



HAL
open science

Algorithmes d'espacement adaptatif de l'apprentissage pour l'optimisation de la maîtrise à long terme de composantes de connaissance

Benoît Choffin

► **To cite this version:**

Benoît Choffin. Algorithmes d'espacement adaptatif de l'apprentissage pour l'optimisation de la maîtrise à long terme de composantes de connaissance. Apprentissage [cs.LG]. Université Paris-Saclay, 2021. Français. NNT : 2021UPASG001 . tel-03216648

HAL Id: tel-03216648

<https://theses.hal.science/tel-03216648v1>

Submitted on 4 May 2021

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.

Algorithmes d'espacement adaptatif de l'apprentissage pour l'optimisation de la maîtrise à long terme de composantes de connaissance

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, ED Sciences et technologies de
l'information et de la communication (STIC)
Spécialité de doctorat : Informatique
Unité de recherche : Université Paris-Saclay, CNRS, Laboratoire
interdisciplinaire des sciences du numérique, 91405, Orsay, France
Réfèrent : CentraleSupélec

Thèse présentée et soutenue à Paris-Saclay,
le 28 janvier 2021, par

Benoît Choffin

Composition du jury :

Bruno Defude Professeur, Télécom SudParis	Président
Vanda Luengo Professeure des universités, Sorbonne Université	Rapporteur & Examinatrice
Pierre-Yves Oudeyer Directeur de recherche, Inria Bordeaux	Rapporteur & Examineur
Armelle Brun Maîtresse de conférences HDR, Université de Lorraine	Examinatrice
Anne-Laure Ligozat Maîtresse de conférences HDR, ENSIIE	Examinatrice
Franck Ramus Directeur de recherche, ENS (Paris)	Examineur
Yolaine Bourda Professeure, CentraleSupélec	Directrice de thèse
Fabrice Popineau Professeur, CentraleSupélec	Co-encadrant de thèse
Jill-Jënn Vie Chargé de recherche, Inria Lille	Invité

Table des matières

Table des matières	iii
Liste des figures	v
Liste des tableaux	ix
Liste des acronymes	xv
Nomenclature	xvii
1 Introduction	3
1.1 Problématiques	7
1.2 Cadre de recherche	7
1.3 Notations	10
1.4 Contributions	10
1.5 Publications	12
1.6 Plan	13
2 État de l’art	15
2.1 Modélisation de l’apprenant	15
2.2 Systèmes d’espace adaptatif et personnalisé de l’apprentissage	23
2.3 Optimisation de séquences d’apprentissage dans les systèmes tutoriels intelligents	32
3 DAS3H : un modèle prédictif et interprétable de l’apprentissage et l’oubli de CC	35
3.1 Introduction	35
3.2 Limites du modèle DASH	36
3.3 Description du modèle DAS3H	36
3.4 Méthodologie expérimentale de validation	41
3.5 Résultats	47
3.6 Discussion	53
3.7 Conclusion	57
4 Heuristiques d’espace adaptatif et personnalisé pour la maîtrise à long terme de CC	59
4.1 Introduction	59
4.2 Protocole expérimental de comparaison des stratégies	60
4.3 Stratégies de sélection	67
4.4 Résultats	72
4.5 Discussion et analyses complémentaires	79
4.6 Conclusion	99

5	Espacement adaptatif de la révision de CC par apprentissage par renforcement profond	101
5.1	Apprentissage par renforcement et espacement adaptatif de l'apprentissage de CC	102
5.2	Notre algorithme AC4S	106
5.3	Expériences	108
5.4	Discussion	112
5.5	Conclusion	115
 6	 Conclusion	 117
6.1	Résumé des contributions de la thèse	117
6.2	Cas d'usage des algorithmes développés dans cette thèse	118
6.3	Perspectives	120
 Bibliographie		 127

Liste des figures

1.1	Illustration de l’espacement de l’apprentissage et de la nécessité de bien choisir l’intervalle d’espacement entre les révisions	4
1.2	Schéma du fonctionnement d’un algorithme d’espacement adaptatif et personnalisé de l’apprentissage	6
1.3	Q-matrice liée au jeu de données RoboMission (EFFENBERGER, 2019). La q-matrice est ici transposée dans un souci de mise en page (en lignes, les CC et en colonnes les items). Les intitulés des CC sont donnés plus bas. Un carré noir indique que la CC de la ligne est mise en jeu par l’item de la colonne.	10
2.1	Schéma du fonctionnement du système de Leitner (LEITNER, 1972), tiré de la page Wikipedia du système de Leitner.	29
2.2	Capture d’écran venant de la plateforme d’apprentissage de <i>flashcards</i> Anki .	30
2.3	Passage d’algorithmes d’espacement adaptatif d’items à celui de CC	32
3.1	Schéma de fonctionnement des fenêtres temporelles du module h_θ dans les modèles DASH (LINDSEY et al., 2014) et DAS3H	39
3.2	Diagramme montrant l’impact conjoint des $\theta_{k,2w+1}$ (coefficients des réussites) et des $\theta_{k,2w+2}$ (coefficients des tentatives) sur la probabilité de réussite future sur la CC k , dans la fenêtre temporelle w	40
3.3	Exemples de courbes d’oubli générées par notre modèle d’apprenant DAS3H. Sur chaque graphique, deux situations sont présentées : en bleu, un apprenant de niveau initial moyen pratique une CC à $t = 0$ mais échoue; en orange, l’apprenant réussit. Ces courbes ont été générées selon notre protocole expérimental principal, décrit dans la sous-section 4.2.3. Ce sont les équivalents pour l’oubli des courbes d’apprentissage idéalisées de GOUTTE et al. (2018).	40
3.4	Comparaison des AUC du modèle DAS3H original et d’une version alternative qui n’utilise pas de fenêtres temporelles, sur tous les jeux de données. Une métrique plus haute est meilleure.	54
4.1	Schéma du processus d’apprentissage et de mémorisation pour un apprenant s dans une exécution de la simulation	63
4.2	Exemples de distributions statistiques des valeurs générées pour les $\alpha_s, \delta_j, \beta_k, \theta_{k,2w+1}$ et $\theta_{k,2w+2}$	64
4.3	Transposée d’une q-matrice aléatoire (10 CC, 10 items par CC). Chaque carré noir indique que l’item de la colonne met en jeu la CC de la ligne. Notons que les items ne peuvent pas inclure de CC postérieure à la CC courante, ce qui explique la forme triangulaire supérieure de la matrice. Nous avons transposé la q-matrice ici pour des raisons de mise en page.	65
4.4	Distributions des scores <i>PMR</i> individuels durant la période d’apprentissage et durant la période de rétention (expérience de la section 4.4)	74

4.5	Matrice montrant la significativité des différences de scores PMR_A entre chaque stratégie (sauf la stratégie sans révision). La couleur et la valeur indiquée dans chaque carré représentent la différence entre le score PMR_A moyen de la stratégie de la ligne et le score PMR_A moyen de la stratégie de la colonne : rouge quand cette différence est positive et significative et bleu quand elle est négative et significative. Le nombre d'étoiles indique le résultat du test statistique : *** quand la p -valeur est plus petite que 0,001, ** quand elle est plus petite que 0,01 et * quand elle est plus petite que 0,05.	76
4.6	Matrice montrant la significativité des différences de scores PMR_R entre chaque stratégie (sauf la stratégie sans révision). La couleur et la valeur indiquée dans chaque carré représentent la différence entre le score PMR_R moyen de la stratégie de la ligne et le score PMR_R moyen de la stratégie de la colonne : rouge quand cette différence est positive et significative et bleu quand elle est négative et significative. Le nombre d'étoiles indique le résultat du test statistique : *** quand la p -valeur est plus petite que 0,001, ** quand elle est plus petite que 0,01 et * quand elle est plus petite que 0,05.	77
4.7	Comparaison de l'évolution temporelle du score PMR entre différentes stratégies de sélection de CC. Pour les stratégies paramétrées, seul le meilleur paramètre est affiché. Une PMR supérieure implique une meilleure maîtrise des CC.	78
4.8	Comparaison de l'évolution temporelle du score PMR relatif entre différentes stratégies de sélection de CC. Pour les stratégies paramétrées, seul le meilleur paramètre est affiché. Une PMR supérieure implique une meilleure maîtrise des CC.	79
4.9	Exemples de courbes d'oubli générées par notre modèle d'apprenant DAS3H, avec des $\theta_{k,w}$ échantillonnés depuis la zone 6 de la figure 3.2. Deux situations sont présentées : en bleu, un apprenant de niveau initial moyen pratique la CC à $t = 0$ mais échoue ; en orange, l'apprenant réussit. Ce sont les équivalents pour l'oubli des courbes d'apprentissage idéalisées de GOUTTE et al. (2018).	83
4.10	Distributions des révisions <i>directes</i> de CC par semaine, par stratégie (expériences de la section 4.4).	92
4.11	Comparaison du score PMR_R par stratégie de sélection de CC (expériences de la section 4.4). Pour les stratégies paramétrées, seul le meilleur paramètre est affiché. Une PMR supérieure implique une meilleure maîtrise des CC.	93
4.12	Comparaison du nombre d'items qui mettent en jeu chaque CC de la q-matrice aléatoire générée selon notre protocole à la figure 4.3 (voir sous-section 4.2.3)	93
4.13	Comparaison du score PMR_R moyen, relatif à la stratégie de sélection de CC <i>Random</i> , pour différentes valeurs de θ (θ -threshold, sélection d'item <i>Random</i> CC)	96
4.14	Comparaison du score PMR_R par indice d'introduction de CC et paramètre θ (stratégie θ -threshold, expériences de la section 4.4)	96
4.15	Comparaison du score PMR_R moyen, relatif à la stratégie de sélection de CC <i>Random</i> , pour différentes valeurs de θ (θ -threshold (multi.), expériences de la section 4.4)	97
4.16	Comparaison du score PMR_R par indice d'introduction de CC et paramètre θ (stratégie θ -threshold (multi.), expériences de la section 4.4)	97
4.17	Comparaison du score PMR_R moyen en fonction de différentes valeurs de μ (stratégie μ -back, expériences de la section 4.4)	98
4.18	Comparaison du score PMR_R par indice d'introduction de CC et paramètre μ (stratégie μ -back, expériences de la section 4.4)	98

5.1	Diagramme du fonctionnement de notre algorithme AC4S. Les nombres au-dessus des rectangles désignent le nombre d'unités de cette partie du réseau de neurones et « Couche CC » signifie « couche complètement connectée ». L'opérateur \odot représente quant à lui la multiplication élément par élément entre deux vecteurs. Enfin, y_t représente la réponse de l'apprenant à l'item a_t .	108
5.2	Évolution de la PMR_R relative lissée d'AC4S en fonction des épisodes. Les droites horizontales représentent les PMR_R relatives des heuristiques concurrentes, moyennées sur tous les épisodes et toutes les simulations. La zone ombrée autour de la courbe orange représente un intervalle de confiance à 95 % pour AC4S.	110
5.3	Évolution de la PMR_R lissée d'AC4S en fonction des épisodes. Les droites horizontales représentent les PMR_R des heuristiques concurrentes, moyennées sur tous les épisodes et toutes les simulations. La zone ombrée autour de la courbe orange représente un intervalle de confiance à 95 % pour AC4S.	111
5.4	Évolution du compteur lissé d'interactions avec les CC du programme par apprenant en fonction des épisodes. Les droites horizontales représentent les compteurs des heuristiques concurrentes, moyennés sur tous les épisodes et toutes les simulations.	113
5.5	Comparaison des distributions des difficultés des items sélectionnés par les heuristiques et par AC4S, toutes simulations confondues.	114
5.6	Comparaison des distributions des difficultés des CC sélectionnées par les heuristiques et par AC4S, toutes simulations confondues.	114
6.1	Capture d'écran du plugin Moodle PLUME (CHOFFIN, 2018), servant à générer des feuilles d'exercices personnalisées	120

Liste des tableaux

2.1	Tableau comparatif des caractéristiques des différents algorithmes d'espace adaptatif décrits dans la section 2.2, triés par ordre d'introduction dans l'état de l'art. Nous n'incluons pas dans ce tableau les algorithmes non-adaptatifs. APR signifie « Apprentissage par renforcement »	25
3.1	Statistiques sur les jeux de données utilisés après pré-traitement	44
3.2	Caractéristiques considérées par chaque modèle d'apprenant	45
3.3	Nombre de paramètres à estimer en fonction du modèle d'apprenant, quand $d = 0$	46
3.4	Comparaison des performances des différents modèles sur le jeu de données algebra05 (généralisation <i>forte</i>). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu'une métrique plus haute (plus basse) est meilleure. 48	48
3.5	Comparaison des performances des différents modèles sur le jeu de données algebra05 (généralisation <i>semi-forte</i>). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu'une métrique plus haute (plus basse) est meilleure.	48
3.6	Comparaison des performances des différents modèles sur le jeu de données bridge06 (généralisation <i>forte</i>). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu'une métrique plus haute (plus basse) est meilleure. 49	49
3.7	Comparaison des performances des différents modèles sur le jeu de données bridge06 (généralisation <i>semi-forte</i>). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu'une métrique plus haute (plus basse) est meilleure.	49
3.8	Comparaison des performances des différents modèles sur le jeu de données assist12 (généralisation <i>forte</i>). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu'une métrique plus haute (plus basse) est meilleure. 50	50
3.9	Comparaison des performances des différents modèles sur le jeu de données assist12 (généralisation <i>semi-forte</i>). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu'une métrique plus haute (plus basse) est meilleure.	50
3.10	Comparaison des performances des différents modèles sur le jeu de données assist09 (généralisation <i>forte</i>). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu'une métrique plus haute (plus basse) est meilleure. 51	51

3.11	Comparaison des performances des différents modèles sur le jeu de données assist09 (généralisation <i>semi-forte</i>). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu'une métrique plus haute (plus basse) est meilleure.	51
3.12	Comparaison des performances des différents modèles sur le jeu de données robomission (généralisation <i>forte</i>). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu'une métrique plus haute (plus basse) est meilleure.	52
3.13	Comparaison des performances des différents modèles sur le jeu de données robomission (généralisation <i>semi-forte</i>). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu'une métrique plus haute (plus basse) est meilleure.	52
3.14	Comparaison de l'AUC (généralisation <i>forte</i>) sur tous les jeux de données entre DAS3H et DAS3H _{1p} , la version de DAS3H pour laquelle l'influence de la pratique passée ne diffère pas d'une CC à l'autre. Les écarts-types sont indiqués après la moyenne sur les 5 plis. Une métrique plus haute est meilleure.	55
3.15	Comparaison de l'AUC sur assist12 entre DASH [items] et DASH [CC]. Les écarts-types sont indiqués après la moyenne sur les 5 plis. Une métrique plus haute est meilleure.	56
4.1	Hypothèses de distribution des paramètres de nos simulations utilisant le modèle DAS3H. Les $\theta_{k,2w+1}$ et $\theta_{k,2w+2}$ choisis correspondent à la zone 1 dans la figure 3.2.	64
4.2	Domaines pour les paramètres des différentes stratégies de sélection de CC	69
4.3	Tableau comparatif des caractéristiques des différentes heuristiques de sélection de CC utilisées dans nos algorithmes d'espacement adaptatif décrits dans la section 4.3. La colonne « Horizon temporel » indique l'horizon temporel considéré par l'heuristique pour sa décision. Plus le nombre de « + » de la colonne « Difficulté de mise en œuvre » est élevé, plus l'heuristique est difficile à mettre en œuvre.	72
4.4	Comparaison des métriques de performance PMR_A et PMR_R entre toutes les stratégies de sélection des CC. La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.	74
4.5	Comparaison des métriques de performance PMR_A et PMR_R relatives à <i>Random</i> , entre toutes les stratégies de sélection des CC. La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des étudiants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.	74
4.6	Comparaison du nombre moyen de CC pratiquées durant une session d'apprentissage entre les stratégies de sélection d'item <i>Random CC</i> et <i>Max CC</i> pour toutes les stratégies de sélection de CC (expériences de la section 4.4). La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants.	81

4.7	Comparaison des métriques de performance PMR_A et PMR_R entre les stratégies de sélection d'item <i>Random CC</i> et <i>Max CC</i> , pour toutes les stratégies de sélection de CC (expériences de la section 4.4). La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.	81
4.8	Comparaison des métriques de performance PMR_A et PMR_R entre les stratégies de sélection d'item <i>Random CC</i> et <i>Max CC</i> , pour toutes les stratégies de sélection de CC, <i>relatives</i> à la stratégie <i>Random</i> (expériences de la section 4.4). La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.	81
4.9	Comparaison des métriques de performance PMR_A et PMR_R entre toutes les stratégies de sélection des CC. Les paramètres θ sont échantillonnés dans la zone 6 du diagramme de la figure 3.2. La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.	84
4.10	Comparaison des métriques de performance PMR_A et PMR_R , <i>relatives</i> à la stratégie <i>Random</i> , entre toutes les stratégies de sélection des CC. Les paramètres θ sont échantillonnés dans la zone 6 du diagramme de la figure 3.2. La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.	85
4.11	Comparaison des métriques de performance PMR_A et PMR_R entre les stratégies θ - <i>threshold</i> et <i>Greedy</i> , avec les vrais paramètres du modèle ou les estimations de ces paramètres (expériences de la section 4.4). La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.	87
4.12	Comparaison des métriques de performance PMR_A et PMR_R entre toutes les stratégies de sélection des CC. Les courbes d'apprentissage et d'oubli sont identiques entre les différentes CC d'une même exécution (modèle générateur DAS3H _{1p}). La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.	89
4.13	Comparaison des métriques de performance PMR_A et PMR_R , <i>relatives</i> à la stratégie <i>Random</i> , entre toutes les stratégies de sélection des CC. Les courbes d'apprentissage et d'oubli sont identiques entre les différentes CC d'une même exécution de la simulation (modèle générateur DAS3H _{1p}). La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.	89

Liste des algorithmes

1	Processus d'apprentissage et de révision pour un seul apprenant	61
2	Protocole expérimental de comparaison de stratégies sur des apprenants simulés	62
3	Procédure de sélection gloutonne d'un sous-ensemble de CC	71
4	Algorithme AC4S	107

Liste des acronymes

- A2C** *Advantage Actor-Critic*. 106
- AC4S** *Actor-Critic 4 Spacing*. 106
- ACT-R** *Adaptive Character of Thought-Rational*. 20
- AFM** *Additive Factor Model*. 17
- AUC** *Area Under the ROC Curve*. 46
- BKT** *Bayesian Knowledge Tracing*. 16
- CC** *Composante de Connaissance*. 6
- DAS3H** *item Difficulty, student Ability, Skill and Student Skill practice History*. 35
- DASH** *Difficulty, Ability and Student History*. 21
- DKT** *Deep Knowledge Tracing*. 16
- IRT** *Item Response Theory*. 15
- ITS** *Intelligent Tutoring System*. 33
- KC** *Knowledge Component*. 6
- KT** *Knowledge Tracing*. 16
- KTM** *Knowledge Tracing Machines*. 18
- LSTM** *Long Short-Term Memory*. 28
- MCM** *Multiscale Context Model*. 20
- MDP** *Markov Decision Process*. 102
- MIRT** *Multidimensional Item Response Theory*. 16
- NLL** *Negative Log-Likelihood*. 47
- PFA** *Performance Factors Analysis*. 18
- PMR** *Probabilité Moyenne de Réussite*. 65
- POMDP** *Partially Observable Markov Decision Process*. 12
- QCM** *Questionnaire, ou Question à Choix Multiple*. 5
- ReLU** *Rectified Linear Unit*. 106
- RMSE** *Root Mean Square Error*. 47
- RNN** *Recurrent Neural Network*. 17

Nomenclature

α_s	paramètre de niveau de l'apprenant s dans IRT, DASH, DAS3H, page 15
β_k	paramètre de facilité latente de la CC k dans les modèles AFM, PFA et DAS3H, page 17
δ_j	paramètre de difficulté de l'item j dans IRT, MIRT, DASH, DAS3H, page 15
η_k	coefficient associé au compteur de réponses correctes $c_{s,k}$, pour le modèle PFA, page 18
$\gamma_{s,k}$	coefficient associé au compteur de tentatives $a_{s,k}$, pour le modèle AFM, page 17
μ	paramètre de l'heuristique d'espacement μ -back, page 26
ϕ	fonction de répartition d'une loi normale $\mathcal{N}(0, 1)$, page 18
π	politique de l'apprentissage par renforcement, page 103
ρ_k	coefficient associé au compteur de mauvaises réponses $f_{s,k}$, pour le modèle PFA, page 18
σ	fonction logistique, page 15
α_s	vecteur latent de caractéristiques de l'apprenant s dans MIRT, page 16
\mathbf{d}_j	vecteur latent de caractéristiques de l'item j dans MIRT, page 16
τ_0	début de la période de rétention, page 10
θ	paramètre de l'heuristique d'espacement adaptatif θ -threshold, page 26
μ_t	masque d'actions dans l'algorithme d'apprentissage par renforcement, page 107
θ	vecteur de coefficients des compteurs de bonnes réponses et de tentatives dans les modèles DASH et DAS3H, page 21
A	fonction avantage dans l'algorithme AC4S, page 107
a_t	action effectuée par l'agent d'apprentissage par renforcement à l'instant t , page 102
$a_{s,j,w}$	nombre de tentatives de l'apprenant s sur l'item j dans la fenêtre temporelle w , dans le modèle DASH, page 21
$a_{s,k,w}$	nombre de tentatives passées par l'apprenant s sur la CC k dans la fenêtre temporelle w , page 38
$a_{s,k}$	nombre de tentatives de l'apprenant s sur la CC k avant l'instant courant, pour le modèle AFM, page 17
c	nombre d'items générés par CC dans notre cadre de simulation, page 65
$c_{s,j,w}$	nombre de bonnes réponses de l'apprenant s sur l'item j dans la fenêtre temporelle w , dans le modèle DASH, page 21
$c_{s,k,w}$	nombre de fois où l'apprenant s a par le passé répondu correctement à un item mettant en jeu la CC k dans la fenêtre temporelle w , page 38

$c_{s,k}$	nombre de bonnes réponses de l'apprenant s sur la CC k avant l'instant courant, pour le modèle PFA, page 18
d	dimension de plongement vectoriel dans KTM, page 37
$f_{s,k}$	nombre de mauvaises réponses de l'apprenant s sur la CC k avant l'instant courant, pour le modèle PFA, page 18
h_θ	module temporel de DASH et DAS3H, page 21
J	nombre d'items, page 10
j	identifiant de l'item, page 15
K	nombre de CC, page 10
k	identifiant de CC, page 17
N	nombre de caractéristiques, pour le modèle KTM, page 18
o_t	observation fournie à l'agent d'apprentissage par renforcement à l'instant t , page 102
$q_{j,k}$	indiatrice de la relation entre l'item j et la CC k dans la q-matrice q , page 10
r	nombre d'items par séance de révision dans notre cadre de simulation, page 61
r_t	récompense fournie à l'agent d'apprentissage par renforcement à l'instant t , page 102
R_{finale}	récompense finale fournie à l'agent d'apprentissage par renforcement, page 105
S	nombre d'apprenants, page 10
s	identifiant de l'apprenant, page 15
s_t	état de l'environnement à l'instant t , page 102
T	fin de la période de rétention, page 10
t	instant, page 15
V	fonction valeur dans l'algorithme AC4S, page 106
w	indice de fenêtre temporelle, dans les modèles DASH et DAS3H, page 21
$Y_{s,j,t}$	réponse de l'apprenant s sur l'item j à l'instant t , page 15
W	nombre total de fenêtres temporelles dans les modèles DASH et DAS3H, page 21

Remerciements

Je souhaitais tout d'abord remercier les membres du jury d'avoir accepté de lire mon manuscrit et d'avoir été présents à ma soutenance. Au cours de ma thèse, j'ai également eu la chance d'être invité par Andrew S. Lan à passer quelques semaines dans son laboratoire aux États-Unis pour travailler sur ce qui deviendra plus tard le chapitre 5 de ce manuscrit. Merci énormément à toi Andrew et à toi Aritra pour cette collaboration fructueuse et passionnante sur le développement d'algorithmes d'apprentissage par renforcement profond ! Enfin, je ne pouvais pas finir ce paragraphe sans remercier chaleureusement mes deux géniaux encadrants de thèse, vu que vous m'avez quand même supporté pendant plus de trois ans (c'est peut-être pour ça que les thèses ne durent pas plus longtemps ?). Merci donc à toi Fabrice et à toi Yolaine pour m'avoir fait grandir scientifiquement, pour vos relectures patientes et minutieuses et pour votre aide à des heures parfois indues, pour votre chaleur et votre soutien dans les moments plus difficiles ; en un mot, pour votre encadrement attentif et profondément humain. Je suis fier et heureux d'avoir été votre doctorant pendant ces trois années.

Puisqu'on en est à parler de ce que la thèse m'a apporté de plus chouette, comment ne pas évoquer toutes les belles rencontres que j'y ai faites ? Merci d'abord à tous mes amis du LRI qui justifiaient largement les heures passées dans les transports en commun pour rejoindre la contrée lointaine de Paris-Saclay : Julien¹, Adnan, Sophie et Yuting. Presque à l'exact opposé de mon laboratoire se trouve la start-up Didask, que je remercie vivement de m'avoir si gentiment et si régulièrement accueilli². L'un des effets secondaires du doctorat est de vous faire rencontrer des amis à des moments où vous vous y attendez le moins. Ainsi, merci également à toi Alexandre, que je suis heureux d'avoir rencontré lors des RJC EIAH 2018, et merci à Mary et Carlos de m'avoir accueillis chez vous à Amherst et de m'avoir fait me sentir chez moi avec une telle simplicité alors que j'étais à des milliers de kilomètres de la France. Enfin, je voulais remercier du fond du cœur Alice et Jill-Jënn. S'il est vrai que je ne vous ai pas techniquement rencontrés au cours de ma thèse, vous avez été pour moi de véritables modèles doublés d'amis chers : des compagnons de route scientifiques avec qui travailler n'était jamais un effort mais toujours un plaisir.

Trois années de doctorat, ça n'est pas rien, et pourtant, celles-ci sont passées comme un éclair grâce à la présence de mes amis à mes côtés. Dans le désordre le plus assumé, citons ma bande de CheerUpiens, mes « vieux » amis d'Angers et ceux de l'ENSAE. Merci d'avoir été si nombreux lors de ma soutenance, merci pour les brunchs du dimanche matin et les bières du vendredi soir, pour les parties de jeux de rôle comme pour les soirées de concerts de rock au Supersonic. Je voulais plus particulièrement remercier Catherine qui a tenu à venir assister en personne³ à ma soutenance alors que celle-ci était contrainte au distanciel par les conditions sanitaires.

Alors que l'on s'approche de la fin de ces remerciements, je voudrais rendre ici un hommage particulier à deux remarquables représentants de la gent féline : Winston et Bellisarius,

1. Partenaire privilégié de matchs endiablés de baby-foot, de billard ou encore de boxe, qui se soldaient bien souvent par ma cuisante défaite !

2. C'est également dans ces lieux que j'ai rencontré ma très chère compagne, mais nous y reviendrons plus bas.

3. Comprendre : en chair et en os.

merci à vous deux d'avoir contribué avec une redoutable efficacité à égayer mon quotidien de thésard confiné.

Maman, Papa, Claire, Laure et Paul, merci infiniment pour votre affection indéfectible et inconditionnelle. Merci de me soutenir dans tout ce que j'entreprends et d'être toujours présents, dans les moments difficiles comme dans les moments heureux. Il va sans dire que vous avez largement contribué à la réussite de ce doctorat.

Je ne pouvais pas finir ces remerciements sans parler de celle que j'admire le plus : toi, ma très chère et très tendre Svet. Sur le chemin escarpé qui menait à la fin de ce doctorat, tu as su être un guide patient, encourageant et un soutien sans faille pour moi. Toute ma gratitude et tout mon amour te sont acquis.

Chapitre 1

Introduction

Si l'on demandait aujourd'hui à un élève ou à un étudiant si des révisions étalées dans le temps sont bénéfiques à l'apprentissage, il y a fort à parier qu'il répondrait favorablement. Peu d'apprenants remettent en question les avantages qu'il y a à revenir périodiquement sur des connaissances que l'on a acquises auparavant. Toutefois, rares sont ceux qui mettent réellement en pratique cette stratégie pour consolider leurs connaissances à long terme¹ (TARABAN et al., 1999; COHEN et al., 2013) et plus nombreux sont ceux qui choisissent des stratégies d'apprentissage favorisant la mémoire à court terme (ROEDIGER III et McDERMOTT, 2018; MOZER et al., 2016). En effet, malgré les recommandations de nombreux chercheurs en sciences cognitives (WEINSTEIN et al., 2018; PASHLER, BAIN et al., 2007; DUNLOSKY et al., 2013), les programmes scolaires comportent encore trop peu d'incitations à la révision systématique de connaissances et par conséquent à la mémorisation à long terme (LINDSEY et al., 2014). Déjà en 1967, PIMSLEUR notait l'absence presque totale de ce genre d'incitations dans les manuels ou dans la formation des enseignants et des formateurs. Sachant que l'apprentissage de nouvelles connaissances s'appuie souvent sur des connaissances plus anciennes et requiert un investissement temporel ainsi que des efforts cognitifs importants, améliorer la mémorisation à long terme et la pérennité des connaissances est une problématique essentielle de tout apprentissage.

La révision périodique de connaissances est plus connue en psychologie cognitive sous le nom de *répétition espacée* (N. J. CEPEDA, VUL et al., 2008; DEMPSTER, 1989; ROEDIGER III et KARPICKE, 2011; KANG, 2016). Plus formellement, la stratégie de répétition espacée consiste à découper l'apprentissage d'un même ensemble d'informations (par exemple, des mots de vocabulaire dans une langue étrangère) en plus petites sessions d'apprentissage espacées dans le temps. La répétition espacée améliore la mémorisation à long terme de ces informations par rapport à un apprentissage dans une seule session dite « massée »² : ce bénéfice de la répétition espacée est appelé effet d'espacement (ou *spacing effect*, en anglais). Notons ici que cette stratégie est meilleure que l'apprentissage « massé » pour la mémorisation à long terme *même* à durée égale d'apprentissage.

Cependant, si les bénéfices de l'espacement comparés à ceux de l'apprentissage massé sont clairement établis, il reste à savoir comment, en pratique, espacer ses révisions. Pour que la révision de connaissances passées n'interfère pas trop avec l'acquisition de nouvelles connaissances, il est nécessaire pour les apprenants de ne pas planifier plus de séances de révision que nécessaire tout en déterminant au mieux les moments de ces sessions. Ceci est d'autant plus important que la bonne durée d'espacement entre l'acquisition initiale et la révision détermine largement l'ampleur de cet effet (N. J. CEPEDA, VUL et al., 2008). La figure 1.1 illustre la nécessité de choisir correctement l'intervalle d'espacement entre les révisions.

1. Nous suivons plusieurs études de psychologie cognitive en entendant par « long terme » une durée de rétention d'un jour ou plus. Une durée de rétention de l'ordre de la minute ou de l'heure correspond quant à elle à du « court terme ».

2. Le bachotage est une forme bien connue des enseignants et des étudiants d'apprentissage « massé ».

L'axe des abscisses représente le temps et l'axe des ordonnées représente la probabilité de rappel de la connaissance. La courbe pleine commence à $t_{acq.}$ (l'instant de l'acquisition initiale) et montre l'oubli au fil du temps en l'absence de révision. Les deux autres courbes montrent l'effet sur la dynamique de l'oubli, selon le moment d'une *unique* session de révision : $t_{réactiv.}^1$ pour la courbe en pointillés et $t_{réactiv.}^2$ pour la courbe en tirets. On peut voir ici que, en fonction de l'emplacement de la révision, les effets de l'espacement sur la mémoire à long terme sont sensiblement différents. En effet, la courbe en pointillés décroît beaucoup plus vite que la courbe en tirets et après une même durée T de rétention, la probabilité de rappel de la connaissance est bien plus élevée pour la courbe en tirets.

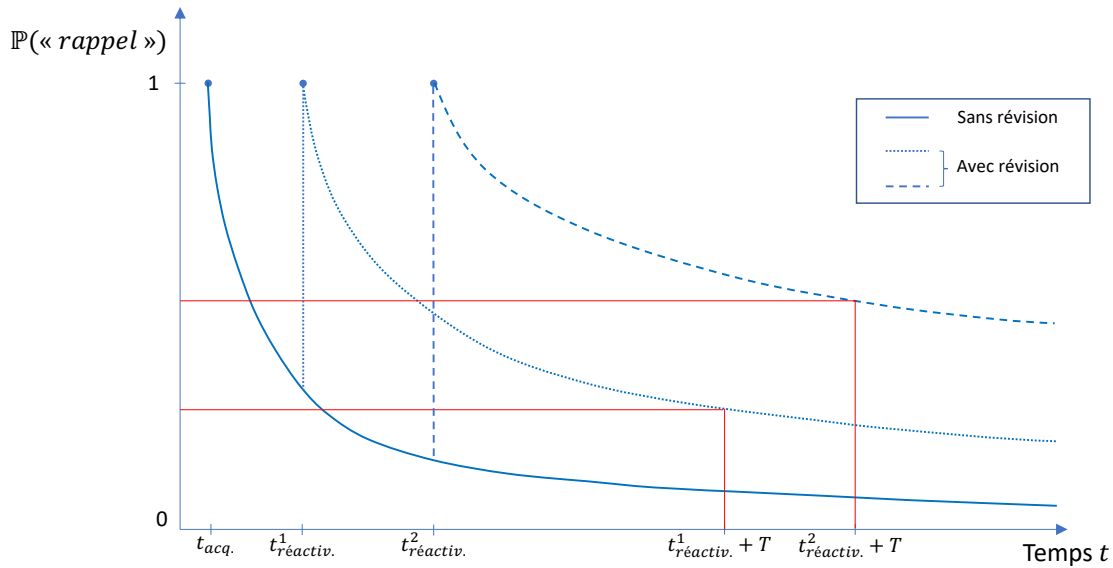


FIGURE 1.1 – Illustration de l'espacement de l'apprentissage et de la nécessité de bien choisir l'intervalle d'espacement entre les révisions

Les résultats de la psychologie cognitive n'offrent pas de recommandation univoque quant à la bonne durée d'espacement entre l'acquisition et la révision. Dans ces expériences, les chercheurs comparent la plupart du temps les effets d'intervalles d'espacement fixes et génériques (c'est-à-dire, partagés au sein d'un groupe de sujets) sur la performance des apprenants à différents tests avec plus ou moins de délai. CARPENTER, N. J. CEPEDA et al. (2012) rappellent que deux types de plannings d'espacement sont régulièrement employés : planning uniforme (l'intervalle entre les différentes révisions reste constant) et planning expansif (l'intervalle entre les différentes révisions croît de révision en révision). Si certaines expériences montrent que l'expansif est meilleur que les autres types de plannings (dont le planning uniforme), d'autres montrent au contraire qu'il n'y a pas de différences entre planning uniforme et planning expansif (METTLER et al., 2016 ; LATIMIER et al., 2020). Pour METTLER et al., cette variabilité des résultats vient précisément des différences entre les expériences dues au contenu pédagogique, aux apprenants et aux conditions d'apprentissage. Les plannings génériques ne sont par essence pas à même de s'adapter à la variabilité des difficultés de récupération en mémoire des différents éléments à réviser. SENSE, BEHRENS et al. (2016) apportent une autre preuve de la sous-optimalité des plannings de révision génériques : les résultats de leur expérience suggèrent que les taux d'oubli des apprenants varient d'un contenu pédagogique à l'autre. Un planning de révision identique pour tous les items à mémoriser dans une expérience risque donc d'être moins efficace qu'un planning qui s'adapte à ces différences.

Fournir aux apprenants des plannings adaptatifs et personnalisés de répétition espacée serait donc plus efficace que des plannings génériques, qui donnent le même planning à tout le monde. Des avancées scientifiques récentes ont permis le développement de systèmes d'es-

acement adaptatif et personnalisé de l'apprentissage pour l'amélioration de la mémorisation à long terme de *flashcards* (TABIBIAN et al., 2019; REDDY, LEVINE et al., 2017; LINDSEY et al., 2014). À partir de l'historique d'apprentissage passé d'un apprenant, ces outils choisissent séquentiellement

- soit l'item (ou question, exercice) optimal à faire réviser à l'apprenant à un instant t ;
- soit le prochain instant auquel faire réviser l'item à l'apprenant.

Définition 1 – Flashcard. Une *flashcard* est une carte utilisée pour la révision périodique d'un élément de connaissance simple, pouvant se décliner sous la forme d'une question et d'une réponse. Par exemple, les *flashcards* peuvent servir à réviser des mots de vocabulaire dans une langue étrangère ou des dates en histoire.

Une *flashcard* peut être physique ou électronique. Elle comporte un endroit et un envers : l'endroit présente à l'apprenant la question à laquelle il doit répondre et l'envers, la réponse à cette question³.

Définition 2 – Item. Un item est une activité pédagogique impliquant une récupération en mémoire de connaissances de la part de l'apprenant. La notion d'item généralise les notions d'exercice, de question, de test,...

Par exemple, une question dans un QCM est un item, tout comme l'est un exercice de mathématiques.

En se concentrant sur les items qui bénéficieraient le plus d'une révision, ces outils sont capables d'améliorer de manière significative la rétention en mémoire des items étudiés à moyen et long terme, par rapport à des plannings fixes d'espacement. Comparé à un espacement identique pour tous les individus, plusieurs expériences montrent que l'espacement adaptatif maintient un plus haut degré d'ancrage en mémoire à long terme des informations apprises en ralentissant la décroissance des courbes d'oubli (METTLER et al., 2016; LINDSEY et al., 2014). Différents algorithmes d'espacement adaptatif de l'apprentissage sont implémentés dans plusieurs systèmes de *flashcards* électroniques, comme Anki⁴, SuperMemo ou bien Mnemosyne⁵.

De tels systèmes forcent l'apprenant à essayer de se remémorer périodiquement chacun des éléments à mémoriser, c'est pourquoi ils mettent aussi en œuvre la stratégie de *récupération en mémoire* (ROEDIGER III et KARPICKE, 2006; CARPENTER, PASHLER et al., 2008). Cette dernière consiste pour un apprenant à se tester sur ses connaissances après y avoir été exposé plutôt que de simplement relire les mêmes contenus pédagogiques s'y rapportant. Ce test peut prendre différentes formes, comme le rappel libre, le rappel indicé, les QCM, les exercices d'application, etc. Dans le cas des *flashcards*, le test est une forme de rappel indicé, l'indice étant alors la question posée sur l'endroit de la carte. Par rapport à de la simple relecture, la récupération en mémoire génère une meilleure mémorisation à long terme : c'est ce bénéfice qu'on appelle effet de test (ou *testing effect* en anglais). L'effet de test a été reproduit par de nombreuses études en psychologie cognitive. Une méta-analyse récente sur cet effet (ADESOPE et al., 2017) a trouvé une taille d'effet générale positive et forte de $g = 0,61$ ⁶ pour la stratégie de récupération en mémoire comparée à des stratégies de révision sans test. Une autre méta-analyse (PAN et al., 2018) a montré que les exercices de récupération en mémoire facilitaient le transfert d'apprentissage (taille d'effet moyenne mais positive de $d = 0,40$ ⁷), c'est-à-dire le fait que l'apprenant réussisse à mobiliser une connaissance dans un contexte différent de celui dans lequel elle a été apprise. Les algorithmes d'espacement adaptatif et personnalisé combinent donc les stratégies de répétition espacée et de récupération en mémoire. Une telle

3. Par exemple « *En quelle année a eu lieu le sacre de Napoléon Bonaparte ? / 1804* » ou « *Quelle est la traduction du mot "behavior" en français ? / comportement* ».

4. <https://apps.ankiweb.net/>

5. <https://mnemosyne-proj.org/>

6. Il s'agit du g de Hedges, une mesure statistique de la taille d'effet.

7. Il s'agit du d de Cohen, une autre mesure statistique de la taille d'effet.

combinaison s'appelle *récupération espacée en mémoire* et consiste ainsi à distribuer dans le temps les épisodes de récupération en mémoire après une première acquisition de la connaissance (LATIMIER et al., 2020).

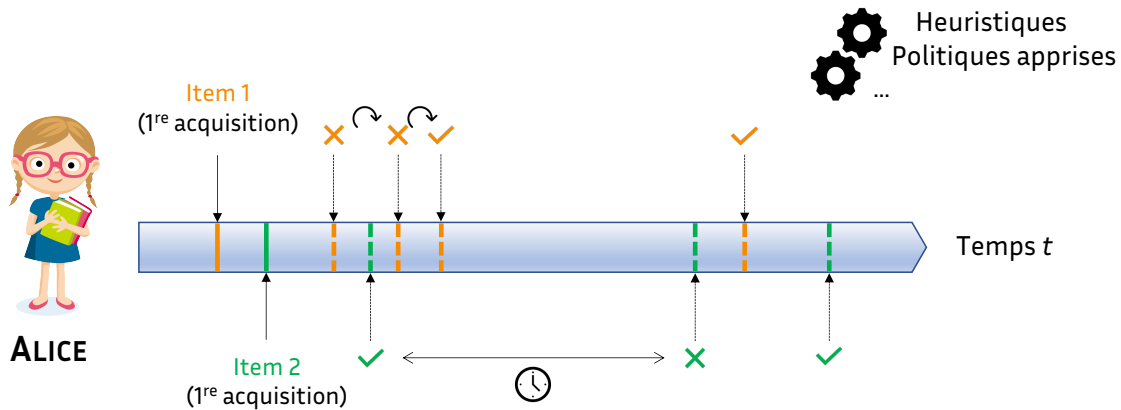


FIGURE 1.2 – Schéma du fonctionnement d'un algorithme d'espacement adaptatif et personnalisé de l'apprentissage

La figure 1.2 présente un schéma du fonctionnement d'un algorithme d'espacement adaptatif de l'apprentissage. Alice doit mémoriser deux items j_1 et j_2 , qu'elle a acquis initialement au début de la frise. Par la suite, le système lui propose une première fois l'item j_1 mais elle ne parvient pas à s'en souvenir. On lui propose alors à nouveau j_1 , plus rapidement cette fois-ci. Elle réussit à se le remémorer mais cela peut provenir de la familiarité de court terme avec j_1 . On lui propose donc à nouveau l'item j_1 , bien plus tard, et elle réussit encore une fois. Pour j_2 , les choses sont différentes : elle réussit à s'en souvenir après la première acquisition. On lui propose à nouveau j_2 bien longtemps après, mais elle échoue à se le remémorer. Elle réussit cependant la dernière fois qu'on lui propose j_2 , ce qui montre que la révision a fait son effet.

Toutefois, ces algorithmes ne fonctionnent actuellement que pour l'apprentissage et la mémorisation pure de connaissances simples, comme des connaissances factuelles (par exemple, des mots de vocabulaire). À notre connaissance, aucun travail de recherche n'a jamais cherché à étendre ces algorithmes quand les connaissances à retenir consistent en un ensemble de composantes de connaissances (CC). Pourtant, l'effet d'espacement ne se cantonne pas à l'apprentissage de vocabulaire ou à la mémorisation pure d'éléments (comme avec les *flashcards*) : il a par exemple été appliqué avec succès à l'acquisition et à la généralisation de concepts scientifiques abstraits (VLACH et al., 2012) et à la pratique de compétences mathématiques dans un contexte éducatif réel (BARZAGAR NAZARI et al., 2019). Il y a donc fort à parier qu'il soit possible de développer des algorithmes d'espacement adaptatif et personnalisé pour l'apprentissage de CC. C'est l'hypothèse qui nous a guidés tout au long de cette thèse, et qui nous a conduits à nous intéresser aux problématiques présentées à la prochaine section.

La notion de CC est régulièrement utilisée dans la recherche en intelligence artificielle pour l'éducation sous le terme anglais de *Knowledge Component* (KC).

Définition 3 – Composante de connaissance (CC). « Nous définissons une composante de connaissance (CC) comme étant une unité *acquise* d'une fonction cognitive ou une structure qui peut être inférée de la performance d'un apprenant sur un ensemble de tâches liées entre elles. [...] En pratique, nous utilisons la notion de composante de connaissance pour généraliser des termes qui décrivent des éléments de cognition ou de connaissance [...] mais également des termes de tous les jours comme concept, principe, fait ou compétence [...]. »⁸ (définition de KOEDINGER et al. (2012)).

8. Passage traduit par nous-mêmes.

Pour être plus clair sur la notion de CC, nous nous appuyons ici sur l'explication proposée par le LearnLab de l'université Carnegie Mellon⁹. Une CC est la « description d'une structure mentale ou d'un processus qu'utilise un apprenant pour résoudre les étapes d'une tâche ou d'un problème. Une CC peut être utilisée seule ou en combinaison avec d'autres CC. » Une CC n'est pas nécessairement formulée explicitement dans l'esprit d'un apprenant : si un apprenant met en œuvre correctement cette CC sans être pour autant capable de la décrire, alors on dit qu'il s'agit d'une CC implicite.

1.1 Problématiques

À la lumière de la littérature, deux grandes problématiques ont guidé les travaux de cette thèse :

1. Comment modéliser l'évolution temporelle de la maîtrise d'un apprenant sur un ensemble de CC à partir de ses réponses passées sur un ensemble d'items ? La prise en compte du phénomène d'oubli au niveau des CC améliore-t-elle significativement les performances prédictives du modèle ?
2. Sachant que l'on dispose pour un apprenant d'un budget fixe d'items de révision et que les périodes de révision sont déterminées par un agent extérieur¹⁰, comment choisir la séquence d'items qui maximisera la maîtrise à long terme de l'apprenant sur les CC du programme ?
 - (a) Peut-on trouver des heuristiques simples et efficaces de sélection de la CC optimale à faire réviser à un instant t ? Sur quelles dimensions peut-on comparer *qualitativement* ces stratégies de révision ? Parmi les heuristiques que nous avons développées, lesquelles optimisent le mieux la mémorisation à long terme de l'apprenant sur l'ensemble des CC ?
 - (b) Nous introduisons dans notre cadre de recherche des items mettant en jeu plusieurs CC à la fois. Peut-on tirer parti de ces relations pour accélérer le processus de révision en sélectionnant un sous-ensemble de CC au lieu d'une seule ? Quel est l'impact sur la complexité des items choisis ?
 - (c) Les stratégies non-adaptatives sont-elles nécessairement moins performantes que les stratégies adaptatives ?
 - (d) Peut-on faire *apprendre* à un algorithme la politique optimale de sélection d'items pour maximiser la mémorisation à long terme des apprenants ? Une telle approche, fondée sur les données, parvient-elle à surpasser des heuristiques conçues à priori ? Combien d'interactions avec des apprenants sont nécessaires avant d'obtenir une politique satisfaisante ?
 - (e) Quel est l'impact du décalage de modèle (en anglais, *model mismatch*) entre modèle générateur de trajectoires d'apprentissage et modèle servant à la recommandation ?

1.2 Cadre de recherche

1.2.1 Introduction

Dans notre cadre de recherche, un apprenant cherche à maîtriser à long terme un ensemble prédéfini et fixe de Composantes de Connaissance (CC) en interagissant séquentiellement avec un système d'espacement adaptatif et personnalisé de l'apprentissage. En général, cet ensemble de CC est défini au préalable par des experts du domaine.

9. https://www.learnlab.org/research/wiki/Knowledge_component

10. Par exemple, un enseignant, un formateur, l'apprenant lui-même,...

À chaque étape, ce système sélectionne un item à soumettre à l'apprenant, par exemple « Calculer $\lim_{x \rightarrow 0} \sin x/x$. ». L'algorithme d'espacement fait ce choix avec l'objectif de maximiser la maîtrise à long terme de l'ensemble de CC à apprendre. Ensuite, l'apprenant résout l'item et le système utilise la justesse de la réponse de l'apprenant pour mettre à jour sa propre estimation du niveau de maîtrise courant et futur de l'apprenant sur les CC mises en jeu par l'item choisi. Enfin, le système fournit le plus souvent à l'apprenant un retour correctif sur l'item.

L'objectif de cet algorithme est d'optimiser la maîtrise à long terme de l'ensemble des CC en sélectionnant soigneusement la séquence d'items sur laquelle chaque apprenant doit s'entraîner. En fonction des points forts et des points faibles de chaque apprenant sur ces CC, l'algorithme peut proposer dynamiquement une séquence différente d'items pour mieux répondre aux besoins de l'apprenant. Nous cherchons pour notre part à étendre les algorithmes d'espacement adaptatif de l'apprentissage dans le cas où les connaissances à maîtriser consistent en l'application de CC sous-jacentes. Dans ce contexte, les apprenants apprennent et renforcent leur niveau de maîtrise en s'entraînant avec des items qui mettent en jeu cette CC. Précisons également que dans notre cadre, cet algorithme choisit l'item optimal à faire réviser à un instant t : cet instant t de révision est décidé par un agent extérieur et non par l'algorithme lui-même.

1.2.2 Hypothèses

Nous nous appuyerons tout au long de cette thèse sur les hypothèses suivantes :

- les connaissances à apprendre et mémoriser sont un ensemble de CC¹¹ ;
- la maîtrise et la mémorisation de CC de l'apprenant s à l'instant t se mesure par sa capacité à généraliser et à transférer ce qu'il a appris pour répondre à un item (inconnu) mettant en jeu cette CC. Cette maîtrise n'est *pas* binaire : elle se mesure sur une échelle continue ;
- chaque nouvelle CC est d'abord introduite de manière théorique aux apprenants mais l'apprentissage se produit par la pratique. Cette première introduction est *indépendante* de l'algorithme d'espacement : elle peut par exemple être décidée par un enseignant qui souhaite présenter une nouvelle CC chaque semaine ;
- les items peuvent faire travailler une ou plusieurs CC et cette information est synthétisée au sein d'une q -matrice binaire (TATSUOKA, 1983). Ainsi, les items ne sont pas indépendants les uns des autres ;
- les CC sont en revanche indépendantes les unes des autres : pratiquer une CC k_1 avec un item aura juste un impact sur la maîtrise de la CC k_1 ;
- les apprenants oublient nécessairement : la maîtrise d'une CC diminue avec le temps passé depuis la dernière interaction avec cette CC ;
- le budget d'items de révision et les moments de ces révisions sont décidés par un agent extérieur à l'algorithme (enseignant, apprenant,...) ;
- les apprenants n'ont aucun contrôle sur les items qu'ils ont à réviser : ce choix repose sur les algorithmes d'espacement ;
- tous les items prennent le même temps de résolution, quel que soit le niveau de maîtrise de l'apprenant ;
- les réponses des apprenants aux items de révision sont binaires : soit une réponse est juste, soit elle est fausse.

11. Ces CC peuvent être organisées sous forme de graphe de prérequis mais cela dépasse le champ d'investigation de cette thèse.

Définition 4 – Q-matrice. Une q-matrice est une matrice binaire dont les lignes correspondent aux items et les colonnes aux CC : à l'intersection de la ligne j et de la colonne k , on trouve un 1 si l'item j met en jeu la CC k , un 0 sinon. Une telle matrice permet donc de définir les relations de dépendance entre items et CC. Sa construction nécessite bien souvent le travail d'experts du domaine de connaissance.

Contrairement à LINDSEY et al., nous ne faisons pas l'hypothèse que les items mettant en jeu une même CC k sont interchangeables : leur difficulté, par exemple, peut varier d'un item à l'autre. Ainsi, il est possible de scinder la phase de sélection en deux volets : (1) la sélection de la (ou des) CC à pratiquer et (2) de l'item à présenter (qui met donc la ou les CC choisies précédemment en œuvre). En théorie, il devrait y avoir au moins un item pour pratiquer chaque CC k ; en pratique, un item serait beaucoup trop peu puisque l'apprenant « surapprendrait » probablement sur cet item. Dans ce cas, remplacer l'item par un item sur modèle (en anglais, *template item*) pourrait être une solution. Cette formalisation généralise celle du cadre des systèmes d'espacement adaptatif de *flashcards* : pour retrouver ce dernier, il suffit d'associer chaque item à une CC distincte. On supprime du même coup la deuxième phase de sélection, celle de l'item.

Dans notre cadre, il y a aussi potentiellement plusieurs sessions de révisions espacées, ainsi que plusieurs items pratiqués dans chaque session (par exemple pour un devoir à la maison hebdomadaire) : nous faisons de l'optimisation de planning de révision intra-session et inter-sessions. Comme elle implique des intervalles d'espacement beaucoup plus importants, la planification de révisions inter-sessions a plus de chances d'améliorer la maîtrise de CC à long terme que l'optimisation intra-session (N. J. CEPEDA, VUL et al., 2008; LINDSEY et al., 2014).

L'algorithme n'a de contrôle que sur la partie de révision. Ainsi, et contrairement au cadre de révision de *flashcards* (REDDY, LABUTOV et al., 2016), nous ne voulons pas optimiser l'arbitrage entre l'introduction de nouvelles CC et la révision d'anciennes CC. Nous préférons plutôt nous concentrer sur l'optimisation de l'arbitrage entre la révision de chacune des différentes CC disponibles. L'ensemble des CC accessibles par le planificateur de révisions à chaque étape est donc défini au préalable par un être humain et constitue une contrainte pour l'algorithme. Par exemple, si l'apprenant a déjà abordé auparavant l'intégralité des CC du programme, alors toutes les CC sont potentiellement accessibles à la révision à chaque étape. À l'inverse, si les révisions se font progressivement, alors les CC de la fin du programme ne sont pas nécessairement disponibles pour être révisées dès le début.

De même, l'algorithme ne décide pas du moment de chaque session de révision. Cette décision est prise par un être humain : par exemple, un enseignant pour toute sa classe ou un apprenant pour préparer un examen spécifique. Cela peut être planifié à l'avance (LINDSEY et al., 2014) ou décidé à la volée par l'apprenant. Par conséquent, notre objectif est de sélectionner la (ou les) meilleure(s) CC à faire réviser à un moment donné, et non de trouver le meilleur moment pour faire réviser chaque CC. En effet, nous pensons qu'une telle problématique est plus réaliste pour la planification inter-sessions, car l'apprenant n'est pas nécessairement disponible pour pratiquer les CC lorsque l'algorithme le décide.

1.2.3 Critère à optimiser : qu'entend-on par « maîtrise à long terme d'un ensemble de CC » ?

Différents objectifs peuvent être envisagés pour l'espacement adaptatif. Par exemple, REDDY, LEVINE et al. utilisent à la fois la vraisemblance de se rappeler tous les items et l'espérance du nombre d'items correctement mémorisés (REDDY, LEVINE et al., 2017) au cours de l'apprentissage. Dans notre cas, la fonction objectif doit tenir compte de l'incertitude concernant les futurs items à résoudre. En effet, si l'objectif de l'utilisateur est de se préparer à un examen, alors le système devra prendre en compte le fait que l'apprenant devra sûrement répondre à des items sur lesquels il ne s'est jamais entraîné.

C'est pour cette raison que nous formulons le critère objectif au niveau des CC, et non au niveau des items. Notre objectif étant de récompenser la maîtrise à long terme de l'ensemble des CC pendant une période dans le futur, nous définissons notre critère à optimiser comme la probabilité de mettre en œuvre correctement n'importe quelle CC du programme pendant l'intervalle de temps $[\tau_0; T]$ avec, respectivement, τ_0 et T le début et la fin de la période de rétention. Typiquement, τ_0 sera bien supérieur à l'instant actuel t_{actuel} .

1.3 Notations

Dans la suite de cette thèse, nous indexerons les apprenants par $s \in \llbracket 1, S \rrbracket$, les items par $j \in \llbracket 1, J \rrbracket$, les CC par $k \in \llbracket 1, K \rrbracket$, et les instants par t . $Y_{s,j,t} \in \{0, 1\}$ indique si l'apprenant s a eu juste ou non à l'item j à l'instant t : 1 si oui, 0 si non. σ est la fonction logistique, régulièrement utilisée dans le domaine de l'apprentissage automatique (en anglais, *machine learning*) :

$$\forall x \in \mathbb{R}, \sigma(x) = 1/(1 + \exp(-x)). \quad (1.1)$$

Un item peut nécessiter une ou plusieurs CC, et ces relations item-CC sont synthétisées dans une q-matrice binaire (TATSUOKA, 1983) :

$$\forall (j, k) \in \llbracket 1, J \rrbracket \times \llbracket 1, K \rrbracket, q_{jk} = \mathbb{1}_{k \in \text{CC}(j)}. \quad (1.2)$$

$\text{CC}(\cdot)$ prend en entrée un indice d'item j et renvoie l'ensemble des CC qui sont mises en jeu dans la résolution de l'item j .

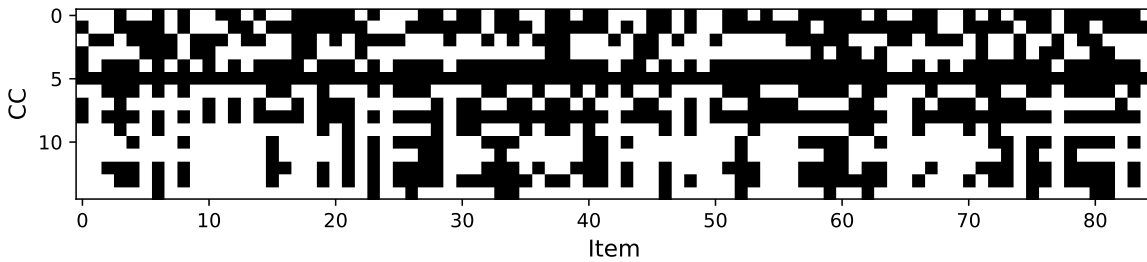


FIGURE 1.3 – Q-matrice liée au jeu de données RoboMission (EFFENBERGER, 2019). La q-matrice est ici transposée dans un souci de mise en page (en lignes, les CC et en colonnes les items). Les intitulés des CC sont donnés plus bas. Un carré noir indique que la CC de la ligne est mise en jeu par l'item de la colonne.

Un exemple de q-matrice est donné dans la figure 1.3. Ce jeu de données concerne une plateforme d'apprentissage de la programmation par blocs. Nous reproduisons ici seulement la q-matrice principale, pas les q-matrices simplifiées du jeu de données (EFFENBERGER, 2019). Voici les intitulés de ses CC :

- | | | |
|---------------------|-------------------------|------------------------|
| 1. <i>teleport</i> | 6. <i>sequences</i> | 11. <i>if</i> |
| 2. <i>collect</i> | 7. <i>while</i> | 12. <i>else</i> |
| 3. <i>obstacle</i> | 8. <i>repeat</i> | 13. <i>test</i> |
| 4. <i>destruct</i> | 9. <i>loop</i> | 14. <i>nested-ctrl</i> |
| 5. <i>lengthlim</i> | 10. <i>nested-loops</i> | 15. <i>compare</i> |

1.4 Contributions

1.4.1 DAS3H : un modèle prédictif et interprétable de l'apprentissage et de l'oubli de CC

Nous avons tout d'abord développé un nouveau modèle prédictif de l'apprenant, que nous avons baptisé DAS3H. DAS3H a la particularité de modéliser l'apprentissage et l'oubli d'un apprenant sur un ensemble de CC, à partir de ses réponses passées à un ensemble d'items. DAS3H s'inscrit à la fois dans la lignée des modèles de l'apprenant à facteurs additifs et dans celle des modèles cognitifs de la mémoire humaine.

DAS3H étend le modèle DASH (LINDSEY et al., 2014) en tenant compte de la structure des relations entre items et CC afin d'améliorer sa prédiction et de tenir compte du transfert de connaissances d'un item à un autre. Plus précisément, DAS3H :

- estime un biais de difficulté par item *et* par CC ;
- estime une courbe d'apprentissage et d'oubli *par* CC et non une seule courbe d'apprentissage et d'oubli pour tous les items ;
- est formulé dans le cadre des *Knowledge Tracing Machines*, ou KTM (VIE et al., 2019), pour enrichir le modèle.

Pour évaluer DAS3H, nous avons comparé ses performances prédictives, mesurées par validation croisée, à quatre autres modèles prédictifs de l'apprenant (IRT/MIRTb, AFM, PFA et DASH) sur cinq jeux de données éducatives. Pour assurer une comparaison juste entre les modèles, nous les avons tous implémentés dans le cadre des KTM et nous avons comparé trois dimensions de plongement vectoriel différentes pour chacun d'entre eux. Nous avons mené en outre des analyses complémentaires pour mieux comprendre ce qui rendait DAS3H plus performant que ses concurrents. Le code Python pour ces expériences a été mis à disposition sur GitHub¹².

Nous avons présenté ce modèle à la conférence EDM 2019 (CHOFFIN, POPINEAU et al., 2019). Cet article a reçu à cette occasion le prix du meilleur article long de la conférence.

1.4.2 Heuristiques d'espacement adaptatif et personnalisé pour la maîtrise à long terme de CC

Pour répondre à notre problématique d'espacement adaptatif de l'apprentissage de CC, nous avons choisi de développer d'abord des stratégies simples et efficaces pour trouver des solutions qui approximent la solution optimale. Nous avons donc proposé et implémenté trois heuristiques pour sélectionner la meilleure CC à faire réviser à un apprenant à un instant t :

- μ -back, qui sélectionne la CC introduite μ semaines auparavant ;
- θ -threshold, qui sélectionne la CC dont la probabilité de bonne réponse est la plus proche d'une valeur fixée en avance θ ;
- Greedy, qui sélectionne la CC dont la révision rapportera le gain de mémorisation à long terme le plus élevé, en espérance.

θ -threshold et Greedy sont deux heuristiques adaptatives d'espacement fondées sur un modèle de l'apprenant ; μ -back est non-adaptative et ne nécessite pas de modèle.

Ces trois heuristiques sélectionnent à chaque instant t de révision la CC optimale (selon elles) à faire réviser. Cependant, nous avons fait l'hypothèse (voir section 1.2.2) dans notre cadre de recherche que les items pouvaient faire travailler plusieurs CC en même temps. Pour tirer parti de cette structure, nous avons introduit une nouvelle procédure de sélection *gloutonne* qui permet de sélectionner, à partir d'une heuristique et son critère de choix de CC, le sous-ensemble de CC le plus prometteur au lieu de la meilleure CC à un instant t . Nous proposons donc une version « multi-CC » pour les heuristiques θ -threshold et Greedy, μ -back

12. <https://github.com/BenoitChoffin/das3h>

n'étant pas compatible avec cette procédure. Nous comparons, de manière qualitative, les différentes caractéristiques de ces heuristiques.

Nous avons ensuite élaboré et implémenté un protocole de simulation de trajectoires d'apprentissage et d'oubli synthétiques pour comparer nos cinq heuristiques entre elles. Ce protocole, qui reproduit la problématique d'un enseignant voulant mettre en place des séances de révision périodiques des CC qu'il introduit au fil de son cours, repose notamment sur le modèle DAS3H (CHOFFIN, POPINEAU et al., 2019). Les performances des différents algorithmes de sélection de CC sont comparées à l'aide de la PMR_R , une métrique décrivant la maîtrise moyenne d'un apprenant sur un ensemble de CC et sur une période future donnée. Nous avons aussi mis à disposition le code Python de ces expériences sur GitHub¹³.

Enfin, nous avons conduit des analyses complémentaires pour évaluer la robustesse et mieux comprendre les performances des heuristiques que nous avons proposées.

1.4.3 Apprentissage par renforcement profond pour l'espace adaptatif de CC

Les stratégies que nous avons proposées et implémentées sont des heuristiques construites à priori. Nous avons également voulu explorer une approche différente pour l'espace adaptatif de l'apprentissage de CC : au lieu de développer en avance la politique de choix de CC, il est aussi possible de créer un algorithme qui apprendrait la politique de sélection optimale en interagissant avec des apprenants (ici, simulés). L'apprentissage par renforcement (profond) fournit des méthodes de choix pour résoudre un tel problème.

Nous avons donc d'abord formulé le problème de l'espace adaptatif et personnalisé de l'apprentissage de CC comme un POMDP (*Partially Observable Markov Decision Process* ou processus de décision markovien partiellement observable), et proposé AC4S, un algorithme de type *Actor-Critic* pour le résoudre. Cet algorithme a la particularité d'utiliser comme observations des représentations abstraites de l'état courant des connaissances de l'apprenant, extraites d'un réseau de neurones récurrent. Ceci nous permet d'étudier l'impact du décalage de modèle (en anglais, *model mismatch*) entre modèle générateur des trajectoires d'apprentissage et d'oubli et modèle utilisé pour l'espace adaptatif.

Nous réutilisons pour ces expériences l'environnement de simulation que nous avons développé auparavant, ainsi que les différentes heuristiques comparées. Nous avons également publié le code Python pour reproduire ces expériences sur GitHub¹⁴.

1.5 Publications

Poster à Rencontres Jeunes Chercheurs en EIAH 2018 Benoît Choffin. « PLUME : un Plugin Moodle pour Générer des Feuilles d'Exercices Personnalisées ». In : *Rencontres Jeunes Chercheurs en EIAH 2018*. 2018. URL : <https://hal.archives-ouvertes.fr/hal-01764376>

Article à MLSP 2018 Benoît Choffin et Naonori Ueda. « Scaling Bayesian Optimization up to Higher Dimensions : a Review and Comparison of Recent Algorithms ». In : *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE. 2018, pp. 1-6.

Présentation à PFIA 2018 Yolaine Bourda, Claude Chaudet, Benoît Choffin, Jeanne Parmentier, Fabrice Popineau et Jill-Jënn Vie. « Tuteurs intelligents : boucler la boucle ». In : *Journée « IA pour l'Éducation », PFIA 2018*. 2018. URL : <https://hal.archives-ouvertes.fr/hal-02197685v1>

¹³. https://github.com/BenoitChoffin/multiskill_adaptive_spacing

¹⁴. <https://github.com/BenoitChoffin/ac4s>

Article long à EDM 2019 Benoît Choffin, Fabrice Popineau, Yolaine Bourda et Jill-Jênn Vie. « DAS3H : Modeling Student Learning and Forgetting for Optimally Scheduling Distributed Practice of Skills ». In : *Proceedings of the Twelfth International Conference on Educational Data Mining (EDM 2019)*. Best Full Paper Award. 2019, pp. 29–38. URL : <https://arxiv.org/abs/1905.06873>.

Article à Paris-Saclay Junior Conference on Data Science and Engineering 2019 Benoît Choffin, Fabrice Popineau, Yolaine Bourda et Jill-Jênn Vie. « DAS3H : Modeling Student Learning and Forgetting for Optimally Scheduling Distributed Practice of Skills ». In : *Paris-Saclay Junior Conference on Data Science and Engineering*. 2019.

Poster à International Congress on Technologies in Education 2019 Benoît Choffin, Alice Latimier et Niluphar Ahmadi. « Making sense of learner behavioral, cognitive and demographic characteristics to improve learner modeling ». In : *International Congress on Technologies in Education 2019*. 2019. URL : <https://hal.archives-ouvertes.fr/hal-02139264>.

Article à ERCIM News 2020 Benoît Choffin, Fabrice Popineau, et Yolaine Bourda. « Modeling Student Learning and Forgetting for Optimally Scheduling Skill Review ». *ERCIM News*, 2020(120), pp. 12-13.

Article invité à LAK 2020 Benoît Choffin, Fabrice Popineau, Yolaine Bourda et Jill-Jênn Vie. « DAS3H : Modeling Student Learning and Forgetting for Optimally Scheduling Distributed Practice of Skills ». Article invité à la 10^e conférence internationale *Learning Analytics and Knowledge (LAK 2020)*.

1.6 Plan

Dans le chapitre 2, nous présentons un état de l’art relatif aux problématiques qui nous intéressent dans cette thèse. Dans un premier temps, nous décrivons des approches issues de deux communautés différentes (fouille de données éducatives et psychologie cognitive) pour la modélisation de l’apprentissage et l’oubli chez l’apprenant. Ensuite, nous faisons une revue de littérature étendue sur les algorithmes d’espacement adaptatif et personnalisé de l’apprentissage.

Dans le chapitre 3, nous proposons un nouveau modèle prédictif de l’apprentissage et l’oubli de CC chez l’apprenant, baptisé DAS3H. Nous comparons DAS3H à quatre autres modèles traditionnels de l’apprenant sur cinq jeux de données différents et montrons qu’il possède une meilleure puissance prédictive que tous ses concurrents.

Dans le chapitre 4, nous proposons tout d’abord un cadre de simulation de trajectoires synthétiques d’apprentissage et d’oubli, fondé sur le modèle DAS3H, pour comparer des politiques d’espacement de l’apprentissage. Nous étendons également trois heuristiques d’espacement adaptatif à notre problématique d’apprentissage de CC, et nous développons une nouvelle procédure gloutonne de sélection de sous-ensemble de CC pour la révision. Nous implémentons ces différents éléments, comparons les heuristiques présentées sur des apprenants synthétiques et présentons nos résultats.

Dans le chapitre 5, nous explorons une nouvelle approche, fondée sur l’apprentissage par renforcement profond, pour l’espacement adaptatif de révisions de CC. Nous proposons un algorithme qui s’inscrit dans cette approche et réutilisons le cadre de simulations introduit au chapitre 4 pour comparer les performances de cet algorithme aux heuristiques de sélection de CC. Nous montrons que cet algorithme est capable d’apprendre une politique de sélection d’item performante, mais au prix d’un grand nombre d’épisodes avant convergence.

Enfin, dans le chapitre 6, nous concluons cette thèse en faisant une synthèse de nos contributions, puis en présentant deux cas d'usage possible pour nos travaux ainsi que des pistes d'amélioration de ces derniers.

Chapitre 2

État de l'art

Comme nous l'avons précisé au chapitre précédent, nous cherchons avec cette thèse à développer des algorithmes d'espacement adaptatif pour la maîtrise à long terme de CC. Il peut être utile, pour la conception de tels algorithmes, de reposer sur un modèle de l'apprenant capable de prédire les performances futures de ce dernier et leur évolution à partir d'un historique d'apprentissage passé. Par exemple, un tel modèle peut servir à anticiper les conséquences de la sélection d'un item à un instant donné sur sa probabilité de rappel à un instant futur, ou bien à déterminer le moment optimal auquel faire réviser un item. Dans ce chapitre, nous commençons donc par présenter différentes approches pour modéliser et prédire les performances d'un apprenant. Nous nous penchons plus précisément sur des modèles qui tiennent compte de la temporalité dans leur prédiction. Certains d'entre eux modélisent explicitement le phénomène d'oubli, d'autres non. Nous montrons en particulier qu'aucun modèle de l'apprenant existant ne tient compte à la fois de l'oubli et des relations multiples entre item et CC. Ensuite, nous décrivons les différents systèmes d'espacement adaptatif et personnalisé de l'apprentissage qui existent dans la littérature. Nous montrons par cette revue de littérature qu'aucun algorithme existant n'est directement adapté à notre problématique d'espacement de l'apprentissage de CC.

2.1 Modélisation de l'apprenant

L'histoire de la littérature scientifique sur la modélisation de l'apprenant est particulièrement riche (DESMARAIS et al., 2012). Nous nous concentrons dans cette section sur le sous-problème de recherche consistant à modéliser l'apprentissage et l'oubli des apprenants à partir de données relatives à leurs performances passées.

Plus précisément, ces données consistent souvent en un ensemble d'apprenants répondant à un ensemble d'items. Le modèle a accès à l'identifiant de l'apprenant et de l'item de chaque interaction, ainsi qu'à la justesse de la réponse de l'apprenant : 0 si l'apprenant a échoué, 1 s'il a réussi. Parfois, quand l'algorithme intègre une notion de séquence ou de temporalité dans les données, il a également accès à la position de l'interaction dans la séquence d'interactions de l'apprenant ou bien tout simplement à l'instant auquel celle-ci a eu lieu.

2.1.1 Approches fondées sur les données pour la modélisation de l'apprenant

Théorie de la réponse à l'item

Le modèle de la théorie de la réponse à l'item (LINDEN et al., 2013), aussi appelé IRT pour *Item Response Theory* en anglais, est un modèle de l'apprenant issu de la psychométrie. Dans sa forme la plus simple, le modèle IRT¹ prédit la probabilité qu'un apprenant s réponde correctement à l'item j de la manière suivante :

1. Alors appelé IRT 1PL, IRT à un paramètre, ou modèle de Rasch.

$$\mathbb{P}(Y_{s,j} = 1) = \sigma(\alpha_s - \delta_j) \quad (2.1)$$

avec :

- s l'identifiant de l'apprenant ;
- j l'identifiant de l'item ;
- $Y_{s,j} \in \{0, 1\}$ la réponse binaire de l'apprenant s sur l'item j ;
- σ la fonction logistique ;
- $\alpha_s \in \mathbb{R}$ le niveau latent de l'apprenant s ;
- $\delta_j \in \mathbb{R}$ la difficulté latente de l'item j .

Les deux paramètres α_s et δ_j sont dits latents car ils ne sont pas directement observables chez l'apprenant : le modèle IRT les extrait en effet à partir d'un ensemble de données. Ces données consistent simplement en les scores binaires des apprenants répondant aux items : aucune notion de séquence ou de temporalité n'est donc prise en compte ici. L'une des principales hypothèses du modèle IRT est en effet que le niveau de l'apprenant est statique et ne peut pas changer avec le temps ou la pratique. C'est d'ailleurs la raison pour laquelle il convient tout particulièrement à des tests standardisés (comme le GMAT ou le GRE), dans lesquels le niveau de l'apprenant n'est pas censé fluctuer d'une réponse à l'autre. Malgré sa simplicité apparente, le modèle IRT s'est révélé être un modèle d'apprenant robuste et fiable, obtenant avec quelques modifications de meilleures performances prédictives que des architectures beaucoup plus complexes telles que *Deep Knowledge Tracing* (WILSON, KARKLIN et al., 2016). Plusieurs fonctions de lien peuvent être utilisées pour passer de la combinaison de facteurs latents à la probabilité de bonne réponse : la fonction logistique (IRT 1PL) ou la fonction de répartition d'une variable aléatoire normale centrée réduite (IRT 1PO).

IRT peut aussi mettre en jeu d'autres paramètres latents : un paramètre de discrimination et un paramètre de chance. Le paramètre de discrimination, propre à chaque item, détermine intuitivement à quel point un item sépare bien les apprenants de niveau inférieur et les apprenants de niveau supérieur à sa difficulté δ_j . Autrement dit, plus le paramètre de discrimination est élevé, plus l'augmentation du niveau d'un apprenant se traduit par une augmentation forte de sa probabilité de bonne réponse à l'item. Le paramètre de chance, quant à lui, s'applique plus à des items pour lesquels l'apprenant peut répondre correctement par chance, comme dans des questionnaires à choix multiple.

Le modèle IRT peut également être étendu pour représenter les biais des apprenants et des items par des vecteurs multidimensionnels plutôt que par des scalaires. Ce modèle s'appelle alors *Multidimensional Item Response Theory* (MIRT) (RECKASE, 2009) :

$$\mathbb{P}(Y_{s,j} = 1) = \sigma(\langle \alpha_s, \mathbf{d}_j \rangle - \delta_j). \quad (2.2)$$

Contrairement à IRT, $\alpha_s \in \mathbb{R}^d$ et $\mathbf{d}_j \in \mathbb{R}^d$ sont ici des vecteurs en dimension d et représentent respectivement les caractéristiques latentes de l'apprenant et de l'item. $\delta_j \in \mathbb{R}$ est toujours le paramètre de difficulté du modèle. L'opérateur $\langle \cdot, \cdot \rangle$ est le produit scalaire canonique entre deux vecteurs.

Séquence et temporalité dans la modélisation de l'apprenant

L'approche *Knowledge Tracing* modélise l'évolution de l'état des connaissances d'un apprenant dans le temps afin de prédire sa future séquence de résultats. Le modèle original (A. T. CORBETT et al., 1994) et toujours le plus répandu de *Knowledge Tracing* est celui de *Bayesian Knowledge Tracing* (BKT). Il est fondé sur un modèle à chaîne de Markov cachée où l'état des connaissances de l'apprenant sur une CC est la variable latente et la maîtrise d'une CC est supposée binaire. Dans ce modèle, un apprenant maîtrise totalement ou pas du tout une CC.

À partir des réponses de l'apprenant sur une CC, le modèle BKT permet de déterminer la probabilité que celui-ci la maîtrise.

Depuis sa création, BKT a été étendu pour pallier ses limitations et tenir compte, par exemple, des différences inter-individuelles entre les apprenants (YUDELSON et al., 2013). Dans (GONZÁLEZ-BRENES, HUANG et al., 2014), les auteurs proposent quant à eux d'étendre le modèle BKT en utilisant des caractéristiques générales pour estimer les paramètres du modèle. Plus récemment, PIECH et al. (2015) ont remplacé le cadre original du modèle à chaîne de Markov cachée par un réseau de neurones récurrent (RNN) et ont proposé un nouveau modèle de *Knowledge Tracing* appelé *Deep Knowledge Tracing* (DKT). Malgré une légère controverse concernant ses performances réelles (WILSON, XIONG et al., 2016; M. KHAJAH et al., 2016), des travaux récents continuent à développer cet axe de recherche (ZHANG et al., 2017; MINN et al., 2018; GHOSH et al., 2020).

Les *modèles à facteurs additifs*² constituent une autre ligne de recherche en modélisation de l'apprenant, qui hérite quant à elle des modèles IRT. Ces modèles tirent leur nom du fait qu'il modélisent la probabilité de bonne réponse de l'apprenant à l'aide d'une somme de caractéristiques décrivant la réponse de l'apprenant sur cet item : par exemple, le nombre de bonnes réponses passées sur des items similaires ou encore les types d'actions pédagogiques rencontrées précédemment. Plus précisément, les modèles à facteurs additifs modélisent la probabilité de bonne réponse de la manière suivante :

$$\mathbb{P}(Y_{s,j} = 1) = \sigma\left(\sum_i \gamma_i m_{s,j,i}\right) \quad (2.3)$$

avec :

- γ_i le paramètre associé à la i -ème caractéristique ;
- $m_{s,j,i}$ la valeur de la caractéristique i décrivant la réponse de l'apprenant s sur l'item j .

Notons que l'on retrouve le modèle IRT 1PL si l'on choisit comme caractéristiques un encodage *one-hot* des identifiants des apprenants et des items.

Au sein de cette ligne, plusieurs modèles tiennent compte de l'historique passé des performances de l'apprenant pour prédire sa probabilité de bonne réponse future et peuvent donc être assimilés au *Knowledge Tracing*. Par exemple, le modèle *Additive Factor Model* (AFM) (CEN et al., 2006; CEN et al., 2008) :

$$\mathbb{P}(Y_{s,j} = 1) = \sigma\left(\alpha_s + \sum_{k \in \text{CC}(j)} \beta_k + \gamma_k a_{s,k}\right) \quad (2.4)$$

avec :

- k identifiant de CC ;
- β_k paramètre de facilité latente de la CC k ;
- $a_{s,k}$ nombre de tentatives de l'apprenant s sur la CC k avant cette tentative ;
- γ_k coefficient associé à ce compteur de tentatives passées, pouvant s'interpréter comme le taux d'apprentissage de la CC k .

Selon AFM, la probabilité de bonne réponse sur un item dépend donc du niveau initial de l'apprenant et de la facilité³ de chacune des CC mises en jeu par l'item j . Mais en pratiquant ces CC, l'apprenant s fait aussi varier cette probabilité : un γ_k positif signifie que la probabilité de bonne réponse d'un apprenant augmente à mesure que celui-ci pratique la CC. À l'inverse, un γ_k négatif implique que la probabilité de bonne réponse de l'apprenant diminue avec la pratique. Plus la valeur absolue de ce coefficient γ_k est élevée, et plus l'effet de la pratique sur la CC k est fort.

2. Nous reprenons ici la dénomination employée par LINDSEY et al. (2014) mais généralisons ce qu'elle recouvre.
3. Ou difficulté : pour passer de l'un à l'autre, il suffit simplement de multiplier le coefficient par -1 .

Le modèle *Performance Factors Analysis* (PFA) (PAVLIK, CEN et al., 2009) s'appuie sur le modèle AFM et décompose la pratique passée en compteurs de réussites et d'échecs sur chacune des CC mises en jeu par l'item, au lieu de simples compteurs d'essais :

$$\mathbb{P}(Y_{s,j} = 1) = \sigma \left(\sum_{k \in \text{CC}(j)} \beta_k + \eta_k c_{s,k} + \rho_k f_{s,k} \right) \quad (2.5)$$

avec :

- $c_{s,k}$ le nombre de bonnes réponses de l'apprenant s sur la CC k avant cette tentative ;
- $f_{s,k}$ nombre de mauvaises réponses de l'apprenant s sur la CC k avant cette tentative ;
- γ_k et ρ_k les coefficients estimés par le modèle et associés à ces compteurs.

On note également que PFA n'estime pas de paramètre spécifique à l'apprenant (comme α_s par exemple) dans cette formulation. PFA module cependant l'effet de la pratique en fonction des réussites et des échecs passés : en effet, on peut raisonnablement imaginer qu'une réussite est un indicateur plus fort de la réussite future sur un item mettant en œuvre la CC qu'un échec.

Plus récemment, le cadre des *Knowledge Tracing Machines* (KTM) (VIE et al., 2019) est venu généraliser plusieurs modèles à facteurs additifs de la littérature en modélisation de l'apprenant. On trouve notamment parmi ces modèles IRT, MIRT, AFM et PFA. Comme les modèles à facteurs additifs, KTM permet à son utilisateur d'utiliser n'importe quel jeu de caractéristiques décrivant les interactions entre un ensemble d'apprenants et un ensemble d'items pour prédire si des interactions futures seront des réussites ou des échecs. Par exemple, ces caractéristiques peuvent être des compteurs de tentatives ou de réussites et d'échecs comme pour AFM et PFA, ou bien les identifiants des apprenants et des items comme pour IRT ou MIRT. Mais la force des KTM est qu'elles peuvent prendre en compte bien d'autres caractéristiques, comme par exemple le temps de réponse ou bien la langue d'origine de l'apprenant⁴.

Les KTM sont fondées sur des *machines à factorisation* et modélisent la probabilité de réponse correcte comme suit :

$$\mathbb{P}(Y_{s,j} = 1) = \psi \left(\mu + \sum_{i=1}^N w_i x_{s,j}^i + \sum_{1 \leq i < \ell \leq N} x_{s,j}^i x_{s,j}^\ell \langle v_i, v_\ell \rangle \right) \quad (2.6)$$

où :

- ψ est une fonction de lien, par exemple la fonction logistique ;
- $\mu \in \mathbb{R}$ est un biais global, estimé par KTM ;
- N est le nombre de caractéristiques du modèle, qu'il s'agisse d'identifiants d'items, de caractéristiques temporelles, etc. ;
- $x_{s,j} \in \mathbb{R}^N$ est un échantillon rassemblant toutes les caractéristiques décrivant l'interaction entre s et j : par exemple l'identifiant de l'apprenant et de l'item, des informations concernant des tentatives antérieures, etc. Chaque composante $x_{s,j,i}$ représente la valeur de la caractéristique i pour cette interaction ;
- $w_i \in \mathbb{R}$ est le biais de la caractéristique i et $v_i \in \mathbb{R}^d$ sa représentation vectorielle continue. Ces éléments sont estimés par KTM.

Chaque caractéristique i prise en compte dans la modélisation possède donc (1) un biais $w_i \in \mathbb{R}$ et (2) sa représentation vectorielle continue multidimensionnelle $v_i \in \mathbb{R}^d$. Ces dernières interagissent par paire grâce à des produits scalaires : KTM modélise donc les interactions entre les variables du modèle, comme par exemple la langue de l'apprenant et l'identifiant d'un item en particulier pour indiquer que certains mots de vocabulaire

4. Ceci peut constituer une variable essentielle pour la modélisation de l'apprentissage d'une langue étrangère (SETTLES, BRUST et al., 2018).

sont plus difficiles ou plus faciles à apprendre selon la langue d'origine de l'apprenant. Les caractéristiques d'un échantillon $x_{s,j}$ sont généralement peu nombreuses, de sorte que cette probabilité peut être calculée efficacement.

Toutefois, on peut voir que les modèles à facteurs additifs présentés plus haut ne tiennent pas compte de la temporalité *précise* des interactions passées de l'apprenant avec les CC : qu'une tentative ait eu lieu il y a une heure ou il y a un mois, l'effet sera le même sur la probabilité de réponse correcte de l'apprenant. De même, VIE et al. utilisent dans leur article des caractéristiques décrivant les tentatives passées d'un apprenant sur les CC mais pas les instants auxquels ces interactions ont eu lieu. Les modèles *PFA-decay* (GONG et al., 2011) et *Recent-PFA* (GALYARDT et al., 2015) s'attaquent à ce problème au sein du modèle PFA en s'appuyant sur une représentation plus sophistiquée de la pratique passée. En effet, ces deux modèles accordent plus de poids aux interactions et résultats passés plus récents. EKANADHAM et al. (2015) font un pas de plus pour tenir compte de la temporalité dans le modèle IRT et étendent le modèle IRT ogive à deux paramètres (modèle 2PO) en modélisant l'évolution du niveau de l'apprenant comme un processus de Wiener. Ils incorporent également dans la structure du modèle des connaissances à priori sur les relations de prérequis entre les CC pour guider l'inférence du modèle.

2.1.2 Approches fondées sur la théorie pour la modélisation de l'oubli

Parallèlement à la littérature sur la modélisation de l'apprenant, une autre ligne de recherche, issue de la psychologie cognitive, est axée spécifiquement sur la modélisation cognitive de la mémoire humaine.

L'un des tout premiers travaux sur la modélisation de la mémoire humaine remonte à EBBINGHAUS (1885), qui a étudié sur lui-même le phénomène d'oubli. Après avoir appris une liste de syllabes par cœur, EBBINGHAUS a mesuré à différents intervalles le temps qu'il gagnait lorsqu'il cherchait à apprendre de nouveau la même liste. Il a alors proposé la formule suivante pour modéliser la décroissance de la trace en mémoire :

$$y = \frac{100k}{(\log \Delta t_{init})^c + k} \quad (2.7)$$

avec :

- y le temps économisé lors du réapprentissage, exprimé en pourcentage du temps d'apprentissage initial ;
- Δt_{init} le temps écoulé depuis l'apprentissage initial ;
- k et c deux constantes.

Ce modèle de l'oubli possède déjà la forme globale que prendront tous les modèles suivants : une décroissance rapide après l'acquisition initiale, puis la courbe s'aplanit progressivement. Depuis ces travaux initiaux, de nombreuses autres relations fonctionnelles ont été proposées pour rendre compte de la décroissance des traces en mémoire. RUBIN et al. (1996) comparent la qualité d'ajustement de 105 modèles différents de la mémoire aux données de 210 expériences en psychologie sur ce sujet. Les modèles comparés ont tous deux paramètres à estimer. Parmi ces modèles, on trouve notamment le *modèle exponentiel*, l'un des modèles les plus utilisés pour décrire le phénomène d'oubli :

$$y = be^{-m\Delta t} \quad (2.8)$$

où :

- y est le pourcentage d'items correctement mémorisés ;
- e la base des logarithmes naturels ;
- Δt le temps écoulé depuis la dernière interaction ;

— b et m deux constantes.

La constante m peut être vue comme la demi-vie de l'item en mémoire, c'est-à-dire qu'ajouter $0,69/m$ unités de temps à t réduit de moitié y . La constante b représente quant à elle le degré d'apprentissage initial : à $t = 0$, c'est le pourcentage d'items correctement mémorisés. Le modèle exponentiel a aussi été utilisée par REDDY, LABUTOV et al. (2016) pour modéliser la probabilité de rappel d'un item dans le futur. Mais dans l'étude de RUBIN et al., les *fonctions puissance*, ou loi de puissance, se sont avérées mieux ajustées aux données des expériences compilées. Elles s'expriment comme suit :

$$y = b\Delta t^{-m} \quad (2.9)$$

avec les mêmes notations qu'au-dessus. Ici, b représente encore le degré d'apprentissage initial et m exprime le rythme de décroissance de la trace en mémoire : plus m est élevée, plus la mémoire décroît vite au fil du temps. Une version alternative à trois paramètres de la loi de puissance (WICKELGREN, 1974), définie en $t = 0$, a ensuite été proposée :

$$y = b(1 + a\Delta t)^{-m} \quad (2.10)$$

Ici, a est un facteur d'échelle, utilisé pour moduler l'échelle temporelle et rendre le modèle utilisable aussi bien avec des secondes qu'avec des mois. Cette formulation a été baptisée loi de puissance de Wickelgren (WIXTED et al., 2007). Elle a été par exemple utilisée avec succès avec quelques variantes par RIDGEWAY et al. (2017) pour modéliser l'oubli sur des données massives de 125000 sujets qui apprenaient l'espagnol à partir du logiciel Rosetta Stone.

Plus récemment, des modèles plus sophistiqués de la mémoire humaine ont été proposés : par exemple *Adaptive Character of Thought-Rational* (ACT-R) (ANDERSON et al., 1997) et *Multiscale Context Model* (MCM) (PASHLER, N. CEPEDA et al., 2009). Dans le modèle ACT-R, une trace mémorielle est ajoutée à chaque fois qu'un item est travaillé : chaque trace décroît selon une loi de puissance. Pour obtenir l'activation d'un item à un moment donné, ACT-R utilise le logarithme d'une combinaison linéaire des différentes traces. L'activation représente l'ancrage de l'item en mémoire : plus l'activation est élevée, et plus l'item sera facile à se remémorer. Ce modèle original a ensuite été étendu (PAVLIK et ANDERSON, 2005) pour tenir compte de l'effet d'espacement : PAVLIK et ANDERSON ajoutent l'hypothèse selon laquelle le rythme de décroissance d'une trace en mémoire dépend du niveau d'activation de l'item au moment de la trace. Si le niveau d'activation est élevé, alors l'item sera certes facile à se rappeler, mais la trace associée à cette révision décroîtra plus vite. MCM modélise quant à lui dès l'origine l'effet d'espacement de l'apprentissage.

Les modèles cognitifs de la mémoire continuent encore à être affinés pour rendre compte de plus de mécanismes de la mémoire. Par exemple, WALSH, GLUCK, GUNZELMANN, T. JASTRZEMBSKI, KRUSMARK et al. (2018) offrent une comparaison de trois modèles computationnels de la mémoire humaine : le modèle de mémoire déclarative ACT-R (PAVLIK et ANDERSON, 2008), le modèle *Predictive Performance Equation* (PPE) (WALSH, GLUCK, GUNZELMANN, T. JASTRZEMBSKI et KRUSMARK, 2018) et une généralisation du modèle *Search of Associative Memory* (SAM) (RAAIJMAKERS, 2003). Ces modèles diffèrent dans la manière dont ils prédisent l'impact de l'espacement sur le réapprentissage ultérieur, après un long intervalle de rétention. PPE est le seul à prédire que l'espacement peut accélérer le réapprentissage ultérieur ("*spacing accelerated relearning*"), un effet qui a été empiriquement souligné par leur expérience. PPE a également montré une meilleure adéquation aux données expérimentales, comparé à SAM et à ACT-R.

L'avantage de ces modèles issus de la psychologie est qu'ils sont directement interprétables en termes cognitifs et qu'ils incorporent explicitement des éléments théoriques établis. Cependant, ce sont des modèles qui sont souvent utilisés sur des données expérimentales agrégées, au niveau des individus ou au niveau des items (ou même au niveau des deux). Ils peuvent être utilisés (MOZER et al., 2016) pour prédire la probabilité de rappel d'un individu sur un ensemble d'items, la probabilité de rappel d'un ensemble d'individus sur un seul item,

mais il est difficile de les utiliser pour prédire la probabilité de rappel d'un individu sur un seul item, car cela nécessiterait d'estimer trop de paramètres. Le fait que l'on dispose rarement de beaucoup d'échantillons de données d'un individu sur un même item complique encore plus cette estimation. En outre, les entrées de ces modèles consistent souvent en des réponses d'apprenants à des items : ces modèles n'ont pas nécessairement la flexibilité nécessaire pour intégrer d'autres types de caractéristiques, comme par exemple la langue de l'apprenant. Certains de ces modèles, comme le modèle PPE (SENSE, T. S. JASTRZEMBSKI et al., 2019), font toutefois exception et peuvent être utilisés en pratique pour planifier des révisions.

2.1.3 Approches hybrides

Plusieurs modèles de l'apprenant tirent profit des deux approches citées plus haut pour construire des modèles plus riches. Le modèle DASH (LINDSEY et al., 2014) s'appuie ainsi sur deux types de modélisations de l'apprenant : les modèles à facteurs additifs (PAVLIK, CEN et al., 2009; CEN et al., 2006) et les modèles cognitifs de la mémoire humaine MCM (PASHLER, N. CEPEDA et al., 2009) et ACT-R (ANDERSON et al., 1997). DASH signifie *Difficulty, Ability, and Student History*. Plus précisément, LINDSEY et al. modélisent la probabilité qu'un apprenant s réponde correctement à l'item j à l'instant t de la manière suivante :

$$\mathbb{P}(Y_{s,j,t} = 1) = \sigma(\alpha_s - \delta_j + h_\theta(t_{s,j,1:l}, y_{s,j,1:l-1})) \quad (2.11)$$

avec :

- $Y_{s,j,t} \in \{0, 1\}$ la justesse de la réponse de l'apprenant s sur l'item j à l'instant t ;
- $t_{s,j,1:l}$ les moments des interactions passées de s avec l'item j (incluant le moment de l'interaction courante);
- $y_{s,j,1:l-1}$ les résultats binaires de ces interactions (sauf, bien entendu, le résultat de l'interaction courante);
- h_θ une fonction qui synthétise l'effet des $l - 1$ tentatives précédentes de l'apprenant s avec l'item j et les résultats binaires de ces tentatives. C'est plus précisément cette fonction qui est inspirée d'ACT-R et MCM.

Les coefficients α_s , δ_j et θ sont appris par le modèle sur les historiques d'interactions d'un ensemble d'apprenants avec un ensemble d'items. DASH requiert donc les instants auxquels ont lieu chacune des réponses d'un ensemble d'apprenants sur un ensemble d'items, en plus des résultats de ces interactions.

LINDSEY et al. formulent principalement la fonction h_θ de la manière suivante :

$$h_\theta(t_{s,j,1:l}, y_{s,j,1:l-1}) = \sum_{w=0}^{W-1} \theta_{2w+1} \log(1 + c_{s,j,w}) - \theta_{2w+2} \log(1 + a_{s,j,w}) \quad (2.12)$$

avec :

- w indexant un ensemble de W fenêtres temporelles de taille croissante;
- $\theta \in \mathbb{R}^{2W}$ vecteur de coefficients des compteurs de bonnes réponses et de tentatives;
- $c_{s,j,w}$ le nombre de bonnes réponses de l'apprenant s sur l'item j dans la fenêtre temporelle w ;
- $a_{s,j,w}$ le nombre de tentatives de l'apprenant s sur l'item j dans la fenêtre temporelle w .

Comme PFA, DASH sépare la représentation de l'historique de l'apprenant entre tentatives et réussites (et donc, entre réussites et échecs) et permet donc à ces deux compteurs d'avoir un impact différent sur la probabilité future de bonne réponse. L'application du logarithme sur ces compteurs $c_{s,j,w}$ et $a_{s,j,w}$ est inspirée par ACT-R et signifie que la pratique d'un apprenant sur un même item au sein d'une même fenêtre temporelle aura des rendements positifs, mais

décroissants, sur sa probabilité de bonne réponse future. Il est alors nécessaire d'ajouter un 1 à ces compteurs pour éviter d'évaluer les logarithmes en 0 ; cela permet également de garder les logarithmes strictement positifs.

Par ailleurs, les fenêtres temporelles w ne sont pas disjointes et s'étendent sur des intervalles de temps croissants. Elles permettent à DASH de prendre en compte à la fois les processus d'apprentissage et d'oubli et de moduler ses prédictions en fonction de la distribution temporelle des interactions passées entre l'apprenant et l'item.

On peut voir que LINDSEY et al. utilisent le cadre des modèles à facteurs additifs pour prendre en compte l'oubli et les bénéfices de la pratique passée. Leur modèle surpasse en performance prédictive le modèle IRT et un modèle de référence sur leur jeu de données COLT⁵, avec comme métrique l'erreur de prédiction cumulative. Pour éviter le surapprentissage et faciliter l'entraînement du modèle, ils utilisent une régularisation bayésienne hiérarchique.

SETTLES et MEEDER (2016) proposent quant à eux une extension du modèle de *courbe d'oubli exponentielle* avec leur modèle de régression demi-vie. La première différence avec le modèle exponentiel classique est qu'ils modélisent la probabilité de rappel de la manière suivante :

$$\mathbb{P}(Y_{s,j,t} = 1) = 2^{-\frac{\Delta t}{h}} \quad (2.13)$$

h est ici l'inverse du paramètre m dans le modèle exponentiel exposé plus haut. Mais la principale différence avec ce dernier est qu'ils paramétrisent h et le modélisent de la manière suivante :

$$\hat{h}_{\Theta} = 2^{\Theta \cdot x} \quad (2.14)$$

avec :

- Θ un vecteur de coefficients estimés ;
- x un vecteur de caractéristiques (dans leur cas, des compteurs de réussites et d'échecs sur l'item et des indicatrices de lexèmes associés à l'item).

On voit ici que SETTLES et MEEDER prennent un modèle classique de la littérature sur la modélisation de la mémoire et l'améliorent par l'utilisation de caractéristiques comme les compteurs de tentatives. Cela leur permet notamment d'intégrer les effets des révisions successives dans la forme du modèle.

De manière similaire, RIDGEWAY et al. (2017) utilisent une approche hybride en individualisant en fonction de l'apprenant et de son historique d'apprentissage les paramètres d'apprentissage initial b et d'oubli m d'une loi de puissance. Pour cela, RIDGEWAY et al. modélisent chacun de ces deux paramètres comme une combinaison linéaire de caractéristiques décrivant la pratique passée de l'apprenant :

$$b(x_{s,j,t}) = \sum_{i=1}^N w_i x_{s,j,t}^i \quad (2.15)$$

$$m(x_{s,j,t}) = \sum_{i=1}^N v_i x_{s,j,t}^i \quad (2.16)$$

avec :

- $x_{s,j,t} \in \mathbb{R}^N$ un vecteur de caractéristiques décrivant l'historique de la pratique passée d'un apprenant s sur un item j à un instant t ;
- N le nombre de caractéristiques considérées ;
- $w \in \mathbb{R}^N$ et $v \in \mathbb{R}^N$ les deux vecteurs de coefficients correspondant respectivement à b et m .

5. Ces données consistaient en des interactions élève-item sur un système de révisions de mots de vocabulaire espagnol à base de *flashcards*.

RIDGEWAY et al. montrent que ce modèle alternatif obtient de meilleures performances prédictives sur leurs données que le modèle de loi de puissance original. Cependant, le modèle qui individualise à la fois α et β n'obtient pas de performances significativement meilleures que le modèle qui individualise seulement α .

Les trois articles précédents sont des exemples d'utilisation conjointe de modèles issus de la psychologie cognitive et d'une approche fondée sur les données qui parviennent à tirer profit de ces deux traditions pour améliorer les modèles existants. L'une des raisons pour lesquelles ils obtiennent ces performances accrues provient de l'individualisation poussée de leurs prédictions en fonction des caractéristiques de chaque apprenant, et notamment de son historique d'apprentissage.

2.1.4 Limites des modèles exposés

Comme nous l'avons exposé dans cette section, plusieurs modèles intègrent l'oubli de manière explicite ou motivée sur le plan cognitif tandis que d'autres tiennent compte des relations entre items et CC. Cependant, aucun ne prend en compte *ces deux dimensions à la fois*.

EKANADHAM et al. (2015) introduisent la temporalité dans le modèle IRT mais ne modélisent pas explicitement le phénomène d'oubli et ne permettent pas à un item de mettre en jeu plusieurs CC en même temps. Le modèle de *Bayesian Knowledge Tracing* original ne prend en compte que la séquence des interactions et ne considère pas le temps écoulé, par exemple. Ceci peut convenir pour la modélisation de séquences d'activités sur une période courte, mais peut s'avérer problématique quand ces séquences s'étendent sur des périodes plus longues. Pour pallier ce problème, M. KHAJAH et al. (2016) ont par exemple utilisé un modèle de *Bayesian Knowledge Tracing* qui permet à la probabilité d'oublier une CC d'être strictement positive. Cependant, dans ce cas, l'oubli ne dépend pas du temps écoulé entre deux interactions avec une CC mais du nombre total d'interactions (incluant potentiellement des interactions avec d'autres CC) entre deux interactions avec une même CC. Dans (GONZÁLEZ-BRENES et MOSTOW, 2013), les auteurs introduisent un modèle de *Knowledge Tracing* qui permet aux items de dépendre de plusieurs CC en même temps; l'oubli pourrait en théorie se produire mais ici aussi, le temps est considéré discret. Dans (QIU et al., 2011), les auteurs étendent le modèle de *Bayesian Knowledge Tracing* en permettant à la probabilité d'oubli de changer d'un jour à l'autre. Cependant, ils ne tiennent pas compte de multiples CC. Dans (NAGATANI et al., 2019), les auteurs étendent le modèle de *Deep Knowledge Tracing* pour tenir compte de l'oubli humain en y incorporant le temps écoulé entre deux interactions et le nombre de tentatives passées. Néanmoins, leur modèle n'est pas directement adapté à des items mettant en jeu plus d'une CC à la fois. Leur prise en compte de l'oubli est en outre assez rudimentaire et ne tient pas compte des apports fournis par la psychologie cognitive en la matière.

LINDSEY et al. rassemblent des items similaires⁶ en groupes homogènes, qu'ils appellent « composantes de connaissance ». Ceci revient à supposer un transfert total de connaissance entre deux items appartenant à la même CC. Toutefois, ils ne laissent pas la possibilité à un item de mettre en jeu plusieurs CC en même temps.

2.2 Systèmes d'espace adaptatif et personnalisé de l'apprentissage

Les systèmes d'espace adaptatif et personnalisé de l'apprentissage sont des outils servant à améliorer la mémorisation à long terme d'un ensemble de connaissances chez des apprenants, par la planification minutieuse des révisions de chacune de ces connaissances. Plus précisément, ces algorithmes cherchent à présenter à chaque apprenant la séquence de révisions qui lui sera la plus bénéfique pour sa mémorisation à long terme, en tenant

6. Par exemple, ils rassemblent au sein d'un même groupe d'items les différentes conjugaisons d'un même verbe ou la version masculine, féminine ou au pluriel d'un nom.

compte de ses besoins. Ces besoins sont le plus souvent inférés par l'algorithme à partir de l'historique d'apprentissage de l'apprenant (tentatives précédentes de rappel des items à apprendre, résultats de ces tentatives de rappel, instants auxquels ces révisions ont eu lieu, temps de réponse, etc.). Nous qualifions ces systèmes d'*adaptatifs* et *personnalisés* car ils prennent en compte, en temps réel, les performances des apprenants pour modifier et proposer un planning de révisions plus adapté à leurs besoins.

Dans la plupart des algorithmes, chacune de ces révisions consiste pour l'apprenant à répondre à une question que lui pose le système d'espacement : il s'agit d'une forme de *rappel indicé* de l'élément de connaissance, l'indice étant la question posée par le système. Les algorithmes d'espacement adaptatif de l'apprentissage utilisent donc deux stratégies d'apprentissage efficace bien connues de la littérature en sciences cognitives : la *répétition espacée* car ils espacent de manière optimale les révisions successives d'une même connaissance et la *récupération en mémoire* puisque chaque révision demande à l'apprenant de faire l'effort de se remémorer l'élément en question plutôt que de lire directement la réponse.

Historiquement, ces algorithmes se sont concentrés sur la planification de révisions pour des éléments simples de connaissance : par exemple, des mots de vocabulaire dans une langue étrangère (PIMSLEUR, 1967; PAVLIK et ANDERSON, 2008; METZLER-BADDELEY et al., 2009; LINDSEY et al., 2014; MOZER et al., 2016), la localisation de pays africains sur une carte (METTLER et al., 2016) ou l'apprentissage de faits historiques. Pour représenter les connaissances à retenir, on a souvent employé le concept de *flashcards* : voir le chapitre 1 pour une définition des *flashcards*. D'abord physiques (LEITNER, 1972), les *flashcards* électroniques se sont développées (WOZNIAK et al., 1994) avec l'avènement de l'enseignement assisté par ordinateur dans la deuxième partie du 20^e siècle. Les *flashcards* électroniques présentent plusieurs avantages pratiques⁷, mais surtout, elles permettent d'utiliser des algorithmes d'espacement adaptatifs plus fins qui optimiseront la séquence de révisions pour une mémorisation prolongée. De nos jours, plusieurs logiciels d'espacement adaptatif de l'apprentissage sont disponibles au grand public : par exemple, Anki⁸, SuperMemo⁹ et Mnemosyne¹⁰.

Dans cette sous-section, nous faisons une revue de la littérature des travaux concernant l'espacement adaptatif et personnalisé de l'apprentissage. Pour simplifier la tâche de notre lecteur, nous synthétisons cet état de l'art dans la table 2.1. La colonne « Choix par l'algorithme » indique si l'algorithme choisit l'instant auquel faire réviser un item ou bien l'item à proposer à un moment donné.

7. Il est en effet facile de les partager entre plusieurs utilisateurs ou de les modifier, et des applications comme Anki proposent même à leurs utilisateurs de les rendre publiques.

8. <https://apps.ankiweb.net/>

9. <https://www.supermemo.com/>

10. <https://mnemosyne-proj.org/>

Nom	Type d'algorithme	Introduit par	Utilisé par	Modèle	Choix par l'algorithme	Horizon temporel	Testé sur apprenants réels
\emptyset	Score de priorité	PAVLIK et ANDERSON, 2008	PAVLIK et ANDERSON, 2008; VAN RIJN et al., 2009; SENSE, BEHRENS et al., 2016	Avec	Instant/item	Intra-session	Oui
θ - <i>threshold</i>	Score de priorité	M. M. KHAJAH et al., 2014	M. M. KHAJAH et al., 2014; LINDSEY et al., 2014; MOZER et al., 2016; REDDY, LEVINE et al., 2017; HUNZIKER et al., 2019; TABIBIAN et al., 2019	Avec	Item	Intra/inter-sessions	Oui
\emptyset	Optimisation myope	ATKINSON, 1972	ATKINSON, 1972; PAVLIK et ANDERSON, 2008; METTLER et al., 2011	Avec	Item	Intra-session	Oui
<i>Greedy</i>	Optimisation myope	HUNZIKER et al., 2019	HUNZIKER et al., 2019; YANG et al., 2020	Avec	Item	Intra/inter-sessions	Oui
<i>Memorize</i>	Contrôle optimal	TABIBIAN et al., 2019	TABIBIAN et al., 2019; UPADHYAY et al., 2018	Avec	Instant	Intra/inter-sessions	Non
TADS	APR	YANG et al., 2020	YANG et al., 2020	Avec	Item	Intra/inter-sessions	Non
TRPO	APR	REDDY, LEVINE et al., 2017	REDDY, LEVINE et al., 2017	Sans	Item	Intra-session	Non
TPPRL	APR	UPADHYAY et al., 2018	UPADHYAY et al., 2018	Sans	Instant	Intra/inter-sessions	Non
ARTS	Score de priorité	METTLER et al., 2011	METTLER et al., 2011; METTLER et al., 2016	Sans	Item	Intra-session	Oui
Système de Leitner	Heuristique à base de règles	LEITNER, 1972	LEITNER, 1972; REDDY, LEVINE et al., 2017; YANG et al., 2020	Sans	Item	Intra/inter-sessions	Oui
<i>SuperMemo</i>	Heuristique à base de règles	WOZNIAK et al., 1994	REDDY, LEVINE et al., 2017; WOZNIAK et al., 1994; METZLER-BADDELEY et al., 2009; YANG et al., 2020	Sans	Instant	Intra/inter-sessions	Oui

TABLE 2.1 – Tableau comparatif des caractéristiques des différents algorithmes d'espace adaptatif décrits dans la section 2.2, triés par ordre d'introduction dans l'état de l'art. Nous n'incluons pas dans ce tableau les algorithmes non-adaptatifs. APR signifie « Apprentissage par renforcement ».

En suivant l'exemple de DOROUDI (2019) dans sa thèse de doctorat, nous divisons les travaux de recherche sur les algorithmes d'espacement adaptatif de l'apprentissage en deux grandes catégories : les algorithmes *fondés sur un modèle* et les algorithmes *sans modèle*. Néanmoins, d'autres caractéristiques distinguent les systèmes d'espacement adaptatif et personnalisé que nous décrivons plus bas, aussi avons-nous indiqué ces différentes caractéristiques dans la table synthétique. Enfin, nous faisons le choix d'inclure dans cet état de l'art des algorithmes non-adaptatifs mais qui ont eu une influence sur cet axe de recherche.

Algorithmes fondés sur un modèle

Parmi les algorithmes fondés sur un modèle, une option fréquemment utilisée consiste à choisir l'item pour lequel un score de priorité de révision est le plus proche d'une valeur donnée θ . La plupart de ces algorithmes nécessitent de pouvoir modéliser statistiquement la mémoire humaine. Par exemple, PAVLIK et ANDERSON (2008) utilisent une version étendue du modèle de mémoire déclarative ACT-R pour construire un planificateur adaptatif d'optimisation de révisions d'items dans un temps limité, au cours d'une session de révision. Le modèle ACT-R est à l'origine capable de prédire la réussite d'un apprenant sur un item et la vitesse de rappel en tenant compte du temps écoulé depuis la dernière interaction et de la fréquence de la pratique. PAVLIK et ANDERSON (2008) étendent ACT-R pour qu'il capture notamment l'effet d'espacement. Le planificateur adaptatif qu'ils proposent utilise l'estimation par le modèle ACT-R du gain de rétention en mémoire par unité de temps au moment du test final pour décider du moment où il est optimal de présenter une paire de mots à un apprenant.

VAN RIJN et al. (2009) comparent quant à eux quatre algorithmes, dont l'algorithme de PAVLIK et ANDERSON (2008) et deux versions de celui-ci qui utilisent la performance individuelle de l'apprenant et le temps de réponse pour améliorer encore la personnalisation et planifier les révisions de 20 paires de mots français-néerlandais au cours d'une seule session d'apprentissage. Leur algorithme plus sophistiqué, utilisant le temps de réponse, a surpassé deux autres algorithmes comparés. SENSE, BEHRENS et al. (2016) utilisent également cet algorithme dans leurs expériences.

M. M. KHAJAH et al. (2014) utilisent des simulations computationnelles fondées sur deux modèles cognitifs de la mémoire humaine (PAVLIK et ANDERSON, 2008; PASHLER, N. CEPEDA et al., 2009) pour comparer l'efficacité sur la mémoire à long terme des apprenants de deux heuristiques de sélection d'items : μ -back et θ -threshold. Ils simulent un environnement d'apprentissage avec K CC, introduites séquentiellement à une semaine d'intervalle; chaque semaine, l'apprenant révise une CC précédemment introduite en essayant de s'en souvenir correctement. La stratégie de rappel μ -back sélectionne l'item qui a été vu μ semaines auparavant et la stratégie θ -threshold, l'item pour lequel la probabilité de rappel correct est la plus proche d'une valeur fixe θ . Ils soutiennent que cette heuristique est cohérente avec la notion de « difficultés désirables » de l'apprentissage humain introduite par BJORK (1994) car elle cherche à faire réviser l'item qui est sur le point d'être oublié par l'apprenant, c'est-à-dire celui dont l'ancrage en mémoire est le plus proche d'une valeur donnée. Ils ont découvert qu'une stratégie "2-back", c'est-à-dire qui fait réviser aux apprenants l'item qui a été présenté deux semaines auparavant, obtient de bonnes performances et peut être suffisante en l'absence d'outils technologiques plus sophistiqués. Par exemple, une telle stratégie, non-adaptative, pourrait être très utile aux enseignants. Cependant, ils ont aussi constaté qu'une politique θ -threshold bien paramétrée (avec θ autour de 0,4) permettait d'obtenir des performances proches de l'optimum pour les apprenants. Ce résultat suggère qu'avec l'accès aux données des apprenants, un algorithme d'espacement adaptatif permettrait aux apprenants d'obtenir une meilleure rétention en mémoire que des plannings fixes de révision. Néanmoins, l'heuristique θ -threshold de M. M. KHAJAH et al. exige d'avoir un modèle d'apprenant fiable pour pouvoir obtenir la probabilité de rappel sur chacun des items. M. M. KHAJAH et al. utilisent

quant à eux leur modèle simulateur¹¹ pour cela, ce qui ne serait pas possible en situation réelle, où le modèle de l'apprenant nous est inconnu. De plus, dans cet article, les auteurs font l'hypothèse que toutes les courbes d'oubli des CC sont les mêmes au sein d'un même processus d'apprentissage simulé, ce qui pourrait potentiellement sous-estimer les bénéfices de l'utilisation de technologies adaptatives à la place de planificateurs d'espacement génériques.

Cette stratégie θ -*threshold* a aussi été utilisée avec succès dans une expérience en classe sur 179 élèves de 4^e d'un collège américain par LINDSEY et al. (2014) et MOZER et al. (2016). L'objectif était d'évaluer pendant un semestre entier l'efficacité d'un planificateur adaptatif et personnalisé de révisions de mots de vocabulaire en espagnol lors de séances hebdomadaires. À chaque fois qu'il devait choisir un item à faire réviser à un apprenant, ce planificateur sélectionnait celui dont la probabilité de rappel estimée¹² était la plus proche de 0,33. Il s'agit donc de l'heuristique θ -*threshold*, avec $\theta = 0,33$. Les planificateurs alternatifs utilisés étaient le planificateur fixe "1-back" (révision des mots de vocabulaire appris la semaine passée) et un planificateur massé (révision des mots de vocabulaire introduits la semaine courante). L'ensemble d'items consistait en 409 paires de mots (ou phrases) anglais-espagnol, rassemblées en 221 CC. Lors d'un examen cumulatif 28 jours après la fin du semestre, le planificateur personnalisé, fondé sur DASH, a permis une amélioration significative de la rétention des mots de 10 % par rapport au planificateur fixe et de 16,5 % par rapport au planificateur massé.

Certains travaux dans cet axe de recherche et qui s'appuient sur des modèles de l'apprenant choisissent plutôt d'utiliser de l'optimisation myope, ce qui signifie qu'ils ne prennent en compte que les conséquences attendues de leur prochaine sélection d'items sur la mesure de performance qu'ils tentent d'optimiser. C'est le cas pour ATKINSON (1972) qui optimise le nombre d'items connus une semaine après la fin de la session d'apprentissage en choisissant à chaque étape l'item qui est le plus susceptible d'être appris définitivement après la révision. Lors d'une expérience impliquant 120 sujets et 84 paires de mots allemand-anglais, ATKINSON compare à deux stratégies de référence deux versions de sa modélisation de l'état de connaissance de l'apprenant et de ses dynamiques : celle qui considère des paramètres différents pour chaque item produit la meilleure rétention une semaine après la session d'apprentissage. Le modèle de Markov utilisé ici considère trois états possibles pour un item : connu définitivement, connu temporairement et inconnu.

Plus récemment, HUNZIKER et al. (2019) proposent un algorithme de sélection d'items glouton fondé sur l'optimisation discrète pour la planification de révisions de *flashcards*. En se fondant sur l'historique de pratique de l'apprenant (c'est-à-dire quels items ont été révisés par le passé, à quel moment et pour quel résultat) l'algorithme choisit de manière itérative l'item¹³, qui générera en espérance le gain de probabilité de bonne réponse le plus élevé sur tous les concepts¹⁴ et sur tous les instants futurs jusqu'à la fin de la période d'apprentissage. Intuitivement, cela revient à choisir l'item dont la révision augmentera le plus la rétention sur tout le reste de la période d'apprentissage si jamais c'était le dernier item à être révisé. Une telle décision requiert l'utilisation d'un modèle de l'apprentissage et de l'oubli de l'apprenant pour déduire l'impact futur du choix à l'instant t de n'importe quel concept. En plus de fournir des garanties de performance pour leur algorithme, HUNZIKER et al. le testent sur des apprenants simulés et des sujets humains recrutés sur Mechanical Turk. Bien que la fonction qu'ils cherchent à optimiser ne tienne pas compte de ce qui se passe après la période d'apprentissage, HUNZIKER et al. évaluent dans leurs expériences la mémorisation de leurs apprenants à des temps postérieurs.

TABIBIAN et al. (2019) abordent le problème de la répétition espacée adaptative et per-

11. Le modèle utilisé pour la simulation de trajectoires d'apprenants et celui utilisé par θ -*threshold* dans ces simulations est donc rigoureusement le même.

12. DASH était le modèle employé pour cette prédiction.

13. Ils parlent dans leur article de concept.

14. En théorie donc, il est possible que la révision d'un item influence la mémorisation d'autres items qui lui sont proches.

sonnalisée d'une manière différente. Ils le formalisent en effet dans le cadre des processus ponctuels temporels marqués comme un problème de contrôle optimal stochastique. Les dynamiques des probabilités de rappel et des taux d'oubli pour chaque item sont quant à elles modélisées par des équations différentielles stochastiques (EDSs) avec sauts. Bien qu'applicable en théorie à n'importe quel modèle d'apprenant, leur algorithme, baptisé *Memorize*, nécessite de modéliser explicitement l'oubli pour pouvoir dériver analytiquement la solution optimale du problème de contrôle optimal : TABIBIAN et al. le font pour trois modèles cognitifs de la mémoire humaine, dont le modèle de courbe d'oubli exponentielle (EBBINGHAUS, 1885). Intuitivement, *Memorize* cherche à maximiser la probabilité de rappel de l'apprenant¹⁵, selon le modèle, au cours de l'apprentissage de l'apprenant, tout en limitant le nombre de révisions d'un même item. Il renvoie alors pour chaque item le prochain moment optimal de révision. Enfin, ils comparent leur algorithme à un algorithme fondé sur un seuil (LINDSEY et al., 2014) et à une stratégie de planification uniforme de révisions sur des données publiées par Duolingo (SETTLES et MEEDER, 2016).

Comme les méthodes d'apprentissage par renforcement ont été utilisées depuis les débuts des systèmes de tutorat intelligent (DOROUDI, ALEVEN et al., 2019), elles ont également été appliquées récemment pour planifier de manière optimale les révisions dans des algorithmes d'espacement adaptatif de l'apprentissage humain (SINHA, 2019). Ainsi, YANG et al. (2020) ont proposé un nouvel algorithme d'apprentissage par renforcement profond fondé sur un modèle, baptisé TADS pour *Time-Aware scheduler with Dyna-Style planning*. Ce dernier a la particularité d'utiliser un modèle de la mémoire de l'apprenant¹⁶ et un modèle prédictif du prochain moment où l'apprenant demandera à réviser pour simuler des trajectoires de révision et améliorer l'efficacité de l'échantillonnage. Dans leur article, ces deux modèles sont des LSTM (HOCHREITER et al., 1997). TADS laisse le choix à l'apprenant du moment où il révisera la prochaine fois et est comparé sur des données simulées à plusieurs politiques de planification de révisions incluant d'autres méthodes d'apprentissage par renforcement.

Algorithmes sans modèle

La plupart des travaux utilisant des méthodes d'apprentissage par renforcement pour la planification de révisions de *flashcards* ne s'appuient pas sur un modèle de l'apprenant sous-jacent. REDDY, LEVINE et al. (2017) utilisent une architecture par renforcement profond pour aborder le problème de l'espacement adaptatif de l'apprentissage. Ce dernier est alors formulé comme un processus de décision markovien partiellement observable (POMDP). Leur algorithme est sans modèle et a seulement accès à l'identifiant de l'item présenté juste avant, si l'apprenant a réussi ou non à se le remémorer, et le temps écoulé entre cette dernière révision et l'instant courant. Ils testent leur algorithme sur des apprenants simulés, avec trois modèles de la mémoire humaine : le modèle d'oubli exponentiel (REDDY, LABUTOV et al., 2016), la régression demi-vie (SETTLES et MEEDER, 2016) et un modèle de loi de puissance généralisée (MOZER et al., 2016). Ils le comparent à plusieurs références, dont une variante de l'algorithme *SuperMemo* (WOZNIAC et al., 1994) et l'heuristique θ -*threshold* de LINDSEY et al. Leurs résultats suggèrent que l'apprentissage par renforcement profond pour la planification de révisions de *flashcards* est une méthode suffisamment flexible et performante pour concurrencer des heuristiques répandues comme *SuperMemo*. Cependant, l'algorithme proposé reçoit à chaque itération une récompense dont le calcul requiert la connaissance des probabilités de rappel immédiates sur chacun des items ; il serait donc difficilement utilisable en l'état avec de vrais apprenants.

S'inspirant de TABIBIAN et al., UPADHYAY et al. (2018) formalisent un nouveau problème d'apprentissage par renforcement profond dans lequel les actions de l'agent et ce que renvoie

15. Dans leurs expérimentations, ils évaluent bien entendu les performances des différents algorithmes comparés avec une métrique plus réaliste, celle du taux d'oubli empirique.

16. Ce modèle-ci sert également à fournir des pseudo-récompenses à l'algorithme d'apprentissage par renforcement.

l'environnement sont des événements stochastiques et asynchrones en temps continu. Ces événements sont alors modélisés par les auteurs par des processus ponctuels temporels marqués. UPADHYAY et al. appliquent la méthode qu'ils ont développée pour résoudre ce type de problème à la planification optimale de révisions de *flashcards*, sur des données simulées. L'un des avantages de leur algorithme est qu'il est indépendant du modèle spécifique de l'apprenant ; il suppose cependant que l'apprenant peut pratiquer les items à n'importe quel moment.

Nous avons décrit plus haut différents algorithmes d'espacement adaptatif de l'apprentissage humain fondés sur des modèles et qui utilisent un score de priorité pour planifier les révisions successives des items. Des algorithmes similaires, ceux-ci sans modèle prédictif de l'apprenant, existent également dans la littérature. Ainsi, METTLER et al. (2016) comparent un planificateur d'espacement adaptatif (METTLER et al., 2011) (appelé ARTS pour *Adaptive Response-Time-based Sequencing*) à deux conditions d'espacement fixe de l'apprentissage. ARTS utilise les temps de réponse, les performances et le nombre d'essais des apprenants pour calculer dynamiquement un score de priorité pour la planification adaptative de la pratique des items : à chaque révision, l'algorithme compare les scores de priorité de tous les items et choisit celui qui est le plus élevé pour le présenter à l'apprenant. Ce score est calculé de telle sorte que :

- les items que l'apprenant n'arrive pas à se remémorer soient représentés après un court délai, pour éviter qu'ils soient toujours présents dans sa mémoire de travail ;
- les items correctement remémorés soient présentés de nouveau après un délai qui est fonction du temps de réponse sur l'item.

Le temps de réponse est en effet ici utilisé comme un indicateur de la difficulté de récupération en mémoire et donc de l'ancrage en mémoire. Leurs résultats montrent que leur algorithme ARTS produit non seulement une meilleure rétention lors d'un test avec délai mais qu'en plus il améliore les temps de réponse lors de ce test.

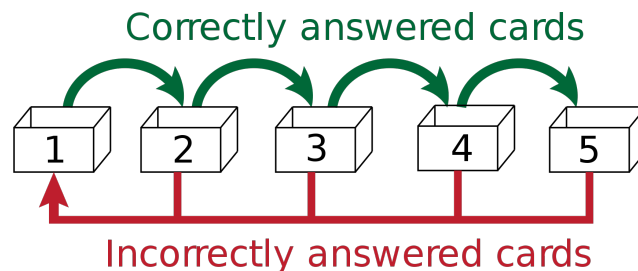


FIGURE 2.1 – Schéma du fonctionnement du système de Leitner (LEITNER, 1972), tiré de la page Wikipedia du système de Leitner.

On peut également trouver des stratégies plus simples et sans modèle dans la littérature. C'est par exemple le cas du système de Leitner (LEITNER, 1972), l'un des premiers algorithmes d'espacement adaptatif de l'apprentissage. La figure 2.1 montre le fonctionnement du système de Leitner. Dans ce système, il existe différentes boîtes, chacune d'entre elles se voyant attribuer une fréquence de révision différente (par exemple, tous les jours, tous les deux jours,...). Lorsqu'un nouvel item (une nouvelle *flashcard*) entre dans le système, il est automatiquement placé dans la boîte la plus fréquemment révisée, c'est-à-dire la première. À chaque session de révision, l'apprenant se voit présenter séquentiellement toutes les *flashcards* qu'il doit réviser ce jour-là. S'il ne parvient pas à se rappeler l'item, ce dernier est placé dans la boîte du dessous¹⁷

17. Dans certains cas, il est placé dans la boîte la plus fréquemment révisée après l'échec du rappel, quelle que soit la boîte dans laquelle il se trouvait.

(plus fréquemment révisée) et s'il parvient à se le rappeler, l'item est placé dans la boîte du dessus (moins fréquemment révisée). À la fin, si l'apprenant se souvient correctement d'un item qui se trouvait auparavant dans la boîte la moins fréquemment révisée, la *flashcard* est considérée comme maîtrisée et quitte le système de Leitner. Celui-ci peut être considéré comme un prototype des futurs algorithmes d'espacement adaptatif, car il adapte bien au fil du temps la fréquence de révision de chaque item en fonction des performances de l'apprenant, présentant moins souvent (respectivement, plus souvent) les items que l'apprenant réussit (échoue) à se rappeler. Une version mathématiquement formalisée du système de Leitner a depuis lors été proposée (REDDY, LABUTOV et al., 2016) : dans cet article, REDDY, LABUTOV et al. se placent dans un régime où le nombre d'items à retenir est très élevé (comme dans l'apprentissage de mots de vocabulaire d'une langue étrangère), modélisent le système de Leitner comme un réseau de files d'attente (une pour chaque paquet de cartes) et utilisent une heuristique pour approximer le planning de révisions optimal.

Notons également que les systèmes de Leitner et de Pimsleur, bien qu'étant conçus à l'origine sans modèle de l'apprenant sous-jacent, ont depuis été ramenés à des cas particuliers du modèle de *régression demi-vie* de SETTLES et MEEDER (2016). De la même façon, TABIBIAN et al. (2019) ramènent le système de Leitner à un cas particulier de leur algorithme *Memorize*.

Plus récemment que le système de Leitner, WOZNIAK et al. (1994) ont développé l'algorithme d'espacement adaptatif *SuperMemo*. Cet algorithme, fonctionnant avec un système de règles posées à priori, a connu différentes versions au fil des années et est aujourd'hui commercialisé¹⁸. Dans la version décrite dans (WOZNIAK et al., 1994), l'algorithme calcule l'intervalle optimal avant la prochaine révision de telle sorte qu'une petite partie seulement (5 %) des items appris soit oubliée¹⁹. Ce calcul est aussi modulé par la difficulté de rappel déclarée des items par l'apprenant. *SuperMemo* est utilisé dans Anki²⁰ et Mnemosyne²¹ et a également montré de meilleures performances en termes de mémorisation qu'un planificateur aléatoire dans une expérience impliquant de vrais sujets humains (METZLER-BADDELEY et al., 2009). La figure 2.2 montre une capture d'écran de la plateforme Anki : on peut voir ici que la difficulté de rappel déclarée par l'apprenant détermine l'intervalle d'espacement.



FIGURE 2.2 – Capture d'écran venant de la plateforme d'apprentissage de *flashcards* Anki

Néanmoins, le manque de garanties de performance théoriques et parfois empiriques parmi les algorithmes d'espacement adaptatif fondés sur des règles posées à priori a depuis

18. <https://www.supermemo.com/>

19. On voit donc ici que l'algorithme *SuperMemo* se rapproche des algorithmes avec un score de priorité, décrits plus haut.

20. <https://faqs.ankiweb.net/what-spaced-repetition-algorithm.html>

21. <https://mnemosyne-proj.org/help/memory-research>

lors soulevé quelques inquiétudes (TABIBIAN et al., 2019) et a motivé une formalisation accrue de la problématique d'espace adaptatif de l'apprentissage.

2.2.1 Algorithmes non-adaptatifs

Certains algorithmes d'espace de l'apprentissage ne sont pas adaptatifs ni personnalisés, bien qu'ils soient régulièrement cités dans la littérature sur le sujet. Nous avons donc choisi de leur accorder une place dans cette sous-section. C'est le cas par exemple de la méthode de Pimsleur (PIMSLEUR, 1967) : ce dernier propose dans cet article de faire des « rappels à intervalles gradués » pour planifier les révisions ultérieures d'un même item dans un contexte d'apprentissage de mots de vocabulaire. Ceci consiste à augmenter de manière exponentielle les intervalles entre les révisions successives d'un item. La toute première révision doit être planifiée quand la probabilité de réponse correcte de l'item atteint un seuil (60 % dans son article) et chaque intervalle suivant est calculé en multipliant le précédent par la durée du tout premier intervalle. Par exemple, si l'intervalle entre l'acquisition initiale et la 1^{re} révision dure 5 secondes, alors celui entre la 1^{re} et la 2^e durera $5^2 = 25$ secondes et le suivant $5^3 = 125$ secondes, et ainsi de suite. PIMSLEUR préfigure donc ici la stratégie θ -threshold qui sera utilisée plus tard par LINDSEY et al. (2014) mais on voit qu'il ne s'adapte pas dynamiquement aux performances et aux besoins de l'apprenant : tout au plus recommande-t-il de moduler la durée du tout premier intervalle en fonction de la difficulté de l'item. La méthode de Pimsleur a depuis été commercialisée²² mais nous ne savons pas si elle a évolué par rapport aux travaux initiaux de son créateur.

NOVIKOFF et al. (2012), quant à eux, formalisent mathématiquement et de manière abstraite le problème de l'espace de révisions. Ils étudient la faisabilité et l'efficacité de séquences de révisions quand la k -ème révision optimale d'un même item par un apprenant doit avoir lieu entre a_k et b_k temps discrets après sa $k - 1$ -ème révision. Ils supposent également que le planning de révisions est expansif, c'est-à-dire que l'intervalle entre deux révisions croît avec le nombre de révisions :

$$\forall k, a_k \leq a_{k+1} \text{ et } b_k \leq b_{k+1}. \quad (2.17)$$

Enfin, une macro Excel²³ utilisant une suite de Fibonacci²⁴ pour espacer automatiquement les CC à aborder et générer des quiz a été rendue publique pour aider les enseignants à planifier leur programme. Cette macro n'a cependant pas, à notre connaissance, été testée lors d'une expérience contrôlée.

2.2.2 Limites des algorithmes exposés

L'objectif de cette thèse est de développer des algorithmes d'espace adaptatif et personnalisé de l'apprentissage de CC et non plus seulement de *flashcards*. Comme nous l'avons vu dans l'état de l'art plus haut, à notre connaissance, tous les algorithmes d'espace adaptatif de la littérature s'appliquent à l'apprentissage de connaissances simples, qui peuvent être contenues dans un unique item. Il s'agit souvent de *paires* : paires de mots langue d'origine/langue étrangère, paires de noms de pays/localisations sur une carte, etc. En outre, certains travaux évoquent parfois la possibilité de regrouper plusieurs items au sein d'une seule CC (M. M. KHAJAH et al., 2014; LINDSEY et al., 2014) ou que la révision d'un item puisse aider à la mémorisation d'items proches (HUNZIKER et al., 2019) mais aucun n'aborde spécifiquement la question de la planification adaptative de révisions lorsqu'un item *unique* peut mettre en jeu *plusieurs* CC différentes et que l'objectif est de faire maîtriser sur le long terme ces CC. Pourtant, ce type de situation est commun dans des situations éducatives réelles, comme dans l'apprentissage des langues ou en mathématiques.

22. <https://www.pimsleur.com/the-pimsleur-method>

23. <https://zenodo.org/record/573764>

24. <https://www.learningscientists.org/blog/2017/5/11-1>

La différence entre notre problématique et les travaux précédents est donc double :

- l'objectif est de faire maîtriser aux apprenants un ensemble de CC *par la pratique* d'un ensemble d'items ;
- chaque item peut mettre en jeu plusieurs CC en même temps.

Nous illustrons ces différences avec la figure 2.3, qui schématise notre problématique.

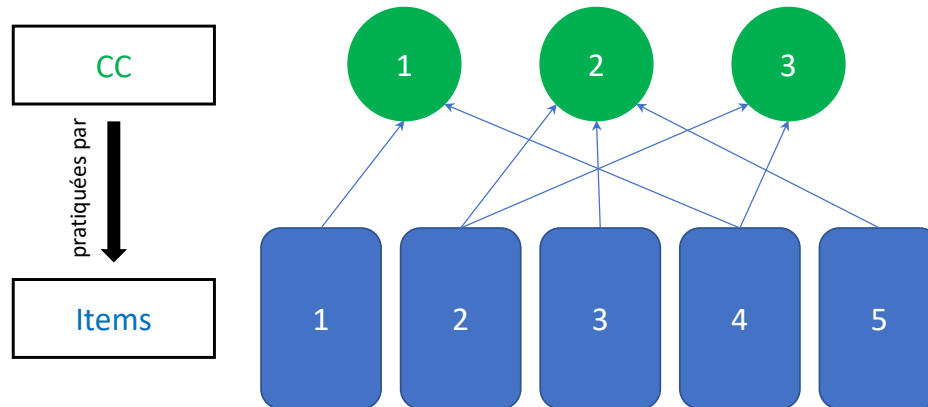


FIGURE 2.3 – Passage d'algorithmes d'espacement adaptatif d'items à celui de CC

Les algorithmes que l'on trouve dans la littérature ne sont pas directement exploitables pour notre problématique car leur objectif est de faire apprendre un ensemble d'*items*, symbolisés par les rectangles bleus du bas dans la figure 2.3. Pour nous, les *items* sont utilisés par les algorithmes pour faire pratiquer un ensemble de CC, symbolisées par les ronds verts du haut : il n'est pas possible de faire réviser directement ces CC car elles sont abstraites. Dans un item de mathématiques par exemple, l'objectif n'est pas de maîtriser *cet* item-là en particulier, mais bien de pouvoir transférer sa connaissance à d'autres items similaires, c'est-à-dire des items qui partagent une ou plusieurs CC avec celui-ci. À l'inverse, les systèmes d'espacement adaptatifs traditionnels cherchent à faire apprendre et retenir les items particuliers aux apprenants.

En outre, dans notre cadre de recherche, la pratique d'un item peut faire réviser plusieurs CC en même temps : il est alors possible de tenir compte des relations entre items et CC pour améliorer les performances des algorithmes d'espacement. HUNZIKER et al. (2019) sont les plus proches de nos travaux de thèse car ils évoquent la possibilité pour la révision d'un item d'influencer la maîtrise d'autres items connexes. Cependant, ils ne testent pas leurs algorithmes dans un tel régime.

2.3 Optimisation de séquences d'apprentissage dans les systèmes tutoriels intelligents

Cette thèse s'inscrit aussi plus généralement dans le cadre de l'optimisation et la personnalisation automatiques des séquences d'apprentissage proposées aux apprenants. RAFFERTY et al. (2011) formulent ce problème comme un problème de planification sur un processus de décision markovien partiellement observable (POMDP). WHITEHILL et al. (2018) étendent ces travaux mais utilisent une architecture de contrôle hiérarchique pour choisir les actions pédagogiques optimales. CLEMENT et al. (2015) utilisent des bandits manchots pour choisir la meilleure séquence d'activités pédagogiques à fournir à l'apprenant. Ils testent leurs deux algorithmes, RiARiT et ZPDES sur 400 élèves entre 7 et 8 ans. A. S. LAN et al. (2016) utilisent

quant à eux des bandits contextuels pour choisir la meilleure activité pédagogique en utilisant une estimation du profil de connaissances de l'apprenant. Plusieurs systèmes tutoriels intelligents (ITS pour *Intelligent Tutoring Systems* en anglais) utilisent la méthode de *mastery learning* (A. CORBETT, 2000 ; DOROUDI, 2020) au sein du cadre du *Knowledge Tracing* (A. T. CORBETT et al., 1994) : il s'agit de faire travailler les apprenants sur une CC en particulier jusqu'à ce que le système infère que l'apprenant maîtrise la CC avec une certaine probabilité (souvent 0,95). L'apprenant passe ensuite à une autre CC.

Le *machine teaching* s'attache à trouver le jeu de données d'entraînement optimal²⁵ pour faire apprendre approximativement les paramètres d'un modèle donné θ^* à un modèle élève. À l'origine étudié dans le cadre de l'apprentissage automatique (*machine learning* en anglais), plusieurs travaux ont formalisé l'optimisation de l'apprentissage humain comme un problème de *machine teaching* (ZHU, 2015 ; ZHU et al., 2018). Si l'on possède un bon modèle cognitif de l'apprenant sur une tâche on peut alors en théorie optimiser les exemples à lui présenter pour l'entraîner à cette tâche. La tâche peut par exemple être l'apprentissage d'une langue étrangère et les exemples à présenter, différentes activités pédagogiques. Par exemple, MAC AODHA et al. (2018) développent un algorithme pour entraîner des humains à reconnaître des images de plusieurs catégories visuelles (différentes espèces de papillons, différents caractères chinois écrits à la main). Cet algorithme choisit les meilleurs exemples à présenter aux apprenants et leur fournit également automatiquement un retour correctif qui met en valeur la partie de l'image la plus importante pour la classification.

Toutefois, la spécificité de nos travaux par rapport à cette ligne de recherche plus générale est que notre objectif est d'améliorer la rétention à long terme sur un ensemble de connaissances : la métrique que l'on cherche à optimiser se focalise donc sur la durabilité de l'apprentissage, pas sur la performance à court terme ou la rapidité de la maîtrise. Pour atteindre ce but, nous utilisons deux stratégies d'apprentissage efficace issues de la littérature en psychologie cognitive : la répétition espacée et la récupération en mémoire. La temporalité et non pas seulement la séquence des activités, est donc essentielle, car la littérature sur l'espacement (N. J. CEPEDA, VUL et al., 2008) montre l'importance de l'emplacement d'une révision sur la rétention à long terme.

25. Une mesure de l'optimalité peut être par exemple la taille de ce jeu : il s'agit alors de trouver le jeu de données minimal qui permette à l'élève d'apprendre le modèle désiré.

Chapitre 3

DAS3H : un modèle prédictif et interprétable de l'apprentissage et l'oubli de CC

3.1 Introduction

Nous avons décrit dans le chapitre 2 les différents systèmes d'espacement adaptatif et personnalisé pour l'apprentissage de *flashcards* de la littérature : il y apparaît que ces algorithmes peuvent se scinder en deux grandes catégories, les algorithmes *avec* ou *sans* modèle de l'apprenant. L'approche avec modèle est utilisée dans de nombreux travaux car un modèle de l'apprenant permet notamment d'inférer la probabilité de rappel courante de l'apprenant sur un item (LINDSEY et al., 2014) et donc de sélectionner un item sur le point d'être oublié. Un modèle offre en outre la possibilité d'évaluer les conséquences attendues de la sélection d'un item sur l'évolution future de cette probabilité (HUNZIKER et al., 2019; TABIBIAN et al., 2019) pour optimiser un planning de révisions.

Pour commencer, nous avons choisi de développer un algorithme *avec* modèle pour notre problématique de l'espacement personnalisé des révisions de CC. Or, la plupart des modèles de l'apprenant que l'on peut trouver dans la littérature (voir chapitre 2) soit (1) intègrent les relations entre items et CC dans leur structure, soit (2) modélisent l'oubli, mais à notre connaissance, aucun ne fait les deux à la fois. Il nous est alors nécessaire de développer un modèle de l'apprenant qui prenne ces deux dimensions en compte pour pouvoir l'intégrer ensuite dans un algorithme d'espacement adaptatif de l'apprentissage de CC.

Dans ce chapitre, nous nous intéressons donc à notre première série de questions de recherche (cf. chapitre 1) :

Comment modéliser l'évolution temporelle de la maîtrise d'un apprenant sur un ensemble de CC à partir de ses réponses passées sur un ensemble d'items ? La prise en compte du phénomène d'oubli au niveau des CC améliore-t-elle significativement les performances prédictives du modèle ?

Dans ce chapitre, nous présentons notre modèle DAS3H, un modèle prédictif et interprétable de l'apprentissage et de l'oubli de CC que nous avons développé. Dans un premier temps, nous détaillons les limites du modèle DASH (LINDSEY et al., 2014) sur lequel s'appuie DAS3H, et expliquons pourquoi ces limites le rendent inadapté à notre problématique. Ensuite, nous décrivons notre modèle DAS3H et son fonctionnement. DAS3H étend le modèle DASH et s'inscrit dans la tradition des modèles à facteurs additifs de la littérature en fouille de données éducatives. Il inclut une représentation de la distribution temporelle de la pratique passée sur les CC mises en jeu par un item pour tenir compte de l'oubli et des bénéfices de la pratique. En particulier, DAS3H permet aux courbes d'apprentissage et d'oubli de varier d'une CC à une autre. Puis, nous présentons notre méthodologie expérimentale pour comparer notre modèle

DAS3H à quatre modèles de l'apprenant concurrents, sur cinq jeux de données éducatives réelles. Nous présentons nos résultats expérimentaux et montrons que DAS3H obtient de meilleures performances prédictives que tous ses concurrents, sur l'intégralité des jeux de données considérés. Enfin, nous explorons dans la discussion différentes questions pour mieux comprendre nos résultats expérimentaux. Nos résultats suggèrent que l'incorporation des relations items-CC et de l'effet d'oubli dans la structure du modèle améliore les performances de celui-ci, plus que si l'on avait considéré seulement l'une ou l'autre de ces informations.

3.2 Limites du modèle DASH

Notre modèle s'intitule DAS3H pour *item Difficulty, student Ability, Skill and Student Skill practice History*. Il nous a été inspiré par le modèle DASH (LINDSEY et al., 2014), décrit dans le chapitre 2.

Le modèle DASH présente deux limites principales que nous essayons de dépasser :

Limite 1 LINDSEY et al. font mention dans leur article des composantes de connaissance mais ils utilisent ce concept pour regrouper ensemble des mots similaires (par exemple, toutes les conjugaisons d'un même verbe). Ainsi, dans leur situation, un item possède une et une seule CC; des items différents peuvent appartenir à la même CC s'ils sont assez proches sémantiquement. Leur modèle DASH ne peut pas s'appliquer à des données pour lesquelles un item met en jeu plus d'une seule CC, ce qui est pourtant souvent le cas dans l'apprentissage d'autres tâches en langues ou dans d'autres disciplines, comme les mathématiques. Ainsi, dans le jeu de données *Algebra 2005-2006* (STAMPER et al., 2010) de la *KDD Cup 2010* concernant l'apprentissage de l'algèbre, les items ont en moyenne 1,363 étiquettes de CC.

Limite 2 En outre, LINDSEY et al. supposent que l'impact de la pratique passée sur la probabilité de bonne réponse ne change pas d'une CC à l'autre : en effet, DASH possède seulement deux biais par fenêtre temporelle w , θ_{2w+1} pour les réussites passées et θ_{2w+2} pour les tentatives passées. Cela signifie que le modèle estimera nécessairement que chaque item s'apprend et s'oublie au même rythme que les autres, puisque les coefficients θ_w qui décrivent les courbes d'apprentissage et d'oubli sont partagés entre tous les items. Cela peut être une hypothèse pertinente pour prévenir le surapprentissage du modèle quand le nombre de CC est élevé car cela réduit grandement le nombre de paramètres à estimer. Néanmoins, cela peut aussi gêner du sous-apprentissage et affecter les performances prédictives du modèle si l'ensemble de CC est suffisamment divers et peu homogène pour justifier l'estimation d'une courbe d'apprentissage et d'oubli par CC.

3.3 Description du modèle DAS3H

Dans cette section, nous décrivons les caractéristiques de notre modèle DAS3H ainsi que sa formulation mathématique.

3.3.1 Caractéristiques de DAS3H

Le modèle DAS3H que nous avons développé s'attaque à ces deux problèmes à la fois en intégrant dans sa structure les relations entre l'ensemble d'items et l'ensemble de CC, sous la forme d'une q-matrice (TATSUOKA, 1983). Dans la formulation originale de DASH, l'estimation des paramètres θ_w pour chaque item aurait probablement entraîné le surapprentissage du modèle, compte tenu du nombre d'items individuels que peuvent avoir certains systèmes d'apprentissage en ligne (des dizaines de milliers pour ASSISTments, par exemple). Mais il est plus facile d'estimer θ_w pour chaque CC, car il y en a généralement un plus petit nombre

(quelques centaines seulement pour ASSISTments, par exemple). Ainsi, DAS3H permet de faire varier l'influence de la distribution temporelle de la pratique passée et des résultats binaires de ces tentatives d'une CC à l'autre. Concrètement, cela signifie que l'on autorise le modèle à estimer des courbes d'apprentissage et d'oubli différentes pour chaque CC. En outre, DAS3H se démarque également de DASH en autorisant un item à mettre en jeu plusieurs CC à la fois. On pourrait avancer que nous pourrions regrouper chaque combinaison existante de CC en une CC distincte pour éviter le traitement de CC multiples. Cependant, cette solution ne serait pas satisfaisante car le modèle qui en résulterait ne pourrait pas, par exemple, capturer les similarités entre deux items qui ont en commun toutes les CC sauf une. L'utilisation d'une représentation des CC multiples permet de tenir compte du transfert de connaissances d'un item à l'autre (DESMARAIS et al., 2012).

Enfin, nous avons utilisé le cadre des *Knowledge Tracing Machines* (VIE et al., 2019), décrit dans le chapitre 2, pour DAS3H afin d'enrichir le modèle en estimant pour chaque caractéristique (apprenant, item, CC, etc.) une représentation vectorielle continue (autrement appelée *plongement vectoriel*) en dimension $d \geq 0$. Ces représentations vectorielles continues permettent de modéliser les interactions deux à deux entre chaque paire de caractéristiques prises en compte dans DAS3H par le biais de produits scalaires. Jusqu'à présent, et à notre connaissance, les KTM n'ont pas été utilisées avec des variables décrivant la mémoire humaine. Nous justifions ce choix également par le fait que les machines à factorisation (*Factorization Machines* en anglais), type de modèle à la racine des KTM, sont appropriées quand la matrice X décrivant les caractéristiques du jeu de données est particulièrement creuse (RENDLE, 2012), ce qui est le cas ici. En effet, des modèles d'apprenant comme AFM, PFA, ou encore IRT, peuvent tous être estimés en calculant pour chaque échantillon un vecteur creux qui sera une nouvelle représentation plus commode des données X initiales. Plus de détails sur le calcul de tels vecteurs creux sont disponibles dans l'article présentant les KTM (VIE et al., 2019).

En somme, nous avons étendu le modèle DASH de trois manières :

- Gestion des situations où un item peut mettre en jeu plusieurs CC à la fois : pour cela, nous avons modifié le module temporel h_θ en permettant à l'influence de la pratique passée d'avoir un impact différent sur les performances futures selon les CC ;
- Estimation de biais latents de difficulté (ou facilité) pour *chaque* item j et CC k ;
- Utilisation du cadre des *Knowledge Tracing Machines* (VIE et al., 2019) à la place d'une simple régression logistique pour enrichir le modèle.

3.3.2 Formulation mathématique de DAS3H

Dans sa formulation la plus simple (avec une dimension de $d = 0$ pour la représentation vectorielle continue), le terme quadratique des KTM s'efface et DAS3H se lit comme suit :

$$\mathbb{P}(Y_{s,j,t} = 1) = \sigma \left(\alpha_s - \delta_j + \sum_{k \in \text{CC}(j)} \beta_k + h_\theta(t_{s,j,t}, y_{s,j,t}) \right). \quad (3.1)$$

Ainsi, la probabilité de réussite $Y_{s,j,t}$ de l'apprenant s sur l'item j à l'instant t dépend de son aptitude latente α_s ¹, de la difficulté latente de l'item δ_j et de la somme des biais latents de facilité β_k des CC mises en jeu par l'item j . Plus α_s ou β_k sont élevés, ou plus δ_j est faible, plus la probabilité de bonne réponse augmente. En revanche, puisque $\beta_k \in \mathbb{R}$, l'addition de β_k supplémentaires ne diminue la probabilité de bonne réponse que si $\beta_k < 0$: un item qui met en jeu plus de CC qu'un autre ne sera pas nécessairement plus difficile à résoudre. Tout dépend de la difficulté des CC supplémentaires.

1. α_s peut donc être vu comme le niveau initial de l'apprenant, en l'absence de toute pratique sur les CC du programme.

Cette probabilité dépend également de la distribution temporelle et des résultats de la pratique passée de l'apprenant s , synthétisés par h_θ . C'est ce module temporel qui est en charge de modéliser l'apprentissage et l'oubli des CC à partir de l'historique des résultats de l'apprenant. Nous inspirant de LINDSEY et al., nous choisissons :

$$h_\theta(\mathbf{t}_{s,j,t}, \mathbf{y}_{s,j,t}) = \sum_{k \in \text{CC}(j)} \sum_{w=0}^{W-1} \theta_{k,2w+1} \log(1 + c_{s,k,w}) + \theta_{k,2w+2} \log(1 + a_{s,k,w}). \quad (3.2)$$

avec :

- $\mathbf{t}_{s,j,t}$ les temps auxquels ont eu lieu les interactions passées de s avec les CC mises en jeu par j . Ceci inclut en particulier l'instant courant t ;
- $\mathbf{y}_{s,j,t}$ les résultats binaires de ces interactions. Ceci n'inclut bien entendu pas le résultat binaire $Y_{s,j,t}$ puisque c'est ce qu'on cherche à inférer;
- $w \in \llbracket 0, W-1 \rrbracket$ les indices des fenêtres temporelles;
- $c_{s,k,w}$ le nombre de fois où l'apprenant s a par le passé répondu correctement à un item mettant en jeu la CC k dans la fenêtre temporelle w ;
- $a_{s,k,w}$ le nombre de tentatives passées par s sur la CC k dans la fenêtre temporelle w ;
- $\theta \in \mathbb{R}^{K \times 2W}$ la matrice de coefficients associés aux compteurs de réussites et de tentatives. Ces coefficients décrivent la courbe d'apprentissage et d'oubli d'une CC k .

Penchons-nous maintenant plus précisément sur le fonctionnement du module temporel h_θ , en charge de la modélisation de l'apprentissage et l'oubli. DAS3H estime une courbe d'apprentissage et d'oubli par CC : ce sont les coefficients $\theta_{k,w}$ qui décrivent l'évolution de la probabilité de bonne réponse en fonction de la pratique (instants des interactions et résultats) d'un apprenant. Le module temporel h_θ combine ensuite ces courbes en les sommant. Intuitivement, h_θ peut être interprété comme une somme de traces mémorielles, une pour chaque CC k mise en jeu par l'item j .

Comme pour DASH (LINDSEY et al., 2014), nous ajoutons aux compteurs $a_{s,k,w}$ et $c_{s,k,w}$ un 1 pour éviter que les logarithmes soient évalués en 0. La présence de logarithmes implique que la pratique d'une même CC au sein d'une fenêtre temporelle w a un impact décroissant sur la probabilité de bonne réponse sur un item impliquant cette CC.

Les fenêtres temporelles w sont également de taille croissante et sont toutes ancrées en l'instant t . Prenons un exemple simple pour montrer comment elles fonctionnent : nous nous appuyerons sur la figure 3.1 pour notre explication. Fixons l'ensemble de fenêtres temporelles à $\{7, 10\}$ (en jours). La fenêtre temporelle à 7 jours est symbolisée sur le schéma par la barre orange et celle à 10 jours par la barre verte. Supposons que l'apprenant s rencontre la CC k pour la première fois au temps $t_0 = 0$. Cette tentative est symbolisée par la croix noire sur le schéma. Si nous souhaitons prédire si s répondra correctement à un item mettant en jeu k trois jours après l'avoir rencontrée (à $t = 3$, donc), toutes les fenêtres temporelles sont actives, car la tentative initiale s'est produite trois jours auparavant et est donc incluse dans les intervalles $[t_{\text{courant}} - 7; t_{\text{courant}}]$ et $[t_{\text{courant}} - 10; t_{\text{courant}}]$. C'est le cas présenté dans le deuxième schéma de la figure 3.1. Six jours encore après ($t_{\text{courant}} = 9$), si l'on souhaite faire la même prédiction, l'unique tentative de s sur k sera seulement comptée dans la fenêtre temporelle à 10 jours. En effet, le schéma montre que la tentative en $t = 0$ n'est plus comprise dans la première fenêtre temporelle puisque k a été vue plus de 7 jours auparavant. Avec la pratique sur une même CC, les compteurs de tentatives sont incrémentés mais alors que le temps passe, les tentatives sont progressivement retirées des différentes fenêtres temporelles glissantes.

Maintenant que l'on a précisé le fonctionnement des fenêtres temporelles w , nous pouvons détailler le rôle de θ dans la modélisation. La figure 3.2 représente l'impact conjoint des $\theta_{k,2w+1}$ et des $\theta_{k,2w+2}$ sur la probabilité d'une future réponse correcte, en supposant que pour la CC k

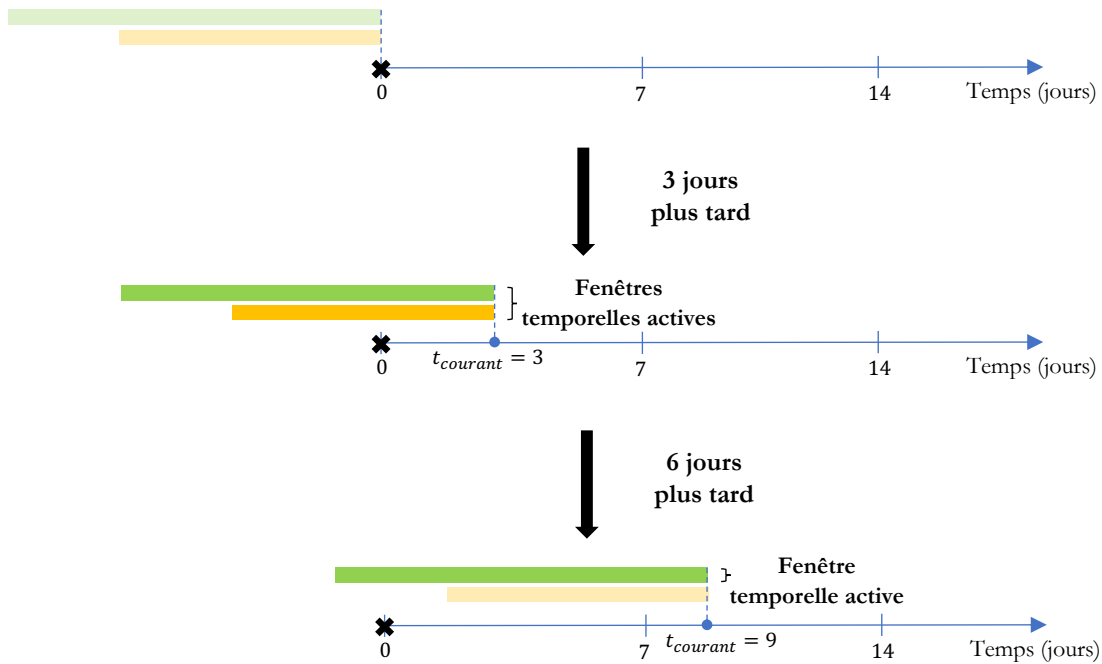


FIGURE 3.1 – Schéma de fonctionnement des fenêtres temporelles du module h_θ dans les modèles DASH (LINDSEY et al., 2014) et DAS3H

tous les $\theta_{k,w}$ sont échantillonnés dans la même zone de la figure 3.2. On voit par ce diagramme que comme pour le modèle DASH, le module temporel h_θ de notre modèle DAS3H est assez expressif pour rendre compte d'une grande diversité de situations d'apprentissage et d'oubli :

- Dans la zone 1, $\theta_{k,2w+1}$ et $\theta_{k,2w+2}$ sont tous deux positifs et les chances de réponse correcte future sur la CC k augmentent avec le nombre d'échecs et de réussites dans la fenêtre temporelle w , mais l'augmentation marginale de ces chances est plus importante pour les réussites. Cela signifie qu'une bonne réponse augmente plus la probabilité de bonne réponse future qu'un échec sur la CC k .
- Dans la zone 2, $\theta_{k,2w+1}$ et $\theta_{k,2w+2}$ sont tous deux négatifs et les chances de réussite diminuent avec le nombre d'échecs et de réussites dans la fenêtre temporelle w , mais la diminution marginale est plus importante (en valeur absolue) pour les réussites. Cela signifie qu'une bonne réponse diminue plus la probabilité de bonne réponse future qu'une mauvaise réponse sur la CC k .
- Dans la zone 3, les échecs diminuent les chances de bonne réponse future, ainsi que les réussites mais la diminution marginale est alors plus faible en valeur absolue. Cela signifie qu'une bonne réponse diminue moins la probabilité de bonne réponse future qu'une mauvaise réponse sur la CC k .
- Dans la zone 4, les échecs diminuent les chances de bonne réponse future, mais les réussites augmentent ces chances.
- Dans la zone 5, les échecs augmentent les chances de bonne réponse future, mais les réussites diminuent ces chances.
- Enfin, dans la zone 6, les échecs et les réussites augmentent les chances de bonne réponse future, mais l'augmentation marginale est plus faible pour les réussites. Cela signifie qu'une bonne réponse augmente moins la probabilité de bonne réponse future qu'une mauvaise réponse sur la CC k .

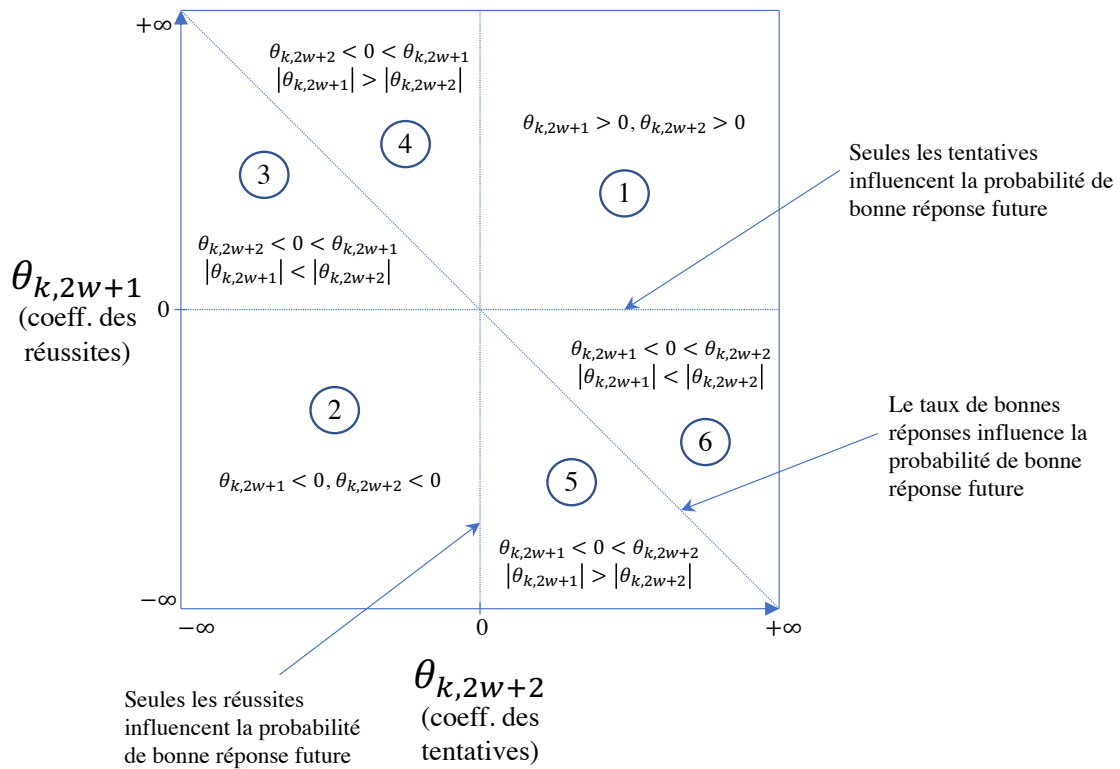


FIGURE 3.2 – Diagramme montrant l’impact conjoint des $\theta_{k,2w+1}$ (coefficients des réussites) et des $\theta_{k,2w+2}$ (coefficients des tentatives) sur la probabilité de réussite future sur la CC k , dans la fenêtre temporelle w .

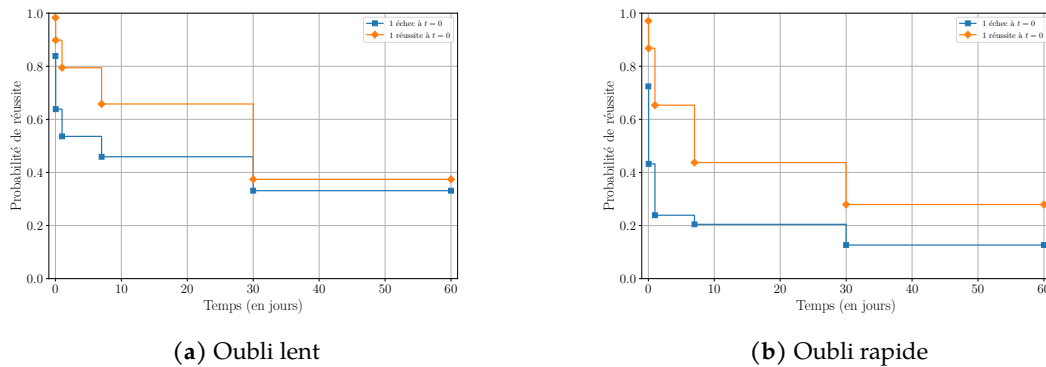


FIGURE 3.3 – Exemples de courbes d’oubli générées par notre modèle d’apprenant DAS3H. Sur chaque graphique, deux situations sont présentées : en bleu, un apprenant de niveau initial moyen pratique une CC à $t = 0$ mais échoue ; en orange, l’apprenant réussit. Ces courbes ont été générées selon notre protocole expérimental principal, décrit dans la sous-section 4.2.3. Ce sont les équivalents pour l’oubli des courbes d’apprentissage idéalisées de GOUTTE et al. (2018).

La figure 3.3 montre deux courbes d’oubli générées aléatoirement² à partir de DAS3H selon une loi normale $\mathcal{N}(0, 2)$. Cela correspond à la zone 1 dans le diagramme 3.2. Dans chaque cas, les courbes représentent l’évolution temporelle de la probabilité de réussite d’un apprenant de niveau moyen répondant à un item de difficulté moyenne ne concernant qu’une

2. C’est-à-dire que les paramètres $\theta_{k,w}$ décrivant ces courbes ont été générés aléatoirement selon la distribution de probabilité indiquée.

seule CC k . À $t = 0$, l'apprenant fait un essai sur un item mettant en jeu cette même CC k : les courbes bleues représentent l'oubli si l'apprenant ne réussit pas à répondre correctement, les courbes orange l'oubli si l'apprenant réussit.

Pour des dimensions de plongement vectoriel supérieures $d > 0$ la formulation de DAS3H se complexifie. La fonction σ devient la fonction probit ϕ (RENDLE, 2012), c'est-à-dire la fonction de répartition d'une variable aléatoire normale centrée réduite $\mathcal{N}(0, 1)$.

Pour implémenter un modèle au sein du cadre des KTM, il faut décider quelles variables encoder dans le vecteur creux x . Dans notre cas, nous avons choisi l'apprenant s , l'item j , les CC $k \in CC(j)$, les réussites $c_{s,k,w}$ et les tentatives $a_{s,k,w}$ pour chaque fenêtre temporelle w . Le modèle DAS3H s'écrit alors comme suit :

$$\mathbb{P}(Y_{s,j,t} = 1) = \phi \left(\underbrace{\mu + \sum_{i=1}^N w_i x_{s,j,t}^i}_A + \underbrace{\sum_{1 \leq i < \ell \leq N} x_{s,j,t}^i x_{s,j,t}^\ell \langle v_i, v_\ell \rangle}_B \right) \quad (3.3)$$

avec :

- ϕ la fonction probit ;
- μ le biais global du modèle ;
- chaque $x_{s,j,t}^i$ la valeur de la caractéristique i du modèle pour l'apprenant s répondant à j à l'instant t ;
- w_i le coefficient réel estimé pour cette caractéristique ;
- v_i la représentation vectorielle continue de cette caractéristique.

Un exemple de caractéristique, pour le modèle DAS3H, peut être l'encodage *one-hot* de l'apprenant s ou bien le logarithme du compteur de tentatives passées dans une fenêtre temporelle sur la CC k . La deuxième somme B contient les interactions par paires entre les représentations vectorielles continues de chacune des variables prises en compte par DAS3H. Cette deuxième somme B enrichit DAS3H car elle permet de modéliser les interactions entre chaque paire de variables. Cela peut par exemple être utile si certains apprenants ont plus de mal sur certaines CC que d'autres.

3.4 Méthodologie expérimentale de validation

Avant toute utilisation dans un algorithme d'espacement adaptatif de l'apprentissage de CC, il est nécessaire d'évaluer la performance prédictive de notre modèle DAS3H et notamment de la comparer à des modèles plus classiques de la littérature sur la modélisation de l'apprenant. L'objet de cette section est de présenter notre méthodologie expérimentale de comparaison de modèles de l'apprenant.

3.4.1 Validation croisée

L'objectif de l'apprentissage automatique supervisé consiste à trouver une fonction f (un modèle) qui permette de prédire à partir d'un ensemble de caractéristiques X une variable Y . Tout l'enjeu réside dans le fait que l'on souhaite avoir confiance dans les prédictions que fera f sur des données nouvelles ; un bon modèle d'apprentissage automatique est un modèle qui possède un fort pouvoir de généralisation, c'est-à-dire que ce qu'il a appris sur un ensemble de données lui permet de prédire avec précision des données qu'il n'aurait jamais vues. Le problème est que l'on ne dispose bien évidemment souvent pas d'un moyen de générer les données X et Y selon leur distribution de probabilité sous-jacente et que l'on doit donc trouver un moyen d'évaluer cette erreur de généralisation avec un ensemble de données fini.

Plusieurs méthodes ont été proposées dans la littérature en apprentissage automatique pour évaluer l'erreur de prédiction d'un algorithme d'apprentissage automatique. La plus

simple et la plus utilisée est celle de la validation croisée (HASTIE et al., 2009) : cette méthode consiste à diviser les données X entre données d'entraînement $X_{\text{entraînement}}$ et données de test X_{test} . Le modèle est d'abord entraîné sur les données d'entraînement ($X_{\text{entraînement}}, Y_{\text{entraînement}}$), puis on le fait prédire les étiquettes \hat{Y}_{test} des données de test, à partir des caractéristiques X_{test} . Ces données n'auront jamais été vues par le modèle pendant sa phase d'entraînement. On calcule ensuite son erreur de prédiction en comparant les vraies étiquettes Y_{test} avec les étiquettes prédites \hat{Y}_{test} .

Pour notre part, nous avons choisi d'utiliser la méthode de validation croisée à k -plis : celle-ci consiste à séparer notre jeu de données initial en k parties (k « plis ») de taille équivalente. La procédure d'évaluation de la performance du modèle comporte elle aussi k parties : à chaque itération, le modèle est entraîné sur $k - 1$ plis de données et testé sur le k -ème pli. À la fin, les métriques de performance prédictive sur chacun des plis sont collectées et moyennées, pour obtenir une estimation de l'erreur de généralisation du modèle. Dans nos expériences, nous avons fixé k à 5, une valeur communément adoptée en apprentissage automatique.

Nos données étant constituées de trajectoires temporelles d'apprenants répondant à des items au fil du temps (et donc de multiples séries temporelles), il nous fallait faire attention à notre méthode de validation croisée. En outre, nous souhaitions que notre méthode de partition des données pour la validation croisée reflète des situations réelles, que l'on pourrait retrouver sur une plateforme d'apprentissage en ligne. Nous avons donc choisi de séparer nos données de deux façons, que l'on a nommées respectivement *généralisation forte* et *généralisation semi-forte* :

- *généralisation forte* : nous séparons notre population d'apprenants en k groupes disjoints de taille équivalente et nous appliquons la validation croisée à k -plis sur ces k groupes. Les apprenants du pli de test ne sont donc à chaque fois jamais vus par le modèle avant la phase de prédiction. Cela permet de reproduire le problème du *démarrage à froid* quand un nouvel apprenant arrive sur une plateforme d'apprentissage en ligne et d'évaluer la performance de nos modèles dans de telles conditions, quand ils ne disposent d'aucune donnée sur les apprenants qu'ils cherchent à prédire. Cette méthode d'évaluation est aussi celle utilisée par VIE et al. (2019). Cela signifie donc aussi que tous les paramètres spécifiques aux apprenants de test du modèle (par exemple, leur niveau α_s pour IRT) ne peuvent pas être estimés par les modèles et sont donc fixés à une valeur moyenne.
- *généralisation semi-forte* : la procédure est similaire à celle de la généralisation forte, à la différence qu'on retire de chaque pli de test les 20 % premières interactions (dans l'ordre chronologique de l'historique) de chacun des apprenants de test et qu'on les rajoute aux plis d'entraînement. Les modèles peuvent donc avoir une première estimation des paramètres des apprenants de test. C'est une procédure qui permet de reproduire la situation dans laquelle, sur une plateforme d'apprentissage en ligne, nous disposerions d'un historique complet pour un certain nombre d'apprenants, qui nous permet donc d'extraire des connaissances concernant par exemple les items, et que nous disposerions également d'un début d'historique pour une partie des apprenants de la plateforme. Nous cherchons alors à savoir à quel point nous sommes capables de prédire avec précision les résultats des futures interactions de ces apprenants-là.

On notera deux différences avec la validation croisée par k -plis canonique :

- la taille des plis peut ne pas être strictement équivalente si jamais certains apprenants ont des historiques d'apprentissage beaucoup plus longs que d'autres. Cependant, si la répartition des longueurs d'historique d'apprentissage est plutôt homogène, en moyenne, les tailles de plis seront similaires.
- pour la généralisation semi-forte, l'union des k plis ne forme pas une partition de X et donc une partie des données ne sera jamais testée par les modèles (les 20 % premières interactions).

3.4.2 Jeux de données utilisés

Les données que nous avons utilisées consistent en des interactions d'un ensemble d'apprenants avec un ensemble d'items. Chaque interaction, c'est-à-dire chaque réponse, est juste ou fausse. La problématique principale était de trouver des jeux de données qui comportaient à la fois des informations sur le moment auquel chaque interaction a eu lieu et sur les CC que mettait en jeu chaque item, l'idéal étant d'avoir des items qui mettent en jeu en moyenne plusieurs CC. En outre, il était préférable d'avoir des jeux de données publiques car (1) nous ne disposions que de peu de jeux de données privées et (2) cela permet aux autres chercheurs du domaine de reproduire nos résultats. Malheureusement, de tels jeux de données sont relativement rares, et concernent souvent les mathématiques. Il serait nécessaire de reproduire les résultats que nous avons obtenus sur des jeux de données concernant des domaines de connaissances plus divers (par exemple, l'histoire) afin de voir si les résultats sont les mêmes. Nous décrivons dans la suite les jeux de données que nous avons utilisés pour nos expériences avec le modèle DAS3H.

ASSISTments 2009-2010 et 2012-2013 (FENG et al., 2009)

ASSISTments est un système de tutorat intelligent en ligne qui a été créé en 2004 dans l'état du Massachusetts aux États-Unis et qui est toujours activement développé aujourd'hui. L'objectif de cette plateforme³ est de fournir un suivi quantitatif de l'apprentissage des élèves (la partie *assessment*) tout en les aidant à progresser en les assistant dans leur apprentissage (la partie *assistance*). ASSISTments propose des exercices de mathématiques⁴ pour des élèves de niveau 4^e (8th grade aux États-Unis). Si les apprenants se trompent à un problème, le système leur propose une décomposition de celui-ci en plusieurs étapes intermédiaires, destinées à la fois à faciliter leur progression et à identifier précisément les raisons de l'échec de l'apprenant. Des indices sont également disponibles, et des messages d'aide personnalisés en fonction de l'erreur de l'apprenant s'affichent parfois.

Les données que nous avons utilisées concernent des interactions entre des apprenants et la plateforme ASSISTments lors des années scolaires 2009-2010⁵ et 2012-2013⁶. Pour l'année scolaire 2009-2010, nous avons seulement utilisé les données "skill builder", dans lesquelles les apprenants pratiquent les CC jusqu'à réussir trois problèmes d'affilée. Toutes ces données sont en accès public sur Internet.

Dans la suite, nous abrégons ces deux jeux de données, respectivement, en `assist09` et `assist12`.

KDD Cup EDM 2010 Challenge : Algebra I 2005-2006 et Bridge to Algebra 2006-2007 (STAMPER et al., 2010)

Ces données ont été fournies lors d'une compétition visant à prédire les performances d'apprenants à partir d'un ensemble de données d'interactions avec un système de tutorat intelligent. Ces données concernent l'apprentissage de l'algèbre par le biais des tuteurs cognitifs *Algebra* et *Bridge to Algebra*, développés par *Carnegie Learning*, pendant les années scolaires 2005-2006 et 2006-2007. Comme pour ASSISTments, ces tuteurs cognitifs (en anglais, *cognitive tutors*) décomposent les problèmes qui sont présentés aux élèves en plusieurs étapes. Si le système remarque que l'apprenant est confus ou a mal compris une notion, il fournira à l'apprenant un message d'aide adapté à la situation ; de même, l'apprenant peut de lui-même

3. Objectif qui a d'ailleurs guidé le choix de son nom : *assisting* et *assessment*.

4. Tous les exercices sont tirés d'annales d'un examen de mathématiques de niveau 4^e, obligatoire pour une partie des élèves du Massachusetts.

5. <https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data/skill-builder-data-2009-2010>

6. <https://sites.google.com/site/assistmentsdata/home/2012-13-school-data-with-affect>

Données	Apprenants	Items	CC	Interactions	Score Moyen	CC par item	Délai moyen	Durée d'étude moy.
assist12	22589	52850	265	2682211	0,696	1,000	4,65	106,9
assist09	3002	17690	123	277459	0,658	1,197	3,01	66,9
bridge06	1130	129263	493	1817393	0,832	1,014	0,5	150,1
algebra05	567	173113	112	606983	0,755	1,363	1,73	110,2
robomission	7066	85	15	146523	0,895	7,412	0,57	5,3

TABLE 3.1 – Statistiques sur les jeux de données utilisés après pré-traitement

demander de l'aide au système, qui lui fournira un indice en fonction de son avancée dans le problème.

Ces données sont aussi en accès public sur Internet. Nous n'avons pas utilisé les données de test pour *Bridge to Algebra 2006-2007*. Dans la suite, nous les abrègerons respectivement en *algebra05* et *bridge06*.

RoboMission (EFFENBERGER, 2019)

RoboMission⁷ est un système d'introduction à l'apprentissage de la programmation : l'objectif des apprenants est de coder un programme qui guidera un vaisseau spatial à travers un monde découpé en grille et lui fera collecter tous les diamants. L'ordre de progression des apprenants est linéaire : l'ensemble des tâches à réaliser disponible dans RoboMission est découpé en 9 niveaux et chacun de ces niveaux est découpé en 3 sous-niveaux. Après chaque tâche réalisée, le système propose à l'apprenant une nouvelle tâche à pratiquer : celle-ci est choisie au hasard parmi les tâches du premier sous-niveau qui n'a pas encore été maîtrisé par l'apprenant. Le système dispose d'un moyen d'estimer si l'apprenant a maîtrisé un sous-niveau ou pas.

Nous avons utilisé les données de décembre 2019⁸ pour nos expériences. Nous représenterons ce jeu de données par *robomission* dans la suite.

Pour tous ces jeux de données, nous avons retiré les apprenants pour lesquels le nombre d'interactions était inférieur à 10. Nous avons aussi retiré les interactions sans indication de CC car nous craignons qu'elles soient trop peu homogènes pour pouvoir être rassemblées sous une même CC générale et abstraite. Cela aurait pu affecter négativement les performances des modèles AFM, PFA et DAS3H qui estiment, entre autres, des paramètres *par CC*. Nous avons aussi retiré les interactions pour lesquelles le résultat n'était pas 0 ou 1, car les modèles comparés n'y sont pas adaptés. Enfin, nous avons aussi retiré tous les doublons d'interactions. Dans les jeux de données d'ASSISTments, la variable utilisée pour identifier les items était `problem_id` et pour les jeux de données *KDD Cup*, cette variable provenait de la concaténation des identifiants du problème et de l'étape, comme recommandé par les organisateurs de la compétition.

Nous avons détaillé les caractéristiques des jeux de données après pré-traitement dans la table 3.1. Le délai moyen par CC correspond à l'intervalle de temps moyen (en jours) entre deux interactions avec la même CC, et la durée d'étude moyenne correspond à la différence temporelle moyenne entre la dernière et la première interaction pour chaque apprenant (en jours).

Pour faciliter la convergence des modèles, nous avons choisi de normaliser nos données de manière à ce que la valeur absolue maximale de chaque variable considérée soit 1 en utilisant le `MaxAbsScaler` de `scikit-learn`. Comme la plupart de nos variables sont des indicatrices (d'apprenant, d'item ou de CC), cette normalisation a seulement modifié les variables de

7. <https://en.robomise.cz/>

8. <https://github.com/adaptive-learning/adaptive-learning-research/tree/master/data/robomission-2019-12>

	appr.	items	CC	réussites	échecs	tentatives	fenêtres temp.
DAS3H	✓	✓	✓	✓		✓	CC
DASH	✓	✓		✓		✓	items
IRT/MIRTb	✓	✓					∅
PFA			✓	✓	✓		∅
AFM	✓		✓			✓	∅

TABLE 3.2 – Caractéristiques considérées par chaque modèle d’apprenant

compteurs. Elle consistait donc, puisque toutes nos variables ne prennent que des valeurs positives ou nulles, à diviser chaque variable de compteur par la valeur maximale prise par ce compteur sur tout le jeu de données.

3.4.3 Modèles concurrents

Nous avons comparé notre modèle DAS3H aux modèles DASH, IRT, PFA et AFM dans le cadre des KTM, pour que les comparaisons soient équitables. Les KTM (VIE et al., 2019) recouvrent facilement la plupart des modèles d’apprenant de la fouille des données éducatives : il suffit de choisir le bon ensemble de prédicteurs (par exemple, « apprenants, items » pour IRT). La table 3.2 résume les caractéristiques utilisées par chaque modèle comparé.

Nous n’avons pas comparé nos modèles à DKT (*Deep Knowledge Tracing*) à cause de la controverse au sujet de sa performance réelle (WILSON, KARLIN et al., 2016; WILSON, XIONG et al., 2016). Pour DASH, nous avons choisi de considérer des biais spécifiques aux items, et non spécifiques aux CC : en effet, dans leur article (LINDSEY et al., 2014), LINDSEY et al. agrègent les items dans des classes d’équivalence (par exemple plusieurs conjugaisons d’un même verbe). Ce n’était pas toujours possible pour nous puisque les items mettent en jeu en général plusieurs CC différentes dans nos jeux de données. Nous avons toutefois testé cette possibilité dans la sous-section 3.6.

Nous avons aussi utilisé les mêmes fenêtres temporelles pour DAS3H et DASH que LINDSEY et al. : $\{1/24, 1, 7, 30, +\infty\}$. Les unités temporelles sont exprimées en jours.

Dimensions de plongement vectoriel

Nous avons comparé trois dimensions de plongement vectoriel pour chaque modèle : 0, 5 et 20. Nous avons choisi ces trois valeurs parce que nous voulions :

1. limiter le nombre de coefficients à estimer par KTM en fixant le maximum à 20;
2. éviter de multiplier les comparaisons de modèles et ainsi faciliter la compréhension des résultats en limitant le nombre de dimensions différentes à 3.

Quand $d > 0$, IRT devient MIRTb, une variante de MIRT qui considère un biais par utilisateur.

Nombre de paramètres par modèle

La table 3.3 donne le nombre de paramètres à estimer en fonction du modèle d’apprenant. S représente le nombre d’apprenants, J le nombre d’items, K le nombre de CC et W le nombre de fenêtres temporelles choisies (5 dans notre cas). On peut voir que DAS3H utilise plus de variables que n’importe quel autre modèle auquel il est comparé ici. Notamment, le fait d’utiliser des fenêtres temporelles augmente beaucoup le nombre de variables liées aux CC. Si les données ne sont pas en quantité suffisante, il est toujours possible de suivre l’exemple de LINDSEY et al. et d’utiliser seulement deux paramètres par fenêtre temporelle pour faciliter l’apprentissage du modèle. Cela signifierait que les paramètres θ sont partagés entre toutes les CC; autrement dit, l’apprentissage et l’oubli se feraient au même rythme pour chaque CC. Nous avons testé cette possibilité à travers des expériences additionnelles, dans la discussion.

Modèle	Paramètres à estimer
IRT	$S + J$
AFM	$S + 2K$
PFA	$3K$
DASH	$S + J + 2W$
DAS3H	$S + J + (2W + 1)K$

TABLE 3.3 – Nombre de paramètres à estimer en fonction du modèle d’apprenant, quand $d = 0$

Régularisation des paramètres

Comme dans des travaux précédents (RENDLE, 2012; VIE et al., 2019), nous utilisons différents priors sur les paramètres pour éviter le surapprentissage des modèles et faciliter l’inférence :

- quand $d = 0$: nous utilisons une régularisation L_2 , ce qui revient à supposer que les paramètres suivent une distribution à priori $\mathcal{N}(0, \lambda^{-1})$ avec $\lambda > 0$;
- quand $d > 0$: chaque paramètre ω du modèle (scalaires et composants des vecteurs de représentation continue) suit une distribution gaussienne à priori $\mathcal{N}(\mu_\omega, 1/\lambda_\omega)$, où μ_ω et λ_ω suivent les hyperpriors $\mu_\omega \sim \mathcal{N}(0, 1)$ et $\lambda_\omega \sim \Gamma(1, 1)$. L’utilisation d’une régularisation bayésienne hiérarchique, comme ici, a plusieurs avantages (RENDLE, 2012) : (1) elle réduit encore plus le risque de surapprentissage par rapport à une régularisation classique, (2) elle diminue le nombre d’hyperparamètres à fixer pour l’entraînement du modèle, (3) elle permet, après entraînement, d’avoir une distribution a posteriori sur les paramètres du modèle. Cette distribution a posteriori peut fournir une bonne estimation initiale des paramètres pour lesquels le modèle n’a pas encore vu d’échantillon, ce qui est le cas par exemple pour les apprenants des plis de test dans nos expériences (cf. sous-section 3.4.1).

Dans leur article (LINDSEY et al., 2014), LINDSEY et al. ont opté pour une approche similaire mais ont supposé que les α_s et les δ_j suivaient différentes distributions. Contrairement à nous, ils ont choisi de ne pas régulariser les paramètres θ_w associés à l’historique de l’apprenant : notre situation est différente parce que nous avons plus de paramètres à estimer qu’ils n’en avaient.

3.4.4 Métriques de performance

Pour comparer la performance d’un modèle d’apprentissage automatique à un autre, il faut avoir un moyen d’évaluer quantitativement à quel point un modèle prédit avec précision les étiquettes des données de test. Dans notre cas, ces étiquettes sont binaires (1 si l’apprenant a répondu correctement à l’item, 0 sinon) et la tâche est une tâche de classification. Nous avons donc choisi les métriques suivantes pour notre tâche :

AUC (*Area Under the ROC Curve*)

L’AUC correspond à l’aire sous la courbe ROC (*Receiver Operating Characteristic*). Cette courbe représente graphiquement le taux de vrais positifs en fonction du taux de faux positifs et ce, pour différents choix de seuil de classifieur. Le taux de vrais positifs (ou sensibilité) correspond à la part d’étiquettes positives qui ont été effectivement prédites positives par le modèle ; le taux de faux positifs correspond quant à lui à la part d’étiquettes négatives qui ont été prédites positives. Un classifieur possède un seuil, c’est-à-dire une valeur numérique au-delà de laquelle le modèle prédira que l’étiquette est positive. En faisant varier ce seuil, on modifie à la fois le taux de faux positifs et le taux de vrais positifs.

Intuitivement, l'AUC représente, pour deux échantillons tirés au hasard dans le jeu de données, l'un négatif et l'autre positif, la probabilité que le modèle en question assigne une probabilité supérieure à l'échantillon positif qu'à l'échantillon négatif. Plus l'AUC est proche de 1, plus le modèle est prédictif. L'AUC permet de tenir compte du déséquilibre entre étiquettes positives et étiquettes négatives dans un jeu de données, ce qui est le cas ici, comme le montre la colonne du score moyen dans la table 3.1. En effet, on peut voir par exemple que `bridge06` a un score moyen de 0,832 : cela signifie que 83,2 % des étiquettes de ce jeu de données sont positives et seulement 16,8 % sont négatives.

NLL (Log-vraisemblance négative moyenne)

Elle se calcule de la manière suivante :

$$NLL = -\frac{1}{n} \sum_{i=1}^n (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i))$$

avec n le nombre d'échantillons, y_i l'étiquette de l'échantillon i et \hat{y}_i l'étiquette prédite pour l'échantillon i . Une NLL plus faible signifie que le modèle est plus prédictif.

RMSE (Root Mean Square Error)

Elle se calcule de la manière suivante :

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Cette métrique met plus de poids sur les écarts importants entre la prédiction et la réalité que sur les écarts plus faibles. Une RMSE plus faible signifie que le modèle commet moins d'erreurs et est plus prédictif.

3.4.5 Implémentation

Nos modèles ont été implémentés en Python et nous avons publié le code pour reproduire nos résultats sur GitHub⁹.

Comme VIE et al. (2019), nous avons utilisé la librairie `pywFM`¹⁰ comme *wrapper* pour `libfm`¹¹ (RENDLE, 2012) pour les modèles avec $d > 0$. Nous avons fixé à 500 le nombre d'itérations pour l'échantillonneur Gibbs MCMC dans ce cas.

Quand $d = 0$, nous avons utilisé la librairie `scikit-learn` (PEDREGOSA et al., 2011) et son implémentation de la régression logistique avec régularisation L_2 . Nous avons choisi SAGA (DEFAZIO et al., 2014) comme algorithme d'optimisation et $C = 1$ comme paramètre de régularisation.

3.5 Résultats

Dans cette section, nous décrivons nos résultats expérimentaux de comparaison de DAS3H à quatre modèles de l'apprenant sur cinq jeux de données différents.

Les résultats obtenus pour la généralisation *forte* sont détaillés dans les tables 3.4, 3.6, 3.8, 3.10 et 3.12. Les résultats pour la généralisation *semi-forte* sont quant à eux détaillés dans les tables 3.5, 3.7, 3.9, 3.11 et 3.13. Dans chacune de ces tables, nous avons reporté l'AUC, la NLL

9. <https://github.com/BenoitChoffin/das3h>

10. <https://github.com/jfloff/pywFM>

11. <http://libfm.org/>

Modèle	Dim.	AUC \uparrow	RMSE \downarrow	NLL \downarrow
DAS3H	0	0,829 \pm 0,002	0,361 \pm 0,005	0,411 \pm 0,010
DAS3H	5	0,828 \pm 0,002	0,362 \pm 0,005	0,412 \pm 0,009
DAS3H	20	0,827 \pm 0,002	0,363 \pm 0,005	0,414 \pm 0,010
DASH	0	0,774 \pm 0,005	0,381 \pm 0,007	0,454 \pm 0,014
DASH	20	0,773 \pm 0,006	0,382 \pm 0,006	0,457 \pm 0,012
DASH	5	0,773 \pm 0,006	0,382 \pm 0,007	0,457 \pm 0,012
MIRTb	5	0,772 \pm 0,003	0,383 \pm 0,006	0,458 \pm 0,012
IRT	0	0,771 \pm 0,005	0,382 \pm 0,007	0,455 \pm 0,014
MIRTb	20	0,771 \pm 0,006	0,383 \pm 0,006	0,459 \pm 0,011
PFA	20	0,768 \pm 0,005	0,385 \pm 0,005	0,458 \pm 0,010
PFA	5	0,767 \pm 0,005	0,385 \pm 0,005	0,458 \pm 0,010
PFA	0	0,754 \pm 0,005	0,390 \pm 0,006	0,468 \pm 0,011
AFM	20	0,729 \pm 0,005	0,395 \pm 0,007	0,481 \pm 0,013
AFM	5	0,729 \pm 0,005	0,395 \pm 0,007	0,481 \pm 0,013
AFM	0	0,722 \pm 0,005	0,398 \pm 0,007	0,486 \pm 0,013

TABLE 3.4 – Comparaison des performances des différents modèles sur le jeu de données algebra05 (généralisation *forte*). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu’une métrique plus haute (plus basse) est meilleure.

Modèle	Dim.	AUC \uparrow	RMSE \downarrow	NLL \downarrow
DAS3H	20	0,838 \pm 0,002	0,356 \pm 0,003	0,401 \pm 0,006
DAS3H	0	0,838 \pm 0,002	0,356 \pm 0,003	0,401 \pm 0,006
DAS3H	5	0,837 \pm 0,002	0,356 \pm 0,003	0,401 \pm 0,006
DASH	0	0,798 \pm 0,002	0,373 \pm 0,004	0,437 \pm 0,008
DASH	5	0,798 \pm 0,002	0,373 \pm 0,004	0,437 \pm 0,008
DASH	20	0,798 \pm 0,002	0,373 \pm 0,004	0,437 \pm 0,008
IRT	0	0,797 \pm 0,002	0,374 \pm 0,004	0,438 \pm 0,008
MIRTb	5	0,797 \pm 0,002	0,373 \pm 0,004	0,438 \pm 0,008
MIRTb	20	0,797 \pm 0,002	0,373 \pm 0,004	0,438 \pm 0,008
PFA	20	0,773 \pm 0,005	0,383 \pm 0,004	0,455 \pm 0,007
PFA	5	0,772 \pm 0,005	0,383 \pm 0,004	0,455 \pm 0,007
PFA	0	0,760 \pm 0,006	0,388 \pm 0,004	0,464 \pm 0,007
AFM	20	0,732 \pm 0,007	0,394 \pm 0,006	0,478 \pm 0,012
AFM	5	0,732 \pm 0,007	0,394 \pm 0,006	0,479 \pm 0,012
AFM	0	0,725 \pm 0,007	0,397 \pm 0,006	0,484 \pm 0,012

TABLE 3.5 – Comparaison des performances des différents modèles sur le jeu de données algebra05 (généralisation *semi-forte*). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu’une métrique plus haute (plus basse) est meilleure.

Modèle	Dim.	AUC \uparrow	RMSE \downarrow	NLL \downarrow
DAS3H	0	0,802 \pm 0,002	0,333 \pm 0,001	0,362 \pm 0,001
DAS3H	5	0,794 \pm 0,003	0,335 \pm 0,001	0,367 \pm 0,001
DAS3H	20	0,794 \pm 0,003	0,336 \pm 0,001	0,367 \pm 0,001
PFA	20	0,756 \pm 0,002	0,346 \pm 0,001	0,390 \pm 0,002
PFA	5	0,756 \pm 0,002	0,346 \pm 0,001	0,390 \pm 0,002
DASH	0	0,749 \pm 0,003	0,347 \pm 0,001	0,393 \pm 0,001
PFA	0	0,748 \pm 0,003	0,348 \pm 0,001	0,393 \pm 0,002
DASH	5	0,748 \pm 0,003	0,349 \pm 0,000	0,398 \pm 0,000
IRT	0	0,748 \pm 0,003	0,348 \pm 0,001	0,393 \pm 0,001
DASH	20	0,748 \pm 0,003	0,350 \pm 0,001	0,399 \pm 0,001
MIRTb	20	0,747 \pm 0,003	0,350 \pm 0,001	0,399 \pm 0,001
MIRTb	5	0,747 \pm 0,003	0,350 \pm 0,001	0,399 \pm 0,001
AFM	20	0,709 \pm 0,004	0,356 \pm 0,001	0,410 \pm 0,002
AFM	0	0,709 \pm 0,004	0,356 \pm 0,001	0,410 \pm 0,002
AFM	5	0,709 \pm 0,004	0,356 \pm 0,001	0,410 \pm 0,002

TABLE 3.6 – Comparaison des performances des différents modèles sur le jeu de données `bridge06` (généralisation *forte*). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu’une métrique plus haute (plus basse) est meilleure.

Modèle	Dim,	AUC \uparrow	RMSE \downarrow	NLL \downarrow
DAS3H	0	0,806 \pm 0,002	0,334 \pm 0,002	0,364 \pm 0,003
DAS3H	20	0,801 \pm 0,002	0,336 \pm 0,002	0,367 \pm 0,003
DAS3H	5	0,798 \pm 0,001	0,336 \pm 0,002	0,369 \pm 0,003
PFA	20	0,759 \pm 0,004	0,348 \pm 0,001	0,394 \pm 0,002
PFA	5	0,759 \pm 0,004	0,348 \pm 0,001	0,394 \pm 0,002
DASH	5	0,751 \pm 0,002	0,349 \pm 0,002	0,396 \pm 0,004
DASH	20	0,751 \pm 0,002	0,349 \pm 0,002	0,396 \pm 0,004
PFA	0	0,750 \pm 0,004	0,350 \pm 0,001	0,397 \pm 0,002
MIRTb	5	0,750 \pm 0,002	0,350 \pm 0,002	0,397 \pm 0,004
MIRTb	20	0,750 \pm 0,002	0,350 \pm 0,002	0,397 \pm 0,004
DASH	0	0,748 \pm 0,002	0,350 \pm 0,002	0,398 \pm 0,004
IRT	0	0,747 \pm 0,002	0,350 \pm 0,002	0,398 \pm 0,004
AFM	5	0,711 \pm 0,004	0,358 \pm 0,002	0,415 \pm 0,003
AFM	20	0,711 \pm 0,004	0,358 \pm 0,002	0,415 \pm 0,003
AFM	0	0,711 \pm 0,004	0,358 \pm 0,002	0,415 \pm 0,003

TABLE 3.7 – Comparaison des performances des différents modèles sur le jeu de données `bridge06` (généralisation *semi-forte*). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu’une métrique plus haute (plus basse) est meilleure.

Modèle	Dim.	AUC ↑	RMSE ↓	NLL ↓
DAS3H	5	0,747 ± 0,001	0,420 ± 0,001	0,529 ± 0,002
DAS3H	20	0,743 ± 0,001	0,421 ± 0,001	0,531 ± 0,002
DAS3H	0	0,743 ± 0,001	0,422 ± 0,001	0,532 ± 0,002
DASH	20	0,703 ± 0,001	0,433 ± 0,001	0,557 ± 0,002
DASH	0	0,703 ± 0,001	0,434 ± 0,001	0,557 ± 0,002
DASH	5	0,703 ± 0,001	0,433 ± 0,001	0,557 ± 0,002
IRT	0	0,702 ± 0,001	0,434 ± 0,001	0,558 ± 0,002
MIRTb	20	0,702 ± 0,001	0,434 ± 0,001	0,558 ± 0,002
MIRTb	5	0,701 ± 0,001	0,434 ± 0,001	0,558 ± 0,002
PFA	5	0,679 ± 0,001	0,440 ± 0,001	0,571 ± 0,002
PFA	20	0,679 ± 0,001	0,440 ± 0,001	0,571 ± 0,002
PFA	0	0,666 ± 0,002	0,442 ± 0,001	0,576 ± 0,002
AFM	0	0,610 ± 0,001	0,452 ± 0,001	0,597 ± 0,002
AFM	5	0,610 ± 0,001	0,452 ± 0,001	0,597 ± 0,002
AFM	20	0,610 ± 0,001	0,452 ± 0,001	0,597 ± 0,002

TABLE 3.8 – Comparaison des performances des différents modèles sur le jeu de données `assist12` (généralisation *forte*). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu’une métrique plus haute (plus basse) est meilleure.

Modèle	Dim.	AUC ↑	RMSE ↓	NLL ↓
DAS3H	5	0,770 ± 0,001	0,412 ± 0,001	0,512 ± 0,003
DAS3H	20	0,769 ± 0,001	0,413 ± 0,001	0,513 ± 0,003
DAS3H	0	0,766 ± 0,001	0,414 ± 0,001	0,516 ± 0,003
DASH	0	0,737 ± 0,001	0,425 ± 0,001	0,539 ± 0,003
DASH	5	0,737 ± 0,001	0,425 ± 0,001	0,539 ± 0,002
DASH	20	0,737 ± 0,001	0,425 ± 0,001	0,539 ± 0,002
IRT	0	0,736 ± 0,001	0,426 ± 0,001	0,540 ± 0,002
MIRTb	5	0,736 ± 0,001	0,426 ± 0,001	0,540 ± 0,002
MIRTb	20	0,735 ± 0,001	0,426 ± 0,001	0,540 ± 0,002
PFA	5	0,680 ± 0,002	0,440 ± 0,001	0,573 ± 0,003
PFA	20	0,680 ± 0,002	0,440 ± 0,001	0,573 ± 0,003
PFA	0	0,667 ± 0,003	0,443 ± 0,002	0,578 ± 0,003
AFM	0	0,608 ± 0,004	0,453 ± 0,002	0,599 ± 0,003
AFM	20	0,608 ± 0,004	0,453 ± 0,002	0,599 ± 0,003
AFM	5	0,608 ± 0,004	0,453 ± 0,002	0,599 ± 0,003

TABLE 3.9 – Comparaison des performances des différents modèles sur le jeu de données `assist12` (généralisation *semi-forte*). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu’une métrique plus haute (plus basse) est meilleure.

Modèle	Dim.	AUC \uparrow	RMSE \downarrow	NLL \downarrow
DAS3H	5	0,757 \pm 0,006	0,429 \pm 0,002	0,546 \pm 0,005
DAS3H	20	0,753 \pm 0,005	0,430 \pm 0,002	0,549 \pm 0,004
DAS3H	0	0,749 \pm 0,006	0,433 \pm 0,002	0,554 \pm 0,005
PFA	20	0,722 \pm 0,006	0,439 \pm 0,002	0,570 \pm 0,005
PFA	5	0,722 \pm 0,006	0,439 \pm 0,002	0,570 \pm 0,005
PFA	0	0,706 \pm 0,006	0,444 \pm 0,003	0,580 \pm 0,006
DASH	5	0,680 \pm 0,006	0,455 \pm 0,002	0,601 \pm 0,004
DASH	20	0,680 \pm 0,007	0,455 \pm 0,002	0,602 \pm 0,004
DASH	0	0,679 \pm 0,006	0,455 \pm 0,002	0,601 \pm 0,004
IRT	0	0,679 \pm 0,006	0,455 \pm 0,002	0,601 \pm 0,004
MIRTb	20	0,678 \pm 0,007	0,456 \pm 0,002	0,602 \pm 0,004
MIRTb	5	0,678 \pm 0,007	0,456 \pm 0,002	0,603 \pm 0,004
AFM	20	0,635 \pm 0,007	0,461 \pm 0,003	0,616 \pm 0,007
AFM	5	0,635 \pm 0,007	0,461 \pm 0,003	0,616 \pm 0,007
AFM	0	0,628 \pm 0,007	0,463 \pm 0,003	0,618 \pm 0,007

TABLE 3.10 – Comparaison des performances des différents modèles sur le jeu de données `assist09` (généralisation *forte*). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu’une métrique plus haute (plus basse) est meilleure.

Modèle	Dim.	AUC \uparrow	RMSE \downarrow	NLL \downarrow
DAS3H	20	0,797 \pm 0,006	0,411 \pm 0,005	0,509 \pm 0,010
DAS3H	5	0,796 \pm 0,005	0,412 \pm 0,004	0,510 \pm 0,009
DAS3H	0	0,787 \pm 0,005	0,416 \pm 0,004	0,519 \pm 0,009
DASH	20	0,746 \pm 0,005	0,434 \pm 0,004	0,555 \pm 0,009
MIRTb	20	0,746 \pm 0,004	0,434 \pm 0,004	0,556 \pm 0,008
DASH	5	0,745 \pm 0,004	0,434 \pm 0,004	0,556 \pm 0,008
MIRTb	5	0,745 \pm 0,004	0,434 \pm 0,004	0,556 \pm 0,008
DASH	0	0,745 \pm 0,004	0,434 \pm 0,004	0,557 \pm 0,008
IRT	0	0,744 \pm 0,004	0,434 \pm 0,004	0,557 \pm 0,008
PFA	20	0,729 \pm 0,004	0,438 \pm 0,004	0,567 \pm 0,009
PFA	5	0,729 \pm 0,004	0,438 \pm 0,004	0,568 \pm 0,009
PFA	0	0,713 \pm 0,005	0,443 \pm 0,004	0,578 \pm 0,010
AFM	5	0,636 \pm 0,004	0,462 \pm 0,004	0,617 \pm 0,008
AFM	20	0,636 \pm 0,004	0,462 \pm 0,004	0,617 \pm 0,008
AFM	0	0,630 \pm 0,004	0,463 \pm 0,004	0,620 \pm 0,007

TABLE 3.11 – Comparaison des performances des différents modèles sur le jeu de données `assist09` (généralisation *semi-forte*). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu’une métrique plus haute (plus basse) est meilleure.

Modèle	Dim.	AUC \uparrow	RMSE \downarrow	NLL \downarrow
DAS3H	20	0,853 \pm 0,005	0,275 \pm 0,004	0,253 \pm 0,005
DAS3H	0	0,851 \pm 0,004	0,273 \pm 0,004	0,252 \pm 0,006
PFA	20	0,851 \pm 0,002	0,267 \pm 0,002	0,245 \pm 0,003
PFA	5	0,849 \pm 0,003	0,269 \pm 0,002	0,247 \pm 0,003
DAS3H	5	0,849 \pm 0,004	0,277 \pm 0,004	0,256 \pm 0,005
PFA	0	0,828 \pm 0,004	0,275 \pm 0,003	0,260 \pm 0,003
DASH	0	0,821 \pm 0,005	0,288 \pm 0,005	0,274 \pm 0,008
IRT	0	0,819 \pm 0,005	0,288 \pm 0,005	0,275 \pm 0,007
DASH	5	0,818 \pm 0,005	0,290 \pm 0,005	0,278 \pm 0,007
DASH	20	0,818 \pm 0,005	0,290 \pm 0,004	0,277 \pm 0,007
AFM	20	0,818 \pm 0,004	0,281 \pm 0,003	0,267 \pm 0,004
MIRTb	5	0,816 \pm 0,005	0,290 \pm 0,004	0,278 \pm 0,007
MIRTb	20	0,815 \pm 0,005	0,290 \pm 0,004	0,278 \pm 0,007
AFM	5	0,815 \pm 0,004	0,282 \pm 0,003	0,269 \pm 0,004
AFM	0	0,793 \pm 0,004	0,287 \pm 0,003	0,280 \pm 0,005

TABLE 3.12 – Comparaison des performances des différents modèles sur le jeu de données robomission (généralisation *forte*). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu’une métrique plus haute (plus basse) est meilleure.

Modèle	Dim.	AUC \uparrow	RMSE \downarrow	NLL \downarrow
DAS3H	20	0,839 \pm 0,002	0,295 \pm 0,002	0,288 \pm 0,003
DAS3H	0	0,836 \pm 0,002	0,294 \pm 0,003	0,288 \pm 0,004
DAS3H	5	0,834 \pm 0,004	0,297 \pm 0,003	0,292 \pm 0,005
PFA	20	0,831 \pm 0,001	0,291 \pm 0,003	0,286 \pm 0,005
PFA	5	0,827 \pm 0,002	0,292 \pm 0,003	0,288 \pm 0,005
DASH	0	0,804 \pm 0,004	0,308 \pm 0,003	0,311 \pm 0,004
PFA	0	0,802 \pm 0,002	0,300 \pm 0,003	0,304 \pm 0,005
IRT	0	0,802 \pm 0,004	0,308 \pm 0,003	0,312 \pm 0,005
DASH	5	0,802 \pm 0,004	0,310 \pm 0,002	0,314 \pm 0,003
DASH	20	0,801 \pm 0,005	0,310 \pm 0,003	0,315 \pm 0,004
MIRTb	20	0,800 \pm 0,005	0,309 \pm 0,002	0,314 \pm 0,004
MIRTb	5	0,800 \pm 0,004	0,310 \pm 0,003	0,315 \pm 0,004
AFM	20	0,791 \pm 0,005	0,306 \pm 0,003	0,312 \pm 0,004
AFM	5	0,789 \pm 0,004	0,307 \pm 0,003	0,314 \pm 0,004
AFM	0	0,759 \pm 0,006	0,313 \pm 0,003	0,328 \pm 0,005

TABLE 3.13 – Comparaison des performances des différents modèles sur le jeu de données robomission (généralisation *semi-forte*). Les métriques sont moyennées sur les 5 plis à chaque fois, et nous avons indiqué les écarts-types après le symbole \pm . \uparrow et \downarrow indiquent respectivement qu’une métrique plus haute (plus basse) est meilleure.

et la RMSE moyennes sur 5 plis pour chaque modèle et jeu de données. Les écarts-types sur les 5 plis sont également indiqués.

On peut constater que les performances prédictives de l'ensemble des modèles sont pour la plupart meilleures pour la généralisation *semi-forte* que pour la généralisation *forte*, pour tous les jeux de données sauf *robomission*. Ceci s'explique d'abord par la taille supérieure des données d'entraînement pour tous les modèles. En outre, pour les modèles qui estiment des biais spécifiques aux apprenants (comme IRT, DAS3H), le fait d'avoir accès à des échantillons (20 % des interactions) concernant les apprenants des données de test permet d'obtenir une première estimation de ces biais, contrairement à la généralisation *forte*. On peut d'ailleurs voir que pour les modèles qui n'estiment pas de biais spécifiques aux apprenants (AFM et PFA), il y a peu de différences entre la généralisation *forte* et la généralisation *semi-forte*. Par exemple, l'AUC moyenne d'AFM ($d = 0$) est de 0,630 sur *assist09* en généralisation *semi-forte*, contre 0,628 en généralisation *forte*.

On peut voir également que notre modèle DAS3H possède de meilleures performances prédictives que tous les autres modèles comparés sur chacun des jeux de données. Le jeu de données *robomission* est le seul sur lequel un autre modèle (PFA) parvient à obtenir des performances comparables, voire même un peu meilleures si l'on considère la RMSE et la NLL (tables 3.12 et 3.13).

On observe aussi que DASH et IRT/MIRTb ont souvent des performances similaires. Ceci provient du fait que la différence entre DASH et IRT/MIRTb réside dans le module temporel h_θ de DASH : ce module compte les interactions d'un apprenant avec un même item. Étant donné le grand nombre d'items différents dans ces jeux de données (cf. table 3.1), il est peu probable qu'un même apprenant revienne plusieurs fois sur le même item : les compteurs d'interactions de DASH sont donc probablement souvent nuls.

Nos résultats expérimentaux montrent que DAS3H est capable de modéliser la performance des apprenants plus précisément que les autres modèles quand le jeu de données possède les informations sur les liens entre items et CC ainsi que les instants auxquels les interactions ont été réalisées. Nous faisons l'hypothèse que ce gain de performance prédictive provient d'une modélisation temporelle plus fine de l'influence de la pratique passée des CC sur la performance future.

3.6 Discussion

Dans cette section, nous menons des analyses supplémentaires pour mieux comprendre les résultats que nous avons présentés à la section précédente. Nous structurons cette section en différentes questions de recherche.

3.6.1 Quel est l'impact de la dimension des représentations vectorielles continues sur la performance prédictive ?

L'impact des représentations vectorielles continues et des interactions entre chaque paire de variables, rendu possible par les KTM, semble à la fois très faible et variable. En effet, la différence de performance prédictive entre les différentes dimensions de plongement vectoriel pour un même modèle est rarement élevée : elle est souvent de l'ordre du millième d'AUC (par exemple 0,679, 0,680 et 0,680 pour DASH respectivement avec $d = 0$, $d = 5$ et $d = 20$ sur *assist09*, généralisation *forte*). En de rares occasions, elle peut être cependant plus élevée, comme en témoigne la différence d'AUC pour PFA sur *robomission* (généralisation *semi-forte*) : 0,802 pour $d = 0$, 0,827 pour $d = 5$ et 0,831 pour $d = 20$.

En outre, le classement des dimensions de plongement vectoriel en termes de performances prédictives varie en fonction des modèles, des jeux de données, et même des types de généralisation. Par exemple,

- pour algebra05 (généralisation *forte*), la dimension $d = 0$ est meilleure que $d = 5$, qui est meilleure que $d = 20$ pour DAS3H, tandis que pour PFA, $d = 20$ est meilleure que $d = 5$ qui est elle-même meilleure que $d = 0$;
- pour algebra05 (généralisation *forte*), $d = 20$ est meilleure pour PFA, suivie de $d = 5$ et $d = 0$; tandis que pour assist12, c'est $d = 5$ qui est meilleure pour PFA, suivie de $d = 20$ et $d = 0$;
- pour assist09 (généralisation *semi-forte*), la dimension $d = 20$ est meilleure que $d = 5$ qui est meilleure que $d = 0$ pour DAS3H. Au contraire, pour la généralisation *forte* sur ce même jeu de données, c'est $d = 5$ qui est meilleure, suivie de $d = 20$ et de $d = 0$.

Ces résultats s'appliquent également à notre modèle DAS3H. Nos résultats confirment ceux de VIE et al. : le rôle de la dimension d semble être plutôt limité. Une étude plus complète de l'impact de la dimension des plongements vectoriels sur la performance prédictive permettrait de déterminer s'il existe bien des différences de performance prédictive entre les différentes dimensions de plongement vectoriel et si ces différences sont liées par exemple au modèle employé.

3.6.2 La représentation fine de l'historique des interactions de chaque apprenant explique-t-elle la performance supérieure de DAS3H ?

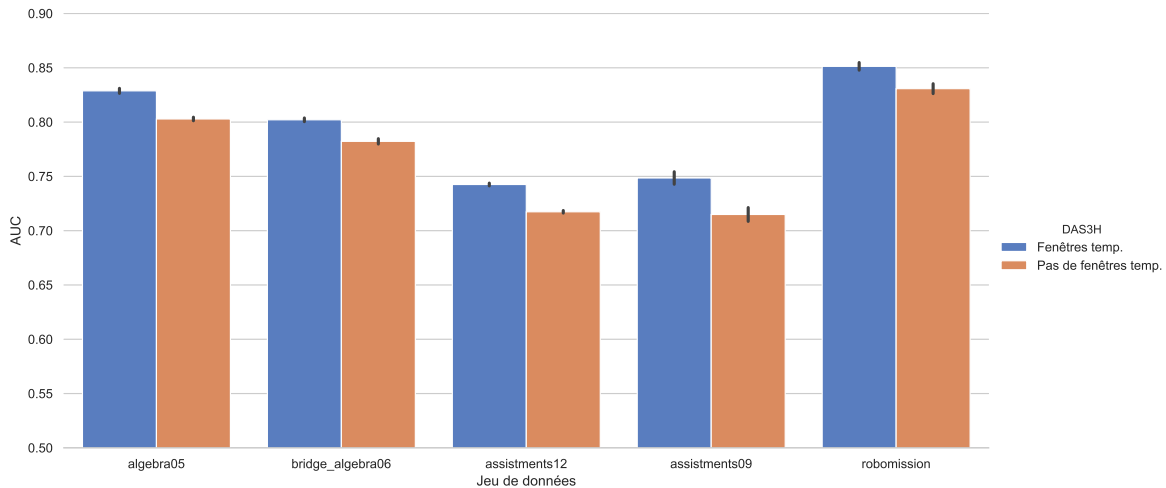


FIGURE 3.4 – Comparaison des AUC du modèle DAS3H original et d'une version alternative qui n'utilise pas de fenêtres temporelles, sur tous les jeux de données. Une métrique plus haute est meilleure.

Pour tenter de comprendre nos résultats, nous voulons savoir ce qui rend notre modèle DAS3H plus prédictif que les autres modèles auxquels nous l'avons comparé. Notre hypothèse est que l'utilisation d'une représentation plus précise de l'historique de la pratique passée (interactions et résultats) au niveau des CC permet au modèle de capturer des phénomènes plus complexes que la simple pratique : par exemple, l'oubli. Pour tester cette hypothèse, nous réalisons ici plusieurs tests d'ablation. Nous mesurons empiriquement la différence d'AUC sur nos jeux de données (généralisation forte) quand les variables avec fenêtres temporelles sont remplacées par de simples compteurs de réussites et d'échecs sur les interactions passées. Pour chaque jeu de données, nous comparons donc l'AUC moyen du modèle DAS3H d'origine avec un modèle similaire mais pour lequel le module temporel h_θ est remplacé par

$$\sum_{k \in \text{CC}(j)} \eta_k^{c_{s,k}} + \rho_k^{a_{s,k}}. \quad (3.4)$$

Cette formulation se rapproche donc plus de celle de PFA. Notons toutefois que nous avons aussi retiré les logarithmes de h_θ ici.

	d	bridge06	algebra05	assist12	assist09	robomission
DAS3H	0	0,802 \pm 0,002	0,829 \pm 0,002	0,743 \pm 0,001	0,749 \pm 0,006	0,851 \pm 0,004
	5	0,794 \pm 0,003	0,828 \pm 0,002	0,747 \pm 0,001	0,757 \pm 0,006	0,849 \pm 0,004
	20	0,794 \pm 0,003	0,827 \pm 0,002	0,743 \pm 0,001	0,753 \pm 0,005	0,853 \pm 0,005
DAS3H _{1p}	0	0,793 \pm 0,002	0,820 \pm 0,002	0,737 \pm 0,001	0,739 \pm 0,005	0,842 \pm 0,004
	5	0,794 \pm 0,003	0,825 \pm 0,001	0,749 \pm 0,001	0,750 \pm 0,005	0,837 \pm 0,004
	20	0,794 \pm 0,003	0,823 \pm 0,001	0,747 \pm 0,002	0,748 \pm 0,005	0,839 \pm 0,004

TABLE 3.14 – Comparaison de l’AUC (généralisation *forte*) sur tous les jeux de données entre DAS3H et DAS3H_{1p}, la version de DAS3H pour laquelle l’influence de la pratique passée ne diffère pas d’une CC à l’autre. Les écarts-types sont indiqués après la moyenne sur les 5 plis. Une métrique plus haute est meilleure.

Les résultats obtenus sont représentés dans la figure 3.4. La moyenne et l’écart-type sur les 5 plis sont indiqués. Par simplicité, et puisqu’aucune dimension ne se démarque par ses performances supérieures, nous avons choisi une dimension de plongement vectoriel de $d = 0$.

On peut voir que l’utilisation de variables à fenêtres temporelles améliore systématiquement et de manière substantielle l’AUC du modèle. Ceci suggère que l’utilisation d’une représentation plus fine de l’historique des interactions de l’apprenant avec les CC du programme améliore les performances du modèle par rapport à de simples compteurs de réussites et de tentatives.

3.6.3 Des CC différentes ont-elles des courbes d’apprentissage et d’oubli différentes ?

Nous cherchons également à savoir si notre hypothèse selon laquelle l’impact de la pratique sur la performance future pouvait différer d’une CC à l’autre est pertinente. C’est pourquoi nous comparons ici notre modèle DAS3H dans sa formulation d’origine à une version différente, plus proche de la formulation du modèle DASH (LINDSEY et al., 2014). Dans cette version alternative de DAS3H, baptisée DAS3H_{1p}, les CC partagent les mêmes paramètres θ_{2w+1} et θ_{2w+2} au sein d’une fenêtre temporelle w donnée. Ceci signifie que nous estimons alors une seule courbe d’apprentissage et d’oubli pour l’ensemble des CC. En d’autres termes :

$$\forall k \in \llbracket 1, K \rrbracket, \forall w \in \llbracket 1, 2W \rrbracket \quad \theta_{k,w} = \theta_w. \quad (3.5)$$

Les résultats de cette comparaison sont donnés dans la table 3.14, qui détaille l’AUC de chacun de ces modèles sur tous les jeux de données, avec la généralisation *forte*. Sur la plupart des jeux de données (sauf *assist12*), l’utilisation de différents paramètres pour différentes CC dans h_θ augmente la performance prédictive, mesurée par l’AUC. Le gain d’AUC est en moyenne de +0,01. Ceci suggère que certaines CC ont des courbes d’apprentissage et d’oubli significativement différentes.

3.6.4 Faut-il compter les interactions au niveau de l’item ou au niveau de la CC dans DASH ?

On pourrait avancer aussi que la comparaison que nous avons faite entre DAS3H et DASH plus haut n’était pas totalement juste. Dans leurs articles, LINDSEY et al. regroupent les items similaires entre eux pour former des CC disjointes. Cela n’était pas possible pour nous sur les données issues de la *KDD Cup* ou de *RoboMission* puisque certains items ont été étiquetés par les experts avec plusieurs CC en même temps.

DASH	$d = 0$	$d = 5$	$d = 20$
items	$0,703 \pm 0,001$	$0,703 \pm 0,001$	$0,703 \pm 0,001$
CC	$0,740 \pm 0,002$	$0,751 \pm 0,002$	$0,751 \pm 0,002$

TABLE 3.15 – Comparaison de l’AUC sur `assist12` entre DASH [items] et DASH [CC]. Les écarts-types sont indiqués après la moyenne sur les 5 plis. Une métrique plus haute est meilleure.

Néanmoins, le jeu de données `assist12` ne possède *que* des items avec une unique CC. Pour évaluer si considérer la distribution temporelle et les résultats de la pratique passée au niveau des CC (DASH [CC]) ou au niveau des items (DASH [items]) produirait de meilleurs résultats, nous avons comparé ces deux formulations de DASH sur `assist12`. Notons que DASH [CC] correspond ici à $\text{DAS3H}_{1,p}$ sans estimation des paramètres de facilité des CC β_k . Les résultats détaillés de cette comparaison, pour la généralisation *forte*, sont disponibles dans la table 3.15. On peut voir que DASH [CC] obtient de meilleures performances prédictives que DASH [items] pour toutes les dimensions de plongement vectoriel. Ces résultats ne sont guère étonnants : comme mentionné précédemment, le faible nombre de CC dans `assist12` comparé au nombre considérable d’items indique que le retour d’un apprenant sur un même item est peu probable. Néanmoins, cette formulation de DASH, qui se rapproche de DAS3H, n’est utilisable que sur les jeux de données sur lesquels un item met en jeu au maximum une CC. L’une des forces de DAS3H est au contraire d’être adapté à des jeux de données contenant des items qui mettent en jeu plusieurs CC à la fois.

Enfin, nous souhaitons illustrer nos résultats avec DAS3H en prenant deux exemples de CC tirées du jeu de données `algebra05`, une pour laquelle la courbe d’oubli estimée est pentue, l’autre pour laquelle elle est plus plate. Nous avons choisi d’utiliser la différence de probabilité de bonne réponse quand une « réussite » (c’est-à-dire une bonne réponse sur un item mettant en jeu la CC) quittait une seule fenêtre temporelle. Nous avons calculé cette différence pour chaque CC, pour chaque couple de fenêtres temporelles et pour chaque pli. Nous avons ensuite moyenné les différences sur les 5 plis et sur les différentes fenêtres temporelles, ce qui donne pour chaque CC la diminution moyenne de probabilité de bonne réponse quand une « réussite » quitte une unique fenêtre temporelle. Une des CC pour laquelle la trace en mémoire diminue le plus lentement concerne l’ombrage de zones pour lesquelles une valeur donnée est inférieure à un seuil : en moyenne et toutes choses égales par ailleurs, la probabilité de bonne réponse pour un item qui met en jeu cette CC diminue de 1,15 % quand une seule « réussite » quitte une fenêtre temporelle. Une telle CC n’est en effet pas difficile à mémoriser avec seulement quelques révisions périodiques. Au contraire, la CC qui concerne l’application des exposants est plus difficile à maîtriser sur le long terme : pour cette CC, la probabilité de bonne réponse diminue de 2,74 % quand une « réussite » quitte une fenêtre temporelle, ce qui est plus que le double de la valeur précédente et ce qui est cohérent avec la description de la CC. Ceci vient également conforter notre hypothèse initiale selon laquelle les CC s’apprennent et s’oublient à des rythmes différents.

Pour conclure cette section, nous avons vu que notre modèle DAS3H proposait de meilleures performances prédictives que les autres modèles d’apprenant auxquels nous l’avons comparé, y compris le modèle DASH. L’utilisation de variables à fenêtres temporelles au lieu de simples compteurs de réussites et de tentatives sur les CC et l’estimation de paramètres différents pour différentes CC améliorent substantiellement la performance du modèle. Sachant que DAS3H surpasse ses versions alternatives (cf. figure 3.4 et table 3.14) et DASH, ces résultats suggèrent qu’inclure à la fois les relations entre items et CC ainsi que le phénomène d’oubli est plus efficace que considérer seulement l’un ou l’autre quand on conçoit un modèle d’apprenant. Contre toute attente, l’utilisation de plongements vectoriels pour les caractéristiques du modèle, cependant, n’a pas semblé apporter de représentations plus riches de la pratique passée.

Mais au-delà de ses performances, DAS3H a l'avantage d'être adapté au problème de planification adaptative et personnalisée de révisions de CC que nous avons décrit à la section 1.2. En effet, il contient une estimation du niveau de maîtrise actuel de l'apprenant s sur chaque CC ou combinaison de CC. Il peut aussi être utilisé pour inférer l'évolution future de cette maîtrise et donc pour optimiser un planning de révisions adaptatif et personnalisé de CC.

3.7 Conclusion

Dans ce chapitre, nous avons posé la première brique du développement d'un algorithme adaptatif et personnalisé de répétition espacée de la pratique de CC. Ainsi, nous avons souhaité créer un nouveau modèle de l'apprenant qui prendrait en compte l'oubli et la pratique des apprenants sur toutes les CC à maîtriser. Ce modèle pourra rentrer (comme nous le verrons au chapitre suivant) dans la composition d'un algorithme fondé sur un modèle; cette option est régulièrement utilisée pour le développement d'algorithmes d'espacement adaptatif pour la pratique de *flashcards*, comme nous l'avons rappelé au chapitre 2.

Ce modèle que nous avons décrit plus haut s'appelle DAS3H et il étend le modèle DASH proposé par LINDSEY et al. Contrairement à DASH, DAS3H permet à chaque item de mettre en jeu un nombre arbitraire de CC : il intègre pour cela l'information fournie par une q -matrice (TATSUOKA, 1983). De plus, DAS3H estime une courbe d'apprentissage et d'oubli pour *chaque* CC, tandis que DASH suppose que tous les items s'apprennent et s'oublient au même rythme. Enfin, DAS3H est basé sur le modèle récent des *Knowledge Tracing Machines* (VIE et al., 2019) car nous pensions que les représentations vectorielles continues de variables en plusieurs dimensions et les interactions par paires entre variables pouvaient donner des modèles plus riches et plus expressifs. À notre connaissance, les KTM n'ont jamais été utilisées avec des variables décrivant la mémoire ou l'oubli jusqu'ici.

Enfin, nous avons montré que DAS3H surpassait plusieurs modèles classiques de l'apprenant dans la littérature en fouille de données éducatives sur cinq jeux de données éducatives réelles incluant des informations sur les dates d'interaction et les CC de chaque item. Nous avons montré qu'ajouter des variables à fenêtre temporelle et supposer que différentes CC devraient avoir différentes courbes d'apprentissage et d'oubli améliorerait significativement la performance prédictive, mesurée par l'AUC sur les données de test.

Ce modèle a été introduit dans le papier (CHOFFIN, POPINEAU et al., 2019) que nous avons présenté à la conférence *Educational Data Mining* (EDM) 2019. Il a reçu à cette occasion le prix du meilleur article long de la conférence (*Best Full Paper Award*). Pour cette raison, nous avons eu la chance de pouvoir le présenter en tant qu'invité à la conférence *Learning Analytics and Knowledge* (LAK) 2020.

Chapitre 4

Heuristiques d'espacement adaptatif et personnalisé pour la maîtrise à long terme de CC

4.1 Introduction

Nous avons exposé au chapitre 2 différents travaux de recherche qui ont permis de développer des algorithmes d'espacement adaptatif et personnalisé de l'apprentissage. Plusieurs études montrent que, comparés à des plannings fixes de révision, ces systèmes améliorent la mémorisation à long terme des connaissances révisées par la personnalisation de l'espacement entre deux révisions successives d'une même connaissance.

Néanmoins, les algorithmes d'espacement adaptatif existants sont conçus pour l'apprentissage et la révision de connaissances simples, telles que des mots de vocabulaire ou des faits historiques. Ils ne gèrent notamment pas la possibilité pour un item de mettre en jeu plusieurs CC, ce qui est pourtant commun dans beaucoup de situations éducatives réelles. Or, d'autres études montrent que l'effet d'espacement s'étend au-delà de la mémorisation pure de connaissances simples, par exemple à des concepts et des procédures en sciences (BARZAGAR NAZARI et al., 2019; ROHRER et al., 2006; VLACH et al., 2012; REYNOLDS et al., 1964) ou encore à l'apprentissage de la syntaxe d'une langue étrangère (BIRD, 2011). Il y a donc fort à parier qu'il soit possible de développer des algorithmes d'espacement adaptatif et personnalisé pour la maîtrise à long terme de CC.

Dans ce chapitre, nous nous penchons donc sur la deuxième problématique principale qui nous a guidés durant cette thèse :

Sachant que l'on dispose pour un apprenant d'un budget fixe d'items de révision et que les périodes de révision sont déterminées par un agent extérieur (comme un enseignant), comment choisir la séquence d'items qui maximisera la maîtrise à long terme de l'apprenant sur les CC du programme ?

Parmi les algorithmes d'espacement adaptatif de l'apprentissage de *flashcards*, plusieurs reposent sur des heuristiques simples qui présentent malgré tout de bonnes performances sur l'amélioration de la mémorisation des apprenants à long terme (M. M. KHAJAH et al., 2014; LINDSEY et al., 2014; METTLER et al., 2016). Parmi eux, plusieurs sont des algorithmes *avec* modèle. On peut naturellement se demander s'il est possible d'étendre de telles heuristiques d'espacement adaptatif à notre problématique de maîtrise à long terme d'un ensemble de CC.

Nous nous intéressons donc plus précisément aux trois questions de recherche suivantes :

- *Peut-on trouver des heuristiques simples et efficaces de sélection de la CC optimale à faire réviser à un instant t ? Sur quelles dimensions peut-on comparer qualitativement ces stratégies de révision ? Parmi les heuristiques que nous développons, lesquelles optimisent le mieux la mémorisation à long terme de l'apprenant sur l'ensemble des CC ?*

- *Nous introduisons dans notre cadre de recherche des items mettant en jeu plusieurs CC à la fois. Peut-on tirer parti de ces relations pour accélérer le processus de révision en sélectionnant un sous-ensemble de CC au lieu d'une seule? Quel est l'impact sur la complexité des items choisis?*
- *Les stratégies non-adaptatives sont-elles nécessairement moins performantes que les stratégies adaptatives?*

Dans ce chapitre, nous présentons tout d'abord notre protocole expérimental de simulation de trajectoires d'apprentissage et d'oubli synthétiques sur des cohortes d'apprenants. Ces simulations nous permettent de comparer dans des environnements contrôlés et idéalisés les performances en termes de mémorisation à long terme des stratégies de sélection de CC pour l'espacement adaptatif et personnalisé de l'apprentissage.

Dans un deuxième temps, nous détaillons les trois heuristiques de sélection de CC pour l'espacement adaptatif et personnalisé de l'apprentissage que nous avons développées et implémentées. Nous utilisons dans deux de ces heuristiques le modèle DAS3H que nous avons introduit au chapitre précédent. Nous proposons également une nouvelle procédure gloutonne pour sélectionner à n'importe quel instant le sous-ensemble de CC le plus prometteur à choisir, au lieu de la CC la plus prometteuse. Nous étendons deux des heuristiques citées précédemment avec cette procédure.

Ensuite, nous présentons les résultats des expériences que nous avons menées sur plusieurs cohortes de 500 apprenants synthétiques, afin de comparer les performances des heuristiques d'espacement développées précédemment. Nous avons rendu public sur la plateforme GitHub tout le code Python de ces expériences. Nous détaillons également plusieurs expériences complémentaires que nous avons menées afin de tester la robustesse de nos algorithmes et mieux comprendre leur fonctionnement.

Enfin, nous concluons ce chapitre en synthétisant nos résultats.

4.2 Protocole expérimental de comparaison des stratégies

Dans cette section, nous décrivons le protocole expérimental que nous avons développé pour comparer par la suite des heuristiques d'espacement adaptatif de l'apprentissage. Comme nous ne disposions pas de terrain d'expérimentation pour tester sur de vrais apprenants nos différents algorithmes d'espacement de l'apprentissage de CC, nous avons décidé de simuler des trajectoires synthétiques, mais réalistes, d'apprenants auxquels nous avons assigné l'une ou l'autre de ces stratégies d'espacement. L'objectif de ce protocole est de pouvoir malgré tout bénéficier d'une première évaluation de leurs performances respectives, dans un environnement contrôlé et idéalisé.

Nous détaillons dans la sous-section 4.2.3 la manière dont nous générons les différents paramètres de nos simulations et nous présentons enfin en sous-section 4.2.4 les différentes métriques servant à comparer les performances des algorithmes d'espacement.

4.2.1 Modèle générateur de trajectoires d'apprentissage et d'oubli

Nous choisissons notre modèle DAS3H (CHOFFIN, POPINEAU et al., 2019), présenté au chapitre précédent, comme modèle simulateur de trajectoires d'apprenants à la fois pour (i) ses meilleures performances prédictives sur des données éducatives réelles par rapport à d'autres modèles d'apprenant et (ii) car c'est à notre connaissance le seul modèle d'apprenant qui intègre à la fois les relations entre les items et de multiples CC ainsi que l'effet d'oubli dans sa structure pour prédire la justesse des réponses futures des apprenants. Il nous permet donc d'extraire des informations sur la dynamique de maîtrise des CC en fonction de l'historique d'interactions des apprenants. Les paramètres de DAS3H offrent enfin une interprétation cognitive directe, ce qui permet de simuler plus facilement des trajectoires d'apprentissage réalistes et de contrôler les comportements d'apprentissage et d'oubli des apprenants simulés.

Nous choisissons de simuler selon le modèle DAS3H avec une dimension de plongement vectoriel de 0. Nous faisons ce choix parce que (i) nous avons montré au chapitre précédent que l'intégration de représentations vectorielles continues n'améliorait pas systématiquement les performances prédictives du modèle et (ii) pour des raisons d'interprétabilité des paramètres du modèle généré. En effet, si l'interprétation des paramètres de DAS3H est relativement aisée sans représentation vectorielle continue des caractéristiques, elle devient beaucoup plus complexe quand la dimension de plongement vectorielle est strictement positive : les paramètres se trouvent alors dans un espace multidimensionnel. Enfin, nous utilisons les mêmes fenêtres temporelles qu'au chapitre précédent : $\{1/24, 1, 7, 30, +\infty\}$ (en jours).

4.2.2 Description des simulations

Nous simulons un processus d'apprentissage et de révision simple mais réaliste, proche du plan d'expérience synthétique de M. M. KHAJAH et al. (2014) et réel de LINDSEY et al. (2014) et qui tire parti des effets d'*espacement* et de *test* de la littérature en sciences cognitives. Nous voulons qu'il reflète des situations d'enseignement réelles où des enseignants humains voudraient utiliser un algorithme d'espacement adaptatif dans leur classe pendant une session de révision dédiée, car c'est le cas d'usage principal que nous imaginons pour nos algorithmes. Néanmoins, notre protocole expérimental simple pourrait facilement être adapté à d'autres situations, à condition que l'optimisation de l'algorithme concerne la CC à faire réviser à un moment donné, et non le meilleur moment pour réviser la CC k (cf. section 1.2).

Algorithme 1 Processus d'apprentissage et de révision pour un seul apprenant

```

1: procédure LEARN AND REVIEW (stratégie de sélection de CC, stratégie de sélection d'item,  $K$ 
   CC, paramètres DAS3H,  $r$  items par session, délais de rétention)
2:   pour semaine de 1 à  $K$  faire                                ▷ Phase d'apprentissage et de révision
3:     Apprend initialement la CC de la semaine courante : ajoute 1 tentative à cette CC ;
4:     si semaine > 1 alors
5:       pour  $i$  de 1 à  $r$  faire
6:          $k^* \leftarrow$  SELECT SKILL (stratégie de sélection de CC)
7:          $j^* \leftarrow$  SELECT ITEM ( $k^*$ , stratégie de sélection d'item)
8:         Résout l'item  $j^*$  et met à jour l'historique de pratique de l'apprenant ;
9:         Mesurer la  $PMR$  à semaine ;
10:    pour tout  $\tau$  dans les délais de rétention faire                ▷ Phase de rétention
11:      Mesurer la  $PMR$  à  $K + \tau$  ;
12:    retourne  $PMR_A$  et  $PMR_R$ 
13: fin procédure

```

L'algorithme 1 décrit le processus d'apprentissage et de révision simulé que suit chaque apprenant : il génère une trajectoire d'apprentissage et de révision pour un seul apprenant selon une stratégie de sélection de CC et selon des paramètres de simulation générés. Nous détaillons dans ce qui suit le fonctionnement de cet algorithme.

L'apprenant commence sans aucune expérience connue dans aucune des K CC du programme qu'il a à acquérir et à retenir. Ses connaissances préalables potentielles sont synthétisées dans son paramètre α_s qui, combiné avec les difficultés de l'item et de la (ou des) CC δ_j et β_k , donne la probabilité que l'apprenant s n'ayant aucune pratique connue au préalable sur les CC mises en jeu par l'item j , puisse y répondre correctement (cf. équation 3.1).

Chaque semaine (c'est-à-dire tous les sept jours), une toute nouvelle CC est introduite à l'apprenant. Ainsi, la CC i est d'abord introduite et apprise à la semaine i . Contrairement aux instants t , nous faisons commencer les indices des semaines à 1¹ pour faciliter l'interprétation

1. La semaine 1 s'étend donc entre $t = 0$ et $t = 604800$ (en secondes).

Algorithme 2 Protocole expérimental de comparaison de stratégies sur des apprenants simulés

```

1: procédure SIMULATION(stratégies de sélection de CC, stratégies de sélection d'item,  $N$ 
   itérations,  $K$  CC,  $S$  apprenants,  $J$  items,  $r$  items par session, délais de rétention)
2:   pour  $i$  de 1 à  $N$  faire
3:     Générer la q-matrice;
4:     Générer les paramètres de DAS3H :  $\alpha_s, \beta_k, \delta_j, \theta_{k,w}$ ;
5:     pour tout apprenant dans  $\{\alpha_s\}_{s \in \llbracket 1, S \rrbracket}$  faire
6:       pour toute stratégie de sélection de CC faire
7:         pour toute stratégie de sélection d'item faire
8:            $\sigma \leftarrow$  LEARN AND REVIEW(stratégie de sélection de CC, stratégie de sélection
   d'item,  $K$ , paramètres DAS3H,  $r$ , délais de rétention)
9:           Effacer l'historique d'apprentissage ( $c_{s,k,w}$  et  $a_{s,k,w}$ ) de  $s$ ;
10:        Enregistrer les résultats de l'itération  $i$ ;
11:   retourne Métriques de performance moyennes  $\bar{\sigma}$  pour chaque stratégie de révision
12: fin procédure

```

de nos résultats. Lorsqu'une nouvelle CC est introduite, nous synthétisons cette connaissance théorique supplémentaire en ajoutant une unique tentative à la semaine w à son historique d'apprentissage² sur la CC $k_{courante}$.

Ensuite, juste après l'introduction de la nouvelle CC k , l'apprenant se voit présenter un nombre fixe (pour toute la simulation, identique pour chaque apprenant) r d'items de révision qu'il essaie alors de résoudre séquentiellement. Ces items sont utilisés pour revoir les CC introduites précédemment : la ou les CC sont d'abord choisies par l'heuristique d'espacement qui lui est associée (routine SELECT SKILL dans l'algorithme 1) et un item mettant en jeu la ou les CC choisies est enfin déterminé par la stratégie de sélection d'item (routine SELECT ITEM dans l'algorithme 1). Ni la CC qui a été introduite la même semaine, ni les CC suivantes (non encore introduites) ne peuvent être révisées. Par conséquent, les sessions de révision commencent à la semaine 2. L'apprenant résout alors l'item et on met à jour son historique d'interactions en fonction de sa réussite ou non. Nous supposons que les résultats binaires des apprenants aux items sont déterministes (c'est-à-dire que si $\mathbb{P}(\ll \text{réponse correcte} \gg) > 0,5$, alors la réponse de l'apprenant sera correcte), et que le temps pris pour résoudre un item ne change pas d'un apprenant (ou d'un item) à l'autre, contrairement à PAVLIK et ANDERSON (2008) : les apprenants se voient attribuer un nombre fixe d'items de révision à résoudre chaque semaine, indépendamment du temps nécessaire pour les résoudre.

La seule différence entre les parcours d'apprentissage de deux apprenants réside dans les items spécifiques que chacun d'entre eux résout pendant les sessions de révision : sinon, chaque apprenant suit le même processus d'apprentissage au même moment. Après cette première *phase d'apprentissage*, les apprenants simulés sont laissés sans rien réviser pendant la *phase de rétention*. À l'issue de cette période finale, la simulation s'arrête. Des mesures de performance des apprenants sont enregistrées tout au long des phases d'apprentissage et de rétention et sont utilisées pour comparer les stratégies de sélection des CC : voir ci-dessous dans la sous-section 4.2.4. L'algorithme 1 renvoie enfin les métriques PMR_A et PMR_R pour l'apprenant simulé.

Le protocole de simulation total est décrit dans l'algorithme 2. Nous détaillons son fonctionnement dans ce qui suit. Afin de mesurer la robustesse de nos résultats et de prendre en compte une grande diversité de scénarios d'apprentissage et de comportements d'oubli, nous exécutons N fois le processus de simulation, avec à chaque fois un groupe d'apprenants, d'items, de CC et une q-matrice différents et générés de manière aléatoire. De cette façon, nous pouvons également calculer des mesures de variabilité d'une simulation à l'autre. Pour com-

2. Cet historique est donc vide sur la CC k avant cette tentative initiale.

parer l'efficacité de nos heuristiques concurrentes d'espacement adaptatif, chaque apprenant au sein d'une simulation donnée est assigné à chaque stratégie de révision. Plus précisément, dans une exécution de la simulation, chaque apprenant suit le processus d'apprentissage et de rétention décrit dans l'algorithme 1 pour chaque heuristique d'espacement des révisions. La figure 4.1 schématise ce fonctionnement. Bien entendu, lorsqu'un apprenant entame un nouveau processus d'apprentissage, son historique de pratique antérieure est complètement remis à zéro. Nous obtenons en faisant cela un plan d'expérience *mixte* avec un facteur de variabilité inter-sujets, les différentes simulations et un facteur intra-sujets, les stratégies de révision. À la fin, l'algorithme 2 renvoie les métriques de performance de chaque stratégie, moyennées sur toutes les simulations et tous les apprenants.

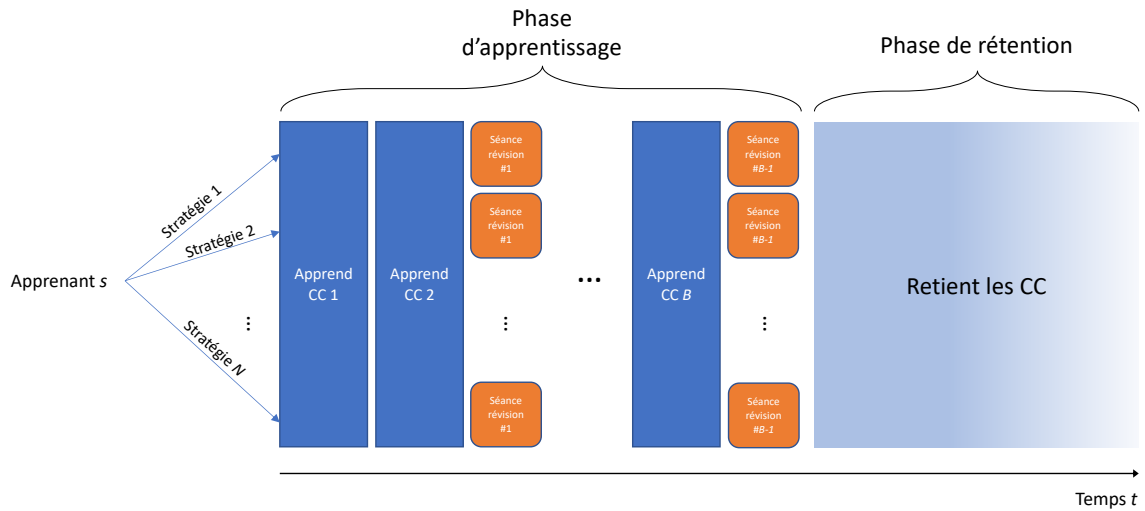


FIGURE 4.1 – Schéma du processus d'apprentissage et de mémorisation pour un apprenant s dans une exécution de la simulation

4.2.3 Paramètres des simulations

Paramètres générés

Génération des paramètres de DAS3H Pour simuler avec notre modèle DAS3H, nous choisissons d'échantillonner aléatoirement et de manière indépendante les paramètres des apprenants (α_s), des items (δ_j), des CC (β_k) et les paramètres d'apprentissage et d'oubli ($\theta_{k,w}$) à partir de distributions de probabilité spécifiées. Générer aléatoirement ces paramètres nous permet de contrôler les caractéristiques de nos simulations, et notamment les dynamiques d'apprentissage et d'oubli, par le biais des distributions de probabilité des paramètres. Cela nous permet également de prendre en compte une grande diversité de comportements d'apprentissage et d'oubli.

Par le passé, d'autres travaux ont également fait le choix de comparer leurs algorithmes d'espacement adaptatif sur des données simulées, à l'aide de modèles d'oubli et de paramètres échantillonnés aléatoirement (REDDY, LEVINE et al., 2017; YANG et al., 2020) ou fixés a priori (M. M. KHAJAH et al., 2014).

La table 4.1 donne plus de détails sur les distributions que nous avons choisies. Un exemple de valeurs échantillonnées est représenté sous la forme d'histogrammes dans la figure 4.2 : nous avons pour l'occasion généré 500 apprenants de niveau initial α_s , 200 items de difficulté δ_j , 10 CC de facilité β_k et 100 paramètres $\theta_{k,w}$ en tout pour ces CC. Ces paramètres $\theta_{k,w}$ décrivent la courbe d'apprentissage et d'oubli de la CC k (cf. section 3.3).

Paramètre	Distribution
α_s	$\mathcal{N}(0, 1)$
δ_j	$\mathcal{N}(1, 1)$
β_k	$\mathcal{N}(-1, 1)$
$\theta_{k,2w+1}, \theta_{k,2w+2}$	$\mathcal{U}_{[0,2]}$

TABLE 4.1 – Hypothèses de distribution des paramètres de nos simulations utilisant le modèle DAS3H. Les $\theta_{k,2w+1}$ et $\theta_{k,2w+2}$ choisis correspondent à la zone 1 dans la figure 3.2.

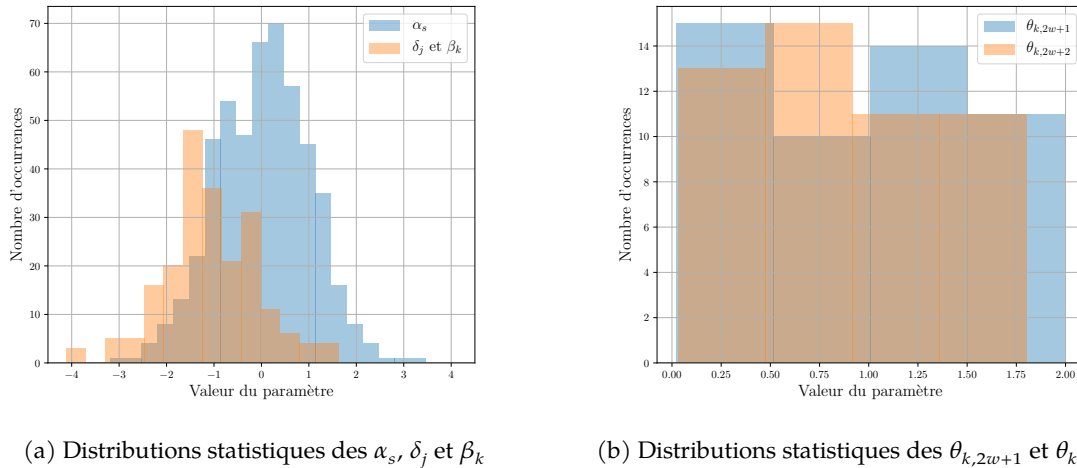


FIGURE 4.2 – Exemples de distributions statistiques des valeurs générées pour les $\alpha_s, \delta_j, \beta_k, \theta_{k,2w+1}$ et $\theta_{k,2w+2}$.

En termes d'apprentissage et de mémorisation, ces choix de distributions de probabilité pour les paramètres α_s, δ_j et β_k signifient que sans aucune pratique, un apprenant de niveau initial moyen aura très peu de chances (environ 0,12) de résoudre correctement un item de difficulté moyenne qui met en œuvre une unique CC, de difficulté moyenne elle aussi. Si toutefois l'apprenant parvenait à résoudre correctement l'item, cela pourrait vouloir dire soit que l'item ou la CC était particulièrement facile, soit que l'apprenant avait des connaissances préalables sur la CC abordée. Quand le nombre de CC supplémentaires qu'utilise un item augmente, la probabilité de bonne réponse de l'apprenant peut augmenter ou diminuer avec l'addition des β_k dans l'équation 3.1, selon le signe de chaque β_k .

Concernant les paramètres $\theta_{k,w}$: nous avons choisi leurs distributions de probabilité de manière à ce qu'échouer à appliquer correctement une CC dans un item améliore la probabilité de bonne réponse future sur cette CC, mais moins que réussir à appliquer cette CC. Cela signifie que $\theta_{k,2w+1} > 0$ et $\theta_{k,2w+2} > 0$ (zone 1 dans la figure 3.2). C'est également le comportement d'oubli qui a été choisi par exemple par REDDY, LEVINE et al. (2017), UPADHYAY et al. (2018) et TABIBIAN et al. (2019). Nous voulions générer des courbes d'apprentissage et d'oubli diversifiées, exhibant une large palette de comportements : nous avons donc choisi une loi uniforme $\mathcal{U}_{[0,2]}$ pour ces paramètres. Pour tenir compte de contextes d'apprentissage encore plus divers, nous avons également conduit des analyses complémentaires dans la discussion de ce chapitre en générant les paramètres $\theta_{k,w}$ à partir d'une zone différente du diagramme présenté dans la figure 3.2.

Génération des q-matrices Il était aussi nécessaire de générer aléatoirement les q-matrices pour obtenir les relations entre items et CC. Rappelons tout d'abord que l'indice d'une CC correspond à son ordre d'introduction dans le programme. Nous avons généré pour chaque

CC k le même nombre d'items de révision : ensuite, nous avons ajouté un nombre aléatoire d'étiquettes de CC additionnelles à chacun de ces items. Ainsi, un item associé à la CC k met nécessairement en jeu la CC k mais peut également mettre en jeu d'autres CC, antérieures dans le programme d'apprentissage. Plus précisément, le nombre de nouvelles CC ajoutées pour un item associé à la CC k était à chaque fois généré aléatoirement selon $\mathcal{U}_{\llbracket 0; \min(k-1, 2) \rrbracket}$. Une q-matrice générée aléatoirement selon notre protocole est présentée dans la figure 4.3. Les items de cette q-matrice comportent en moyenne 1,88 étiquettes de CC.

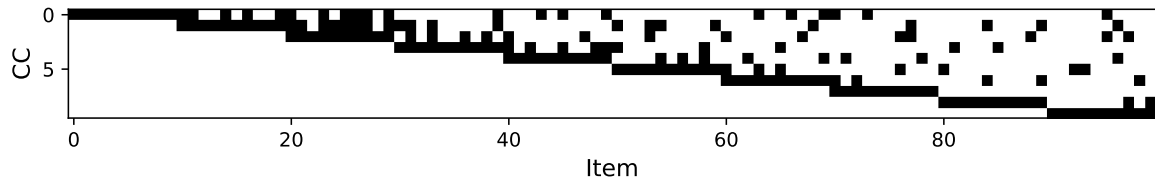


FIGURE 4.3 – Transposée d'une q-matrice aléatoire (10 CC, 10 items par CC). Chaque carré noir indique que l'item de la colonne met en jeu la CC de la ligne. Notons que les items ne peuvent pas inclure de CC postérieure à la CC courante, ce qui explique la forme triangulaire supérieure de la matrice. Nous avons transposé la q-matrice ici pour des raisons de mise en page.

Cette procédure génère des q-matrices réalistes car :

- elle imite un programme d'enseignement et sa séquence d'introduction des différentes CC;
- chaque CC apparaît au moins k fois dans une q-matrice et un item peut posséder au maximum 3 étiquettes de CC;
- si l'on veut faire réviser la CC k et qu'elle a déjà été introduite, il est toujours possible de trouver un item qui met en jeu k et *seulement* des CC antérieures.

Paramètres fixés

Dans nos expériences, nous fixons :

- N le nombre d'exécutions du processus de simulation à 100;
- S le nombre d'apprenants par exécution à 500;
- K le nombre de semaines d'apprentissage et de CC du programme à 10;
- le nombre de semaines de rétention à 6;
- r le nombre d'items à sélectionner par session de révision à 3;
- c le nombre d'items générés par CC à 20;
- le nombre maximal de CC additionnelles par item à 2.

4.2.4 Métriques de performance

Nous décrivons dans cette sous-section les métriques spécifiques auxquelles nous nous sommes intéressés pour mesurer la performance des apprenants et comparer l'efficacité des stratégies.

Nous nous sommes intéressés à la mesure et à la comparaison de la Probabilité Moyenne de Réussite (PMR) sur l'ensemble des CC, pour tous nos apprenants synthétiques. Cette PMR est inspirée par HUNZIKER et al. (2019), KANG et al. (2014) et YANG et al. (2020). Ici, nous donnons la formule du calcul de la PMR pour un seul apprenant³ s , au temps t :

3. Rappelons que chaque apprenant simulé est assigné à toutes les stratégies ; cette métrique PMR est donc calculée pour chaque apprenant, pour chaque stratégie de révision.

$$PMR_{s,t} = \sum_{k=1}^K \mathbb{P}(Y_{s,k,t} = 1) \quad (4.1)$$

avec :

- s l'indice de l'apprenant ;
- K le nombre de CC ;
- t l'instant auquel la PMR est calculée ;
- $Y_{s,k,t}$ la justesse de la réponse de l'apprenant s sur un item de difficulté moyenne (inconnue) utilisant *seulement* la CC k à l'instant t .

Puisque DAS3H ne renvoie pas directement de probabilité de bonne réponse pour une CC donnée, pour calculer la PMR , nous remplaçons dans le modèle la difficulté de l'item δ_j par l'espérance de la distribution de probabilité utilisée pour échantillonner les difficultés de l'item (dans notre cas, -1). Cela nous permet d'obtenir la probabilité de rappel désirée. Nous procédons en réalité comme pour les courbes d'apprentissage idéalisées de GOUTTE et al. (2018).

Cependant, et contrairement à d'autres travaux antérieurs (M. M. KHAJAH et al., 2014), nous soutenons qu'évaluer les performances des apprenants en n'utilisant que quelques estimations ponctuelles ne correspond pas de manière adéquate à notre objectif d'optimisation de la rétention de la mémoire à long terme. Pour cette raison, nous introduisons une version alternative de la métrique PMR qui mesure les performances des apprenants pendant une période donnée :

$$PMR_{s,t_0:T} = \frac{1}{T - t_0 + 1} \sum_{t=t_0}^T PMR_{s,t} \quad (4.2)$$

avec :

- t_0 le début de la période ;
- T la fin de la période ;

Intuitivement, cette $PMR_{s,t_0:T}$ mesure la capacité de l'apprenant s à appliquer l'ensemble des CC du programme entre les instants t_0 et T . Puisque nous ne mesurons la PMR qu'à des instants discrets, cette métrique n'est qu'une estimation de la véritable probabilité moyenne de bonne réponse au fil du temps. Notons qu'il existe une relation affine, soulignée par HUNZIKER et al., entre cette métrique PMR et l'aire discrète sous la courbe d'évolution temporelle de la probabilité de bonne réponse de l'apprenant s pour la CC k .

Dans nos expériences, nous comparons les performances des heuristiques d'espacement adaptatif en moyennant les PMR sur tous les apprenants assignés à cette stratégie, dans toutes les exécutions de la simulation. Pour réduire le nombre de notations, et sauf indication contraire, nous nous reposons dans ce qui suit sur la définition suivante de la PMR :

Définition 5 – Probabilité Moyenne de Réussite (PMR) d'une heuristique. La PMR d'une heuristique h , notée PMR_t^h , est égale à la moyenne des PMR individuelles des apprenants assignés à cette heuristique, sur toutes les simulations.

$$PMR_t^h = \frac{1}{|S_h| \times |R|} \sum_{r \in R} \sum_{s \in S_h} PMR_{s,r,t} \quad (4.3)$$

avec :

- S_h l'ensemble des apprenants associés à l'heuristique h , et $|S_h|$ son cardinal ;
- R l'ensemble des simulations et $|R|$ son cardinal ;

Plus précisément, nous avons calculé cette métrique pour deux périodes : PMR_A (apprentissage) et PMR_R (rétention). La PMR_R formalise mathématiquement le critère à optimiser que nous avons défini dans la sous-section 1.2.3.

Définition 6 – PMR_A d'une heuristique. La PMR_A^h mesure les performances des apprenants assignés à une heuristique h pendant la période d'apprentissage :

$$PMR_A^h = \frac{1}{10} \sum_{t=1}^{10} PMR_t^h \quad (4.4)$$

Définition 7 – PMR_R d'une heuristique. La PMR_R^h mesure les performances des apprenants assignés à une heuristique h pendant la période de rétention :

$$PMR_R^h = \frac{1}{6} \sum_{t=11}^{16} PMR_t^h \quad (4.5)$$

Dans ce qui suit, lorsque cela n'entraîne pas de confusion, nous retirons l'indice h de ces deux métriques.

Pendant la phase d'apprentissage, pour collecter les mesures des performances de l'apprenant, nous calculons la PMR chaque semaine, juste après que le dernier item de la semaine soit présenté à l'apprenant⁴. Pendant la phase de rétention, nous calculons la PMR chaque semaine entre la fin de la phase d'apprentissage et le délai de rétention maximum que nous avons spécifié dans l'algorithme. Dans nos simulations, ce délai est de 6 semaines. La PMR_A est plus intéressante pour notre problématique que la PMR_R car elle mesure la rétention après la dernière session de révision ; cependant, l'analyse de la PMR_A présente aussi l'intérêt de montrer l'impact d'une stratégie sur la performance des apprenants *pendant* l'apprentissage.

La PMR telle que nous la définissons pondère tous les instants et les CC de la même manière, mais il serait très simple d'affecter des coefficients de pondération différents pour favoriser certaines CC ou certaines sous-périodes de la phase de rétention. Cela rend la PMR modulable. Il s'agit en somme d'une mesure synthétique et flexible de la maîtrise d'un ensemble de CC sur une période future.

4.3 Stratégies de sélection

La sélection d'items de révisions dans notre cadre de recherche est plus complexe que pour l'apprentissage de connaissances simples, comme des mots de vocabulaire dans une langue étrangère. En effet, nous voulons choisir le meilleur item pour réviser un ensemble de CC sous-jacentes. La sélection d'un item nécessite donc deux étapes : la sélection de la (ou des) CC, puis la sélection d'un item qui met en œuvre cette (ou ces) CC choisies. Dans la suite de cette section, nous détaillons d'abord les différentes heuristiques de sélection de CC que nous avons développées, implémentées et comparées dans nos expériences. En particulier, nous présentons une nouvelle procédure gloutonne de sélection de *sous-ensemble* de CC. Enfin, nous décrivons les deux stratégies de sélection d'item que nous avons considérées dans nos expériences.

Les deux définitions suivantes concernent les notions de CC et d'item *acceptables*, essentielles pour cette section.

Définition 8 – CC acceptable. Une CC *acceptable* est une CC qui a déjà été vue par l'apprenant par le passé, exception faite de la semaine courante.

Définition 9 – Item acceptable. Un item *acceptable* est un item qui utilise la (ou les) CC choisie(s) et seulement des CC *acceptables*.

4. Donc, pendant la phase d'apprentissage, notre métrique implique dans son calcul la probabilité de bonne réponse sur des CC qui n'ont pas encore été vues par l'apprenant.

4.3.1 Stratégies de sélection de CC

La procédure `SELECT SKILL` de l'algorithme 1 sélectionne la ou les CC à faire réviser à un instant t . Nous rappelons à notre lecteur que dans nos simulations, la CC i est acquise initialement par chaque apprenant à la semaine i . Dans ce qui suit, k^* désigne l'indice de la CC choisie par l'heuristique.

Sélection d'une unique CC

Voici les stratégies de sélection de CC que nous proposons :

Baseline Pas de révision.

Random Choisit une CC uniformément au hasard parmi les CC acceptables.

$$\text{Échantillonne la CC } k^* \text{ selon } \mathcal{U}_{\llbracket 1, k_{actuelle}-1 \rrbracket} \quad (4.6)$$

avec $k_{actuelle}$ la CC qui a été introduite la semaine courante. $k_{actuelle} \geq 2$ parce que les apprenants commencent leurs révisions seulement à partir de la semaine 2 dans nos simulations.

μ -back Choisit la CC qui a été vue μ semaines auparavant, c'est-à-dire μ blocs en arrière. μ peut prendre des valeurs entières strictement positives. Cette stratégie est adaptée de la littérature en sciences cognitives et de M. M. KHAJAH et al. (2014). C'est une stratégie non-adaptative qui fournit le même planning de révision de CC pour tous les apprenants partageant le même paramètre μ et qui ne s'adapte pas aux performances des apprenants. Cependant, sa force réside justement dans sa simplicité, et elle a l'avantage de ne pas faire appel à un modèle prédictif de l'apprenant. Elle est de plus directement utilisable par un enseignant humain, par exemple.

$$\text{Choisit la CC } k^* = \max(k_{actuelle} - \mu, 1) \quad (4.7)$$

θ -threshold Choisit la CC pour laquelle la probabilité actuelle de réponse correcte est la plus proche d'une valeur fixée $\theta \in [0, 1]$. Cette stratégie est aussi adaptée de la littérature en sciences cognitives et de M. M. KHAJAH et al. (2014). Intuitivement, cette stratégie cherche à faire réviser la CC qui est sur le point d'être oubliée : elle est donc alignée avec la notion introduite par BJORK de « difficulté désirable » (BJORK, 1994). Cette stratégie a besoin d'un modèle prédictif de l'apprenant capable de renvoyer la probabilité immédiate que l'apprenant réponde correctement à un item de difficulté moyenne mettant en jeu la CC k .

$$\text{Choisit la CC } k^* \text{ t.q. } k^* \in \underset{k \in \llbracket 1, k_{actuelle}-1 \rrbracket}{\operatorname{argmin}} |\mathbb{P}(Y_{s,k,t_{actuel}} = 1) - \theta| \quad (4.8)$$

avec t_{actuel} l'instant courant.

Greedy Cette stratégie est adaptée de l'article (HUNZIKER et al., 2019) à notre problématique de sélection de CC. Elle consiste à choisir la CC k pour laquelle le gain de performance estimé de l'apprenant, en espérance, durant la période de rétention et sur toutes les CC, serait le plus fort après avoir sélectionné la CC k . Comme θ -threshold, Greedy requiert également un modèle prédictif de l'apprenant, mais ici, les prédictions du modèle couvrent une période plus grande que la simple probabilité immédiate de bonne réponse. Ainsi, le modèle devra être particulièrement fiable puisque la fonction objectif à optimiser inclut des probabilités de bonne réponse à plusieurs moments dans le futur. Cependant et contrairement à θ -threshold, Greedy ne nécessite pas de choisir un hyperparamètre. Comme nous souhaitons optimiser la mémorisation à long terme, nous choisissons un horizon temporel différent de celui choisi

Stratégie de sélection	Domaine
<i>Baseline</i>	\emptyset
<i>Random</i>	\emptyset
μ -back	\mathbb{N}^{*+}
θ -threshold	$[0, 1]$
<i>Greedy</i>	\emptyset

TABLE 4.2 – Domaines pour les paramètres des différentes stratégies de sélection de CC

par HUNZIKER et al. (2019) : au lieu d'optimiser la *PMR* sur toutes les périodes, nous nous concentrons sur la période de rétention.

$$\text{Choisit la CC } k^* \text{ t.q. } k^* \in \underset{k \in [1, k_{\text{actuelle}} - 1]}{\operatorname{argmax}} U(k | \sigma_{1:t_{\text{actuel}}-1}, y_{1:t_{\text{actuel}}-1}, \tau_0) \quad (4.9)$$

avec :

$$U(k | \sigma_{1:t-1}, y_{1:t-1}, \tau_0) = \mathbb{E}_{y_t} [f(\sigma_{1:t-1} \oplus k, y_{1:t-1} \oplus y_t; \tau_0) - f(\sigma_{1:t-1}, y_{1:t-1}; \tau_0)] \quad (4.10)$$

$$f(\sigma_{1:t}, y_{1:t}; \tau_0) = \frac{1}{K(T - \tau_0 + 1)} \sum_{k=1}^K \sum_{\tau=\tau_0}^T \mathbb{P}(Y_{s,k,\tau} = 1 | \sigma_{1:t}, y_{1:t}) \quad (4.11)$$

et :

- $\sigma_{1:t}$ la séquence des CC rencontrées par s jusqu'à t (inclus) ;
- $y_{1:t}$ les résultats binaires de ces interactions ;
- τ_0 le début de la période de rétention ;
- T la fin de la période de rétention ;
- K le nombre de CC au total.

L'espérance est calculée sur la justesse (inconnue, donc) de la réponse de l'apprenant s répondant à un item de difficulté moyenne, étiqueté simplement par la CC k au temps t . Le symbole \oplus représente quant à lui l'opération de concaténation. Intuitivement, f représente la probabilité de bonne réponse sur toutes les CC et tous les instants futurs de la période de rétention, sachant l'historique courant de résultats de l'apprenant s : $(\sigma_{1:t}, y_{1:t})$.

Mais puisque nous faisons l'hypothèse que nos CC sont indépendantes les unes des autres, pratiquer une CC k a juste un impact sur la probabilité de bonne réponse future sur *cette* CC k . En outre, nos CC sont pondérées uniformément dans notre fonction objectif U . Pour cette raison, nous pouvons remplacer notre fonction f par une version plus simple et moins coûteuse computationnellement que celle-ci, dans laquelle nous avons retiré la deuxième somme sur les CC :

$$f^*(\sigma_{1:t}, y_{1:t}; k, \tau_0) = \frac{1}{T - \tau_0 + 1} \sum_{\tau=\tau_0}^T \mathbb{P}(Y_{s,k,\tau} = 1 | \sigma_{1:t}, y_{1:t}) \quad (4.12)$$

La table 4.2 synthétise les domaines des paramètres de toutes les stratégies de sélection que nous avons employées.

Deux stratégies de révision doivent avoir accès à la probabilité qu'un apprenant réponde correctement à un item mettant en jeu une CC donnée, maintenant ou à l'avenir : θ -threshold et *Greedy* (et leurs versions multi-CC). Ainsi, elles auraient besoin dans une situation réelle de disposer d'un jeu de données d'interactions passées d'apprenants répondant aux items et peut-être même d'interactions passées des apprenants pour qui l'on cherche à adapter

les plannings de révision. C'est un des inconvénients de ces heuristiques : elles nécessitent d'avoir un modèle fiable de l'apprenant pour fonctionner correctement : ce modèle devra être réentraîné périodiquement pour prendre en compte les dernières données produites par les apprenants. Cela rend également ces deux heuristiques plus complexes à utiliser et à mettre en œuvre. Dans nos expériences, nous avons choisi d'utiliser le modèle DAS3H pour obtenir ces estimations de probabilité. Dans (LINDSEY et al., 2014), les auteurs estiment les paramètres du modèle pendant l'expérience elle-même : puisque leur modèle DASH utilise la même courbe d'apprentissage et d'oubli pour chaque item, ils sont en mesure d'avoir des échantillons pour chaque paramètre à estimer dès le début de l'expérience. Dans notre cas, puisque nous introduisons chaque semaine une nouvelle CC dans le programme (cf. le protocole expérimental dans la sous-section 4.2), cela donnerait de mauvaises estimations des $2W$ paramètres pour chaque CC dans le modèle DAS3H (avec W le nombre de fenêtres temporelles dans DAS3H). Ainsi, pour évaluer comment le fait d'avoir des paramètres estimés au lieu des vrais paramètres du modèle affecte les performances de ces stratégies, nous avons implémenté des versions estimées de θ -threshold et Greedy. Dans ce cas, un plan d'expérience spécifique nous permet d'utiliser des paramètres estimés au lieu des vrais paramètres du modèle et d'évaluer la sensibilité de ces stratégies aux paramètres estimés (voir à ce sujet la discussion dans la section 4.4). Notons toutefois que les heuristiques Greedy et θ -threshold sont agnostiques au modèle de l'apprenant choisi et pourraient être utilisées directement avec tout autre modèle de l'apprenant, à condition qu'ils puissent produire des prédictions précises pour les probabilités de bonne réponse au niveau des CC.

Nous avons choisi de ne pas comparer nos algorithmes d'espacement au système de Leitner (LEITNER, 1972; REDDY, LABUTOV et al., 2016) parce que nous n'avons pas réussi à l'étendre au problème de sélection de CC. Le principal problème que nous avons rencontré était en fait qu'un item pouvait utiliser plusieurs CC en même temps : il était alors difficile de mettre à jour les piles dans le système puisqu'une CC pouvait se trouver à plusieurs endroits dans les différentes piles, contrairement au système de Leitner d'origine dans lequel une *flashcard* n'apparaît qu'une unique fois. De plus, nous n'avons pas comparé nos stratégies de sélection à l'algorithme SuperMemo (WOZNIAK et al., 1994) car il n'était pas adapté à notre cadre expérimental avec des sessions de révision fixes à une semaine d'intervalle ni à notre simulateur de trajectoires d'apprenant (qui renvoie des probabilités de bonne réponse).

Sélection d'un sous-ensemble de CC

L'une des contributions principales de ce chapitre est que nous développons une nouvelle procédure gloutonne⁵ pour sélectionner le sous-ensemble de CC le plus prometteur à faire réviser à l'apprenant, au lieu de l'unique CC la plus prometteuse. Cette procédure est générique et peut s'adapter à n'importe quelle stratégie de sélection, à la condition que celle-ci puisse fournir un classement de priorité des CC à présenter à l'instant t . Par exemple, elle ne peut pas s'appliquer à μ -back car μ -back ne produit pas de tel classement.

L'une des différences principales de notre cadre de recherche avec celui des autres travaux sur les algorithmes d'espacement adaptatif est que les items utilisent potentiellement plusieurs CC en même temps : ces informations sont synthétisées dans une q -matrice binaire spécifiée par l'expert du domaine de connaissances (TATSUOKA, 1983). Nous faisons l'hypothèse que le fait de faire pratiquer à un apprenant un item qui met en jeu plusieurs CC en même temps améliorera la maîtrise de cet apprenant dans *chacune* de ces CC mises en jeu. C'est cette hypothèse qui nous conduit à développer la procédure gloutonne de sélection de sous-ensemble de CC, dont le fonctionnement est détaillé dans l'algorithme 3.

5. Notons que cette procédure, bien que qualifiée de « gloutonne » est bien *différente* de l'heuristique appelée Greedy. Greedy sélectionne une CC à faire réviser à un instant t tandis que la procédure dont nous parlons sélectionne, à partir d'une heuristique compatible de sélection de CC unique, le sous-ensemble de CC optimal à faire réviser à un instant t .

Algorithme 3 Procédure de sélection gloutonne d'un sous-ensemble de CC

```

1: procédure SELECT MULTI-KCs(stratégie de sélection de CC, semaine courante  $k$ , q-matrice
   Q)
2:    $\sigma \leftarrow \text{RANKKCs}(\text{stratégie de sélection de CC}, k)$             $\triangleright$  Classe les CC acceptables
3:    $CC\text{choisies} \leftarrow \{\}$ 
4:   pour  $\sigma_i$  dans  $\sigma$  faire
5:      $\Lambda \leftarrow \text{GETITEMSLIST}(Q, CC\text{choisies} \cup \{\sigma_i\}, k)$ 
6:     si  $\Lambda \neq \{\}$  alors
7:        $CC\text{choisies} \leftarrow CC\text{choisies} \cup \{\sigma_i\}$ 
8:    $\Lambda \leftarrow \text{GETITEMSLIST}(Q, CC\text{choisies}, k)$ 
9:    $j^* \leftarrow \mathcal{U}_\Lambda$ 
10:  retourne  $j^*$ 
11: fin procédure

```

Tout d'abord, la procédure appelle la routine `RANKKCs`, qui classe les indices de CC acceptables par ordre décroissant de priorité de révision, selon la stratégie de sélection de CC choisie. La procédure parcourt ensuite les indices de CC du classement σ dans l'ordre : s'il existe des items acceptables qui mettent en jeu l'intégralité des CC déjà choisies et la CC considérée, alors celle-ci est ajoutée à la liste des CC choisies ; sinon elle ne l'est pas, et on passe à la CC suivante. `GETITEMSLIST(Q, L)` retourne pour cela la liste des items acceptables de la q-matrice Q qui mettent en jeu toutes les CC de la liste L . La sélection s'arrête lorsque la liste des CC a été parcourue en entier. Enfin, l'algorithme choisit uniformément au hasard un item parmi la liste des items acceptables qui mettent en œuvre l'ensemble des CC choisies⁶ : contrairement aux procédures de sélection d'unique CC, cette procédure retourne donc un item, et pas une CC. Il s'agit d'un critère de sélection glouton. En général, les items sélectionnés avec ces stratégies utilisent au moins deux CC, mais ils peuvent tout à fait ne pas en mettre plus en jeu (s'il n'y a pas d'item disponible qui réponde à tous les critères). Selon la q-matrice générée et l'item choisi, les stratégies de sélection d'une seule CC peuvent elles aussi faire réviser plusieurs CC en même temps, mais les autres CC sont alors sélectionnées aléatoirement.

Nous proposons donc deux versions multi-CC de nos heuristiques, une pour θ -threshold et une pour *Greedy*. C'est la routine `RANKKCs` qui les distingue :

θ -threshold (multi.) Applique la procédure `SELECT MULTI-KCs` avec la routine `RANKKCs` qui classe les CC par ordre décroissant de proximité (en valeur absolue) avec la valeur $\theta \in [0, 1]$.

***Greedy* (multi.)** Applique la procédure `SELECT MULTI-KCs` avec la routine `RANKKCs` qui classe les CC par ordre décroissant de gain attendu f^* (cf. équation 4.3.1). Puisque les CC sont indépendantes les unes des autres, nous pouvons sélectionner itérativement chaque nouvelle CC à ajouter dans le sous-ensemble sélectionné ; grâce à cette propriété également, nous n'avons pas à recalculer le classement chaque fois que nous ajoutons une CC à l'ensemble des CC choisies.

Pour faciliter la tâche de notre lecteur, nous rassemblons les différentes heuristiques de sélection de CC que nous avons utilisées dans les expériences de la section 4.4 dans la table 4.3. Nous indiquons également les principales caractéristiques de ces algorithmes dans cette table.

Enfin, nous utilisons différentes grilles de valeurs d'hyperparamètres pour les trois heuristiques avec hyperparamètres θ -threshold, θ -threshold (multi.) et μ -back :

6. Il n'y a pas de risque que cette liste soit vide car la 1^{re} CC du classement est nécessairement ajoutée et on n'ajoute ensuite des CC que s'il existe des items disponibles pour toutes les faire réviser.

Heuristique	Modèle	Adaptatif	Hyperparamètre	Multi-CC	Horizon temporel	Difficulté de mise en œuvre
<i>Random</i>	Sans	Non	Sans	Non	Aucun	+
<i>μ-back</i>	Sans	Non	Avec	Non	Aucun	+
<i>θ-threshold</i>	Avec	Oui	Avec	Non	Court	++
<i>θ-threshold (multi.)</i>	Avec	Oui	Avec	Oui	Court	+++
<i>Greedy</i>	Avec	Oui	Sans	Non	Long	+++
<i>Greedy (multi.)</i>	Avec	Oui	Sans	Oui	Long	++++

TABLE 4.3 – Tableau comparatif des caractéristiques des différentes heuristiques de sélection de CC utilisées dans nos algorithmes d’espacement adaptatif décrits dans la section 4.3. La colonne « Horizon temporel » indique l’horizon temporel considéré par l’heuristique pour sa décision. Plus le nombre de « + » de la colonne « Difficulté de mise en œuvre » est élevé, plus l’heuristique est difficile à mettre en œuvre.

- *θ -threshold* et *θ -threshold (multi.)* : 11 θ qui séparent l’intervalle $[0,1]$ en 10 intervalles de même taille;
- *μ -back* : $\{1,2,3\}$

Dans les résultats de nos expériences, sauf indication contraire, nous indiquons uniquement les performances pour l’hyperparamètre optimal, c’est-à-dire celui qui optimise la PMR_R .

4.3.2 Stratégies de sélection d’item

La procédure `SELECT ITEM` de l’algorithme 1 sélectionne, à partir d’une CC ou d’un ensemble de CC, l’item à présenter à un apprenant.

Après la sélection de la CC ou de l’ensemble de CC, nous devons ensuite choisir l’item précis qui sera présenté à l’apprenant s . Deux stratégies de sélection d’item sont proposées ici et étudiées dans nos expériences :

- *Random CC* : choisit un item uniformément au hasard parmi l’ensemble des items acceptables.
- *Max CC* : choisit un item au hasard parmi l’ensemble des items acceptables qui ont le *plus grand* nombre de CC. Dans la réalité, une telle stratégie consisterait à choisir l’un des items les plus complexes, par exemple un problème en mathématiques.

Dans les expériences que nous décrivons ensuite, nous utilisons la stratégie de sélection d’item *Random CC*. La stratégie *Max CC* est, quant à elle, comparée à *Random CC* dans la discussion (sous-section 4.5.1).

4.4 Résultats

Dans cette section, nous décrivons les résultats de nos expériences de comparaison de stratégies d’espacement adaptatif de l’apprentissage de CC. Le code Python pour reproduire nos résultats a été publié sur la plateforme GitHub⁷.

Nos principaux résultats, suivant le protocole expérimental décrit dans la section 4.2, sont présentés dans les tables 4.4 et 4.5. Nous calculons les métriques PMR_A et PMR_R pour chaque trajectoire d’apprentissage et la moyenne des mesures de performance de chaque couple (stratégie de sélection des CC, paramètre de stratégie) pour obtenir ces résultats. Les erreurs types de la moyenne sont également indiquées. Pour les stratégies *μ -back* et *θ -threshold*, nous ne présentons les résultats que pour les paramètres μ et θ qui maximisent la métrique PMR_R , et la colonne *Meilleur paramètre* donne ce paramètre. Des analyses complémentaires sur l’impact de ces paramètres sur les performances des heuristiques sont données en sous-section 4.5.6.

7. https://github.com/BenoitChoffin/multiskill_adaptive_spacing

Les distributions des scores PMR des apprenants simulés, par période, sont représentées sur la figure 4.4.

Contrairement à M. M. KHAJAH et al. (2014), nous ne pouvons pas calculer le planning optimal exact (c'est-à-dire le planning qui maximise la PMR_R des apprenants) de révisions à chaque nouvelle exécution de la simulation. Ceci provient du nombre élevé de combinaisons de CC que permet notre cadre de recherche à multiples CC. Ainsi, si l'on note :

- K le nombre de CC à apprendre en tout ;
- r le nombre d'items à sélectionner par session de révision ;
- c le nombre d'items générés par CC.

et que l'on suppose que les CC sont ordonnées dans le temps et qu'un apprenant peut réviser plusieurs fois le même item, alors, le nombre de plannings différents est de :

$$(r \times c) \times (r \times (c \times 2)) \times \dots \times (r \times (c \times K)) = \prod_{i=1}^K (r \times c \times i) \quad (4.13)$$

En fixant comme dans nos expériences K à 10, r à 3 et c à 20, on obtient une quantité très importante de plannings différents : cela n'est évidemment pas possible à évaluer. On pourrait réduire le nombre de combinaisons en considérant le nombre de combinaisons différentes de CC dans la q-matrice au lieu du nombre d'items mais cela ne réduirait pas suffisamment le nombre de plannings différents. Pour pouvoir tout même permettre une comparaison plus lisible des performances des heuristiques, nous indiquons dans la table 4.5 les performances de chaque stratégie de sélection des CC, relatives à la stratégie de sélection *Random*. Appelons :

- $PMR_{s,t,i}^h$ le score PMR de l'apprenant s à l'instant t avec la stratégie h dans la simulation i ;
- $PMR_{s,t,i}^{random}$ le score PMR de l'apprenant s à l'instant t avec la stratégie *Random* dans la simulation i ;
- $\rho_{s,t,i}^h$ le score PMR relatif de l'apprenant s à l'instant t avec la stratégie h dans la simulation i .

Alors nous définissons :

$$\rho_{s,t,i}^h = \frac{PMR_{s,t,i}^h - PMR_{s,t,i}^{random}}{PMR_{s,t,i}^{random}} \quad (4.14)$$

Le score relatif ρ exprime donc à quel point la performance de la stratégie h est meilleure (ou moins bonne) que celle de la stratégie *Random*, en pourcentage de la PMR de *Random*. Dans ce qui suit, nous n'indiquons pas les résultats de la stratégie de sélection de CC *Random* dans ces tables, car ils sont par définition égaux à 0. Ensuite, comme pour les scores PMR , les scores relatifs sont moyennés par période (apprentissage et rétention) entre les apprenants et les simulations. Il convient de noter que chaque score PMR relatif est calculé *durant* les simulations, puis moyenné par la suite : cela explique pourquoi le calcul de cette mesure à partir de la table 4.4 ne donne pas exactement les mêmes résultats que dans la table 4.5.

Stratégie de sélection	Meilleur paramètre	PMR_A	PMR_R
Pas de révision	\emptyset	$0,318 \pm 0,001$	$0,297 \pm 0,001$
<i>Random</i>	\emptyset	$0,567 \pm 0,000$	$0,695 \pm 0,001$
<i>μ-back</i>	1	$0,589 \pm 0,000$	$0,753 \pm 0,001$
<i>θ-threshold</i>	0,4	$0,599 \pm 0,000$	$0,754 \pm 0,001$
<i>θ-threshold (multi.)</i>	0,2	$0,604 \pm 0,000$	$0,814 \pm 0,000$
<i>Greedy</i>	\emptyset	$0,579 \pm 0,000$	$0,759 \pm 0,001$
<i>Greedy (multi.)</i>	\emptyset	$0,592 \pm 0,000$	$0,816 \pm 0,000$

TABLE 4.4 – Comparaison des métriques de performance PMR_A et PMR_R entre toutes les stratégies de sélection des CC. La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.

Stratégie de sélection	Meilleur paramètre	PMR_A relative	PMR_R relative
Pas de révision	\emptyset	$-0,404 \pm 0,001$	$-0,593 \pm 0,001$
<i>μ-back</i>	1	$0,036 \pm 0,000$	$0,093 \pm 0,000$
<i>θ-threshold</i>	0,4	$0,050 \pm 0,000$	$0,089 \pm 0,000$
<i>θ-threshold (multi.)</i>	0,2	$0,059 \pm 0,000$	$0,189 \pm 0,001$
<i>Greedy</i>	\emptyset	$0,017 \pm 0,000$	$0,100 \pm 0,000$
<i>Greedy (multi.)</i>	\emptyset	$0,039 \pm 0,000$	$0,194 \pm 0,000$

TABLE 4.5 – Comparaison des métriques de performance PMR_A et PMR_R relatives à *Random*, entre toutes les stratégies de sélection des CC. La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des étudiants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.

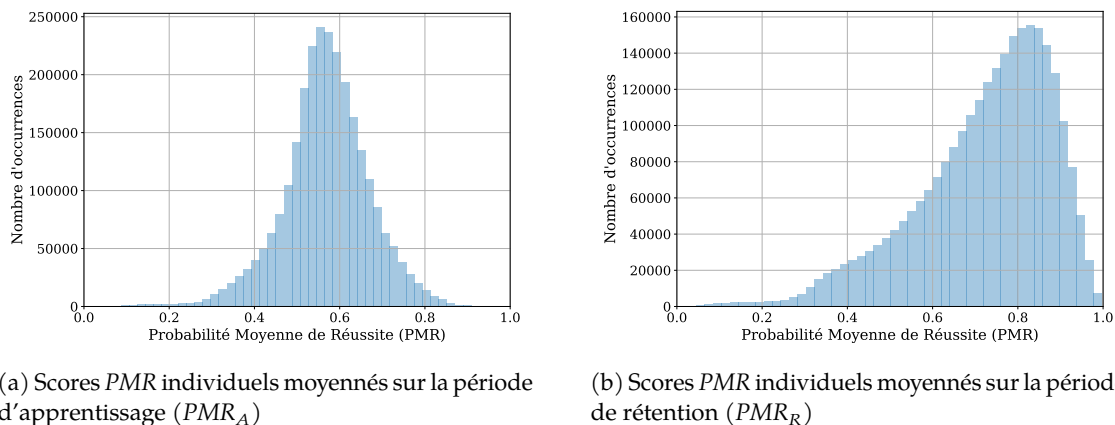


FIGURE 4.4 – Distributions des scores PMR individuels durant la période d'apprentissage et durant la période de rétention (expérience de la section 4.4)

4.4.1 Description des résultats

Comme prévu, les apprenants sans révision sont moins performants que ceux de toutes les stratégies de sélection des CC, à la fois pendant l'apprentissage et la rétention. Cela s'explique par le faible nombre de fois où ces apprenants de référence ont vu chaque CC (une par semaine, soit 10 interactions en tout). C'est également la seule stratégie à avoir une métrique PMR_A supérieure à la PMR_R . En effet, la moyenne de la PMR_A est calculée sur toutes les semaines : ainsi, les toutes premières semaines de la période d'apprentissage ont une métrique faible parce que l'apprenant n'a pas encore vu toutes les CC. Pendant la période de rétention,

au contraire, toutes les CC ont été vues au moins une fois par chaque apprenant.

La stratégie θ -*threshold* est la stratégie de sélection de CC unique la plus performante pendant l'apprentissage avec une PMR_A de 0,599 en moyenne ; μ -*back* atteint 0,589 de PMR_A , surpassant ainsi *Greedy* pendant l'apprentissage. En effet, celle-ci optimise localement la probabilité moyenne de réussite de toutes les CC pendant la rétention et ne vise pas à optimiser la probabilité de réussite pendant l'apprentissage, ce qui peut expliquer ces résultats. En outre, θ -*threshold* sélectionne la CC dont la probabilité de réussite immédiate est la plus proche d'une valeur fixée θ (avec ici $\theta^* = 0,4$) : cela pourrait expliquer pourquoi elle obtient d'aussi bons résultats pendant l'apprentissage.

Cependant, pendant la rétention, cette tendance change : θ -*threshold* donne des résultats équivalents à ceux de μ -*back*, alors que *Greedy* ne surpasse que légèrement les deux autres stratégies. Nous supposons que les relatives mauvaises performances de *Greedy* viennent du fait qu'elle ne prend en compte que l'impact direct du choix d'une CC, sans tenir compte de la séquence entière des révisions.

Pendant l'apprentissage, θ -*threshold* (multi.), la version multi-CC de θ -*threshold*, surpasse toutes les autres stratégies mais *Greedy* (multi.) n'atteint qu'une PMR_A de 0,592, ce qui est inférieur à θ -*threshold* et à peine supérieur à μ -*back*. Ces résultats multi-CC sont cohérents avec les résultats de leurs homologues à CC unique. D'autre part, les deux algorithmes multi-CC surpassent sensiblement toutes les autres heuristiques en termes de PMR_R avec respectivement 0,814 et 0,816 en moyenne pour θ -*threshold* (multi.) et *Greedy* (multi.). Ces résultats suggèrent que l'utilisation d'algorithmes multi-CC pour sélectionner plusieurs CC en même temps pourrait être meilleur que la sélection d'une seule CC.

4.4.2 Tests de significativité des différences entre stratégies

Pour vérifier si ces résultats sont significativement différents d'une stratégie de sélection des CC à l'autre, nous effectuons une série de tests statistiques : pour chaque paire de stratégies, nous testons à l'aide d'un *test bilatéral des signes* (CONOVER, 1998) si les deux stratégies produisent des performances statistiquement différentes chez les apprenants simulés. Le test des signes est le plus ancien des tests statistiques non-paramétriques : il s'applique quand on cherche à tester la différence entre deux séries d'échantillons $\{X_i\}_{i=1,\dots,n}$ et $\{Y_i\}_{i=1,\dots,n}$ de taille n et que chaque couple (X_i, Y_i) forme une paire⁸. Dans notre cas :

- n est le nombre total d'apprenants sur toutes les simulations ;
- comme chaque apprenant a été assigné à chacune des heuristiques d'espacement, une paire correspond à une paire de scores PMR d'un même apprenant sur deux stratégies différentes.

Nous utilisons des tests bilatéraux des signes parce que :

1. c'est un test simple à mettre en œuvre qui ne nécessite que peu d'hypothèses : indépendance mutuelle des paires, échelle ordinale de mesure au sein de chaque paire ;
2. nous ne pouvons pas nous assurer des hypothèses pour des tests similaires mais avec plus de puissance statistique, comme le test de Student apparié (hypothèse de normalité de la distribution des différences au sein de chaque paire).

Les résultats de ces tests, pour chaque période, sont reportés dans deux matrices (figures 4.5 et 4.6). Toutes les performances des stratégies de sélection sont statistiquement différentes les unes des autres au niveau 0,001.

8. Par exemple, dans le domaine médical, la mesure de la pression sanguine avant et après la prise d'un médicament par un même individu.

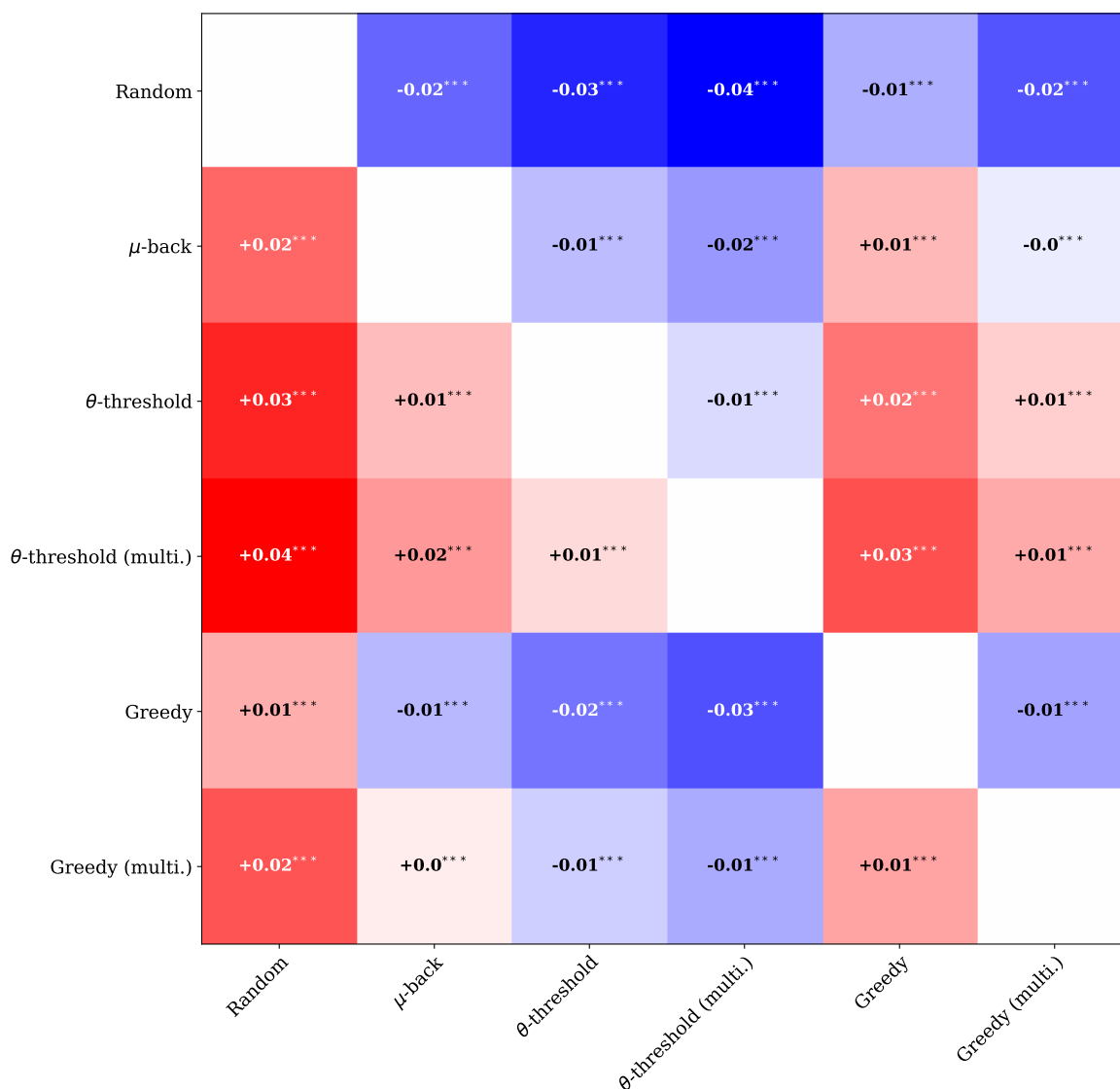


FIGURE 4.5 – Matrice montrant la significativité des différences de scores PMR_A entre chaque stratégie (sauf la stratégie sans révision). La couleur et la valeur indiquée dans chaque carré représentent la différence entre le score PMR_A moyen de la stratégie de la ligne et le score PMR_A moyen de la stratégie de la colonne : rouge quand cette différence est positive et significative et bleu quand elle est négative et significative. Le nombre d'étoiles indique le résultat du test statistique : *** quand la p -valeur est plus petite que 0,001, ** quand elle est plus petite que 0,01 et * quand elle est plus petite que 0,05.

4.4.3 Évolution temporelle des scores PMR

Pour visualiser d'une autre manière nos résultats, la figure 4.7 montre l'évolution temporelle du score PMR , moyenné sur l'ensemble des simulations et des apprenants, et ventilé par stratégie de sélection des CC. La figure 4.8 représente quant à elle la même évolution mais pour le score PMR relatif à la place, comme dans la table 4.5. Là encore, nous n'affichons que les résultats pour le meilleur paramètre trouvé pour les stratégies μ -back et θ -threshold. Pour assurer une lisibilité maximale des figures, et puisque nous avons déjà reporté des mesures de variabilité et des tests de significativité pour ces résultats auparavant, nous n'indiquons pas, à dessein, les intervalles de confiance par semaine dans ces figures.

On peut constater que, sans aucune révision, les apprenants n'augmentent que légèrement leur PMR pendant le temps d'apprentissage (cette augmentation est due à la tentative ajoutée automatiquement chaque semaine pour la CC courante dans l'historique d'apprentissage de

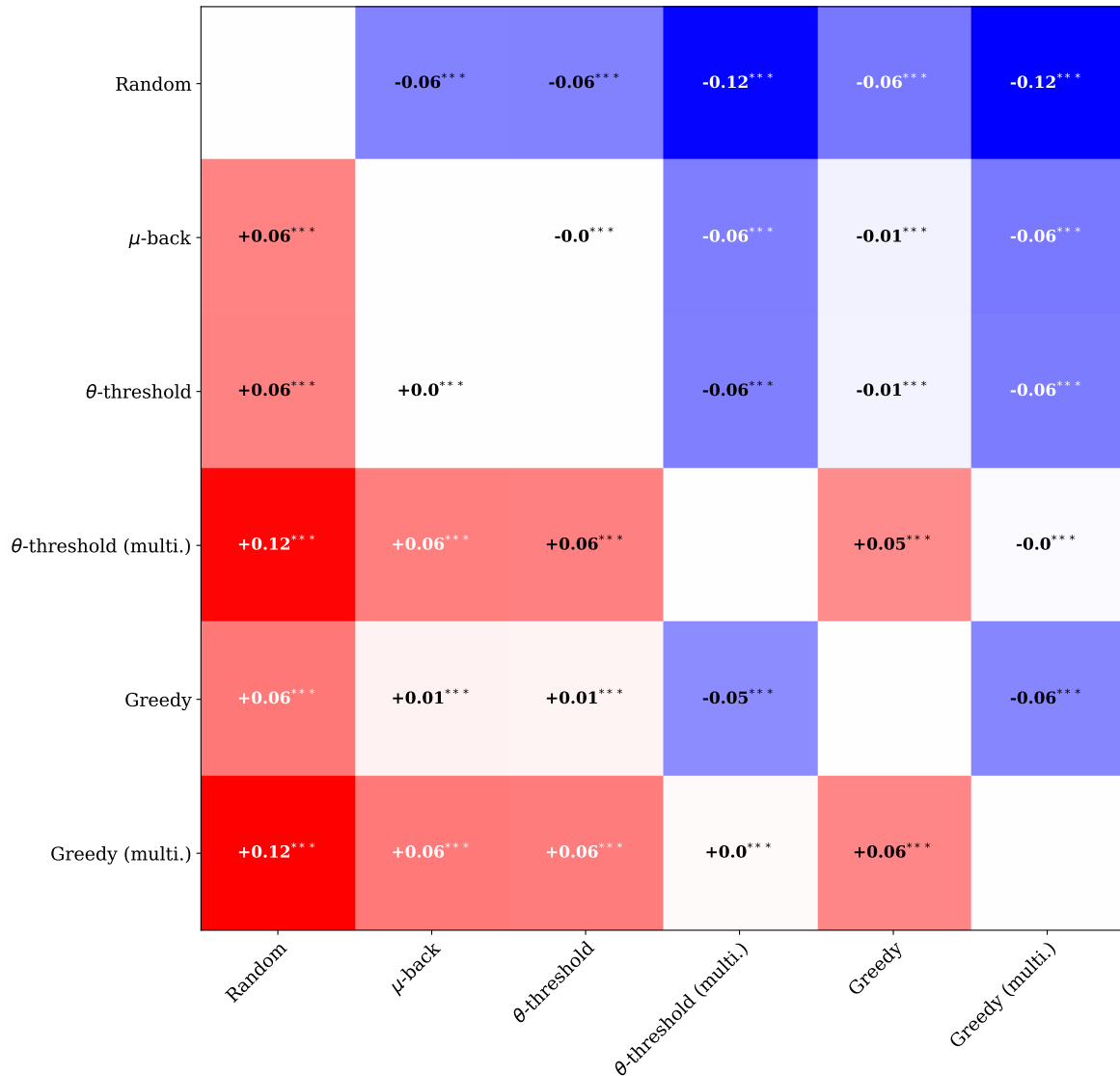


FIGURE 4.6 – Matrice montrant la significativité des différences de scores PMR_R entre chaque stratégie (sauf la stratégie sans révision). La couleur et la valeur indiquée dans chaque carré représentent la différence entre le score PMR_R moyen de la stratégie de la ligne et le score PMR_R moyen de la stratégie de la colonne : rouge quand cette différence est positive et significative et bleu quand elle est négative et significative. Le nombre d'étoiles indique le résultat du test statistique : *** quand la p -valeur est plus petite que 0,001, ** quand elle est plus petite que 0,01 et * quand elle est plus petite que 0,05.

chaque apprenant) ; pendant la rétention, et comme toute autre stratégie, leur PMR diminuent. Pour toutes les autres stratégies, toutes les courbes d'évolution de la PMR sont confondues jusqu'à la semaine 2 et se différencient progressivement après cela. Ces courbes sont confondues à la semaine 2 parce qu'à ce moment précis, les stratégies n'ont qu'un seul choix possible : faire réviser un item qui utilise la CC 1 et seulement cette CC.

Au cours de l'apprentissage, les stratégies θ -threshold surpassent toutes les autres stratégies, y compris Greedy (multi.), ce qui corrobore nos conclusions de la sous-section 4.4.1 (cf. table 4.4). Dès le début de la période de rétention, θ -threshold commence à diminuer de manière drastique et finit comme la stratégie la moins performante (si l'on excepte Random et Baseline). La stratégie μ -back a la pente de PMR la plus constante ; Greedy la surpasse pendant la majeure partie de la période de rétention, avant de terminer à un niveau similaire. En ce qui concerne les algorithmes multi-CC, ils sont tous deux les plus performants pendant toute la période de

rétention. Au début, θ -threshold (multi.) surpasse légèrement *Greedy* (multi.) mais termine à une *PMR* inférieure. Ces résultats combinés suggèrent que *Greedy* pourrait être plus adapté à la mémorisation à long terme, tandis que θ -threshold se concentre davantage sur la maîtrise à court et moyen terme.

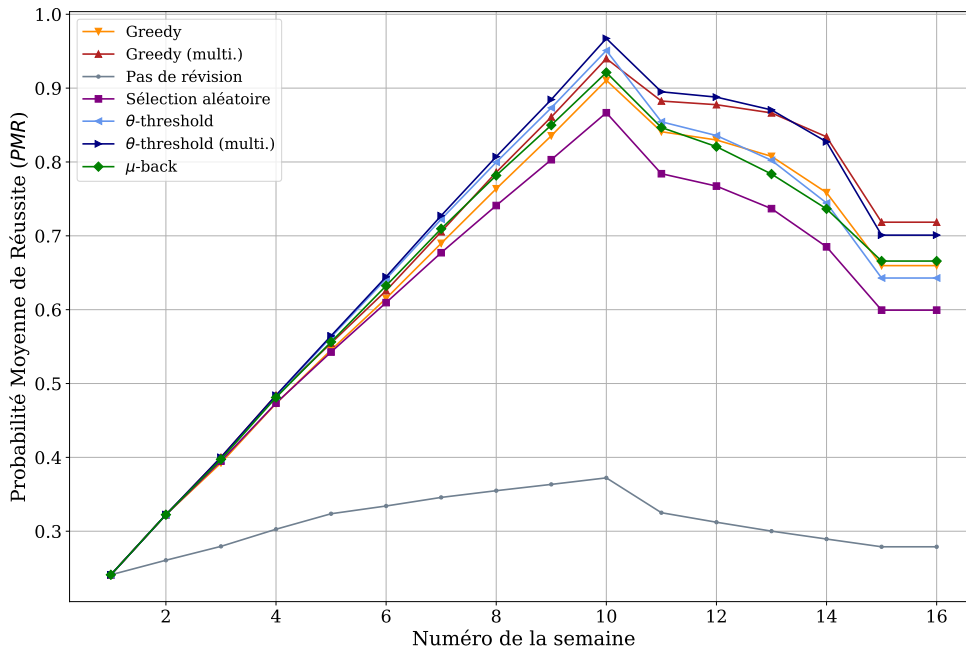


FIGURE 4.7 – Comparaison de l'évolution temporelle du score *PMR* entre différentes stratégies de sélection de CC. Pour les stratégies paramétrées, seul le meilleur paramètre est affiché. Une *PMR* supérieure implique une meilleure maîtrise des CC.

4.4.4 Synthèse des résultats

Pour résumer nos expériences conduites dans cette section, nous avons comparé différentes stratégies de sélection des CC, y compris des heuristiques qui sélectionnaient un ensemble de CC à la fois. Nous avons montré que θ -threshold et μ -back ont des performances similaires pendant la rétention, mais nous avons aussi constaté que θ -threshold améliorait les performances des apprenants pendant l'apprentissage également. Cependant, la stratégie *Greedy* n'a que légèrement surpassé ses concurrentes. Nous émettons l'hypothèse que cela provient de la procédure d'optimisation locale, incapable de prendre en compte toute la séquence des révisions futures. Enfin, nous avons également montré que les versions multi-CC de deux heuristiques proposées surpassent toutes les autres heuristiques de sélection de CC unique.

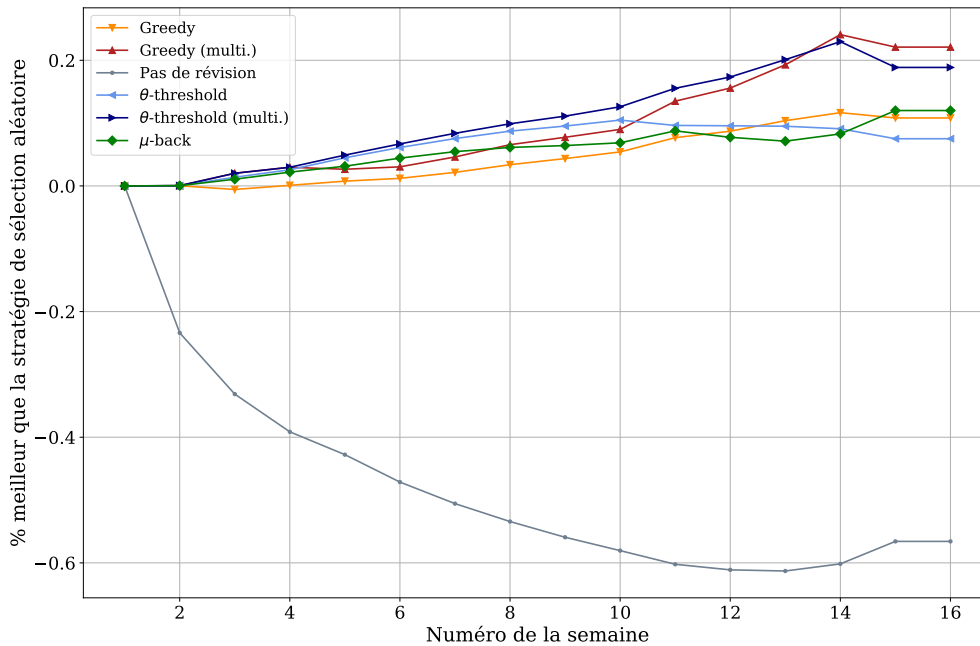


FIGURE 4.8 – Comparaison de l'évolution temporelle du score *PMR relatif* entre différentes stratégies de sélection de CC. Pour les stratégies paramétrées, seul le meilleur paramètre est affiché. Une *PMR* supérieure implique une meilleure maîtrise des CC.

4.5 Discussion et analyses complémentaires

Dans cette section, nous prolongeons l'analyse des résultats présentés à la section précédente. Plus précisément, nous souhaitons évaluer la robustesse de nos algorithmes dans des conditions différentes de la section 4.4 (sous-sections 4.5.2, 4.5.3 et 4.5.4) mais également mieux comprendre leur fonctionnement et leurs performances (sous-sections 4.5.1, 4.5.5 et 4.5.6). Nous structurons les différents axes que nous explorons par le biais de questions.

4.5.1 Les performances supérieures des heuristiques multi-CC sont-elles dues seulement à un plus grand nombre moyen de révisions de CC?

Motivation

Dans nos expériences de la section 4.4, nous avons utilisé la stratégie de sélection d'items *Random CC* pour toutes les stratégies de sélection de CC unique. Or, les stratégies de sélection de CC multiples ont, de par leur conception, tendance à sélectionner des items avec plusieurs CC. Au contraire, la stratégie de sélection d'items *Random CC* sélectionne de manière aléatoire des items parmi l'ensemble d'items acceptables : certains sont des items avec une seule CC, d'autres des items avec plusieurs CC. Nous pouvons en avoir une preuve supplémentaire en lisant la colonne « Nombre moyen de révisions (*Random CC*) » de la table 4.6, qui donne la quantité moyenne de CC pratiquées (soit par acquisition initiale, soit par révision directe ou indirecte) pendant tout le processus d'apprentissage des apprenants de l'expérience de la section 4.4. En moyenne, dans un processus d'apprentissage et de rétention, les algorithmes multi-CC sélectionnent vingt CC de plus que leurs homologues à CC unique. Cela pourrait

potentiellement constituer un avantage injuste pour les stratégies de sélection à CC unique⁹. C'est pourquoi nous analysons dans ce qui suit si les performances supérieures des deux heuristiques de sélection multi-CC proviennent juste d'un plus grand nombre moyen de révisions de CC, ou bien si elles parviennent à choisir plus intelligemment le sous-ensemble de CC que leurs concurrentes à CC unique.

Résultats

Pour avoir une comparaison plus juste entre les heuristiques de sélection de CC unique et multiples, nous effectuons une expérience additionnelle, en utilisant le même ensemble de paramètres générés et de q-matrices que dans notre expérience de la section 4.4. La différence est que nous utilisons pour toutes les stratégies à CC unique¹⁰ la stratégie de sélection d'items *Max CC* : cette stratégie choisit un item au hasard parmi l'ensemble des items acceptables qui sont également étiquetés par le plus grand nombre de CC. Nous tenons à souligner que cette stratégie de sélection d'items a été choisie uniquement à des fins de recherche. Dans une situation réelle, il ne serait pas forcément pertinent de sélectionner à chaque fois le plus grand nombre de CC. Par exemple, cela pourrait décourager les apprenants de devoir se confronter à des items trop complexes. Pour tenir compte de cela dans nos stratégies de sélection multi-CC, nous pourrions ajouter une pénalité qui récompenserait le choix d'items qui mettent en jeu moins de CC. Cela permettrait de contrebalancer la tendance de ces algorithmes à choisir plusieurs CC en même temps.

Les résultats (bruts et relatifs à la stratégie *Random* de sélection de CC pour une comparaison plus claire) sont présentés dans les tables 4.7 et 4.8. Nous indiquons le nombre moyen de CC pratiquées par un apprenant pendant toute une session d'apprentissage dans la table 4.6. Nous pouvons voir que le nombre moyen de CC sélectionnées lors de l'utilisation de la stratégie de sélection d'items *Max CC* est maintenant similaire entre les stratégies de sélection des CC : il est même légèrement plus élevé pour les stratégies de sélection de CC unique. Cela vient du fait que les stratégies de sélection multi-CC sont fortement contraintes par la q-matrice générée. Alors qu'il est possible de sélectionner pour une CC donnée un item qui utilise le nombre maximum d'étiquettes de CC supplémentaires, ce n'est pas le cas pour les stratégies de sélection de CC multiples. Par exemple, supposons qu'une stratégie de sélection multi-CC sélectionne les CC k_1 et k_3 . Cela signifie que cette combinaison de CC existe dans la q-matrice. Mais il est possible qu'aucun item n'utilise les CC k_1, k_3 ainsi qu'une *autre* CC. Dans ce cas, l'algorithme choisirait un item parmi l'ensemble des items qui utilisent *seulement* ces deux CC.

Comme prévu, les résultats ne diffèrent pas entre les stratégies *Random CC* et *Max CC*, ni pour les stratégies de sélection multi-CC, ni pour les apprenants *sans révision*. Toutefois, nous constatons également que l'utilisation d'une stratégie de sélection d'items *Max CC* augmente considérablement les métriques de performance des apprenants par rapport à la stratégie de sélection d'items *Random CC* pour les stratégies de sélection à CC unique. Cela s'explique facilement par le nombre plus élevé de CC qui sont alors révisées au cours d'une seule session d'apprentissage. Cependant, nous observons également que cette augmentation est généralement plus importante pour la PMR_R que pour la PMR_A . Nous remarquons aussi que l'écart de performance de PMR_R entre les stratégies de sélection multi-CC et les autres stratégies s'est globalement réduit mais demeure toujours positif. Par exemple, la différence de PMR_R entre *Greedy* (multi.) et μ -back a diminué de 6,3 points de pourcentage à 1,5 point de pourcentage. Plus intéressant encore, nous constatons que les performances des stratégies de sélection de CC unique combinées à la stratégie de sélection d'items *Max CC* sont toujours

9. Nous insistons toutefois sur le fait que le nombre d'items de révisions proposés aux apprenants dans un processus d'apprentissage est bien le même, quelle que soit l'heuristique de sélection de CC considérée. Les différences proviennent du nombre de CC mises en jeu dans les items choisis par les différentes heuristiques.

10. Il n'est pas possible d'utiliser la stratégie de sélection d'item *Max CC* avec les stratégies de sélection multi-CC parce qu'elles renvoient *déjà* l'ensemble exact de CC à pratiquer : ni plus, ni moins.

Stratégie de sélection	Nombre moyen de révisions	
	(<i>Random CC</i>)	(<i>Max CC</i>)
Pas de révision	10,0	10,0
<i>Random</i>	61,0	82,0
μ -back	59,6	82,0
θ -threshold	60,5	82,0
θ -threshold (multi.)	81,2	81,2
<i>Greedy</i>	60,4	82,0
<i>Greedy</i> (multi.)	80,9	80,9

TABLE 4.6 – Comparaison du nombre moyen de CC pratiquées durant une session d'apprentissage entre les stratégies de sélection d'item *Random CC* et *Max CC* pour toutes les stratégies de sélection de CC (expériences de la section 4.4). La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants.

Stratégie de sélection	PMR_A		PMR_R	
	<i>Random CC</i>	<i>Max CC</i>	<i>Random CC</i>	<i>Max CC</i>
Pas de révision	$0,318 \pm 0,001$	$0,318 \pm 0,001$	$0,297 \pm 0,001$	$0,297 \pm 0,001$
<i>Random</i>	$0,567 \pm 0,000$	$0,583 \pm 0,000$	$0,695 \pm 0,001$	$0,742 \pm 0,000$
μ -back	$0,589 \pm 0,000$	$0,599 \pm 0,000$	$0,753 \pm 0,001$	$0,801 \pm 0,000$
θ -threshold	$0,599 \pm 0,000$	$0,603 \pm 0,000$	$0,754 \pm 0,001$	$0,796 \pm 0,000$
θ -threshold (multi.)	$0,604 \pm 0,000$	$0,604 \pm 0,000$	$0,814 \pm 0,000$	$0,814 \pm 0,000$
<i>Greedy</i>	$0,579 \pm 0,000$	$0,594 \pm 0,000$	$0,759 \pm 0,001$	$0,799 \pm 0,000$
<i>Greedy</i> (multi.)	$0,592 \pm 0,000$	$0,592 \pm 0,000$	$0,816 \pm 0,000$	$0,816 \pm 0,000$

TABLE 4.7 – Comparaison des métriques de performance PMR_A et PMR_R entre les stratégies de sélection d'item *Random CC* et *Max CC*, pour toutes les stratégies de sélection de CC (expériences de la section 4.4). La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.

Stratégie de sélection	PMR_A		PMR_R	
	<i>Random CC</i>	<i>Max CC</i>	<i>Random CC</i>	<i>Max CC</i>
Pas de révision	$-0,404 \pm 0,001$	$-0,414 \pm 0,001$	$-0,593 \pm 0,001$	$-0,616 \pm 0,001$
μ -back	$0,036 \pm 0,000$	$0,023 \pm 0,000$	$0,093 \pm 0,000$	$0,088 \pm 0,000$
θ -threshold	$0,050 \pm 0,000$	$0,029 \pm 0,000$	$0,089 \pm 0,000$	$0,077 \pm 0,000$
θ -threshold (multi.)	$0,059 \pm 0,000$	$0,030 \pm 0,000$	$0,189 \pm 0,001$	$0,104 \pm 0,000$
<i>Greedy</i>	$0,017 \pm 0,000$	$0,015 \pm 0,000$	$0,100 \pm 0,000$	$0,083 \pm 0,000$
<i>Greedy</i> (multi.)	$0,039 \pm 0,000$	$0,011 \pm 0,000$	$0,194 \pm 0,000$	$0,109 \pm 0,000$

TABLE 4.8 – Comparaison des métriques de performance PMR_A et PMR_R entre les stratégies de sélection d'item *Random CC* et *Max CC*, pour toutes les stratégies de sélection de CC, relatives à la stratégie *Random* (expériences de la section 4.4). La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.

moins bonnes que celles de leurs homologues multi-CC et que le nombre moyen de CC révisées au cours d'une session est légèrement inférieur pour les stratégies de sélection multi-CC que pour les autres stratégies. Les mêmes tests statistiques qu'à la section 4.4 indiquent que toutes les différences par paire sont significatives au niveau 0,001, sauf la différence entre *μ -back* et *Greedy* pour la PMR_A qui est seulement significative au niveau 0,05.

Ces résultats suggèrent que les stratégies de sélection multi-CC sont plus performantes que les stratégies de sélection à CC unique, non seulement parce qu'elles sélectionnent plus de CC, mais aussi parce que chaque CC du sous-ensemble est sélectionnée avec plus de soin.

4.5.2 Les performances des algorithmes d'espacement adaptatif sont-elles sensibles à des comportements atypiques d'oubli ?

Motivation

Dans nos expériences de la section 4.4, nous avons échantillonné les paramètres $\theta_{k,w}$ de la zone 1 de la figure 3.2. Cela signifie que la résolution correcte d'un item mettant en jeu une CC donnée par un apprenant améliorerait plus la probabilité de bonne réponse future sur cette CC que d'échouer à cet item. Toutefois, d'autres zones du diagramme pourraient également constituer des zones d'échantillonnage plausibles et réalistes pour nos expériences. C'est le cas de la zone 6 : l'échantillonnage de nos paramètres $\theta_{k,w}$ dans cette zone signifierait que le fait de réussir à résoudre un item qui utilise une CC donnée améliore toujours la probabilité de bonne réponse future sur cette CC, mais moins que le fait d'échouer à cet item. Des exemples de courbes d'oubli générées à partir de cette zone sont présentés dans la figure 4.9.

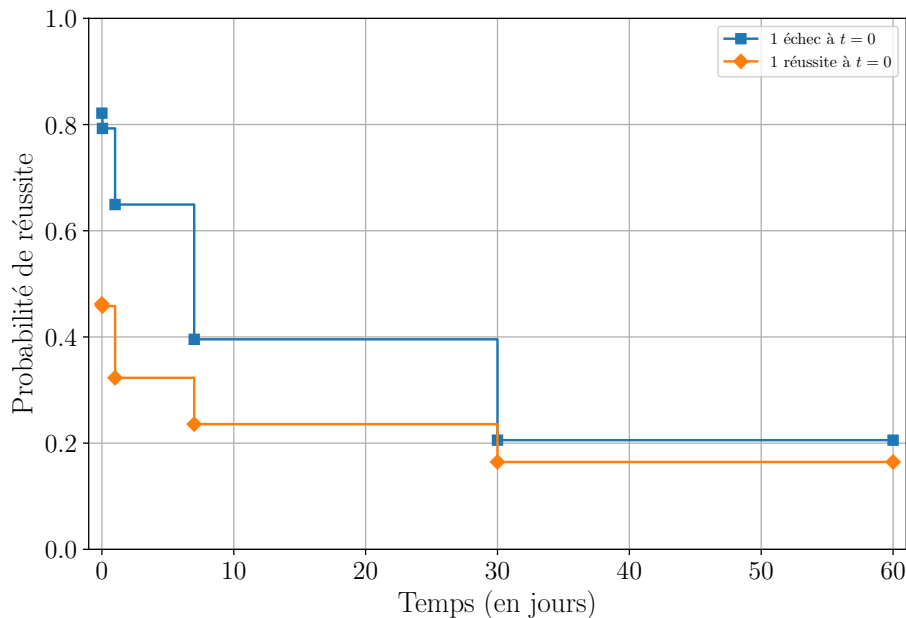


FIGURE 4.9 – Exemples de courbes d'oubli générées par notre modèle d'apprenant DAS3H, avec des $\theta_{k,w}$ échantillonnés depuis la zone 6 de la figure 3.2. Deux situations sont présentées : en bleu, un apprenant de niveau initial moyen pratique la CC à $t = 0$ mais échoue ; en orange, l'apprenant réussit. Ce sont les équivalents pour l'oubli des courbes d'apprentissage idéalisées de GOUTTE et al. (2018).

Cette situation pourrait sembler contre-intuitive. Après tout, une bonne réponse est a priori un signal plus fort de la maîtrise d'une CC qu'une mauvaise réponse. Une interprétation possible d'une telle situation provient de résultats tirés de la littérature en sciences cognitives (ROWLAND, 2014). Dans cette méta-analyse, ROWLAND montre en particulier que des performances initiales élevées chez les apprenants peuvent dans certains cas être associées à un effet de test moins fort que pour des performances initiales faibles. C'est notamment le cas lorsqu'un retour correctif est fourni à l'apprenant après chaque réponse. En l'absence de ce retour correctif, les réponses correctes ont tendance à augmenter la probabilité de bonne réponse future plus que les mauvaises réponses. Nous avons aussi montré (CHOFFIN, LATIMIER et al., 2019) que des apprenants répondant à des QCM dans une expérience en psychologie cognitive avaient tendance à passer plus de temps sur le retour correctif quand leur réponse était fautive. Dans ce cas, on peut imaginer qu'une mauvaise réponse incite l'apprenant à comprendre son erreur et génère donc une plus grande augmentation de la probabilité de bonne réponse future qu'une bonne réponse. Cette configuration n'a, à notre connaissance,

Stratégie de sélection	Meilleur paramètre	PMR_A	PMR_R	Nombre moyen de révisions
Pas de révision	\emptyset	$0,316 \pm 0,001$	$0,304 \pm 0,001$	10,0
Sélection aléatoire	\emptyset	$0,465 \pm 0,001$	$0,418 \pm 0,001$	60,8
μ -back	2	$0,453 \pm 0,001$	$0,415 \pm 0,001$	60,7
θ -threshold	0,5	$0,468 \pm 0,001$	$0,419 \pm 0,001$	60,7
θ -threshold (multi.)	0,2	$0,505 \pm 0,001$	$0,434 \pm 0,001$	81,6
Greedy	\emptyset	$0,470 \pm 0,001$	$0,441 \pm 0,001$	60,7
Greedy (multi.)	\emptyset	$0,482 \pm 0,001$	$0,458 \pm 0,001$	81,3

TABLE 4.9 – Comparaison des métriques de performance PMR_A et PMR_R entre toutes les stratégies de sélection des CC. Les paramètres θ sont échantillonnés dans la zone 6 du diagramme de la figure 3.2. La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.

jamais été utilisée auparavant dans la littérature sur l'espacement adaptatif de l'apprentissage (HUNZIKER et al., 2019 ; TABIBIAN et al., 2019 ; REDDY, LEVINE et al., 2017). Avec cette expérience supplémentaire, nous voulons tester la robustesse de nos stratégies de sélection de CC dans des situations d'apprentissage et d'oubli alternatives mais réalistes : celle que nous utilisons en est un exemple.

Résultats

Par conséquent, nous effectuons les mêmes expériences que dans la section 4.4 mais nous échantillonnons les paramètres $\theta_{k,w}$ à partir de la zone 6 de la figure 3.2. Plus précisément, nous générons d'abord les paramètres $\theta_{k,2w+2}$ selon $\mathcal{U}_{[0,2]}$ et ensuite

$$\forall k, w \in \llbracket 1, K \rrbracket \times \llbracket 1, W \rrbracket, \theta_{k,2w+1} \sim \mathcal{U}_{[-\theta_{k,2w+2}, 0]}. \quad (4.15)$$

Ceci force les coefficients des réussites à être positifs mais plus petits que les coefficients des tentatives. À part cette simple différence, l'ensemble du protocole expérimental reste identique.

Les résultats de cette expérience complémentaire sont donnés dans les tables 4.9 et 4.10. Nous effectuons également des tests de significativité des différences de performance par paire entre toutes les stratégies de sélection de CC comme dans la section 4.4 et toutes ces différences sont significatives au niveau 0,001. On peut constater que, dans l'ensemble, les performances de toutes les heuristiques de sélection de CC diminuent considérablement, par rapport aux résultats expérimentaux principaux. En outre, tous les scores PMR_A sont désormais supérieurs aux scores PMR_R , ce qui signifie que l'impact de l'oubli pèse ici plus lourdement sur la maîtrise à long terme de l'apprenant.

Ici, la stratégie μ -back est moins performante que la stratégie *Random* et son meilleur paramètre est maintenant $\mu = 2$. Nous pensons que ce résultat s'explique par le fait que par construction, μ -back a tendance à choisir des CC récemment introduites, ce qui augmente la probabilité de bonne réponse de l'apprenant lors de la révision. Étant donné que, dans ce contexte alternatif, les échecs ont un impact plus positif sur la probabilité de bonne réponse que les résolutions correctes, cela pourrait expliquer pourquoi μ -back a de moins bons résultats que *Random*. La stratégie θ -threshold souffre également de cette situation : même si elle surpasse la stratégie μ -back, elle parvient à peine à faire mieux que les performances de la stratégie *Random*. Il est intéressant de noter que *Greedy* parvient à s'adapter à cette situation différente et surpasse sensiblement toutes les autres stratégies de sélection de CC unique pendant la période de rétention tout en réalisant des performances équivalentes aux stratégies *Random* et θ -threshold pendant l'apprentissage. De même, *Greedy* (multi.) surpasse toutes les autres stratégies de sélection des CC au cours des deux périodes, à l'exception de θ -threshold

Stratégie de sélection	Meilleur paramètre	PMR_A relative	PMR_R relative	Nombre moyen de révisions
Pas de révision	\emptyset	$-0,318 \pm 0,001$	$-0,303 \pm 0,001$	10,0
μ -back	2	$-0,021 \pm 0,000$	$-0,007 \pm 0,000$	60,7
θ -threshold	0,5	$0,007 \pm 0,000$	$0,005 \pm 0,000$	60,7
θ -threshold (multi.)	0,2	$0,083 \pm 0,000$	$0,040 \pm 0,000$	81,6
Greedy	\emptyset	$0,012 \pm 0,000$	$0,073 \pm 0,000$	60,7
Greedy (multi.)	\emptyset	$0,035 \pm 0,000$	$0,112 \pm 0,000$	81,3

TABLE 4.10 – Comparaison des métriques de performance PMR_A et PMR_R , relatives à la stratégie *Random*, entre toutes les stratégies de sélection des CC. Les paramètres θ sont échantillonnés dans la zone 6 du diagramme de la figure 3.2. La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.

(multi.) pendant l'apprentissage. Nous supposons que cela provient de la flexibilité accrue des algorithmes *Greedy* que nous proposons et qui leur permet de s'adapter à diverses situations d'apprentissage.

Pour conclure, nos résultats suggèrent que parmi les heuristiques de sélection de CC que nous proposons, seules les heuristiques *Greedy* et *Greedy* (multi.) sont à même de s'adapter à des comportements atypiques d'apprentissage et d'oubli.

4.5.3 Est-ce que l'utilisation de paramètres *estimés* au lieu des paramètres *générés* du modèle détériore les performances des stratégies θ -*threshold* et *Greedy*?

Motivation

Contrairement à leurs homologues, les stratégies θ -*threshold* et *Greedy* (ainsi que leurs homologues multi-CC) nécessitent de pouvoir inférer statistiquement la probabilité de bonne réponse actuelle et la probabilité de bonne réponse pendant n'importe quelle période future pour tout apprenant s : ainsi, elles font toutes deux partie des stratégies *avec modèle*.

Dans nos expériences de la section 4.4, θ -*threshold* et *Greedy* utilisent les *vrais paramètres*¹¹ du modèle de l'apprenant pour sélectionner les items à choisir à n'importe quel moment. C'est également le cas pour M. M. KHAJAH et al. (2014), REDDY, LEVINE et al. (2017) et YANG et al. (2020), qui utilisent les vrais paramètres du modèle de l'apprenant pour la stratégie θ -*threshold*.

Cette utilisation des vrais paramètres du modèle de l'apprenant offre une sorte de borne supérieure des performances des stratégies *avec modèle*, mais n'est pas très réaliste. Dans une situation réelle, le modèle générateur des réponses des apprenants nous est en effet inconnu, et par conséquent ses paramètres le sont aussi : il est alors nécessaire de les estimer à partir d'un ensemble de réponses d'apprenants sur les items et les CC.

Par conséquent, nous souhaitons savoir en quoi la performance de ces stratégies dépend de la précision des estimations des paramètres du modèle de l'apprenant sur lequel elles reposent. L'utilisation de simulations nous offre un cadre expérimental pour tester l'impact de l'utilisation de paramètres estimés sur les stratégies d'espacement adaptatif avec modèle. Dans cette partie, nous menons donc des expériences supplémentaires pour voir si l'utilisation pour la planification de révisions de paramètres estimés au lieu des vrais paramètres du modèle de l'apprenant affecte les performances de θ -*threshold*, *Greedy*, et leurs homologues multi-CC.

Résultats

Pour ce faire, au début de chaque simulation, nous avons ajouté une phase de simulation supplémentaire au cours de laquelle une cohorte de 200 apprenants synthétiques répondent pendant 10 semaines consécutives à des items tirés aléatoirement de l'ensemble d'items généré pour l'expérience courante ; chaque semaine, chaque apprenant répond à 3 items, comme dans la section 4.4. Au total, cela représente 6000 échantillons, ce qui est relativement peu au regard du nombre de paramètres à estimer par DAS3H (cf. table 3.3). Les apprenants sont simulés selon les mêmes hypothèses de distribution que pour nos expériences de la section 4.4. Les résultats sont ensuite collectés et les paramètres $(\delta_j, \beta_k, \theta_{k,w})$ sont estimés en utilisant une régression logistique avec régularisation L_2 ($C = 1$) du module `scikit-learn` (PEDREGOSA et al., 2011). Ensuite, ces paramètres estimés sont utilisés à la place des vrais paramètres du modèle dans les stratégies θ -*threshold* et *Greedy* ; pour calculer les trajectoires réelles des apprenants, ensuite, nous utilisons bien sûr toujours les vrais paramètres du modèle.

Malheureusement, nous ne pouvons pas estimer les paramètres du modèle DAS3H pendant l'expérience elle-même, avec les données des apprenants simulés, car les différentes CC sont introduites à une semaine d'intervalle. Ainsi, et contrairement à DASH (LINDSEY et al., 2014), les données du début de chaque expérience ne peuvent pas être utilisées pour prédire les performances futures sur des CC encore non introduites pour les apprenants.

Les résultats de ces analyses complémentaires sont donnés dans la table 4.11. On peut constater que l'utilisation de paramètres estimés au lieu des vrais paramètres du modèle de l'apprenant n'a qu'un faible impact sur les performances des stratégies fondées sur un modèle. Cet impact est modulé à la fois par la stratégie de sélection de CC et par la période

11. Par *vrais paramètres* du modèle, nous entendons les paramètres exacts, ceux qui servent à générer la probabilité de rappel de chaque apprenant dans les simulations.

Stratégie de sélection	PMR_A		PMR_R	
	Param. générés	Param. estimés	Param. générés	Param. estimés
θ -threshold	0,599 ± 0,000	0,599 ± 0,000	0,754 ± 0,001	0,753 ± 0,001
θ -threshold (multi.)	0,604 ± 0,000	0,604 ± 0,000	0,814 ± 0,000	0,815 ± 0,000
Greedy	0,579 ± 0,000	0,586 ± 0,000	0,759 ± 0,001	0,757 ± 0,001
Greedy (multi.)	0,592 ± 0,000	0,597 ± 0,000	0,816 ± 0,000	0,812 ± 0,000

TABLE 4.11 – Comparaison des métriques de performance PMR_A et PMR_R entre les stratégies θ -threshold et Greedy, avec les vrais paramètres du modèle ou les estimations de ces paramètres (expériences de la section 4.4). La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.

considérée. En effet, l'écart de performance est plus important en valeur absolue pour les algorithmes Greedy que pour les stratégies θ -threshold, confirmant notre hypothèse selon laquelle des modèles précis d'apprentissage et d'oubli de l'apprenant sont essentiels pour cette heuristique. Plus intéressant encore, les vrais paramètres du modèle permettent d'obtenir des performances des apprenants inférieures à celles des paramètres estimés pendant la période d'apprentissage pour les algorithmes Greedy ; cependant, ce n'est plus le cas pour la période de rétention, où l'on observe l'effet inverse.

Nous effectuons également une série de tests statistiques de significativité de différence par paires avec des tests bilatéraux des signes, comme dans la section 4.4, avec les versions estimées de l'heuristique θ -threshold et de l'heuristique Greedy. Toutes les différences par paires sont significatives au niveau 0,001, à l'exception de la différence entre θ -threshold et θ -threshold (est.) pour la PMR_R qui n'est pas significative au niveau 0,05.

4.5.4 Est-ce que supposer une dynamique *unique* d'apprentissage et d'oubli entre les CC réduit les performances des algorithmes d'espacement adaptatif ?

Motivation

Dans le chapitre 3 de cette thèse, nous avons comparé notre modèle DAS3H à une version alternative, appelée DAS3H_{1p}, dans laquelle toutes les CC partageaient les mêmes paramètres θ_w pour une même fenêtre temporelle; en d'autres termes,

$$\forall k \in \llbracket 1, K \rrbracket, \forall w \in \llbracket 1, 2W \rrbracket \quad \theta_{k,w} = \theta_w. \quad (4.16)$$

Cela revient à supposer que l'apprentissage et l'oubli se font au même rythme pour chaque CC; c'est d'ailleurs l'hypothèse faite par LINDSEY et al. (2014) dans leur modèle DASH. Nous avons vu que DAS3H_{1p} était certes moins prédictif que DAS3H sur la plupart des jeux de données mais qu'il parvenait tout de même à obtenir de bonnes performances prédictives, si on le compare aux autres modèles considérés au chapitre 3.

Dans (M. M. KHAJAH et al., 2014), les auteurs supposent que pour une simulation donnée, chaque CC (introduite dans la séquence des K blocs) a la même courbe d'apprentissage et d'oubli que les autres. Pour nous, cela pourrait sous-estimer les gains potentiels liés à l'adaptativité du planning d'espacement des révisions puisque toutes les CC se comportent alors de la même manière : dans ce cas, il ne serait par exemple pas possible de se concentrer sur les CC qui nécessitent une attention plus soutenue que les autres CC. Nous faisons donc l'hypothèse que supposer une courbe d'apprentissage et d'oubli unique et commune à toutes les CC du programme réduirait les performances des algorithmes d'espacement adaptatif développés à la section 4.3.

Résultats

Pour tester cette hypothèse et comparer nos résultats à ceux de M. M. KHAJAH et al. (2014), nous effectuons ici le même type d'expériences que dans la section 4.4. La différence est qu'au lieu d'avoir K courbes d'apprentissage et d'oubli distinctes (une pour chaque CC k), nous supposons que toutes les CC ont la même courbe d'apprentissage et d'oubli. À part cela, le reste du protocole expérimental est identique. Cette configuration est plus proche du protocole expérimental de M. M. KHAJAH et al. (2014).

Nos résultats se trouvent dans les tables 4.12 et 4.13. « est. » signifie qu'il s'agit de la version de l'heuristique qui utilise les paramètres estimés du modèle (cf. sous-section 4.5.3). On peut tout d'abord constater que ces résultats sont similaires à nos résultats expérimentaux principaux en termes de tendances : le classement des stratégies est conservé, sauf entre θ -*threshold* (multi.) et *Greedy* (multi.). Cependant, on note tout de même une baisse globale des scores PMR entre DAS3H et DAS3H_{1p}, mais cette baisse ne semble pas plus marquée pour les stratégies adaptatives (θ -*threshold* et *Greedy*) que pour les stratégies non-adaptatives (*Random* et μ -*back*). Toutes les différences par paire entre les stratégies ont été testées de la même manière que dans la section 4.4 et sont toutes significatives au niveau 0,001, sauf entre θ -*threshold* et θ -*threshold* (est.), pour qui la différence de PMR_R n'est pas significative au niveau 0,05. On constate également que les paramètres optimaux ne changent pas beaucoup entre les deux configurations expérimentales : en effet, la seule différence concerne les stratégies θ -*threshold* (multi.), qui, comme nous l'avons vu dans la figure 4.15, ont des performances similaires pour $\theta \in [0; 0,5]$. Ici, les θ optimaux restent dans cet intervalle. Ceci suggère que notre hypothèse selon laquelle considérer les mêmes comportements d'apprentissage et d'oubli pour toutes les CC ne permettrait pas de mettre en évidence la véritable efficacité de l'espacement adaptatif n'est pas pertinente.

Enfin, cette expérience supplémentaire nous permet de tester l'impact du décalage de modèle (*model mismatch* en anglais) entre le modèle générateur de trajectoires d'apprenants et les modèles utilisés par les planificateurs d'espacement adaptatif θ -*threshold* et *Greedy*.

Stratégie de sélection	Meilleur paramètre	PMR_A	PMR_R	Nombre moyen de révisions
Pas de révision	\emptyset	$0,309 \pm 0,001$	$0,284 \pm 0,001$	10,0
<i>Random</i>	\emptyset	$0,562 \pm 0,000$	$0,679 \pm 0,001$	60,4
<i>μ-back</i>	1	$0,585 \pm 0,000$	$0,733 \pm 0,001$	59,2
<i>θ-threshold</i>	0,4	$0,593 \pm 0,000$	$0,739 \pm 0,001$	60,2
<i>θ-threshold (est.)</i>	0,5	$0,592 \pm 0,000$	$0,737 \pm 0,001$	60,2
<i>θ-threshold (multi.)</i>	0,5	$0,601 \pm 0,000$	$0,795 \pm 0,001$	81,1
<i>θ-threshold (multi., est.)</i>	0,5	$0,602 \pm 0,000$	$0,799 \pm 0,001$	81,0
<i>Greedy</i>	\emptyset	$0,572 \pm 0,000$	$0,739 \pm 0,001$	60,1
<i>Greedy (est.)</i>	\emptyset	$0,580 \pm 0,000$	$0,738 \pm 0,001$	60,1
<i>Greedy (multi.)</i>	\emptyset	$0,584 \pm 0,000$	$0,793 \pm 0,001$	81,1
<i>Greedy (multi., est.)</i>	\emptyset	$0,591 \pm 0,000$	$0,793 \pm 0,001$	81,2

TABLE 4.12 – Comparaison des métriques de performance PMR_A et PMR_R entre toutes les stratégies de sélection des CC. Les courbes d'apprentissage et d'oubli sont identiques entre les différentes CC d'une même exécution (modèle générateur DAS3H_{1p}). La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.

Stratégie de sélection	Meilleur paramètre	PMR_A relative	PMR_R relative	Nombre moyen de révisions
Pas de révision	\emptyset	$-0,414 \pm 0,001$	$-0,599 \pm 0,001$	10,0
<i>μ-back</i>	1	$0,038 \pm 0,000$	$0,082 \pm 0,000$	59,2
<i>θ-threshold</i>	0,4	$0,050 \pm 0,000$	$0,090 \pm 0,000$	60,2
<i>θ-threshold (est.)</i>	0,5	$0,049 \pm 0,000$	$0,085 \pm 0,000$	60,2
<i>θ-threshold (multi.)</i>	0,5	$0,066 \pm 0,000$	$0,198 \pm 0,001$	81,1
<i>θ-threshold (multi., est.)</i>	0,5	$0,067 \pm 0,000$	$0,202 \pm 0,001$	81,0
<i>Greedy</i>	\emptyset	$0,015 \pm 0,000$	$0,093 \pm 0,000$	60,1
<i>Greedy (est.)</i>	\emptyset	$0,030 \pm 0,000$	$0,090 \pm 0,000$	60,1
<i>Greedy (multi.)</i>	\emptyset	$0,036 \pm 0,000$	$0,193 \pm 0,001$	81,1
<i>Greedy (multi., est.)</i>	\emptyset	$0,050 \pm 0,000$	$0,194 \pm 0,001$	81,2

TABLE 4.13 – Comparaison des métriques de performance PMR_A et PMR_R , relatives à la stratégie *Random*, entre toutes les stratégies de sélection des CC. Les courbes d'apprentissage et d'oubli sont identiques entre les différentes CC d'une même exécution de la simulation (modèle générateur DAS3H_{1p}). La moyenne des résultats est calculée sur l'ensemble des expériences, des CC et des apprenants. Les erreurs types de la moyenne sont indiquées. Une PMR supérieure implique une meilleure maîtrise des CC.

En effet, dans cette expérience, notre modèle générateur est le modèle DAS3H_{1p}. En outre, nous utilisons au début de chaque simulation le même procédé d'estimation de paramètres du modèle que nous avons décrit en sous-section 4.5.3. Pour récupérer les estimations des paramètres, le modèle d'apprenant que nous entraînons au début de chaque simulation est le modèle DAS3H d'origine. DAS3H (modèle utilisé pour la recommandation) estime différentes courbes d'apprentissage et d'oubli pour différentes CC au lieu d'une seule et unique courbe d'apprentissage et d'oubli, comme le fait DAS3H_{1p} (modèle utilisé pour la simulation). Cette expérience supplémentaire est donc l'occasion pour nous d'étudier une forme de décalage de modèle et son impact sur les performances de *θ -threshold* et *Greedy*. On peut voir dans les résultats des versions estimées de nos heuristiques que ce décalage de modèle entre le générateur et le planificateur n'a pas beaucoup d'impact sur les performances de ces stratégies avec modèle.

Dans cette partie, nous avons cherché à savoir si supposer que toutes les CC d'une simulation s'apprenaient et s'oubliaient au même rythme réduisait les performances des algorithmes d'espacement adaptatif de l'apprentissage. Ce questionnement est important car, si les per-

performances des algorithmes étaient modifiées, cela pourrait par exemple guider le choix de l'heuristique dans une situation réelle où les différences entre CC sont faibles. Contre toute attente, nos résultats suggèrent que les performances des heuristiques ne sont que marginalement affectées dans cette situation. Surtout, les différences que l'on observe ne semblent pas varier entre stratégies adaptatives et non-adaptatives. Enfin, cette expérience a aussi l'avantage de présenter une forme faible de décalage de modèle entre modèle générateur de trajectoires et modèle utilisé pour la planification. Dans une situation réelle, nous n'avons pas accès au modèle exact de l'apprenant, il est donc crucial de pouvoir évaluer en amont la sensibilité des heuristiques d'espacement à ce décalage de modèle. Nos résultats suggèrent que nos algorithmes sont robustes au problème du décalage de modèle.

4.5.5 Quelles CC sont favorisées par chaque stratégie ? Quel impact cela a-t-il sur la PMR_R des apprenants ?

Motivation

Nous avons observé dans la section 4.4 que les stratégies que nous avons développées présentent des différences de performances. Une des explications possibles de ces résultats est que les stratégies ne se focalisent pas sur les mêmes CC et que ces différences ont un impact sur leurs performances. Nous cherchons donc dans cette partie à mieux comprendre les différences de performance de nos heuristiques en décomposant ces performances par ordre d'introduction des CC dans le programme d'apprentissage.

Résultats

Dans la figure 4.10, nous représentons pour chaque stratégie de sélection de CC la distribution des sélections directes de CC à chaque semaine du processus d'apprentissage initial. Les différentes semaines sont représentées par des barres successives dans les graphiques et dans chaque barre, chaque couleur représente la proportion globale de révisions directes qui ont été consacrées à cette CC. Ce sont donc les CC qui ont été directement sélectionnées par chaque stratégie : en d'autres termes, nous ne comptons ici ni les CC qui ont été indirectement sélectionnées par un algorithme, ni l'acquisition initiale d'une nouvelle CC à chaque semaine. Pour les stratégies θ -*threshold* et μ -*back*, nous choisissons le meilleur paramètre, celui qui maximise le score PMR_R . Les données utilisées dans ces figures concernent la stratégie de sélection d'items *Random CC*, dans l'expérience principale. On peut constater qu'il y a une certaine variabilité dans les distributions de révisions : il est notamment étonnant de constater qu'une stratégie comme μ -*back* a des performances similaires à celles de θ -*threshold* mais des sélections totalement différentes. Ceci nous incite à poursuivre l'analyse en décomposant les performances des différentes heuristiques par ordre d'introduction des CC dans le programme.

Dans la figure 4.11, nous comparons ainsi pour chaque stratégie de sélection de CC le score PMR_R ventilé par l'indice d'introduction de la CC. Là encore, nous n'affichons que les résultats pour le meilleur paramètre trouvé pour μ -*back* et θ -*threshold* (PMR_R). On constate que la plupart des stratégies de révision suivent la même tendance générale : à mesure que le numéro de la CC augmente, le score PMR_R diminue légèrement. Cela peut sembler contre-intuitif en raison de l'effet d'oubli, mais cela vient du fait qu'en raison de notre cadre de révision de multiples CC, les premières CC ont beaucoup plus de possibilités d'être révisées indirectement que les CC ultérieures. Dans la figure 4.12, nous montrons qu'en raison de l'ordre d'apparition des CC dans nos simulations, les CC du début ont tendance à être révisées (directement ou indirectement) plus souvent que les CC suivantes. Ceci provient du plus grand nombre d'items qui mettent en jeu les CC du début que les CC de la fin.

Pour revenir à la figure 4.11, cette tendance est différente pour μ -*back* qui parvient à maintenir une PMR_R stable sur l'ensemble des CC ; cette analyse suggère que la raison pour laquelle μ -*back* obtient de si bons résultats dans la section 4.4 est due aux CC 8 et 9, dont les PMR sont nettement moins bonnes pour toutes les autres stratégies de sélection de CC unique. En outre, la stratégie *Random* présente des résultats semblables aux stratégies de sélection multi-CC pour les toutes premières CC, mais souffre d'une diminution plus importante pour toutes les CC. Nous constatons également que la stratégie θ -*threshold* (multi.) surpasse la stratégie *Greedy* (multi.) pour les premières CC mais qu'elle est ensuite surpassée par la stratégie *Greedy* (multi.) pour les dernières. Toutes les stratégies de révision aboutissent à la même PMR_R pour la dernière CC parce que celle-ci n'a pas pu être révisée en raison de notre protocole expérimental, dans lequel les apprenants ne pouvaient réviser que les CC qu'ils avaient vues lors des semaines précédentes.

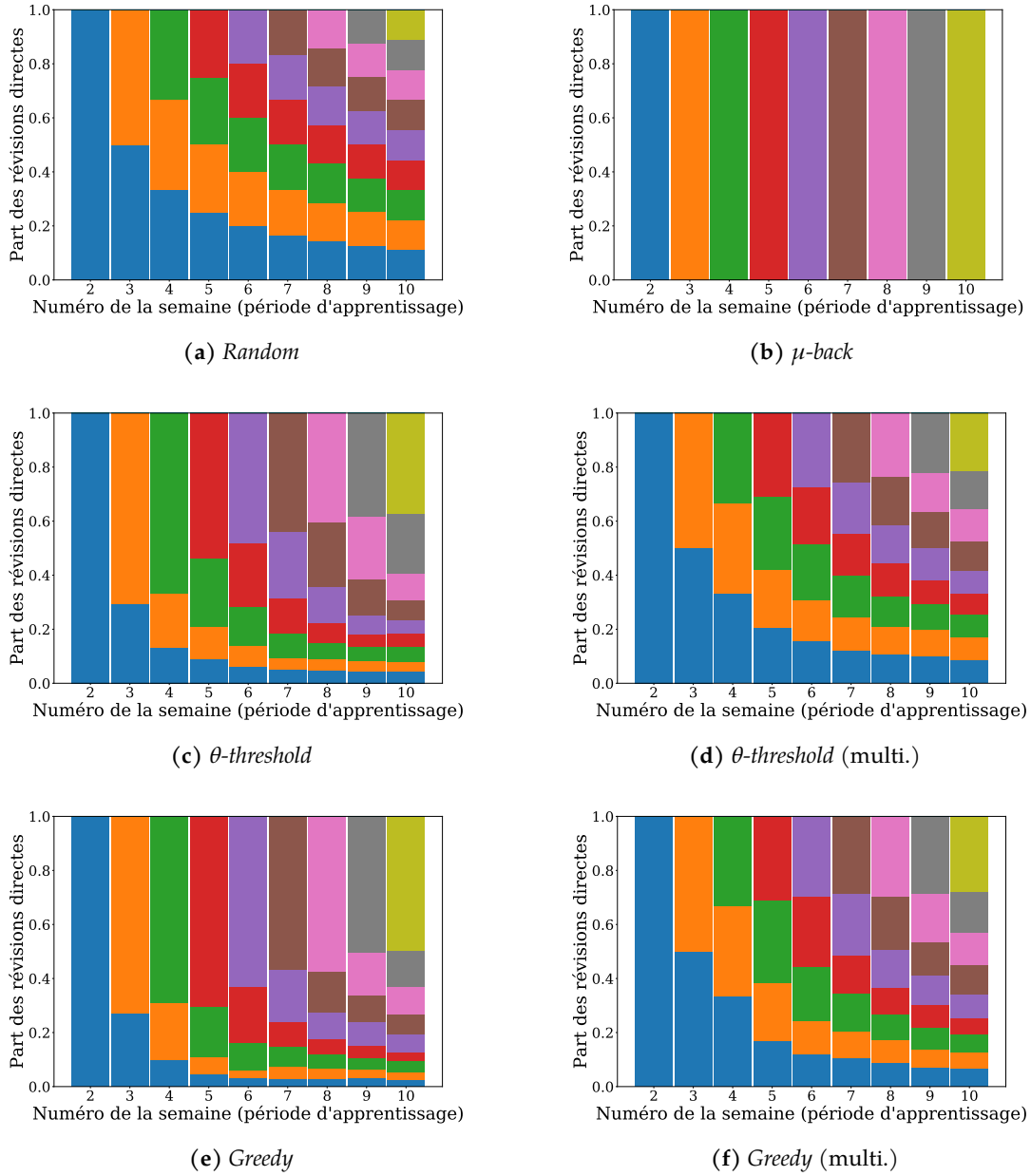


FIGURE 4.10 – Distributions des révisions *directes* de CC par semaine, par stratégie (expériences de la section 4.4).

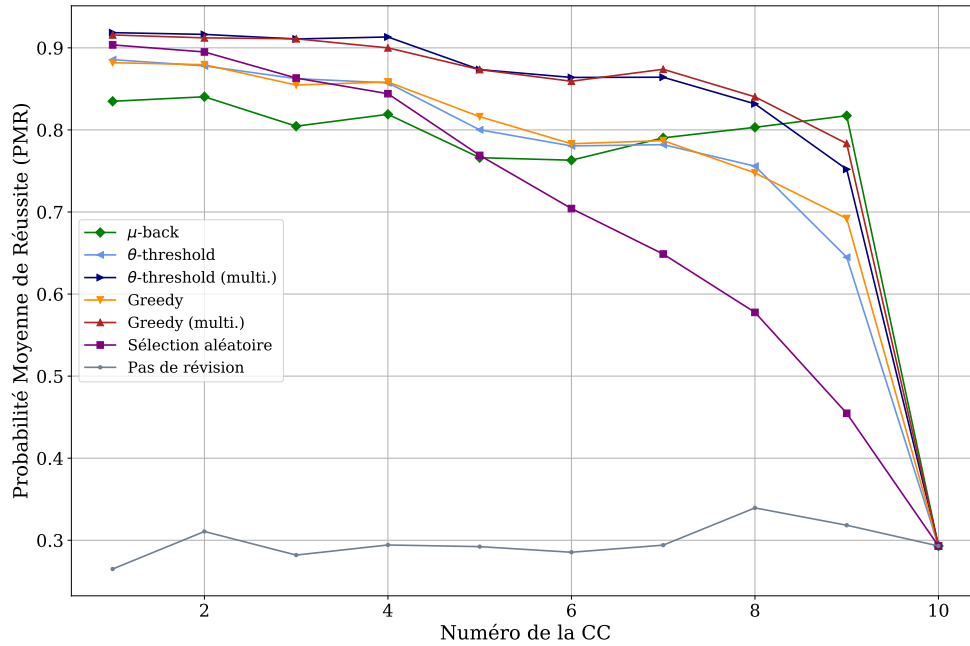


FIGURE 4.11 – Comparaison du score PMR_R par stratégie de sélection de CC (expériences de la section 4.4). Pour les stratégies paramétrées, seul le meilleur paramètre est affiché. Une PMR supérieure implique une meilleure maîtrise des CC.

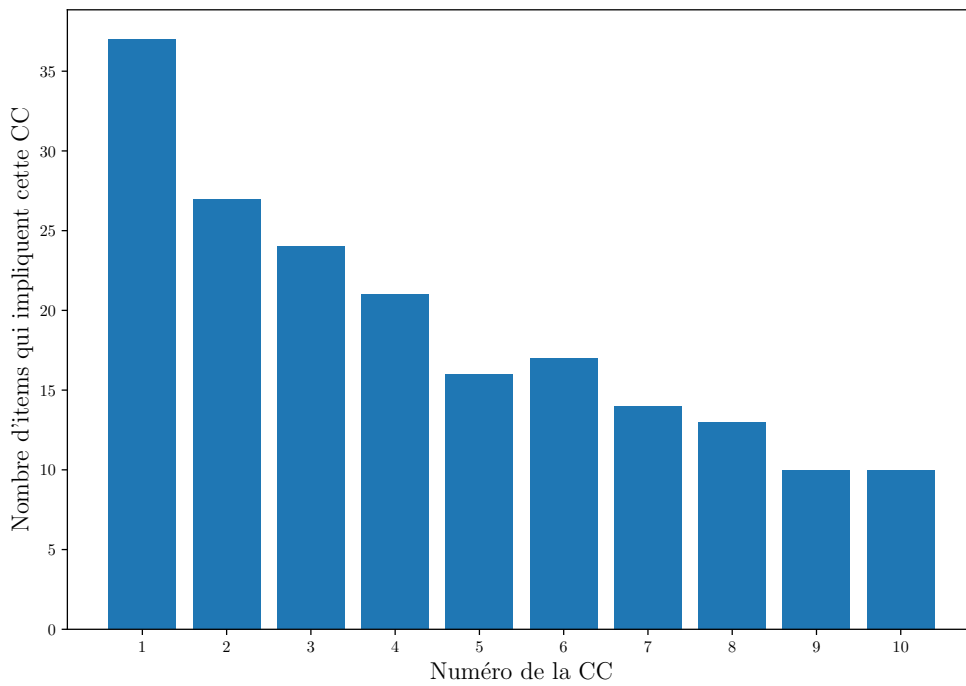


FIGURE 4.12 – Comparaison du nombre d'items qui mettent en jeu chaque CC de la q-matrice aléatoire générée selon notre protocole à la figure 4.3 (voir sous-section 4.2.3)

4.5.6 Quel est l'impact du choix du paramètre des stratégies θ -*threshold* et μ -*back* sur la performance des apprenants ?

Motivation

Dans (M. M. KHAJAH et al., 2014), les auteurs comparent les performances relatives d'une grille de paramètres θ et μ pour leurs stratégies θ -*threshold* et μ -*back*. La performance relative de chaque stratégie est calculée par rapport à la performance obtenue par la séquence optimale de révisions d'items.

Notre modèle simulateur d'apprenant (DAS3H) étend le modèle DASH (LINDSEY et al., 2014). Le module temporel de DASH h_θ , responsable de la modélisation de l'apprentissage et de l'oubli dans ce modèle, a lui-même été inspiré par ACT-R et MCM, deux modèles de la mémoire humaine qui ont été utilisés par M. M. KHAJAH et al. (2014). Nous utilisons également deux stratégies heuristiques de sélection de CC, θ -*threshold* et μ -*back*, qui sont similaires à celles employées par M. M. KHAJAH et al. Nous avons enfin proposé une extension de la stratégie θ -*threshold* pour sélectionner un item mettant en jeu *plusieurs* CC prometteuses.

Compte tenu de ces similitudes (modèles et stratégie de sélection) et de ces différences (sélection de plusieurs CC en même temps dans notre situation), nous voulons savoir si nos résultats concordent avec ceux de M. M. KHAJAH et al. (2014). En outre, la problématique de cette partie revêt une importance pratique essentielle : puisqu'il serait coûteux de tester sur de vrais apprenants une grille de paramètres pour les stratégies θ -*threshold* et μ -*back*, il est préférable de les analyser au préalable à l'aide de simulations, comme (M. M. KHAJAH et al., 2014; LINDSEY et al., 2014).

Résultats

θ -*threshold* Ainsi, nous comparons les performances des apprenants pendant toute la période de rétention pour différents paramètres θ , pour la stratégie θ -*threshold* de base (sélection d'item *Random CC*) et la stratégie θ -*threshold* (multi.). Nos résultats, relatifs à la stratégie de sélection de CC *Random*, sont représentés dans les figures 4.13 et 4.15. Ces résultats sont issus de nos expérimentations de la section 4.4. Pour mieux comprendre ces résultats, nous mettons à chaque fois ces figures en regard d'une comparaison du score PMR_R de la stratégie, ventilé par paramètre θ et par indice d'introduction de CC (figures 4.14 et 4.16).

Notons toutefois que cette comparaison avec les résultats de M. M. KHAJAH et al. (2014) doit être évaluée avec prudence, puisque M. M. KHAJAH et al. ont eu la possibilité de calculer le planning optimal de révisions d'items, ce qui n'était pas possible pour nous du fait du grand nombre de combinaisons de CC dans chaque q-matrice générée (cf. section 4.4). Leurs résultats ont ensuite été calculés relativement aux résultats du planning optimal. Nous nous tenons donc à des comparaisons de tendances, sans chercher à comparer les métriques de performance brutes.

Nous observons des tendances similaires entre les différentes figures et ces tendances sont en outre cohérentes avec les résultats obtenus par M. M. KHAJAH et al. (2014) : les performances sont très stables entre $\theta = 0$ et $\theta = 0,5$ mais elles diminuent très fortement après ce seuil. Cette diminution est plus marquée pour la version multi-CC de θ -*threshold*. Ces résultats peuvent provenir d'un arbitrage entre proposer des items trop difficiles (sur lesquels les apprenants sont donc le plus susceptibles d'échouer, ce qui entraîne une mémorisation plus faible en raison de la zone d'échantillonnage de θ qui a été choisie dans nos expériences de la section 4.4) et la proposition d'items trop faciles (qui ne ferait pratiquer l'apprenant que ce qu'il maîtrise déjà).

Dans les figures 4.14 et 4.16, nous représentons la métrique PMR_R moyennée par indice d'introduction de CC (les indices plus faibles concernent les CC introduites au début du processus d'apprentissage) d'une grille de valeurs θ pour θ -*threshold* et θ -*threshold* (multi.). Ces données proviennent de l'expérience de la section 4.4, la stratégie de sélection d'items

est donc la stratégie *Random CC*. On peut voir que les valeurs élevées (courbes rouges) des θ tendent à préférer la sélection de CC déjà connues, en particulier les CC qui ont été apprises dans les premiers blocs. Cependant, ce choix a des rendements décroissants. Cela pourrait provenir d'un effet « boule de neige » : au départ, les stratégies ne peuvent sélectionner que les CC qui ont été introduites au début. Pour toutes les stratégies, ces CC commencent à être connues et la probabilité de bonne réponse sur elles est donc élevée. Mais au fur et à mesure que l'apprenant progresse, les stratégies avec les θ les plus faibles deviennent meilleures que celles avec les θ les plus élevés, car celles avec les θ les plus élevés restent bloquées à faire réviser les CC déjà maîtrisées, et délaissent donc les CC nouvellement introduites. L'oubli sur les CC du début ne doit pas être suffisant pour faire réviser les CC les plus récentes. Nous observons le même type de résultats dans la figure 4.16, pour la version multi-CC de θ -*threshold*. Cependant, cette tendance est ici accentuée, en raison de la caractéristique multi-CC de la stratégie.

μ -back Nous réalisons le même type d'analyse avec les paramètres μ (stratégie μ -back, stratégie de sélection d'item *Random CC*). Les résultats sont représentés dans la figure 4.17. On peut voir ici que la diminution de la performance de la stratégie est linéaire en le paramètre μ .

La figure 4.18 représente le score PMR_R , ventilé par ordre d'introduction de CC et par la valeur du paramètre μ . Ici, il est logique que les CC les plus anciennes soient les meilleures, car elles ont été révisées plusieurs fois au début et pendant l'apprentissage (davantage d'items de révision concernent en effet ces CC). Il est également compréhensible que la stratégie du *3-back* connaisse une forte diminution après la CC 7 car c'est la dernière CC qui a pu être revue; on observe la même diminution (avec la même forme) mais 1 (respectivement 2) CC après pour les stratégies *2-back* et *1-back*. Étant donné que la stratégie *1-back* est meilleure que ses homologues (voir paragraphe 4.5.6), sa capacité à revoir les CC de la fin du programme doit compenser la perte due au fait qu'elle ne fait pas autant réviser les premières CC que ses homologues.

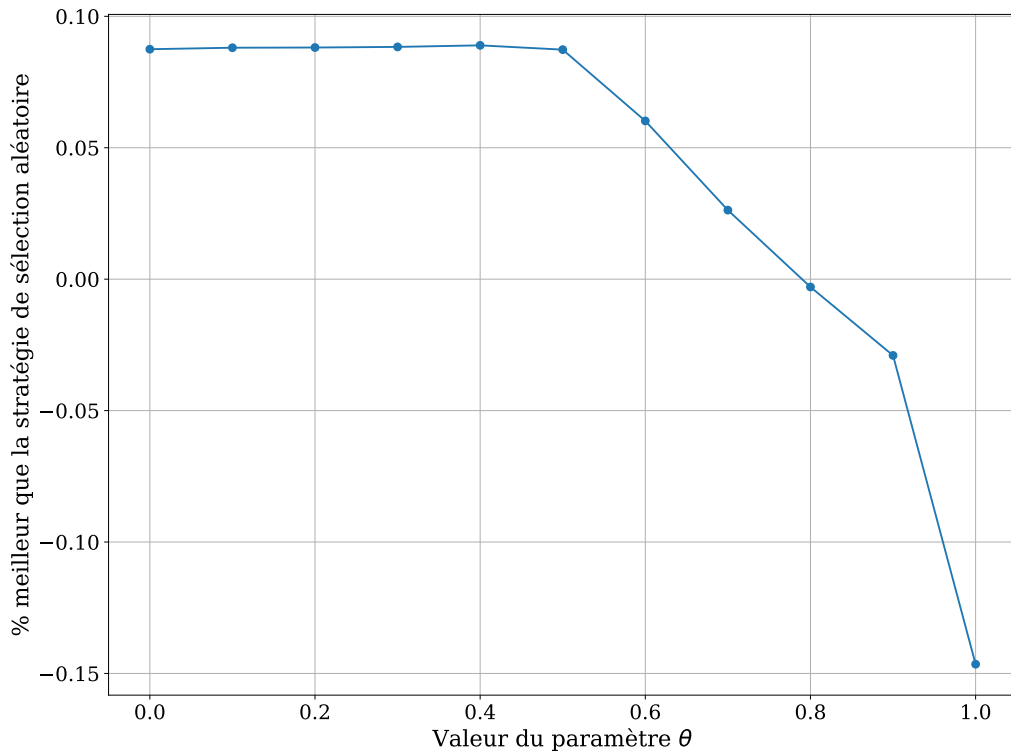


FIGURE 4.13 – Comparaison du score PMR_R moyen, relatif à la stratégie de sélection de CC *Random*, pour différentes valeurs de θ (θ -threshold, sélection d’item *Random CC*)

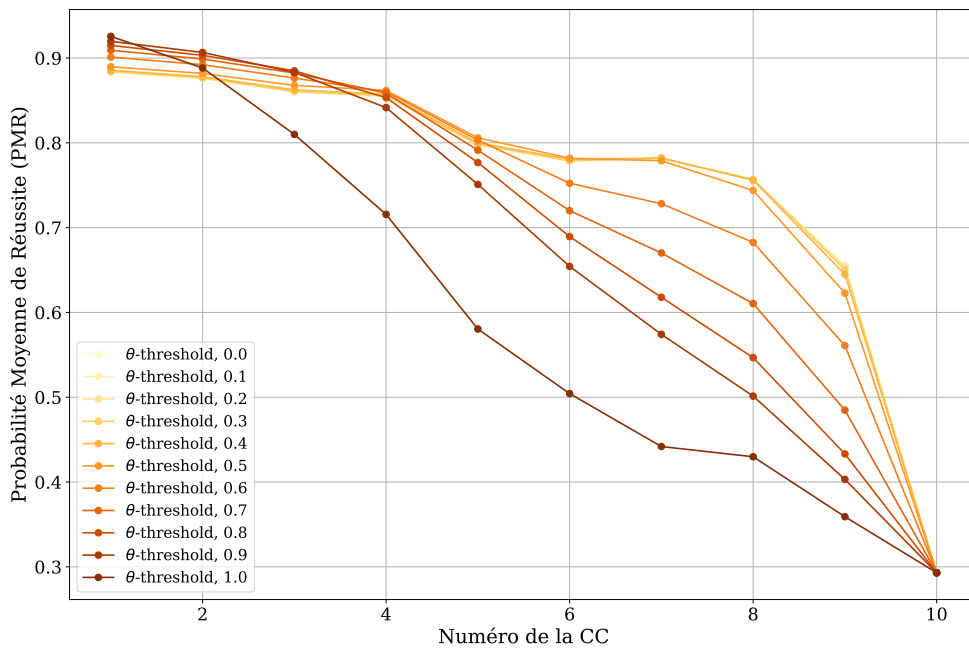


FIGURE 4.14 – Comparaison du score PMR_R par indice d’introduction de CC et paramètre θ (stratégie θ -threshold, expériences de la section 4.4)

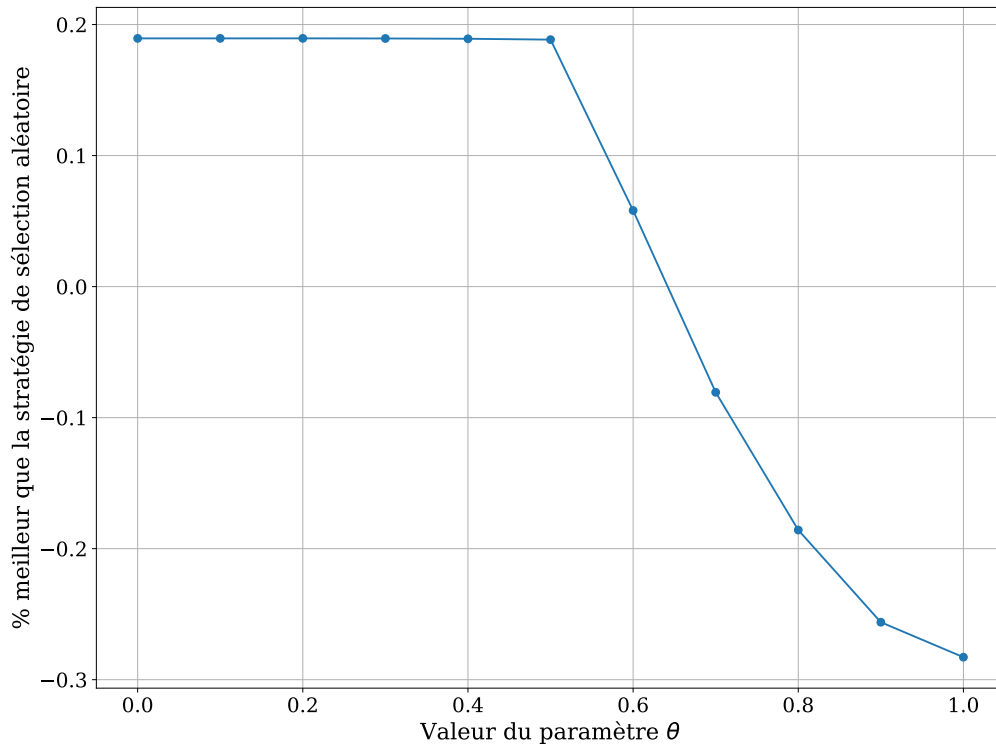


FIGURE 4.15 – Comparaison du score PMR_R moyen, relatif à la stratégie de sélection de CC *Random*, pour différentes valeurs de θ (θ -threshold (multi.), expériences de la section 4.4)

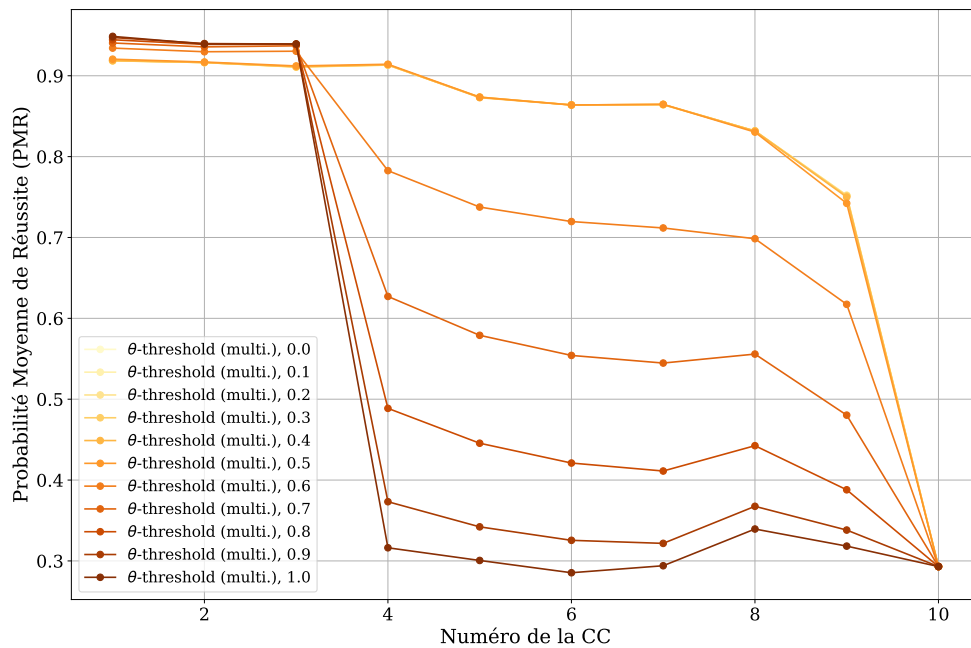


FIGURE 4.16 – Comparaison du score PMR_R par indice d'introduction de CC et paramètre θ (stratégie θ -threshold (multi.), expériences de la section 4.4)

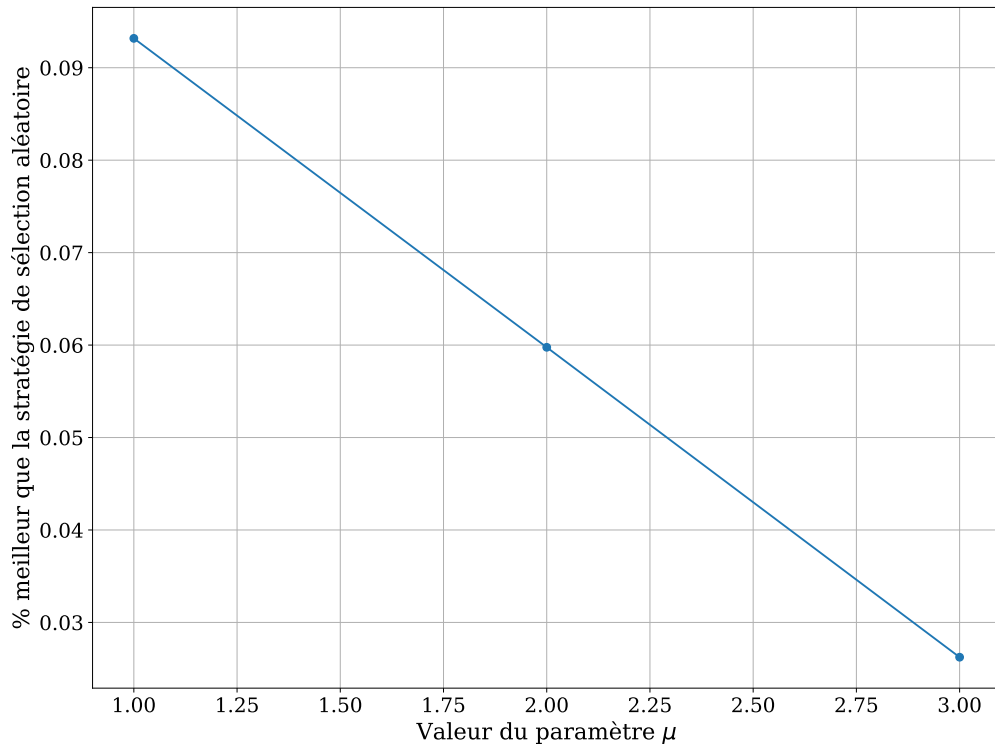


FIGURE 4.17 – Comparaison du score PMR_R moyen en fonction de différentes valeurs de μ (stratégie μ -back, expériences de la section 4.4)

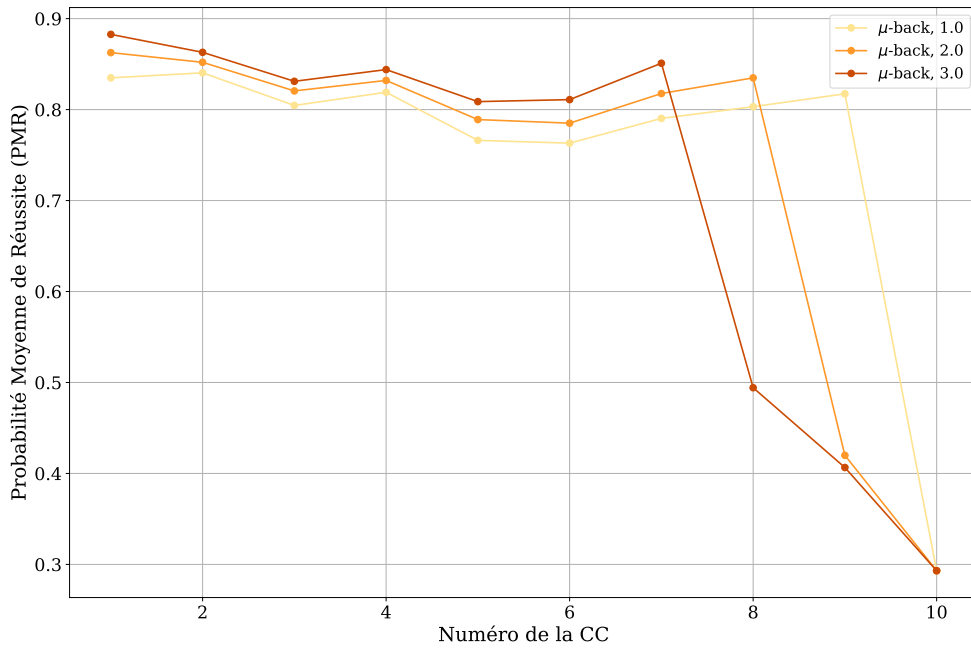


FIGURE 4.18 – Comparaison du score PMR_R par indice d'introduction de CC et paramètre μ (stratégie μ -back, expériences de la section 4.4)

4.6 Conclusion

Dans ce chapitre, nous nous sommes intéressés à l'optimisation de la maîtrise à long terme de CC par des apprenants à l'aide d'heuristiques de sélection pour les algorithmes d'espacement adaptatif et personnalisé de l'apprentissage humain.

Nous avons d'abord construit un protocole de comparaison des heuristiques d'espacement, décrit à la section 4.2. Ce protocole est réaliste et fondé sur des simulations de trajectoires d'apprentissage et de rétention chez des apprenants synthétiques. Nous avons utilisé pour ce faire le modèle DAS3H, présenté au chapitre 3. Les paramètres de DAS3H ainsi que les q -matrices de chacune des simulations ont été générés aléatoirement selon des distributions de probabilité spécifiées, afin d'induire une grande variabilité de comportements d'apprentissage et d'oubli tout en contrôlant les caractéristiques générales de ces comportements.

À la section 4.3, nous avons ensuite adapté à notre cadre de recherche trois algorithmes d'espacement adaptatif de l'apprentissage, conçus à l'origine pour la révision de *flashcards*. Deux de ces algorithmes reposent sur un modèle de l'apprenant, comme le modèle DAS3H. Nous avons également proposé une nouvelle procédure gloutonne permettant de sélectionner de manière itérative le sous-ensemble de CC le plus prometteur au lieu de choisir uniquement la meilleure CC à n'importe quel moment. Grâce à cette procédure, nous avons étendu deux des heuristiques proposées pour sélectionner plusieurs CC à la fois. Avant la comparaison quantitative des performances de ces cinq heuristiques d'espacement à la section 4.4, nous en avons proposé une comparaison qualitative.

Ensuite, nous avons mené des expériences sur 500 apprenants simulés afin de comparer l'efficacité de ces heuristiques de sélection de CC sur la rétention en mémoire à long terme des apprenants. Pour nous assurer de la robustesse de nos résultats, nous avons lancé 100 fois ces simulations en générant aléatoirement un nouvel ensemble de paramètres à chaque fois. Nous avons nous-mêmes implémenté en Python l'intégralité des expérimentations, du protocole de simulation jusqu'aux stratégies d'espacement. Ce code, qui permet de répliquer nos expériences, a été rendu public sur GitHub¹². Nous avons montré avec ces expériences qu'une simple stratégie non-adaptative telle que μ -back pouvait être un planificateur de révisions presque aussi efficace que des heuristiques adaptatives dans la plupart des situations. Bien que présentant des performances similaires à μ -back du point de vue de la rétention en mémoire, θ -threshold présente également l'avantage d'améliorer la probabilité de réussite de l'apprenant *durant* l'apprentissage, pas seulement *après*. À l'inverse, notre algorithme *Greedy* a montré de mauvaises performances pendant l'apprentissage des apprenants et n'était que légèrement meilleur que toutes les autres stratégies de sélection de CC unique pendant la phase de mémorisation. Il s'est cependant avéré plus robuste à la variabilité des comportements d'apprentissage et d'oubli que toutes les autres stratégies de sélection de CC unique. Enfin, nous avons également montré que la procédure gloutonne de sélection de sous-ensemble de CC, proposée à la section 4.3, permettait d'obtenir chez les heuristiques compatibles (*Greedy* et θ -threshold) de meilleures performances que toutes les stratégies de sélection de CC unique.

12. https://github.com/BenoitChoffin/multiskill_adaptive_spacing

Chapitre 5

Espacement adaptatif de la révision de CC par apprentissage par renforcement profond

Dans le chapitre précédent, nous avons proposé différentes heuristiques d’espacement adaptatif de l’apprentissage et de la révision de CC et nous avons comparé leurs performances sur des trajectoires d’apprenants synthétiques. Nous avons montré que ces heuristiques se distinguaient au niveau de leurs performances, de leur robustesse, et de leur difficulté de mise en œuvre. En particulier, la procédure gloutonne de sélection de sous-ensemble de CC que nous proposons s’est avérée plus performante que toutes les heuristiques de sélection de CC unique.

Ces heuristiques présentent cependant des limites. Pour reprendre la classification proposée par DOROUDI, ALEVEN et al. (2019), la plus sophistiquée de ces stratégies, *Greedy*, sélectionne la meilleure CC à faire réviser en faisant de l’*optimisation myope*. Ceci signifie qu’elle ne considère que les conséquences attendues de sa prochaine sélection, en faisant comme si celle-ci était la dernière de la séquence de révisions. Par conséquent, on peut se demander si la prise en compte de plusieurs révisions dans le futur, au lieu d’une seule, ne pourrait pas améliorer les performances des algorithmes. En outre, ce sont des heuristiques construites à priori, qui ne s’améliorent pas au fil des interactions avec des apprenants.

Nous proposons dans ce chapitre de changer de perspective pour le développement d’algorithmes d’espacement adaptatif de l’apprentissage de CC. Au lieu de nous appuyer sur des stratégies construites à priori, bien que motivées par la théorie en psychologie cognitive, nous explorons le développement d’algorithmes qui *apprennent* la politique optimale de sélection de CC en interagissant avec des apprenants.

Nous utilisons pour cela le cadre de l’apprentissage par renforcement pour optimiser des politiques d’espacement de révisions de CC. Nous faisons ce choix car l’apprentissage par renforcement est particulièrement adapté à la prise de décision séquentielle dans un environnement incertain. Par l’exploration des différentes actions disponibles (les items de révision) dans différentes situations, l’apprentissage par renforcement affine peu à peu sa politique de sélection pour maximiser un critère objectif donné. La grande force de ces méthodes est qu’elles ne nécessitent pas forcément de connaître ni la manière dont les apprenants apprennent et oublient ni l’objectif à atteindre (vitesse d’apprentissage, temps passé sur les révisions, etc.), mais simplement de pouvoir interagir avec un environnement et de recevoir des récompenses pour les actions prises. Nous sommes particulièrement intéressés par la comparaison de nos heuristiques conçues à priori avec une telle politique, qui est optimisée par des interactions avec des apprenants. Notons toutefois que l’on ne cherche pas dans ce chapitre à épuiser le sujet de l’utilisation de l’apprentissage par renforcement pour l’espacement adaptatif : au contraire, nous utilisons ici ces méthodes pour concevoir un algorithme qui nous permette de répondre à nos problématiques d’une autre manière.

Par ailleurs, nos simulations offrent un cadre de comparaison favorable aux stratégies *fondées sur un modèle*, puisqu'elles utilisent le même modèle pour leur sélection que celui utilisé pour la simulation de trajectoires. Comme le rappellent DOROU DI, ALEVEN et al. (2017), cette méthode risque de surestimer les performances des politiques de recommandation¹. Nous souhaitons donc évaluer l'effet du décalage de modèle entre modèle simulateur de trajectoires et modèle utilisé pour la recommandation dans ce cadre d'apprentissage par renforcement.

Nous investiguons ici les deux dernières problématiques évoquées au chapitre 1 :

1. *Peut-on faire apprendre à un algorithme la politique optimale de sélection d'items pour maximiser la mémorisation à long terme des apprenants? Une telle approche, fondée sur les données, parvient-elle à surpasser des heuristiques conçues à priori? Combien d'interactions avec des apprenants sont nécessaires avant d'obtenir une politique satisfaisante?*
2. *Quel est l'impact du décalage de modèle (en anglais, "model mismatch") entre modèle générateur de trajectoires d'apprentissage et modèle servant à la recommandation?*

Dans ce chapitre, nous commençons par faire quelques rappels sur l'apprentissage par renforcement, puis nous formulons le problème de l'espacement adaptatif et personnalisé de révisions de CC comme un problème d'apprentissage par renforcement. Nous proposons et implémentons ensuite AC4S, notre nouvel algorithme d'espacement adaptatif de CC fondé sur une architecture d'apprentissage par renforcement profond. Nous détaillons également les expériences que nous menons pour comparer ce nouvel algorithme aux heuristiques présentées au chapitre 4 : nous précisons d'abord quelques détails sur notre protocole expérimental avant d'en venir aux résultats. La section suivante est consacrée à une analyse plus poussée de ces résultats à l'aide d'expériences complémentaires. Enfin, nous concluons ce chapitre en synthétisant nos résultats.

5.1 Apprentissage par renforcement et espacement adaptatif de l'apprentissage de CC

5.1.1 Rappels sur l'apprentissage par renforcement (profond)

L'apprentissage par renforcement (SUTTON et al., 1998) est une sous-branche de l'apprentissage automatique qui s'intéresse à l'optimisation de prises de décision séquentielle en environnement incertain. Plus précisément, elle s'attache à développer des agents qui apprennent à partir de leur expérience d'interactions passées avec leur environnement à agir au mieux pour optimiser une certaine fonction objectif. À force d'essais et d'erreurs, l'agent d'apprentissage par renforcement apprend progressivement à agir de manière optimale dans un environnement qu'il ne maîtrise pas : l'apprentissage par renforcement ne nécessite en effet pas que l'agent connaisse le fonctionnement de son environnement. Il a simplement besoin de pouvoir interagir avec celui-ci.

Dans sa formulation la plus générale, l'apprentissage par renforcement concerne l'optimisation de la prise de décision d'un agent A interagissant en temps discret avec son environnement E . À chaque étape, A effectue une action a_t sur la base d'une observation o_t que lui fournit l'environnement. E lui envoie alors une récompense r_t , passe de l'état s_t à s_{t+1} et envoie l'observation o_{t+1} à A . L'objectif de l'agent est de maximiser une somme pondérée des récompenses reçues tout au long des interactions avec l'environnement.

Généralement, l'apprentissage par renforcement s'appuie sur des processus de décision markoviens (*Markov Decision Processes*, ou MDP en anglais). Les MDP sont des processus stochastiques de contrôle en temps discret, définis comme des tuples $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$:

1. Bien que dans nos simulations, les apprenants synthétiques suivent la décision prise par l'algorithme, des apprenants réels seraient libres de respecter ou non la sélection de l'algorithme. Nous parlons donc ici de *recommandation* pour désigner le choix optimal déterminé par la politique.

- \mathcal{S} l'espace d'états ;
- \mathcal{A} l'espace d'actions ;
- $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ la fonction de transition, qui définit la probabilité de passer à l'état s_{t+1} sachant l'état précédent s_t et l'action effectuée par l'agent a_t ;
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathcal{R}$ la fonction de récompense, qui définit la récompense obtenue en passant de l'état s_t à s_{t+1} après avoir choisi l'action a_t ;
- $\gamma \in [0, 1]$ le facteur d'escompte, qui sert à pondérer les récompenses futures de l'agent. Plus γ est élevé, plus les récompenses futures comptent dans la prise de décision actuelle de l'agent.

Comme on peut le voir ici, l'agent observe directement l'état réel du processus à un instant t pour effectuer l'action a_t . il n'y a donc pas d'observations distinctes des états dans un MDP.

L'objectif de l'algorithme d'apprentissage par renforcement est de trouver la politique π qui maximise l'espérance de récompenses actualisées futures. Une politique peut être déterministe ou stochastique. Si elle est déterministe, elle décrit de manière déterministe la relation entre un état s et l'action à choisir a . Si elle est stochastique au contraire, $\pi(s|a)$ est définie comme la probabilité d'effectuer l'action $a \in \mathcal{A}$ à l'état $s \in \mathcal{S}$.

L'apprentissage par renforcement profond (FRANÇOIS-LAVET et al., 2018), quant à lui, combine à la fois l'apprentissage par renforcement traditionnel et des méthodes héritées de l'apprentissage profond. Il est souvent utilisé quand l'espace d'états est en grande dimension, et sert alors à apprendre des représentations abstraites mais en plus faible dimension des états de l'environnement pour les exploiter ensuite dans l'algorithme par renforcement.

5.1.2 L'espace adaptatif et personnalisé des révisions de CC comme un problème d'apprentissage par renforcement profond

Formalisation en POMDP

Nous proposons ici de formuler notre problème d'espace adaptatif et personnalisé de la révision de CC (cf. section 1.2) comme un problème d'apprentissage par renforcement profond. Cette formulation a déjà été employée par le passé (REDDY, LEVINE et al., 2017; YANG et al., 2020) dans le cadre des révisions personnalisées de *flashcards*. Cependant, elle n'a à notre connaissance jamais été étendue au cas de l'apprentissage et de la révision de CC.

Nous réutilisons le même cadre de simulation que dans le chapitre 4, avec quelques ajustements que nous détaillons plus bas. Nous formulons donc notre problème de la manière suivante : un agent A (le système d'espace adaptatif) interagit avec son environnement E (l'apprenant). Les moments des révisions sont définis à l'avance (par un enseignant, par exemple). Donc, chaque étape consiste pour A à choisir pour E l'item sur lequel s'exercer à l'instant t de manière à maximiser la mémorisation à long terme d'un ensemble de CC. Les items et les CC sont donc reliés par une q-matrice. Après cette sélection, l'apprenant résout l'item et A observe la réponse binaire (juste ou fausse) de l'apprenant.

On voit ici une première différence avec le cadre général décrit à la sous-section 5.1.1 : A n'observe pas le véritable état des connaissances de l'apprenant, qui serait l'état s_t de l'environnement à un instant t . À la place, il observe la réponse de l'apprenant sur un item j , à un instant t . Nous employons donc, à la place des MDP, le cadre des POMDP, c'est-à-dire des processus décisionnels de Markov partiellement observables (*Partially Observable Markov Decision Processes* en anglais). Les POMDP sont définis par un tuple qui reprend les mêmes composantes que celles des MDP, mais en rajoute deux autres puisque des informations importantes sur l'environnement sont inobservables :

- \mathcal{O} l'espace d'observations ;
- $O : \mathcal{O} \times \mathcal{S} \rightarrow [0, 1]$ la probabilité d'observer o_t en l'état s_t ;

On suppose que les observations $o \in \mathcal{O}$ sont indicatrices de l'état latent $s \in \mathcal{S}$ de l'apprenant.

Composants du POMDP

Espace d'états \mathcal{S} Comme notre modèle générateur de trajectoires d'apprenants est DAS3H, un état comprend :

- α_s le niveau de l'apprenant ;
- les paramètres de difficulté δ_j et de facilité β_k , respectivement des n items et des K CC ;
- $a_{s,k,w}$ et $c_{s,k,w}$ les compteurs de tentatives et de réussites de s sur chacune des CC k dans chacune des fenêtres temporelles w ;
- θ le vecteur de coefficients associés à ces compteurs ;
- la q-matrice Q .

Notre espace d'états \mathcal{S} est donc :

$$\mathcal{S} = \mathbb{R}^{n+1} \times (\mathbb{R} \times \mathbb{R}^{2W} \times \mathbb{N}^{2W})^K \times \{0, 1\}^{n \times K}. \quad (5.1)$$

Espace d'actions \mathcal{A} Contrairement aux heuristiques que nous avons proposées à la section 4.3, notre algorithme d'apprentissage par renforcement choisit directement, à chaque instant déterminé, l'item à faire réviser à un apprenant. Nous faisons ce choix pour plusieurs raisons :

1. Nous voulons faire en sorte que l'algorithme gère le processus de sélection du début à la fin, sans introduire d'étape de sélection d'item.
2. Nous voulons laisser la possibilité à l'algorithme de choisir plusieurs CC à la fois.
3. Nous souhaitons voir si l'algorithme est capable de parvenir à une bonne politique de sélection même s'il n'a pas accès aux relations entre items et CC (par exemple avec une q-matrice).

Cependant, nous rappelons à notre lecteur que notre protocole expérimental (décrit à la section 4.2) prévoit que les CC soient introduites séquentiellement et dans un ordre précis. Cela signifie donc que toutes les CC ne peuvent pas être révisées à n'importe quel moment (seules celles qui ont été introduites au préalable le peuvent) et par conséquent, tous les items ne peuvent pas non plus être choisis à n'importe quel moment. Nous avons introduit à la section 4.3 les notions de CC et d'items acceptables pour cela.

Par conséquent, l'espace d'actions de l'algorithme n'est pas fixe : il dépend en réalité de l'ordre d'introduction des CC et de l'instant courant. Si l'on note donc \mathcal{A}_t l'espace d'actions de l'algorithme à un instant t , alors nous avons :

$$\mathcal{A}_t = \llbracket 1, c \times w_t \rrbracket \quad (5.2)$$

car les items sont ordonnés dans le temps et le nombre d'items disponibles à la révision à un instant t correspond dans notre protocole au numéro de cette semaine w_t multiplié par le nombre d'items générés par CC.

Notons que l'on pourrait facilement retirer cette dépendance de l'espace d'actions au temps t , si jamais dans un protocole expérimental différent il s'avère que toutes les CC sont disponibles à la révision à tout moment.

Récompenses Dans notre cadre, l'horizon temporel est *fini* car le nombre de révisions de l'apprenant (et donc, le nombre d'interactions entre A et E) est lui aussi fini.

La récompense que nous utilisons dans notre algorithme d'apprentissage par renforcement est la même que celle que nous avons proposée à la sous-section 4.2.4, la PMR_R fournie par le modèle générateur de trajectoires d'apprenants (DAS3H). Cette récompense est donc fournie à l'algorithme à la fin d'un épisode. Contrairement à SINHA (2019) et REDDY, LEVINE et al. (2017), nous n'utilisons pas de récompense immédiate r_t à chaque action que choisit l'agent A . Le décalage entre les actions effectuées par l'algorithme et la récompense à la fin rendent

cette tâche complexe à résoudre. Il est alors plus difficile pour A de déterminer quelle action dans quel état est responsable de l'augmentation ou la diminution de la récompense finale R_{finale} : il s'agit du problème de *credit assignment*. Nous discutons dans les perspectives de cette thèse (section 6.3) de quelques manières de rendre cette tâche plus aisée pour notre algorithme d'apprentissage par renforcement.

Définition 10 – Épisode. Un épisode de la simulation consiste en une unique trajectoire d'apprenant, depuis le début de la période d'apprentissage jusqu'à la fin de la période de rétention.

L'objectif de notre algorithme est de trouver une politique π qui maximise l'espérance conditionnelle suivante :

$$\mathbb{E}_{s_0, a_0, \dots} \left[\sum_{t=\tau_0}^T \sum_{k=1}^K \mathbb{P}(Y_{s,k,t} = 1) \middle| \pi \right] \quad (5.3)$$

avec τ_0 le début de la période de rétention et T sa fin.

Utiliser la PMR_R calculée avec DAS3H constitue à nos yeux un accès légitime de l'algorithme d'apprentissage par renforcement au modèle générateur de trajectoires. Dans une situation réelle, il serait nécessaire de faire passer aux apprenants des tests à différents intervalles de rétention, dans lesquels leur maîtrise des différentes CC du programme seraient examinées. C'est d'ailleurs ce protocole expérimental qui est généralement utilisé dans les expériences en psychologie sur l'espacement de l'apprentissage (N. J. CEPEDA, VUL et al., 2008). Les résultats à ces différents tests jouent le rôle d'indicateur de la maîtrise des CC du programme. Dans notre cas, par simplicité, nous préférons donner directement comme récompense la véritable PMR_R plutôt que de simuler différents tests à nos apprenants synthétiques. Nous supposons donc que les CC doivent être équitablement représentées dans les items des différents tests.

Transition La fonction de transition T est définie par le modèle DAS3H. Les paramètres statiques $\alpha_s, \delta_j, \beta_k, \theta$ ainsi que la q-matrice restent constants d'un état à l'autre, tandis que les compteurs de réussites et de tentatives dans chacune des tentatives sont augmentés en fonction de la q-matrice et de la réussite ou non de l'apprenant.

Observations Pour tester l'impact du décalage de modèle sur l'efficacité de notre algorithme, nous utilisons comme observation un vecteur dense décrivant l'état des connaissances de l'apprenant. Cet état correspond à la couche cachée d'un LSTM, entraîné au préalable sur des données issues du même modèle générateur, et mis à jour pour l'apprenant actuel à chaque fois que celui-ci résout un item dans le processus décrit à l'algorithme 1. Ici, le modèle utilisé pour obtenir cette observation est différent du modèle utilisé pour générer les trajectoires, ce qui nous permet d'évaluer l'impact de ce décalage.

Le modèle utilisé ici est une version modifiée du modèle *Deep Knowledge Tracing* (PIECH et al., 2015), dans laquelle l'entrée est une séquence de vecteurs x_1, x_2, \dots, x_T et où chaque x_t est un encodage one-hot de l'identifiant de l'item à l'instant t et de la justesse de la réponse de l'apprenant à cet item. Donc, $x_t \in \{0, 1\}^{2n}$ (avec n le nombre total d'items). La sortie du modèle y_t est un vecteur dont la composante i correspond à la probabilité, sachant les interactions passées et leurs résultats, que l'apprenant réponde correctement à l'item i . Il est nécessaire pour nous d'avoir les probabilités de bonne réponse au niveau des items et non pas au niveau des CC car notre algorithme d'apprentissage par renforcement choisit l'item et non pas la ou les CC pour l'apprenant. Nous estimons un seul modèle pour tous les apprenants.

Le modèle utilisé pour DKT est ici un LSTM (HOCHREITER et al., 1997), modèle connu des réseaux de neurones récurrents. Il comporte, dans l'ordre :

1. une couche de plongement vectoriel de dimension 256;

2. le LSTM avec une couche cachée de dimension 128;
3. une couche entièrement connectée de dimension 512 (fonction d'activation ReLU);
4. une couche softmax de dimension n le nombre d'items.

Nous utilisons de la régularisation par abandon (*dropout*), avec un taux d'abandon de 0,2. Nous utilisons également l'algorithme d'optimisation Adam (KINGMA et al., 2015) avec un taux d'apprentissage de 10^{-3} . Nous entraînons le modèle sur 100 itérations et gardons les paramètres du modèle à l'itération où l'AUC est maximale sur l'ensemble de données de validation.

Comme à la sous-section 4.5.3, nous générons d'abord un ensemble de trajectoires d'apprenants avec le modèle DAS3H et ses paramètres. Ces données initiales servent uniquement à entraîner DKT. Ensuite, ses paramètres sont gelés et utilisés à chaque étape pour obtenir l'état des connaissances latent de l'apprenant selon DKT. Cette observation o_t est donc de taille 128. DKT prend en compte chaque réponse de l'apprenant au fur et à mesure qu'elles arrivent afin de modifier l'état des connaissances estimé de l'apprenant et de mettre à jour l'observation o_t .

Nous tenons à insister sur le fait que nous utilisons ce LSTM *uniquement* pour obtenir une représentation *abstraite* et *dense* de l'état de l'apprenant à un instant t , générée par un modèle différent du modèle générateur de trajectoires. Nous n'utilisons pas ce modèle comme une approximation de la fonction de transition T de l'environnement.

5.2 Notre algorithme AC4S

Nous proposons dans cette section notre nouvel algorithme d'espace adaptatif des révisions de CC fondé sur une architecture d'apprentissage par renforcement profond. Nous appelons cet algorithme AC4S pour *Actor-Critic 4 Spacing*. Contrairement aux algorithmes d'apprentissage par renforcement fondés sur la valeur (*value-based*, ou algorithmes *action-valeur*) qui estiment soit la valeur d'un état ou d'un couple état-action pour dériver ensuite la politique optimale, nous choisissons une approche dite de « gradient de politique » (*policy gradient*). Ces méthodes estiment directement la politique à suivre en optimisant une fonction objectif et l'utilisent ensuite pour choisir les actions à effectuer².

Nous employons un algorithme de la famille des algorithmes *Actor-Critic* (SUTTON et al., 1998). Les algorithmes *Actor-Critic* estiment à la fois la politique $\pi(a|s)$ (appelée l'*acteur*) et un critique (par exemple, la fonction valeur $V(s)$). Intuitivement, l'acteur apprend à choisir en fonction de l'état de l'environnement l'action qui maximise la somme actualisée des récompenses futures tandis que le critique évalue les actions effectuées par l'acteur. L'acteur et le critique peuvent être modélisés par un réseau de neurones. Au fur et à mesure que l'agent accumule de l'expérience en interagissant avec son environnement, acteur et critique sont mis à jour et deviennent meilleurs dans leurs rôles respectifs.

Plus précisément, nous utilisons une variante de l'algorithme *Advantage Actor-Critic* (A2C). A2C est une version synchrone de l'algorithme *Asynchronous Advantage Actor-Critic* (A3C) (Mnih et al., 2016). A3C utilise plusieurs agents indépendants qui interagissent en parallèle avec une instance différente de l'environnement et qui mettent à jour périodiquement (et de manière asynchrone) l'acteur et le critique. Une fois qu'un agent termine un épisode, il récupère les paramètres globaux de l'acteur et du critique et met à jour son acteur et son critique.

Le fonctionnement d'AC4S est synthétisé dans l'algorithme 4 et schématisé dans la figure 5.1. L'algorithme commence par initialiser la politique $\pi(a_t|o_t, \mu_t; \theta)$ et la fonction valeur $V(o_t; \theta)$. Notons tout d'abord que les notations ont ici un peu changé : μ_t représente en effet le masque d'actions appliqué à la politique π à l'instant t et o_t représente l'observation que

2. Elles peuvent également maintenir en parallèle une estimation des fonctions valeur ou Q-valeur mais celles-ci ne sont pas utilisées directement pour le choix de l'action a_t .

Algorithme 4 Algorithme AC4S

-
- 1: **procédure** LEARN AC4S POLICY(N nombre d'épisodes, $n_{révisions}$ nombre total de révisions, DKT)
 - 2: **pour** *épisode* de 1 à N **faire** ▷ Cf. algorithme 2.
 - 3: Mettre le gradient à zéro : $d\theta \leftarrow 0$;
 - 4: Recevoir l'observation initiale o_0 à partir de DKT;
 - 5: **pour** t de 1 à $n_{révisions}$ **faire** ▷ Cf. algorithme 1.
 - 6: Calculer le masque d'actions μ_t ;
 - 7: Échantillonner l'action a_t selon la politique $\pi(a_t|o_t, \mu_t; \theta)$;
 - 8: Sauvegarder $\pi(a_t|o_t, \mu_t; \theta)$ et $V(o_t; \theta)$;
 - 9: Passer de l'état s_t à s_{t+1} ;
 - 10: Recevoir la nouvelle observation o_t de DKT et la récompense $r_t = 0$; ▷ Pas de récompense immédiate
 - 11: Recevoir récompense finale $R_{finale} \leftarrow PMR_R$;
 - 12: **pour** $i \in \{1, \dots, n_{révisions}\}$ **faire**
 - 13: $A_i \leftarrow R_{finale} - V(o_i; \theta)$; ▷ Calcul de la fonction avantage
 - 14: $d\theta \leftarrow d\theta + \underbrace{\nabla_{\theta'} \log \pi(a_i|o_i, \mu_i; \theta')}_{\text{Gradient pour l'acteur}} \times A_i + \underbrace{\nabla_{\theta'} (L_{Huber}(V(o_i; \theta'), R_{finale}))}_{\text{Gradient pour le critique}}$
 - 15: Mettre à jour θ par rétropropagation;
 - 16: Effacer la sauvegarde des actions $\pi(a_t|o_t, \mu_t; \theta)$ et $V(o_t; \theta)$;
 - 17: **fin procédure**
-

reçoit AC4S de DKT. μ_t empêche que l'algorithme puisse choisir des actions (c'est-à-dire des items) qui mettent en jeu des CC non encore introduites. Concrètement, cela consiste à mettre à zéro les probabilités de tels items. Nous modélisons π et V par un réseau de neurones commun. Nous représentons ses paramètres par le symbole θ dans l'algorithme et ce qui suit. Ce réseau prend en entrée les observations o_t de l'environnement (cf. partie 5.1.2) et est constitué d'une seule couche cachée de 10 unités avec la fonction d'activation ReLU (ou Unité Linéaire Rectifiée). La sortie du modèle dépend de l'acteur et du critique : pour la tête de la politique, nous avons une couche softmax avec $n_{actions}$ unités, et une seule unité pour la tête liée à la valeur. L'algorithme d'optimisation est Adam (KINGMA et al., 2015), avec un taux d'apprentissage de 2×10^{-3} .

Chaque épisode correspond à la trajectoire d'apprentissage et de mémorisation d'un nouvel apprenant simulé. À chaque révision planifiée, l'algorithme échantillonne l'action a_t selon la distribution de probabilité $\pi(a_t|o_t, \mu_t; \theta)$. L'algorithme sauvegarde ensuite la probabilité de l'action choisie $\pi(a_t|o_t, \mu_t; \theta)$ et la valeur de l'observation o_t , $V(o_t; \theta)$, pour qu'elles soient utilisées par la suite dans la mise à jour des paramètres θ de l'acteur et du critique. L'environnement met à jour son état s_t à partir de la réponse y_t de l'apprenant sur a_t et DKT met également à jour son estimation de l'état des connaissances de l'apprenant pour fournir la prochaine observation à AC4S.

Comme indiqué à la partie 5.1.2, notre algorithme ne reçoit pas de récompenses immédiates au cours d'un épisode. Ceci revient dans notre implémentation à donner $r_t = 0$ à l'algorithme pour tout t non final. Une fois que l'épisode est terminé, la PMR_R de l'apprenant courant est fournie comme récompense finale pour tout l'épisode.

Après la fin d'un épisode, nous pouvons enfin mettre à jour la politique $\pi(a|o, \mu; \theta)$ et la fonction valeur $V(o; \theta)$ grâce à l'expérience accumulée dans l'épisode. Nous utilisons pour cela la fonction avantage (MNIH et al., 2016) :

$$A(o_t, a_t) = Q(o_t, a_t) - V(o_t) = \underbrace{r_{t+1} + \gamma V(o_{t+1})}_{=R_{finale}} - V(o_t) \quad (5.4)$$

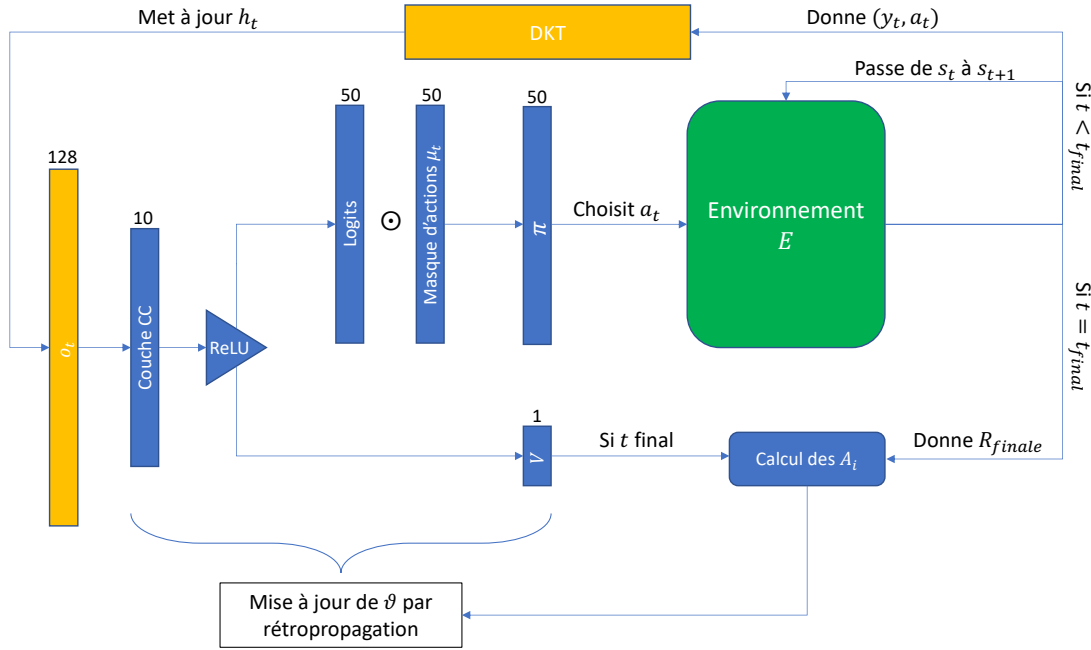


FIGURE 5.1 – Diagramme du fonctionnement de notre algorithme AC4S. Les nombres au-dessus des rectangles désignent le nombre d’unités de cette partie du réseau de neurones et « Couche CC » signifie « couche complètement connectée ». L’opérateur \odot représente quant à lui la multiplication élément par élément entre deux vecteurs. Enfin, y_t représente la réponse de l’apprenant à l’item a_t .

Intuitivement, cette fonction avantage $A(o, a)$ capture la différence de valeur à l’observation o (ou l’état s) entre l’action a et la moyenne des valeurs de toutes les autres actions. Si $A(o, a)$ est positif, cela signifie qu’il faut favoriser cette action, et θ est alors mis à jour dans cette direction. Si $A(o, a)$ est négatif, c’est l’inverse qui se produit. En outre, cette fonction avantage permet également de réduire la variance du gradient de la politique (Mnih et al., 2016), problème soulevé par les méthodes *policy gradient*.

Comme nous partageons le réseau de neurones paramétré par θ entre l’acteur et le critique, nous mettons à jour θ par rétropropagation à l’aide de la fonction de perte de l’acteur et celle du critique. Pour le critique, nous utilisons la fonction de perte Huber. Nous définissons cette dernière de la manière suivante :

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \quad L_{Huber}(x, y) = \frac{1}{n} \sum_i z_i \quad (5.5)$$

$$\text{avec } z_i = \begin{cases} \frac{1}{2}(x_i - y_i)^2 & \text{si } |x_i - y_i| < 1; \\ |x_i - y_i| - \frac{1}{2} & \text{sinon.} \end{cases} \quad (5.6)$$

5.3 Expériences

Dans cette section, nous commençons par détailler le protocole expérimental que nous utilisons pour nos expériences (paramètres des simulations, métriques de performance, stratégies concurrentes considérées et implémentation). Enfin, nous décrivons les résultats de ces expériences.

5.3.1 Protocole expérimental

À part quelques différences que nous décrivons dans ce qui suit, nous employons à nouveau en grande partie le protocole expérimental que nous avons introduit au chapitre 4.

Pour des raisons de commodité, nous ne réutilisons pas les mêmes apprenants dans chaque stratégie, au sein d’une même exécution de la simulation. À la place, nous choisissons de générer aléatoirement à chaque fois un nouvel apprenant, selon le protocole décrit au chapitre 4. En outre, au lieu de générer la réponse d’un apprenant de manière déterministe, nous l’échantillonons selon une loi de Bernoulli de paramètre $p = \mathbb{P}(Y_{s,j,t} = 1)$. Nous ne rajoutons plus de tentative à l’historique de l’apprenant pour symboliser l’introduction d’une nouvelle CC pour éviter de complexifier inutilement l’environnement. Enfin, nous autorisons les algorithmes à choisir des items impliquant la CC qui est introduite la semaine courante, afin de permettre aux algorithmes de choisir au moins une fois la dernière CC du programme. Ces modifications mineures n’entraînent que des changements marginaux dans les résultats des heuristiques.

Paramètres des simulations

Les paramètres de nos simulations sont identiques à ceux décrits à la sous-section 4.2.3. La seule différence réside dans le nombre d’items par CC que nous générons : au lieu d’être fixé à 20 items par CC, nous le réduisons à 5 par CC. Nous faisons ce choix pour faciliter la tâche de l’algorithme AC4S, car cela permet d’avoir un espace d’actions de taille 50 au lieu de taille 200. En outre, nous générons 1000 trajectoires d’apprenants synthétiques pour les heuristiques, 5000 pour AC4S et nous lançons 100 fois ce protocole.

Comme indiqué précédemment, nous avons deux modèles différents dans notre situation : un modèle *générateur*, qui sert à simuler des trajectoires d’apprenants synthétiques, et un modèle utilisé par AC4S. Ce dernier modèle est une version alternative de DKT (PIECH et al., 2015). Pour obtenir une représentation dense et abstraite de l’état de maîtrise des apprenants sur chacune des CC, nous générons avant chaque exécution de la simulation un ensemble de trajectoires synthétiques suivant DAS3H et les mêmes paramètres que pour la simulation principale. DKT est ensuite entraîné sur ces données, avant de pouvoir être utilisé pour obtenir les observations d’AC4S dans chaque simulation. Cette étape est bien entendu répétée au début de chaque nouvelle exécution du protocole principal de simulation, comme à la sous-section 4.5.3. Pour cette génération de trajectoires initiales, nous fixons à :

- $S = 500$ le nombre d’apprenants simulés ;
- $r = 3$ le nombre de révisions par semaine ;
- $K = 10$ le nombre de séances de révision.

Pour obtenir un nombre équivalent d’échantillons entre les CC, nous faisons le choix de sélectionner au hasard chaque item de révision pour ces apprenants parmi l’ensemble des items de la q -matrice. En outre, au sein d’une même séance de révisions, le délai en minutes entre deux révisions est échantillonné selon une loi uniforme discrète $\mathcal{U}_{\llbracket 0,60 \rrbracket}$. Le délai entre deux séances de révisions est quant à lui fixé à une semaine.

Métriques de performance

Pour faciliter la lecture de nos résultats, nous utilisons dans ces expériences seulement la métrique PMR_R (cf. sous-section 4.2.4). Comme au chapitre 4, nous mesurons la PMR pour chaque apprenant, à chaque semaine de la phase de rétention, et agrégeons ensuite ces résultats.

Stratégies concurrentes

Nous utilisons les mêmes heuristiques qu’à la section 4.4. Nous avons montré qu’elles obtenaient en moyenne de meilleures performances que la stratégie *Random* et nous espérons ainsi fournir des stratégies de référence fortes auxquelles comparer AC4S. DOROUDI, ALEVEN et al. (2019) montrent en effet que l’un des facteurs expliquant les résultats positifs obtenus

par un algorithme d'apprentissage par renforcement dans le domaine éducatif est la faiblesse des politiques de référence avec lesquelles il est comparé.

En ce qui concerne les paramètres des stratégies μ -back et θ -threshold, nous choisissons les paramètres optimaux que nous avons trouvés :

- $\theta = 0,4$ pour les stratégies θ -threshold ;
- $\mu = 0$ pour μ -back.

Implémentation

Nous implémentons l'environnement et les différents algorithmes sous Python. Nous utilisons notamment pour cela les bibliothèques `pytorch` et `gym`. Tout le code Python servant aux expérimentations décrites dans ce chapitre a été publié sur GitHub³.

5.3.2 Résultats

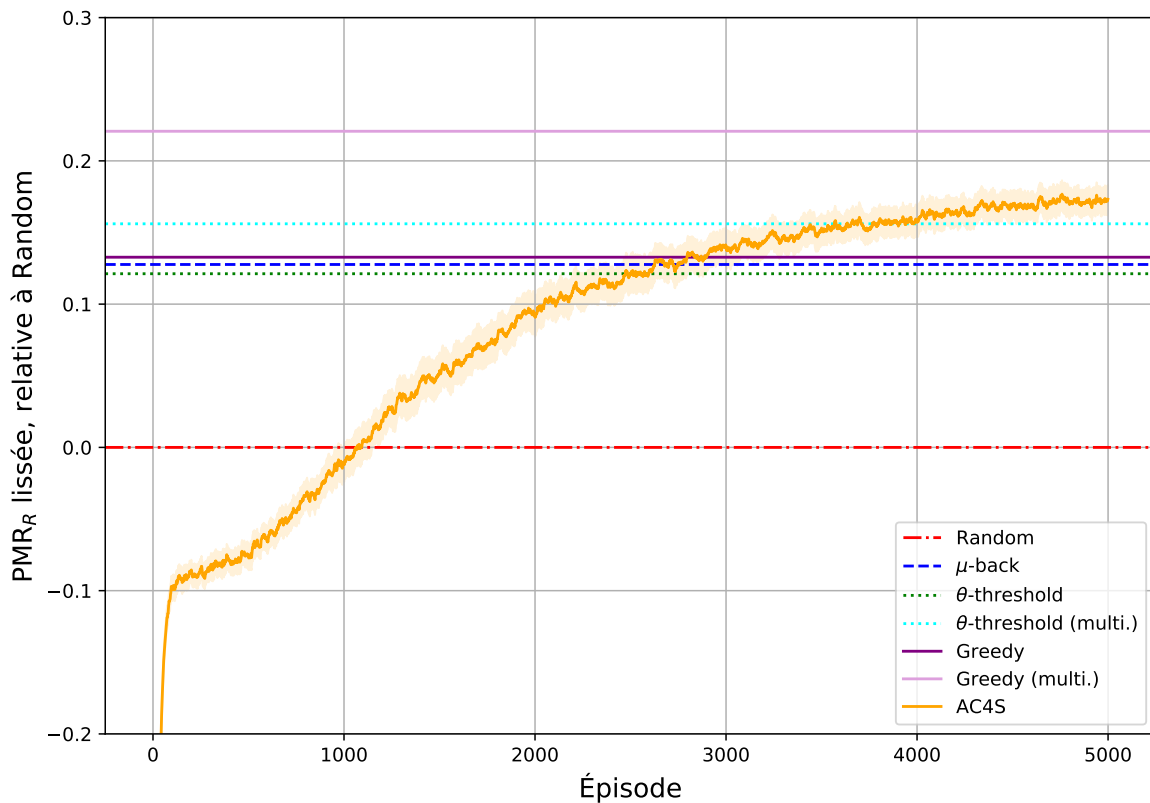


FIGURE 5.2 – Évolution de la PMR_R relative lissée d'AC4S en fonction des épisodes. Les droites horizontales représentent les PMR_R relatives des heuristiques concurrentes, moyennées sur tous les épisodes et toutes les simulations. La zone ombrée autour de la courbe orange représente un intervalle de confiance à 95 % pour AC4S.

Les résultats de ces expériences sont représentés dans les figures 5.2 et 5.3. L'évolution de la PMR_R relative (à *Random*, cf. section 4.4) et brute en fonction des épisodes d'AC4S est représentée en orange. Quant aux autres heuristiques d'espacement adaptatif, nous les avons représentées par leur moyenne puisque contrairement à AC4S, elles ne s'améliorent pas au fil des épisodes. Nous avons lissé l'évolution de la PMR_R d'AC4S à l'aide d'une moyenne mobile exponentielle, afin d'obtenir une visualisation moins bruitée de l'évolution des performances d'AC4S. La particularité de la moyenne mobile exponentielle par rapport à la moyenne mobile

3. <https://github.com/BenoitChoffin/ac4s>

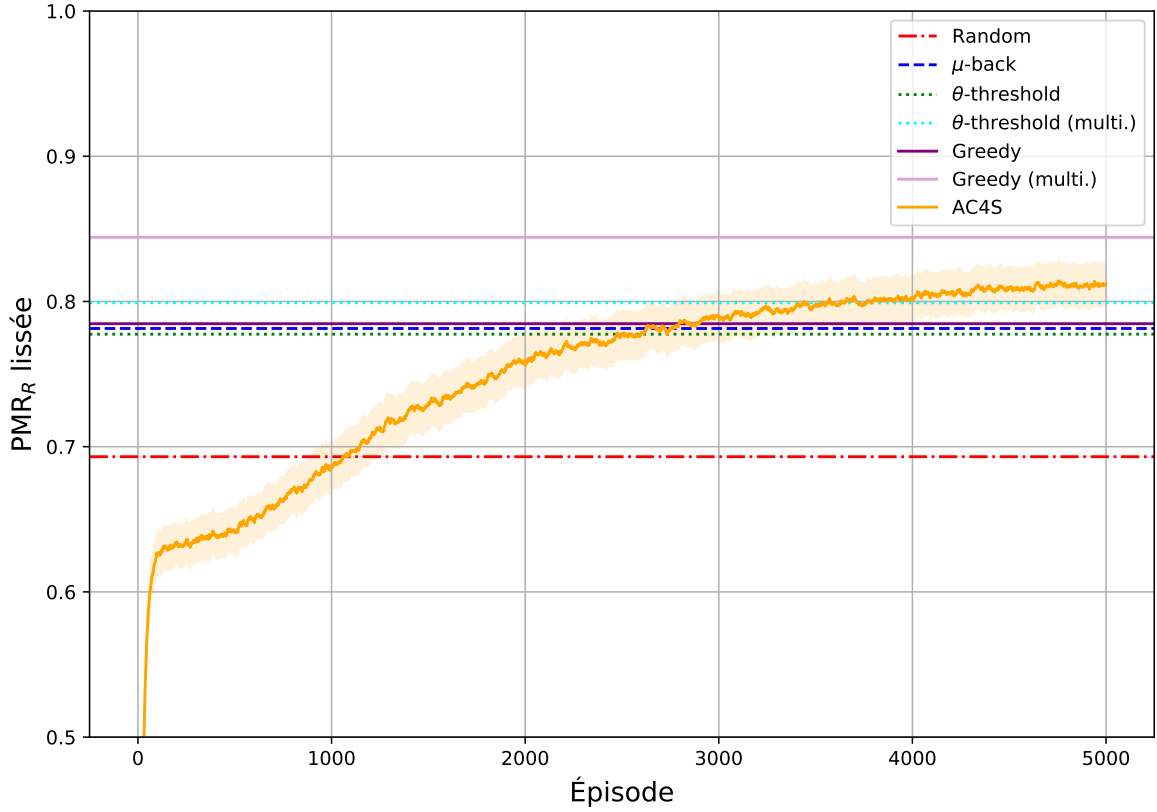


FIGURE 5.3 – Évolution de la PMR_R lissée d'AC4S en fonction des épisodes. Les droites horizontales représentent les PMR_R des heuristiques concurrentes, moyennées sur tous les épisodes et toutes les simulations. La zone ombrée autour de la courbe orange représente un intervalle de confiance à 95 % pour AC4S.

classique est qu'elle associe à chaque échantillon du passé une pondération qui décroît ensuite de manière exponentielle. Ainsi, cette méthode de lissage accorde un poids plus élevé aux observations les plus récentes mais ne supprime toutefois jamais totalement les observations les plus anciennes. La constante de lissage que nous utilisons ici est $\alpha = 0,05$. Appelons Y_t la valeur de notre série temporelle brute et $Y_t^{lissée}$ la valeur de la série lissée à l'instant t . Alors, si l'on fixe $Y_0^{lissée} = 0$, pour tout $t > 0$:

$$Y_t^{lissée} = \alpha Y_t + (1 - \alpha) Y_{t-1}^{lissée}. \quad (5.7)$$

Par ailleurs, pour représenter graphiquement la dispersion des moyennes de performances entre chaque exécution de la simulation, nous construisons un intervalle de confiance au niveau $\alpha = 95\%$, représenté par la zone ombrée en orange. Grâce au théorème central-limite, cet intervalle est asymptotiquement exact :

$$\left[\hat{\mu} - t_{\frac{1+\alpha}{2}} \frac{\hat{\sigma}}{\sqrt{n}}; \hat{\mu} + t_{\frac{1+\alpha}{2}} \frac{\hat{\sigma}}{\sqrt{n}} \right] \quad (5.8)$$

avec :

- $\hat{\mu}$ et $\hat{\sigma}$ respectivement la moyenne et la variance empiriques de l'échantillon ;
- n la taille de l'échantillon ;
- t_α le quantile d'ordre α d'une loi de Student à $n - 1$ degrés de liberté.

Tout d'abord, on peut voir que les heuristiques que nous avons déjà évaluées précédemment obtiennent des résultats similaires à ceux décrits à la section 4.4. Nous nous concentrons donc dans ce qui suit sur les performances d'AC4S.

On peut voir dans ces figures qu'AC4S parvient à obtenir de meilleures performances en termes de PMR_R que toutes les heuristiques de sélection de CC unique que nous avons proposées et comparés. Ce résultat est d'autant plus étonnant qu'AC4S n'a pas accès au modèle à l'origine des simulations de trajectoires, à savoir DAS3H. Il utilise à la place une variante de DKT (PIECH et al., 2015) pour modéliser l'évolution des états de connaissance des apprenants et utilise une représentation dense et abstraite de cet état comme observation. Il obtient de meilleures performances que des algorithmes comme *Greedy* qui ont, eux, un accès direct aux paramètres du modèle générateur de trajectoires. En revanche, si AC4S parvient à égaler et même dépasser les performances de θ -*threshold* (multi.), ce n'est pas le cas pour celles de *Greedy* (multi.).

Cependant, ces performances ont un coût assez élevé en termes d'échantillons, puisqu'AC4S ne dépasse les heuristiques développées au chapitre précédent qu'au bout d'environ 3000 épisodes. Dans une situation réelle, il ne serait bien évidemment pas envisageable de faire passer 3000 apprenants sur l'algorithme avant d'obtenir une politique convenable. Il faudrait alors par exemple pré-entraîner la politique sur des apprenants synthétiques puis l'affiner sur les vrais apprenants.

5.4 Discussion

Nous avons vu à la section précédente qu'AC4S parvenait à obtenir des performances équivalentes à celles de θ -*threshold* (multi.), mais que cela se payait par un grand nombre d'épisodes avant convergence. Dans cette section, nous explorons quelques pistes pour mieux comprendre les politiques d'espacement apprises par l'algorithme AC4S.

5.4.1 AC4S doit-il ses performances à un nombre supérieur de CC révisées ?

Le nombre total de CC choisies a une influence sur le score PMR_R d'un algorithme, comme nous l'avons montré dans la sous-section 4.5.1. Il est donc naturel de se demander si les bonnes performances de notre algorithme AC4S proviennent elles aussi d'un plus grand nombre de CC avec lesquelles les apprenants interagissent. Il était possible qu'AC4S apprenne une politique visant à sélectionner uniquement des items avec le nombre maximal de CC, comme la stratégie de sélection d'items *Max CC* de la section 4.3.

La figure 5.4 représente l'évolution du nombre total d'interactions de chaque apprenant (chaque épisode donc) avec les CC du programme. Comme pour les figures 5.2 et 5.3, nous lisons l'évolution de ces compteurs à l'aide d'une moyenne mobile exponentielle (constante de lissage $\alpha = 0,5$). De même, les compteurs des autres stratégies sont moyennés sur tous les épisodes, puis sur toutes simulations, puisqu'ils ne présentent pas d'évolution au cours des épisodes.

On peut voir dans cette figure que le nombre moyen de CC révisées par épisode augmente au fil des épisodes pour AC4S. Il dépasse le nombre moyen de CC des heuristiques de sélection de CC unique autour de l'épisode 1500. Son rythme de croissance diminue cependant à partir de 2500 épisodes. À la fin des 5000 épisodes, AC4S fait réviser en moyenne environ 67 CC par épisode. C'est presque 10 CC de moins que *Greedy* (multi.), et 10 CC de plus que la plupart des heuristiques de sélection de CC unique, qui sélectionnent en moyenne environ 57 CC par épisode.

En conclusion, AC4S se situe entre les stratégies de sélection de CC unique et les stratégies de sélection de CC multiples. Comme nous l'avons montré dans la sous-section 5.3.2, AC4S est en effet capable d'obtenir de meilleures performances que toutes les heuristiques de sélection de CC unique, et d'égaler les performances de θ -*threshold* (multi.). Bien qu'il sélectionne sensiblement plus de CC que les stratégies de sélection de CC unique, il en sélectionne en moyenne 10 de moins par épisode que *Greedy* (multi.).

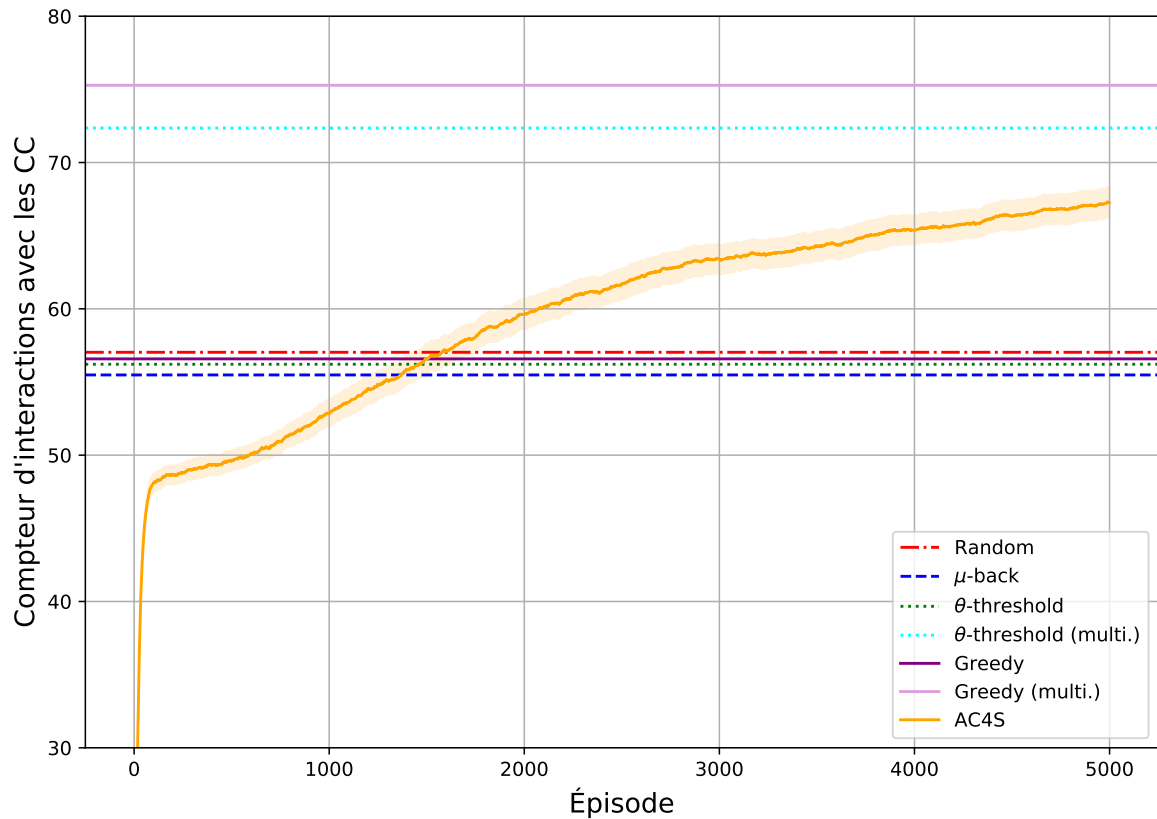


FIGURE 5.4 – Évolution du compteur lissé d'interactions avec les CC du programme par apprenant en fonction des épisodes. Les droites horizontales représentent les compteurs des heuristiques concurrentes, moyennés sur tous les épisodes et toutes les simulations.

5.4.2 AC4S choisit-il des items et des CC plus faciles que la moyenne ?

Ici, nous faisons l'hypothèse qu'AC4S apprend à sélectionner les items les plus faciles du programme. En effet, dans nos simulations, nous avons fixé les paramètres θ décrivant les courbes d'apprentissage et d'oubli de telle sorte que la réussite à une CC améliore plus la probabilité de bonne réponse future qu'un échec à cette CC. En outre, dans (UPADHYAY et al., 2018), les auteurs montrent que leur algorithme TPPRL fondé sur une architecture d'apprentissage par renforcement profond apprend à sélectionner les items les plus faciles. Nous souhaitons savoir si AC4S apprend également cela.

Dans les figures 5.5 et 5.6, nous représentons des diagrammes en boîte des difficultés des items et des CC choisis par AC4S et les différentes heuristiques de sélection de CC. Les données représentées sont les données agrégées sur toutes les exécutions des simulations, tous les épisodes pour les heuristiques et les 1000 derniers épisodes pour AC4S⁴.

Contre toute attente, on peut observer que les distributions des difficultés des items et des CC choisis par les différents algorithmes sont très similaires. AC4S n'apprend donc pas à choisir des items ou des CC spécifiquement plus faciles que le reste des algorithmes (y compris *Random*).

4. Nous ne sélectionnons que les données à partir du moment où la politique se stabilise.

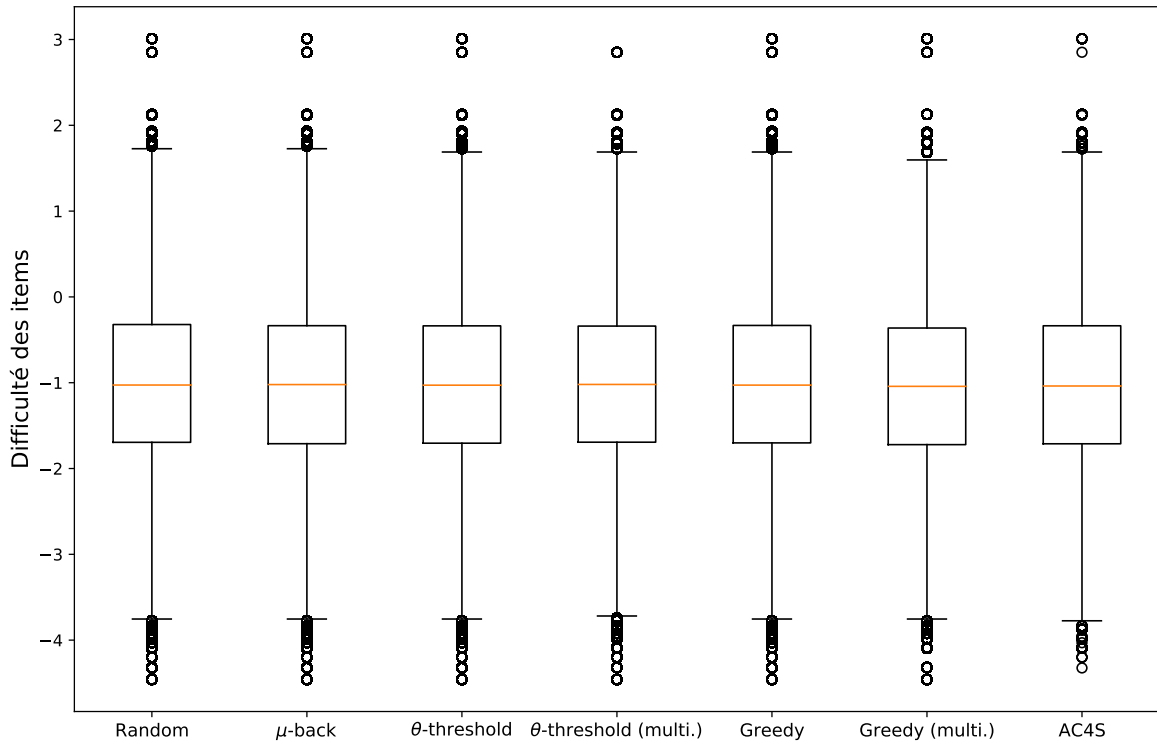


FIGURE 5.5 – Comparaison des distributions des difficultés des items sélectionnés par les heuristiques et par AC4S, toutes simulations confondues.

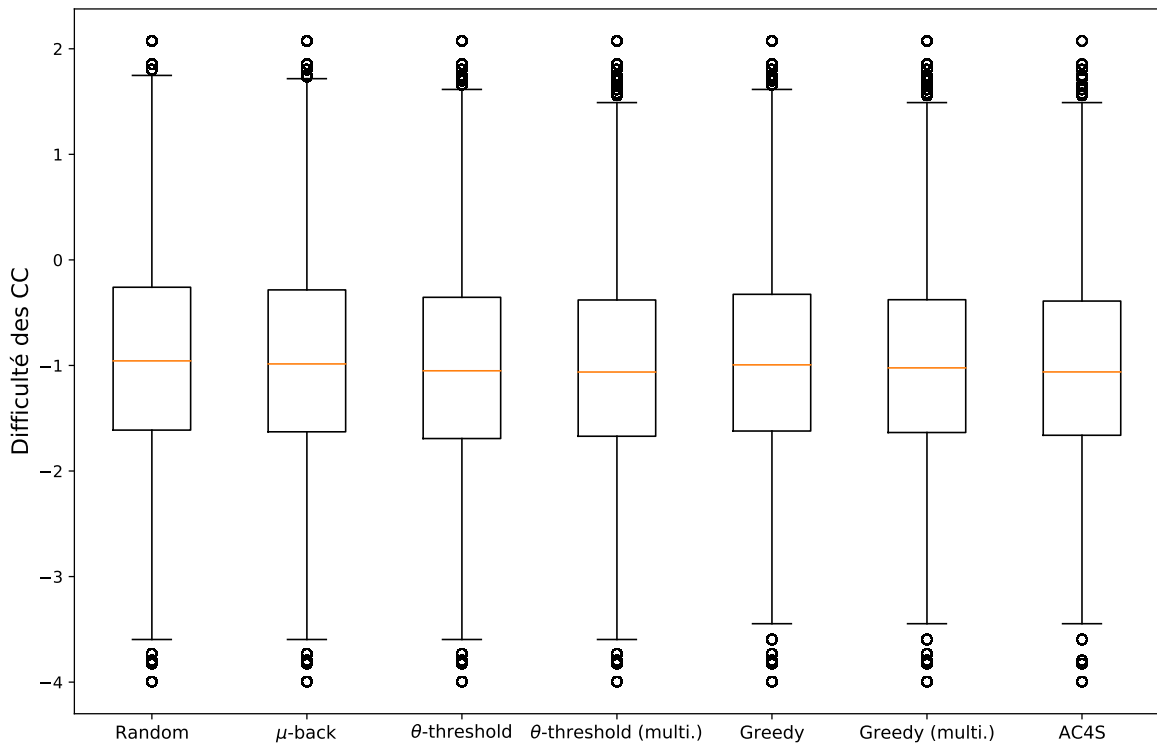


FIGURE 5.6 – Comparaison des distributions des difficultés des CC sélectionnées par les heuristiques et par AC4S, toutes simulations confondues.

5.5 Conclusion

Dans ce chapitre, nous avons proposé AC4S, un nouvel algorithme pour l'optimisation de l'espacement des révisions d'un ensemble de CC. Contrairement aux heuristiques présentées au chapitre 4, cet algorithme est fondé sur une architecture d'apprentissage par renforcement profond et apprend au fil des interactions avec les apprenants la politique optimale de sélection d'item pour maximiser la Probabilité Moyenne de Réussite des apprenants pendant la période de rétention. Il ne s'appuie donc *pas*, au contraire de θ -*threshold* par exemple, sur des connaissances établies en psychologie cognitive concernant le fonctionnement de l'apprentissage humain.

Notre algorithme AC4S utilise une représentation dense et abstraite de l'état des connaissances de l'apprenant pour faire sa recommandation. Cette représentation provient de la couche cachée d'une version alternative du modèle DKT (PIECH et al., 2015), entraînée au préalable sur des trajectoires d'étudiants. La comparaison de cet algorithme à nos heuristiques nous permet, entre autres, d'évaluer l'impact du décalage de modèle (*model mismatch* en anglais) sur ses performances.

Nous comparons cet algorithme d'apprentissage par renforcement profond aux heuristiques développées au chapitre 4, avec le même protocole de simulation que nous avons décrit à la section 4.2. Nous montrons alors que notre algorithme parvient à obtenir des performances supérieures à la plupart des heuristiques proposées à la section 4.3, malgré des observations provenant d'un modèle qui n'est pas le modèle de l'apprenant utilisé pour générer les trajectoires. Seule *Greedy* (multi.) obtient de meilleurs résultats que notre algorithme mais elle a accès au véritable modèle de l'apprenant et ne souffre donc pas du décalage de modèle.

Ces résultats suggèrent qu'une approche qui ignore les théories de l'effet d'espacement et de l'effet de test est à même d'obtenir, au moins sur des données simulées, de meilleures performances que des heuristiques conçues à priori et qui tiennent compte, elles, de ces apports de la psychologie cognitive. Néanmoins, ceci ne doit en aucun cas être interprété comme une preuve de l'inefficacité des méthodes hybrides (DOROUDI, ALEVEN et al., 2019). En premier lieu, c'est la théorie en psychologie cognitive qui guide notre modélisation de l'oubli de l'apprenant (CHOFFIN, POPINEAU et al., 2019), ainsi que l'objectif même de l'algorithme (l'espacement des révisions). Mais surtout, si notre algorithme est capable de parvenir à une politique de recommandation performante, ceci se fait au prix d'un grand nombre d'épisodes. Une approche hybride permettrait par exemple de guider les débuts de l'apprentissage de notre algorithme pour converger plus rapidement vers une politique efficace.

Chapitre 6

Conclusion

6.1 Résumé des contributions de la thèse

L’espacement de l’apprentissage est une stratégie bien connue dans la littérature en psychologie cognitive (N. J. CEPEDA, PASHLER et al., 2006) pour améliorer la rétention en mémoire d’un ensemble d’informations. Bien que les connaissances utilisées dans les expériences sur l’espacement relèvent souvent de la mémorisation pure, plusieurs expériences ont montré que cet effet s’appliquait également à la mémorisation de connaissances plus complexes, telles que des concepts en science ou l’application correcte de règles grammaticales.

Parallèlement, des travaux de recherche récents en fouille de données éducatives ont montré qu’il était possible de développer des algorithmes d’espacement adaptatif et personnalisé pour optimiser la mémorisation pure d’éléments simples, représentés souvent par des *flashcards*. Ces travaux apportent une solution quantitative et automatique au problème suivant : s’il est vrai que l’espacement des révisions bénéficie à la rétention en mémoire, quand faut-il planifier les révisions successives d’une même information ?

Dans cette thèse, nous avons donc cherché à étendre les algorithmes d’espacement adaptatif et personnalisé de l’apprentissage de *flashcards* à la mémorisation de CC. Nous avons développé pour cela un cadre de recherche pour cette problématique, dans lequel les apprenants ont accès à un ensemble d’items de révision pour pratiquer de manière périodique un ensemble de CC. Un item de révision peut mettre en jeu plusieurs CC en même temps. Nous supposons que le fait de pratiquer un item influe sur la probabilité de bonne réponse de toutes les CC qu’il met en jeu. Si les CC sont bien indépendantes les unes des autres, les items ne le sont donc pas. Pour diminuer la complexité de la tâche de planification, nous restreignons notre problématique au choix de l’item optimal à faire pratiquer un apprenant à un instant t : les moments des séances de révisions et le budget total d’items sont donc exogènes. Nous définissons ici l’optimalité en termes de maîtrise moyenne des CC du programme sur une période future donnée, la période de rétention.

Notre première contribution a consisté en le développement et l’étude d’un nouveau modèle de l’apprenant, que nous avons baptisé DAS3H. Ce modèle s’inscrit à la frontière entre la modélisation statistique de l’apprenant et la modélisation cognitive de la mémoire, héritée de la littérature en psychologie cognitive. DAS3H modélise l’apprentissage et l’oubli d’un apprenant sur un ensemble de CC en intégrant dans sa structure les relations entre items et CC. Il permet en outre à chaque CC d’avoir une courbe d’apprentissage et d’oubli qui lui est propre. Nous avons montré au chapitre 3 que DAS3H parvenait à obtenir de meilleures performances prédictives que quatre autres modèles traditionnels de l’apprenant et ce, sur cinq jeux de données éducatives différents. Nous avons également publié le code Python pour reproduire ces résultats en accès libre sur GitHub¹. Nos résultats suggèrent notamment que son module en charge de la modélisation de la mémoire joue un rôle central dans ses performances. Encouragés par ces résultats, nous avons ensuite utilisé ce modèle dans deux

1. <https://github.com/BenoitChoffin/das3h>

objectifs différents : la simulation de trajectoires synthétiques d'apprentissage et d'oubli sur un ensemble de CC et la sélection de CC pour la révision.

Pour s'attaquer au problème de l'espacement adaptatif et personnalisé de CC, nous avons ensuite, au chapitre 4, adapté à notre problématique trois heuristiques de la littérature sur les algorithmes d'espacement : μ -back, θ -threshold et Greedy. Nous avons exposé les différences qualitatives de ces algorithmes, notamment en termes de complexité, d'hypothèses et de soubassements théoriques. Pour tirer parti du fait qu'un item peut engager plusieurs CC à la fois, nous avons aussi proposé une nouvelle procédure gloutonne de sélection du sous-ensemble de CC le plus prometteur parmi les combinaisons disponibles dans la q-matrice à faire réviser à un instant t . Nous avons donc proposé une version multi-CC des heuristiques θ -threshold et Greedy, μ -back n'étant pas compatible avec cette procédure. Afin de comparer les performances respectives de ces heuristiques, nous avons développé et implémenté un cadre de simulation réaliste fondé sur notre modèle DAS3H. Nous avons aussi publié le code Python de ces expériences sur GitHub². Plusieurs résultats centraux ressortent de ces expériences :

- La stratégie μ -back constitue une stratégie d'espacement non-adaptative mais efficace quand elle est bien paramétrée.
- L'heuristique θ -threshold se démarque par ses performances supérieures pendant la période d'apprentissage et elle est relativement aisée à paramétrer.
- Greedy ne surpasse que marginalement les performances de mémorisation de μ -back et θ -threshold mais a l'avantage de ne pas nécessiter d'ajustement de paramètre. Elle s'est en outre avérée plus robuste à des comportements d'apprentissage et d'oubli atypiques.
- On peut améliorer significativement les performances de Greedy et θ -threshold par l'utilisation de la procédure gloutonne de sélection de sous-ensemble de CC.

Une approche différente que nous avons souhaité explorer pour l'espacement adaptatif de la révision de CC réside dans l'apprentissage par renforcement. Cette dernière diffère des heuristiques développées au chapitre 4 à plusieurs égards :

- Elle offre un cadre mathématique dédié à la prise de décision séquentielle en milieu incertain.
- Ce cadre est suffisamment modulaire pour s'adapter par exemple à différents objectifs.
- La politique optimale de sélection d'item est apprise en interagissant avec les apprenants, et s'améliore donc au fil du temps.

Nous avons proposé au chapitre 5 AC4S, un nouvel algorithme d'espacement adaptatif de l'apprentissage de CC, fondé sur une méthode dite *Actor-critic*. Cet algorithme a la particularité de ne pas utiliser directement les observations de son environnement, mais de passer par une représentation abstraite de l'état de connaissances de l'apprenant à un instant t : cet état de connaissances est généré par un modèle différent du modèle simulateur de trajectoires. Nous réemployons le même cadre d'expérience³ qu'au chapitre 4 et montrons que notre algorithme d'apprentissage par renforcement parvient à apprendre une politique de sélection d'items meilleure que l'ensemble des heuristiques de sélection de CC unique mais que (1) sa vitesse de convergence est faible et (2) qu'il sélectionne un nombre moyen supérieur de CC par épisode. Ces résultats suggèrent que l'apprentissage par renforcement est une piste prometteuse pour l'espacement adaptatif des révisions de CC mais qu'elle nécessiterait d'être améliorée.

6.2 Cas d'usage des algorithmes développés dans cette thèse

Dans cette sous-section, nous décrivons deux cas d'usage que nous imaginons pour la mise en œuvre des algorithmes que nous avons développés dans cette thèse.

2. https://github.com/BenoitChoffin/multiskill_adaptive_spacing

3. <https://github.com/BenoitChoffin/ac4s>

6.2.1 En salle de classe

Le premier cas d'usage auquel nous pensons directement concerne une utilisation par un enseignant, dans une salle de classe. Ceci correspond directement au cadre de simulation que nous avons développé et présenté au chapitre 4. En outre, ce cas d'usage nous semble le plus évident à cause de la nécessité de maintenir un ensemble structuré et cohérent d'items et de CC, ce qui est plus facile pour un enseignant dans son cours.

On pourrait imaginer dans ce cadre que l'enseignant souhaite fournir à ses élèves un ensemble d'items de révisions, toutes les semaines. Chaque élève bénéficierait d'une feuille d'items personnalisée, en fonction de ses besoins. Le plugin Moodle PLUME, que nous avons présenté dans (CHOFFIN, 2018), serait tout indiqué pour cet usage. Ce plugin autorise la collecte des données nécessaires à ce type de recommandation sur Moodle et fournit une solution au passage à l'échelle de la génération de feuilles d'exercices individualisées⁴.

L'automatisation repose quant à elle sur la prise en charge par le plugin du suivi individualisé des items donnés à chaque élève ainsi que des résultats de ces derniers, ce qui évite à l'enseignant une gestion fastidieuse et sujette à erreurs. Ce plugin tient en outre registre de l'évolution de la maîtrise des élèves dans chacune des CC du programme. L'enseignant doit d'abord initialiser le plugin en lui fournissant un fichier qui indexe les items par CC et qui spécifie également les dépendances entre CC. L'énoncé des items doit ensuite être enregistré sur le plugin⁵. Si l'enseignant décide en cours d'année de modifier les CC ou les items, il lui suffit simplement d'enregistrer le nouveau fichier sur le plugin et la mise à jour se fait automatiquement. Chaque semaine, l'enseignant sélectionne les CC qui seront traitées. Une feuille d'exercices personnalisée est alors générée et envoyée à chaque élève sur la plateforme au format PDF.

Pour l'instant, la personnalisation se fait automatiquement par tirage aléatoire uniforme parmi les items impliquant chacune des CC indiquées par l'enseignant mais que l'élève ne maîtrise pas encore. De plus, le programme veille à ce qu'un item déjà travaillé par l'élève ne lui soit pas reproposé par la suite. Toutefois, si une feuille générée ne convient pas à l'enseignant, il peut la visualiser et modifier les items au cas par cas, avant de renvoyer la feuille à l'élève. Ainsi, l'enseignant n'a ni à gérer le suivi des CC validées par les élèves, ni à choisir les items envoyés chaque semaine. En outre, le plugin assure un suivi longitudinal des items envoyés à chacun. Une fois que les élèves ont rendu leurs exercices, l'enseignant peut les corriger et indiquer directement sur le plugin, par le biais d'une interface dédiée, si les CC associées sont acquises, en cours d'acquisition, ou non acquises.

6.2.2 En autodidacte

Une autre possibilité d'usage de nos algorithmes serait une utilisation solitaire, en autodidacte. Le support pour cette utilisation pourrait par exemple être un exerciceur (BOURDA et al., 2018), sur lequel les apprenants s'entraîneraient sur une matière donnée, comme la physique. Un expert du domaine (comme un enseignant) mettrait en ligne un ensemble d'items indexés par un ensemble de CC. Dans cette situation, l'apprenant déciderait du moment où il souhaite s'entraîner, et le système lui proposerait les meilleurs items en fonction de ses résultats passés sur les autres items.

4. Il est à noter que cette individualisation se fait bien au niveau des feuilles d'exercices, et pas au niveau des exercices en eux-mêmes. Le plugin pioche dans une base d'exercices aux énoncés fixes pour composer ensuite les feuilles d'exercices personnalisées.

5. Les formats acceptés sont L^AT_EX, PDF ou Word.

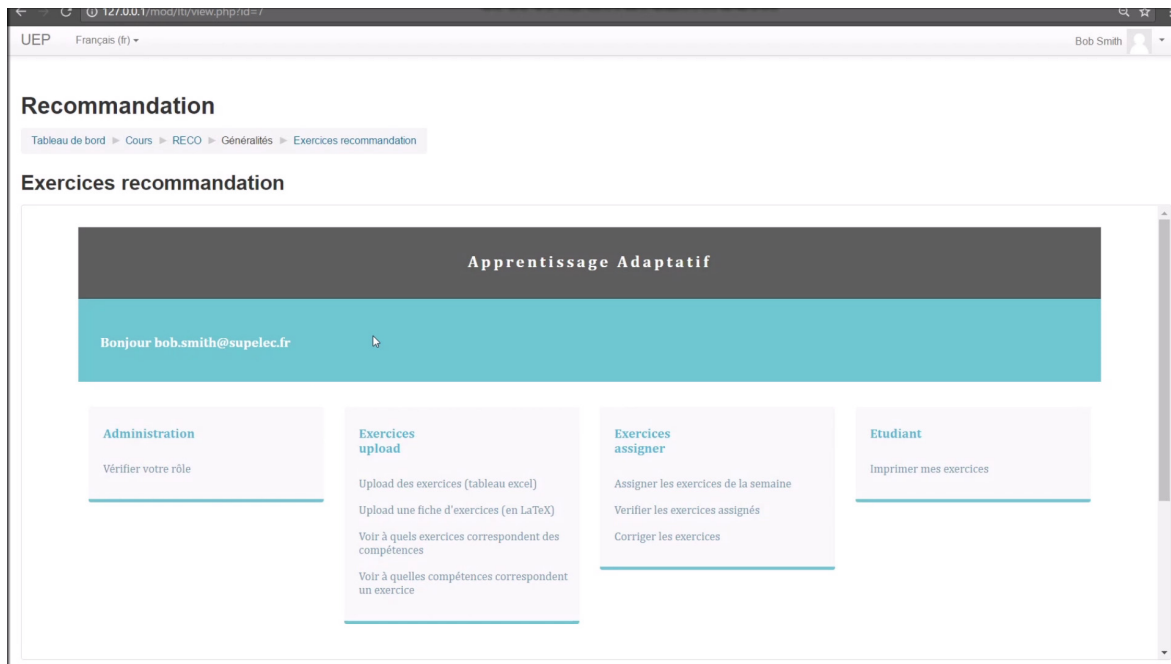


FIGURE 6.1 – Capture d’écran du plugin Moodle PLUME (CHOFFIN, 2018), servant à générer des feuilles d’exercices personnalisées

6.3 Perspectives

6.3.1 Complexifier les relations entre items et CC dans DAS3H

Dans notre modèle DAS3H, nous avons supposé que les relations entre items et CC étaient de nature binaire : soit un item met en jeu une CC, soit il ne la met pas en jeu. Or, dans la réalité, un item peut impliquer des CC à divers degrés. Il serait possible de spécifier des pondérations $\rho_{j,k} \in [0, 1]$ (avec $\forall j \in \llbracket 1, J \rrbracket, \sum_{k=1}^K \rho_{j,k} = 1$) à appliquer aux différents paramètres (β_k et $\theta_{k,w}$) de chaque CC k mise en jeu par un item j pour représenter les liens plus ou moins forts entre un item et les CC qu’il met en jeu.

Cependant, à notre connaissance, peu de jeux de données éducatives fournissent une représentation des relations item-CC qui ne soit pas binaire. Il faudrait donc récolter de tels jeux de données ou bien procéder par exemple à une extraction automatique de q-matrice.

6.3.2 Relâcher l’hypothèse de compensation par la pratique dans DAS3H

La forme additive de notre modèle DAS3H le rend compensatoire. Cela signifie que si un item j met en jeu deux CC k_1 et k_2 , un apprenant pourrait compenser un faible niveau dans k_1 en accroissant indéfiniment sa pratique dans k_2 . Dans des situations réelles, une telle hypothèse peut montrer ses limites : en effet, il serait plus raisonnable de supposer qu’un apprenant doit maîtriser k_1 et k_2 . S’inspirer d’autres modèles non-affines, comme Dealbreaker (A. LAN et al., 2016) ou bien le modèle DINA (DE LA TORRE, 2009) pourrait s’avérer pertinent pour améliorer DAS3H dans cette direction.

6.3.3 Utiliser d’autres sources d’information pour la modélisation de l’apprenant

Dans notre modélisation de l’apprentissage et de l’oubli par le modèle DAS3H, nous ne tenons compte que de sources d’information très faibles : le moment auquel un apprenant a résolu un item, et s’il l’a résolu correctement. Nous pourrions profiter de sources d’informations plus denses, qui sont par exemple fournies dans des jeux de données comme ASSISTments (FENG et al., 2009) pour améliorer la puissance prédictive de DAS3H.

Le temps de réponse d'un apprenant à un item a par exemple été utilisé par METTLER et al. (2016) pour servir d'indicateur de la difficulté de rappel et donc, indirectement, de l'ancrage en mémoire de l'item (ou de la CC, dans notre cas). Plus le temps de réponse est long, et si l'apprenant répond correctement à l'item, alors plus l'ancrage en mémoire de cet item est faible. Il serait intéressant de tester si l'incorporation de cette information dans DAS3H pourrait améliorer la performance du modèle.

Nous pourrions aussi intégrer dans DAS3H une représentation des interactions passées qui tienne compte d'une plus grande diversité d'interactions que les simples réponses aux items (CHI et al., 2011) : par exemple, si l'apprenant a visionné une vidéo, lu un contenu pédagogique,... Enfin, nous pourrions également tenir compte des bonnes réponses partielles : nous avons considéré dans nos travaux qu'une réponse ne pouvait être que juste ou fausse.

6.3.4 Comparer les performances de DAS3H à celles d'autres modèles de l'apprenant et sur d'autres jeux de données

Des travaux futurs pourraient aussi comparer DAS3H à des modèles d'apprenant supplémentaires. Par exemple, R-PFA (*Recent-Performance Factor Analysis*) (GALYARDT et al., 2015) et PFA-decay (GONG et al., 2011) étendent et améliorent PFA grâce à l'utilisation d'une représentation de la pratique passée de l'apprenant qui met plus de poids sur les interactions les plus récentes. Cependant, ils ne prennent pas explicitement en compte la distribution temporelle de la pratique passée pour prédire la performance future d'un apprenant. Des modèles plus profonds comme DKT (PIECH et al., 2015) ou des modèles de la mémoire humaine, comme ACT-R (PAVLIK et ANDERSON, 2008) ou MCM (PASHLER, N. CEPEDA et al., 2009), pourraient aussi être comparés à DAS3H.

Nous avons testé dans le chapitre 3 notre modèle sur cinq jeux de données éducatives réelles collectées sur des plateformes de tutorat intelligent pour l'apprentissage des mathématiques et de la programmation. Pour prolonger ces travaux, il serait aussi envisageable de tester notre modèle DAS3H sur des jeux de données concernant des domaines de connaissance plus larges et divers que les mathématiques et la programmation. De tels jeux de données existent mais contiennent rarement des informations sur les moments de chaque interaction *et* sur les relations entre items et CC : par exemple, le jeu de données Slepemapy (ΠΑΡΟΥΣΗΚ et al., 2016) sur des données d'apprentissage de pays (géographie) ne contient pas de données sur les CC. Ils sont également rarement publics (comme le jeu de données de la Khan Academy utilisé par PIECH et al. (2015)).

6.3.5 Optimiser le jeu de fenêtres temporelles de DAS3H

Une autre manière de continuer ces travaux serait aussi de développer une méthode pour trouver les fenêtres temporelles optimales pour la prédiction dans le modèle DAS3H. Nous avons suivi LINDSEY et al. en choisissant cinq fenêtres temporelles : $\{1/24, 1, 7, 30, +\infty\}$ (en jours). Il faudrait valider empiriquement que les fenêtres temporelles que l'on a fixées sont bien celles qu'il nous faut. Des travaux futurs pourraient explorer l'utilisation de fenêtres temporelles alternatives, par exemple avec des échelles temporelles plus fines.

Nous pourrions aussi imaginer fixer un nombre de fenêtres temporelles ainsi que certaines contraintes (par exemple, de croissance exponentielle) pour apprendre le jeu de fenêtres optimal pour la prédiction. LINDSEY et al. ont développé une version alternative de DASH qu'ils ont nommée DASH [MCM] et qui ne fixe pas de limites aux fenêtres temporelles de h_θ : à la place, les compteurs de réussites $c_{s,j,w}$ et de tentatives $a_{s,j,w}$ décroissent de manière exponentielle à un rythme différent pour chaque fenêtre temporelle. Nous pourrions nous inspirer de ce modèle alternatif pour étendre le nôtre.

6.3.6 Améliorer notre algorithme d'apprentissage par renforcement

Il serait également possible d'améliorer l'algorithme d'apprentissage par renforcement profond que nous utilisons au chapitre 5. Par exemple, pour améliorer la vitesse de convergence de leur planificateur adaptatif de révisions fondé sur un algorithme d'apprentissage par renforcement, YANG et al. (2020) utilisent des récompenses denses à chaque action de leur algorithme. Plus précisément, à chaque action a_t , leur algorithme reçoit une pseudo-récompense qui est la différence entre la moyenne des probabilités de bonne réponse sur tous les items à cette interaction et la même moyenne à l'interaction précédente. Ces probabilités sont des estimations fournies par un modèle interne à leur algorithme. Nous pourrions quant à nous utiliser DKT pour cela. Autrement, nous pourrions aussi utiliser comme récompense immédiate le taux d'oubli empirique observé sur les CC pendant l'apprentissage de l'apprenant.

Plus simplement, nous n'avons pas cherché dans le chapitre 5 à optimiser les hyperparamètres de nos politiques ou de DKT : le faire pourrait permettre d'améliorer les performances des politiques apprises.

6.3.7 Démarrage à froid des algorithmes d'espace adaptatif

Pour une utilisation en situation réelle, il est nécessaire que les algorithmes d'espace adaptatif soient faciles d'utilisation. Notamment, le fait de requérir un modèle d'apprentissage et d'oubli entraîné au préalable sur des données d'apprenants constitue un frein certain⁶ à l'utilisation des algorithmes θ -threshold et Greedy, tout comme l'est la nécessité d'apprendre la politique après un grand nombre d'interactions pour nos algorithmes d'apprentissage par renforcement. Il serait donc nécessaire d'aborder la problématique du démarrage à froid pour les algorithmes que nous avons développés.

Heuristiques θ -threshold et Greedy

Une première solution simple consisterait, comme le font HUNZIKER et al. (2019), à fixer a priori les valeurs $\theta_{k,w}$ des coefficients des fenêtres temporelles des courbes d'apprentissage et d'oubli, en fonction de la difficulté des CC. HUNZIKER et al. montrent que leur algorithme Greedy (dont nous avons proposé une extension au chapitre 4) parvient tout de même à obtenir de meilleures performances que trois stratégies de sélection de référence. Malgré tout, cette approche nous semble seulement convenir dans des situations d'absence totale de données, car elle ne repose pas sur une véritable estimation des paramètres du modèle.

Une autre solution, si l'on choisit de nous reposer sur notre modèle DAS3H, serait d'utiliser sa variante que nous avons baptisée DAS3H_{1p}. Cette version de DAS3H n'estime qu'un seul jeu de paramètres θ_w décrivant les courbes d'apprentissage et d'oubli des CC : ces paramètres sont donc partagés entre toutes les CC. Comme pour DASH (LINDSEY et al., 2014), les données du début du programme pourraient servir à inférer les comportements d'oubli sur les CC ultérieures, sans attendre de récolter des données sur ces dernières.

Apprentissage par renforcement

En ce qui concerne notre algorithme d'apprentissage par renforcement, il serait possible de le pré-entraîner en utilisant des simulations à partir d'un modèle d'apprenant estimé sur des données réelles (UPADHYAY et al., 2018; YANG et al., 2020). Ces données réelles doivent bien entendu correspondre aux items et aux CC du programme sur lequel on souhaite faire réviser nos apprenants.

Une autre piste d'amélioration consisterait à rendre cet algorithme *transférable* à des programmes et des ensembles d'items et de CC qu'il n'aurait jamais vus auparavant. En

6. Sauf pour des cours qui sont redonnés à plusieurs reprises, ou bien des MOOC.

situation réelle, même avec un entraînement au préalable sur des données simulées, il est possible que la convergence prenne du temps. Dans ces conditions, est-il possible de tirer parti de ce que l'algorithme aurait appris sur d'autres programmes d'apprentissage et de révision? Nous pourrions utiliser pour cela du méta-apprentissage par renforcement (RAKELLY et al., 2019).

6.3.8 Adaptation de la difficulté et espacement de l'apprentissage

L'adaptation de la difficulté dans les ITS constitue un champ d'investigation prometteur de la recherche en intelligence artificielle pour l'éducation. En effet, l'adaptation de la difficulté dans les ITS permet d'améliorer les performances des apprenants (SAMPAYO-VARGAS et al., 2013), de maintenir cette amélioration plus longtemps après la fin de l'entraînement (CHOI et al., 2008) et de généraliser cette amélioration à d'autres tâches (AHISSAR et al., 1997).

Par ailleurs, des travaux en psychologie cognitive (BJORK, 1994) ont mis en évidence l'appartenance de l'espacement de l'apprentissage au champ des *difficultés désirables*. Une difficulté désirable est une difficulté qui bénéficie à l'apprentissage et à la mémorisation des apprenants. Dans ce cadre théorique, l'espacement crée une difficulté désirable en forçant l'apprenant à faire un effort pour récupérer en mémoire l'information qu'il avait apprise précédemment.

Nous émettons l'hypothèse que la difficulté d'un item, par exemple définie par le paramètre δ_j dans le modèle DAS3H, peut capturer elle aussi une forme de difficulté désirable. Nous souhaiterions alors savoir s'il serait possible de compenser une difficulté faible d'item par un espacement plus long : en d'autres termes, si faire réviser une CC par un item facile après un long délai est équivalent à faire réviser une CC par un item difficile mais après un délai plus court. Peut-être y a-t-il même une interaction entre la difficulté et l'espacement, de sorte que moduler conjointement difficulté et espacement de l'apprentissage permettrait d'améliorer les bénéfices isolés de chacune des stratégies?

6.3.9 Développer des métriques de mémorisation plus pertinentes

Comme nous l'avons mentionné au chapitre 4, notre métrique PMR_R pondère de manière uniforme les différents instants de la période de rétention et les différentes CC. Il serait tout à fait possible d'assigner un poids différent à différentes périodes ou à différentes CC dans cette métrique, par exemple pour favoriser un apprentissage sur le plus long terme ou pour mettre l'accent sur des CC plus importantes du programme.

En outre, la PMR_R est compensatoire entre les instants et entre les CC. Comme pour DAS3H qui permet à la pratique d'une CC de compenser l'absence de pratique sur une autre CC, il est théoriquement possible qu'une courbe d'oubli qui commence à une probabilité élevée de bonne réponse sur toutes les CC du programme mais qui diminue ensuite rapidement obtienne la même PMR_R qu'une autre courbe d'oubli qui commencerait à un niveau plus faible de probabilité de bonne réponse mais posséderait une courbe d'oubli moins pentue par la suite. Une propriété désirable des courbes d'oubli pourrait être une faible vitesse de décroissance de la probabilité de bonne réponse sur l'ensemble des CC. Pour capturer cette propriété, il faudrait développer d'autres métriques que la PMR .

Enfin, il serait utile de développer des métriques de performance qui prennent directement en compte la *complexité* d'un item, c'est-à-dire le nombre de CC qu'il met en jeu. Cette métrique pourrait ainsi être utilisée au sein de nos algorithmes d'apprentissage par renforcement au lieu de la PMR_R et nous éviterait d'avoir à comparer d'un côté la performance de l'algorithme et de l'autre le nombre de CC qu'il fait réviser au cours d'un épisode.

6.3.10 Améliorer le réalisme de nos simulations

Nous pourrions également prendre en compte d'autres aspects du fonctionnement de la cognition humaine dans nos modèles afin d'améliorer encore le réalisme de nos simulations. Par exemple, il semble que les bénéfices de l'effet de test diminuent avec la complexité d'une tâche à résoudre (VAN GOG et al., 2015). Si nous considérons que le nombre de CC mises en jeu par un item est un indicateur fiable de la complexité de cet item, alors notre modèle simulateur DAS3H ne prend pas explicitement en compte cet effet. De même, sélectionner uniquement des items complexes pourrait décourager les apprenants et les amener à se désengager du système d'espace adaptatif de révisions. Dans ce cas, il serait parfois préférable de choisir des items qui utilisent moins de CC pour l'apprenant. Par exemple, nous pourrions imaginer fixer un nombre maximal de CC à choisir pour la procédure de sélection gloutonne du sous-ensemble de CC le plus prometteur.

Dans nos travaux, nous avons aussi fait l'hypothèse simplificatrice que les items prenaient tous le même temps pour être résolus. Nos algorithmes disposaient donc d'un budget d'items à optimiser. Nous pourrions prendre en compte le temps de résolution des items (PAVLIK et ANDERSON, 2008) et ajuster nos algorithmes pour qu'ils prennent en compte cette donnée et qu'ils tirent parti au maximum d'un budget *temporel* alloué.

Ceci nécessiterait probablement de développer d'autres modèles de l'apprentissage et l'oubli, qui prendraient ces autres dimensions en compte dans la prédiction.

6.3.11 Vers une évaluation plus robuste et plus précise des performances de nos algorithmes d'espace

Diversifier les modèles utilisés dans nos simulations

Dans des travaux futurs, nous pourrions refaire le même type de comparaison de stratégies que dans les chapitres 4 et 5, mais avec des modèles d'apprentissage et d'oubli différents. Des travaux précédents ont montré que des modèles de l'apprenant qui avaient des performances prédictives similaires pouvaient cependant conduire à des séquences d'apprentissage très différentes (ROLLINSON et al., 2015; DOROUDI, ALEVEN et al., 2017) quand ils étaient utilisés pour la planification adaptative de séquences pédagogiques.

Nous avons déjà évalué plusieurs formes de décalage de modèle entre modèle générateur et modèle planificateur aux sous-sections 4.5 et 5.3.2 mais nous aimerions tester des formes plus fortes de décalage en utilisant des modèles plus diversifiés entre simulateur de trajectoires d'apprenants et planificateur d'espace. Ces modèles seraient utilisés dans nos heuristiques *θ -threshold* et *Greedy* ainsi que dans nos algorithmes d'apprentissage par renforcement. Nous pourrions utiliser pour cela la méthode d'Évaluation Matricielle Robuste (ou *Robust Evaluation Matrix*) proposée par DOROUDI, ALEVEN et al. (2017) : elle consiste à évaluer les performances de chaque algorithme de recommandation en utilisant toutes les combinaisons possibles de modèle générateur et modèle planificateur. Tous ces résultats sont ensuite rentrés dans une matrice, qui permet alors de voir la robustesse d'un algorithme de planification au décalage de modèle.

Cependant, à notre connaissance, aucun modèle d'apprenant actuel (hormis le modèle DAS3H) ne permet à la fois d'inférer l'impact de la pratique d'un item sur un ensemble de CC et d'intégrer l'effet d'oubli dans sa structure. Ainsi, un premier pas dans cette direction serait de développer d'autres modèles d'apprentissage et d'oubli des apprenants qui, comme DAS3H (CHOFFIN, POPINEAU et al., 2019), nous permettraient d'inférer la dynamique de la maîtrise sur un ensemble de CC.

Évaluer les politiques d'espace à l'aide de données existantes

Au lieu d'utiliser des trajectoires d'apprenants synthétiques pour comparer nos algorithmes, nous pourrions aussi utiliser des données d'apprentissage existantes.

Par exemple, TABIBIAN et al. (2019) utilisent des données fournies par Duolingo (SETTLES, BRUST et al., 2018) pour comparer les performances de trois algorithmes d’espacement adaptatif de l’apprentissage. Pour ce faire, ils calculent pour chaque paire (apprenant, item) du jeu de données sa proximité avec chacun des trois plannings de révision, en termes de vraisemblance (statistique). Ils sélectionnent enfin pour chaque algorithme les 25 % des trajectoires qui lui sont les plus proches et font des comparaisons agrégées du taux d’oubli empirique sur cette base.

Au lieu de générer aléatoirement des jeux de paramètres avec des distributions de probabilité spécifiées, comme nous le faisons aux chapitres 4 et 5, il serait aussi possible d’estimer les paramètres de DAS3H⁷ sur un jeu de données éducatives réelles. On tirerait également la q-matrice de ce jeu de données. D’autres travaux utilisent cette méthode pour évaluer les performances de différents algorithmes de recommandation en fouille de données éducatives (ROLLINSON et al., 2015; DOROUDI, ALEVEN et al., 2017; YANG et al., 2020). Par exemple, UPADHYAY et al. (2018) utilisent une version plus complexe du modèle de régression demi-vie de SETTLES et MEEDER (2016) et estiment ses paramètres sur des données de Duolingo (SETTLES, BRUST et al., 2018). Ensuite, ils simulent des processus d’apprentissage et de mémorisation à l’aide de ce modèle entraîné et assignent des apprenants simulés à chacun des algorithmes concurrents. Les performances de chacun des algorithmes d’espacement adaptatif qu’ils comparent sont évaluées également à l’aide du modèle qu’ils ont appris sur les données de Duolingo. Nous avons déjà commencé à développer et à implémenter un tel protocole.

6.3.12 Tester les algorithmes développés sur de vrais apprenants

Bien entendu, pour évaluer les performances réelles des algorithmes que nous avons développés dans cette thèse, il serait nécessaire de les comparer avec une expérience contrôlée randomisée sur de vrais apprenants. Les simulations informatiques réalisées aux chapitres 4 et 5 nous ont servi à comparer nos algorithmes dans un cadre idéalisé, tout en contrôlant les comportements d’apprentissage et d’oubli de nos apprenants synthétiques. Elles ont notamment permis de déterminer les sous-ensembles de paramètres qui génèrent les meilleures performances de mémorisation chez l’apprenant, comme M. M. KHAJAH et al. (2014) et LINDSEY et al. (2014). Cependant, une expérience avec de vrais apprenants est la meilleure manière, si le design expérimental est bien conçu, d’obtenir des indications fiables sur la véritable efficacité des méthodes que nous avons développées. Nous serions particulièrement intéressés de savoir si les résultats avec des données simulées sont cohérents avec ceux sur de vrais apprenants.

Il serait envisageable de recruter les sujets de cette expérience (HUNZIKER et al., 2019) sur une plateforme comme Mechanical Turk. Mais à notre sens, une expérience en environnement naturel (avec des élèves du secondaire, par exemple), comme le font CLEMENT et al. ou LINDSEY et al., serait plus réaliste et bien plus adéquate à notre problématique d’apprentissage d’un ensemble de CC. Cependant, il serait probablement plus difficile de trouver un cadre où réaliser ce type d’expérience.

Une telle expérience nécessiterait de nombreuses adaptations par rapport à notre cadre de simulation. Par exemple :

- Pour les heuristiques avec modèle (*θ -threshold* et *Greedy*) qui nécessitent des données d’interactions préalables pour fonctionner, et pour notre algorithme d’apprentissage par renforcement, nous pourrions employer les stratégies décrites dans la sous-section 6.3.7.
- Pour des raisons d’éthique et d’équité entre les élèves, nous pourrions imaginer adopter un design d’expérience « intra-sujets », similaire à celui employé par LINDSEY et al. (2014). Chaque élève serait alors affecté à l’intégralité des stratégies d’espacement : seulement, ces stratégies seraient affectées à la révision de CC différentes selon les élèves.

7. Ou de tout autre modèle que l’on voudrait utiliser.

- Les niveaux de maîtrise sur chaque CC ne seraient bien sûr pas directement accessibles comme dans nos simulations. Il faudrait alors faire pendant la période de rétention différents tests à différents délais pour ensuite agréger et comparer statistiquement les performances des différents algorithmes.

Bibliographie

- ADESOPE, Olusola O, TREVISAN, Dominic A et SUNDARARAJAN, Narayankripa. « Rethinking the use of tests : A meta-analysis of practice testing ». In : *Review of Educational Research* 87.3 (2017), p. 659-701 (cf. p. 5).
- AHISSAR, Merav et HOCHSTEIN, Shaul. « Task difficulty and the specificity of perceptual learning ». In : *Nature* 387.6631 (1997), p. 401 (cf. p. 123).
- ANDERSON, John R, MATESSA, Michael et LEBIERE, Christian. « ACT-R : A theory of higher level cognition and its relation to visual attention ». In : *Human-Computer Interaction* 12.4 (1997), p. 439-462 (cf. p. 20, 21).
- ATKINSON, Richard C. « Optimizing the learning of a second-language vocabulary. » In : *Journal of experimental psychology* 96.1 (1972), p. 124 (cf. p. 25, 27).
- BARZAGAR NAZARI, Katharina et EBERSBACH, Mirjam. « Distributing mathematical practice of third and seventh graders : Applicability of the spacing effect in the classroom ». In : *Applied Cognitive Psychology* 33.2 (2019), p. 288-298 (cf. p. 6, 59).
- BIRD, Steve. « Effects of distributed practice on the acquisition of second language English syntax ». In : *Applied Psycholinguistics* 32.2 (2011), p. 435-452 (cf. p. 59).
- BJORK, Robert A. « Memory and Metamemory Considerations in the Training of Human Beings ». In : *Metacognition : Knowing about knowing* 185 (1994) (cf. p. 26, 68, 123).
- BOURDA, Yolaine, CHAUDET, Claude, CHOFFIN, Benoît, PARMENTIER, Jeanne, POPINEAU, Fabrice et VIE, Jill-Jénn. « Tuteurs intelligents : boucler la boucle ». In : *Journée IA pour l'Éducation, PFIA 2018*. 2018 (cf. p. 119).
- CARPENTER, Shana K, CEPEDA, Nicholas J, ROHRER, Doug, KANG, Sean HK et PASHLER, Harold. « Using spacing to enhance diverse forms of learning : Review of recent research and implications for instruction ». In : *Educational Psychology Review* 24.3 (2012), p. 369-378 (cf. p. 4).
- CARPENTER, Shana K, PASHLER, Harold, WIXTED, John T et VUL, Edward. « The effects of tests on learning and forgetting ». In : *Memory & Cognition* 36.2 (2008), p. 438-448 (cf. p. 5).
- CEN, Hao, KOEDINGER, Kenneth et JUNKER, Brian. « Learning factors analysis—a general method for cognitive model evaluation and improvement ». In : *International Conference on Intelligent Tutoring Systems*. Springer. 2006, p. 164-175 (cf. p. 17, 21).
- « Comparing two IRT models for conjunctive skills ». In : *International Conference on Intelligent Tutoring Systems*. Springer. 2008, p. 796-798 (cf. p. 17).
- CEPEDA, Nicholas J, PASHLER, Harold, VUL, Edward, WIXTED, John T et ROHRER, Doug. « Distributed practice in verbal recall tasks : A review and quantitative synthesis. » In : *Psychological bulletin* 132.3 (2006), p. 354 (cf. p. 117).
- CEPEDA, Nicholas J, VUL, Edward, ROHRER, Doug, WIXTED, John T et PASHLER, Harold. « Spacing effects in learning : A temporal ridge of optimal retention ». In : *Psychological science* 19.11 (2008), p. 1095-1102 (cf. p. 3, 9, 33, 105).
- CHI, Min, KOEDINGER, Kenneth R, GORDON, Geoffrey J, JORDON, Pamela et VANLAHN, Kurt. « Instructional factors analysis : A cognitive model for multiple instructional interventions ». In : *Proceedings of the 4th International Conference on Educational Data Mining (EDM 2011)*. 2011, p. 61-70 (cf. p. 121).

- CHOFFIN, Benoît. « PLUME : un Plugin Moodle pour Générer des Feuilles d'Exercices Personalisées ». In : *Rencontres Jeunes Chercheurs en EIAH 2018*. 2018 (cf. p. 119, 120).
- CHOFFIN, Benoît, LATIMIER, Alice et AHMADI, Niluphar. « Making sense of learner behavioral, cognitive and demographic characteristics to improve learner modeling ». In : *International Congress on Technologies in Education 2019*. 2019 (cf. p. 83).
- CHOFFIN, Benoît, POPINEAU, Fabrice, BOURDA, Yolaine et VIE, Jill-Jênn. « DAS3H : Modeling Student Learning and Forgetting for Optimally Scheduling Distributed Practice of Skills ». In : *Proceedings of the Twelfth International Conference on Educational Data Mining (EDM 2019)*. 2019, p. 29-38 (cf. p. 11, 12, 57, 60, 115, 124).
- CHOI, Younggeun, QI, Feng, GORDON, James et SCHWEIGHOFER, Nicolas. « Performance-based adaptive schedules enhance motor learning ». In : *Journal of motor behavior* 40.4 (2008), p. 273-280 (cf. p. 123).
- CLEMENT, Benjamin, ROY, Didier, OUDEYER, Pierre-Yves et LOPES, Manuel. « Multi-Armed Bandits for Intelligent Tutoring Systems ». In : *Proceedings of the 8th International Conference on Educational Data Mining, EDM 2015*. International Educational Data Mining Society (IEDMS), 2015, p. 21 (cf. p. 32, 125).
- COHEN, Michael S, YAN, Veronica X, HALAMISH, Vered et BJORK, Robert A. « Do students think that difficult or valuable materials should be restudied sooner rather than later ? ». In : *Journal of Experimental Psychology : Learning, Memory, and Cognition* 39.6 (2013), p. 1682 (cf. p. 3).
- CONOVER, William Jay. *Practical nonparametric statistics*. T. 350. John Wiley & Sons, 1998 (cf. p. 75).
- CORBETT, Albert. « Cognitive mastery learning in the ACT programming tutor ». In : *AAAI Tech. Rep. SS-00-01*. 2000 (cf. p. 33).
- CORBETT, Albert T et ANDERSON, John R. « Knowledge tracing : Modeling the acquisition of procedural knowledge ». In : *User modeling and user-adapted interaction* 4.4 (1994), p. 253-278 (cf. p. 16, 33).
- DE LA TORRE, Jimmy. « DINA model and parameter estimation : A didactic ». In : *Journal of educational and behavioral statistics* 34.1 (2009), p. 115-130 (cf. p. 120).
- DEFAZIO, Aaron, BACH, Francis et LACOSTE-JULIEN, Simon. « SAGA : A fast incremental gradient method with support for non-strongly convex composite objectives ». In : *Advances in neural information processing systems*. 2014, p. 1646-1654 (cf. p. 47).
- DEMPSTER, Frank N. « Spacing effects and their implications for theory and practice ». In : *Educational Psychology Review* 1.4 (1989), p. 309-330 (cf. p. 3).
- DESMARAIS, Michel C. et BAKER, Ryan Shaun Joazeiro de. « A review of recent advances in learner and skill modeling in intelligent learning environments ». In : *User Modeling and User-Adapted Interaction* 22.1-2 (2012), p. 9-38 (cf. p. 15, 37).
- DOROUDI, Shayan. « Integrating Human and Machine Intelligence for Enhanced Curriculum Design ». Thèse de doct. Air Force Research Laboratory, 2019 (cf. p. 26).
- « Mastery Learning Heuristics and Their Hidden Models ». In : *International Conference on Artificial Intelligence in Education*. Springer. 2020, p. 86-91 (cf. p. 33).
- DOROUDI, Shayan, ALEVEN, Vincent et BRUNSKILL, Emma. « Robust evaluation matrix : Towards a more principled offline exploration of instructional policies ». In : *Proceedings of the Fourth (2017) ACM Conference on Learning@Scale*. ACM. 2017, p. 3-12 (cf. p. 102, 124, 125).
- « Where's the Reward ? ». In : *International Journal of Artificial Intelligence in Education* 29.4 (2019), p. 568-620 (cf. p. 28, 101, 109, 115).
- DUNLOSKY, John, RAWSON, Katherine A, MARSH, Elizabeth J, NATHAN, Mitchell J et WILLINGHAM, Daniel T. « Improving students' learning with effective learning techniques : Promising directions from cognitive and educational psychology ». In : *Psychological Science in the Public Interest* 14.1 (2013), p. 4-58 (cf. p. 3).
- EBBINGHAUS, Herm. *Ueber das Gedächtnis*. Duncker et Humblot, 1885 (cf. p. 19, 28).

- EFFENBERGER, Tomáš. « Blockly Programming Dataset ». In : *3rd Educational Data Mining in Computer Science Education (CSEDM) Workshop*. 2019 (cf. p. 10, 44).
- EKANADHAM, Chaitanya et KARKLIN, Yan. « T-SKIRT : Online Estimation of Student Proficiency in an Adaptive Learning System ». In : *Machine Learning for Education Workshop at ICML*. 2015 (cf. p. 19, 23).
- FENG, Mingyu, HEFFERNAN, Neil et KOEDINGER, Kenneth. « Addressing the assessment challenge with an online system that tutors as it assesses ». In : *User Modeling and User-Adapted Interaction* 19.3 (2009), p. 243-266 (cf. p. 43, 120).
- FRANÇOIS-LAVET, Vincent, HENDERSON, Peter, ISLAM, Riashat, BELLEMARE, Marc G. et PINEAU, Joelle. « An Introduction to Deep Reinforcement Learning ». In : *Found. Trends Mach. Learn.* 11.3-4 (2018), p. 219-354 (cf. p. 103).
- GALYARDT, April et GOLDIN, Ilya. « Move your lamp post : Recent data reflects learner knowledge better than older data ». In : *Journal of Educational Data Mining* 7.2 (2015), p. 83-108 (cf. p. 19, 121).
- GHOSH, Aritra, HEFFERNAN, Neil T. et LAN, Andrew S. « Context-Aware Attentive Knowledge Tracing ». In : *KDD '20 : The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*. ACM, 2020, p. 2330-2339 (cf. p. 17).
- GONG, Yue, BECK, Joseph E et HEFFERNAN, Neil T. « How to construct more accurate student models : Comparing and optimizing knowledge tracing and performance factor analysis ». In : *International Journal of Artificial Intelligence in Education* 21.1-2 (2011), p. 27-46 (cf. p. 19, 121).
- GONZÁLEZ-BRENES, José, HUANG, Yun et BRUSILOVSKY, Peter. « General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge ». In : *Proceedings of the 7th International Conference on Educational Data Mining*. 2014, p. 84-91 (cf. p. 17).
- GONZÁLEZ-BRENES, José et MOSTOW, Jack. « What and when do students learn ? Fully data-driven joint estimation of cognitive and student models ». In : *Proceedings of the 6th International Conference on Educational Data Mining*. 2013 (cf. p. 23).
- GOUTTE, Cyril, DURAND, Guillaume et LÉGER, Serge. « On the Learning Curve Attrition Bias in Additive Factor Modeling ». In : *International Conference on Artificial Intelligence in Education*. Springer. 2018, p. 109-113 (cf. p. 40, 66, 83).
- HASTIE, Trevor, TIBSHIRANI, Robert et FRIEDMAN, Jerome. *The elements of statistical learning : data mining, inference, and prediction*. Springer Science & Business Media, 2009 (cf. p. 42).
- HOCHREITER, Sepp et SCHMIDHUBER, Jürgen. « Long short-term memory ». In : *Neural computation* 9.8 (1997), p. 1735-1780 (cf. p. 28, 105).
- HUNZIKER, Anette, CHEN, Yuxin, MAC AODHA, Oisín, RODRIGUEZ, Manuel Gomez, KRAUSE, Andreas, PERONA, Pietro, YUE, Yisong et SINGLA, Adish. « Teaching multiple concepts to a forgetful learner ». In : *Advances in Neural Information Processing Systems*. 2019, p. 4050-4060 (cf. p. 25, 27, 31, 32, 35, 65, 66, 68, 69, 84, 122, 125).
- KANG, Sean HK. « Spaced repetition promotes efficient and effective learning : Policy implications for instruction ». In : *Policy Insights from the Behavioral and Brain Sciences* 3.1 (2016), p. 12-19 (cf. p. 3).
- KANG, Sean HK, LINDSEY, Robert V, MOZER, Michael C et PASHLER, Harold. « Retrieval practice over the long term : Should spacing be expanding or equal-interval ? » In : *Psychonomic bulletin & review* 21.6 (2014), p. 1544-1550 (cf. p. 65).
- KHAJAH, Mohammad, LINDSEY, Robert V et MOZER, Michael C. « How deep is knowledge tracing ? » In : *Proceedings of the Ninth International Conference on Educational Data Mining* (2016) (cf. p. 17, 23).
- KHAJAH, Mohammad M, LINDSEY, Robert V et MOZER, Michael C. « Maximizing students' retention via spaced review : Practical guidance from computational models of memory ». In : *Topics in cognitive science* 6.1 (2014), p. 157-169 (cf. p. 25, 26, 31, 59, 61, 63, 66, 68, 73, 86, 88, 94, 125).

- KINGMA, Diederik P et BA, Jimmy. « Adam : A method for stochastic optimization ». In : *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*. 2015 (cf. p. 106, 107).
- KOEDINGER, Kenneth R, CORBETT, Albert T et PERFETTI, Charles. « The Knowledge-Learning-Instruction framework : Bridging the science-practice chasm to enhance robust student learning ». In : *Cognitive science* 36.5 (2012), p. 757-798 (cf. p. 6).
- LAN, Andrew, GOLDSTEIN, Tom, BARANIUK, Richard et STUDER, Christoph. « Dealbreaker : A nonlinear latent variable model for educational data ». In : *International Conference on Machine Learning*. 2016, p. 266-275 (cf. p. 120).
- LAN, Andrew S. et BARANIUK, Richard G. « A Contextual Bandits Framework for Personalized Learning Action Selection ». In : *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016*. 2016, p. 424-429 (cf. p. 32).
- LATIMIER, Alice, PEYRE, Hugo et RAMUS, Franck. « A Meta-Analytic Review of the Benefit of Spacing out Retrieval Practice Episodes on Retention ». In : *Educational Psychology Review* (oct. 2020) (cf. p. 4, 6).
- LEITNER, S. « So lernt man lernen [How to learn] ». In : *Freiburg im Breisgau, Germany : Herder* (1972) (cf. p. 24, 25, 29, 70).
- LINDEN, Wim J van der et HAMBLETON, Ronald K. *Handbook of modern item response theory*. Springer Science & Business Media, 2013 (cf. p. 15).
- LINDSEY, Robert V, SHROYER, Jeffery D, PASHLER, Harold et MOZER, Michael C. « Improving students' long-term knowledge retention through personalized review ». In : *Psychological science* 25.3 (2014), p. 639-647 (cf. p. 3, 5, 9, 11, 17, 21-25, 27, 28, 31, 35, 36, 38, 39, 45, 46, 55, 57, 59, 61, 70, 86, 88, 94, 121, 122, 125).
- MAC AODHA, Oisín, SU, Shihan, CHEN, Yuxin, PERONA, Pietro et YUE, Yisong. « Teaching categories to human learners with visual explanations ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, p. 3820-3828 (cf. p. 33).
- METTLER, Everett, MASSEY, Christine M et KELLMAN, Philip J. « Improving Adaptive Learning Technology through the Use of Response Times. » In : *Grantee Submission* (2011) (cf. p. 25, 29).
- « A comparison of adaptive and fixed schedules of practice. » In : *Journal of Experimental Psychology : General* 145.7 (2016), p. 897 (cf. p. 4, 5, 24, 25, 29, 59, 121).
- METZLER-BADDELEY, Claudia et BADDELEY, Roland J. « Does adaptive training work? » In : *Applied Cognitive Psychology : The Official Journal of the Society for Applied Research in Memory and Cognition* 23.2 (2009), p. 254-266 (cf. p. 24, 25, 30).
- MINN, Sein, YU, Yi, DESMARAIS, Michel C, ZHU, Feida et VIE, Jill-Jènn. « Deep knowledge tracing and dynamic student classification for knowledge tracing ». In : *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2018, p. 1182-1187 (cf. p. 17).
- MNIH, Volodymyr, BADIA, Adria Puigdomenech, MIRZA, Mehdi, GRAVES, Alex, LILICRAP, Timothy, HARLEY, Tim, SILVER, David et KAVUKCUOGLU, Koray. « Asynchronous methods for deep reinforcement learning ». In : *International conference on machine learning*. 2016, p. 1928-1937 (cf. p. 106-108).
- MOZER, Michael C et LINDSEY, Robert V. « Predicting and improving memory retention : Psychological theory matters in the big data era ». In : *Big Data in Cognitive Science*. Psychology Press, 2016, p. 43-73 (cf. p. 3, 20, 24, 25, 27, 28).
- NAGATANI, Koki, ZHANG, Qian, SATO, Masahiro, CHEN, Yan-Ying, CHEN, Francine et OHKUMA, Tomoko. « Augmenting Knowledge Tracing by Considering Forgetting Behavior ». In : *The World Wide Web Conference*. 2019, p. 3101-3107 (cf. p. 23).
- NOVIKOFF, Timothy P, KLEINBERG, Jon M et STROGATZ, Steven H. « Education of a model student ». In : *Proceedings of the National Academy of Sciences* 109.6 (2012), p. 1868-1873 (cf. p. 31).
- PAN, Steven C et RICKARD, Timothy C. « Transfer of test-enhanced learning : Meta-analytic review and synthesis. » In : *Psychological bulletin* 144.7 (2018), p. 710 (cf. p. 5).

- PAPOUŠEK, Jan, PELÁNEK, Radek et STANISLAV, Vít. « Adaptive geography practice data set ». In : *Journal of Learning Analytics* 3.2 (2016), p. 317-321 (cf. p. 121).
- PASHLER, Harold, BAIN, Patrice M, BOTTGE, Brian A, GRAESSER, Arthur, KOEDINGER, Kenneth, MCDANIEL, Mark et METCALFE, Janet. « Organizing Instruction and Study to Improve Student Learning. IES Practice Guide. NCER 2007-2004. » In : *National Center for Education Research* (2007) (cf. p. 3).
- PASHLER, Harold, CEPEDA, Nicholas, LINDSEY, Robert V, VUL, Ed et MOZER, Michael C. « Predicting the optimal spacing of study : A multiscale context model of memory ». In : *Advances in neural information processing systems*. 2009, p. 1321-1329 (cf. p. 20, 21, 26, 121).
- PAVLIK, Philip I et ANDERSON, John R. « Practice and forgetting effects on vocabulary memory : An activation-based model of the spacing effect ». In : *Cognitive Science* 29.4 (2005), p. 559-586 (cf. p. 20).
- « Using a model to compute the optimal schedule of practice. » In : *Journal of Experimental Psychology : Applied* 14.2 (2008), p. 101 (cf. p. 20, 24-26, 62, 121, 124).
- PAVLIK, Philip I, CEN, Hao et KOEDINGER, Kenneth R. « Performance Factors Analysis - A New Alternative to Knowledge Tracing ». In : *Proceedings of the 14th International Conference on Artificial Intelligence in Education, AIED 2009*. 2009, p. 531-538 (cf. p. 18, 21).
- PEDREGOSA, Fabian, VAROQUAUX, Gaël, GRAMFORT, Alexandre, MICHEL, Vincent, THIRION, Bertrand, GRISEL, Olivier, BLONDEL, Mathieu, PRETTENHOFER, Peter, WEISS, Ron, DUBOURG, Vincent et al. « Scikit-learn : Machine learning in Python ». In : *Journal of machine learning research* 12.Oct (2011), p. 2825-2830 (cf. p. 47, 86).
- PIECH, Chris, BASSEN, Jonathan, HUANG, Jonathan, GANGULI, Surya, SAHAMI, Mehran, GUIBAS, Leonidas J et SOHL-DICKSTEIN, Jascha. « Deep knowledge tracing ». In : *Advances in neural information processing systems*. 2015, p. 505-513 (cf. p. 17, 105, 109, 112, 115, 121).
- PIMSLEUR, Paul. « A memory schedule ». In : *The Modern Language Journal* 51.2 (1967), p. 73-75 (cf. p. 3, 24, 31).
- QIU, Yumeng, QI, Yingmei, LU, Hanyuan, PARDOS, Zachary A et HEFFERNAN, Neil T. « Does Time Matter? Modeling the Effect of Time with Bayesian Knowledge Tracing ». In : *Proceedings of the 4th International Conference on Educational Data Mining (EDM 2011)*. 2011, p. 139-148 (cf. p. 23).
- RAAIJMAKERS, Jeroen GW. « Spacing and repetition effects in human memory : Application of the SAM model ». In : *Cognitive Science* 27.3 (2003), p. 431-452 (cf. p. 20).
- RAFFERTY, Anna N, BRUNSKILL, Emma, GRIFFITHS, Thomas L et SHAFTO, Patrick. « Faster teaching by POMDP planning ». In : *International Conference on Artificial Intelligence in Education*. Springer. 2011, p. 280-287 (cf. p. 32).
- RAKELLY, Kate, ZHOU, Aurick, FINN, Chelsea, LEVINE, Sergey et QUILLEN, Deirdre. « Efficient off-policy meta-reinforcement learning via probabilistic context variables ». In : *International conference on machine learning*. 2019, p. 5331-5340 (cf. p. 123).
- RECKASE, Mark D. « Multidimensional item response theory models ». In : *Multidimensional item response theory*. Springer, 2009, p. 79-112 (cf. p. 16).
- REDDY, Siddharth, LABUTOV, Igor, BANERJEE, Siddhartha et JOACHIMS, Thorsten. « Unbounded human learning : Optimal scheduling for spaced repetition ». In : *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM. 2016, p. 1815-1824 (cf. p. 9, 20, 28, 30, 70).
- REDDY, Siddharth, LEVINE, Sergey et DRAGAN, Anca. « Accelerating Human Learning with Deep Reinforcement Learning ». In : *NIPS'17 Workshop : Teaching Machines, Robots, and Humans*. 2017 (cf. p. 5, 9, 25, 28, 63, 64, 84, 86, 103, 104).
- RENDLE, Steffen. « Factorization machines with libfm ». In : *ACM Transactions on Intelligent Systems and Technology (TIST)* 3.3 (2012), p. 57 (cf. p. 37, 41, 46, 47).
- REYNOLDS, James H et GLASER, Robert. « Effects of repetition and spaced review upon retention of a complex learning task. » In : *Journal of Educational Psychology* 55.5 (1964), p. 297 (cf. p. 59).

- RIDGEWAY, Karl, MOZER, Michael C et BOWLES, Anita R. « Forgetting of foreign-language skills : A corpus-based analysis of online tutoring software ». In : *Cognitive science* 41.4 (2017), p. 924-949 (cf. p. 20, 22, 23).
- ROEDIGER III, Henry L et KARPICKE, Jeffrey D. « Test-enhanced learning : Taking memory tests improves long-term retention ». In : *Psychological science* 17.3 (2006), p. 249-255 (cf. p. 5).
— « Intricacies of spaced retrieval : A resolution ». In : *Successful Remembering and Successful Forgetting*. Psychology Press, 2011, p. 41-66 (cf. p. 3).
- ROEDIGER III, Henry L et McDERMOTT, Kathleen B. « Remembering What We Learn ». In : *Cerebrum : the Dana Forum on Brain Science*. T. 2018. Dana Foundation. 2018 (cf. p. 3).
- ROHRER, Doug et TAYLOR, Kelli. « The effects of overlearning and distributed practise on the retention of mathematics knowledge ». In : *Applied Cognitive Psychology : The Official Journal of the Society for Applied Research in Memory and Cognition* 20.9 (2006), p. 1209-1224 (cf. p. 59).
- ROLLINSON, Joseph et BRUNSKILL, Emma. « From Predictive Models to Instructional Policies ». In : *Proceedings of the Eighth International Conference on Educational Data Mining*. 2015 (cf. p. 124, 125).
- ROWLAND, Christopher A. « The effect of testing versus restudy on retention : a meta-analytic review of the testing effect. » In : *Psychological Bulletin* 140.6 (2014), p. 1432 (cf. p. 83).
- RUBIN, David C et WENZEL, Amy E. « One hundred years of forgetting : A quantitative description of retention. » In : *Psychological review* 103.4 (1996), p. 734 (cf. p. 19, 20).
- SAMPAYO-VARGAS, Sandra, COPE, Chris J, HE, Zhen et BYRNE, Graeme J. « The effectiveness of adaptive difficulty adjustments on students' motivation and learning in an educational computer game ». In : *Computers & Education* 69 (2013), p. 452-462 (cf. p. 123).
- SENSE, Florian, BEHRENS, Friederike, MEIJER, Rob R et RIJN, Hedderik van. « An individual's rate of forgetting is stable over time but differs across materials ». In : *Topics in cognitive science* 8.1 (2016), p. 305-321 (cf. p. 4, 25, 26).
- SENSE, Florian, JASTRZEMBSKI, Tiffany S, MOZER, Michael C, KRUSMARK, Michael et RIJN, Hedderik van. « Perspectives on Computational Models of Learning and Forgetting ». In : *International Conference on Cognitive Modeling*. 2019 (cf. p. 21).
- SETTLES, Burr, BRUST, Chris, GUSTAFSON, Erin, HAGIWARA, Masato et MADNANI, Nitin. « Second language acquisition modeling ». In : *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*. 2018, p. 56-65 (cf. p. 18, 125).
- SETTLES, Burr et MEEDER, Brendan. « A trainable spaced repetition model for language learning ». In : *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*. T. 1. 2016, p. 1848-1858 (cf. p. 22, 28, 30, 125).
- SINHA, Sugandh. *Using deep reinforcement learning for personalizing review sessions on e-learning platforms with spaced repetition*. 2019 (cf. p. 28, 104).
- STAMPER, J., NICULESCU-MIZIL, A., RITTER, S., GORDON, G.J. et KOEDINGER, K.R. *Algebra I 2005-2006 and Bridge to Algebra 2006-2007. Development data sets from KDD Cup 2010 Educational Data Mining Challenge*. Find them at <http://psl1cdatashop.web.cmu.edu/KDDCup/downloads.jsp>. 2010 (cf. p. 36, 43).
- SUTTON, Richard S, BARTO, Andrew G et al. *Introduction to reinforcement learning*. T. 135. MIT press Cambridge, 1998 (cf. p. 102, 106).
- TABIBIAN, Behzad, UPADHYAY, Utkarsh, DE, Abir, ZAREZADE, Ali, SCHÖLKOPF, Bernhard et GOMEZ-RODRIGUEZ, Manuel. « Enhancing human learning via spaced repetition optimization ». In : *Proceedings of the National Academy of Sciences* 116.10 (2019), p. 3988-3993 (cf. p. 5, 25, 27, 28, 30, 31, 35, 64, 84, 125).
- TARABAN, Roman, MAKI, William S et RYNEARSON, Kimberly. « Measuring study time distributions : Implications for designing computer-based courses ». In : *Behavior Research Methods, Instruments, & Computers* 31.2 (1999), p. 263-269 (cf. p. 3).

- TATSUOKA, Kikumi K. « Rule space : An approach for dealing with misconceptions based on item response theory ». In : *Journal of educational measurement* 20.4 (1983), p. 345-354 (cf. p. 8, 10, 36, 57, 70).
- UPADHYAY, Utkarsh, DE, Abir et RODRIGUEZ, Manuel Gomez. « Deep reinforcement learning of marked temporal point processes ». In : *Advances in Neural Information Processing Systems*. 2018, p. 3168-3178 (cf. p. 25, 28, 29, 64, 113, 122, 125).
- VAN GOG, Tamara et SWELLER, John. « Not new, but nearly forgotten : the testing effect decreases or even disappears as the complexity of learning materials increases ». In : *Educational Psychology Review* 27.2 (2015), p. 247-264 (cf. p. 124).
- VAN RIJN, Hedderik, MAANEN, Leendert van et WOUDENBERG, Marnix van. « Passing the test : Improving learning gains by balancing spacing and testing effects ». In : *Proceedings of the 9th International Conference of Cognitive Modeling*. T. 2. 2009, p. 7-6 (cf. p. 25, 26).
- VIE, Jill-Jënn et KASHIMA, Hisashi. « Knowledge Tracing Machines : Factorization machines for Knowledge Tracing ». In : *Proceedings of the AAAI Conference on Artificial Intelligence*. T. 33. 2019, p. 750-757 (cf. p. 11, 18, 19, 37, 42, 45-47, 54, 57).
- VLACH, Haley A et SANDHOFER, Catherine M. « Distributing learning over time : The spacing effect in children's acquisition and generalization of science concepts ». In : *Child development* 83.4 (2012), p. 1137-1144 (cf. p. 6, 59).
- WALSH, Matthew M, GLUCK, Kevin A, GUNZELMANN, Glenn, JASTRZEMBSKI, Tiffany et KRUSMARK, Michael. « Evaluating the theoretic adequacy and applied potential of computational models of the spacing effect ». In : *Cognitive science* 42 (2018), p. 644-691 (cf. p. 20).
- WALSH, Matthew M, GLUCK, Kevin A, GUNZELMANN, Glenn, JASTRZEMBSKI, Tiffany, KRUSMARK, Michael, MYUNG, Jay I, PITT, Mark A et ZHOU, Ran. « Mechanisms underlying the spacing effect in learning : A comparison of three computational models. » In : *Journal of Experimental Psychology : General* 147.9 (2018), p. 1325 (cf. p. 20).
- WEINSTEIN, Yana, MADAN, Christopher R et SUMERACKI, Megan A. « Teaching the science of learning ». In : *Cognitive Research : Principles and Implications* 3.1 (2018), p. 2 (cf. p. 3).
- WHITEHILL, Jacob et MOVELLAN, Javier. « Approximately optimal teaching of approximately optimal learners ». In : *IEEE Transactions on Learning Technologies* 11.2 (2018), p. 152-164 (cf. p. 32).
- WICKELGREN, Wayne A. « Single-trace fragility theory of memory dynamics ». In : *Memory & Cognition* 2.4 (1974), p. 775-780 (cf. p. 20).
- WILSON, Kevin H, KARKLIN, Yan, HAN, Bojian et EKANADHAM, Chaitanya. « Back to the basics : Bayesian extensions of IRT outperform neural networks for proficiency estimation ». In : *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016*. 2016, p. 539-544 (cf. p. 16, 45).
- WILSON, Kevin H, XIONG, Xiaolu, KHAJAH, Mohammad, LINDSEY, Robert V, ZHAO, Siyuan, KARKLIN, Yan, VAN INWEGEN, Eric G, HAN, Bojian, EKANADHAM, Chaitanya, BECK, Joseph E et al. « Estimating student proficiency : Deep learning is not the panacea ». In : *Neural Information Processing Systems, Workshop on Machine Learning for Education*. 2016, p. 3 (cf. p. 17, 45).
- WIXTED, John T, CARPENTER, Shana K et al. « The Wickelgren power law and the Ebbinghaus savings function ». In : *Psychological Science* 18.2 (2007), p. 133 (cf. p. 20).
- WOZNIAK, PA et GORZELANCZYK, Edward J. « Optimization of repetition spacing in the practice of learning ». In : *Acta neurobiologiae experimentalis* 54 (1994), p. 59-59 (cf. p. 24, 25, 28, 30, 70).
- YANG, Zhengyu, SHEN, Jian, LIU, Yunfei, YANG, Yang, ZHANG, Weinan et YU, Yong. « TADS : Learning Time-Aware Scheduling Policy with Dyna-Style Planning for Spaced Repetition ». In : *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2020, p. 1917-1920 (cf. p. 25, 28, 63, 65, 86, 103, 122, 125).

- YUDELSON, Michael V, KOEDINGER, Kenneth R et GORDON, Geoffrey J. « Individualized bayesian knowledge tracing models ». In : *International Conference on Artificial Intelligence in Education*. Springer. 2013, p. 171-180 (cf. p. 17).
- ZHANG, Jiani, SHI, Xingjian, KING, Irwin et YEUNG, Dit-Yan. « Dynamic key-value memory networks for knowledge tracing ». In : *Proceedings of the 26th international conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2017, p. 765-774 (cf. p. 17).
- ZHU, Xiaojin. « Machine teaching : An inverse problem to machine learning and an approach toward optimal education ». In : *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 2015 (cf. p. 33).
- ZHU, Xiaojin, SINGLA, Adish, ZILLES, Sandra et RAFFERTY, Anna N. « An overview of machine teaching ». In : *CoRR abs/1801.05927* (2018) (cf. p. 33).

Title : Adaptive spacing algorithms for optimizing long-term mastery of knowledge components

Keywords : Adaptive spacing, review scheduling, knowledge components (KCs), deep reinforcement learning, educational data mining

Abstract : Between acquiring new knowledge and reviewing old knowledge to mitigate forgetting, learners may find it difficult to organize their learning time effectively. Adaptive spacing algorithms, like SuperMemo, can help learners deal with this trade-off. Such algorithms sequentially plan reviews of a given piece of knowledge to adapt to the specific and ongoing needs of each learner. Compared to a fixed and identical temporal spacing between reviews, several experiments have shown that adaptive spacing improves long-term memory retention of the piece of knowledge.

To date, research on adaptive spacing algorithms has focused on the pure memorization of simple pieces of knowledge, which are often represented by flashcards. However, several studies in cognitive psychology have shown that the benefits of spacing out learning episodes on long-term retention also extend to more complex knowledge, such as learning concepts and procedures in mathematics. In this thesis, we have therefore sought to develop adaptive and personalized spacing algorithms for optimizing long-term mastery of knowledge components (KCs).

First, we develop and present a new statistical model of learning and forgetting of knowledge components, coined DAS3H, and we empirically show that DAS3H has better predictive performance than several learner models in educational data mining. Second, we develop several adaptive spacing heuristics for long-term mastery of KCs and compare their performance on simulated data. Two of these heuristics use the DAS3H model to select which KC should be reviewed by a given learner at a given time. In addition, we propose a new greedy procedure to select the most promising subset of KCs instead of the best KC to review. Finally, in the last chapter of this thesis, we develop AC4S, a deep reinforcement learning algorithm for adaptive spacing for KCs. We compare this data-driven approach to the heuristic methods that we presented previously.

Titre : Algorithmes d'espacement adaptatif de l'apprentissage pour l'optimisation de la maîtrise à long terme de composantes de connaissance

Mots-clés : Espacement adaptatif de l'apprentissage, planification de révisions, composantes de connaissance (CC), apprentissage par renforcement profond, fouille de données éducatives

Résumé : Entre acquérir de nouvelles connaissances et revoir les anciennes pour en atténuer l'oubli, les apprenants peuvent avoir du mal à organiser efficacement leur temps d'apprentissage. Les algorithmes d'espacement adaptatif de l'apprentissage, tels SuperMemo, permettent d'aider les apprenants à résoudre cet arbitrage. Ces algorithmes planifient les révisions successives d'une même connaissance de manière optimale et personnalisée en tenant compte des besoins de chaque apprenant. Comparé à un espacement temporel entre les révisions identique pour tous les individus, plusieurs expériences montrent que l'espacement adaptatif maintient un plus haut degré d'ancrage en mémoire à long terme des informations apprises.

Jusqu'ici, la recherche sur l'espacement adaptatif de l'apprentissage s'est concentrée sur la mémorisation pure de connaissances simples, représentées souvent par le biais de *flashcards*. Or, plusieurs études en psychologie cognitive montrent que les bénéfices de l'espacement de l'apprentissage sur la mémorisation à long terme s'étendent aussi à des connaissances plus complexes, telles que l'apprentissage de concepts et de procédures en mathématiques. Dans cette thèse, nous avons donc cherché à développer des algorithmes d'espacement adaptatif et personnalisé de l'apprentissage de composantes de connaissance (CC).

Dans un premier temps, nous proposons un nouveau modèle statistique de l'apprentissage et l'oubli de CC, appelé DAS3H, et montrons empiriquement qu'il possède de meilleures performances prédictives que plusieurs modèles de l'apprenant en fouille de données éducatives. Ensuite, nous développons plusieurs heuristiques d'espacement adaptatif pour la maîtrise à long terme de CC et comparons leurs performances sur des données simulées. Deux de ces heuristiques reposent sur le modèle DAS3H pour sélectionner la CC à faire réviser à un instant donné. Nous proposons en outre une nouvelle procédure gloutonne pour sélectionner le sous-ensemble de CC le plus prometteur au lieu de la meilleure CC à faire réviser. Enfin, dans le dernier chapitre de cette thèse, nous développons AC4S, un algorithme d'apprentissage par renforcement profond pour l'espacement adaptatif de l'apprentissage de CC. Nous comparons cette approche fondée sur les données à nos méthodes heuristiques, présentées précédemment.