



HAL
open science

L'apport combiné de deux algorithmes d'IA à l'optimisation des parcours d'apprentissage dans le projet Adaptiv'Math

François Bouchet, Didier Roy

► To cite this version:

François Bouchet, Didier Roy. L'apport combiné de deux algorithmes d'IA à l'optimisation des parcours d'apprentissage dans le projet Adaptiv'Math. 47ème colloque de la COPIRELEM, Jun 2021, Grenoble, France. pp.855-868. hal-03625573

HAL Id: hal-03625573

<https://hal.science/hal-03625573>

Submitted on 31 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

APPORT COMBINÉ DE DEUX ALGORITHMES D'IA POUR L'OPTIMISATION DES PARCOURS D'APPRENTISSAGE DANS LE PROJET ADAPTIV'MATH

François BOUCHET

Maître de conférences, Sorbonne Université

LIP6, équipe MOCAH

francois.bouchet@lip6.fr

Didier ROY

Chercheur, Inria

équipe Flowers

didier.roy@inria.fr

Résumé

Les algorithmes employant des techniques d'intelligence artificielle se retrouvent de plus en plus dans des solutions à destination des élèves et enseignants, mais il s'agit toutefois généralement de briques isolées. L'objectif de ce travail est de montrer comment deux algorithmes peuvent avoir des apports complémentaires dans le cadre d'une solution d'apprentissage des mathématiques en cycle 2. D'une part, un algorithme utilisant l'apprentissage par renforcement (ZPDES) personnalise le parcours de chaque élève, en suivant en temps réel sa progression et en lui proposant les exercices qui lui sont les plus adaptés. D'autre part, un algorithme de regroupement de profils (SACCOM) définit des groupes homogènes d'élèves autour de critères donnés (réussite à un exercice, nature des erreurs, ...), facilitant la mise en place de pratiques pédagogiques différenciées et compensant la divergence des activités des élèves avec les parcours personnalisés.

Le projet Adaptiv'Math combine ces deux approches, proposant ainsi des phases d'apprentissage en autonomie et des interventions pédagogiques de l'enseignant en petits groupes.

I - INTRODUCTION

Le domaine des Systèmes Tuteurs Intelligents (STI) a vu son essor s'accroître ces dernières années, avec l'apparition des MOOCs, des jeux sérieux à objectif pédagogique, et l'utilisation de dispositifs numériques dans les écoles et les familles. Une question-clé en découle : comment personnaliser au mieux les parcours des utilisateurs afin qu'il soit proposé à chacun les exercices les plus adaptés et les plus motivants ?

Par ailleurs, si le système tuteur intelligent est destiné à être utilisé en classe, alors que l'on demande aux enseignants de mener des pratiques de pédagogie différenciée, un risque pourrait être que la personnalisation de parcours se fasse au détriment de la possibilité pour l'enseignant de savoir précisément ce que fait chacun. Une question-clé supplémentaire vient donc compléter la précédente : comment aider les enseignants à suivre l'activité de leurs élèves au sein du système tuteur intelligent de la meilleure façon possible en faisant émerger des regroupements de profils pour construire les stratégies pédagogiques les plus efficaces ?

Le projet Adaptiv'Math (<https://www.adaptivmath.fr>) s'inscrit dans cette problématique. Adaptiv'Math est une application à destination d'élèves du CP au CE2 visant à faire travailler différents domaines ou objectifs d'apprentissage (sens du nombre, résolution de problèmes...) regroupées au sein de cinq modules thématiques. Deux algorithmes d'Intelligence Artificielle (IA) travaillent conjointement à répondre aux deux questions précédemment posées : d'un côté, l'algorithme d'apprentissage adaptatif ZPDES choisit individuellement les exercices de chaque élève en fonction des compétences identifiées, de l'autre côté, l'algorithme de clustering SACCOM regroupe les élèves selon les exercices effectués et la

manière dont ils sont réalisés par les élèves afin de proposer à l'enseignant des regroupements d'élèves aux caractéristiques similaires. Dans cet article, nous nous proposons de décrire successivement le fonctionnement de chacun de ces deux algorithmes, puis de montrer leur complémentarité au sein de la solution Adaptiv'Math et les perspectives ouvertes par cette solution.

II - IA DANS LE PROJET ADAPTIV'MATH : L'APPROCHE ZPDES

1 Le contexte d'émergence de l'algorithme ZPDES

L'algorithme ZPDES (Zone of Proximal Development and Empirical Success) est issu des travaux de l'équipe Flowers d'Inria sur la modélisation des mécanismes d'apprentissage des humains (Clément, 2018). Les chercheurs de l'équipe conjuguent intelligence artificielle et différentes disciplines du développement (sciences cognitives, psychologie du développement, neurosciences, sciences de l'éducation) pour étudier les mécanismes d'apprentissage.

Le domaine de recherche des travaux de Flowers s'inscrit principalement dans le champ de l'intelligence artificielle développementale. L'intelligence artificielle développementale est un domaine de l'intelligence artificielle qui tire son inspiration de la psychologie développementale, et en particulier, de la façon dont les enfants apprennent en interagissant avec leur environnement. En robotique, l'intelligence artificielle développementale a de nombreuses applications et permet notamment d'envisager l'élaboration de robots qui sont capables de se construire leurs propres représentations du monde au travers de leurs interactions, et donc, d'être plus aptes à évoluer dans des environnements peu prédictibles.

Ainsi, l'équipe construit des algorithmes qui sont ensuite testés sur des machines afin de mieux comprendre comment les enfants humains apprennent, et cette recherche permet également de prototyper des machines ayant des capacités d'apprentissage.

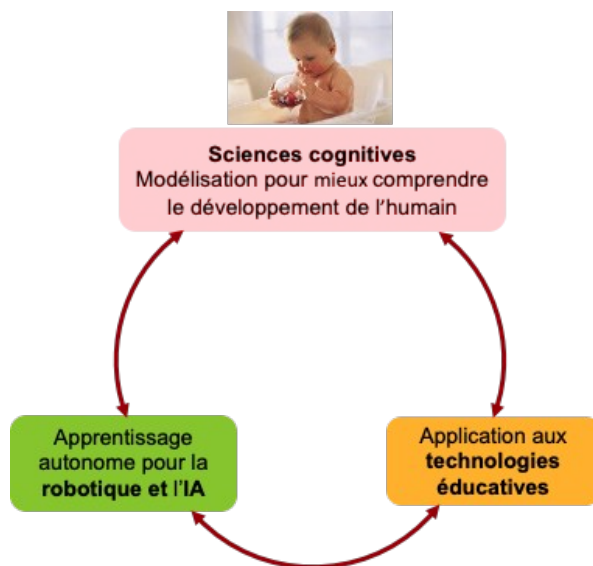


Figure 1. Champs de recherche de l'équipe Inria Flowers

Les algorithmes évoqués sont des algorithmes d'apprentissage automatique (machine learning), qui apprennent à partir de données. Dans le cas d'algorithmes du type de ZPDES, ils apprennent comment l'utilisateur peut progresser à partir de ce qu'il fait dans son parcours avec les exercices qui lui sont proposés. Ils permettent ainsi de déterminer le meilleur parcours d'apprentissage pour un apprenant humain à un instant donné, rendant possible une personnalisation des parcours.

2 La motivation intrinsèque, un moteur de l'apprentissage

Ces algorithmes exploitent des travaux fondamentaux de l'équipe Flowers sur la curiosité comme motivation intrinsèque fondamentale, moteur des apprentissages (humains, et par mimétisme, machines dans ce cas particulier). De même qu'un bébé explore son environnement et exploite les éléments de celui-ci pour acquérir de nouvelles connaissances, ces algorithmes utilisent une "exploration par curiosité" pour personnaliser des parcours d'apprentissage humain en maximisant l'efficacité des exercices proposés et la motivation des apprenants. Chez l'humain, la motivation intrinsèque est définie comme le fait de faire une tâche pour la satisfaction inhérente qu'elle procure plutôt que pour une conséquence distincte. Lorsqu'une personne est intrinsèquement motivée, elle est amenée à agir pour le plaisir ou le défi présenté plutôt que pour des produits, des pressions ou des récompenses externes (Ryan & Deci, 2000).

Dès la naissance, les enfants explorent spontanément leur environnement. Ils font preuve d'une volonté omniprésente d'apprendre et d'explorer sans avoir besoin d'incitations extérieures pour le faire. Cette tendance naturelle à la motivation est un élément essentiel du développement cognitif, social et physique (Blaye & Lemaire, 2007).

L'expérience d'un progrès en apprentissage dans une tâche donnée déclenche en retour une récompense intrinsèque, l'apprentissage en soi influençant donc de manière causale l'état de curiosité et la motivation intrinsèque. Ainsi, il existe une boucle de rétroaction fermée et auto-renforcée entre l'apprentissage et la motivation intrinsèque suscitée par la curiosité (Gottlieb et al., 2013). Ce mécanisme peut être utilisé dans des algorithmes d'apprentissage actif afin de sélectionner des tâches (Baranes & Oudeyer, 2013), et la maximisation des progrès de l'apprentissage empirique permet de générer automatiquement des séquences d'exercices, dont l'efficacité est optimale pour certaines catégories d'exercices, et très efficace en général (Lopes et al., 2012).

Oudeyer et al. (2016) ont montré que le progrès en apprentissage est une mesure pertinente de la qualité des exercices (activités proposées par le système). Ainsi, les algorithmes développés sont basés sur l'idée que déterminer les exercices les plus pertinents, c'est identifier ceux avec lesquels l'apprenant progresse. Or disposer d'une métrique permettant de classer les exercices à destination de l'élève du plus pertinent au moins pertinent à instant donné (en fonction de ce que l'on sait de ses connaissances et capacités à cet instant) est un pré-requis indispensable à un type d'algorithmes dits de « bandit manchot multi-bras ».

3 Les algorithmes de bandit manchot au coeur de ZPDES

Dans le but d'identifier les exercices les plus efficaces pour l'apprenant, les algorithmes étudiés se basent sur des méthodes d'inférence statistique appelés algorithmes de bandit manchot multi-bras (ou MAB pour « multi-armed bandit »). Ces algorithmes permettent de résoudre le problème suivant : un utilisateur, face à des machines à sous (bandits-manchots) possédant des espérances de gain différentes, les récompenses moyennes pour une même somme pariée, doit décider sur quelle machine il est le plus rentable de jouer. L'objectif est de maximiser le gain cumulé de l'utilisateur. Ainsi, en fonction des tirages successifs, l'algorithme va inférer quelle est la machine qui a la plus grande espérance de gain et l'utiliser plus que les autres (on parle « d'exploitation ») tout en continuant d'explorer les autres machines afin de vérifier qu'elles sont moins pertinentes. Pour trouver la meilleure machine, il faut donc dépenser de l'argent à explorer chacune d'entre elles avant de pouvoir toujours déterminer la meilleure. C'est un compromis exploration-exploitation très utilisé dans le domaine de l'apprentissage machine.

Dans le cadre des travaux de l'équipe Flowers avec des systèmes tuteurs intelligents, ce ne sont pas des machines à sous qui sont étudiées, mais des exercices. Le joueur est remplacé par l'algorithme, le choix de la machine est remplacé par un choix de paramètres d'exercices, et le gain en argent est remplacé par un gain en progrès d'apprentissage de l'élève. Ainsi, l'algorithme va choisir les exercices en fonction du progrès qu'elles permettent à l'apprenant de faire, tout en explorant un « graphe d'activités¹ » pour

¹ Nous considérerons dans le contexte de cet article exclusivement des exercices à réaliser par l'élève, mais du point de vue de l'algorithme, il peut tout aussi bien s'agir d'une autre activité (par exemple une vidéo à regarder – la réussite consistant alors à l'avoir vue entièrement).

trouver d'autres exercices pertinents. Une des particularités ici est que si l'espérance de gain d'une machine à sous reste la même tout au long du jeu, il n'en va pas de même pour l'espérance de gain en progrès d'apprentissage avec un exercice qui, elle, est non stationnaire. En effet, un exercice donné cessera de fournir un progrès d'apprentissage lorsque l'élève aura atteint un certain niveau de maîtrise de la compétence qu'il fait travailler. Inversement, un exercice précédemment trop difficile pour élève (ne lui permettant donc pas de progresser) peut, une fois les notions pré-requises maîtrisées, devenir accessible et fournir désormais un progrès d'apprentissage important pour lui. Cela nécessite par conséquent des mécanismes spécifiques pour suivre l'évolution de la récompense en progrès d'apprentissage.

4 Une banque d'exercices à explorer-exploiter avec ZPDES

En supposant un graphe d'activités pédagogiques préalablement défini pour un système tuteur intelligent (par exemple à la manière de (Térouanne et al., 2021)), des méthodes appropriées peuvent alors explorer et exploiter celui-ci pour proposer des exercices à l'apprenant. Ces exercices, issus d'une banque d'exercices, peuvent se présenter de différentes manières telles que des questions à choix multiples, des opérations abstraites à calculer au crayon, des jeux où les objets doivent être comptés par manipulation, etc. Ce formalisme permet de définir les différentes possibilités de sélection par l'algorithme, le défi consistant alors à trouver la séquence d'exercices qui va maximiser le progrès de l'élève et sa motivation.

La banque d'exercices pouvant être très grande, le problème à traiter par l'algorithme de bandit manchot multi-bras peut devenir très complexe et nécessiter beaucoup d'activités à faire par l'apprenant avant d'avoir une sélection optimale des exercices. Pour résoudre ce problème, deux théories sont mobilisées et combinées, celle de la Zone Proximale de Développement (ZPD)² (Vygotsky, 1978) et celle du Flow³ (Csikszentmihalyi 1975 ; Csikszentmihalyi, 1990). Ceci apporte des contraintes dans l'exploration du graphe afin de limiter le nombre d'exercices sélectionnables en même temps, et ainsi de guider l'algorithme de bandit manchot multi-bras dans son exploration (cf. Figure 2).

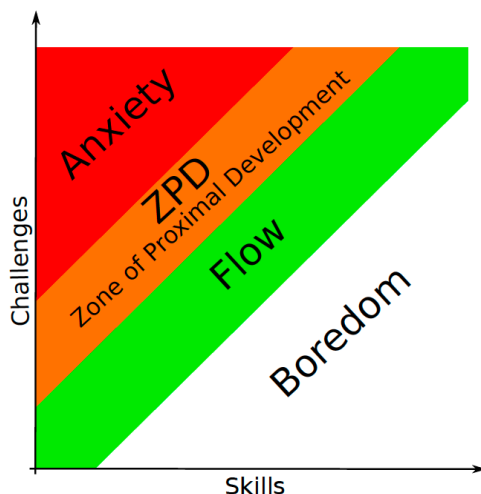


Figure 2. Entre ennui et anxiété, l'importance de choisir des exercices adaptés : ni trop faciles, ni trop difficiles

² Le concept de ZPD a été introduit par Vygotsky (1978) et représente l'ensemble des activités qu'un apprenant peut faire avec de l'aide et ne peut pas faire sans et suggère que cet ensemble d'activités présente une valeur pédagogique particulière.

³ Concept selon lequel, "au-delà de l'ennui et de l'anxiété", les gens sont intrinsèquement motivés par des activités qui présentent un défi optimal leur permettant d'entrer dans l'état de Flow. Il est ensuite (1991) décrit comme l'état mental dans lequel une personne est totalement immergée dans un sentiment de concentration énergique, d'implication totale et de plaisir dans le processus d'une activité conduisant à une expérience optimale. Pour entrer dans cet état, l'habileté de la personne doit correspondre au défi de l'activité, c'est-à-dire présenter un défi optimal, ni trop facile ni trop difficile.

La Figure 3 montre pour un élève donné comment évolue la zone d’exploration-exploitation dans laquelle opère ZPDES en fonction des résultats de cet élève. ZPDES et d’autres algorithmes du même type ont été expérimentés en situation pour en mesurer l’impact.

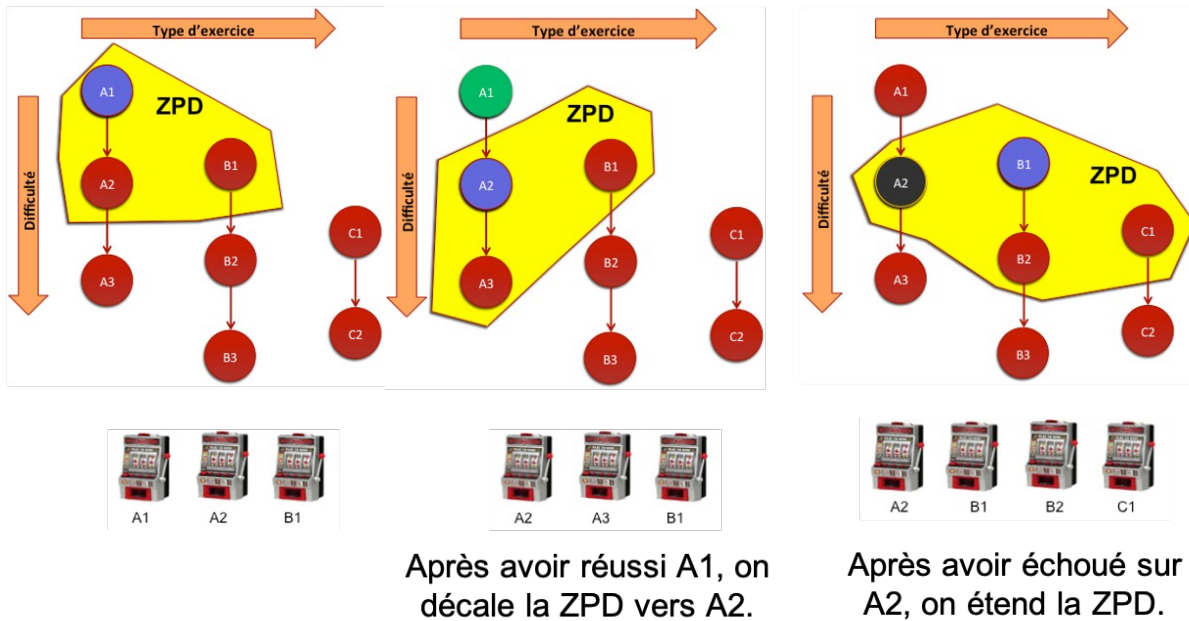


Figure 3. Un exemple d’exploration-exploitation de la banque d’exercices

5 Expérimentations en classe : le projet Kidlearn

L’équipe Flowers travaillant également sur la conception de méthodologies d’évaluation utilisant des outils de psychologie expérimentale pour mesurer les impacts des algorithmes, tels que ZPDES, en terme d’efficacité de l’apprentissage et de motivation des utilisateurs, des expérimentations ont été conduites, en partenariat avec le rectorat de l’académie de Bordeaux, au sein de plusieurs dizaines de classes de CE1, dans le cadre du projet Kidlearn⁴.

L’algorithme ZPDES a été dans un premier temps testé en simulation puis en situation réelle dans le projet Kidlearn auprès de plus de 1000 enfants de CE1 (Clément, 2018). Pour cela, une application sous forme d’un learning game sur l’utilisation de la monnaie a été développé, dans laquelle plusieurs types d’exercices sont proposés. L’élève peut être client ou marchand, avec un ou plusieurs objets, et doit composer des montants avec des pièces et des billets, avec des nombres entiers ou décimaux, les niveaux scolaires couverts s’étendant du CP au CM1 afin d’avoir une banque d’exercices suffisamment importante à explorer (cf. Figure 4) et couvrant des compétences mathématiques suffisamment larges.

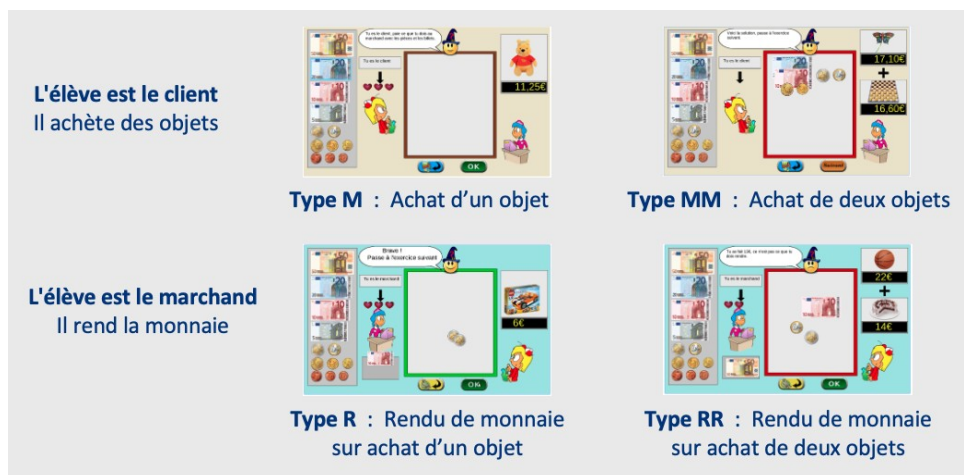


Figure 4. Les différents types de rôles et d’exercices dans Kidlearn (Clément, 2018)

⁴ <https://flowers.inria.fr/research/kidlearn/>

La stratégie de personnalisation menée avec ZPDES consiste à suivre en temps réel les progrès individuels d'apprentissage de chaque apprenant, en analysant ses résultats, pour lui proposer en permanence les exercices qui vont lui être les plus profitables et les plus motivants. Pour cela, ZPDES permet d'explorer des activités du graphe et d'en estimer l'impact sur l'apprentissage de l'élève concerné, puis d'exploiter celles estimées comme étant les plus efficaces.

6 L'algorithme ZPDES en action

Les figures suivantes montrent une comparaison des parcours d'exercices réalisés par les élèves selon qu'ils suivent un parcours pédagogique prédéfini par un enseignant-expert ou ce parcours réorganisé dynamiquement en temps réel par ZPDES pour l'adapter et le personnaliser pour chaque élève.

En ordonnées sont disposés les différents types d'exercices présents dans le jeu d'apprentissage du projet Kidlearn sur l'utilisation de la monnaie. Plus on monte dans les ordonnées, plus la difficulté est importante. Les petits rectangles verticaux représentent les couples élève-activité, en vert si l'exercice a été réussi, en violet si l'exercice n'est pas encore réussi, sans couleur si l'exercice n'a pas encore été réalisé.

On observe qu'avec ZPDES les parcours sont plus variés et amènent les élèves plus loin dans leurs apprentissages (cf. Figures 5 et 6).

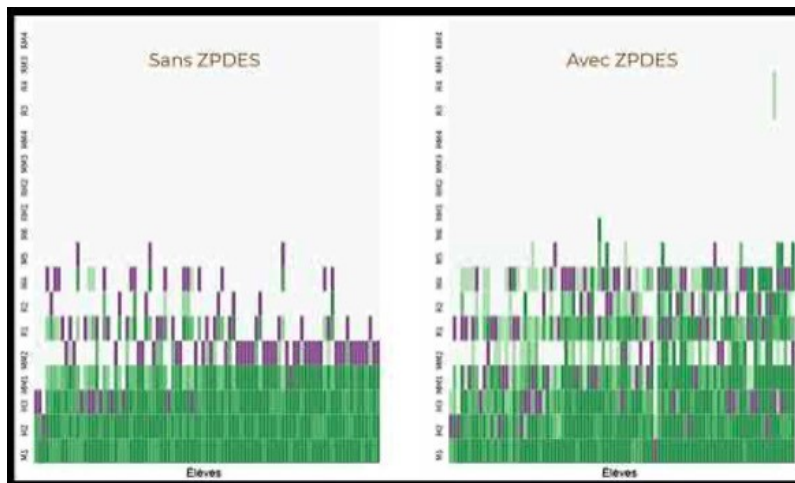


Figure 5. Parcours avec et sans ZPDES pour 100 élèves (en abscisse)

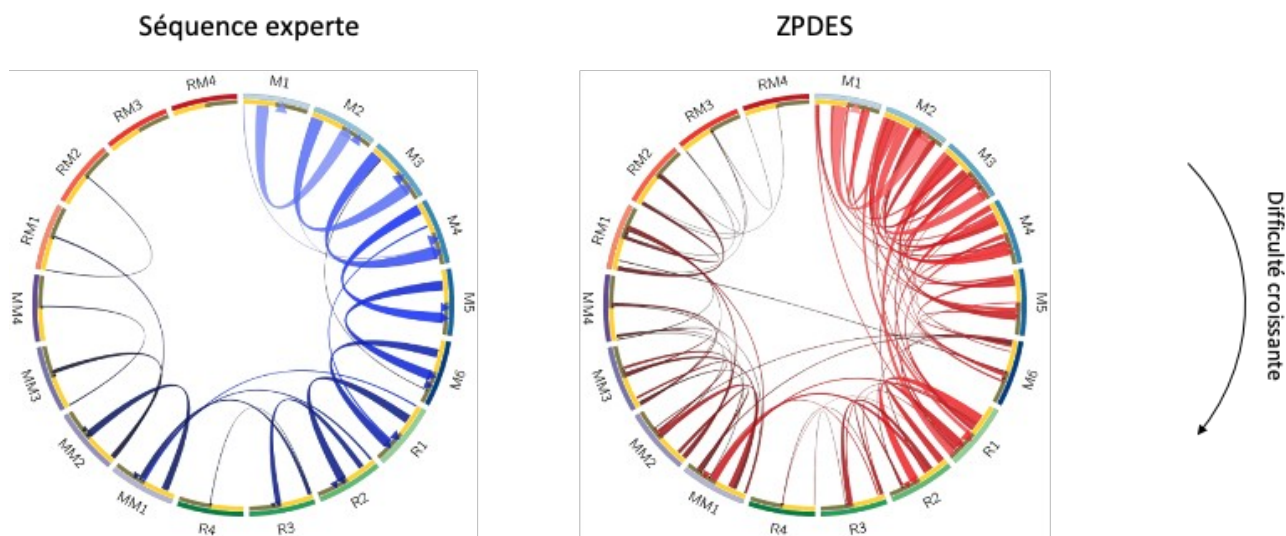


Figure 6. Une autre représentation de la comparaison avec ou sans ZPDES

Sur la Figure 6, l'épaisseur d'un tracé est proportionnelle au nombre d'élèves concernés par le changement de type d'exercice. Nous constatons là aussi que les parcours des élèves sont plus variés avec ZPDES, visibles avec les tracés traversant le disque, et que les niveaux atteints sont plus élevés.

Une thèse a été réalisée sur la conception et l'étude l'algorithme ZPDES (Clément 2018), décrivant l'algorithme lui-même en détails, et également les outils, mesures et questionnaires mis en œuvre pour analyser son impact sur les progrès et la motivation des apprenants. Les conclusions de ce travail de recherche indiquent que ZPDES est efficace pour l'apprentissage des élèves et permet de les maintenir motivés.

III - IA DANS LE PROJET ADAPTIV'MATH : L'ALGORITHME SACCOM

L'algorithme SACCOM (Student Activity Clustering for Classroom Orchestration & Monitoring) a été développé pour le projet Adaptiv'Math en vue de répondre à la deuxième question évoquée en introduction, à savoir comment faire des regroupements d'élèves permettant à l'enseignant de suivre de manière simple et compréhensible la diversité des parcours de ses élèves. Il repose sur des méthodes dites de **clustering** relativement classiques mais modifiées pour prendre en compte les nombreuses spécificités du contexte d'application. Il est développé par l'équipe MOCAH du LIP6 qui se spécialise dans la combinaison des approches d'IA dite numérique (fondée sur l'analyse de données) et symbolique (pour la représentation des connaissances et le raisonnement) en éducation, en s'attachant notamment à la compréhensibilité des décisions par l'utilisateur final (apprenant ou enseignant).

1 Principe général du clustering

Le clustering est une technique d'apprentissage automatique (ou machine learning) dite non supervisée. Dans un algorithme supervisé, on fournit des exemples de ce que l'on souhaite identifier automatiquement (par exemple des photos de chats), des contre-exemples (des photos sans chats) et moyennant un travail (non trivial) de bonne mise en forme des données, on laisse le soin à l'algorithme de déterminer ce qui distingue les éléments à classer dans le but d'identifier aussi bien que possible de futures instances (i.e. dire si dans une photo jamais vue il y a ou non un chat). Dans le cas de l'apprentissage non supervisé, on ne connaît pas a priori les catégorisations à faire. On fournit donc à l'algorithme un ensemble d'exemples mises en forme de manière aussi pertinente que possible, et on lui laisse le soin de les regrouper "au mieux". On peut ensuite analyser les regroupements produits pour leur donner une interprétation a posteriori.

2 Le clustering en contexte éducatif

Des recherches antérieures ont largement proposé des restitutions aux enseignants en constituant des groupes au sein d'une classe à partir des méthodes de clustering pour les aider à suivre la progression des élèves. Ainsi Fadljevic et al. (2020) proposent un système adaptatif visant à constituer des groupes de différents niveaux de compétences en lecture pour des élèves de collège. Les résultats du clustering (algorithme K-Means) ont montré des corrélations positives entre les niveaux de difficulté du texte et le temps de lecture pour les 4 groupes étudiés. Li et Yoo (2006) ont pour leur part proposé une chaîne de Markov basée sur du clustering pour modéliser les comportements en ligne des élèves pendant le processus d'apprentissage pour un enseignement plus efficace et adaptatif. D'autres approches utilisent des méthodes de clustering (algorithme K-Means ou clustering hiérarchique) pour regrouper les séquences d'interactions des élèves dans les MOOCs (Köck & Paramythis, 2011) ou les environnements hybrides (Akpınar et al., 2020). McBroom et al. (2020) s'appuient sur un nouvel algorithme de clustering hiérarchique (DETECT) pour identifier les tendances du comportement des élèves à partir des données temporelles. En particulier, il permet de capturer les comportements des élèves susceptibles d'abandonner sur certains types d'exercices, information qui pourrait nous intéresser par la suite pour analyser les trajectoires individuelles de chaque élève.

Au-delà de ces exemples, les techniques de clustering peuvent également être utilisées à des fins plus diverses dans un contexte éducatif. Ainsi Lopez et al. (2012) ont employé une méthode de clustering par

maximisation des espérances (EM) pour déterminer si la participation des élèves sur Moodle pourrait être un bon prédicteur des notes finales, et Käser et al. (2013) ont utilisé le résultat de l'application de l'algorithme de clustering K-Means pour prédire le pourcentage de réussite/échec des élèves. D'autres ont regroupé les élèves dans des catégories selon leur niveau d'engagement (Bouchet et al., 2013; Khalil et al., 2016) ou selon la nature des questions posées par ceux-ci (Harrak et al., 2020).

3 Du besoin d'adaptation du clustering

III.3.1 Un algorithme en deux temps

Traditionnellement, un algorithme de clustering fonctionne en un temps unique : à partir des données représentant N instances décrites selon M attributs qui lui sont présentées, l'algorithme détermine le meilleur partitionnement de celles-ci en un nombre K de groupes. La valeur de K peut être fixée par l'utilisateur ou déterminée automatiquement. En fonction de ce partitionnement, et du type d'algorithme de clustering utilisé, chaque instance existante peut ensuite être associée à un, plusieurs ou aucun des clusters appris. Dans le contexte d'Adaptiv'Math, N représenterait les élèves d'une classe (donc $N \in [15, 35]$) et est donc relativement faible, ce qui peut poser certains problèmes :

1. des comportements atypiques, mais correspondant à des problèmes réels bien que peu fréquents, risquent de ne se retrouver que chez un ou deux élèves dans la classe. Ils sont alors peu susceptibles d'être capturés par le clustering (soit ils seront filtrés au préalable dans une phase de détection des données aberrantes, soit l'algorithme de clustering risque de les associer, à tort, avec d'autres points distants) ;
2. inversement, certains élèves présentant des comportements atypiques mais pour des raisons extrinsèques (ex : problème avec une machine, interruptions répétées par un camarade ou par l'enseignant pendant qu'il fait des exercices) pourraient de manière artificielle constituer un cluster qui n'aurait que peu de sens d'un point de vue pédagogique. C'est d'autant plus vrai si ces problèmes ont lieu en début d'utilisation, où peu de traces ont été collectées au niveau de la classe.
3. ZPDES induit intrinsèquement une part d'aléatoire dans les exercices présentés aux élèves puisqu'il explore notamment différents objectifs ouverts à chaque élève pour déterminer son niveau sur ces exercices. Sur un petit effectif, il est donc normal d'avoir des exercices traités qui sont assez différents, ce qui complique la recherche de similarité dans les activités des élèves.

Le clustering traditionnel dont nous venons de lister les inconvénients dans notre contexte n'est donc pas idéal dans le cas d'Adaptiv'Math. Nous y référerons sous le terme de "**clustering local**" (puisque l'on apprend que sur les données, locales, d'une classe unique d'élèves). Par opposition à ce clustering local, nous proposons donc une approche en deux temps, inspirée des travaux de Harrak, Bouchet & Luengo (2019):

1. dans un premier temps, une **phase** dite **d'entraînement** permet de déterminer le meilleur partitionnement possible à partir des traces d'activité de l'ensemble des élèves ayant travaillé sur le système. Ce partitionnement permet de former plusieurs groupes ou clusters. Au sein de chaque cluster, il est possible de déterminer un centroïde, qui correspond à un élève prototypique représentatif des caractéristiques de ce cluster.
2. dans un second temps, une **phase** dite de **prédiction** permet d'appliquer le partitionnement précédemment appris pour déterminer à quel cluster appartient un nouvel élève, en fonction de la distance de ce nouvel élève à chacun des centroïdes des différents clusters.

Dans cette version adaptée du clustering, le nombre d'élèves considéré est alors bien supérieur, augmentant les chances d'obtenir des clusters réellement représentatifs de profils d'activités différents puisqu'ils se retrouvent au sein de différentes classes. Nous référerons par la suite à ce type d'approche dans lequel on apprend sur un ensemble de données avant d'appliquer ce partitionnement à de nouvelles données sous le terme de "**clustering global**".

Un des problèmes liés au clustering global est celui dit du “démarrage à froid”. Ce problème est courant dans l'apprentissage automatique : il est lié au fait qu'en absence de données initiales, un algorithme est parfois très peu performant. Ce problème est magnifié dans le cas du clustering global qui demande de disposer de données sur un grand nombre d'élèves avant de pouvoir prédire quoi que ce soit. SACCOM contourne ce problème lors du lancement d'un nouveau module en commençant par un mode “dégradé” utilisant le clustering local, puis lorsque suffisamment de traces sont disponibles, un entraînement est réalisé à partir de l'ensemble des traces avec ensuite un basculement vers le clustering global. Cet entraînement lui-même peut être réalisé à intervalles réguliers (par exemple tous les mois) pour y intégrer les dernières traces, et ce au moins jusqu'à ce qu'une année scolaire complète ait eu lieu pour avoir une vision aussi complète que possible des différentes activités des élèves.

III.3.2 Contraintes pédagogiques

Un algorithme de clustering standard peut proposer un nombre arbitraire de clusters. Certains nombres correspondent toutefois mieux à la structure sous-jacente des données, et on préfère donc utiliser certaines heuristiques (optimisation du BIC [critère d'information bayésien] ou “elbow method”) afin de sélectionner automatiquement ce nombre. Cependant, dans le contexte d'une aide au suivi et au pilotage de classe, il n'est pas souhaitable de se fonder uniquement sur ces heuristiques qui pourraient résulter en deux cas extrêmes :

- un partitionnement avec un unique cluster : si une classe est particulièrement homogène, il pourrait arriver que tous ses élèves se retrouvent dans un cluster unique. Il ne serait alors pas possible pour l'enseignant de distinguer des sous-groupes éventuels au sein de ce cluster unique.
- un partitionnement avec un nombre élevé de clusters (par exemple 10) : si une classe est particulièrement hétérogène, il pourrait arriver que les élèves se retrouvent dans de nombreux clusters, ce qui limite l'intérêt pratique des clusters qui est de réduire le nombre de cas à regarder pour l'enseignant.

Nous avons donc mis en place des heuristiques visant à rendre impossible ces cas extrêmes en garantissant un nombre de clusters systématiquement compris entre 2 et 5. En pratique, cela revient respectivement à scinder en deux les élèves d'un cluster unique, et à regrouper les élèves les plus proches de clusters distincts.

Même en fixant le nombre de clusters souhaité, il n'y a pas de garantie dans l'équilibre du nombre d'élèves associés à chacun d'entre eux. Il est donc possible d'avoir une classe de 25 élèves séparée en 3 clusters, avec 22 élèves d'un côté, 2 élèves dans un deuxième cluster, et 1 élève seul dans le dernier cluster. Pour l'enseignant, cela signifie traiter un groupe constitué quasiment de toute la classe (donc pas de petite taille), et inversement de faire du cas par cas (donc pas de travail en groupe), alors que la pédagogie différenciée préconise plutôt d'avoir recours à des petits groupes. Pour limiter ce phénomène, nous avons donc mis en place des contraintes dans la recherche de clusters qui visent à effectuer des regroupements de clusters comportant en priorité peu d'élèves pour constituer des clusters plus conséquents en taille, en respectant la limite minimale de 2 clusters par classe.

III.3.3 Choix des variables

Bien que non spécifique au contexte d'Adaptiv'Math, toute utilisation d'un algorithme de type clustering pose la question de la modélisation des données, autrement dit, l'identification des M attributs utilisés pour effectuer les regroupements. Plus ces données sont fines, plus les clusters constitués seront sémantiquement pertinents. Inversement, si par exemple on ne dispose que du temps passé sur chaque exercice, les clusters formés ne pourront représenter que les élèves rapides vs. les élèves lents.

Un autre point important, et moins courant, est la question de la restitution de l'information permettant à l'enseignant-utilisateur de comprendre la nature des clusters dans sa classe. En effet, même en limitant à un sous-ensemble des variables existantes celles utilisées pour la constitution des clusters, rien n'empêche d'utiliser l'ensemble pour leur interprétation. Ainsi sur la figure 7 à gauche, chaque instance est représentée par deux coordonnées (X et Y) correspondant donc à deux variables différentes. Bien qu'il soit évidemment possible d'utiliser les dimensions X et Y pour constituer des clusters, il est

également possible de n'utiliser qu'une seule dimension (par exemple X) pour les former, ce qui donne les clusters rouges et bleus sur la figure 7 à droite. Quand il s'agit d'interpréter ce qui distingue les clusters rouges et bleus, il est alors possible de se référer aux variables qui ont été utilisées pour la constitution de ceux-ci (par exemple pour dire que le cluster bleu se caractérise par une valeur de X plus élevée) mais aussi aux variables non utilisées pour leur constitution (par exemple pour dire que le cluster bleu se caractérise aussi par une valeur de Y plus élevée).

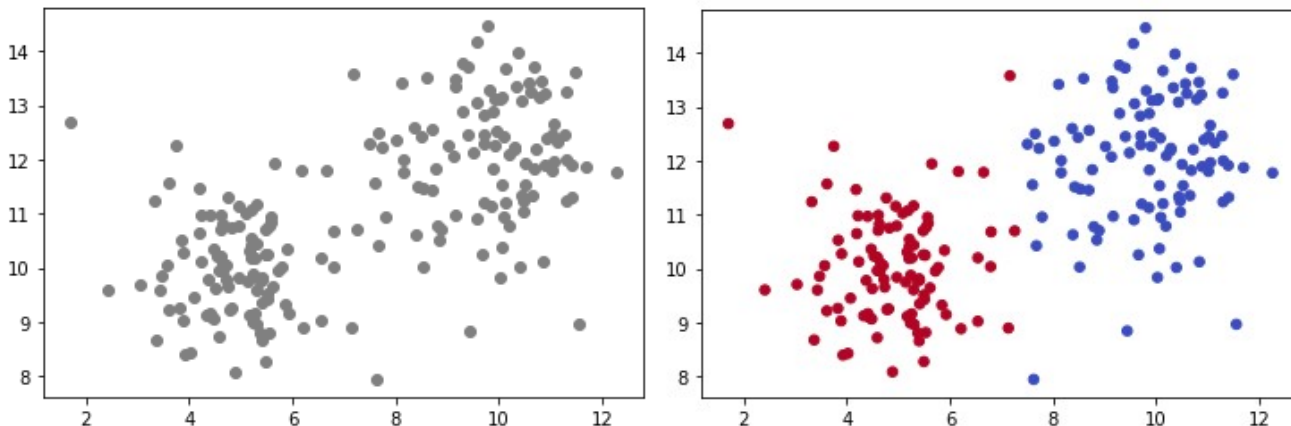


Figure 7. Clusters de points dans un espace en 2 dimensions (X, Y) mais dont la constitution ne dépend que de l'axe X

En pratique, dans le contexte d'Adaptiv'Math, parmi les attributs communs à tous les clusterings (quel que soit le module), pour chaque objectif (15 à 20 par module) et chaque niveau de difficulté (4 à 6 selon les objectifs), on calcule :

- le score moyen obtenu - qui permet de déterminer à quel point un niveau est déjà maîtrisé ou en cours d'acquisition ;
- la durée moyenne passée sur un exercice - qui donne une idée de la facilité avec laquelle un élève réalise un exercice (s'il est réussi) ou du degré d'effort impliqué dans sa réalisation. Associé au score, il peut donc donner une idée du degré de maîtrise d'un objectif ;
- le niveau de difficulté maximum atteint - qui donne une idée de la progression de l'élève sur cet objectif, puisqu'un niveau n ne se débloque que si un niveau $n-1$ est considéré comme maîtrisé par ZPDES (i.e. un certain score a été atteint sur les exercices de niveau $n-1$ dans cet objectif) ;
- le nombre d'exercices - qui donne une idée de la proportion de temps passé par l'élève à travailler ou valider un niveau de difficulté donné.

III.3.4 Fenêtre temporelle

Au fil du temps, chaque élève accumule des traces de ses différentes activités lorsqu'il travaille sur les exercices qui lui sont proposés automatiquement par ZPDES. Dans le contexte d'une utilisation sur une période étendue (plusieurs mois), qui est une hypothèse pertinente dans le cas d'Adaptiv'Math qui intègre plusieurs milliers d'exercices⁵ au sein de chaque module (même si chaque élève n'a pas vocation à tous les faire), une nouvelle contrainte relative au choix des traces est à considérer. En effet, lorsqu'on est au début du mois de juin, il est intuitivement peu pertinent d'accorder autant de poids aux exercices réalisés par un élève au mois de septembre qu'à ceux faits au mois de mai.

Pour résoudre ce problème, nous avons mis en place un système de fenêtre temporelle : lors de l'entraînement, on segmente les traces en période de T_j jours (par exemple 15 jours), ce qui a pour bénéfice d'augmenter également le nombre d'instances N disponibles pour l'entraînement (ex : là où un élève actif pendant deux mois n'aurait généré qu'une seule séquence complète d'activités, il génère

⁵ Chaque module est réalisé par une équipe d'auteurs distinctes qui génèrent et valident manuellement des variantes d'exercices standards initialement proposés

désormais quatre séquences de 15 jours). Pour la prédiction du cluster d'appartenance d'un élève donné, on ne considère alors que les traces d'activité produites au cours des T_j derniers jours. Il est alors nécessaire d'introduire également un nombre minimum d'exercices réalisés sur une période de T_j jours pour considérer l'élève actif sur cette période – dans le cas contraire, on ignore les quelques traces éventuellement produites.

IV - FONCTIONNEMENT CONJOINT DES ALGORITHMES DANS ADAPTIV'MATH

Pour comprendre en pratique comment les algorithmes fonctionnent de concert dans le contexte d'Adaptiv'Math, nous nous proposons dans cette section de décrire le déroulement d'une séquence d'utilisation du logiciel par un élève, tel qu'illustré dans la figure 8.

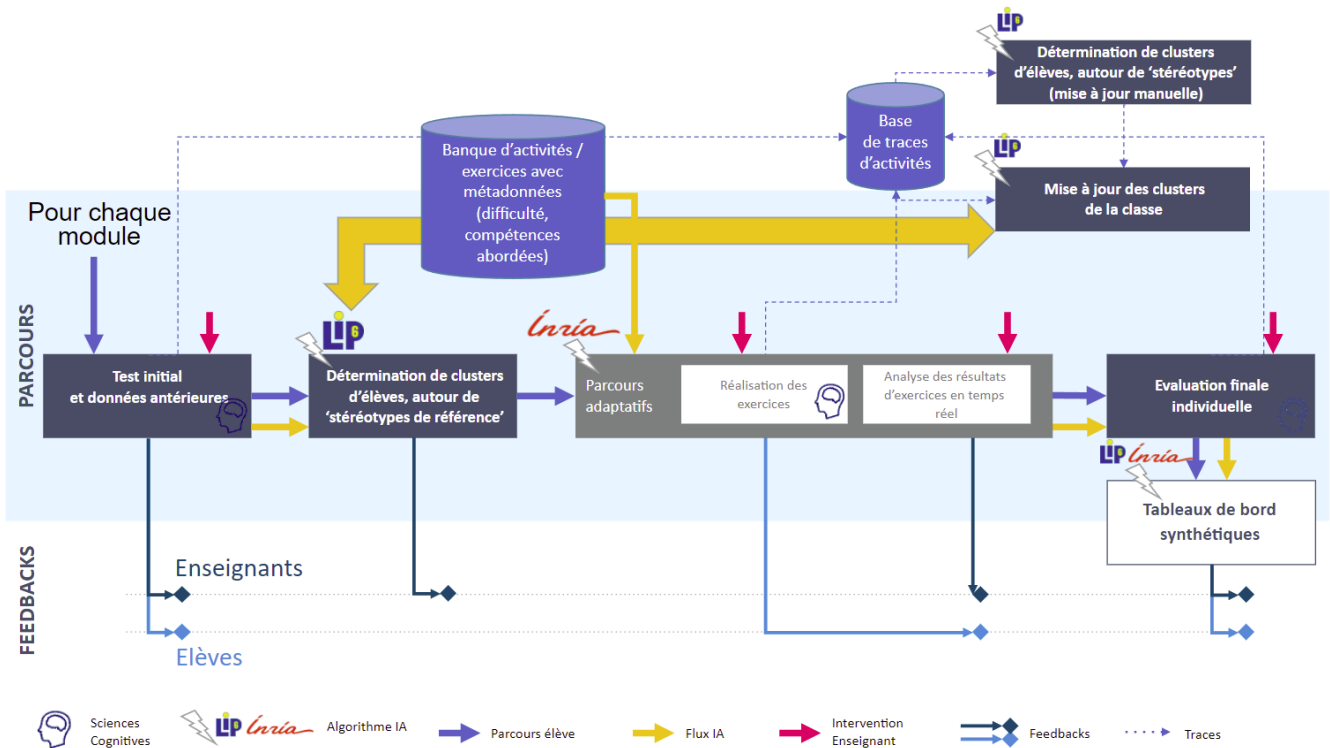


Figure 8. Les différentes étapes d'utilisation d'Adaptiv'Math illustrant l'intervention des deux algorithmes d'IA

Quand l'élève commence à utiliser l'application, il voit les modules qui ont été débloqués pour lui par l'enseignant parmi les cinq disponibles. Quand il commence un module, il doit passer un test initial qui utilise des exercices choisis pour estimer au mieux le niveau de l'élève parmi les exercices réalisés par les experts en sciences cognitives. Toutes les traces d'activité de l'élève (les bonnes/mauvaises réponses mais aussi les essais de glisser-déposer, les hésitations, les temps de réponse...) sont enregistrées dans la base de traces d'activité. Ce sont ces traces qui sont collectées à large échelle (en cas d'accord préalable de l'élève et de l'enseignant) et qui servent à déterminer des clusters comme vu dans la présentation de l'algorithme SACCOCOM. Une fois le test terminé, l'enseignant en est informé via le tableau de bord à partir duquel il peut piloter sa classe (cf. Figure 9). Lorsque tous les élèves de la classe ont réalisé ce test initial, il est possible de déterminer les clusters de la classe en communiquant avec le module d'IA SACCOCOM qui connaît les clusters IA de référence sur toutes les classes, et va renvoyer un partitionnement des élèves propre à la classe mais qui tire partie des connaissances accumulées sur l'ensemble des élèves ayant utilisé Adaptiv'Math. Les exercices de départ sont alors attribués à chaque élève en fonction du cluster auquel il est rattaché (au lieu d'avoir des exercices de départ potentiellement différents pour chaque élève), renforçant ainsi la cohérence des parcours dans la classe tout en gardant une individualisation des parcours par la suite. Chaque élève commence alors à réaliser des exercices de ce

module conçus par des experts en sciences cognitives, selon un parcours propre à chacun grâce à l’algorithme ZPDES. Ces résultats sont analysés en temps réel pour déterminer le meilleur exercice suivant pour l’élève, c’est-à-dire celui qui le fera progresser le plus vite. Il s’appuie donc pour cela sur une banque d’exercices finement indexée, et les traces de ce que font chaque élève dans ces exercices, comme dans le test initial, sont enregistrées. Ces informations individuelles sur ce que fait chaque élève sont bien sûr toujours visibles sur le tableau de bord de l’enseignant, qui peut voir les types d’exercices réalisés par chaque élève et le niveau de difficulté atteint pour chaque type d’exercice. Quand les élèves ont fait suffisamment d’exercices, un nouveau calcul des clusters de la classe peut être proposé à l’enseignant s’il ne le fait pas par lui-même. Ces clusters ou regroupements d’élèves sont bien sûr visualisables par l’enseignant, également sur le tableau de bord de pilotage de sa classe (cf. Figure 10). Finalement, à la fin d’un module, un test synthétique permet de bien vérifier ce que l’élève a pu apprendre. Son résultat est également affiché sur le tableau de bord et ses traces sont enregistrées.

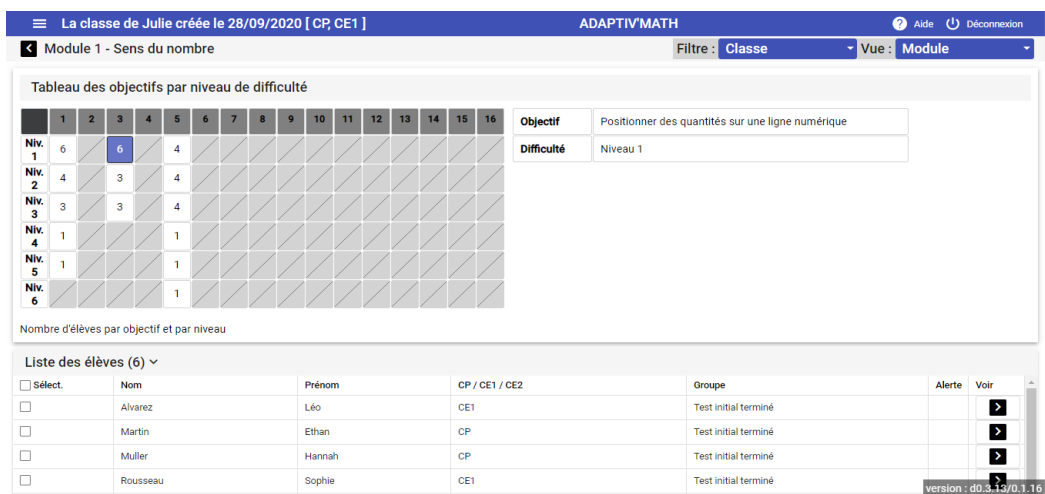


Figure 9. Tableau de bord permettant de voir la progression des élèves au sein des différents objectifs de ZPDES

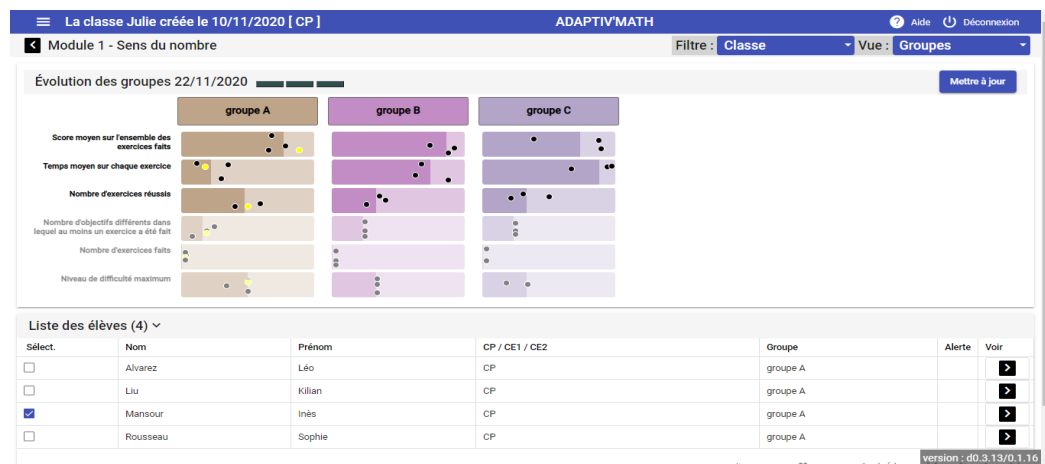


Figure 10. Tableau de bord de suivi des élèves au sein des clusters (3 clusters pour cette classe)

A tout moment l’enseignant garde le contrôle de la progression de sa classe et de chaque élève. Il peut ainsi donner ou non accès aux exercices d’un module à un élève, limiter l’accès à certains exercices dans le graphe d’activités utilisé par ZPDES (s’il veut par exemple que certaines notions ne soient pas encore abordées, même par des élèves en avance), visualiser les clusters et s’en servir pour donner des exercices de remédiation qui viennent s’insérer dans la séquence normalement déterminée par ZPDES. Cet usage de l’intelligence artificielle n’est donc bien sûr pas destiné, de quelque façon que ce soit, à faire remplacer l’enseignant par un dispositif numérique auto-adaptatif, ce qui serait à la fois absurde et inefficace, mais vise à mettre un outil supplémentaire à sa disposition.

V - CONCLUSION

Nous avons présenté ici deux algorithmes, SACCOM et ZPDES, fondés sur deux paradigmes différents d'IA (apprentissage non supervisé et apprentissage par renforcement) qui, une fois associés, offrent une expérience pour l'enseignant qui n'aurait pas été possible en considérant chaque algorithme de manière séparée. L'ambition est de permettre à l'enseignant de construire des temps de classe proposant des contenus personnalisés pour chaque élève, prenant en compte ses réussites et ses échecs. C'est une aide à la réussite de chaque élève, en difficulté scolaire ou non, d'autant plus intéressante que durant ce temps l'enseignant peut accompagner plus spécifiquement les élèves à besoins particuliers.

Début 2021, les deux algorithmes sont exploités au sein de l'application Adaptiv'Math en cours de test parmi un ensemble de 137 classes pilotes (Harrak & Bouchet, 2021), réparties sur plusieurs académies, mais aussi au sein d'Adaptiv'Langue⁶, solution pour les cours de grammaire et d'orthographe en lycée, puisque les algorithmes sont agnostiques par rapport au domaine considéré (il est uniquement nécessaire que les exercices soient indexés et tracés). Parmi les perspectives d'évolution, l'intégration des deux algorithmes pourrait être encore renforcée pour que les regroupements d'élèves ne soient pas utilisés par ZPDES uniquement au début d'un module mais également plus tard. Ce point est cependant complexe car les activités divergent de plus en plus au fil du temps et dans une classe hétérogène, le risque serait réel de retarder les plus rapides ou de forcer les élèves qui sont en cours d'acquisition d'une notion à avancer trop vite. Au niveau de l'algorithme SACCOM, l'exploitation d'attributs basés sur des séquences significatives (ex : échec sur plusieurs exercices successifs) au lieu d'événements individuels (ex : réussite sur un exercice) pourrait permettre d'améliorer la constitution ou au moins l'explication associée à chaque cluster.

VI - RÉFÉRENCES BIBLIOGRAPHIQUES

Akpinar, N.-J., Ramdas, A., & Acar, U. (2020). Analyzing Student Strategies in Blended Courses Using Clickstream Data. *Proc. of the 13th International Conference on Educational Data Mining*. EDM 2020, Ifrane, Morocco.

Baranes, A., & Oudeyer, P. Y. (2013). Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1), 49-73.

Bouchet, F., Harley, J. M., Trevors, G. J., & Azevedo, R. (2013). Clustering and Profiling Students According to their Interactions with an Intelligent Tutoring System Fostering Self-Regulated Learning. *Journal of Educational Data Mining*, 5(1), 104-146.

Clément, B. (2018). *Adaptive Personalization of Pedagogical Sequences using Machine Learning*. Thèse de doctorat. Université de Bordeaux. <https://hal.inria.fr/tel-01968241>

Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience*. New York: Harper & Row.

Csikszentmihalyi, M. (1975). *Beyond boredom and anxiety*. San Francisco: Jossey-Bass.

Fadljević, L., Maitz, K., Kowald, D., Pammer-Schindler, V., & Gasteiger-Klicpera, B. (2020). Slow is good: The effect of diligence on student performance in the case of an adaptive learning system for health literacy. *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, 112-117. <https://doi.org/10.1145/3375462.3375502>

Gottlieb, J., Oudeyer, P. Y., Lopes, M., & Baranes, A. (2013). Information-seeking, curiosity, and attention: computational and neural mechanisms. *Trends in cognitive sciences*, 17(11), 585-593. <https://doi.org/10.1016/j.tics.2013.09.001>.

6 <https://www.dane.ac-versailles.fr/outils/article/adaptiv-langue-et-adaptiv-langue-plus-la-grammaire-par-l-intelligence>

Harrak, F., Bouchet, F., & Luengo, V. (2020). Liens entre performance, assiduité et questions posées et votées en ligne dans le cadre d'une classe inversée. *Sciences et Technologies de L'Information et de La Communication Pour L'Éducation et La Formation*, 27(2). <https://doi.org/10.23709/sticf.27.2.2>

Harrak, F., & Bouchet, F. (2021). Aide au suivi de la progression de groupes d'apprenants pour la mise en place d'une pédagogie différenciée. In M. Lefevre & C. Michel (Éds.), *10e Conférence sur les Environnements Informatiques pour l'Apprentissage Humain* (p. 312-317). <https://hal.archives-ouvertes.fr/hal-03287778>

Käser, T., Busetto, A. G., Solenthaler, B., Kohn, J., von Aster, M., & Gross, M. (2013). Cluster-Based Prediction of Mathematical Learning Patterns. In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Éds.), *Artificial Intelligence in Education* (p. 389-399). Springer. https://doi.org/10.1007/978-3-642-39112-5_40

Khalil, M., & Ebner, M. (2016). Clustering patterns of engagement in Massive Open Online Courses (MOOCs): The use of learning analytics to reveal student categories. *Journal of computing in higher education*, 1-19. <https://doi.org/10.1007/s12528-016-9126-9>

Köck, M., & Paramythis, A. (2011). Activity sequence modelling and dynamic clustering for personalized e-learning. *User Modeling and User-Adapted Interaction*, 21(1), 51-97. <https://doi.org/10.1007/s11257-010-9087-z>

Li, C., & Yoo, J. (2006). Modeling student online learning using clustering. *Proceedings of the 44th annual Southeast regional conference*, 186-191. <https://doi.org/10.1145/1185448.1185490>

Lopes, M., Lang, T., Toussaint, M., & Oudeyer, P. Y. (2012). Exploration in model-based reinforcement learning by empirically estimating learning progress. *Advances in neural information processing systems*, 206-214.

Lopez, M. I., Luna, J. M., Romero, C., & Ventura, S. (2012). Classification via Clustering for Predicting Final Marks Based on Student Participation in Forums. *Proc. of the 5th International Conference on Educational Data Mining*. EDM 2012, Chania, Greece. <https://eric.ed.gov/?id=ED537221>

McBroom, J., Yacef, K., & Koprinska, I. (2020). DETECT: A Hierarchical Clustering Algorithm for Behavioural Trends in Temporal Educational Data. In *International Conference on Artificial Intelligence in Education* (pp. 374-385). Springer, Cham. https://doi.org/10.1007/978-3-030-52237-7_30

Oudeyer, P. Y., Gottlieb, J., & Lopes, M. (2016). Intrinsic motivation, curiosity, and learning: Theory and applications in educational technologies. *Progress in brain research*, 229, 257-284.

Ryan, R. M., & Deci, E. L. (2000). Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. *Contemporary Educational Psychology*, 25(1), 54-67. <https://doi.org/10.1006/ceps.1999.1020>

Térouanne, S., Croset, M.-C., Guiol, D., & Soury-Lavergne, S. (2021). Les concepts didactiques comme outils de conception pour l'Intelligence Artificielle en éducation. *Actes du 47ème colloque de la COPIRELEM*. Grenoble, France.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard university press.