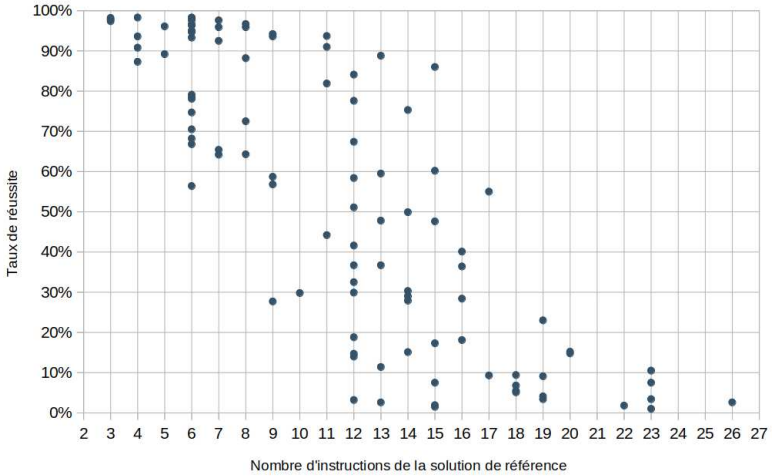

Fig. 3. Diagramme de dispersion du taux de réussite en fonction du nombre d'instructions de la solution de référence (taux de corrélation linéaire -0,81; p-value<0,05 au test de Bravais-Pearson)

D'autre part, la figure 3 confirme, que de manière attendue, le taux de réussite baisse lorsque le nombre d'instructions dans la solution de référence augmente. Cependant, nous remarquons une dispersion des valeurs importante sur l'axe vertical, parfois de plus de 50 unités de pourcentage, ce qui nous indique que d'autres variables de situation ont un effet sur le taux de réussite. L'identification et l'étude de ces variables sont l'objet des sections suivantes. À cette fin, pour chaque caractéristique relevée, nous calculons la médiane des taux de réussite et l'écart interquartile comme indicateurs de la distribution des données.

5.1 Motif visuel

Dans cette section, c'est l'aspect visuel du motif qui importe, indépendamment des actions que doit exécuter le robot virtuel.

Concernant le nombre de cases sur lesquels s'étend le motif visuel, nous pouvons distinguer deux classes de situation de manière très marquée (Fig. 4). Pour une première classe de situation, le motif visuel est constitué d'une seule case de la grille (exemple Fig. 2). Le taux de réussite de ces problèmes est élevé dès l'école élémentaire. L'écart interquartile est faible, ce qui signifie que cette caractéristique est prégnante dans l'explication du taux de réussite. En revanche, l'écart interquartile est beaucoup plus élevé si le motif visuel s'étend sur plusieurs cases (exemples Fig. 6).

Dans ce cas, d'autres variables contribuent significativement à la valeur du taux de réussite.

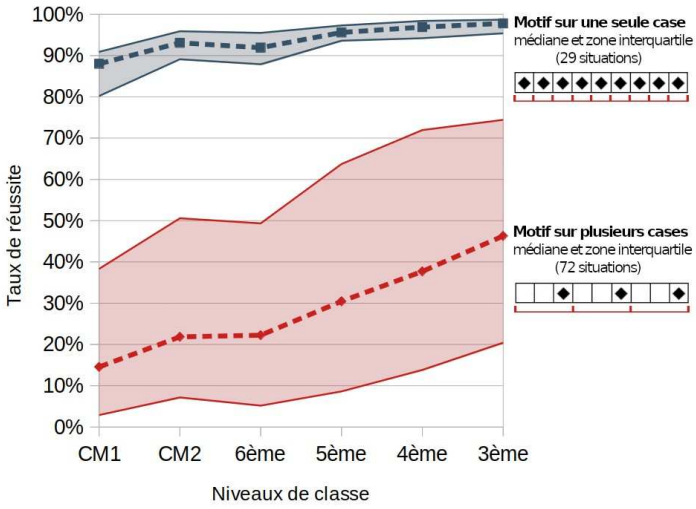

Fig. 4. Deux classes de situation : situations où la case délimite le motif, et situations pour lesquelles le motif s'étend sur plusieurs cases

Pour 70 situations pour lesquelles le motif visuel s'étend sur plusieurs cases, nous étudions les cases adjacentes qui ne font pas partie du même motif.

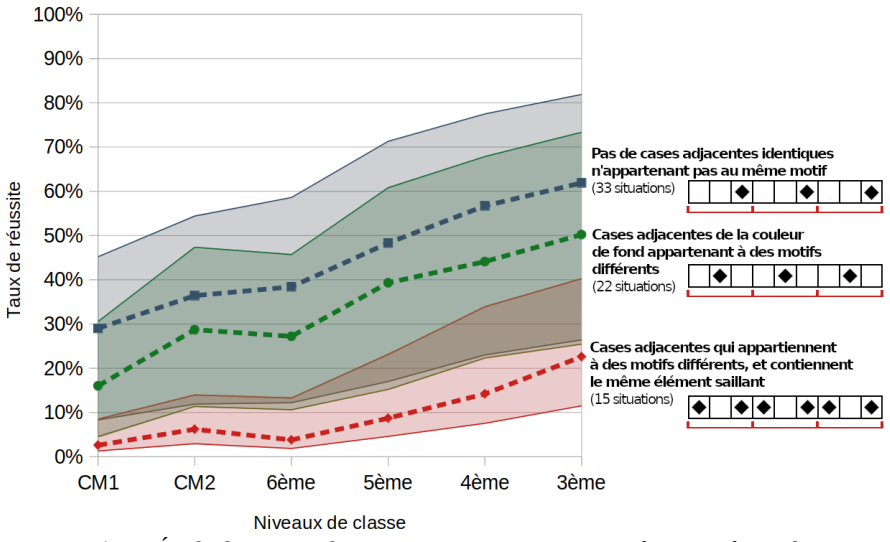

Fig. 5. Étude des cases adjacentes n'appartenant pas au même motif visuel

Lorsque des cases identiques adjacentes comportant un élément saillant visuellement n'appartiennent pas au même motif (exemples Fig. 6 : situations 3 et 4), le taux de réussite est bas (Fig. 5 : courbe rouge), et ce de manière plus marquée chez les plus

jeunes élèves. En revanche, lorsque ce sont deux cases adjacentes de la couleur du fond de la grille qui appartiennent à des motifs différents, le taux de réussite est proche de celui des situations sans cases adjacentes identiques appartenant à des motifs différents. Ce résultat nous amène à penser que les éléments saillants sont pris comme points de repère privilégiés lors de l'identification du motif visuel. Des éléments saillants identiques sur des cases adjacentes sont perçus comme faisant partie d'une même entité visuelle. Lorsque ceux-ci n'appartiennent pas au même motif, cela rend le motif moins visible et donc son identification plus difficile.

Nous montrons de la même manière l'effet de la présence d'éléments de décor sur la grille. Faute de place, nous ne donnons pour chaque modalité que la valeur de la médiane (Q_2) et de l'écart interquartile (EI) pour tous les niveaux de classe pris ensemble, l'unité étant le point de pourcentage du taux de réussite. Suivant la manière dont ils sont disposés, les éléments de décor sont plutôt une aide ou une source de difficulté. Lorsqu'ils contraignent complètement le parcours du robot (Q_2: 60.0, EI: 31.3), ils constituent une aide par rapport à des situations sans éléments de décor (Q_2: 51.1, EI: 0.58). Si ce n'est pas le cas, ils semblent agir comme distracteurs et constituent une source de difficultés (Q_2: 27.7, EI: 49.0). Cette difficulté devient massive lorsque ces éléments de décors rendent certains motifs visuellement différents (Q_2: 3.2, EI: 3.3).

Ainsi l'étude des caractéristiques des motifs visuels montre que la nature des éléments présents sur la grille a un effet sur la complexité de la situation. Plus le motif est facilement isolé visuellement, et plus la situation est bien résolue. À l'inverse, les facteurs qui perturbent la visibilité du motif impactent négativement le taux de réussite de la situation.

5.2 Correspondance entre motif visuel et motif algorithmique

Après avoir identifié le motif visuel sur la grille, il est nécessaire d'en déduire le motif algorithmique correspondant. Concernant la correspondance entre motif visuel et motif algorithmique, nous distinguons 5 classes de situation. Pour les trois premières classes, tous les motifs visuels sont identiques, ce qui n'est plus vrai pour les deux dernières classes.

Une première classe de situation, très distincte, et que nous avons déjà identifiée dans la section précédente concerne les situations où la case délimite le motif (exemple sur la figure 2). Les classes suivantes sont représentées figure 6. Nous regroupons dans une deuxième classe les situations pour lesquelles nous avons une correspondance stricte entre motif visuel et motif algorithmique. Chaque action de déplacement est repérable par la limite entre deux cases et les autres actions sont identifiables par un élément saillant visuellement. Ce sont les situations où le déplacement du robot n'est possible que dans une seule direction et les situations de déplacements en orientation absolue (nord, sud, est, ouest). Une troisième classe correspond aux situations où plusieurs états du robot virtuel sur une même case sont visuellement identiques, rendant partielle la correspondance entre motif visuel et mo-

tif algorithmique. Ce sont les situations en orientation relative pour lesquelles les actions de pivotement du robot ne sont pas observables avant l'exécution du programme. Il est nécessaire de simuler mentalement les actions de pivotement du robot, en se les représentant sur les cases adéquates et en maintenant l'orientation du robot en mémoire. La quatrième classe concerne les situations en orientation relative pour lesquelles la disposition des motifs est cyclique. Les motifs visuels ne sont donc plus identiques qu'à une rotation d'un quart de tour près. Enfin, pour les situations regroupées dans une cinquième classe, la correspondance entre motif visuel et motif algorithmique est entravée. Il est nécessaire de faire abstraction de certains éléments visuels. Soit des éléments saillants ou des éléments de décor sont équivalents mais visuellement différents, soit plusieurs motifs visuels sont partiellement superposés, perturbant la visibilité de chacun d'eux.

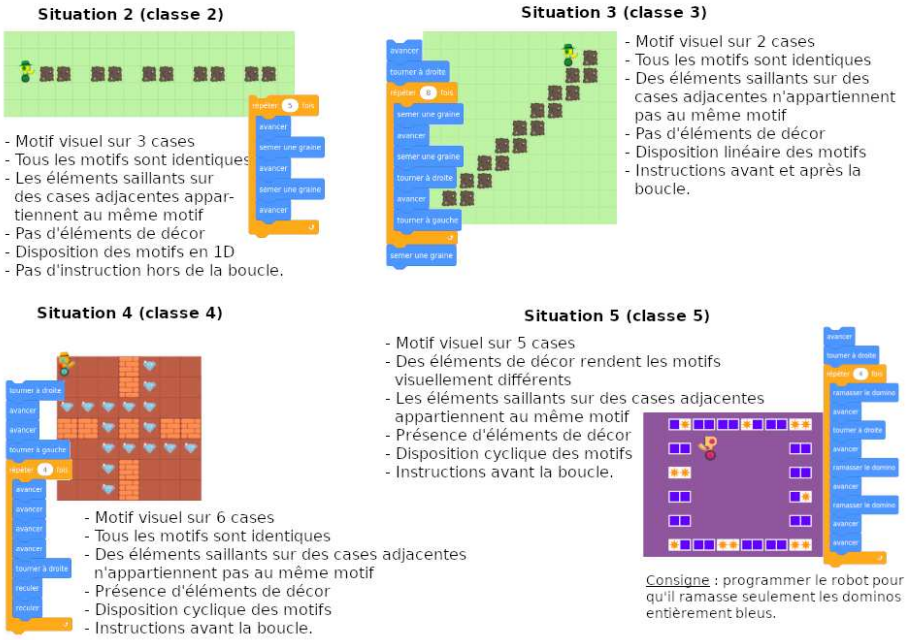

Fig. 6. Exemple type de situation pour chaque classe définie pour la correspondance entre motif visuel et motif algorithmique

Les 5 classes de situation définies précédemment correspondent à une gradation dans la difficulté à mettre en correspondance motif visuel et motif algorithmique (Fig. 7). Les situations de la classe 1, pour lesquelles la correspondance entre les deux motifs est attachée à la case, sont bien réussies par la majorité des élèves dès l'école élémentaire. En revanche, les situations de classe 5, qui nécessitent beaucoup plus de capacités d'abstraction, sont encore difficiles pour la plupart des élèves de fin de collège. Les zones interquartiles des classes 1 et 5 ne chevauchent pas celle des autres classes de situation. Nous en déduisons que le degré de correspondance entre motif visuel et motif algorithmique détermine fortement la difficulté de ces situations. En revanche, les classes 2, 3 et 4 ont des zones interquartiles partiellement superposées,

ce qui signifie que d'autres variables impactent aussi la difficulté de ces situations de manière significative. Ce sont aussi les classes de situation où l'on observe la plus forte progression au cours des 6 niveaux de classe étudiés.

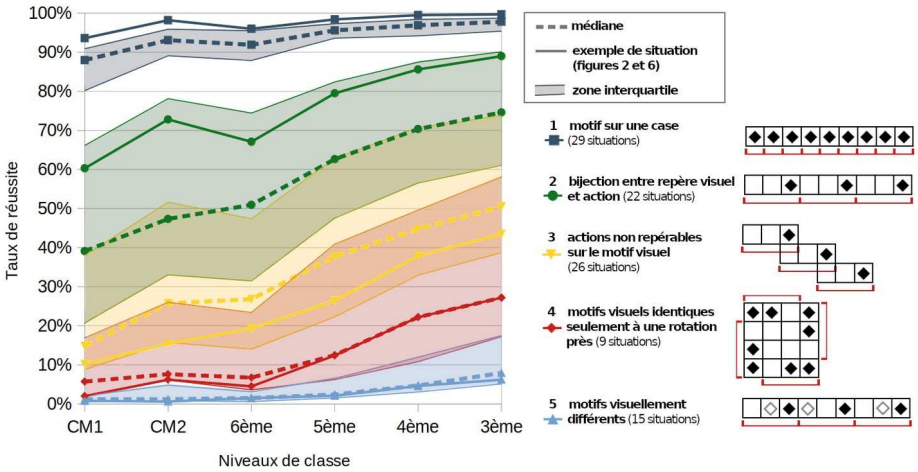

Fig. 7. Étude de la correspondance entre motif visuel et motif algorithmique

Concernant le motif algorithmique exprimé en langage *Scratch*, nous montrons en outre que le taux de réussite est corrélé au nombre d'instructions dans la boucle (taux de corrélation linéaire de -0,79) et que la situation est significativement moins bien résolue lorsqu'il est nécessaire de placer des instructions hors de la boucle, en particulier avant. Nous pensons que cette dernière difficulté est liée au repérage de la position du robot à considérer pour le début de la série de motifs visuels, position qu'il faut anticiper mentalement.

6 Discussion et perspectives

Nous avons montré dans cette étude qu'un problème de programmation de type itératif, même si la solution comporte seulement une boucle, n'est pas forcément un problème simple. Lors de la résolution de ce type de problème, l'identification d'un motif visuel et celle du motif algorithmique correspondant sont essentiels. Nous avons caractérisé des facteurs qui rendent difficile l'identification du motif visuel et nous avons établi une gradation dans les difficultés rencontrées, en particulier pour la mise en correspondance des motifs visuel et algorithmique. Parmi les difficultés identifiées, nous retrouvons celle, déjà repérée dans une étude précédente [8], liée à l'association de la situation de programmation avec de l'orientation dans l'espace.

D'autres travaux sont en cours pour approfondir cette étude. D'une part, peut-on considérer qu'un élève maîtrise la notion de boucle lorsqu'il a résolu les problèmes de programmation en tâtonnant, ce qui est possible dans ce contexte ? D'autre part, nous savons que l'identification de motif n'est pas seule en jeu dans le traitement des situations de type itératif. Une fois le motif identifié, il faut dénombrer les motifs, ce qui

peut induire d'autres difficultés qui restent à analyser. Pour affiner notre compréhension, nous avons besoin de données plus précises. C'est pourquoi, nous avons mis en place une collecte de traces d'activité à plusieurs échelles. Hormis les taux de réussite collectés à l'échelle nationale analysés dans cet article, nous disposons de traces d'activités à l'échelle de classes et d'enregistrements vidéo de participation au concours à l'échelle individuelle. Les traces d'activité à l'échelle des classes devraient nous permettre de distinguer les procédures de résolution expertes et les réussites par tâtonnement. Quant à l'analyse des enregistrements vidéos, nous cherchons à identifier des indicateurs qui traduisent le raisonnement, la *conceptualisation-en-acte* [16] du participant (procédure experte, erreurs). L'objectif sera ensuite d'apparier ces indicateurs avec les traces d'activité afin de passer à l'échelle, c'est-à-dire de faire le lien entre les trois échelles de collecte.

Remerciements

Ce travail est soutenu par le projet Interreg Teach Transition (https://teachtransition.eu). Nous tenons aussi à remercier l'association France-IOI pour la mise à disposition des taux de réussite au concours Algorea.

References

1. Baron, G. L., Drot-Delange, B.: L'informatique comme objet d'enseignement à l'école primaire française? Mise en perspective historique. Revue française de pédagogie. Recherches en éducation, (195), 51-62. (2016).
2. Berry, G. L' Hyperpuissance de l'informatique : Algorithmes, données, machines, réseaux. Odile Jacob. (2017).
3. Collins, M. A., & Laski, E. V.: Preschoolers' strategies for solving visual pattern tasks. Early Childhood Research Quarterly, 32, 204-214. (2015).
4. Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., Woollard, J.: Computational thinking-A guide for teachers. (2015).
5. Gamma, E., Johnson R., Vlissides J., Helm R.: Design Patterns: Elements of Reusable Object-Oriented Software. (1994).
6. Gouws, L. A., Bradshaw, K., Wentworth, P.: Computational thinking in educational activities : An evaluation of the educational game light-bot. In: Proceedings of the 18th ACM conference on Innovation and technology in computer science education, 10-15. (2013).
7. Hsu, T.-C., Chang, S.-C., Hung, Y.-T.: How to learn and how to teach computational thinking : Suggestions based on a review of the literature. In: Computers & Education, 126, 296-310. (2018).
8. Léonard, M., Peter, Y., Secq, Y.: Reconnaissance et synthèse de motifs redondants avec des élèves de 6-7 ans MOTIFS.MOTIFS.MOTIFS. ⟹ 3 x MOTIFS. 3 x MOTIFS. In : Colloque DIDAPRO 8 -DIDASTIC - L'informatique, objets d'enseignements enjeux épistémologiques, didactiques et de formation. (2020).
9. Liljedahl, P.: Repeating pattern or number pattern : The distinction is blurred. Focus on learning problems in mathematics, 26(3), 24-42. (2004).
10. Papic, M.: Promoting Repeating Patterns with Young Children—More Than Just Alternating Colours! Australian Primary Mathematics Classroom, 12(3), 8. (2007).

11. Peter, Y., Secq, Y., Léonard, M. : Reconnaissance de motifs redondants et répétitions : Introduction à la Pensée Informatique. STICEF (Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation), 27(2) (2020).
12. Rich, K. M., Strickland, C., Binkowski, T. A., Moran, C., Franklin, D.: K-8 Learning Trajectories Derived from Research Literature : Sequence, Repetition, Conditionals. Proceedings of the 2017 ACM Conference on International Computing Education Research, 182-190. (2017).
13. Rogalski, J.: Acquisition de savoirs et de savoir-faire en informatique. In: Cahiers de Didactique des Mathématiques, 43. (1987).
14. Rogalski, J., Vergnaud, G.: Didactique de l'informatique et acquisitions cognitives en programmation. Psychologie française, 32.4. (1987).
15. Spach, M.: Activités robotiques à l'école primaire et apprentissage de concepts informatiques: quelle place du scénario pédagogique? Les limites du co-apprentissage (Doctoral dissertation, Université Sorbonne Paris Cité). (2017).
16. Vergnaud, G.: La théorie des champs conceptuels. In Recherches en didactique des mathématiques: Vol. 10/2.3. La Pensée Sauvage. (1991).
17. Vergnaud, G., Durand, C.: Structures additives et complexité psychogénétique. Revue française de pédagogie, 28-43. (1976).
18. Warren, E., Cooper, T.: Using Repeating Patterns to Explore Functional Thinking. Australian Primary Mathematics Classroom, 11(1), 9. (2006).
19. Warren, E., Miller, J.: Exploring Four Year Old Indigenous Students' Ability to Pattern. International Research in Early Childhood Education, 1(2), 42-56. (2010).
20. Warren, E., Miller, J., Cooper, T.: Repeating patterns : Strategies to assist young students to generalise the mathematical structure. Australasian Journal of Early Childhood, 37(3), 111-120. (2012).
21. Wing, J. M.: Computational thinking. In: Communications of the ACM, 49(3), 33-35. (2006).