Sequential machine learning for intelligent tutoring systems

Julien Seznec

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LILLE
École Doctorale Sciences Pour l'Ingénieur
Spécialité : **Informatique**

Thèse de Doctorat présentée par

Julien SEZNEC

Apprentissage automatique séquentiel pour les systèmes éducatifs intelligents

sous la direction de MM. Michal Valko et Alessandro Lazaric,
et l'encadrement de M. Jonathan Banon.

Rapporteurs :	M.	Aurélien	**GARIVIER**	ENS de Lyon
	M.	Gilles	**STOLTZ**	Université Paris Saclay & CNRS

Soutenue le **15 décembre 2020** devant le jury composé de

M.	Gilles	**STOLTZ**	Univ. Paris Saclay & CNRS	Rapporteur
M.	Aurélien	**GARIVIER**	ENS de Lyon	Rapporteur
M.	Steffen	**GRÜNEWÄLDER**	University of Lancaster	Examinateur
M.	Manuel	**LOPES**	Instituto Superior Tecnico	Examinateur
Mme	Mathilde	**MOUGEOT**	Univ. Paris Saclay & ENSIIE	Présidente du jury
M.	Michal	**VALKO**	INRIA Lille & Deepmind	Directeur
M.	Alessandro	**LAZARIC**	INRIA Lille & FAIR	Co-Directeur
M.	Jonathan	**BANON**	Lelivrescolaire.fr	Encadrant

Centre de Recherche en Informatique, Signal et Automatique de Lille (CRIStAL),
UMR 9189 Équipe SequeL, 59650, Villeneuve d'Ascq, France

lelivrescolaire.fr Université de Lille CRIStAL Centre de Recherche en Informatique, Signal et Automatique de Lille Inria

À mes grands-parents: Denise, Jean et Germaine.

Résumé

Proposer des séquences adaptatives d'exercices dans un Environnement informatique pour l'Apprentissage Humain (EIAH) nécessite de caractériser les lacunes de l'élève et d'utiliser cette caractérisation dans une stratégie pédagogique adaptée. Puisque les élèves ne font que quelques dizaines de questions dans une session de révision, ces deux objectifs sont en compétition. L'apprentissage automatique appelle *problème de bandits* ces dilemmes d'exploration-exploitation dans les prises de décisions séquentielles. Dans cette thèse, nous étudions trois problèmes de bandits pour une application dans les systèmes éducatifs adaptatifs.

Les *bandits décroissants au repos* sont un problème de décision séquentiel dans lequel la récompense associée à une action décroît lorsque celle-ci est sélectionnée. Cela modélise le cas où un élève progresse quand il travaille et l'EIAH cherche à sélectionner le sujet le moins maîtrisé pour combler les plus fortes lacunes. Nous présentons de nouveaux algorithmes et nous montrons que pour un horizon inconnu T et sans aucune connaissance sur la décroissance des K bras, ces algorithmes atteignent une borne de regret dépendante du problème $\mathcal{O}(\log T)$, et une borne indépendante du problème $\widetilde{\mathcal{O}}(\sqrt{KT})$. Nos résultats améliorent substantiellement l'état de l'art, ou seule une borne minimax $\widetilde{\mathcal{O}}(K^{1/3}T^{2/3})$ avait été atteinte. Ces nouvelles bornes sont à des facteurs polylog des bornes optimales sur le problème stationnaire, donc nous concluons : les bandits décroissants ne sont pas plus durs que les bandits stationnaires.

Dans les *bandits décroissants sans repos*, la récompense peut décroître à chaque tour pour toutes les actions. Cela modélise des situations différentes telles que le vieillissement du contenu dans un système de recommandation. On montre que les algorithmes conçus pour le problème "au repos" atteignent les bornes inférieures agnostiques au problème et une borne dépendante du problème $\mathcal{O}(\log T)$. Cette dernière est inatteignable dans le cas général où la récompense peut croître. Nous concluons : l'hypothèse de décroissance simplifie l'apprentissage des bandits sans repos.

Viser le sujet le moins connu peut être intéressant avant un examen, mais pendant le cursus - quand tous les sujets ne sont pas bien compris - cela peut mener à l'échec de l'apprentissage de l'étudiant. On étudie un Processus de Décision Markovien Partiellement Observable (POMDP, selon l'acronyme anglais) dans lequel on cherche à maîtriser le plus de sujets le plus rapidement possible. On montre que sous des hypothèses raisonnables sur l'apprentissage de l'élève, la meilleure stratégie oracle sélectionne le sujet le plus connu sous le seuil de maîtrise. Puisque cet oracle optimal n'a pas besoin de connaître la dynamique de transition du POMDP, nous proposons une stratégie apprenante avec des outils "bandits" classiques, en évitant ainsi les méthodes gourmandes en données de l'apprentissage de POMDP.

Abstract

Designing an adaptive sequence of exercises in Intelligent Tutoring Systems (ITS) requires to characterize the gaps of the student and to use this characterization in a relevant pedagogical strategy. Since a student does no more than a few tens of exercises in a session, these two objectives compete. Machine learning called these exploration-exploitation trade-offs in sequential decision making the *bandits problems*. In this thesis, we study different bandits setups for intelligent tutoring systems.

The *rested rotting bandits* are a sequential decision problem in which the reward associated with an action may decrease when it is selected. It models the situation where the student improves when he works and the ITS aims the least known subject to fill the most important gaps. We design new algorithms and we prove that for an unknown horizon T, and without any knowledge on the decreasing behavior of the K arms, these algorithms achieve problem-dependent regret bound of $\mathcal{O}(\log T)$, and a problem-independent one of $\widetilde{\mathcal{O}}(\sqrt{KT})$. Our result substantially improves over existing algorithms, which suffers minimax regret $\widetilde{\mathcal{O}}(K^{1/3}T^{2/3})$. These bounds are at a polylog factor of the optimal bounds on the classical stationary bandit; hence our conclusion: rotting bandits are not harder than stationary ones.

In the *restless rotting bandits*, the reward may decrease at each round for all the actions. They model different situations such as the obsolescence of content in recommender systems. We show that the rotting algorithms designed for the rested case match the problem-independent lower bounds and a $\mathcal{O}(\log T)$ problem-dependent one. The latter was shown to be unachievable in the general case where rewards can increase. We conclude: the rotting assumption makes the restless bandits easier.

Targeting the least known topic may be interesting before an exam but during the curriculum - when all the subjects are not yet understood - it can lead to failure in the learning of the student. We study a Partially Observable Markov Decision Process in which we aim at mastering as many topics as fast as possible. We show that under relevant assumptions on the learning of the student, the best oracle policy targets the most known topic under the mastery threshold. Since this optimal oracle does not need to know the transition dynamics of the POMDP, we design a learning policy with classical bandits tools, hence avoiding the data-intensive methods of POMDP learning.

Acknowledgments

I deeply thank my PhD advisors, Michal and Alessandro, for everything that I have learned during these four years. They taught me about bandits theory during the numerous brainstorming sessions on the white board, but I have also learned from them how to carry on a scientific project; from selecting the most promising research questions to the writing of the papers.

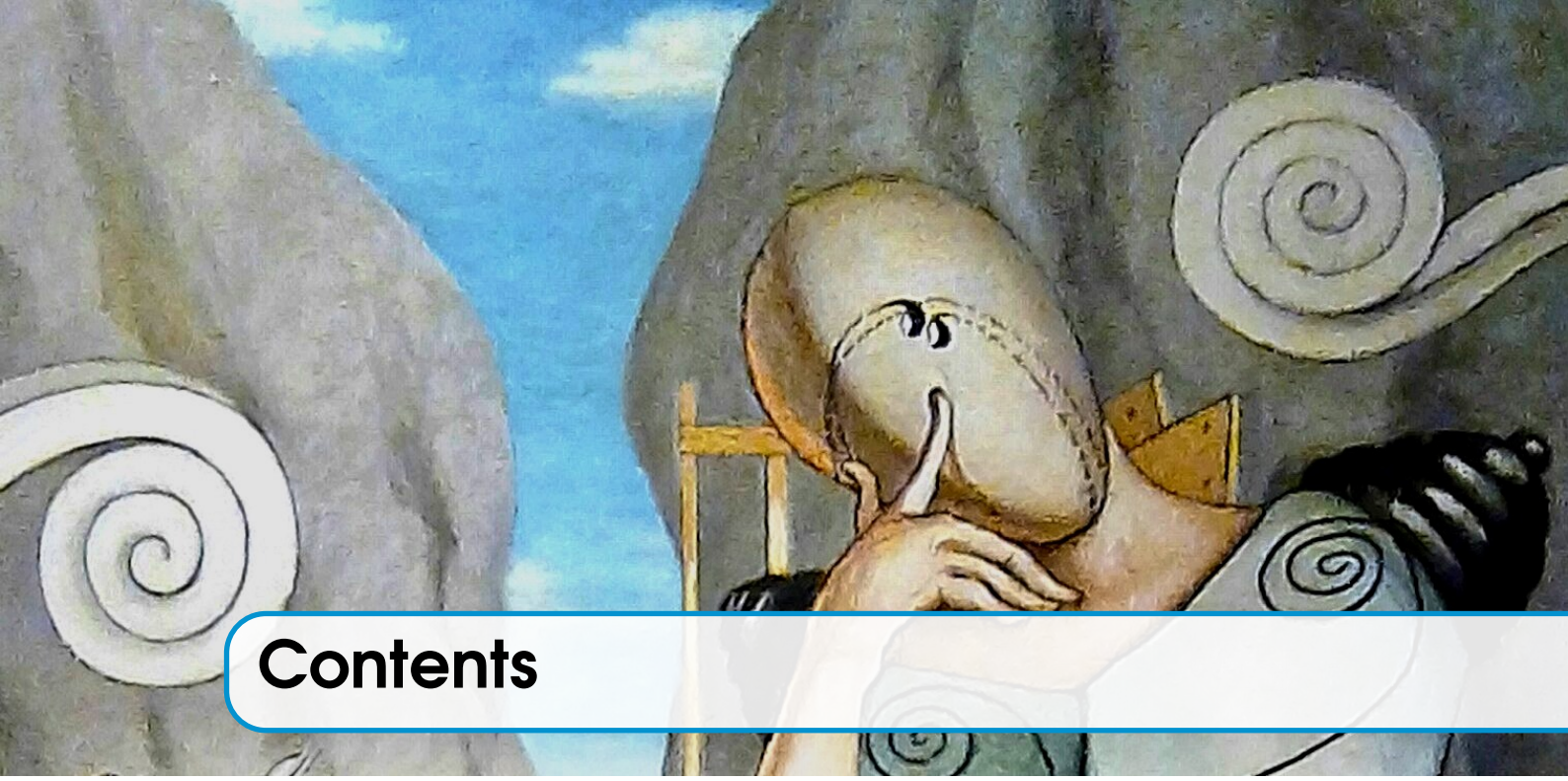
Je remercie Aurélien Garivier et Gilles Stoltz pour leurs relectures du manuscrit. Je n'ignore pas la quantité de travail que cela représente, et je suis sincèrement touché qu'ils aient accepté d'être Rapporteurs. I also would like to thank the examiners - Manuel Lopes and Steffen Grünewälder - for accepting my invitation. Je remercie aussi Mathilde Mougeot pour avoir présidé le jury de soutenance.

Un grand merci à Lelivrescolaire.fr dans son ensemble pour m'avoir permis de réaliser cette thèse sur un sujet qui me portait à coeur, et cela au plus proche des débouchés réels. Merci en premier lieu à Jonathan ainsi qu'à l'ensemble du pôle produit, pour m'avoir accompagné et fait monter en compétence sur de nombreuses technologies. Je suis très reconnaissant envers Raphaël et Emilie pour avoir appuyé le projet. Merci aux différentes personnes ayant travaillé avec moi sur Afterclasse tout au long de ses quatre années : Aymeric, Guillaume, Marine, Jef, Anna, Gaelle. Enfin, j'ai une pensée particulière pour les yeswerunners des origines - Robin et Claire - pour les kilomètres de discussions aux bords de la Saone.

Les résultats présentés dans ce manuscrit doivent beaucoup aux chercheurs avec qui j'ai pu collaborer. En particulier, les résultats sur les rested rotting bandits sont issus d'une fructueuse collaboration avec Andrea et Alexandra, desquels j'ai énormément appris durant ma première année de thèse. Merci à Pierre, qui a collaboré au second article sur les rotting bandits. Finalement, je remercie Emilie, Odalric et Lilian de m'avoir permis de collaborer sur le projet GLR-UCB.

J'ai été très heureux d'intégrer l'équipe SCOOL (anciennement SEQUEL) de l'INRIA, et je remercie l'ensemble des chercheurs, doctorants et post-doctorants que j'ai pu rencontrer lors de mes passages à Lille. J'ai une pensée particulière pour les nombreux thés partagés avec mes cobureaux, Emilie et Sadegh; ainsi que pour le voyage au Japon en compagnie de Jean. Merci aussi à l'ensemble des organisateurs de la *Reinforcement Learning Summer School*; Nicolas, Omar, Yannis, Edouard, Mathieu, Xuedong et Florian. Je remercie également Yohan et Juliette, la team bandit de la RLSS.

Beaucoup de personnes m'ont soutenu et poussé au cours de ces quatre années étirées entre Lyon, Lille et Paris. Un grand merci aux lyonnais - Josselin, Paloma, Pierre, John, Johanna, Romain, Jean-Baptiste - ainsi qu'à François qui m'aura hébergé lors de mes visites à Lille. Je remercie également mes amis de plus longues dates pour leur soutien: Edouard et Claire-Marie, Laureen, Hugo, Eugénie, Romain, Thomas, Bastien et tous le groupe des chaptaliens. Finalement, je souhaiterais remercier mes deux frères - Arthur et Corentin - ainsi que mes parents pour leur soutien indéfectible.



Contents

I Introduction

1 Afterclasse 25

1.1 A revision website 25

1.2 Exercises: format and data 25

1.3 Scope of the Ph.D.: how to choose the next exercise? 29

1.4 Users and usages 29

1.5 Appendix: contextual elements on Lelivrescolaire.fr 31

2 Exploration in online learning 33

2.1 The multi-armed bandits model 33

2.2 Stochastic bandits 34

2.2.1 Regret minimization 34

2.2.2 Upper confidence bound methods 36
2.2.3 Bayesian methods 38
2.3 Adversarial bandits 39
2.3.1 Pseudo-regret 39
2.3.2 Adversarial methods 40
2.4 Non-stationary bandits 42
2.5 Contextual bandits 43
2.6 Beyond bandits: Reinforcement Learning 44
3 Applications to Intelligent Tutoring Systems 47
3.1 Shortcomings in the bandits model 47
3.1.1 Observation is reward. 48
3.1.2 Comparing to the best action. 49
3.1.3 Actions do not impact observations 49
3.1.4 Learning is quite slow. 49
3.2 Exploration methods in Adaptive Intelligent Tutoring Systems 50
3.2.1 Target the largest improvement 50
3.2.2 Target the least known subject 53
3.2.3 Target faster learning 54
II Rotting bandits
4 Rested rotting bandits are not harder than stationary ones 59
4.1 Rested rotting bandit: model and preliminaries 59
4.1.1 The offline problem (Heidari et al. 2016) 62

4.1.2 The noiseless online problem (Heidari et al. 2016) 63
4.1.3 Levine et al. (2017): wSWA, a first policy for the noisy problem 66
4.1.4 Experimental benchmarks 72
4.1.5 Open problems 75
4.2 FEWA and RAW-UCB: Two adaptive window algorithms 76
4.2.1 Towards adaptive windows 77
4.2.2 FEWA: Filtering on expanding window average 78
4.2.3 RAW-UCB: Rotting Adaptive Window Upper Confidence Bound 82
4.3 Regret Analysis 83
4.3.1 Problem-independent bound 84
4.3.2 Problem-dependent bound 85
4.3.3 Proof 87
4.4 Experimental benchmarks 93
4.4.1 Simulated benchmark #1 (2 arms). 93
4.4.2 Simulated benchmark #2 (10 arms). 95
4.5 Efficient algorithms 98
4.5.1 The numerical cost of adaptive windows 98
4.5.2 The efficient update trick 98
4.5.3 The delay in EFF_UPDATE 102
4.5.4 EFF-FEWA (π_{EF}) and EFF-RAW-UCB (π_{ER}) 109
4.5.5 Regret analysis 110
4.5.6 Experimental Results 115
4.5.7 Conclusion 118

4.6 How harder are rotting bandits ? 118

4.6.1 RAW-UCB++ . . . 118

4.6.2 Experiments . . . 119

4.6.3 Towards a theoretical analysis of RAW-UCB++ . . . 120

4.7 Linear rotting bandits are impossible to learn 122

4.7.1 Linear bandits . . . 122

4.7.2 Linear rested rotting bandits . . . 123

4.7.3 The offline problem . . . 125

4.7.4 The noise-free online problem . . . 126

4.7.5 Proofs . . . 126

5 The rotting assumption makes restless bandits easier 131

5.1 Restless rotting bandits 131

5.1.1 Restless bandits model . . . 131

5.1.2 Piece-wise stationary bandits . . . 132

5.1.3 Variation budget bandits . . . 135

5.1.4 The restless rotting assumption . . . 136

5.2 Analysis of adaptive window policies on restless rotting bandits. 140

5.2.1 Proofs . . . 142

5.3 Real-word data experiment on Yahoo! Front Page 152

5.4 Restless and rested rotting bandits 157

5.4.1 The general case . . . 157

5.4.2 Rested rotting bandits with a restless envelope . . . 158

5.4.3 Proofs . . . 158

III Beyond rotting bandits

6 Master topics as soon as possible 163

6.1 Beyond rotting bandits: some motivations 163

6.2 Setup 164

6.3 Optimal Oracle: Focus on the largest under the threshold 168

6.3.1 The FLUT oracle 168

6.3.2 Optimality 169

6.3.3 Proof of Theorem 6.3.1 169

6.3.4 Technical Lemmas 175

6.4 What does random progression mean? 180

6.5 Learning Perspectives 182

6.5.1 Regret 182

6.5.2 Counter-examples and a new learning assumption 182

6.5.3 Focus on the Largest Under the Threshold with Exploration (FLUT-E) . . 183

6.5.4 Regret upper bound perspectives 185

6.6 Practical considerations for ITS applications 186

6.6.1 Including prior knowledge 186

6.6.2 The exercises population is finite 187

6.6.3 Tuning Δ with ε 187

6.6.4 Managing difficulty with a Zone of Proximal Development 187

IV References

References 191

Chapter header: Giorgio de Chirico, *Edipo e la Sfinge*, 1968, olio su tela, cm 90x70, ©Fondazione Giorgio e Isa de Chirico, Roma.

List of items

Figures

1.1 Grades (Red rectangle) and Subjects (Blue rectangle). 26

1.2 "Multiple choice" is a question with several propositions which can be either true or false. 27

1.3 The "single choice" is a question with several propositions (mostly, 2 or 4) and only one good answer. 27

1.4 "Box" is a type of exercises in which the student should sort several elements (e.g. 5) between two categories ("boxes"). 27

1.5 "Link" exercises consist in linking each element of two different sets. 28

1.6 A "timeline" consists in ordering different elements. It can be some dates in history, but it can be used for any kind of ordered sets. 28

1.7 "Input" is a question with one or several blank holes that the learner should fill. 28

1.8 Active logged-in students and their exercises per month 30

1.9 The churn rate is the proportion of students which do not connect $n+1$ times among the ones which connect n times. 30

3.1 Clement et al. (2015)'s algorithms 51

4.1 **Top:** Regret at the end of the game for different values of L. **Bottom:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region. 73

4.2 **Left:** Regret at the end of the game for different values of L. **Middle, Right:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region. 75

4.3 Three rotting reward functions (red dash line) and associated reward samples: Why should we use a single fixed window size to compare these three arms? 77

4.4 **Top:** Regret at the end of the game for different values of L. **Bottom:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region. 94

4.5 **Left:** Regret at the end of the game for different values of L. **Middle, Right:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region. 96

4.6 Normalized delay d_j/h_j after N_i pulls for each j-th statistic $\widehat{\mu}^{h_j}_{i,\texttt{eff}}$. We display in white the rounds at which statistic j is not created yet. 104

4.7 Normalized delay d_j/h_j after N_i pulls for each j-th statistic $\widehat{\mu}^{h_j}_{i,\texttt{eff}}$. We display in white the rounds at which statistic j is not created yet. 104

4.8 Impact of m on the minimum, maximum, average and median ratio among $\left\{m\omega_j/(m-1)h_j\right\}_j$. 106

4.9 Regret across time. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region. 115

4.10 **Top:** Regret at the end of the game for different values of L. **Bottom:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region. 117

4.11 **Left:** Regret at the end of the game for different values of L. **Middle, Right:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region. 117

4.12 Stationary experiments 120

4.13 **Top:** Regret at the end of the game for different values of L. **Bottom:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region. 121

4.14 **Left:** Regret at the end of the game for different values of L. **Middle, Right:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region. 122

5.1 The reward functions μ and μ'. A policy with low regret on μ cannot achieve low regret on μ'. 133

5.2 **Left:** reward functions from the Yahoo! dataset **Right:** average regret of policies over 500 runs 154

Tables

4.1 Average running time for the 10-arms experiment in seconds. 97

4.2 Average running time and comparison with RAW-UCB for the efficient benchmark. 116

5.1 Average computational time in seconds for each algorithm in each experiment. 153

Algorithms

1 Greedy Oracle π_O (or $\mathscr{A}_0$, Heidari et al. (2016)) 62

2 Greedy Bandit π_G (or $\mathscr{A}_2$, Heidari et al. (2016)) 63

3 SWA (Levine et al. 2017) 67

4 wSWA (Levine et al. 2017) 71

5 FEWA 79

6 FILTER 79

7 RAW-UCB 82

8 EFF_UPDATE 100

9 Focus on the Largest Under the Threshold (FLUT or $\widetilde{\pi}^{\star}$) 169

10 Focus on the Largest Under the Threshold with Exploration (FLUT-E) 185

Chapter header: *Portrait of Luca Pacioli*, traditionally attributed to Jacopo de' Barbari, 1495

Plan détaillé

Cette thèse peut s'approcher de deux manières différentes. En version longue, elle traite de la possibilité d'appliquer des modèles d'apprentissage par renforcement pour améliorer les séquences d'exercices données à un élève sur une plateforme en ligne. En version courte, elle peut se lire comme une thèse apportant des contributions fondamentales à des problèmes de bandits à plusieurs bras non-stationnaires. Commençons par présenter la version longue.

Cette thèse s'est déroulée dans le cadre du dispositif CIFRE avec l'entreprise Lelivrescolaire.fr. Cette entreprise a développé Afterclasse, un site de révision en ligne gratuitement accessible et massivement utilisé. Dans le Chapitre 1, on donne quelques éléments de contexte concernant Afterclasse et Lelivrescolaire.fr. Le but de la thèse y est précisé : améliorer la séquence d'exercices en fonction des résultats de chaque élève, avec un focus sur le court terme. En effet, les élèves "bachotent" sur la plateforme et sont assez peu engagés dans le temps. Il est donc naturel de chercher à les aider sur le court terme. Pour cela, il nous faut évaluer rapidement ce qu'ils savent et ce qu'ils ne savent pas et utiliser cette connaissance pour les réorienter vers les exercices qui leurs seraient les plus utiles.

Le Chapitre 2 présente les modèles les plus classiques d'exploration-exploitation en machine learning. Cette revue de littérature peu mathématisée (par rapport au standard du domaine) a pour but de présenter de manière détaillée mais abordable les questions et les réponses apportées par la communauté du machine learning sur ce dilemme naturel entre exploration et exploitation. Une emphase particulière est mise sur les problèmes de bandits à plusieurs bras. Dans ces problèmes, un agent choisit séquentiellement parmi plusieurs actions et obtient une récompense dépendant de l'action choisit. Le but est de réussir à repérer les actions qui mènent aux plus fortes récompenses. Pour cela, le joueur doit accepter d'explorer - et donc de se tromper - les différentes options afin d'améliorer sa connaissance du problème.

Plusieurs modèles de récompenses sont détaillés. Dans le modèle stochastique stationnaire (Section 2.2), chaque bras est lié à une distribution aléatoire. Le but est donc d'évaluer la moyenne de la distribution tout en quantifiant l'information manquante. Dans le modèle adversarial (Section 2.3), la récompense est choisie par un adversaire et on cherche donc à repérer les actions les plus prolifiques tout en étant suffisamment imprédictible par l'adversaire. Dans les modèles stochastiques non-stationnaires (Section 2.4), la distribution associée à chaque bras peut changer au cours du temps ou en fonction des actions choisies. Cela mène à un double problème : d'estimation statistique d'une part (Quelle est la valeur courante ?), et de stratégie d'autre part (Comment puis-je me prémunir contre la non-stationarité ?). Enfin, les bandits contextuels (Section 2.5) lient les différentes actions à l'aide d'éléments de contexte. Ces éléments permettent d'envisager un très grand nombre d'actions qui ne seront pas toutes explorées : l'information sur les unes permettant d'extrapoler la valeur des autres.

Finalement, on présente en Section 2.6 quelques fondamentaux d'apprentissage par renforcement (*reinforcement learning*). Ce modèle est beaucoup plus général que les modèles de bandits car le joueur possède un état qu'il doit contrôler pour rester dans des zones

à fortes récompenses. Ce modèle est très riche et permet d'apprendre des taches très complexes. Pour autant, il est très consommateur de données et ses meilleures réussites ont souvent lieu dans des dispositifs ou des données peuvent être simulées facilement.

Le Chapitre 3 est consacré aux possibilités d'adaptation du modèle de bandits pour un usage éducatif. Lorsqu'on donne une question à un élève, on observe la réponse à cette question, ce qui est très similaire au *feedback* du modèle de bandit. Cependant, nous notons quatre écueils fondamentaux (Section 3.1) qui entravent un tel usage. Tout d'abord, il est difficile d'associer la réponse d'un élève à une récompense pour l'algorithme. Faut-il privilégier les questions que l'élève réussit ou celles qu'il ne connaît pas ? Une fois la récompense posée, l'algorithme va tenter de la maximiser "brutalement": il faut donc faire très attention au proxy utilisé.

Deuxièmement, les modèles de bandits comparent souvent la performance obtenue à la performance de la "meilleure" action (inconnue). Dans le cas d'une séquence d'exercices, il est assez naturel qu'un exercice puissent être très intéressant à un instant donné, et beaucoup moins intéressant à l'avenir (par exemple, une fois que l'élève le maîtrise parfaitement). Il est donc essentiel de pouvoir se doter de points de comparaison plus intelligents que celui des bandits stationnaires ou adversariaux.

Troisièmement, le modèle stationnaire suppose que ce qu'on observe ne réagit pas à nos actions. Dans le cas d'un système d'apprentissage, on souhaite que nos actions améliorent les performances de l'élève et que *in fine* cela change les réponses qu'il envoie. Autrement dit, on souhaiterait pouvoir incorporer une modélisation de l'interaction entre l'élève et la machine.

Quatrièmement, les élèves font quelques dizaines de questions dans une séquence. C'est très peu, y compris pour un problème simple comme le problème des bandits stationnaires. Cet écueil s'oppose aux trois autres : là où ces derniers suggéraient une modélisation ambitieuse; ici, on doit se contenter de problèmes très simples.

À la Section 3.2, on détaille l'état de l'art de l'usage des modèles de bandits dans les systèmes d'apprentissage intelligents. Ces études empiriques très poussées s'attaquent à plusieurs objectifs, et montrent le plus souvent les comportements intéressants des algorithmes de bandits dans ces systèmes. Cependant, aucun de ces travaux ne s'attaque directement aux problèmes fondamentaux évoqués précédemment.

Dans le Chapitre 4, on étudie la possibilité de faire travailler un élève sur le sujet le moins connu alors que celui-ci progresse au cours de ses révisions. Ainsi, on associe une récompense positive lorsque l'algorithme trouve une question qui n'est pas connue par l'élève (écueil 1). Au fur et à mesure que l'élève se perfectionne (écueil 3), il y a des moins en moins de récompenses sur le sujet sélectionné. Ce dispositif est donc nommé "bandits pourrissant au repos", ce qui traduit la raréfaction des récompenses et l'absence d'évolution sur les sujets non-sélectionnés. Dans ce dispositif, il est nécessaire de changer de sujet lorsque l'élève a suffisamment progressé sur un chapitre peu connu initialement (écueil 2). L'étude statistique du problème permet de conclure que malgré sa très forte richesse, le problème n'est pas plus dur à apprendre que le problème stationnaire (écueil 4). Ainsi, on

montre que le dispositif des bandits au repos apporte une contribution significative aux quatre problèmes fondamentaux détaillés au Chapitre 3. Le Chapitre 5 étudie un autre dispositif de bandits où la récompense décroît indépendamment des choix d'actions. Ce problème est donc nommé "bandits pourrissant sans repos". Ce modèle est fortement lié au premier, bien qu'il ne soit pas motivé par des applications éducatives.

Dans le Chapitre 6, on considère à nouveau que l'élève progresse lors de ses révisions, mais l'objectif n'est plus de maximiser la récompense cumulée. On définit un seuil de maîtrise et on cherche à atteindre ce seuil sur tous les chapitres. Ce problème se place donc dans le cadre plus large des Problèmes de Décisions Markoviens Partiellement Observés (POMDP). La récompense est beaucoup plus implicite qu'elle ne l'est dans le cas des bandits classiques. À l'aide de quelques hypothèses bien choisies, on montre que la meilleure politique oracle est de choisir le chapitre le mieux connu sous le seuil de maîtrise. Cela contraste fortement avec les bandits décroissants: on cherche désormais à viser les chapitres les plus faciles jusqu'à ce qu'ils soient maîtrisés. Fort de notre très bonne compréhension du problème oracle, on propose une heuristique pour la politique apprenante.

Les résultats fondamentaux sur les bandits se concentrent dans les Chapitres 4 et 5. Le Chapitre 4 se décompose de la manière suivante. La Section 4.1 présente le modèle et les deux principaux travaux préliminaires de manière exhaustive. Heidari et al. (2016) ont étudié le problème avec une récompense non-bruitée tandis que Levine et al. (2017) ont étudié le problème bruité. Ils proposent `SWA`, un algorithme avec une borne de regret indépendante du problème $\widetilde{\mathcal{O}}\left(T^{2/3}\right)$.

Cet algorithme utilise un mécanisme de fenêtre glissante de taille fixe. Nous avons proposé deux algorithmes (Section 4.2), `FEWA` et `RAW-UCB`, qui utilisent pour chaque bras des statistiques balayant toutes les fenêtres possibles. `FEWA` utilise ces statistiques pour filtrer les bras les moins bons en partant des échantillons les plus récents.`RAW-UCB` calcule de multiples indices UCB pour chaque bras et utilise le plus petit pour comparer les bras entre eux. L'étude de ces algorithmes (Section 4.3) a permis de prouver une borne de regret indépendante du problème $\widetilde{\mathcal{O}}\left(\sqrt{T}\right)$ et une borne dépendante du problème $\mathcal{O}\left(\log T\right)$. Ces bornes sont comparables avec le problème stationnaire ce qui nous permet de suggérer que ce problème - bien que plus général - n'est pas plus dur que le problème stationnaire.

La performance empirique des algorithmes est testée sur des données simulées (Section 4.4). Non seulement `RAW-UCB` et `FEWA` obtiennent les meilleurs résultats, mais le détail des expériences montrent en plus des différence qualitatives notables dans la forme du regret comparé à `SWA`. En particulier, nos algorithmes sont agnostiques à tous les paramètres du problème à l'exception du niveau de bruit σ. C'est une forte amélioration par rapport à `SWA` qui doit connaître en plus la décroissance maximum ainsi que l'horizon de temps pour obtenir ses meilleures performances.

Nos algorithmes utilisent $\mathcal{O}\left(T\right)$ statistiques par tour et souffrent donc d'une complexité algorithmique prohibitive (en temps et en espace). Cependant, on montre à la Section 4.5 qu'il est possible de réduire les statistiques utilisées (et la complexité afférente) à $\mathcal{O}\left(K\log T\right)$ par tour, tout en retrouvant les mêmes bornes de regret que pour les algo-

rithmes originaux. C'est une meilleure complexité que celle de SWA ($\mathscr{O}\left(T^{2/3}\right)$), bien que, en pratique, SWA soit beaucoup plus rapide. Nous avons essayé d'étendre nos résultats aux bandits linéaires. Cependant, nous avons pu montrer que le problème proposé n'était pas apprenable, même en l'absence de bruit. En effet, la non-stationnarité au repos se comporte mal avec le contexte vectoriel.

Le Chapitre 5 commence par une revue de la littérature sur les bandits sans repos (Section 5.1). Nous reprenons les deux modèles les plus étudiés dans la littérature : les bandits stationnaires par morceaux (Υ_T morceaux) et les bandits avec budgets d'évolution globaux (V_T budget). On rappelle que les bornes minimax sont respectivement $\widetilde{\mathscr{O}}\left(\sqrt{K\Upsilon_T T}\right)$ et $\widetilde{\mathscr{O}}\left(K^{1/3}V_T^{1/3}T^{2/3}\right)$. À la Section 5.2, nous montrons que FEWA et RAW-UCB, sans aucune modification par rapport au chapitre précédent, sont capables d'atteindre ces taux sans connaître les paramètres T, V_T et Υ_T. Plus important encore, ces algorithmes sont capables d'obtenir une borne de regret logarithmique dépendante du problème. Cette borne est inatteignable dans le cas ou la récompense peut croître. En effet, Garivier and Moulines (2011) montre que les algorithmes minimax ont une borne inférieure en $\mathscr{O}\left(\sqrt{T}\right)$ sur tous les problèmes stationnaires. On conclut donc que l'hypothèse de décroissance permet de simplifier les bandits sans repos.

La Section 5.3 propose une évaluation empirique sur des données réelles issues du journalisme en ligne. C'est un jeu de données très utilisé pour les problèmes de bandits non-stationnaires. Cette expérience permet de confirmer nos découvertes théoriques : le regret montre une courbe logarithmique sur les portions stationnaires du problème. Finalement, nous proposons une modélisation avec une non-stationnarité croisée sans repos et avec repos (Section 5.4). Cependant, comme dans le cas linéaire, les deux problèmes sont incompatibles puisque l'on peut montrer une borne inférieur $\mathscr{O}(T)$.

Le Chapitre 6 propose un problème d'exploration-exploitation assez original, intermédiaire entre le RL et les bandits. Il ne contient qu'un seul résultat technique : il s'agit de la preuve de l'optimalité de la politique oracle. Ce résultat est surprenant dans la mesure ou cette politique oracle n'utilise pas sa connaissance de l'opérateur de transition. Comme dans le cas des bandits étudiés aux chapitres précédents, seule la connaissance des valeurs courantes est nécessaire pour se comporter optimalement. Bien que ce chapitre ne contient pas d'analyse complète d'une politique apprenante, il nous paraît être une perspective prometteuse de travaux futurs.

I Introduction

1 Afterclasse 25

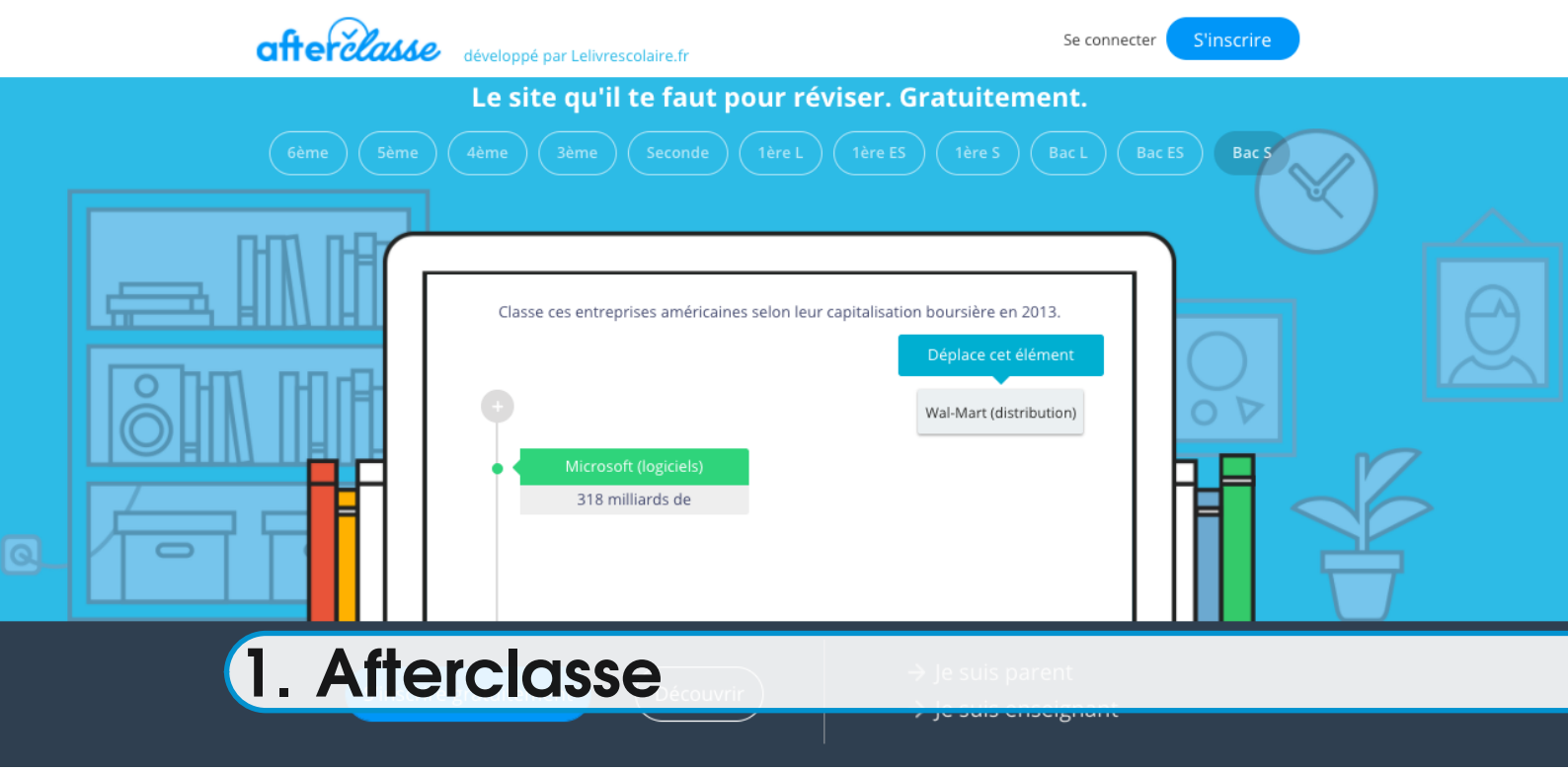
1.1 A revision website
1.2 Exercises: format and data
1.3 Scope of the Ph.D.: how to choose the next exercise?
1.4 Users and usages
1.5 Appendix: contextual elements on Lelivrescolaire.fr

2 Exploration in online learning ... 33

2.1 The multi-armed bandits model
2.2 Stochastic bandits
2.3 Adversarial bandits
2.4 Non-stationary bandits
2.5 Contextual bandits
2.6 Beyond bandits: Reinforcement Learning

3 Applications to Intelligent Tutoring Systems 47

3.1 Shortcomings in the bandits model
3.2 Exploration methods in Adaptive Intelligent Tutoring Systems



1.1 A revision website

Afterclasse.fr is a revision website released in 2015 by Lelivrescolaire.fr, a French EdTech company (section 1.5). While most of the educational content on Lelivrescolaire.fr were thought and designed to be used by teachers with their students (in class or at home), Afterclasse is a platform which is designed directly for students to work independently after classroom hours.

The content covers the official program of the French Education Nationale ministry for middle school - from 11 to 15 years old with the corresponding grades *6ème 5ème 4ème 3ème* - and high school- from 15 to 18 years old with the grades *2nde 1ère Terminale*. In each grade, there are several subjects: French, Mathematics, History and Geography, Physics and Chemistry, Natural Science, Sociology and Economics, English, Spanish, and Philosophy.

For a given grade, the content of a subject is divided into chapters. Each chapter is associated with a revision sheet and a self-assessment mode. The revision sheet gathers the main information about the chapter: a lesson plan, definitions, dates, biography of the main characters or authors, etc.

1.2 Exercises: format and data

In the self-assessment mode, the exercises are given sequentially to students. For each chapter, there are roughly one hundred different exercises. An exercise session contains

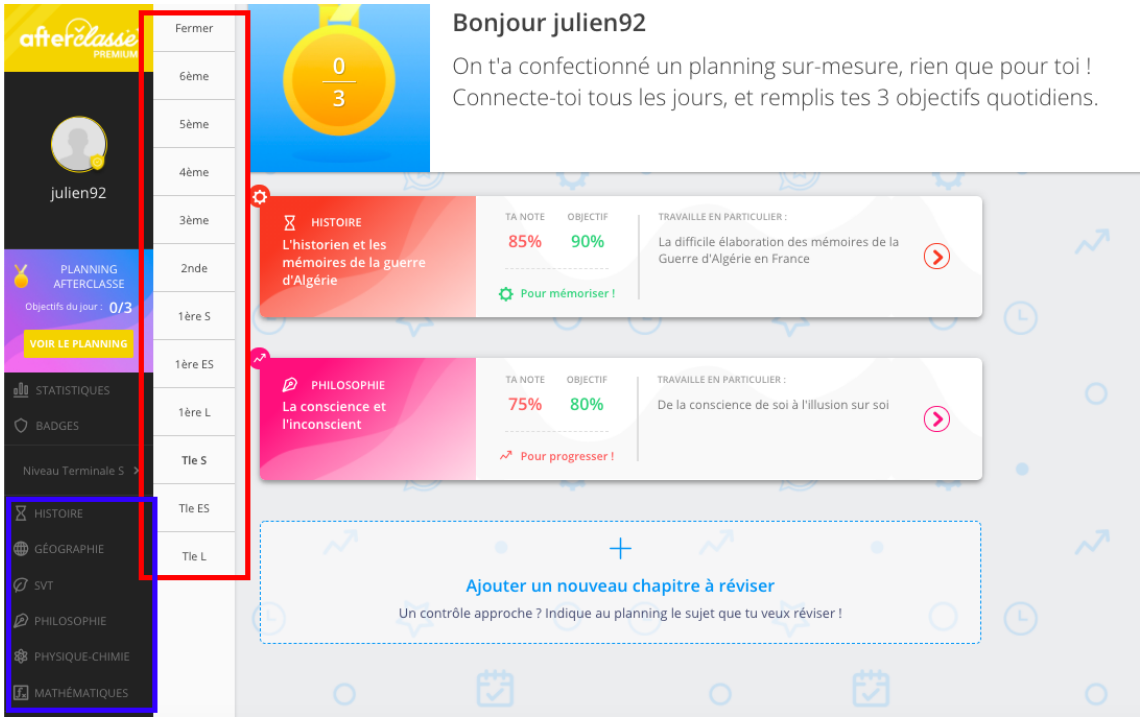


Figure 1.1: Grades (Red rectangle) and Subjects (Blue rectangle).

few (~ 8) exercises. At the end of the session, statistics about the session are displayed to the user, and s/he can choose to restart a new session.

Each exercise is associated with a topic, a difficulty level, and a type. A topic is a subdivision of a chapter. It usually corresponds to a section of the lesson plan. There are two to four topics for each chapter. There are three levels of difficulty in the course. The easy questions are direct applications from the course or involve prerequisite knowledge from the previous chapters. The medium questions are the core of the course. The difficult questions are either complex exercises either questions which involve knowledge slightly beyond the scope of the course. Notice that the difficulty is tagged by a teacher. It does not necessarily quantify the fraction of students which had succeeded it. Indeed, there are questions that might be easy to answer but involve complex knowledge. There are 7 types of exercises: multiple choice (Figure 1.2), single choice (Figure 1.3), box (Figure 1.4), link (Figure 1.5), timeline (Figure 1.6), and input (Figure 1.7).

Figure 1.2: "Multiple choice" is a question with several propositions which can be either true or false.

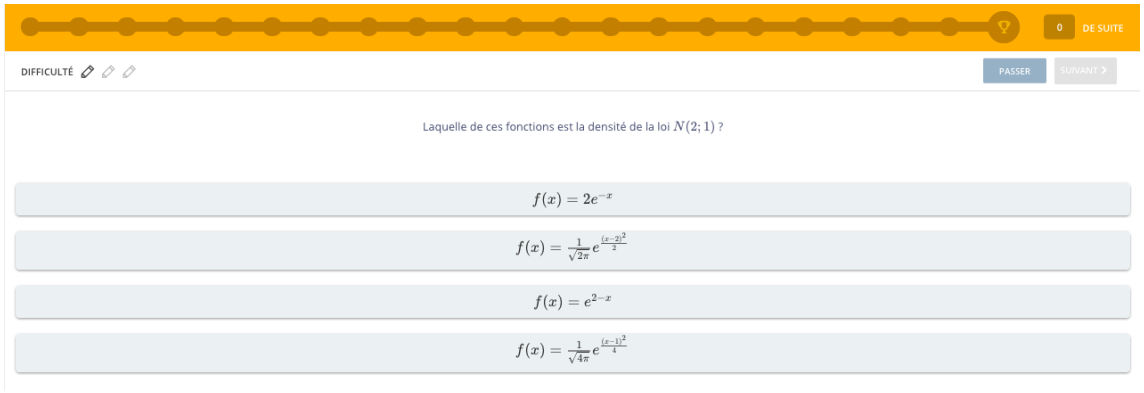


Figure 1.3: The "single choice" is a question with several propositions (mostly, 2 or 4) and only one good answer.

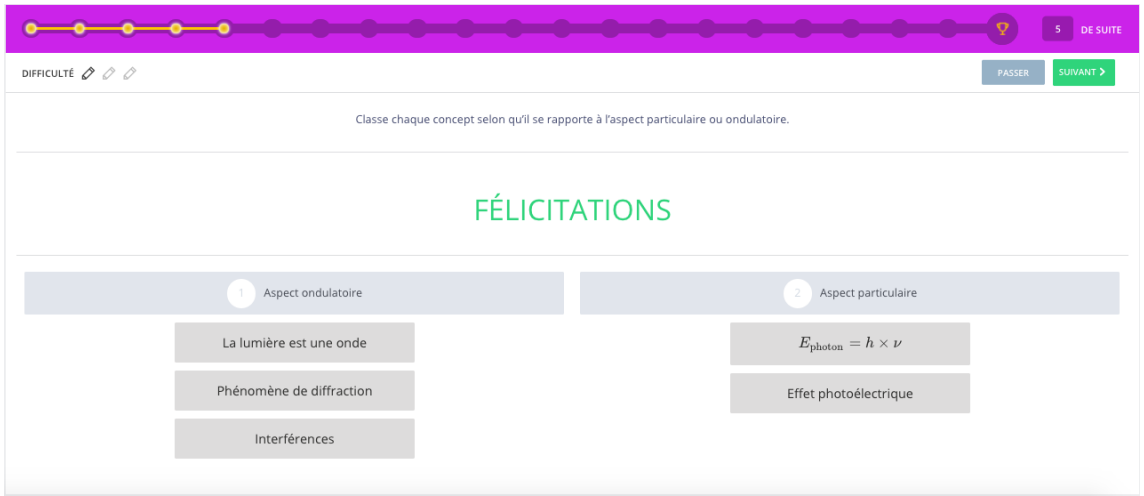


Figure 1.4: "Box" is a type of exercises in which the student should sort several elements (e.g. 5) between two categories ("boxes").

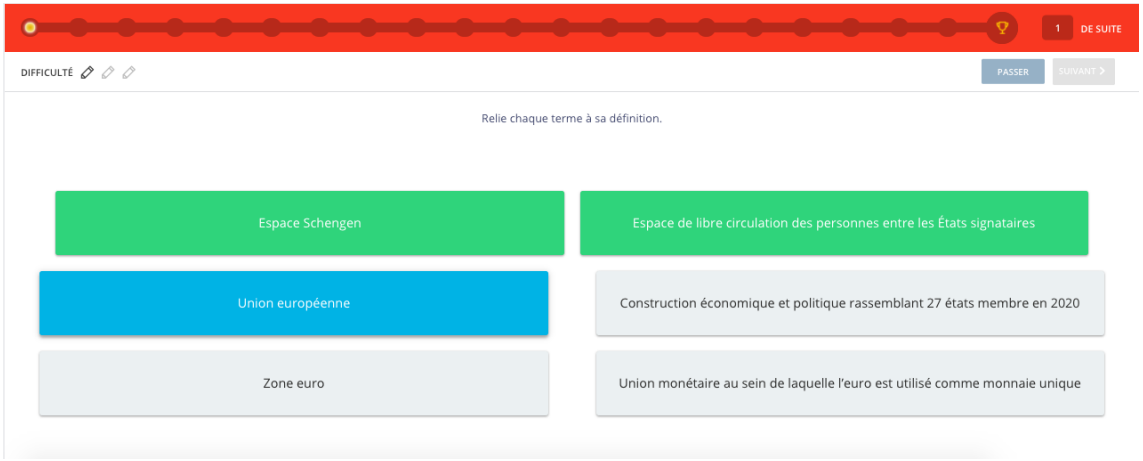


Figure 1.5: "Link" exercises consist in linking each element of two different sets.

Figure 1.6: A "timeline" consists in ordering different elements. It can be some dates in history, but it can be used for any kind of ordered sets.

Figure 1.7: "Input" is a question with one or several blank holes that the learner should fill.

1.3 Scope of the Ph.D.: how to choose the next exercise?

The goal of this Ph.D. is to improve the way we choose the sequence of questions. In particular, we would like to select the next question in an adaptive way. It means that we would like to ask different questions to students depending on their respective estimated proficiency.

There is a two-in-one objective: we would like to (1) accurately estimate the proficiency in order to (2) recommend more relevant questions. There is a trade-off to find between asking a question to measure the proficiency and asking a question because we think they are relevant. This problem is known in machine learning as the exploration-exploitation trade-off. In Chapter 2, we will present the main questions and answers the machine learning community brings to the topic of active exploration-exploitation.

In Chapter 3, we present the main gaps and limits of these general methods for application to educative systems. We also present other attempts to use these methods in adaptive educational systems.

The next three Chapters (4, 5, and 6) are dedicated to the specific bandits problems we consider in this thesis.

1.4 Users and usages

Afterclasse content (both exercises and sheets) is free to use. A premium version is available with convenient tools: a printing option for the sheets, a revision planning to help the student organizing its exam revision on several days, etc. Because the content is free, it is very used nationwide: every year, tens of thousands of students complete several millions of exercises.

In June, there is a peak in the usage before the middle school (*brevet des collèges*) and high school (*baccalauréat*) exams (Figure 1.8). The number of active students in June is multiplied by three compared to the month of October. The total number of exercises is also multiplied by more than six, which shows that the average student is also doing twice more exercises in June than in October.

However, studying is not an addictive behavior and we notice a rather high churn rate (Figure 1.9). Almost half of the logged-in students do not connect more than once to the website. After the first connection, we still loose a large fraction of the students between two connections. This fraction decreases to reach 10% asymptotically. The fact that the churn rate converges to a positive number means that the number of students connecting at least n times decreases exponentially with n.

Most of the students revise only one chapter per logged-in day. The repartition changes significantly just before the exams: half of the students revise 7 chapters per week or more in June while they are only 10% in January.

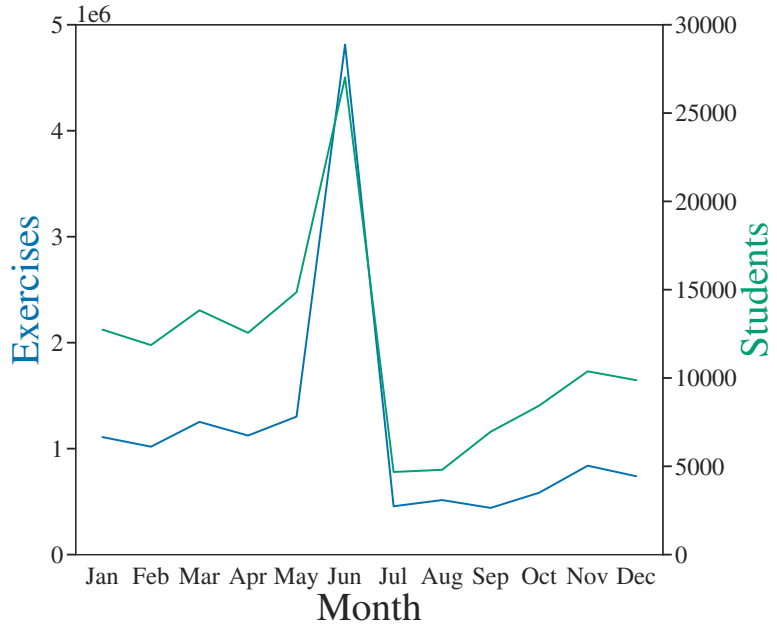


Figure 1.8: Active logged-in students and their exercises per month

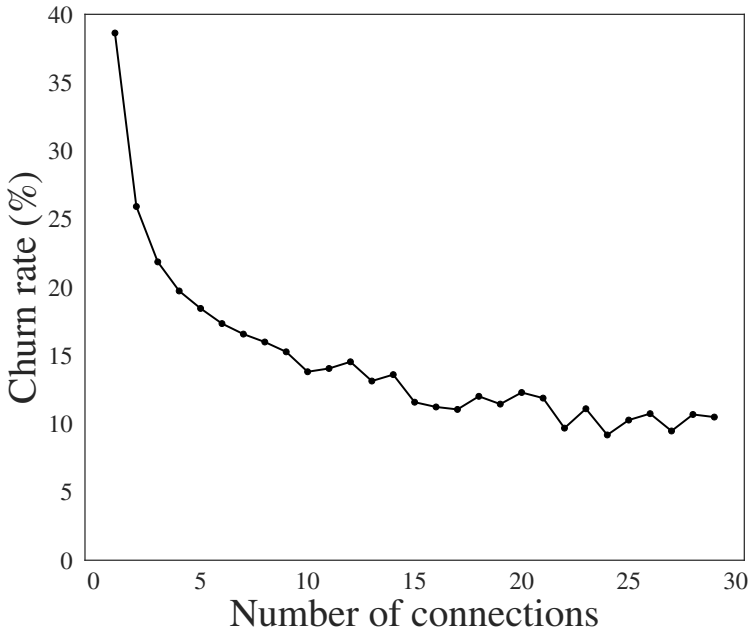


Figure 1.9: The churn rate is the proportion of students which do not connect $n+1$ times among the ones which connect n times.

In brief, the students are cramming on the website: their revisions are short-term, intensive, and targeted (except during the final exams period in which they are broader). Due to the lack of long-term usage (and data), we will not address long-term objectives such

as improving memory retention. Our upcoming approaches will focus on pragmatic short-term objectives: what are the gaps of the student *now* and how can we fill them.

1.5 Appendix: contextual elements on Lelivrescolaire.fr

Lelivrescolaire.fr[1] Editions (or lelivrescolaire.fr) publishes since 2010 educative contents and technologies for the French market. Within a few years, the company became one of the main actors of the national education ecosystem by gathering double expertise: strong publishing know-how and the ability to develop innovative technologies.

Lelivrescolaire.fr brings up three key elements to distinguish themselves from other textbooks publishers. First, they promote free usage of their digital content. Indeed, each textbook's content is available for free on the website (www.lelivrescolaire.fr), which is a unique fact in the French textbook publishing ecosystem. There are more than 2 million visitors each month on the website, which makes their digital textbooks the most used educative digital resources in France.

Second, they write their educative content in a collaborative way. Lelivrescolaire.fr associates more than 100 teachers to the conception of each textbook. In total, it is more than 3000 teachers who have participated in middle school and high school textbooks conception for the last 4 years.

Third, the company promotes constant innovation in the contents and digital media. Each textbook is available on the website and in desktop and tablet Apps. Moreover, the company released many features to enhance digital textbooks experience such as an embedded Python console for the programming exercises, an audio recorder for language learning, or multiple embedded tools for the science curriculum.

More than half of the teachers and one million students in France are using digital content from Lelivrescolaire.fr. For printed books, Lelivrescolaire.fr represents more than 16% of the market for many textbooks, which makes the company one of the leading actors in the French textbook publishing sector. The firm employs 50 workers dispatched in the technology division (engineers, developers, designers), the publishing division (publishers, graphic designers, community managers), and the customer support division (instructors, technical assistant). In January 2020, Lelivrescolaire.fr joined Hachette Livre, the third publishing company in the world.

[1] "le livre scolaire" means "the school textbook".

2. Exploration in online learning

Il était tard lorsque K. arriva. Une neige épaisse couvrait le village. La colline était cachée par la brume et par la nuit, nul rayon de lumière n'indiquait le grand Château. K. resta longtemps sur le pont de bois qui menait de la grand-route au village, les yeux levés vers ces hauteurs qui semblaient vides.

Franz Kafka, Le Château, *Chapitre Premier.*

2.1 The multi-armed bandits model

The multi-armed bandits (MAB) model is a sequential decision process in which the machine learner faces many possible actions. At each round $t \in \{1, \dots, T\}$, it selects one of these actions $i_t \in \mathscr{K}$ (also called "arms") and receives an observation $r_{i_t,t}$ which measures the benefits of this action (also called "reward"). A common goal is to maximize the sum of the rewards collected,

$$J_T = \sum_{t=1}^{T} r_{i_t,t}.$$

In order to do so, the learner should try the different options and discover which action yields the largest rewards. The more the learner try different actions, the more they will be accurate in the future. Yet, there is an inherent cost of "trying" options. Multi-armed bandits methods focus on solving this *exploration-exploitation dilemma*.

The model was first studied in 1933 by Thompson (1933). The denomination "multi-armed bandits" was coined in the 80's in reference to the slot machines. Indeed, in a casino, a gambler may face several machines and wonder which one is the most profitable. Of

Chapter header: *Wanderer above the Sea of Fog*, Caspar David Friedrich, 1818.

course, the model aims at optimizing more interesting or useful trial and error processes like clinical trials (Villar et al. 2015), recommender systems (Tracà and Rudin 2015) or intelligent tutoring systems (Clement et al. 2015; Pike-Burke 2019).

Yet, before going any further in the modelization, we should stress the assumptions that we already made. First, what we observe is connected to the action we took. In particular, we don't observe the reward associated with other actions. This is known as *bandits feedback*. Second, the observation is revealed just after the action choice. Third, the observation measures how good the action is. Last, the sum of the observations is our final objective. It means that rewards are exchangeable, we can trade-off reward in the present for reward in the future.

2.2 Stochastic bandits

2.2.1 Regret minimization

Up to this point, we did not precise how the environment generates rewards. A popular assumption associates each arm i to a stochastic distribution with mean μ_i. Each time an action is selected, the environment outputs an independent reward sample from the arm's distribution. The mean of the distribution can be seen as the intrinsic value of the action. This intrinsic value is only accessible to the learner through the noisy reward.

For instance, in clinical trials, consider many patients who are affected by the same disease. The different actions are the different drugs that could heal the patients. The goal could be to cure as many patients as possible. The learner observes if a patient heals or not. Each drug has its own probability of success that we don't know *before testing*.

If the learner knew in advance the means, he would select the arm with the largest μ_i to maximize the cumulative reward in expectation. How can the learner compare to this oracle strategy? In order to answer to this question, we define the (expected) regret after T rounds, which is the expected difference between the cumulative reward of the oracle strategy and the cumulative reward gathered by the learner,

$$R_T(\pi) \triangleq \mathbb{E}\left[\sum_{t=1}^{T} \mu_\star - \mu_{i_t}\right]$$

with $\mu_\star = \max_{i \in \mathscr{K}} \mu_i$. The expected regret is positive, as the oracle policy obtains the best possible performance in expectation.

How small the regret of the learner can be? In fact, a policy that selects always arm 1 will have zero regret as soon as arm 1 is optimal. However, this policy suffers a regret which scales linearly with the number of rounds T when arm 1 is suboptimal. Thus, this kind of policy is not adaptive at all.

What do we mean by *adaptive*? In fact, for any arms' distributions set, we would like the policy to make fewer and fewer mistakes as it receives feedback. A policy is called

uniformly fast convergent (UFC) when its number of mistakes grows slower than any power of T for any problem parameters μ: $\forall \alpha \in [0,1], R_T(\pi) = o(T^\alpha)$.

What is the cost of being adaptive? T. L. Lai and Robbins (1985) and Burnetas and Katehakis (1996) show that the expected regret per suboptimal arm i for uniformly fast convergent policies is lower bounded asymptotically by $\Omega\left(\frac{\log T}{(\mu_\star - \mu_i)}\right)$ for gaussian noise with known variance [1]. This is the minimal cost on each bandit game to be quite good (*i.e.* consistent) on every one. This is a *problem-dependent* bound because it depends on the value of the arms' means. Later, we will describe famous policies that are proven to get this logarithmic rate (asymptotically) on each bandit game. These policies are called asymptotic optimal because one cannot get better asymptotic performance on any bandit game without suffering very large regret on another problem.

This logarithmic rate is optimal only asymptotically. In fact, when the gap $\Delta_i = \mu_\star - \mu_i$ tends to zero, the rate diverges at finite horizon T. Yet, we cannot have infinite regret as we cannot do more than T mistakes of size Δ_i, *i.e.* at most $T\Delta_i$ regret for arm i. Hence, when Δ_i tends to zero, the regret at finite-time also tends to zero. When Δ_i is large, the cost of each mistake is large, but a good learner can quickly learn from these mistakes and reach the logarithmic asymptotic regime. In between, we have difficult problems, where the learner struggles to detect significant differences between arms and yet suffers a rather large error at each mistake.

What is the worst possible regret a good learner can get for finite-horizon T? Auer et al. (2003) give a quantitative version of the last argument. With K arms, they design a bandit problem where the best arm's mean is separated from the others by a distance of $\mathcal{O}\left(\sqrt{K/T}\right)$. Then, they show that this difference is small enough so the learner does not see significant differences between arms. Hence, in expectation, the optimal arm cannot be pulled a lot more than the others, which is T/K times. Thus, we do roughly $\mathcal{O}(T)$ mistakes of size $\mathcal{O}\left(\sqrt{K/T}\right)$ in this setting, *i.e.* a worst-case (or *problem-independent*) regret rate of at least $\mathcal{O}\left(\sqrt{KT}\right)$.

We will later present some algorithms which match this rate in the worst-case (with an increased constant factor compared to the lower bound). These algorithms are called minimax optimal. The denomination "minimax" comes from game theory, where a player tries to maximize its performance knowing that its adversary will later try to minimize it. Here, the adversary is the environment, which chooses the worst possible gaps between arms.

We presented two types of performance criteria, one which depends on the specific parameter of the bandits we are considering and the other which holds in the worst case. Another point of view is to consider the weighted average performance across multiple bandits games. The weight used in the average is called the *prior* probability distribution across bandit games. This prior represents how likely a bandit game is according to our belief

[1] The original proof by T. L. Lai and Robbins (1985) considered Bernoulli rewards. We present gaussian bandits for the sake of simplicity.

before the game has started. One may recognize the language of Bayesian statistics, and this objective measure is called the Bayesian regret. Bayesian regret is weaker than the problem-dependent bound in the sense that we can deduce a Bayesian regret bound from the problem-dependent bound by averaging. Also, the worst-case regret upper bounds the Bayesian regret.

2.2.2 Upper confidence bound methods

In stochastic bandits, we know that arms have intrinsic values. Each time we pull an arm, we get an observation which is useful in two different ways: first, it is an instantaneous reward; second, it brings some information about the intrinsic value of the arm. The ultimate goal is the cumulative reward the learner gathers, so we would like to estimate how much the extra information is worth in terms of future reward. With this estimation, we could estimate the value of pulling an arm by adding the reward with the value of information.

We call *index policies*, the policies which compute a value for each arm based on the arm's history and select the arm with the largest value. The UCB algorithm uses as index an upper confidence bound on the value of the arm. For instance, if arms are gaussians with known variance σ^2, UCB computes the following indexes,

$$\text{ind}(i) = \widehat{\mu}_{i,t} + \sqrt{\frac{2\sigma^2 \log 1/\delta}{N_{i,t}}}. \tag{2.1}$$

with $\widehat{\mu}_{i,t}$ the average of the $N_{i,t}$ values of arm i at each round t. The average can be seen as the estimate of the instantaneous reward we should get, and the Hoeffding confidence bound term as the value we are willing to pay for the information that the $N_{i,t}+1$-th reward sample should bring.

Yet, this estimated value depends on a parameter δ: how should we tune it? It is possible to show that UCB with $\delta = 1/t$ is asymptotic optimal in the case of gaussian arms with known variance. However, for other distributions, how should we set σ in Equation 2.1? One possibility is to upper-bound the variance. For instance, for Bernoulli distribution, we can use Equation 2.1 with $\sigma^2 = 1/4$. By doing so, we can get a near-optimal logarithmic regret rate, *i.e.* a logarithmic regret rate with a slightly worse problem-dependent factor than the T. L. Lai and Robbins (1985)'s lower bound.

Indeed, upper-bounding the variance means that we "buy" new information at a higher price than what it is worth. For instance, for Bernoulli distribution with a small probability $p \sim 0$, the variance is $p(1-p) \sim p \sim 0$ which is much smaller than $1/4$ when $p = 1/2$. UCB-V (Audibert et al. 2009) is an extension of UCB which estimates the variance empirically. While UCB-V shows improved results over classical UCB in the general case, it is not yet shown to be asymptotic optimal.

In order to get the asymptotic optimal rate, we need better statistical tools. KL-UCB (Cappé et al. 2013) uses the Kullback-Leibler divergence which measures how plausible is a

distribution p' given that data are generated with an other distribution p. More precisely, it computes as index of an arm,

$$\text{ind}(i) = \sup\left\{\mu \in [0,1] \;\middle|\; \mathbb{KL}(\widehat{\mu}_{i,t}, \mu) \leqslant \frac{\log(t) + c\log(\log(t))}{N_{i,t}}\right\} \tag{2.2}$$

The expression of the KL-divergence depends on the family of distributions which is considered. `KL-UCB` uses the KL-divergence of the bernoulli distribution and is shown to be asymptotic optimal for bounded distributions in $[0,1]$[2]. `kl-UCB` uses a very similar index but with a KL-divergence which is specific to a parametric distribution. When the distributions are gaussians with fixed and known variance, `kl-UCB` is equivalent to `UCB`. Yet, in general, the KL-indexes cannot be computed with a closed formula, and we need to use standard optimization software to approximate the index.

`KL-UCB` is asymptotic optimal but only near-minimax optimal $\mathcal{O}\left(\sqrt{KT\log T}\right)$ bounds were proven (even for the simple gaussian bandits' case). We can conjecture that the extra $\sqrt{\log T}$ factor is not an artefact of the proof. It has a clear interpretation: `UCB` buys information at a $\mathcal{O}\left(\sqrt{\log t}\right)$ price. This cost will be paid off asymptotically, but at finite-time, when arms are too close to each other to be distinguished, this information is rather useless. In the early work of T. L. Lai (1987), they suggest to use a refined confidence level $\delta = {}^{N_{i,t}}/_{t}$ in the ucb such that we do not buy information for the most pulled arms. Yet, when the K arms are close to each other $N_{i,t} \sim {}^{t}/_{K}$, so we still buy information at a $\mathcal{O}\left(\sqrt{\log K}\right)$ cost.

The Minimax Optimal Stochastic Strategy `MOSS` (Audibert and Bubeck 2009; Degenne and Perchet 2016) suggests to use $\delta = {}^{KN_{i,t}}/_{t}$. As its name suggests, `MOSS` is minimax optimal. It is also asymptotic optimal for the gaussian case (Lattimore and Szepesvári 2020). Ménard and Garivier (2017) suggested `KL-UCB++`, an algorithm which is minimax and asymptotic optimal for many famous distributions (the single-parameter exponential family). This algorithm uses the tuning of the confidence levels of `MOSS` with the KL divergence upper-confidence bound of `KL-UCB`. Garivier et al. (2018) show similar results for bounded non-parametric distributions. They suggest `KL-UCB-switch`, an algorithm which switches between the index of `MOSS` and the one of `KL-UCB` depending on the allocation of the pulls.

Lattimore (2018) suggests that asymptotic and minimax optimality may not be enough. When there are many arms, but only one suboptimal arm is close to the optimal value (with a distance Δ), it is effectively a two-arm bandit problem. The other arms weigh very little in terms of both regret and number of pulls (for a good policy). Yet, `MOSS` tunes δ with K. In particular, the exploration bonus is canceled after ${}^{T}/_{K}$ pulls, which only guarantees (with high probability) to pull the optimal arm $\mathcal{O}\left({}^{T}/_{K}\right)$ times at the beginning of the game. By contrast, `UCB` keeps exploring the two arms such that they are pulled ${}^{T}/_{2}$ at the beginning of the game. During this starting phase, the two arms' values are not well identified by the algorithm, and the expected regret is linear. This linear phase ends once each arm has

[2] It can be understood by noticing that a Bernoulli of parameter p is the bounded distribution with mean p with the maximal variance. Hence, it can be seen as the worst case from an information theoretic point of view.

been pulled $\mathcal{O}\left(1/\Delta^2\right)$, hence it is K times longer for MOSS than for UCB. In fact, at the end of this phase for MOSS, its expected regret is K times larger than UCB. That is why Lattimore (2018) suggests the sub-UCB criteria, which ensures that the policy is at a constant factor of the performance of UCB at any round t. He also suggests the policy ADA-UCB, which computes for each arm the number of other arms that are "competing" with this arm, and they plug this number instead of K in the confidence level tuning. ADA-UCB is proven to be sub-ucb, asymptotic optimal, and minimax optimal.

We have discussed how optimistic strategies based on upper-confidence bound indexes can achieve multiple optimality criteria. However, the main advantage of UCB could be its simplicity. Indeed, it is a deterministic algorithm, that is, an algorithm that outputs always the same action given the same data. Arguably, this is a desirable property for explainability as well as for an implementation purpose. It is worth noticing that one of the most quoted paper (Auer et al. 2002) in the bandit literature studies a suboptimal version of UCB (namely UCB1) with $\delta = 1/t^4$. It gives a simple proof that leads to a finite-time and problem-dependent regret bound which holds with high-probability and from which we can derive near-optimal minimax and asymptotic bounds. From a research perspective, this simplicity is desirable as it gives a simple starting point when one studies a more complex setup than the stochastic stationary multi-armed bandits.

2.2.3 Bayesian methods

In his early work, Thompson (1933) suggests pulling an arm according to its probability of being the best given the data. It is difficult to compute this probability directly. Hence, we compute for each arm the probability of the parametric distribution beyond it given the data and a prior. Then, we sample a model for each arm according to this distribution and we select the arm with the best mean according to this sampling. This procedure is known as Thompson Sampling (TS).

Though TS is very old, it was only shown recently (Kaufmann et al. 2012b; Agrawal and Goyal 2013) that it is asymptotic optimal (when it is fed with an uninformative prior). Borrowing the idea of canceling the exploration for arms with $N_{i,t} = T/K$ from MOSS, Jin et al. (2020) suggested the Minimax Optimal Thompson Sampling (MOTS) which clipped the posterior distribution at a quantile $\delta = T/KN_{i,t}$. MOTS is minimax and asymptotic optimal.

Bayes-UCB (Kaufmann et al. 2012a) is another asymptotic optimal Bayesian algorithm. It computes an optimistic index based on an optimistic quantile of the posterior. Bayes-UCB and TS have empirical performance very similar to KL-UCB.

The posterior distribution can sometimes be computed explicitly, for instance with the Beta distribution for Bernoulli reward. When it is not possible, one can use Markov-Chain Monte-Carlo (MCMC, Andrieu et al. (2003)). This technique can sample from a probability distribution p, if we know the probability ratio $p(x)/p(y)$ for all x and y. Indeed, when we use the Bayes rules, we often have an unknown normalization factor which can be hard to compute.

2.3 Adversarial bandits

2.3.1 Pseudo-regret

Another popular assumption is to consider the environment fully adversarial (Auer et al. 2003), which means that rewards are generated by an adversary who wants to maximize our regret. But how do we define the regret in this setting? In adversarial bandits, it is not possible to compete with the oracle who would know in advance what reward is beyond each arm at every step. Indeed, let's consider an adversary who rewards one arm uniformly at random at every step, and set the reward of the other arms to zero. An oracle can select the right arm at every step, but a learning policy can only try to guess what is the right arm. "Guessing" an independent random variable cannot be improved with past feedback (by definition of independent), and hence the learner suffers a linear regret rate compared to the best possible sequence.

Thus, we will target a more reasonable objective: we will compare to the best arm in hindsight, *i.e.* we take as reference the best policy (for this reward sequence) among the ones which select always the same arm. Formally, with $r_{i,t}$ the reward of arm i at each round t, we define the pseudo-regret,

$$R_T(\pi) = \max_{i \in \mathcal{K}} \left(\sum_{t=1}^{T} r_{i,t} - r_{i_t,t} \right).$$

The adversarial multi-armed bandit may look much harder than the stochastic bandits due to the latitude the adversary has to trick us. However, Auer et al. (2003) have designed `Exp3` (Exponential weight for exploration-exploitation), an algorithm with a proven worst-case regret upper bound of $\mathcal{O}\left(\sqrt{KT\log(K)}\right)$. This rate was further refined by `INF` (Audibert and Bubeck 2009) to $\mathcal{O}\left(\sqrt{KT}\right)$ when the range of rewards is bounded and known by the learner. It shows that stochastic bandits are not much easier than adversarial bandits from the minimax perspective. More recently, Zimmert and Seldin (2018) designed a variant `Tsallis-INF` which is minimax optimal in both adversarial and stochastic settings and near-asymptotic optimal in the stochastic setting. They also show relevant results in intermediate settings. It tends to show that we can have simultaneously the best of both worlds (without knowing in advance in which world the learner is). Yet, we emphasize that `Tsallis-INF` is not completely asymptotic optimal as it does not recover the right multiplicative constant in the regret rate.

The adversarial bandit framework is a bit odd: on the one hand, the learner tries to compare to the best arm in hindsight; on the other hand, there is no mechanism behind the reward generation of each arm which guarantees any coherence in the sequence. Let's go back to the casino: if the gambler acknowledges that slot machines are just some black boxes the casino uses to diminish its performance, why would they care about comparing to the best machine in hindsight?

There is no fully satisfying answer to this question. An important point is that the learner has to believe in something (because they will suffer linear regret in the worst-case if they

compare to any possible sequence of actions), and the meaning of this belief is not included in the model. A popular extension of the adversarial bandits computes the regret against the best policy in a predefined set of E experts. Auer et al. (2003) suggests Exp4 which is proven to achieve a regret rate of $\mathcal{O}\left(\sqrt{\min(K,E)\,T\log E}\right)$. Notice the logarithmic dependence with the number of experts: we can have a rather high number of experts, but if we consider all the possible sequences of choices, *i.e.* $E = K^T$, the upper bound rate becomes linear with T.

The learner may believe that there is an inner mechanism beyond each arm, such that it makes sense to compare to the best arm. In an old-time casino, each machine may have an independent non-stochastic mechanism such that one is more rewarding than the others. Yet, the mechanism may be complex to model and the learner may be lazy and assume the reward adversarial. The aforementioned "best of both worlds" results may encourage him in that way. However, one should be cautious: low regret compared to bad policies can mean low reward. For instance, if the arms have periodic and synchronous rewards (the reward of arm 1 is low when the reward of arm 2 is high) competing against the best fixed-arm policy may be much less rewarding than competing against experts which are aware of the periodicity.

2.3.2 Adversarial methods

Adversarial games are very different from stochastic games. In the stochastic setting, when we observe the reward for all the actions (a.k.a the full information setting), the learner can follow the actions with the largest current average reward. Indeed, the learner does not need to explore like in the bandit setting, and *Follow the Leader* (FTL) is guaranteed to do less than a constant regret (with respect to T). Yet, in the adversarial full-information setting, FTL suffers a linear regret. Indeed, the adversary can alternate the reward between two arms such as the current "leader" is never rewarded.

In fact, in the adversarial setting, every deterministic policy (like UCB) would fail because a good adversary may know our strategy Hence, it can set to zero the reward of the action we select. That is why we need to design probabilistic strategies that output a probability distribution across actions. We already presented TS, a probabilistic policy. Yet, this policy suffers linear regret in the adversarial setting. Indeed, it is fairly easy to trick optimistic strategies: during the first quarter of the game, we may reward only one arm such that an optimistic stationary policy is very confident that it is the best arm. Then, the adversary can increase the reward of another arm. This arm will be pulled only at a logarithmic pace and even when it is pulled the high reward will be averaged with older lower rewards such that it will take a very long time to realize that something has changed. Recently, Zimmert and Seldin (2018) empirically show that TS suffers near-linear regret even in an intermediate setup called "stochastically constrained adversarial regime". In this setup, the rewards are generated stochastically but the probability distributions beyond arms change a few times during the game without changing the best arm identity. Once again, the key is to exploit the "inertia" of this stationary bandit policy, which average rewards from different distributions.

In adversarial games, the output probability distribution needs to take into account the data while being sufficiently unpredictable for the adversary. This is the spirit of the Follow the Regularized Leader (FTRL) policy. This full-information policy selects the probability distribution which maximizes the expected performance (according to the current data) plus a regularization term that penalized probability distributions that are too concentrated. More formally, with p_t the output probability distribution on arms at each round t, D_t the sum of the observed reward for each arm at each round t, and L a regularizing function,

$$p_t \in \arg\max_p \{< p|D_t > +L(p)\}. \tag{2.3}$$

In the bandit setting, we do not have access to D_t, the sum of the reward for each arm from the beginning of the game to round t. We can estimate D_t with importance weighted estimator, that is, we add $\widehat{r}_{i,t} = \mathbb{1}[i_t = i] r_{i,t}/p_{i,t}$ to the sum at each round. This quantity is equal to zero for all the arms which are not selected and for which we don't know $r_{i,t}$. For the arm which is selected, the reward $r_{i,t}$ is normalized by the probability of selecting the arm. This weighting strategy is unbiased in the sense that $\mathbb{E}[\widehat{r}_{i,t}] = r_{i,t}$ (the expectation is taken on the algorithm randomization conditionally on the observed history before round t).

This estimator is unbiased but has a large variance when $p_{i,t}$ is small and $r_{i,t}$ is large. Indeed, in this case, $\widehat{r}_{i,t}$ will have a very different value depending on whether we pull arm i at the round t or not. This variance will be transmitted to $\widehat{D}_{t,i} = \sum_{s=1}^{t} \widehat{r}_{i,s}$ that we want to use instead of $D_{t,i}$ in Equation 2.3. It means that when we observe a good reward for an arm that is pulled with low probability, it can squash all the other probabilities to almost zero. Then, the algorithm may never recover because it will keep selecting this arm and adding a positive weighted reward to $\widehat{D}_{t,i}$. Yet, if the algorithm did not pull the arm i at the round t in the first place, it would have very different behavior for the same data sequences generated by the adversary.

The solution is to work with losses instead of rewards. We can define the losses $l_{i,t} = 1 - r_{i,t}$, the importance-weighted estimator of the losses $\widehat{l}_{i,t} = \mathbb{1}[i_t = i] l_{i,t}/p_{i,t}$ and the estimated sum of reward $\widehat{D}_{t,i} = \sum_{s=1}^{t} 1 - \widehat{l}_{i,s}$. In that case, the variance can also be high but when $p_{i,t}$ is small and $r_{i,t}$ is small. If we select arm i at a round t, $\widehat{l}_{i,t}$ will be very large and it will reduce $\widehat{D}_{t,i} = \sum_{s=1}^{t} 1 - \widehat{l}_{i,s}$. Hence, according to Equation 2.3, it will squash $p_{i,t+1}$ to zero. This is arguably better for the stability of the algorithm than squashing all the other probabilities to zero. Moreover, arm i may recover after few rounds because $\widehat{D}_{t,i}$ is increased by 1 every time arm i is not pulled.

Up to this point, we did not precise what regularizer L we should use. A good L will penalize probability vectors which are too predictable. In information theory, a classical measure of how predictable is a probability distribution is its (Shannon) entropy: $-\sum_{i \in \mathcal{K}} p_i \log(p_i)$. The larger is the entropy the more unpredictable it is. Hence, we could use the negentropy as a regularizer.

We now have all the ingredients beyond the aforementioned Exp3 algorithm (Auer et al. 2003). Exp3 is equivalent to FTRL (see Equation 2.3) where we use the loss-based importance weighted estimator $\widehat{D}_{t,i} = \sum_{s=1}^{t} 1 - \widehat{l}_{i,s}$ and the unormalized negentropy as regularizer $F(p) = \sum_{i \in \mathcal{K}} p_i \log(p_i) - p_i$. Notice that with this regularizer, there exists a closed-form formula for p_t instead of the implicit formulation in Equation 2.3. This expression is useful for implementation but it hides the main ideas beyond Exp3.

Most of the recent adversarial algorithms use slight (but powerful) modifications of the aforementioned ideas. For instance, we already advertised Tsallis-INF (Zimmert and Seldin 2018), which improves over Exp3 from the minimax adversarial perspective and recovers logarithmic asymptotic bound for the stochastic stationary bandits' case. Tsallis-INF uses Online Mirror Descent (OMD) instead of FTRL. Without going into the details, the two algorithms share deep similarities. In fact, they are even equivalent for some regularizers (McMahan 2011). Zimmert and Seldin (2018) also use a different regularizing function known as Tsallis entropy (Tsallis 1988) and they finally discussed another unbiased estimation scheme of the losses.

2.4 Non-stationary bandits

Since the early stages of the research in bandits (Thompson 1933; Whittle 1980), one of the most desirable properties for a learner would be to adapt to actions whose *value changes over time* (Whittle 1988), as it happens in non-stationary environments. In fact, from applications in medical trials (where the patient can become more resistant to antibiotics) to a modern applications in recommender systems (Chapelle and Li 2011; Tracà and Rudin 2015), assuming that the environment is *stationary is very limiting*.

In the adversarial bandit setting, rewards do not have to be generated by a stationary stochastic process. However, the objective is strongly stationary as the pseudo-regret definition competes against a *fixed* set of policies (e.g. the stationary policies which select always the same arm). As in stationary bandits, we would like to define the regret against the best oracle, or, at least, a good enough one. Indeed, depending on the non-stationarity, it can be challenging to compute the best oracle (also called the offline policy), especially when the choice of the learner impacts the non-stationarity.

With bandit feedback, it can be meaningful to assume that the arms evolve only when they are pulled. In that *rested* case, the learner observes (often with noise) every value. Bouneffouf and Féraud (2016) consider the case where all the arms are evolving with a known trend which depends on the number of pulls. They compare to the greedy oracle which selects the largest available reward at each round. They design a variant of UCB which uses as index the product of the classical ucb index by the trend. This algorithm achieves a logarithmic asymptotic bound similar to UCB's on stationary bandits. Heidari et al. (2016) consider the two monotonic rested cases without noise in the observation. Levine et al. (2017) consider the parametric and non-parametric decreasing (or rotting) rested case with noise. We give a detailed review of their result on the non-parametric rotting case in Chapter 4.

In the *restless* setting, the arms can evolve even when they are not pulled. Hence, the learner does not know the last reward state beyond each arm. Whittle (1988) first consider a very general restless setting: arms are associated with Markov chains with different transition probabilities depending on whether an arm is selected. While Whittle (1988) suggested a heuristic known as the *Whittle's index* policy, the restless bandits problem was later shown to be PSPACE-hard even to approximate (Papadimitriou and Tsitsiklis 1994).

Assuming that the transition probabilities do not depend on the action of the user simplifies the restless setup. Indeed, in that case, the optimal oracle is straightforward: one should pull the arm with the current largest expected reward. When the evolution is deterministic, *i.e.* the Markov chains are replaced by functions of the round, it is possible to approximate this optimal oracle with an online bandit policy when the changes are either not too frequent (Garivier and Moulines 2011) or not too big (Besbes et al. 2014). We give a detailed literature review of the restless bandits with independent evolution in Chapter 5.

While the general restless bandits are unlearnable, some authors studied some specific instances of the restless bandits. For instance, Immorlica and Kleinberg (2018), Pike-Burke and Grunewalder (2019), and Cella and Cesa-Bianchi (2020) studied different models of recharging bandits, where arms' rewards decrease when they are selected, and increase back when the arm has not been pulled for a while. In these problems, the optimal oracle policy for the full horizon regret is hard to compute and the authors often consider approximated oracle policies (Immorlica and Kleinberg 2018; Cella and Cesa-Bianchi 2020) or weakened regret definition (Pike-Burke and Grunewalder 2019).

2.5 Contextual bandits

The contextual bandits framework (Tewari and Murphy 2017) assumes that at each round contextual information is given to the learner. The reward is associated with the action and context such that an action can be good for a given context and bad in another one. Of course, if we have to explore from scratch every time we receive a new context, it can be quite expensive.

A classical assumption is that actions and contexts can be embedded in a vector space such that the reward is smooth enough in that space - e.g. it is a linear form (Abe and Long 1999; Auer 2002; Abbasi-yadkori et al. 2011; Lattimore and Szepesvari 2017) though more complex structures were also considered (Filippi et al. 2010; Valko et al. 2013; Valko et al. 2014). It allows for a potentially infinite number of contexts and actions, as long as there is a finite number of unknown parameters that determine the rewards. Interestingly, while optimistic strategies were shown to perform quite well in this setting (Abbasi-yadkori et al. 2011), Lattimore and Szepesvari (2017) recently advocates that they could not reach asymptotic optimality, unlike in the multi-armed bandits setup.

2.6 Beyond bandits: Reinforcement Learning

Reinforcement learning (Sutton and Barto 2018) extends the former *contextual bandits* so that the context (renamed "state") is controlled by the learner through the actions. The goal is not only to find and exploit the function which relates states and actions to rewards but also to discover the relation between actions and states.

Formally, the Markov Decision Process (MDP, Howard 1960) models this situation as a quadruplet $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}\}$ where $\mathcal{S}$ is the states space, $\mathcal{A}$ the actions space, $\mathcal{T}(s,a)$ the transition operator which associates an origin state and an action to a probability density over the destination states, $\mathcal{R}(s,a,s')$ which associates a probability density over the reward to a transition from s to s' after choosing action a. It is often easier and more meaningful to consider the discounted cumulative reward rather than the cumulative reward at finite-horizon T. Indeed, in some setups, the termination rule may not be known and the learner may discount the reward to take into account the probability of termination.

The exact state of the learner may be observed partially. For instance, the knowledge of students revising on an intelligent tutoring system is not directly observable: each answer gives only limited information about what they know or do not know. In order to model this situation, Partially Observable Markov Decision Process (POMDP, Astrom 1965) adds a set of observation (Ω) and a probability distribution over this observation set for each states-action transition ($\mathcal{O}(s,a,s')$).

Unlike in stationary bandits, finding an optimal policy when we know the MDP parameters is not straightforward. Dynamic programming (Bellman 1966) is a general method which uses a recursive relation - the Bellman equation - on the state value, which is the cumulative value that the agent can expect in a given state if s/he follows a given policy.

Reinforcement learning aims at finding the optimal policy when the reward function $\mathcal{R}$ and transition probabilities $\mathcal{T}$ are not known. There are several quantitative objectives associated with the maximization of the reward. As in the bandit case, we can define the regret for the optimal policy. However, this objective is ambitious in RL as in some problems a single mistake can send irreversibly the learner in a sub-optimal region of the state space. A weaker objective is to minimize the sample complexity, which is the number of rounds after which the policy behaves near optimally with high probability.

The framework of RL models much more complex situations than bandits. However, it is possible to adapt the `UCB` strategy and its optimistic paradigm for RL. `UCRL2` (Jaksch et al. 2009) selects the most optimistic model in a confidence region built around the empirical means and then uses classical dynamic programming method to get the optimal policy. Then, it runs this policy for a while and restarts the procedure. Being optimistic about the transition probabilities is not as straightforward as it is for the reward parameters. Indeed, for the reward, we can simply increase the reward with the confidence bound. For the transition, we cannot simply increase each transition, because, 1) the probabilities would not be normalized; and 2) increasing the probability to reach a low reward region of the state space is a pessimistic choice. Yet, we can find with an optimization software (with complexity $\mathcal{O}(|\mathcal{S}|)$) an optimistic MDP whose transition probabilities lies within a

confidence band around the empirical average and maximize the reward that the learner can get given the optimistic estimation of the reward function. If the diameter of the MDP is finite - that is, one can (with the right policy) reach any state from any other state in a finite expected number of rounds - `UCRL2` recovers near-optimal regret bounds.

`UCRL2` models the environment by fitting the environment (that is, $\mathscr{R}$ and $\mathscr{T}$). This type of method is called *model-based* RL. By contrast, model-free reinforcement learning directly tries to fit the policy without modeling the environment. For instance, `Q-Learning` (Watkins 1989) plays a behavior policy and tries to measure the total values that the learner can get from selecting each action in each state (the q-value function). At the end of the learning phase, it outputs a target policy that selects in each state the action with the largest q-value. The fact that the policy which is used in the learning phase differs from the output one is called *off-policy learning*. The value of a state-action pair is the sum of the expected instantaneous reward - for which we get a noisy yet objective sample - and the value of the destination state when we play the optimal policy. Of course, the optimal policy is unknown but we can approximate the aforementioned value by considering the maximal state-action value we have estimated for the destination state. The fact that we reinforce our estimated values with other estimated values is called *bootstrapping*. Notice that at the beginning of the learning, the values are just a random guess, and bootstrapping may propagate the error to other nodes. Yet, the q-values converge with high probability under mild conditions on the learning rate (Watkins and Dayan 1992).

There exist many other policies than the two we have quickly described. Yet, none of them can learn anything but toy models without extra assumptions. Indeed, there are a priori $|\mathscr{A}||\mathscr{S}|^2$ transition parameters and $|\mathscr{A}||\mathscr{S}|$ reward parameters. When the state space is not very small, it is much more than the $K = |\mathscr{A}|$ reward parameters in the K-armed bandit case ($|\mathscr{S}| = 1$). Hence, it will take thousands of rounds to `UCRL2` to get a basic understanding of a fairly small environment with ten states and ten actions. The problem is even worse for *model-free* methods like `Q-Learning`. Indeed, model-based methods use every sample to estimate the model. In model-free learning, samples are forgotten either because they were only used to evaluate one policy (*on-policy* learning) or because bootstrapping updates by using the (inaccurate) current belief. Hence, they need multiple visits of each state-action pair to converge.

Deep Reinforcement Learning tries to mitigate this issue by using deep neural networks to generalize the experience the learner obtains. For instance, Deep Q-network (Mnih et al. 2013) uses deep networks to learn the q-value function. Yet, using supervised predictive methods in an online active environment is nothing but straightforward. Indeed, in the supervised learning setting, we learn a function that maps observations X to results Y. The way X is generated is assumed to be stationary between the training and production phases. Moreover, Y is assumed to be an objective value that is given to the learner. In the online setting, the observations X are heavily dependent on the policy which is played. With off-policy learning methods, if the policy which is used in the learning phase output a very different X proportion than the optimal policy, then it will bias the neural network. On the other hand, with bootstrapping, the target values Y do not correspond to purely objective values. Indeed, they are constructed using the current belief of the model on the value destination state. The combination of bootstrapping, off-policy learning, and

(supervised) function approximation was called *the deadly triad* because it can lead to unstable algorithms that do not converge to the optimal policy.

Surprisingly, using sophisticated deep networks instead of more classical and simple supervised models leads to more stable algorithms. Indeed, deep networks are trained with mini-batches: we only use a subset of the data to estimate the network's parameters gradient. This is arguably a key feature for online learning applications where incoming data are natural mini-batches for continuous training. However, this feature alone is not sufficient to solve the whole deadly triad problem.

We will not review in detail all the ideas (experience replay, double-Q-network ...) which have improved the stability of DRL methods. Yet, we advertise that this line of work led to superhuman performances in many complex games such as the board game of Go (Silver et al. 2016; Silver et al. 2017) or Starcraft II (Vinyals et al. 2019). It shows that given a potentially infinite source of data - and enough computational power to process it - DRL methods can learn very complex tasks. However, these methods are still sample-inefficient: AlphaZero (Silver et al. 2017) played several million games before reaching superhuman performance in both Chess and Go. For many real-life applications, one may not have a simulator that can produce a tremendous amount of accurate and cheap data.

Improving the sample efficiency is a hot research topic (Yu 2018; Yarats et al. 2019), and there exist more efficient methods than the ones which have been designed for applications with accurate simulators. However, one should notice that the interaction of planning and exploration makes the methods much more data-intensive than in bandits. In a small data situation, it is preferable to frame a given problem as a bandit than to rely on the too general reinforcement learning paradigm. In this thesis, we will mainly focus on bandits problem in order to take into account this small data constraint.

3. Applications to Intelligent Tutoring Systems

Là-haut, le Château, déjà étrangement sombre, que K. avait espéré atteindre dans la journée, recommençait à s'éloigner. Mais, comme pour saluer K., à l'occasion de ce provisoire adieu, le Château fit retentir un son de cloche, un son ailé, un son joyeux, qui faisait trembler l'âme un instant : on eût dit – car il avait aussi un accent douloureux – qu'il vous menaçait de l'accomplissement des choses que votre cœur souhaitait obscurément.

Franz Kafka, Le Château, *Chapitre Premier*.

3.1 Shortcomings in the bandits model

When we ask a question to a student, we observe their answer to this particular question. This is a good example of *bandits feedback*. Facing this partial feedback, the machine learner has to explore the different options to understand what to do. Handling this exploration is the main question of the bandits' literature. This is quite relevant for adaptive Intelligent Tutoring Systems (ITS): if we think that different students should have different learning paths, we have to characterize to which extends a student is different.

Of course, a good *exploration* strategy depends on what we want to achieve. In the previous section, we have presented the most famous objective, *i.e.* the cumulative reward maximization. The main objective is to balance between gathering new information and using this information to collect rewards. This exploration-exploitation dilemma is relevant for ITS: characterizing the student is only a tool to improve the learning. A good ITS should size the effort spent on characterizing the student versus the estimated benefits of such characterization.

Yet, the cumulative reward maximization makes strong assumptions about the benefits. These assumptions strongly orientate the answers the bandits' community gives to this

Chapter header: *À la croisée des mondes (Le Miroir d'ambre)*, John Howe, 2002. (Droits réservés)

exploration-exploitation dilemma. In the following, we will discuss four limits of applying classical bandits methods to Intelligent Tutoring Systems.

3.1.1 Observation is reward.

In the cumulative reward maximization setup, there is an identification between observation and reward. For an ITS, it is rather unclear what is the reward that can be associated with the student answer. One should be careful: the reward measures how well the ITS behaves and not how well the student answers the question. If we reward the ITS for the success of the student, the ITS will find very easy questions for the students, which is arguably not the best pedagogical strategy. In Section 3.2, we will present different rewards that were used for ITS applications.

We advertise some objectives in the bandits' literature that are different from cumulative reward maximization. The Best Arm Identification (BAI) (Audibert and Bubeck 2010; Gabillon et al. 2012) is a pure exploration objective where the learner should output the best arm at the end of the game. There are several quantitative objectives associated with the best arm identification. In the fixed budget setting, one may want to minimize the *simple regret* (Audibert and Bubeck 2010), that is, the difference between the true best arm's and the identified arm's values. Another possibility is to target the probability of outputting the best arm (Carpentier and Locatelli 2016). In the fixed confidence setting (Garivier and Kaufmann 2016; Kaufmann et al. 2016), the learner outputs as fast as possible with high-probability $1-\delta$ an arm at a residual distance ε from the best arm. The algorithms designed for cumulative regret do not work very well in the BAI setting. Indeed, at least from the asymptotical perspective, these algorithms spend $\mathcal{O}(\log T)$ in exploration and most of their budget in exploiting the identified best arm.

The Best Arm Identification still considers observation as "reward", in the sense that the motivation of targeting the arm with the largest observation is based on the identification "large observation = good". This is not the case for Thresholding Bandits (Locatelli et al. 2016; Garivier et al. 2017; Mukherjee et al. 2017) where the learner wants to separate the arms according to their position relative to a threshold. Interestingly, Locatelli et al. (2016) found a near-optimal algorithm which is fully agnostic. This problem is interesting from an educational perspective: if we have several topics with corresponding sets of related questions, we may want to know which topics are mastered by a student. We could define a threshold above which the topic is considered as mastered and use a Thresholding Bandits algorithm.

More generally, *exploration* can be intrinsically interesting for ITS if the goal is to send information to the teacher. That is why Y.-E. Liu et al. (2014) and Erraqabi et al. (2017) considered a setup where the objective combines the cumulative reward and the error the learner do on the estimation of each arm. Erraqabi et al. (2017) show that naive `UCB` algorithm fails in this setting. They describe `ForcingBalance`, an algorithm that directly targets the optimal allocation of pulls for this problem.

3.1.2 Comparing to the best action.

In both the stationary and the adversarial setups, the performance is compared with the policy which selects always the "best" action. For ITS applications, we believe that the best thing to do is not to recommend always the same type of exercises.

In Sections 2.4 and 2.5, we presented two lines of work where the optimal action is not always the same during the game. These approaches have some limits. Contextual bandits need a meaningful representation for the context such that the reward is a simple function in this space. This representation can be learned using offline data, but it breaks the online paradigm. Moreover, contextual algorithms often use more sophisticated techniques. For instance, the simple averages which are used in the classical multi-armed bandits' framework (e.g., for computing the ucb) are replaced by regression methods which are computationally more expensive. Non-stationary bandits also have some important drawbacks. They often require much more exploration than in the stationary case. This is especially true in the restless case, when arms which are not pulled can change. Indeed, this type of non-stationarity is particularly challenging with bandit feedback.

3.1.3 Actions do not impact observations

In the classical adversarial and stochastic setting, the learner has no impact on the observations (rewards) output by the arms. This is a strong limitation for tutoring systems, as we expect a teaching strategy to modify the student's knowledge.

Reinforcement Learning (Section 2.6) models a more general setup where the learner has a state which is changed by the action. In some non-stationary bandits setups (Section 2.4), arms reward is changing when the arm is pulled. It could be reformulated as a "state" which is impacted by the actions of the learner. There is a difference of perspective between RL and non-stationary bandits: bandits methods often focus on tracking the changing rewards to target the (often short-sighted) best action while RL methods design policies that monitor the state's dynamics to remain in rewarding regions of the state space.

3.1.4 Learning is quite slow.

The last paragraphs suggest increasing the complexity of the classical bandits model with context, state, or non-stationarity. However, the stationary stochastic bandits are already quite hard to learn when the horizon is small. Indeed, students often do no more than a few tens (or hundreds) questions. By contrast, bandits algorithm are often evaluated for longer horizon $T > 10^4$ (e.g., Chapelle and Li (2011)).

From the theoretical perspective, the asymptotic rate $\mathcal{O}\left(\log T/\Delta_i\right)$ is larger than the maximal regret $T\Delta_i$ for many gaps $\Delta_i < 0.2$ when the horizon is small ($T = 100$). It means that the asymptotic analysis is not meant for such small horizons (except when the gap is very large). Even the minimax rate $\mathcal{O}\left(\sqrt{KT}\right)$ is not very different from the worst possible rate T for small T.

This *small data* situation is particularly challenging. Special care should be taken to overcome this issue: The learning problem should not be too ambitious, the setup should be correctly designed. In particular, the number of arms (or unknown reward parameters for contextual bandits) should be reduced. Prior information should be included in the model, algorithms should target finite-time and empirical performance.

3.2 Exploration methods in Adaptive Intelligent Tutoring Systems

In this section, we will review previous work which involves bandits and reinforcement learning methods in Intelligent Tutoring Systems. We will focus on the work where the action-observation loop corresponds to the sequence of question-answer of a single student. In these setups, the goal is to explore the student's knowledge and exploit this knowledge to improve educational actions.

Notice that there are also different exploration scenarios where the feedback loop is the sequence of incoming students. The goal is to refine the instructional policy from one student to another. The objective can be to choose the courses that maximize the final grade of the student (Lan and Baraniuk 2016; Xu et al. 2016), or to find the teaching examples sequence that maximizes the score at the test (Lindsey et al. 2013).

3.2.1 Target the largest improvement

Clement et al. (2015) suggest an ITS which selects sequentially a question linked to a knowledge component (KC) and receives the answer of the student. They suggest selecting the action which leads to the largest increase in student's performance. Besides maximizing the learning gain, it is also the action that motivates the most the learner (Gottlieb et al. 2013).

They present two similar algorithms which we reproduce in Figure 3.1. Each algorithm has two components: the first one computes a *Zone of Proximal Developement* (ZPD, Luckin (2001)), the second selects one knowledge component in the ZPD. The ZPD aims to exclude the KCs on which the student is either too good, too bad, or the ones on which s/he does not progress. These algorithms don't use any model nor parametric assumption on the way the student progresses. They compute non-parametric statistics to estimate the current level or the current progress of the student on a KC.

Once the ZPD is set, a bandit algorithm selects a KC. The reward is a difference between the last samples and the before last ones. In the first algorithm (Zone of Proximal Development and Empirical Success - `ZPDES`), they use the average of the $d/2$ last samples minus the average $d/2$ before last samples. In the second algorithm (Right Activity at the Right Time - `RiARiT`), they use the last sample minus a discounted average of the previous ones. These two statistics measure the recent progress of the students.

They claim to use a variant of `Exp4` (Auer et al. 2003). Like `Exp4`, this algorithm is

Algorithm 1 RiARiT and ZPDES ITS algorithms based on Multi-armed Bandits

Require: Set of n_c Knowledge Componets C, set of n_a activities A
Require: γ rate of exploration
Require: distribution for parameter exploration ξ_u
1: Initialize w_a uniformly
2: **if** RiARiT **then**
Require: R Table
3: Initialize estimated competence levels c^L
4: **end if**
5: **while** *learning* **do**
6: Initialize ZPD
7: {Generate exercise:}
8: **for** $a \in ZPD$ **do**
9: $\tilde{w}_a = \frac{w_a}{\sum_j w_j}$
10: $p_a = \tilde{w}_a(1-\gamma) + \gamma\xi_u$
11: Sample a proportional to p_a
12: **end for**
13: Propose activity a
14: Get student answer and compute reward
15: **if** RiARiT **then**
16: Compute reward (Eq. 2)
17: Update competence levels (Eq. 3)
18: Update ZPD based on competence levels
19: **end if**
20: **if** ZPDES **then**
21: Compute reward (Eq. 1)
22: Update ZPD based on pre-requisites graph
23: **end if**
24: $w_a \leftarrow \beta w_a + \eta r$ {Update quality of activity}
25: **end while**

Figure 3.1: Clement et al. (2015)'s algorithms

probabilistic (see Line 11 in Fig. 3.1). The output probability distribution pulls arm according to weights, which are a weighted sum of rewards. This probability distribution also has a uniform component ξ_u, like the vanilla Exp4. Notice that this component was later proven to be unnecessary (Bubeck and Cesa-Bianchi 2012), even to recover high probability guarantees (Neu 2015).

Yet, this bandit algorithm also has major differences with Exp4. First, there are no experts recommendations which are a necessary input of Exp4. Hence, this algorithm is closer to Exp3. In Subsection 2.3, we presented the three ideas beyond Exp3: Follow the regularized leader, a specific regularization, and an unbiased estimation scheme based on importance weight. The specific regularization is responsible for the exponential weights, which are absent from the algorithm of Clement et al. (2015). The importance weights of the loss - the loss is divided by the probability of pulling the arm - are replaced by fixed weights β and ν on the reward (see Line 24). These fixed weights are closer to discounted statistics used in non-stationary bandits (like D-UCB, Kocsis and Szepesvári (2006) and Garivier and Moulines (2011)).

Anyway, the main feature of Exp3 is to guarantee $\widetilde{\mathcal{O}}\left(\sqrt{KT}\right)$ regret compared to the sum of the reward for the policies which select always the same arm. In this setup, we believe that the interest of this result is limited for two reasons. First, the policies which selects always the same KCs are not the most interesting policies from the educational point of view. Second, the rewards are weighted differences of past observations. Hence, there is a telescoping effect in the cumulative reward. For some choices of parameters in the reward definition (e.g. $d = 2$ for ZPDES), this telescoping can be total such that the sum of the rewards is simply the last observation minus the first, *i.e.* a $\mathcal{O}(1)$ quantity. In that case, the $\widetilde{\mathcal{O}}\left(\sqrt{KT}\right)$ guarantee is meaningless. Even for other choices of parameters, the telescoping reduces the cumulative reward range. The more that range is reduced *by construction*, the less interesting is the theoretical guarantee of Exp3.

Hence, we believe that the modifications suggested by Clement et al. (2015) are indeed more interesting than the classical Exp3. Their algorithm targets a more pragmatic goal: selecting randomly the KCs (to ensure diversity in the tasks) while favoring smoothly the KCs which demonstrate recent progress. They provide empirical evidence of the benefits of their algorithms. In a simulated experiment, they show that their algorithms are more adaptable than an expert sequence to the profile of some simulated students. In an in-class experiment on real students, they show that students who were learning with their algorithms achieve more balanced performance between KCs than a control group that was using the expert sequence. They also demonstrate qualitative differences between the behavior of their algorithm and the expert sequence.

As noticed by Pike-Burke (2019), this paper is arguably one of the most advanced works using bandits in ITS. The objective - targeting the topic on which the student progresses - is very appealing. The ZPD design allows some timely exploration by unlocking progressively the most advanced topics. The experiments bring many insightful comparisons between the studied algorithms and the expert sequence. However, this work only partially address the aforementioned shortcomings 3.1.2, 3.1.3, and 3.1.4.

In particular, there are statistical issues with respect to shortcomings 3.1.2 and 3.1.3. The goal of the paper is to aim at the arm with the current largest increase. It is not clear that this problem fall under the cumulative reward maximisation perspective (see our discussion on the telescoping effect). Even in the algorithm, it is not clear that taking a discounted sum of rewards, which are themselves differences of past observations is a statistically efficient way to measure recent progress. We also notice that these algorithms are quite difficult to tune. They have 4 parameters: γ, ν, β and an other parameter for the reward computation.

The work of Clement et al. (2015) was further extended by Mu et al. (2018) to take into account the forgetting of the student and the learning of the ZPD structure.

We also advertise the works of Rollinson and Brunskill (2015) and Käser et al. (2016), which also try to track the progress of the student. These works do not aim at choosing the knowledge component among several possibilities. Instead, they try to decide when one should stop the work on a given skill. Rollinson and Brunskill (2015) suggest stopping when there is a sufficient probability that the prospective learning gain associated with

the next question is below a threshold. The prospective learning gains are estimated with a student model. Notice that these models are trained with the data of many other students such as it reflects the "average" student. The models assume a specific shape of the progression. Hence, different models with the same input sequence can lead to different stopping times, even when they have comparable predictive performance. The issue is that the predictive performance is evaluated on several students, the goal is to predict correctly on average. However, when they are used in instructional policies, these models are required to explain and predict quantitatively the learning of a specific student given a small amount of data. This instructional policy was further extended by Käser et al. (2016) to be able to stop when a student's performance diverges from the model (for instance, for wheel-spinning student) and to include more complex student models such as deep belief network.

3.2.2 Target the least known subject

Melesko and Novickij (2019) suggest targeting the less known subjects. The idea is that the student has more to learn from their mistakes than from their successes. Hence, they suggest rewarding the failed questions and to not reward the succeeded ones.

Rewarding the system for finding the failed exercises has some limits. Some skills are harder to get, and it could be useful to start with the simplest one. It can also be the case that there are some prerequisite dependencies between the different skills. From the motivational point of view, recommending too hard questions may disengage the student.

Yet, consider a student that is learning some geography facts. S/he wants to check if s/he know their lesson. The different topics in the lesson are as hard to learn *a priori*. Yet, the student could have studied a lot the first part of the course and did not spend too much effort on the other parts. The goal of the ITS could be to try to spot the weaker part of the course and teach them with some questions.

Another motivation highlighted by Melesko and Novickij (2019) is the pure-exploration setup, where the goal of the ITS is to find the weakest topic to send the information to the teacher.

Melesko and Novickij (2019) suggest using the classical `UCB` algorithm. They carry many very small data experiments where the number of topics (arms of the bandits) is of the same order of magnitude as the number of questions (horizon). In this context, they recommend the usage of smaller confidence intervals than classical `UCB`. The experiments show improved performance for `UCB` compared to the random strategy.

In their work, Melesko and Novickij (2019) neglect the impact of the questions on the knowledge of the student. The goal is not really to teach through questions, but to find the least understood topic. However, it is surprising that they use an exploration-exploitation algorithm instead of a specialized algorithm from the best arm identification literature.

Teng et al. (2018) also target to find the least known questions with multi-armed bandits

methods. They suggest an algorithm adapted from the linear bandits' literature where the reward depends linearly on an embedding. The algorithm uses several graphs structuring questions, users, and concepts. These graphs are used to infer a vector representation of the users and questions. They bring theoretical and empirical evidence of the performance of their algorithm.

3.2.3 Target faster learning

Rafferty et al. (2016) suggest minimizing the time the student spends to understand a concept. Hence, the cost (negative reward) associated with each pedagogical action is the time the action takes to be completed. They formulate their problem as a Partially Observable Markov Decision Process. Indeed, the student has a knowledge state which is only partially observable by the teacher. The teacher has several actions: some *examples* which teaches the concept to the student, some *quizzes* which retrieves information about the knowledge state of the student, and some *questions with feedback* which do both at the same time. The goal of the learner is to track the state of the student (which encodes what the student does not know) with questions to show some relevant examples.

They test this framework with several student models and several learning scenarios. The algorithm shows significant time reduction compared to random policies. Some student models are better than others. In particular, modeling long-term memory improves performance compared to models that react only to the last seen example.

II Rotting bandits

4 Rested rotting bandits are not harder than stationary ones 59

4.1 Rested rotting bandit: model and preliminaries
4.2 FEWA and RAW-UCB: Two adaptive window algorithms
4.3 Regret Analysis
4.4 Experimental benchmarks
4.5 Efficient algorithms
4.6 How harder are rotting bandits ?
4.7 Linear rotting bandits are impossible to learn

5 The rotting assumption makes restless bandits easier 131

5.1 Restless rotting bandits
5.2 Analysis of adaptive window policies on restless rotting bandits.
5.3 Real-word data experiment on Yahoo! Front Page
5.4 Restless and rested rotting bandits

Decreasing reward

In Subsection 3.2.2, we presented a line of work that aims at asking questions from the least known subject to a student. In the multi-armed bandits' formulation, it associates positive reward to failed questions. Yet, none of these works consider the impact of the questions on the knowledge of the student. When the answer is given to the student after her trial, questions are a powerful learning tool. Therefore, the more the student work on a topic, the better he becomes, and the smaller is the reward for this topic.

Other situations can be modeled with decreasing rewards caused by the repetition of an action. For instance, the more we recommend an item to a user in a recommender system, the more he might get bored (Warlop et al. 2018). In medicine, the efficiency of antibiotics is diminishing with the overall use due to bacteria's mutation (Ventola 2015a; Ventola 2015b).

In microeconomics, the law of diminishing marginal utility states that the utility associated with each unit of goods is decreasing with the number of goods a consumer holds. It is an *ad hoc* explanation to justify that rational consumers, who maximize their total utility, may select different goods. In production theory, the law of diminishing returns (Cannan 1892) states that the increment of production caused by the increment of a factor of production (labor, capital) by one unit is decreasing. Again, there is the idea that repeating always the same action - buying one good, investing in a project - may become suboptimal even though the returns were high at the beginning.

Motivated by these broad applications, Heidari et al. (2016) and Levine et al. (2017) study this non-stationarity with bandits feedback. Heidari et al. (2016) study the case where the rewards are directly observed without noise under the name *decaying bandits*. Levine et al. (2017) study the problem with noisy rewards under the name *rotting bandits*. In the following, we call this problem the *rested rotting bandits* to emphasize that actions cause the rewards' decay. We also mention the works of Immorlica and Kleinberg (2018), Warlop et al. (2018), and Pike-Burke and Grunewalder (2019) which model boredom effects in recommender systems as a rested decaying bandit problem but with restless recharging effects.

In Chapter 4, we synthesized our contributions to the rested rotting bandits problem (Seznec et al. 2019; Seznec et al. 2020): we present new algorithms and we prove that for an unknown horizon T, and without any knowledge on the decreasing behavior of the K arms, these algorithms achieve problem-dependent regret bound of $\widetilde{\mathcal{O}}(\log(T))$, and a problem-independent one of $\widetilde{\mathcal{O}}(\sqrt{KT})$. Our result substantially improves over the algorithm of Levine et al. (2017), which suffers regret $\widetilde{\mathcal{O}}(K^{1/3}T^{2/3})$. These bounds are at a polylog factor of the optimal bounds on the stationary problem; hence our conclusion: rotting bandits are not harder than stationary ones.

Another decaying setup is when the reward decreases no matter what the agent is doing. It models different situations such as the aging of content in recommender systems. Louëdec et al. (2016) models obsolescence of appearing arms (e.g. piece of news) with a known exponential rate. Komiyama and Qin (2014) study a parametric decay in restless bandits

where rewards are linear combinations of known decaying functions. However, the rotting assumption was not studied in the well-studied non-parametric restless bandit setting(Garivier and Moulines 2011; Besbes et al. 2014; F. Liu et al. 2018; Auer et al. 2019; Besson and Kaufmann 2019; Cao et al. 2019; Chen et al. 2019; Cheung et al. 2019; Russac et al. 2019). That is why we consider the *restless rotting bandits* problem in Chapter 5 which is adapted from Seznec et al. (2020). We show that the rotting algorithms designed for the rested case match the problem-independent lower bound and a problem-dependent $\mathcal{O}(\log T)$. The latter was shown to be unachievable in the general case where rewards can increase. We conclude: the rotting assumption makes the restless bandits easier.

Since the same algorithms work in both setups, we investigate in Section 5.4 the joint setup where the reward can decrease with the number of pulls and the rounds. Yet, we show that the optimal oracle policy cannot be approached at a nontrivial rate by a learning policy.

4. Rested rotting bandits are not harder than stationary ones

This rested rotting bandit seems quite stationary to me.

4.1 Rested rotting bandit: model and preliminaries

Feedback loop

At each round t, an agent chooses an arm $i_t \in \mathscr{K} \triangleq \{1,...,K\}$ and receives a noisy reward o_t. The reward associated to each arm i is a σ^2-sub-gaussian random variable with expected value of $\mu_i(n)$, which depends on the number of times n it was pulled before; $\mu_i(0)$ is the initial expected value. We use $\mu_i(n)$ for the expected value of arm i *after* n *pulls* instead of when it is pulled *for the* n*-th time*. Let $\mathscr{H}_t \triangleq \{\{i_s, o_s\}, \forall s < t\}$ be the sequence of arms pulled and rewards observed until round t, then

$$o_t \triangleq \mu_{i_t}(N_{i_t,t-1}) + \varepsilon_t \;\text{ with } \mathbb{E}[\varepsilon_t|\mathscr{H}_t] = 0 \;\text{ and } \forall \lambda \in \mathbb{R},\ \mathbb{E}\left[e^{\lambda \varepsilon_t}\right] \leq e^{\frac{\sigma\lambda^2}{2}}, \tag{4.1}$$

where $N_{i,t} \triangleq \sum_{s=1}^{t} \mathbb{I}\{i_s = i\}$ is the number of times arm i is pulled after round t.

Definition 4.1.1 We introduce $\mathscr{L}_L$, the set of non-increasing reward functions with bounded decay L,

$$\mathscr{L}_L \triangleq \left\{\mu : \mathbb{N} \to [-\infty, L] \;\middle|\; 0 \leq \mu(n) - \mu(n+1) \leq L \text{ and } \mu(0) \in [0, L]\right\}.$$

R We define the set of constant reward functions in $[0,L]$:

$$\mathscr{S}_L \triangleq \left\{\mu : \mathbb{N} \to [0, L] \;\middle|\; \mu(n) = \mu_i\right\}.$$

Chapter header: Photo of a tag by Nemo, Paris, Rue Le Brun.

We have that $\mathscr{S}_L \subset \mathscr{L}_L$. Hence, we can conclude that the rotting bandits model includes all the stationary bandits problems.

Online and offline objectives

In this chapter, we will only consider deterministic agents which output an arm i at each round t. They are degenerate cases of probabilistic agents, which outputs a probability distribution over arms at each round. For the sake of simplicity, we present only the deterministic formalism.

We will distinguish two types of policies. On the one hand, an offline (or oracle) policy $\pi \in \Pi_{\mathrm{O}}$ is a function which maps the round t and the set of reward functions $\mu \triangleq \{\mu_i\}_{i\in\mathscr{K}}$ to arms, i.e. $\pi(t,\mu) \in \mathscr{K}$. On the other hand, an online (or learning) policy $\pi \in \Pi_{\mathrm{L}}$ is a function from the history of observations at time t (which includes the knowledge of the round t) to arms, i.e., $\pi(\mathscr{H}_t) \in \mathscr{K}$. For both types of policies, we often use the shorter notation $\pi(t)$, where the dependencies on μ or $\mathscr{H}_t$ is implicit.

For a policy π, let $N_{i,t}^{\pi} \triangleq \sum_{s=1}^{t} \mathbb{I}\{\pi(s) = i\}$ be the number of pulls of arm i at the end of round t. The performance of a policy π is measured by the (conditionally expected) rewards accumulated over time,

$$J_T(\pi) \triangleq \sum_{t=1}^{T} \mu_{\pi(t)}\left(N_{\pi(t),t-1}\right) = \sum_{i\in\mathscr{K}} \sum_{n=0}^{N_{i,T}^{\pi}-1} \mu_i(n). \tag{4.2}$$

The cumulative reward depends only on the number of pulls of each arm at the horizon T: it does not depend on the specific pulling order of the arms. Hence, two distinct policies with the same pulling allocation at the horizon T, *i.e.* $N_{i,T}^{\pi_1} = N_{i,T}^{\pi_2}$ for all i, have the same cumulative reward.

We notice that $\pi \in \Pi_{\mathrm{L}}$ uses the (random) history observed over time, and thus $J_T(\pi)$ is also random for learning policies. The goal of the learning agent is to maximize the expected reward $\mathbb{E}[J_T(\pi)]$.

On the contrary, oracle policies do not depend on the (random) history. They can be computed entirely before the start of the game. Hence, finding $\pi^\star \in \arg\max_{\pi\in\Pi_{\mathrm{O}}} J_T(\pi)$ is called the *offline problem*. For a given problem μ, there is a finite number (K^T) of policies, hence the maximum always exists and it could be found by brute-force with infinite computational power.

We set a policy $\pi^\star \in \arg\max_{\pi\in\Pi_{\mathrm{O}}} J_T(\pi)$. Calling $J_T^\star = J_T(\pi^\star)$ the largest cumulative reward achievable, one can measure the regret of any policy (learning or oracle) compared to the optimal one,

$$R_T(\pi) \triangleq J_T^\star - J_T(\pi). \tag{4.3}$$

Let $N_{i,T}^{\star} \triangleq N_{i,T}^{\pi^{\star}}$ be the number of times that arm i is pulled by the oracle policy $\pi^{\star}$ up to time T (excluded). Using Equation 4.2, we can conveniently rewrite the regret as,

$$\begin{aligned} R_T(\pi) &= \sum_{i \in \mathscr{K}} \left(\sum_{n=0}^{N_{i,T}^{\star}-1} \mu_i(n) - \sum_{n=0}^{N_{i,T}^{\pi}-1} \mu_i(n) \right) \\ &= \sum_{i \in \text{UP}} \sum_{n=N_{i,T}^{\pi}}^{N_{i,T}^{\star}-1} \mu_i(n) - \sum_{i \in \text{OP}} \sum_{n=N_{i,T}^{\star}}^{N_{i,T}^{\pi}-1} \mu_i(n), \end{aligned} \tag{4.4}$$

where we define $\text{UP} \triangleq \left\{ i \in \mathscr{K} | N_{i,T}^{\star} > N_{i,T}^{\pi} \right\}$ and likewise $\text{OP} \triangleq \left\{ i \in \mathscr{K} | N_{i,T}^{\star} < N_{i,T}^{\pi} \right\}$ as the sets of arms that are respectively under-pulled and over-pulled by π with respect to the optimal policy. In the following, when there is no possible confusion about the policy π, we simply call $N_{i,t}^{\pi} = N_{i,t}$.

R The regret is measured against an optimal allocation over arms rather than a fixed-arm policy as it is a case in adversarial and stochastic bandits. Therefore, even the adversarial algorithms that one could think of applying in our setting (e.g., `Exp3` of Auer et al. (2002)) are not known to provide any guarantee for our definition of regret. Moreover, for constant $\mu_i(n)$-s, our problem and definition of regret reduce to the ones of stationary stochastic bandits.

We give an upper bound on the regret that holds for any policy and will be used in the analysis of all the presented learning policies. First, we upper-bound all the rewards in the first double sum - the underpulls - by their maximum $\mu_T^+(\pi) \triangleq \max_{i \in \mathscr{K}} \mu_i(N_{i,T})$. Indeed, for any overpulls $\mu_i(n_i)$ (with $n_i \geq N_{i,T}$), we have that

$$\mu_i(n_i) \leq \mu_i(N_{i,T}) \leq \mu_T^+(\pi) \triangleq \max_{i \in \mathscr{K}} \mu_i(N_{i,T}),$$

where the first inequality follows by the non-increasing property of μ_is; and the second by the definition of the maximum operator. Second, we notice that there are as many underpulls than overpulls (terms of the second double sum) because both policies π and $\pi^{\star}$ pull T arms. Notice that this does *not* mean that for each arm i, the number of overpulls equals to the number of underpulls, which cannot happen anyway since an arm cannot be simultaneously underpulled and overpulled. Therefore, we keep only the second double sum,

$$R_T(\pi) \leq \sum_{i \in \text{OP}} \sum_{n=N_{i,T}^{\star}}^{N_{i,T}-1} \left(\mu_T^+(\pi) - \mu_i(n) \right). \tag{4.5}$$

The *online problem* is to find a learning policy that maximizes the expected cumulative reward (or equivalently minimizes the expected regret). In the next sections, we will present the main results of Heidari et al. (2016), which has solved the offline problem and the online problem in the absence of noise, and Levine et al. (2017), which has presented the first learning policy with nontrivial guarantees for rotting bandits with noise.

4.1.1 The offline problem (Heidari et al. 2016)

We consider the greedy oracle policy π_O (Alg. 1) which selects at each round the arm with the next best value.

Algorithm 1 Greedy Oracle π_O (or $\mathscr{A}_0$, Heidari et al. (2016))

Require: $\{\mu_i\}_{i\in\mathscr{K}}$
1: Initialize $N_i \leftarrow 0$ for all $i \in \mathscr{K}$
2: **for** $t \leftarrow 1, 2, \dots$ **do**
3: PULL [a] $i_t \in \arg\max_{i\in\mathscr{K}} \mu_i(N_i)$
4: $N_{i_t} \leftarrow N_{i_t} + 1$
5: **end for**

[a] One can choose the tie break selection rule arbitrarily, e.g. by selecting the arm with the smallest index.

Proposition 4.1.1 — Heidari et al. (2016). For any reward functions $\mu \in \mathscr{L}_{+\infty}^K$ and any horizon T, $\pi_O \in \arg\max_{\pi\in\Pi_O} J_T(\pi)$.

Proof. At each round t, π_O collects the largest reward that can be available in the future, *i.e.*

$$\forall i \in \mathscr{K},\ \forall n_i \geq N_{i,t},\ \mu_{\pi_O(t)}\left(N_{\pi_O(t),t}\right) \geq \mu_i\left(N_{i,t}\right) \geq \mu_i(n_i).$$

The first inequality is due to the selection rule of the policy; the second is due to the decreasing reward functions.

A direct consequence is that, at the round T, π_O has selected the T largest reward samples among the KT possible ones. Therefore, any other policy which would select other reward samples can only have a worse cumulative reward. ■

For a given horizon T, all the policies with the same number of pulls of each arm than π_O at the round T have the optimal cumulative reward. Yet, we show in the following Proposition that π_O is the only optimal policy at every round.

Proposition 4.1.2 Let π such that $\pi(t) \notin \arg\max_{i\in\mathscr{K}} \mu_i(N_{i,t})$.

Then, $J_t(\pi) < J_t(\pi_O) = \max_{\pi\in\Pi_O} J_t(\pi)$.

Proof. Let $i_t^\star \in \arg\max_{i\in\mathscr{K}} \mu_i(N_{i,t})$. We consider the policy π^+ which selects the same arm than π during the $t-1$ first rounds and selects $i_t^\star$ at a round t. Therefore, the two policies π and π^+ collects the same rewards except the last one. Notice that before the last round t, the two policies have the same pulling allocation $N_{j,t-1}^{\pi} = N_{j,t-1}^{\pi^+}$ for all $j \in \mathscr{K}$. Hence, there is only a difference between the two last reward samples,

$$J_t(\pi^+) - J_t(\pi) = \mu_{i_t^\star}(N_{i_t^\star,t-1}^{\pi^+}) - \mu_{\pi(t)}(N_{\pi(t),t-1}^{\pi}) = \mu_{i_t^\star}(N_{i_t^\star,t-1}^{\pi}) - \mu_{\pi(t)}(N_{\pi(t),t-1}^{\pi}) > 0.$$

The inequality follows from $\pi(t) \notin \arg\max_{i\in\mathscr{K}} \mu_i(N_{i,t})$ and $i_t^\star \in \arg\max_{i\in\mathscr{K}} \mu_i(N_{i,t})$. ■

R **Complexity.** We have already highlighted that the offline problem is a computational problem. Indeed, the optimal solution can always be computed by brute force by iterating all the possible policies, i.e. with exponential time complexity per round $\mathcal{O}(K^T)$. By contrast, π_O can be computed with space complexity $\mathcal{O}(K)$ and time complexity per round $\mathcal{O}(\log K)$. Indeed, at each round one should find the maximum among K values. Yet, from one round to another, there is only one value which changes: the value of the last selected arm. Thus, one can store a sorted list of the K arm's value and change one element at each round which costs $\mathcal{O}(\log K)$. Then, accessing the first element of the sorted list is a $\mathcal{O}(1)$ operation.

To conclude, π_O solves the offline problem in the sense that it provides a cheap way to compute the optimal policy without any knowledge of the horizon T. Interestingly, π_O takes the optimal decision by being greedy on the current values. It shows that there is no planning aspect in this problem: the learner never has to sacrifice rewards in the present to get more rewards in the future.

4.1.2 The noiseless online problem (Heidari et al. 2016)

In the online problem, the learner does not have access to the current value of the arms. Can they track the best current value using only the observed past values? Heidari et al. (2016) first studied the simpler noise-free problem ($\sigma = 0$), where the learner observes the true value of an arm after selecting it (instead of a noisy sample). They suggested the greedy bandit π_G (Alg. 2), a policy that selects greedily the arm with the largest last observed value. Indeed, instead of looking at the (unavailable) current values as π_O, π_G looks at the closest past.

Algorithm 2 Greedy Bandit π_G (or $\mathscr{A}_2$, Heidari et al. (2016))

Require:

Initialize $\widehat{\mu}_i^1 \leftarrow +\infty$ for all $i \in \mathscr{K}$
for $t \leftarrow 1, 2, \dots$ **do**
    PULL [a] $i_t \in \arg\max_{i \in \mathscr{K}} \widehat{\mu}_i^1$; RECEIVE o_t
    $\widehat{\mu}_{i_t}^1 \leftarrow o_t$
end for

[a] One can choose the tie break selection rule arbitrarily, e.g. by selecting the arm with the smallest index.

Proposition 4.1.3 — Heidari et al. (2016). For any problem $\mu \in \mathscr{L}_L^K$ and any horizon T,

$$R_T(\pi_G) \leq (K-1)L.$$

The worst case regret is upper-bounded by a constant with respect to T. This is surprising as the reward can change at every round. Yet, it is impossible to trick π_G to do more than one mistake per arm.

Indeed, consider the two arm bandit scenario where $\mu_1(n) = -(n-0.5)$ and $\mu_2(n) = -2n$ ($L = 2$). After the two first round-robin pulls, π_G selects arm 2 and collects $\mu_2(1) = -2$

reward instead of $\mu_1(1) = -1.5$ for arm 1. Hence, it is the first mistake on arm 2. At the fourth and fifth pulls, it selects arm 1 twice with value $\mu_1(1) = -1.5$ and $\mu_1(2) = -2.5$ which is better than the current value of arm 2. At the sixth pull, it pulls arm 2 with value $\mu_2(2) = -4$ instead of arm 1 with value $\mu_1(3) = -3.5$. This is a second mistake for arm 2. However, the first mistake was canceled by the decay. Indeed, the regret at a round t compares with the t largest reward value. Hence, pulling $\mu_2(1)$ is a mistake at the round 3 because it is the fourth largest value among all the possible rewards. Yet, it is not a mistake anymore at the round 6 when π_{G} pulls the suboptimal value.

An important consequence of this argument is that the regret at t can decrease with t. Indeed, for any policy, if an arm i_2 becomes optimal after the decay of another arm i_1, any mistake which was potentially done on arm i_2 becomes henceforth an optimal pull, in the sense that it is selected by the optimal policy. It shows the forgiving nature of the rested rotting setup.

Proof. We start from Equation 4.5 applied to policy π_{G},

$$R_T(\pi_{\mathrm{G}}) \leq \sum_{i \in \mathrm{OP}} \sum_{n=N^\star_{i,T}}^{N_{i,T}-1} \left(\mu_T^+(\pi_{\mathrm{G}}) - \mu_i(n)\right). \tag{4.6}$$

Let $i \in \mathscr{K}$ an arm which is pulled at least twice at the end of the game $N_{i,T} \geq 2$. We call $t_i \triangleq \min\{t \leq T \mid N_{i,t} = N_{i,T}\}$ the last round at which i is pulled. For any arm $j \in \mathscr{K}$ pulled at least once at the end of the game $N_{j,T} \geq 1$, and for all $n_i \leq N_{i,T} - 2$,

$$\mu_i(n_i) \geq \mu_i(N_{i,T} - 2) = \mu_i(N_{i,t_i-1} - 1) \geq \mu_j(N_{j,t_i-1} - 1). \tag{4.7}$$

The first inequality follows by the non-increasing hypothesis on the reward function. The equality follows by definition of t_i. The last inequality is by definition of the policy: at time t_i, π_{G} selects $i \in \arg\max_{j \in \mathscr{K}} \mu_j(N_{j,t_i-1} - 1)$, the largest last observed sample.

We choose j such that $\mu_j(N_{j,T}) = \mu_T^+(\pi_{\mathrm{G}}) \left(\triangleq \max_{j' \in \mathscr{K}} \mu_{j'}(N_{j',T})\right)$.
Since $t_i \leq T$, $N_{j,t_i-1} - 1 < N_{j,T}$. By the rotting assumption,

$$\mu_j(N_{j,t_i-1} - 1) \geq \mu_j(N_{j,T}) = \mu_T^+(\pi_{\mathrm{G}}). \tag{4.8}$$

Gathering Equations 4.7 and 4.8, we have that

$$\forall i \in \left\{i' \in \mathscr{K} | N_{i',T} \geq 2\right\}, \forall n_i \leq N_{i,T} - 2,\ \ \mu_i(n_i) \geq \mu_T^+(\pi_{\mathrm{G}}). \tag{4.9}$$

Therefore, we can upper-bound all the before last terms in each second sum in Equation 4.6 by zero. Hence,

$$\begin{aligned} R_T(\pi_{\mathrm{G}}) &\leq \sum_{i \in \mathrm{OP}} \left(\mu_T^+(\pi_{\mathrm{G}}) - \mu_i(N_{i,T} - 1)\right) \\ &\leq \sum_{i \in \mathrm{OP}} \left(\mu_T^+(\pi_{\mathrm{G}}) - (\mu_i(N_{i,T} - 2) - L)\right) \\ &\leq |\mathrm{OP}| L \\ &\leq (K-1)\, L \end{aligned}$$

In the second inequality, we used $\mu_i \in \mathscr{L}_L$ (see Definition 4.1.1). The third inequality follows from Equation 4.9. We can conclude by noticing that they are at most $K-1$ overpulled arm. Indeed, there are as many overpulls than underpulls since the two policies $\pi^\star$ and π_{G} both pull $T-1$ sample. Hence, if there is at least one overpulled arm, there is necessary at least one underpulled arm. ■

In the next proposition, we state that this rate is minimax optimal at the first order in K/T.

Proposition 4.1.4 — Heidari et al. (2016). For any policy $\pi \in \Pi_{\mathrm{L}}$ and any horizon $T \geq K-1$, there exists a K-arm stationary bandit problem $\mu \in \mathscr{S}_L \subset \mathscr{L}_L$ (see the remark following Definition 4.1.1),

$$R_T(\pi) \geq \frac{(K-1)L}{1+K-1/T}.$$

This proposition is slightly more precise than the one in Heidari et al. (2016). Indeed, while they show only a $\mathscr{O}(K)$ worst case rate, we show that π_G is minimax optimal up to a second order term in $\mathscr{O}(K/T)$. Even for $K \sim T$, π_{G} is optimal up to a factor 2. Moreover, we show that this lower bound holds for the easier stationary problem. Hence, it shows that, without noise, rotting bandits are not harder than stationary ones.

Proof. We consider a set of K problems where

- the first arm has always a constant value equals to $L\left(1-\alpha\frac{K-1}{T}\right)$ with α a number that we will specify later;
- problem $p=1$ has all the other arms with a value 0;
- problem $p \in \{2,\dots,K\}$ has arm p with value L and the other arms $i \in \mathscr{K} \setminus \{1,p\}$ with a value 0.

The learner can distinguish between problem $p \in \{2,\dots,K\}$ and problem 1 only by pulling arm p once. If the learner $\pi \in \Pi_{\mathrm{L}}$ pulls every arm $i \in \{2,\dots,K\}$ at least once, it suffers on problem 1,

$$R_T^1(\pi) \geq (K-1)L\left(1-\alpha\frac{K-1}{T}\right).$$

If there exists an arm $i \in \{2,\dots,K\}$ which is never pulled, π suffers on problem i,

$$R_T^i(\pi) \geq T\left(L - L\left(1-\alpha\frac{K-1}{T}\right)\right) = \alpha L(K-1).$$

We can choose $\alpha = \frac{1}{1+K-1/T}$ to balance the two costs. Therefore, we have that for any π, there exists a stationary problem $\mu \in \mathscr{S}_L$ such that,

$$R_T(\pi) \geq \frac{(K-1)L}{1+K-1/T}.$$

■

R Heidari et al. (2016) have also studied rested bandits with increasing and concave reward functions (without noise). The offline analysis shows that the optimal policy selects always the same arm. This is very different from the rotting case, where the optimal allocation may pull several arms. They suggest an online policy that plays Round-robin on an active set of arms. An arm is excluded from this active set if the optimistic projection of its total available reward until the end of the game (which can be computed thanks to the concavity assumption) is lower than the pessimistic projection (when an arm's reward stays constant) of any other arm. They prove no better than a $o(T)$ regret bound (even in the noise-less case) for this algorithm. While they do not provide a lower bound, it suggests that the increasing rested non-stationarity is harder than the decreasing one, where the minimax rate is only in $\mathcal{O}(KL)$.

4.1.3 Levine et al. (2017): wSWA, a first policy for the noisy problem

Sliding-Window Average (SWA)

When the feedback is noisy ($\sigma > 0$), selecting greedily on the last observed reward may be very risky. Indeed, a sample from an optimal pull could be underestimated by $\sim \sigma$. π_G may not pull this good underestimated arm for a long time, because it only estimates the value of the arm with the last sample. This behavior may cause a regret of $\Omega(\sigma T)$ which can be much larger than the noise-free rate $\Theta(KL)$.

Levine et al. (2017) suggested to use the Sliding-Window Average (SWA) policy, a policy which selects the arm with the largest average of its h last sample. Averaging in the presence of noise is a straightforward idea. Yet, it is unclear how the learner should choose h. Before going through the detailed analysis, we give the high-level idea. First, we notice that when $h = 1$, SWA reduces to π_G. Indeed, intuitively, the smaller the noise, the less averaging we need. On the one hand, with a window h, the learner should expect to do $\mathcal{O}(h)$ overpulls for an arm which abruptly decays at $N^\star_{i,T}$ with drop size B. Indeed, its estimator $\widehat{\mu}_i^h$ will be positively biased during the next h pulls. Hence, the learner may suffer up to $\mathcal{O}(KBh)$ due to this bias. On the other hand, the learner takes slightly wrong decisions due to the variance of their estimators $\mathcal{O}(\sigma/\sqrt{h})$ which can cost up to $\widetilde{\mathcal{O}}(\sigma T/\sqrt{h})$ on the long run. Choosing $h = \widetilde{\mathcal{O}}\left((\sigma T/KB)^{2/3}\right)$, we get the regret rate of $\widetilde{\mathcal{O}}\left(B^{1/3}\sigma^{2/3}K^{1/3}T^{2/3}\right)$.

R SWA uses a rested sliding-window mechanism. Indeed, the window of arm i slides only when arm i is selected. Notice the difference with the restless sliding-window of SW-UCB (Garivier and Moulines 2011), which slides for all arms at every round.

Analysis

The analysis of Levine et al. (2017) uses the set of bounded decaying functions instead of $\mathcal{L}_L$.

Algorithm 3 SWA (Levine et al. 2017)

Require: h

for $t \leftarrow 1, 2, \ldots, Kh$ **do**
    PULL ROUND-ROBIN $i_t \leftarrow t\%h$;
    RECEIVE o_t
end for
for $t \leftarrow Kh+1, Kh+2, \ldots$ **do**
    PULL [a] $i_t \in \arg\max_{i \in \mathcal{K}} \widehat{\mu}_i^h$ (see Equation 4.10);
    RECEIVE o_t
end for

[a] One can choose the tie break selection rule arbitrarily, e.g. by selecting the arm with the smallest index.

Definition 4.1.2 Let $\mathscr{B}_{B,x}$, the set of non-increasing reward functions with bounded amplitude B,

$$\mathscr{B}_{B,x} \triangleq \left\{ \mu : \mathbb{N} \to [x, x+B] \;\middle|\; \mu(n) \geq \mu(n+1) \right\}.$$

The choice of origin x is not important. Without loss of generality, we will carry the analysis on $\mathscr{B}_B \triangleq \mathscr{B}_{B,0}$.

R We have that $\mathscr{B}_L \subset \mathscr{L}_L$. Hence, any guarantee of any algorithm on $\mathscr{L}_L$ applies on $\mathscr{B}_B$ by setting $L := B$. We also have that $\mathscr{L}_L \subset \mathscr{B}_{LT,-L(T-1)}$. Hence, any guarantee of any algorithm on $\mathscr{B}_{B,x}$ applies on $\mathscr{L}_L$ by setting $B := LT$.

Estimators For policy π, we define the average of the last h observations of arm i at time t as

$$\widehat{\mu}_i^h(t,\pi) \triangleq \frac{1}{h} \sum_{s=1}^{t-1} \mathbb{1}\left(\pi(s) = i \wedge N_{i,s} > N_{i,t-1} - h\right) o_s \tag{4.10}$$

and the average of the associated means as

$$\overline{\mu}_i^h(t,\pi) \triangleq \frac{1}{h} \sum_{s=1}^{t-1} \mathbb{1}\left(\pi(s) = i \wedge N_{i,s} > N_{i,t-1} - h\right) \mu_i(N_{i,s-1}). \tag{4.11}$$

We notice that $\overline{\mu}_i^h(t,\pi) = \frac{1}{h}\sum_{h'=1}^{h} \mu_i(N_{i,t-1} - h')$ is independent of t and π given $N_{i,t-1}$. Hence, we call $\overline{\mu}_i^h(N_{i,t-1}) \triangleq \overline{\mu}_i^h(t,\pi)$. With a slight abuse of notation, we will also use $\widehat{\mu}_i^h(N_{i,t}) \triangleq \widehat{\mu}_i^h(t,\pi)$. Indeed, the average of the observations depends on the realization of the noise $\{\varepsilon_t\}_t$, hence it is not fully determined by $N_{i,t}$. Yet, these h samples of noise are i.i.d. and thus do not perturb the analysis.

A favorable event

Proposition 4.1.5 For a confidence level $\delta_T \triangleq T^{-2}$, let

$$\xi_{\text{SWA}} \triangleq \left\{ \forall i \in \mathscr{K},\ \forall n \in \{h,\dots,T-1\},\ \left| \widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n) \right| \leq c(h,\delta_T) \right\} \tag{4.12}$$

be the event under which all the possible estimates constructed up to the round T are all accurate up to $c(h,\delta_T) \triangleq \sqrt{2\sigma^2 \log(2/\delta_T)/h}$. Then, for a policy which pulls every arm h times at the beginning (like SWA),

$$\mathbb{P}\left[\overline{\xi_{\text{SWA}}}\right] \leq \frac{K}{T}.$$

Proof. We want to upper bound the probability

$$\mathbb{P}\left[\overline{\xi_{\text{SWA}}}\right] = \mathbb{P}\left[\exists i \in \mathscr{K}, \exists n \in \{h,\dots,T-1\}, \left|\widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n)\right| > c(h,\delta_T)\right].$$

For $N_{i,t-1} = n$, we have that,

$$\widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n) = \frac{1}{h}\sum_{s=1}^{T} \mathbb{1}\left(i_s = i \,|\, n-h < N_{i,s} \leq n\right) \varepsilon_s .$$

By Doob's optional skipping (e.g. see Chow and Teicher (1997), Section 5.3) there exists a sequence of random independent variables $(\varepsilon'_l)_{l\in\mathbb{N}}$, σ^2 sub-gaussian such that

$$\widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n) = \frac{1}{h}\sum_{s=1}^{T-1} \mathbb{1}\left(i_s = i \,|\, N_{i,s} > n-h\right) \varepsilon_s = \frac{1}{h}\sum_{l=n-h+1}^{n} \varepsilon'_l \triangleq \widehat{\varepsilon}_n^h.$$

Hence,

$$\begin{aligned}
&\mathbb{P}\left[\exists i \in \mathscr{K}, \exists n \in \{h,\dots,T-1\}, \left|\widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n)\right| > c(h,\delta_T)\right] \\
&\quad = \mathbb{P}\left[\exists i \in \mathscr{K}, \exists n \in \{h,\dots,T-1\}, |\widehat{\varepsilon}_n^h| > c(h,\delta_T)\right] \\
&\quad \leq \sum_{i\in\mathscr{K}} \sum_{n=h}^{T-1} \mathbb{P}\left[|\widehat{\varepsilon}_n^h| > c(h,\delta_T)\right] \\
&\quad \leq KT\delta_T \\
&\quad \leq \frac{K}{T},
\end{aligned}$$

where we used the Chernoff inequality at the before last line and $\delta_T = T^{-2}$ at the last one. ■

Regret upper-bound

Proposition 4.1.6 — Levine et al. (2017). For a problem $\mu \in \mathscr{B}_B^K$, the expected regret of SWA tuned with h is bounded as,

$$\mathbb{E}\left[R_T(\pi_{\text{SWA}})\right] \leq 4\sigma T \cdot \sqrt{\frac{\log\left(\sqrt{2}T\right)}{h}} + K(h+1)B$$

Proof. If $T \leq Kh$, we can bound the regret by the maximum regret (T errors of magnitude B)

$$\mathbb{E}\left[R_T(\pi_{\text{SWA}})\right] \leq TB \leq KhB \leq 4\sigma T \cdot \sqrt{\frac{\log\left(\sqrt{2}T\right)}{h}} + K(h+1)B.$$

If $T > Kh$, we notice that any arm i is pulled at least h times, *i.e.* $N_{i,T} \geq h$. We split the regret on the events ξ_{SWA} and $\overline{\xi_{\text{SWA}}}$,

$$\mathbb{E}\left[R_T(\pi_{\text{SWA}})\right] \leq \mathbb{E}\left[\mathbb{1}\left[\xi_{\text{SWA}}\right] R_T(\pi_{\text{SWA}})\right] + \mathbb{E}\left[\mathbb{1}\left[\overline{\xi_{\text{SWA}}}\right] R_T(\pi_{\text{SWA}})\right].$$

The regret on the unfavorable event $\mathbb{1}\left[\overline{\xi_{\text{SWA}}}\right]$ can be bounded by the maximal regret BT (since $\mu \in \mathscr{B}_B^K$),

$$\mathbb{E}\left[R_T(\pi_{\text{SWA}})\right] \leq \mathbb{E}\left[\mathbb{1}\left[\xi_{\text{SWA}}\right] R_T(\pi_{\text{SWA}})\right] + \mathbb{P}\left[\overline{\xi_{\text{SWA}}}\right] BT.$$

Using Proposition 4.1.5, we get,

$$\mathbb{E}\left[R_T(\pi_{\text{SWA}})\right] \leq \mathbb{E}\left[\mathbb{1}\left[\xi_{\text{SWA}}\right] R_T(\pi_{\text{SWA}})\right] + KB. \tag{4.13}$$

We will now bound the regret on the favorable event,

$$R_T(\pi_{\text{SWA}}|\xi_{\text{SWA}}) \triangleq \mathbb{1}\left[\xi_{\text{SWA}}\right] R_T(\pi_{\text{SWA}})$$

We start from Equation 4.5 applied to policy `SWA`,

$$R_T(\pi_{\text{SWA}}|\xi_{\text{SWA}}) \leq \mathbb{1}\left[\xi_{\text{SWA}}\right] \sum_{i \in \text{OP}} \sum_{n=N_{i,T}^\star}^{N_{i,T}-1} \left(\mu_T^+(\pi_{\text{SWA}}) - \mu_i(n)\right). \tag{4.14}$$

The remaining of the proof is similar to the proof of Proposition 4.1.3 about algorithm π_{G}. Instead of showing that the before last terms in the sums are equals to zeros, we will show that the terms before the h last one cost less than $2c(h,\delta_T)$. Let $i \in \mathscr{K}$ an arm which is pulled at least $h+1$ times at the end of the game $N_{i,T} \geq h+1$. We call $t_i \triangleq \min\{t \leq T \mid N_{i,t} = N_{i,T}\}$ the last round at which i is pulled. Notice that $t_i > Kh$ because $N_{i,T} > h$ and the Kh first pulls corresponds to the round-robin period. Hence, for any arm $j \in \mathscr{K}$, $N_{j,t_i-1} \geq h$. For all $n_i \leq N_{i,T} - (h+1)$,

$$\begin{aligned}
\mu_i(n_i) &\geq \mu_i(N_{i,T} - (h+1)) \\
&\geq \overline{\mu}_i^h(N_{i,t_i-1}) \\
&\geq \widehat{\mu}_i^h(N_{i,t_i-1}) - c(h,\delta_T) \\
&\geq \widehat{\mu}_j^h(N_{j,t_i-1}) - c(h,\delta_T) \\
&\geq \overline{\mu}_j^h(N_{j,t_i-1}) - 2c(h,\delta_T).
\end{aligned} \tag{4.15}$$

The first inequality follows by the non-increasing hypothesis on the reward function. The second inequality is because $\overline{\mu}_i^h(N_{i,t_i-1})$ is the average of h reward sample of arm i after

the $N_{i,T}-(h+1)$-th (according to the definition of t_i). The third and fifth one use the concentration of all the constructed estimates on the event ξ_{SWA}. The fourth inequality follows by definition of the policy: at time t_i, π_{SWA} selects $i \in \arg\max_{j\in\mathscr{K}} \widehat{\mu}_j^h(N_{j,t_i-1})$.

We choose j such that $\mu_j(N_{j,T}) = \mu_T^+(\pi_{\text{SWA}})$ $\left(\triangleq \max_{j'\in\mathscr{K}} \mu_{j'}(N_{j',T})\right)$.
Since $t_i \leq T$, by the rotting assumption,

$$\overline{\mu}_j^h(N_{j,t_i-1}) \geq \mu_j(N_{j,T}) = \mu_T^+(\pi_{\text{SWA}}). \tag{4.16}$$

Gathering Equations 4.15 and 4.16, we have that

$$\forall n_i \leq N_{i,T}-(h+1)\,,\ \left(\mu_T^+(\pi_{\text{SWA}}) - \mu_i(n_i)\right) \leq 2c(h,\delta_T). \tag{4.17}$$

Therefore, in Equation 4.14, we can split the sum on $N_{i,T}-h$. Hence,

$$\begin{aligned}
R_T(\pi_{\text{SWA}}|\xi_{\text{SWA}}) \leq& \mathbb{1}\left[\xi_{\text{SWA}}\right] \sum_{i\in\text{OP}} \sum_{n=N^\star_{i,T}}^{N_{i,T}-1} \left(\mu_T^+(\pi_{\text{SWA}}) - \mu_i(n)\right) \\
=& \mathbb{1}\left[\xi_{\text{SWA}}\right] \sum_{i\in\text{OP}} \sum_{n=N^\star_{i,T}}^{N_{i,T}-(h+1)} \left(\mu_T^+(\pi_{\text{SWA}}) - \mu_i(n)\right) \\
&+ \mathbb{1}\left[\xi_{\text{SWA}}\right] \sum_{i\in\text{OP}} \sum_{n=N_{i,T}-h}^{N_{i,T}-1} \left(\mu_T^+(\pi_{\text{SWA}}) - \mu_i(n)\right) \\
\leq& 2Tc(h,\delta_T) + KhB.
\end{aligned} \tag{4.18}$$

In the last inequality, we used Equation 4.17 and that there is less than T overpulls in the first sums. We also use $\mu \in \mathscr{B}_B$ to bound each term in the second sums by B. Finally, we can conclude by plugging Equation 4.18 in Equation 4.13 and by using the definition of $c(h,\delta_T)$ and $\delta_T = T^{-2}$ in Proposition 4.1.5,

$$\mathbb{E}\left[R_T(\pi_{\text{SWA}})\right] \leq 4\sigma T \cdot \sqrt{\frac{\log\left(\sqrt{2}T\right)}{h}} + K(h+1)B$$

■

Corollary 4.1.7 — Levine et al. (2017). Let C such that $h := \left\lceil C\left(\frac{\sigma T}{KB}\right)^{2/3} \log\left(\sqrt{2}T\right)^{1/3}\right\rceil$.
Then, for reward functions in $\mathscr{B}_B$,

$$R_T(\pi_{\text{SWA}}) \leq \left(\frac{4}{\sqrt{C}} + C\right)\left(\sigma^2 BKT^2 \log\left(\sqrt{2}T\right)\right)^{1/3} + 2KB.$$

Hence, if the learner knows T and the ratio σ/B, s/he can set $h := \left\lceil \left(\frac{2\sigma T}{KB}\right)^{2/3} \log\left(\sqrt{2}T\right)^{1/3}\right\rceil$

(*i.e.* $C = 2^{2/3}$) and be guaranteed to perform,

$$R_T(\pi_{\text{SWA}}) \leq 5\left(\sigma^2 BKT^2 \log\left(\sqrt{2}T\right)\right)^{1/3} + 2KB.$$

Doubling trick Wrapper for SWA (`wSWA`): an anytime algorithm.

The theoretical window choice requires the knowledge of the horizon T, the subgaussian parameter σ, and the reward range B (or at least the ratio σ/B). Levine et al. (2017) suggest `wSWA` (Alg. 4), which wraps `SWA` with the doubling trick. The algorithm is initialized with a first (small) guess of the horizon. When the horizon is reached, the algorithm is fully reinitialized and restarted with a doubled horizon. This is a classic trick in the literature: it is known to recover the problem-independent rate of a given algorithm (with a worse constant factor), but the empirical performance is often significantly reduced (Besson and Kaufmann 2018). In the case of `wSWA`, the doubling trick erases all the history $\mathbf{H}_t$ and increases the window.

Algorithm 4 `wSWA` (Levine et al. 2017)

Require: α, σ, $T \leftarrow 1$

$h \leftarrow \left\lceil \alpha \left(\frac{4\sigma T}{K}\right)^{2/3} \left(\log\left(\sqrt{2}T\right)\right)^{1/3} \right\rceil$

for $t \leftarrow 1, 2, \ldots, T$ **do**

    RUN `SWA` (h)

end for

CLEAN `SWA`'s MEMORY

`wSWA` (α, σ, $2T$)

Corollary 4.1.8 — Levine et al. (2017). The regret of `wSWA` tuned with α can be bounded by,

$$R_T(\pi_{\text{wSWA}}) \leq 8\left(\alpha B + \frac{1}{\sqrt{\alpha}}\right)\left(\sigma^2 KT^2 \log\left(\sqrt{2}T\right)\right)^{1/3} + 3KB\left(\log_2(T) + 1\right).$$

The best theoretical tuning corresponds to $\alpha := (2B)^{-2/3}$ which necessitates the prior knowledge of B. In their experimental section, Levine et al. (2017) select $\alpha := 0.2$ by grid-search on one problem. Yet, the reader should not forget that the tuning of α depends on the rotting magnitude B.

4.1.4 Experimental benchmarks

Simulated benchmark #1: Impact of B (2 arms).

Experiments We consider rotting bandits with two arms defined as,

$$\mu_1(n) = 0, \ \forall n \leq T \quad \text{and} \quad \mu_2(n) = \begin{cases} \frac{L}{2} & \text{if } n < \frac{T}{4}, \\ -\frac{L}{2} & \text{if } n \geq \frac{T}{4}. \end{cases}$$

The rewards are then generated by applying a gaussian i.i.d. noise $\mathcal{N}(0, \sigma = 1)$. The optimal allocation for this two-armed setting is $N^\star_{1,T} = \lceil 3T/4 \rceil$ and $N^\star_{2,T} = \lfloor T/4 \rfloor$. The reward lies on a bounded interval of size $B := L$. In this specific case, L also defines the gap between the arms $\Delta = |\mu_1(n_1) - \mu_2(n_2)| = L/2$, which is known to heavily impact the performance in stochastic bandits. We set $T = 10000$ and we consider 30 different values of L dispatched on a geometric grid between $[0.02, 20]$.

Algorithms We compare `wSWA` tuned with three different values of parameter $\alpha \in \{0.002, 0.02, 0.2\}$, including the recommendation of Levine et al. (2017), $\alpha = 0.2$. We remind the reader that α is a multiplicative constant to tune the averaging window. The window h is increased at each restart, and reaches the values $\{3, 28, 272\}$ (for the three values of α) at the horizon $T = 10000$. In general, the smaller the α, the smaller the averaging window and the more reactive the algorithm is to large drops. Nonetheless, a small α increases the variance of the sliding window indexes. Thus, the regret increases in stationary regimes where gaps between arms are small compared to the variance of these indexes. On the other hand, a large value of α may reduce variance but increase the bias in the case of rapidly rotting arms.

Results In Fig. 4.1, we compare the performance of the three versions of `wSWA`. The top plot shows the regret at the last round $T = 10000$ for 30 different values of L. The bottom plots show the regret as a function of time for $L = 0.233$ and $L = 4.24$.

Each curve of the top plot has three different parts. First, we observe a linear increase for small values of L (exponential shape on the semi-log plot). Indeed, when $\Delta = L/2 \lesssim \sigma/\sqrt{h}$, the variance of the indexes are greater than the gap between arms. Therefore, the algorithms are unable to consistently choose the good arm and they do $\mathcal{O}(T)$ mistakes of size Δ. Hence, the regret grows linearly with $\mathcal{O}(T\Delta)$ and ultimately with L.

Then, the regret stagnates (red curve) or sharply decreases (green and blue curve). When $\Delta \gtrsim \sigma/\sqrt{h}$, the variance of the indexes is smaller than the gap between arms. Hence, the number of mistakes decrease at an exponential rate with L^2 (according to Hoeffding inequality) from $\mathcal{O}(T)$ to $\widetilde{\mathcal{O}}(h)$. Notice that there is indeed a factor $\sqrt{\alpha_{blue}/\alpha_{green}} \sim 3$ between the x-coordinate of the green and blue peaks: it matches the order of magnitude of $L_{peak} \sim \sigma/\sqrt{h}$.

Yet, in this setup, the concentration of the index can only reduce the number of overpulls of arm 2 up to $\sim h/2$. Indeed, the expected value of the index of arm 2 is larger than the

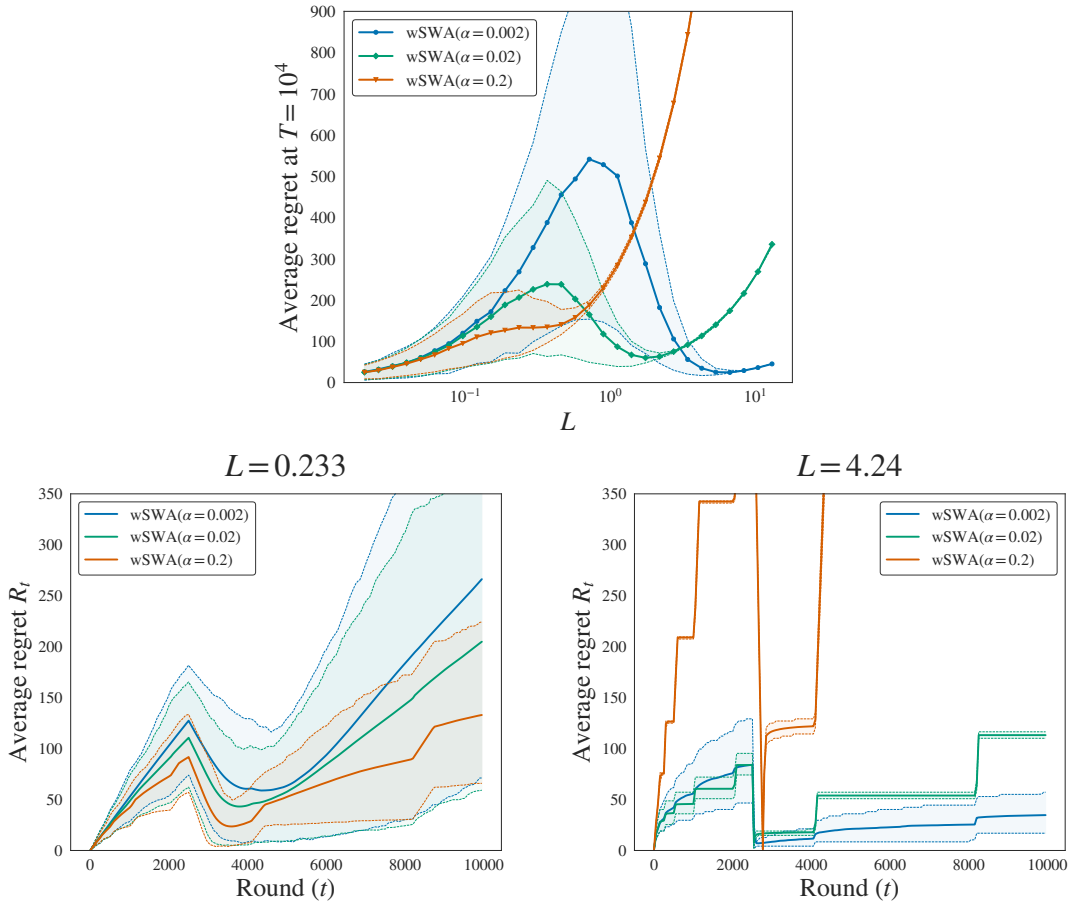


Figure 4.1: **Top:** Regret at the end of the game for different values of L. **Bottom:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region.

expected value of the index of arm 1 until we do $h/2$ overpulls of arm 2 because of the bias caused by the pulls before the breakpoint. Moreover, at each restart of the algorithm, both arms are pulled h times. Thus, the regret is lower-bounded by $\widetilde{\mathcal{O}}(Lh)$ for any L due to the doubling trick restart and the bias of the index. That is why we observe a linear increase (exponential shape on the semi-log plot) of the regret at the end of the green and red curves. Notice that there is a factor 10 between the red and green curves for large values of L, which confirms the $\mathcal{O}(Lh)$ regret rate. We highlight that the red curve does not decrease because this exponential increase takes over the decrease due to the concentration of the index.

The bottom plots show the evolution of the regret for two different values of L. We notice that the regret first increases, then decreases at $t = \lceil T/4 \rceil$ and increase again. The regret decreases because at $t = \lceil T/4 \rceil$ the arms' value for the optimal policy are 0 and $-L/2$ while any sub-optimal policy can obtain either 0 or $+L/2$. Therefore, the regret cannot increase until `wSWA` has pulled arm 2 for $\lfloor T/4 \rfloor$ times. At this round, the regret is 0 because we are at the optimal pulling allocation. Notice that we display the expected regret, which might not be equal to 0 because the different runs do not reach 0 at the same round. After that,

the regret increases again as wSWA may select arm 2 with sub-optimal value $-L/2$.

For $L = 0.233$, $\alpha = 0.2$ is the best tuning. Indeed, the difference between arms is only of $\Delta = 0.1$ which is small compared to $\sigma = 1$. Therefore, we need a reasonably large averaging window to decrease the variance of the indexes below the gaps between arms, *i.e.* $h \sim \sigma^2/\Delta^2 \sim 100$ which is the order of magnitude when $\alpha = 0.2$ ($h = 272$ at the end of the game). For $L = 4.24$, $\alpha = 0.002$ is the best tuning. Indeed, following the same reasoning, we need $h \sim 1$ which is coherent with $h = 3$ at the end of the game when $\alpha = 0.002$.

At each full restart due to the doubling trick wrapper, the two arms are pulled h times which generates $hL/2$ extra regret. The cost of this operation is particularly prohibitive when either h or L is large. For instance, when $L = 4.24$, we see periodic sharp increments in the regret when $t = 2^i$, especially for the larger values of α.

Simulated benchmark #2: Learning against several drops (10 arms)

Experiments. We also tested a rotting setting with 10 arms. The mean of 1 arm is constant with value 0 while the means of 9 arms abruptly decrease after 1000 pulls from $+\Delta_i$ to $-\Delta_i$. We use nine different values of Δ_i which are ranging from 0.001 to 10 in a geometric sequence. In this setting, the regret can be written as $R_T(\pi) = \sum_{i=1}^{9} h_{i,T}\Delta_i$, with $h_{i,T}$ the number of overpulls of arm i at round T. We define the regret per arm, $R_T^i(\pi) \triangleq \Delta_i h_{i,T}$.

Algorithms. We keep the three versions of wSWA with $\alpha \in \{0.002, 0.02, 0, 2\}$. We add to our benchmark famous stationary, adversarial and non-stationary algorithms: UCB (T. L. Lai and Robbins 1985), Exp3, Exp3.S (Auer et al. 2003), D-UCB, SW-UCB (Garivier and Moulines 2011) and GLR-UCB (Besson and Kaufmann 2019).

For UCB we use the asymptotic optimal tuning of the confidence bounds for stationary gaussian bandits. For Exp3 and Exp3.S, we use theoretical tuning using the number of breakpoints and the horizon. For SW-UCB and D-UCB, we select the forgetting parameters $\tau = 200$ and $\gamma = 0.997$ with grid-search for best performance on this problem. For GLR-UCB, we use the theoretical value $\delta = \sqrt{T}^{-1}$ for the change-point sensitivity and we set the probability of random exploration to 0. Indeed, the random exploration is used to detect (restless) increment in the sub-optimal arms value which is irrelevant for our rested rotting setup.

Results. We display the average regret through the rounds and the regret per arm at the end of the horizon in Figure 4.2.

We do not display the results for UCB and Exp3 because they obtain very large regret after the first drop in the reward (20 000 at the end of the game). Indeed, these two algorithms are designed for the fixed-arm regret and are unable to adapt to change of the best arm identity.

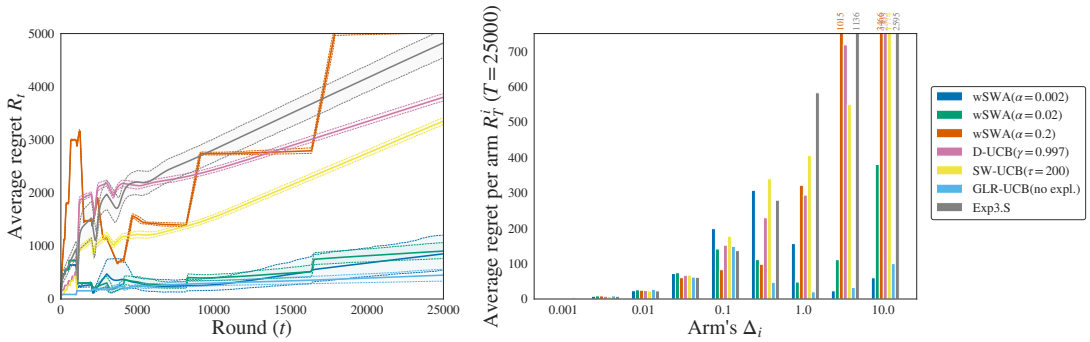


Figure 4.2: **Left:** Regret at the end of the game for different values of L. **Middle, Right:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region.

`SW-UCB`, `D-UCB`, and `Exp3.S` show large regret even though `SW-UCB` and `D-UCB` were optimally tuned for this problem. These algorithms use random exploration and/or restless forgetting of the data associated with each arm. This is harmful because it leads to multiple pulls of a very suboptimal arm. Yet, with the rested rotting non-stationarity, an identified bad arm has no reason to improve.

`SWA` shows better performance when it is rightly tuned. In this problem, we have multiple sizes of drops and one should choose a value α which trades-off between these different sizes. $\alpha = 0.2$, which obtains the best value in Levine et al. (2017)'s experiment, has a very large regret in our benchmark. Indeed, the best tuning depends on the maximal size of the drops, which is quite large in our setting. We also notice the cost of the doubling trick.

Last, `GLR-UCB` shows good performance when random exploration is turned off. Indeed, the change detection mechanism reset the history of the arm when there is significant evidence of a change. Hence, the number of mistakes after a drop is adaptive to the difficulty to detect the change.

4.1.5 Open problems

Minimax rate

We report existing regret bounds for two special cases. First, in Proposition 4.1.4, Heidari et al. (2016) show that in the absence of noise, the regret is lower bounded by $\mathcal{O}(KL)$. Second, we recall the minimax regret lower bound for stochastic stationary bandits.

Proposition 4.1.9 — Auer et al. 2003. [Thm. 5.1] For any learning policy π and any horizon T, there exists a stochastic stationary problem $\left\{\mu_i(n) \triangleq \mu_i\right\}_i$ with K σ-sub-gaussian arms such that π suffers a regret

$$\mathbb{E}[R_T(\pi)] \geq \frac{\sigma}{10} \min\left(\sqrt{KT}, T\right).$$

where the expectation is w.r.t. both the randomization over rewards and algorithm's internal randomization.

Any problem in the two settings above is a rotting problem with parameters (σ, L). Therefore, the performance of any algorithm on the noisy rotting problem is also bounded by these two lower bounds. For reward functions in $\mathscr{B}_B$, SWA is guaranteed to achieve $\mathscr{O}\left(T^{2/3}\right)$ regret rate. Yet, Levine et al. (2017) do not provide a lower bound while they suggest it could be an interesting future work direction.

Problem-dependent rate

SWA starts by pulling every arm h times. It means that even for simple stationary problem with large difference $\Delta_i > \sigma$ between suboptimal and optimal arms, SWA makes at least $h = \mathscr{O}\left(T^{2/3}\right)$ mistakes per suboptimal arms which is much more than the stationary asymptotic optimal pulling rate $\mathscr{O}\left(\sigma \log(T)/\Delta_i^2\right)$.

Agnostic algorithm

SWA requires the knowledge of the horizon T, the subgaussian parameter σ, and the reward range B to tune the window h. We showed empirically that the doubling trick leads to large regret increases at each restart. We also show that not knowing the amplitude of the drops B could lead to very suboptimal tuning.

R Levine et al. (2017) suggest in wSWA to use the classical doubling trick, with a full restart of the memory of the algorithm. It is an easy way to generalize the $\widetilde{\mathscr{O}}\left(T^{2/3}\right)$ bound when we do not know T. However, in practice, in this rested setup, there is no good reason to clean the memory of wSWA (see Line 5). We could simply increase the window h and keep the current history of the arms in order to diminish the cost of the restart. The empirical investigation of this algorithm showed improved results compared to wSWA without completely removing the extra cost of the doubling trick.

Global budget or Budget per round

The analysis of SWA was carried in the global rotting budget setting while the analysis of the noiseless case was carried in the per round budget setting. We can translate the $\widetilde{\mathscr{O}}\left(B^{1/3}T^{2/3}\right)$ bound by setting $B = LT$ which leads to linear regret (see the remark following Definition 4.1.2). Hence, no algorithm is proved to achieve a $o(T)$ regret bound in the rotting budget per round setting with noise.

4.2 FEWA and RAW-UCB: Two adaptive window algorithms

4.2.1 Towards adaptive windows

Since the expected rewards μ_i change from one pull to another, the main difficulty in the rested rotting bandits is that we cannot rely on all samples observed until time t to predict which arm is likely to return the highest reward in the future. In fact, the older a sample, the less representative it is for future rewards. This suggests constructing estimates using more recent samples. Nonetheless, discarding older rewards reduces the number of samples used in the estimates, thus increasing their variance.

In Figure 4.1, we showed different setups in which the regret of wSWA scales either with $\mathcal{O}(KLh)$ (for large L) or with $\widetilde{\mathcal{O}}(T\sigma/\sqrt{h})$ (for small L). wSWA chooses a window h which balances between these two costs. Yet, the two situations are quite different. In Figure 4.3, we show three different reward functions with the associated data. The first one has a large decay $L > \sigma$ at the end of the sequence. The second one has a rather small decay in the middle. The last one is stationary. For these three arms, we should probably not use the same window to estimate the three values.

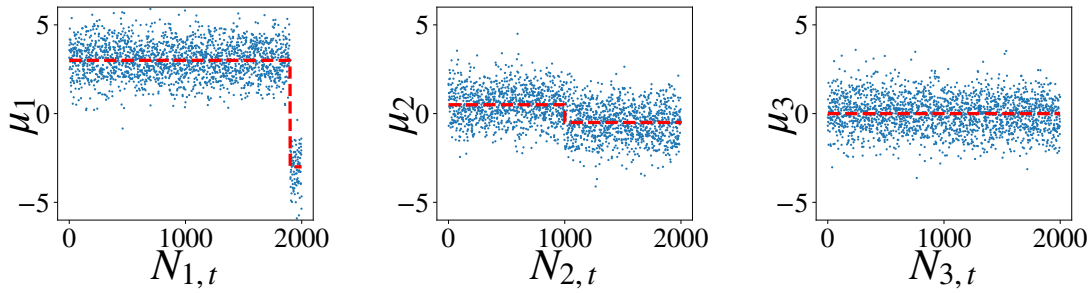


Figure 4.3: Three rotting reward functions (red dash line) and associated reward samples: Why should we use a single fixed window size to compare these three arms?

A favorable event for adaptive windows

Proposition 4.2.1 For any round t and confidence $\delta_t \triangleq 2t^{-\alpha}$, let

$$\xi_t^\alpha \triangleq \left\{ \forall i \in \mathcal{K},\ \forall n \leq t-1,\ \forall h \leq n, \left| \widehat{\mu}_i^h(t,\pi) - \overline{\mu}_i^h(t,\pi) \right| \leq c(h,\delta_t) \right\} \tag{4.19}$$

be the event under which the estimates at a round t are all accurate up to $c(h,\delta_t) \triangleq \sqrt{2\sigma^2 \log(2/\delta_t)/h}$. Then, for a policy π which pulls each arms once at the beginning, and for all $t > K$,

$$\mathbb{P}\left[\overline{\xi_t^\alpha}\right] \leq \frac{Kt^2\delta_t}{2} = Kt^{2-\alpha}.$$

Proof. We want to upper bound the probability

$$\mathbb{P}\left[\overline{\xi_t^\alpha}\right] = \mathbb{P}\left[\exists i \in K, \exists n \leq t-1, \exists h \leq n, \left| \widehat{\mu}_i^h(n) - \overline{\mu}_i^h(n) \right| > c(h,\delta_t)\right].$$

Following the same argument as in Proposition 4.1.5, there exists a sequence of random independent variable $(\varepsilon'_l)_{l\in\mathbb{N}}$, σ^2 sub-Gaussian such that for $\widehat{\varepsilon}^h_n \triangleq (1/h)\sum_{l=n-h+1}^{n}\varepsilon'_l$ we get,

$$\begin{aligned}
\mathbb{P}\Big[\exists n \le t-1, \exists h \le n, \big|\widehat{\mu}^h_i(t-1,\pi) - \overline{\mu}^h_i(t-1,\pi)\big| > c(h,\delta_t)\Big] \\
&= \mathbb{P}\Big[\exists n \le t-1, \exists h \le n, |\widehat{\varepsilon}^h_n| > c(h,\delta_t)\Big] \\
&\le \sum_{n=1}^{t-1}\sum_{h=1}^{n}\mathbb{P}\Big[|\widehat{\varepsilon}^h_n| > c(h,\delta_t)\Big] \\
&\le \frac{t(t-1)}{2}\cdot\delta_t\,,
\end{aligned}$$

where we used the Chernoff inequality in the last line. Thus, a union bound over the arms allows us to conclude that

$$\mathbb{P}\Big[\overline{\xi^{\alpha}_t}\Big] \le \frac{K\delta_t t^2}{2}.$$

■

R Compared to the unique favorable event we used for SWA (see Equation 4.12), we use a favorable event for each round t. It will be helpful to obtain anytime guarantees for our algorithms. Moreover, ξ^{α}_t control the deviation of any statistic $\widehat{\mu}^h_i(n)$ for any possible h, i and n. This is different from ξ_{SWA} which uses a fixed h.

4.2.2 FEWA: Filtering on expanding window average

In Alg. 5, we introduce FEWA (or π_{F}) that at each round t, relies on estimates using windows of increasing length to filter out arms that are sub-optimal with high probability and then pulls the least pulled arm among the remaining arms.

We first describe the subroutine FILTER in Alg. 6, which receives a set of active arms $\mathscr{K}_h$, a window h, a confidence bound tuning parameter α and the subgaussian parameter σ as input and returns an updated set of arms $\mathscr{K}_{h+1}$. For each arm $i \in \mathscr{K}_h$ (that has all been pulled $n \ge h$ times), the algorithm has stored an estimate $\widehat{\mu}^h_i$ that averages the h most recent rewards observed from i. The subroutine FILTER discards all the arms whose mean estimate (built with window h) from $\mathscr{K}_h$ is lower than the empirically best arm by more than twice a threshold $c(h,\delta_t)$ constructed by standard Hoeffding's concentration inequality (see Prop. 4.2.1).

The FILTER subroutine is used in FEWA to incrementally refine the set of active arms, starting with a window of size 1, until the condition at Line 13 is met. As a result, $\mathscr{K}_{h+1}$ only contains arms that passed the filter for all windows from 1 up to h. Notice that it is important to start filtering arms from a small window and to keep refining the previous set of active arms. In fact, the estimates constructed using a small window use recent rewards,

Algorithm 5 FEWA

Require: $\mathscr{K}$, σ, α

1: **for** $t \leftarrow 1, 2, \ldots, K$ **do** ▷ *Pull each arm once*
2: PULL $i_t \leftarrow t$; RECEIVE o_t
3: $N_{i_t} \leftarrow 1$
4: $\left\{\widehat{\mu}_{i_t}^h\right\}_h \leftarrow \texttt{UPDATE}(\left\{\widehat{\mu}_{i_t}^h\right\}_h, o_t)$
5: **end for**
6: **for** $t \leftarrow K+1, K+2, \ldots$ **do**
7: $h \leftarrow 1$ ▷ *initialize bandwidth*
8: $\mathscr{K}_1 \leftarrow \mathscr{K}$ ▷ *initialize with all the arms*
9: $i_t \leftarrow \texttt{none}$
10: **while** i_t is none **do**
11: $\mathscr{K}_{h+1} \leftarrow \texttt{FILTER}(\mathscr{K}_h, h, \alpha, \sigma, t)$
12: $h \leftarrow h+1$
13: **if** $\exists i \in \mathscr{K}_h$ such that $N_i = h$ **then**
14: PULL[a] $i_t \in \{i \in \mathscr{K}_h | N_{i_t} = h\}$; RECEIVE o_t
15: **end if**
16: **end while**
17: $N_{i_t} \leftarrow N_{i_t} + 1$
18: $\left\{\widehat{\mu}_{i_t}^h\right\}_h \leftarrow \texttt{UPDATE}(\left\{\widehat{\mu}_{i_t}^h\right\}_h, o_t)$
19: **end for**

[a] One can choose the tie break selection rule arbitrarily, e.g. by selecting the arm with the smallest index.

Algorithm 6 FILTER

Require: $\mathscr{K}_h$, h, α, σ, t

$c(h, \delta_t) \leftarrow \sqrt{2\alpha\sigma^2 \log(t)/h}$
$\widehat{\mu}_{\max}^h \leftarrow \max_{i \in \mathscr{K}_h} \widehat{\mu}_i^h$
for $i \in \mathscr{K}_h$ **do**
$\Delta_i \leftarrow \widehat{\mu}_{\max}^h - \widehat{\mu}_i^h$
if $\Delta_i \leq 2c(h, \delta_t)$ **then**
add i to $\mathscr{K}_{h+1}$
end if
end for

Ensure: $\mathscr{K}_{h+1}$

which are closer to the future value of an arm. As a result, if there is enough evidence that an arm is suboptimal already at a small window h, it should be directly discarded. On the other hand, a sub-optimal arm may pass the filter for small windows as the threshold $c(h, \delta_t)$ is large for small h (i.e., because a few samples are used in constructing $\widehat{\mu}_i^h$, the estimation error may be high). Thus, FEWA keeps refining $\mathscr{K}_h$ for larger windows in the attempt of constructing more accurate estimates and discard more sub-optimal arms. This process stops when we reach a window as large as the number of samples for at least one arm in the active set $\mathscr{K}_h$ (i.e., Line 13). At this point, increasing h would not bring any additional evidence that could refine $\mathscr{K}_h$ further (recall that $\widehat{\mu}_i^h$ is not defined for $h > N_i$).

Finally, FEWA selects the active arm i_t whose number of samples matches the current window, i.e., the least pulled arm in $\mathscr{K}_h$. The set of available rewards and the number of pulls are then updated accordingly.

Core guarantee on the favorable event We derive an important lemma that provides support for the arm selection process obtained by a series of refinements through the FILTER subroutine. Recall that at any round t, after pulling arms $\{N_{i,t-1}\}_i$ the greedy (oracle) policy would select an arm

$$i_t^\star\left(\left\{N_{i,t-1}\right\}_i\right) \in \arg\max_{i\in\mathscr{K}} \mu_i\left(N_{i,t-1}\right).$$

We recall that $\mu_t^+(\pi_{\mathrm{F}}) \triangleq \max_{i\in\mathscr{K}} \mu_i(N_{i,t-1})$ the reward that could be obtained by pulling $i_t^\star$ at a round t. While FEWA cannot directly match the performance of the oracle arm, the following lemma guarantees that the past performance of the selected arm is close enough compared to the current best arm value.

Lemma 4.2.2 For FEWA tuned with α, on the favorable event ξ_t^α, if an arm i passes through a filter of window h at a round t, i.e., $i \in \mathscr{K}_h$, then the average of its h last pulls satisfies

$$\overline{\mu}_i^h(N_{i,t-1}) \geq \mu_t^+(\pi_{\mathrm{F}}) - 4c(h,\delta_t). \tag{4.20}$$

Therefore, at a round t, on favorable event ξ_t^α, if arm i is selected by FEWA (α), for any $h \leq N_{i,t-1}$, the average of its h last pulls cannot deviate significantly from the best available arm at that round, i.e.,

$$\overline{\mu}_{i_t}^h(N_{i,t-1}) \geq \mu_t^+(\pi_{\mathrm{F}}) - 4c(h,\delta_t).$$

Proof. We will prove this property for a more general rotting feedback model than the rested rotting one presented in Equation 4.1. We will use this more general proof in the next Chapter 5. If arm i is selected at the round t, the learner π receives,

$$o_t \triangleq \mu_{i,t} + \varepsilon_t \;\text{ with } \mathbb{E}\left[\varepsilon_t|\mathscr{H}_t\right] = 0 \;\text{ and } \forall\lambda\in\mathbb{R},\; \mathbb{E}\left[e^{\lambda\varepsilon_t}\right] \leq e^{\frac{\sigma\lambda^2}{2}},$$

with $\{\mu_{i,t}\}_{t\leq T}$ a non-increasing sequence. We do not specify how the reward is rotting, while it was assumed in Equation 4.1 that the reward function was evolving with the number of pulls $N_{i,t-1}$ of arm i at the round t. With this reward model, we cannot use $\overline{\mu}_i^h(N_{i,t-1}^\pi)$ to refer to $\overline{\mu}_i^h(t,\pi)$, the average of the h last means associated to arm i (see the definition in Equation 4.11 and the following remark). We also extend the definitions of $i_t^\star \in \arg\max_{i\in\mathscr{K}} \mu_{i,t}$ and $\mu_t^+ \triangleq \max_{i\in\mathscr{K}} \mu_i(N_{i,t-1})$.

Let $i \in \mathscr{K}_h$ be an arm that passed a filter of window h at the round t. First, we use the confidence bound for the estimates and we pay the cost of keeping all the arms up to a distance $2c(h,\delta_t)$ of $\widehat{\mu}_{\max,t}^h \triangleq \max_{j\in\mathscr{K}_h} \widehat{\mu}_i^h(t,\pi_{\mathrm{F}})$,

$$\overline{\mu}_i^h(t,\pi_{\mathrm{F}}) \geq \widehat{\mu}_i^h(t,\pi_{\mathrm{F}}) - c(h,\delta_t) \geq \widehat{\mu}_{\max,t}^h - 3c(h,\delta_t) \geq \max_{j\in\mathscr{K}_h} \overline{\mu}_j^h(t,\pi_{\mathrm{F}}) - 4c(h,\delta_t), \tag{4.21}$$

where in the last inequality, we used that for all $j \in \mathscr{K}_h$,

$$\widehat{\mu}^h_{\max,t} \geq \widehat{\mu}^h_j(t,\pi_F) \geq \overline{\mu}^h_j(t,\pi_F) - c(h,\delta_t).$$

Second, we call $t_{i,t} < t$ the last round at which arm i was selected. Since the means of arms are decaying, we know that

$$\begin{aligned}\mu^+_t(\pi_F) &\triangleq \mu_{i^\star_t,t} \\ &\leq \mu_{i^\star_t,t_{i,t}} = \overline{\mu}^1_{i^\star_t}(t,\pi_F) \\ &\leq \max_{j\in\mathscr{K}} \overline{\mu}^1_j(t,\pi_F) = \max_{j\in\mathscr{K}_1} \overline{\mu}^1_j(t,\pi_F). \end{aligned} \tag{4.22}$$

Third, we show that the largest average of the last h' means of arms in $\mathscr{K}_{h'}$ is increasing with h',

$$\forall h' \leq h, \max_{j\in\mathscr{K}_{h'+1}} \overline{\mu}^{h'+1}_j(t,\pi_F) \geq \max_{j\in\mathscr{K}_{h'}} \overline{\mu}^{h'}_j(t,\pi_F).$$

To show the above property, we remark that thanks to our selection rule, the arm that has the largest average of means, always passes the filter. Formally, we show that $\arg\max_{j\in\mathscr{K}_{h'}} \overline{\mu}^{h'}_j(t,\pi_F) \subseteq \mathscr{K}_{h'+1}$. Let $i^{h'}_{\max} \in \arg\max_{j\in\mathscr{K}_{h'}} \overline{\mu}^{h'}_j(t,\pi_F)$. Then, for such $i^{h'}_{\max}$, we have

$$\widehat{\mu}^{h'}_{i^{h'}_{\max}}(t,\pi_F) \geq \overline{\mu}^{h'}_{i^{h'}_{\max}}(t,\pi_F) - c(h',\delta_t) \geq \overline{\mu}^{h'}_{\max,t} - c(h',\delta_t) \geq \widehat{\mu}^{h'}_{\max,t} - 2c(h',\delta_t),$$

where the first and the third inequality are due to concentration of the estimates on ξ^α_t, while the second one is due to the definition of $i^{h'}_{\max}$.

Since the arms are decaying, the average of the last $h'+1$ mean values for a given arm is always greater than the average of the last h' mean values and therefore,

$$\max_{j\in\mathscr{K}_{h'}} \overline{\mu}^{h'}_j(t,\pi_F) = \overline{\mu}^{h'}_{i^{h'}_{\max}}(t,\pi_F) \leq \overline{\mu}^{h'+1}_{i^{h'}_{\max}}(t,\pi_F) \leq \max_{j\in\mathscr{K}_{h'+1}} \overline{\mu}^{h'+1}_j(t,\pi_F), \tag{4.23}$$

because $i^{h'}_{\max} \in \mathscr{K}_{h'+1}$. Gathering Equations 4.21, 4.22, and 4.23 leads to the first claim of the lemma,

$$\begin{aligned}\overline{\mu}^h_i(t,\pi_F) &\overset{(4.21)}{\geq} \max_{j\in\mathscr{K}_h} \overline{\mu}^h_j(t,\pi_F) - 4c(h,\delta_t) \\ &\overset{(4.23)}{\geq} \max_{j\in\mathscr{K}_1} \overline{\mu}^1_j(t,\pi_F) - 4c(h,\delta_t) \\ &\overset{(4.22)}{\geq} \mu^+_t(\pi_F) - 4c(h,\delta_t).\end{aligned}$$

To conclude, we remark that if i is pulled at the round t, then by the condition at Line 13 of Algorithm 5, it means that i passes through all the filters from $h=1$ up to $N_{i,t-1}$. Therefore, for all $h \leq N_{i,t-1}$,

$$\overline{\mu}^h_i(t,\pi_F) \geq \mu^+_t(\pi_F) - 4c(h,\delta_t). \tag{4.24}$$

■

4.2.3 RAW-UCB: Rotting Adaptive Window Upper Confidence Bound

Algorithm 7 RAW-UCB

Require: $\mathscr{K}$, σ, α

for $t \leftarrow 1, 2, \dots, K$ **do** ▷ *Pull each arm once*
  PULL $i_t \leftarrow t$; RECEIVE o_t
  $N_{i_t} \leftarrow 1$
  $\left\{\widehat{\mu}_{i_t}^h\right\}_h \leftarrow \texttt{UPDATE}(\left\{\widehat{\mu}_{i_t}^h\right\}_h, o_t)$
end for
for $t \leftarrow K+1, K+2, \dots$ **do**
  PULL [a] $i_t \in \arg\max_i \min_{h \le N_i} \left(\widehat{\mu}_i^h + c(h, \delta_t)\right)$; RECEIVE o_t ▷ *cf.* (4.25);
  $N_{i_t} \leftarrow N_{i_t} + 1$
  $\left\{\widehat{\mu}_{i_t}^h\right\}_h \leftarrow \texttt{UPDATE}(\left\{\widehat{\mu}_{i_t}^h\right\}_h, o_t)$
end for

[a] One can choose the tie break selection rule arbitrarily, e.g. by selecting the arm with the smallest index.

We will study a single class of policies which select at each round t the arm with the maximal index of the form

$$\mathrm{ind}(i,t,\delta_t) \triangleq \min_{h \le N_{i,t-1}} \left(\widehat{\mu}_i^h(N_{i,t-1}) + c(h,\delta_t)\right) \quad \text{with} \quad \delta_t \triangleq \frac{2}{t^\alpha}. \tag{4.25}$$

We set and call this algorithm Rotting Adaptive Window UCB (RAW-UCB). There is a bias-variance trade-off for the window choice: more variance for smaller sizes of the window h and more bias for larger h. The goal of RAW-UCB is to adaptively select the right window to compute the tightest UCB. RAW-UCB uses the indexes of UCB1 computed on all the slices of each arm's history which include the last pull. When the rewards are rotting, all these indexes are upper confidence bounds on the *next value*. Thus, RAW-UCB simply selects the tightest (minimum) one as the index of the arm: it is a pure UCB-index algorithm. By contrast, when the reward can increase, the learner can only derive upper-confidence bound on past values which are loosely related to the next value. Hence, all the UCB-index algorithms in the restless non-stationary literature need to add change-detection sub-routine, active random exploration, or passive forgetting mechanism.

Core guarantee on the favorable event

Lemma 4.2.3 At the round t, on favorable event ξ_t^α, if arm i_t is selected by RAW-UCB (α), for any $h \le N_{i_t,t-1}$, the average of its h last pulls cannot deviate significantly from the best available arm at that round, i.e.,

$$\overline{\mu}_{i_t}^h(N_{i_t,t-1}) \ge \max_{i \in \mathscr{K}} \mu_i(N_{i,t-1}) - 2c(h,\delta_t)\,.$$

This lemma is comparable with Lemma 4.2.2 about the algorithm FEWA. Yet, RAW-UCB has tighter guarantees than FEWA (2 versus 4 confidence bands), which is the benefit of upper confidence bounds index policies over confidence bound filtering policies.

Proof. Like for Lemma 4.2.2 (see its proof), our proof is done in a more general rotting framework that can be used in the next chapter.

We denote by $i_t^\star \in \arg\max_{i\in\mathcal{K}} \mu_{i,t}$, a best available arm at time t and

$$h_{i,t}^{\min} \in \operatorname*{arg\,min}_{h\leq N_{i,t-1}} \widehat{\mu}_i^h(t,\pi) + c(h,\delta_t),$$

a window which minimizes `RAW-UCB` index at time t for arm i. Hence, because the reward functions are non-increasing, we know that

$$\mu_{i_t^\star,t} \leq \overline{\mu}_{i_t^\star}^{1}(t,\pi) \leq \cdots \leq \overline{\mu}_{i_t^\star}^{h_{i_t^\star,t}^{\min}}(t,\pi).$$

On the high-probability event ξ_t, we know that the true average of the means cannot deviate significantly from the average of the observed quantity,

$$\overline{\mu}_{i_t^\star}^{h_{i_t^\star,t}^{\min}}(t,\pi) \leq \widehat{\mu}_{i_t^\star}^{h_{i_t^\star,t}^{\min}}(t,\pi) + c(h_{i_t^\star,t}^{\min},\delta_t).$$

We know that the selected arm i_t at time t has the largest index, hence,

$$\widehat{\mu}_{i_t^\star}^{h_{i_t^\star,t}^{\min}}(t,\pi) + c(h_{i_t^\star,t}^{\min},\delta_t) \leq \widehat{\mu}_{i_t}^{h_{i_t,t}^{\min}}(t,\pi) + c(h_{i_t,t}^{\min},\delta_t).$$

From $h_{i,t}^{\min}$ definition, we know that this quantity is below any upper-confidence bound for any other window h

$$\widehat{\mu}_{i_t}^{h_{i_t,t}^{\min}}(t,\pi) + c(h_{i_t,t}^{\min},\delta_t) \leq \widehat{\mu}_{i_t}^{h}(t,\pi) + c(h,\delta_t).$$

Finally, using again the concentration of the average on the ξ_t^α,

$$\widehat{\mu}_{i_t}^{h}(t,\pi) + c(h,\delta_t) \leq \overline{\mu}_{i_t}^{h}(t,\pi) + 2c(h,\delta_t).$$

Hence, putting all the equations together, we can write

$$\overline{\mu}_{i_t}^{h}(t,\pi) \geq \max_{i\in\mathcal{K}} \mu_i(t,N_{i,t-1}) - 2c(h,\delta_t).$$

■

4.3 Regret Analysis

In the last section, we presented two algorithms which have very different behaviours. Yet, they show two main similarities. First, for each arm they compute several statistics $\widehat{\mu}_i^h(N_{i,t-1})$ for different windows $h \leq N_{i,t-1}$. Second, on the same favorable events ξ_t^α (on which all these aforementioned statistics are well concentrated around their means, see Prop. 4.2.1), we have shown that both algorithms share a guarantee with similar shape that we restate in a general form,

Corollary 4.3.1 — Lemmas 4.2.2 and 4.2.3. At a round t, on favorable event ξ_t^α, if arm i_t is selected by $\pi(\alpha) \in \{\pi_R, \pi_F\}$, for any $h \leq N_{i_t,t-1}$, the average of its h last pulls cannot deviate significantly from the best available arm at that round, i.e.,

$$\overline{\mu}_{i_t}^h(N_{i_t,t-1}) \geq \max_{i\in\mathcal{K}} \mu_i(N_{i,t-1}) - \frac{C_\pi}{\sqrt{2\alpha}} c(h,\delta_t) = \max_{i\in\mathcal{K}} \mu_i(N_{i,t-1}) - C_\pi \sigma \sqrt{\frac{\log(t)}{h}},$$
$$\text{with } C_{\pi_R} = 2\sqrt{2\alpha} \text{ and } C_{\pi_F} = 4\sqrt{2\alpha}.$$

We will see that this Corollary is the only characterization we need in our analysis. We first give problem-independent regret bound for FEWA and RAW-UCB and sketch its proof in Subsection 4.3.1. Then, we discuss problem-dependent guarantees in Subsection 4.3.2. Finally, we give a detailed analysis in Subsection 4.3.3.

4.3.1 Problem-independent bound

Theorem 4.3.2 For any rotting bandit scenario with means $\{\mu_i\}_i \in \mathcal{L}_L^K$ and any time horizon T, $\pi \in \{\pi_R, \pi_F\}$ run with $\alpha \geq 5$ suffers an expected regret of

$$\mathbb{E}[R_T(\pi)] \leq C_\pi \sigma \sqrt{\log(T)} \left(\sqrt{KT} + K\right) + 3KL \text{ with } \begin{cases} C_{\pi_R} = 2\sqrt{2\alpha} \\ C_{\pi_F} = 4\sqrt{2\alpha} \end{cases}.$$

Comparison to Levine et al. (2017) The regret of SWA is bounded by $\widetilde{\mathcal{O}}(B^{1/3}K^{1/3}T^{2/3})$ for bounded rotting functions in $\mathcal{B}_B$. According to Subsection 4.1.5, the regret guarantee translate in $\mathcal{O}(T)$ for rotting functions in $\mathcal{L}_L$. Thus, according to its original analysis, SWA may not be able to learn for our general setting. On the other hand, we could use FEWA or RAW-UCB with rotting functions in $\mathcal{B}_B$ and recover the same regret bound with $L := B$. In this case, our two algorithms suffer a regret of $\widetilde{\mathcal{O}}(\sqrt{KT})$, thus significantly improving over SWA.

The improvement is mostly because FEWA and RAW-UCB use adaptive window mechanisms to smoothly track changes in the value of each arm. Indeed, SWA relies on a fixed exploratory phase where all arms are pulled in a round-robin way and the tracking is performed using averages constructed with a fixed window. According to Proposition 4.1.6, this fixed window trades off between the cost of biased estimates $\mathcal{O}(KBh)$ - for scenarios where the arms abruptly decay and their values are overestimated during at most h rounds - and the cost of the variance of estimators $\widetilde{\mathcal{O}}(\sigma T/\sqrt{h})$ - for scenarios where the arms keep their value close to each other for $\mathcal{O}(T)$ rounds. In Theorem 4.3.2, the regret of FEWA and RAW-UCB is also bounded by an additive decomposition between the terms depending on the noise level σ and the terms depending on the rotting level L. Yet, adaptive window algorithms do not need to trade-off: their regret is bounded by $\mathcal{O}(KL) + \widetilde{\mathcal{O}}(\sigma\sqrt{KT})$. It evidence that our algorithms can take decision based on a relevant $h \in \{1, \dots, N_{i,t-1}\}$ depending on the scenarios.

Last, our algorithms are anytime and agnostic to L (or B), while the tuning of SWA requires to know B and T (or to resort to a doubling trick, which performs poorly in practice).

Comparison to stationary stochastic bandits The regret upper bounds of FEWA and RAW-UCB match the worst-case optimal regret bound of the standard stochastic bandits (i.e., $\mu_i(n)$s are constant) up to a logarithmic factor. Whether an algorithm can achieve $\mathcal{O}(\sqrt{KT})$ regret bound is an open question. On one hand, our analysis needs confidence bounds to hold for different windows at the same time, which requires an additional union bound and thus larger confidence intervals w.r.t. UCB1. On the other hand, our worst-case analysis shows that some of the difficult problems that reach the worst-case bound of Thm. 4.3.2 are realized with constant functions, which is the standard stochastic bandits, for which MOSS-like (Audibert and Bubeck 2009) algorithms achieve regret guarantees without the $\sqrt{\log T}$ factor. Thus, the necessity of the extra $\sqrt{\log T}$ factor for the worst-case regret of rotting bandits remains an open problem.

4.3.2 Problem-dependent bound

Since our setting generalizes the stationary stochastic bandit setting, a natural question is whether we pay any price for this generalization. While the result of Levine et al. (2017) suggested that learning in rotting bandits could be more difficult, in Thm. 4.3.2 we actually proved that FEWA and RAW-UCB nearly match the problem-independent regret rate $\widetilde{\mathcal{O}}(\sqrt{KT})$. We may wonder whether this is true for the *problem-dependent* regret as well.

R Consider a stationary stochastic bandit setting with expected rewards $\{\mu_i\}_i$ and $\mu_\star \triangleq \max_i \mu_i$. For $\pi \in \{\pi_F, \pi_R\}$, on the favorable event ξ_t^α with $\delta_t \geq 2/T^\alpha$, we can apply Corollary 4.3.1 at the last time arm i is pulled (*i.e.* after $N_{i,T}-1$ pulls) for $h = N_{i,T}-1$,

$$\mu_\star - \mu_i \leq \frac{C_\pi}{\sqrt{2\alpha}} c\left(N_{i,T}-1, \delta_t\right) = C_\pi \sigma \frac{\sqrt{\log(T)}}{N_{i,T}-1},$$
$$\text{or equivalently, } N_{i,T} \leq 1 + \frac{C_\pi^2 \sigma^2 \log(T)}{(\mu_\star - \mu_i)^2}. \tag{4.26}$$

Therefore, for $\alpha > 4$[1], our algorithms match the lower bound of T. L. Lai and Robbins (1985) up to a constant factor $C_\pi^2/2$.

With a similar argument, we can show a similar bound on the number of overpulls $h_{i,T}$ of arm i in the general rested rotting bandits case. Indeed, we show in Lemma 4.3.7 that $h_{i,T}$ is smaller than a problem-dependent quantity $h_{i,T}^+$ which is itself smaller by construction

[1] α should be large enough to control the cost of the unfavorable events, see Lemma 4.3.5.

than a function of "gaps" $\Delta_{i,h^+_{i,T}-1}$,

$$h^+_{i,T} \triangleq \max\left\{h \in \{1,\dots T\} \,|\, h \leq 1+\frac{C_\pi^2\sigma^2\log(T)}{\Delta^2_{i,h-1}}\right\}$$
$$\text{with } \Delta_{i,h} \triangleq \min_{j\in\mathscr{K}} \mu_j\left(N^\star_{j,T}-1\right) - \overline{\mu}^h_i\left(N^\star_{i,T}+h\right). \quad (4.27)$$

R Notice that for stationary bandits, we have for all h, $\Delta_{i,h} = \Delta_i = \mu_\star - \mu_i$. In fact, $\Delta_{i,h}$ extends the notion of gap to our non-stationary setting: it is the average gap between the smallest value pulled by the optimal policy and the average value of the h first overpulls of arm i. We also highlight that $h^+_{i,t}$ is always defined because $h = 1$ always verify the self-bounding property.

Moreover, on the favorable event ξ_t^α, we can show that the regret of $h_{i,T}$ overpulls of arm i is bounded by $\mathscr{O}(\sqrt{h_{i,T}})$ (see Lemma 4.3.6, in Subsection 4.3.3). Hence, we bound $h_{i,T}$ by $h^+_{i,T}$ and we use the self-bounding property in the definition of $h^+_{i,T}$ (Equation 4.27) to get a $\mathscr{O}(\log(T))$ problem-dependent bound for our algorithms on any rotting bandit scenario.

Theorem 4.3.3 For any rotting bandit scenario with means $\{\mu_i\}_i \in \mathscr{L}^K_L$ and any time horizon T, $\pi \in \{\pi_R, \pi_F\}$ run with $\alpha \geq 5$ suffers an expected regret of

$$\mathbb{E}[R_T(\pi)] \leq \sum_{i\in\mathscr{K}}\left(\frac{C_\pi^2\sigma^2\log(T)}{\Delta_{i,h^+_{i,T}-1}} + C_\pi\sigma\sqrt{\log(T)} + 3L\right),$$
$$\text{with } \begin{cases} C_{\pi_R} = 2\sqrt{2\alpha} \\ C_{\pi_F} = 4\sqrt{2\alpha} \\ \Delta_{i,h} \text{ and } h^+_{i,T} \text{ defined in Equation 4.27.} \end{cases}$$

R The problem-dependent guarantee of `RAW-UCB` is 4 times smaller than the guarantee of `FEWA`: this is the benefits of upper-confidence bound index policies over confidence bound filtering ones. However, for $\alpha = 5$, our guarantee for `RAW-UCB` is still at a factor $C^2_{\pi_R}/2 = 20$ of the lower bound of T. L. Lai and Robbins (1985) for stationary bandits.

This is mostly due to our proof technique. Indeed, Auer et al. (2002) also use a similar high-probability proof for `UCB1` and also get a large factor compared to the lower bound and an over-conservative tuning of the confidence bounds[2]. Yet, even compared to `UCB1`, we have to use a more conservative tuning of the confidence bounds. On the first hand, we use a larger number of estimators at each round: Kt^2 instead of Kt for `UCB`. Hence, after taking the union bound, we need to increase α by

[2]To make the results comparable to the one of Auer et al. (2002), we need to replace σ^2 by $1/4$ for sub-Gaussian noise.

one to have the same probability of the unfavorable event as for UCB1 (see Prop. 4.1.5). On the other hand, for reward functions in $\mathscr{L}_L$, the maximal possible regret at a round t is bounded by Lt which is larger than the constant cost L for the stationary case. Thus, we have to increase α by one to control the cost of the unfavorable event. Notice that it is a consequence of our extended setting: we would not need to increase α for reward functions in $\mathscr{B}_B$.

While we presented our Theorems 4.3.2 and 4.3.3 with $\alpha \geq 5$, we could have similar results for $\alpha > 4$ by replacing the additive term $3KL$ by $(1+\zeta(\alpha-3))KL$ with $\zeta(x) \triangleq \sum_n n^{-x}$. For bounded reward functions, we can further reduce $\alpha > 3$. It is still a larger confidence interval than with $\delta_t \sim \frac{1}{t \log t^2}$, which is used in UCB with asymptotic-optimal tuning for sub-gaussian stationary bandits (Lattimore and Szepesvári 2020). We further discuss the notion of asymptotic optimality and confidence level tuning in rotting bandits in Section 4.6.

4.3.3 Proof

Structure of the proof

In Lemma 4.3.4, we split the regret decomposition according to whether the overpulls has been done on the favorable event ξ_t^α or not.

In Lemma 4.3.5, we show that the part of the expected regret due to pulls under $\overline{\xi_t^\alpha}$ is bounded by a constant with respect to T for $\alpha > 4$. Indeed, while we have only trivial bounds on the quality of the pulls on these events, we can control their probabilities thanks to Proposition 4.2.1.

In Lemma 4.3.6, we show that for $h_{i,T}$ overpulls of arm i, we suffer no more than $\widetilde{\mathscr{O}}\left(\sqrt{h_{i,T}}\right)$ on the favorable event. Indeed, thanks to Corollary 4.3.1, we know that the cost of the h before last pulls is bounded by $h \cdot c(h,\delta_t) = \widetilde{\mathscr{O}}\left(\sqrt{h}\right)$.

The proof of Theorem 4.3.2 follows by noticing that $\sum_{i \in \mathscr{K}} h_{i,T} \leq T$ which leads to the $\widetilde{\mathscr{O}}\left(\sqrt{KT}\right)$ rate. Indeed, thanks to the concavity of the $\sqrt{\cdot}$ and to Jensen's inequality, we find that the worst allocation is $h_{i,T} = \frac{T}{K}$.

In Lemma 4.3.7, we construct a problem-dependent bound of $h_{i,T}$ which extends the notion of gaps for rotting bandits using Corollary 4.3.1.

The proof of Theorem 4.3.3 follows by plugging this bound in the result of Lemma 4.3.6.

Full proof

Let $t_i^\pi(n)$ the function such that $t_i^\pi(n) = t$ when policy π selects arm i at time t for the n-th time. We call $\mu_T^+(\pi) \triangleq \max_{i \in \mathscr{K}} \mu_i(N_{i,T})$, *i.e.* the largest available reward for π at the round T+1.

Lemma 4.3.4 Let $h_{i,T} \triangleq |N_{i,T} - N^\star_{i,T}|$. For any policy π, the regret at the round T is no bigger than

$$R_T(\pi) \leq \sum_{i \in \mathrm{OP}} \sum_{h=0}^{h_{i,T}-1} \mathbb{1}\left[\xi^\alpha_{t_i^\pi(N^\star_{i,T}+h)}\right] \left(\mu_T^+(\pi) - \mu_i(N^\star_{i,T}+h)\right) + \sum_{t=1}^{T} \mathbb{1}\left[\overline{\xi^\alpha_t}\right] Lt.$$

We refer to the the first sum above as to A_π and to the second sum as to B.

Proof. We consider the regret at the round T. We start from the upper bound in Eq. 4.5,

$$R_T(\pi) \leq \sum_{i \in \mathrm{OP}} \sum_{h=0}^{h_{i,T}-1} \left(\mu_T^+(\pi) - \mu_i(N^\star_{i,T}+h)\right). \tag{4.28}$$

Then, we need to separate overpulls that are done under ξ^α_t and under $\overline{\xi^\alpha_t}$. We introduce $t_i^\pi(n)$, the round at which π pulls arm i for the n-th time. We now make the round at which each overpull occurs explicit,

$$\begin{aligned}
R_T(\pi) &\leq \sum_{i \in \mathrm{OP}} \sum_{h=0}^{h_{i,T}-1} \sum_{t=1}^{T} \mathbb{1}\left[t_i^\pi\left(N^\star_{i,T}+h\right) = t\right] \left(\mu_T^+(\pi) - \mu_i(N^\star_{i,T}+h)\right) \\
&\leq \underbrace{\sum_{i \in \mathrm{OP}} \sum_{h=0}^{h_{i,T}-1} \sum_{t=1}^{T} \mathbb{1}\left[t_i^\pi\left(N^\star_{i,T}+h\right) = t \wedge \xi^\alpha_t\right] \left(\mu_T^+(\pi) - \mu_i(N^\star_{i,T}+h)\right)}_{A_\pi} \\
&\quad + \underbrace{\sum_{i \in \mathrm{OP}} \sum_{h=0}^{h_{i,T}-1} \sum_{t=1}^{T} \mathbb{1}\left[t_i^\pi\left(N^\star_{i,T}+h\right) = t \wedge \overline{\xi^\alpha_t}\right] \left(\mu_T^+(\pi) - \mu_i(N^\star_{i,T}+h)\right)}_{B}.
\end{aligned}$$

For the analysis of the pulls done under ξ^α_t we do not need to know at which round it was done. Therefore,

$$A_\pi \leq \sum_{i \in \mathrm{OP}} \sum_{h=0}^{h_{i,T}-1} \mathbb{1}\left[\xi^\alpha_{t_i^\pi(N^\star_{i,t}+h)}\right] \left(\mu_T^+(\pi) - \mu_i(N^\star_{i,T}+h)\right).$$

For `FEWA` or `RAW-UCB`, it is not easy to directly guarantee the low probability of overpulls (the second sum). Thus, we upper-bound the regret of each overpull at a round t under $\overline{\xi^\alpha_t}$ by its maximum value Lt. While this is done to ease `FEWA` analysis, this is valid for any policy π. Then, noticing that we can have at most 1 overpull per round t, i.e., $\sum_{i \in \mathrm{OP}} \sum_{h=0}^{h_{i,T}-1} \mathbb{1}\left[t_i^\pi\left(N^\star_{i,T}+h\right) = t\right] \leq 1$, we get

$$B \leq \sum_{t=1}^{T} \left[\overline{\xi^\alpha_t}\right] Lt \left(\sum_{i \in \mathrm{OP}} \sum_{h=0}^{h_{i,T}-1} \mathbb{1}\left[t_i^\pi\left(N^\star_{i,T}+h\right) = t\right]\right) \leq \sum_{t=1}^{T} \left[\overline{\xi^\alpha_t}\right] Lt.$$

Therefore, we conclude that

$$R_T(\pi) \leq \underbrace{\sum_{i \in \mathrm{OP}} \sum_{h=0}^{h_{i,T}-1} \mathbb{1}\left[\xi^\alpha_{t_i^\pi(N^\star_{i,t}+h)}\right] \left(\mu_T^+(\pi) - \mu_i(N^\star_{i,T}+h)\right)}_{A_\pi} + \underbrace{\sum_{t=1}^{T} \left[\overline{\xi^\alpha_t}\right] Lt}_{B}.$$

■

Lemma 4.3.5 Let $\zeta(x) = \sum_n n^{-x}$. Thus, with $\delta_t = 2t^{-\alpha}$ and $\alpha > 4$, we can use Proposition 4.2.1 and get

$$\mathbb{E}[B] \triangleq \sum_{t=1}^{T} p\left(\overline{\xi_t^{\alpha}}\right) Lt \leq \sum_{t=1}^{T} KLt^{3-\alpha} \leq KL\zeta(\alpha-3).$$

In particular, for $\alpha \geq 5$, we have:

$$\mathbb{E}[B] \leq KL\zeta(2) \leq 2KL.$$

Lemma 4.3.6 We define $h_{i,T}^{\xi} \triangleq \max\left\{h \leq h_{i,T} \text{s.t. } \xi^{\alpha}_{t_i^{\pi}(N^{\star}_{i,t}+h)} \text{holds}\right\}$, the largest number of overpulls of arm i pulled under ξ_t^{α} at the round $t = t_i^{\pi}(N_{i,t}^{\star} + h_{i,T}^{\xi}) \leq T$. We also define $\text{OP}_{\xi} \triangleq \left\{i \in \text{OP} |\ h_{i,T}^{\xi} \geq 1\right\}$. For policy $\pi \in \{\pi_{\text{R}}, \pi_{\text{F}}\}$ with parameter α, A_{π} defined in Lemma 4.3.4 is upper-bounded by

$$\begin{aligned} A_{\pi} &\triangleq \sum_{i\in\text{OP}} \sum_{h=0}^{h_{i,T}-1} \mathbb{1}\left[\xi^{\alpha}_{t_i^{\pi}(N^{\star}_{i,T}+h)}\right] \left(\mu_T^{+}(\pi) - \mu_i(N_{i,T}^{\star}+h)\right) \\ &\leq \sum_{i\in\text{OP}_{\xi}} \left(C_{\pi}\sigma\sqrt{\left(h_{i,T}^{\xi}-1\right)\log(T)} + C_{\pi}\sigma\sqrt{\log(T)} + L\right). \end{aligned}$$

Proof. We upper-bound A_{π} by including all the overpulls of arm i until the $h_{i,T}^{\xi}$-th overpull, even the ones under $\overline{\xi_t^{\alpha}}$,

$$\begin{aligned} A_{\pi} &\triangleq \sum_{i\in\text{OP}} \sum_{h=0}^{h_{i,T}-1} \mathbb{1}\left[\xi^{\alpha}_{t_i^{\pi}(N^{\star}_{i,T}+h)}\right] \left(\mu_T^{+}(\pi) - \mu_i(N_{i,T}^{\star}+h)\right) \\ &\leq \sum_{i\in\text{OP}_{\xi}} \sum_{h=0}^{h_{i,T}^{\xi}-1} \left(\mu_T^{+}(\pi) - \mu_i(N_{i,T}^{\star}+h)\right), \end{aligned}$$

where $\text{OP}_{\xi} \triangleq \left\{i \in \text{OP} |\ h_{i,T}^{\xi} \geq 1\right\}$. We can split the second sum of $h_{i,T}^{\xi}$ terms above into two parts: on the one hand, the first $h_{i,T}^{\xi} - 1$ (possibly zero) terms (overpulling differences); and on the other hand, the last $(h_{i,T}^{\xi} - 1)$-th one. Recalling that at the round t_i, arm i was selected under $\xi_{t_i}^{\alpha}$, we apply Corollary 4.3.1 to bound the regret caused by the first $h_{i,T}^{\xi} - 1$ overpulls of i (possibly none),

$$A_{\pi} \leq \sum_{i\in\text{OP}_{\xi}} \mu_T^{+}(\pi) - \mu_i\left(N_{i,T}^{\star} + h_{i,T}^{\xi} - 1\right) + \frac{C_{\pi}}{\sqrt{2\alpha}}\left(h_{i,T}^{\xi}-1\right) c\left(h_{i,T}^{\xi}-1, \delta_{t_i}\right) \quad (4.29)$$

$$\leq \sum_{i\in\text{OP}_{\xi}} \mu_T^{+}(\pi) - \mu_i\left(N_{i,T}^{\star} + h_{i,T}^{\xi} - 1\right) + \frac{C_{\pi}}{\sqrt{2\alpha}}\left(h_{i,T}^{\xi}-1\right) c\left(h_{i,T}^{\xi}-1, \delta_{T}\right) \quad (4.30)$$

$$\leq \sum_{i\in\text{OP}_{\xi}} \mu_T^{+}(\pi) - \mu_i\left(N_{i,T}^{\star} + h_{i,T}^{\xi} - 1\right) + C_{\pi}\sigma\sqrt{\left(h_{i,T}^{\xi}-1\right)\log(T)}. \quad (4.31)$$

The second inequality is obtained because δ_t is decreasing and $c(\cdot,\delta)$ is decreasing as well. The last inequality is the definition of confidence interval in Proposition 4.2.1 with $\delta_T = 2T^{-\alpha}$. If $N^\star_{i,T} = 0$ and $h^\xi_{i,T} = 1$, then,

$$\mu^+_T(\pi) - \mu_i(N^\star_{i,T} + h^\xi_{i,T} - 1) = \mu^+_T(\pi) - \mu_i(0) \leq L,$$

since $\mu^+_T(\pi) \leq \max_{j\in\mathscr{K}} \mu_j(0)$ and $\max_{j\in\mathscr{K}} \mu_j(0) - \mu_i(0) \leq L$ because $\{\mu_i\}_{i\in\mathscr{K}} \in \mathscr{L}^K_L$ (Def. 4.1.1).

We do not have direct guarantees on the value of the last pull because it may have decay. However, we know that it has decay by less than L since the before last pull (term A_2 in the next equation). We also have a guarantee on the value of this before last pull thanks to our adaptive window mechanism (term A_1). That is why, we decompose,

$$\begin{aligned}\mu^+_T(\pi) - \mu_i(N^\star_{i,T} + h^\xi_{i,T} - 1) = &\underbrace{\mu^+_T(\pi) - \mu_i(N^\star_{i,T} + h^\xi_{i,T} - 2)}_{A_1} \\ &+ \underbrace{\mu_i(N^\star_{i,T} + h^\xi_{i,T} - 2) - \mu_i(N^\star_{i,T} + h^\xi_{i,T} - 1)}_{A_2}.\end{aligned}$$

For term A_1, since this $h^\xi_{i,T}$-th overpull is done under $\xi^\alpha_{t_i}$, by Corollary 4.3.1 we have that

$$A_1 = \mu^+_T(\pi) - \overline{\mu}^1_i(N^\star_{i,T} + h^\xi_{i,T} - 1) \leq 1c(1,\delta_{t_i}) \leq 2c(1,\delta_T) \leq C_\pi \sigma \sqrt{\log(T)}.$$

The second difference, $A_2 = \mu_i(N^\star_{i,T} + h^\xi_{i,T} - 2) - \mu_i(N^\star_{i,T} + h^\xi_{i,T} - 1)$ cannot exceed L, since by the assumptions of our setting (Def. 4.1.1), the maximum decay in one round is bounded. Therefore, we further upper-bound Equation 4.31 as

$$A_\pi \leq \sum_{i\in \mathrm{OP}_\xi} \left(C_\pi \sigma \sqrt{\left(h^\xi_{i,T} - 1\right)\log(T)} + C_\pi \sigma \sqrt{\log(T)} + L \right). \tag{4.32}$$

■

Proof of Theorem 4.3.2. In Lemma 4.3.4, we split the regret in two parts. The first one B corresponds to the regret due to unfavorable events $\overline{\xi^\alpha_t}$. We do not derive any guarantee of our algorithms on these events but their probabilities can be controlled thanks to parameter α. Hence, for $\alpha > 4$, we show in Lemma 4.3.5 that the part of the expected regret due to unfavorable events can be bounded by a constant w.r.t. T. Yet, we choose $\alpha \geq 5$ to have a small constant.

The second one A_π corresponds to the regret due to favorable events ξ^α_t which can be bounded for our two algorithms (`FEWA` and `RAW-UCB`) thanks to Lemma 4.3.6. In order to get a problem-independent upper bound, we need to replace $h^\xi_{i,T}$ by a problem-independent quantity. Starting from Lemma 4.3.6,

$$A_\pi \leq \sum_{i \in \mathrm{OP}_\xi} \left(C_\pi \sigma \sqrt{\left(h_{i,T}^\xi - 1 \right) \log(T)} + C_\pi \sigma \sqrt{\log(T)} + L \right).$$

We can upper-bound the number of terms in the above sum by K. Moreover, we recall that $h_{i,T}^\xi \leq h_{i,T}$ and that the total number of overpulls $\sum_{i \in \mathrm{OP}} h_{i,T}$ cannot exceed T. As square-root function is concave we can use Jensen's inequality. Moreover, we can deduce that the worst allocation of overpulls is the uniform one, i.e., $h_{i,T} = T/K$,

$$\begin{aligned} A_\pi &\leq K(C_\pi \sigma \sqrt{\log(T)} + L) + C_\pi \sigma \sqrt{\log(T)} \sum_{i \in \mathrm{OP}} \sqrt{(h_{i,T} - 1)} \\ &\leq K(C_\pi \sigma \sqrt{\log(T)} + L) + C_\pi \sigma \sqrt{KT \log(T)}. \end{aligned} \tag{4.33}$$

Therefore, using Lemma 4.3.4 together with Equations 4.33 and Lemma 4.3.5, we bound the total expected regret as

$$\mathbb{E}[R_T(\pi)] \leq C_\pi \sigma \sqrt{\log(T)} \left(\sqrt{KT} + K \right) + 3KL\cdot \tag{4.34}$$

■

Lemma 4.3.7 We define the smallest reward gathered by the optimal policy μ_T^- and the gap of the h first overpulls of arm i with respect to that value $\Delta_{i,h}$.

$$\begin{aligned} &\mu_T^- \triangleq \min_{i \in \mathscr{K}^\star} \mu_i \left(N_{i,T}^\star - 1 \right) \text{ with } \mathscr{K}^\star \triangleq \left\{ i \in \mathscr{K} | N_{i,T}^\star \geq 1 \right\}, \\ &\Delta_{i,h} \triangleq \mu_T^- - \overline{\mu}_i^h \left(N_{i,t}^\star + h \right). \end{aligned}$$

$h_{i,T}^\xi$ defined in Lemma 4.3.4 is upper-bounded by a problem-dependent quantity,

$$h_{i,T}^\xi \leq h_{i,T}^+ \triangleq \max \left\{ h \leq T \,\middle|\, h \leq 1 + \frac{C_\pi^2 \sigma^2 \log(T)}{\Delta_{i,h-1}^2} \right\} \leq 1 + \frac{C_\pi^2 \sigma^2 \log(T)}{\Delta_{i,h_{i,T}^+ - 1}^2}.$$

Proof. We want to bound $h_{i,T}^\xi$ with a problem dependent quantity $h_{i,T}^+$. We remind the reader that for arm i, the $h_{i,T}^\xi$-th overpull is pulled under $\xi_{t_i}^\alpha$ at the round t_i. Therefore, Corollary 4.3.1 applies and we have

$$\begin{aligned} \overline{\mu}_i^{h_{i,T}^\xi - 1} \left(N_{i,T}^\star + h_{i,T}^\xi - 1 \right) &\geq \mu_T^+(\pi) - \frac{C_\pi}{\sqrt{2\alpha}} c \left(h_{i,T}^\xi - 1, \delta_{t_i} \right) \\ &\geq \mu_T^+(\pi) - \frac{C_\pi}{\sqrt{2\alpha}} c \left(h_{i,T}^\xi - 1, \delta_T \right) \\ &\geq \mu_T^+(\pi) - C_\pi \sigma \sqrt{\frac{\log(T)}{h_{i,T}^\xi - 1}}, \end{aligned}$$

Hence, we have that

$$h^{\xi}_{i,T} \leq 1 + \frac{C_\pi^2 \sigma^2 \log(T)}{\left(\mu_T^+(\pi) - \overline{\mu}_i^{h^{\xi}_{i,T}-1}\left(N^\star_{i,T} + h^{\xi}_{i,T} - 1\right)\right)^2}. \tag{4.35}$$

We will justify in few lines that $\mu_T^+ \geq \overline{\mu}_i^{h^{\xi}_{i,T}-1}\left(N^\star_{i,T} + h^{\xi}_{i,T} - 1\right)$ when the regret is not null. Yet, this upper bound still depends on random quantities such as $\mu_T^+(\pi)$ or $h^{\xi}_{i,T}$ on the denominator. Consider the smallest value collected by the optimal policy,

$$\mu_T^- \triangleq \min_{i \in \mathscr{K}^\star} \mu_i\left(N^\star_{i,T} - 1\right) \text{ with } \mathscr{K}^\star \triangleq \left\{i \in \mathscr{K} | N^\star_{i,T} \geq 1\right\}.$$

We recall that the greedy oracle π_{O} selects the rewards in the decreasing order (see the proof of Proposition 4.1.1). Therefore, μ_T^- (the smallest value selected at the round T) is the T-th largest value among the KT possible ones. Moreover, the overpulls - which are the values that are not among the T largest ones selected by π_{O} - are all smaller than μ_T^-. Since $\overline{\mu}_i^{h^{\xi}_{i,T}-1}\left(N^\star_{i,T} + h^{\xi}_{i,T} - 1\right)$ is an average of overpulls' values, we have,

$$\mu_T^- \geq \overline{\mu}_i^{h^{\xi}_{i,T}-1}\left(N^\star_{i,T} + h^{\xi}_{i,T} - 1\right).$$

Moreover, $\mu_T^- > \mu_T^+(\pi)$ implies that the regret is 0. Indeed, in that case $\mu_T^+(\pi)$ - the pull with the largest value among the remaining values at the end of the game for π - is *strictly smaller* than μ_T^- - the T-th largest reward sample. Therefore, π has collected the T largest values and has zero regret. Hence, we focus on the case $\mu_T^- \leq \mu_T^+(\pi)$, for which the regret may not be zero. In that case, we can upperbound the RHS term Equation 4.35 by replacing the random quantity $\mu_T^+(\pi)$ by the smaller quantity μ_T^-. We do have that $\overline{\mu}_i^{h^{\xi}_{i,T}-1}\left(N^\star_{i,T} + h^{\xi}_{i,T} - 1\right) \leq \mu_t^- \leq \mu_T^+$ when the regret is not null. Hence,

$$h^{\xi}_{i,T} \leq 1 + \frac{C_\pi^2 \sigma^2 \log(T)}{\left(\mu_T^+(\pi) - \overline{\mu}_i^{h^{\xi}_{i,T}-1}\left(N^\star_{i,T} + h^{\xi}_{i,T} - 1\right)\right)^2} \leq 1 + \frac{C_\pi^2 \sigma^2 \log(T)}{\Delta^2_{i,h^{\xi}_{i,T}-1}},$$

with $\Delta_{i,h} \triangleq \mu_T^- - \overline{\mu}_i^h\left(N^\star_{i,t} + h\right)$, the difference between the lowest mean value of the arm pulled by $\pi^\star$ and the average of the h first overpulls of arm i. Yet, this self-bounding property of $h^{\xi}_{i,T}$ is not a proper problem-dependent upper bound. We will consider the largest h which satisfies this self-bounding property,

$$h^+_{i,T} \triangleq \max\left\{h \leq T \middle| \; h \leq 1 + \frac{C_\pi^2 \sigma^2 \log(T)}{\Delta^2_{i,h-1}}\right\}.$$

We have that,

$$h^{\xi}_{i,T} \leq h^+_{i,T} \leq 1 + \frac{C_\pi^2 \sigma^2 \log(T)}{\Delta^2_{i,h^+_{i,T}-1}}.$$

■

Proof of Theorem 4.3.3. We use Lemmas 4.3.6 and Lemma 4.3.7 to bound A_π (see Lemma 4.3.4). Indeed, since the square-root function is increasing, we can upper-bound the result in Lemma 4.3.6 by replacing $h^{\xi}_{i,T}$ by its upper bound in Lemma 4.3.7

$$\begin{aligned} A_\pi &\leq \sum_{i \in \mathrm{OP}_\xi} \left(C_\pi \sigma \sqrt{\log(T)} \left(1 + \sqrt{h^+_{i,T} - 1} \right) + L \right) \\ &\leq \sum_{i \in \mathrm{OP}_\xi} \left(C_\pi \sigma \sqrt{\log(T)} \left(1 + \frac{C_\pi \sigma \sqrt{\log(T)}}{\Delta_{i,h^+_{i,T}-1}} \right) + L \right). \end{aligned}$$

Notice that the quantity $\mathrm{OP}_\xi \subset \mathscr{K}$. Therefore, we have

$$A_\pi \leq \sum_{i \in \mathscr{K}} \left(\frac{C_\pi^2 \sigma^2 \log(T)}{\Delta_{i,h^+_{i,T}-1}} + C_\pi \sigma \sqrt{\log(T)} + L \right). \tag{4.36}$$

Using Lemmas 4.3.4, 4.3.5, and Equation 4.36 we get

$$\begin{aligned} \mathbb{E}[R_T(\pi)] &= \mathbb{E}[A_\pi] + \mathbb{E}[B] \\ &\leq \sum_{i \in \mathscr{K}} \left(\frac{C_\pi^2 \sigma^2 \log(T)}{\Delta_{i,h^+_{i,T}-1}} + C_\pi \sigma \sqrt{\log(T)} + L \right) + 2KL \\ &\leq \sum_{i \in \mathscr{K}} \left(\frac{C_\pi^2 \sigma^2 \log(T)}{\Delta_{i,h^+_{i,T}-1}} + C_\pi \sigma \sqrt{\log(T)} + 3L \right). \end{aligned}$$

■

4.4 Experimental benchmarks

We use the two benchmarks described in Subsection 4.1.4.

4.4.1 Simulated benchmark #1 (2 arms).

Algorithms. We display the performance of `RAW-UCB` and `FEWA` for two versions of each algorithm: with the theoretical tuning $\alpha = 4$; and with the empirical tuning $\alpha_{\mathrm{R}} = 1.4$ and $\alpha_{\mathrm{F}} = 0.06$. These two values are selected by grid-search. Though there are 30 different problems (for different L), the best tuning of α is the same for all the considered problem. We also include the three versions of `wSWA` that we displayed in Subsection 4.1.4.

Results - `RAW-UCB` versus `FEWA`. We compare `RAW-UCB` and `FEWA` both for theoretical and empirical tuning. For theoretical tuning, we see in Figure 4.4 (top), that `RAW-UCB` outperforms `FEWA` on all sizes of decays by a factor ~ 4 which is predicted by our theory. Indeed, there is also a factor 4 between the two problem-dependent upper-bounds (Theorem 4.3.3).

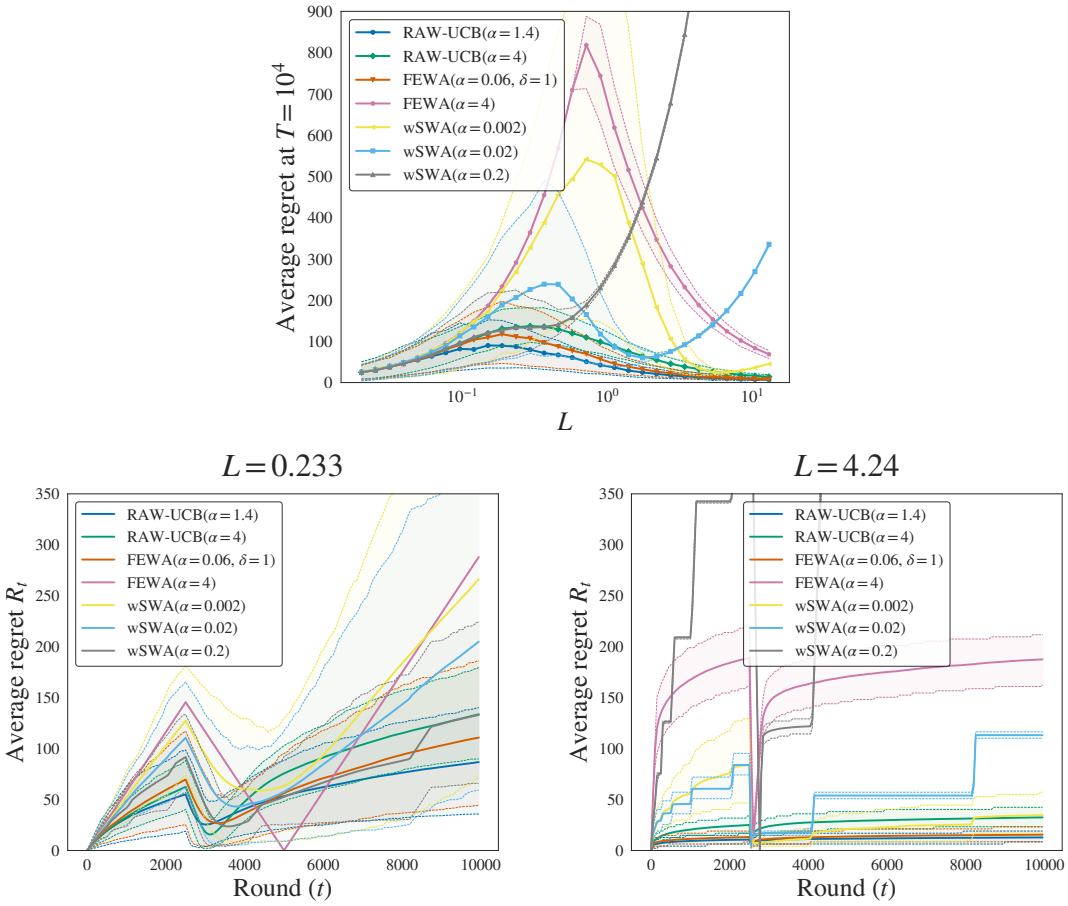


Figure 4.4: **Top:** Regret at the end of the game for different values of L. **Bottom:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region.

Surprisingly, for empirical tuning, the average performances of the two algorithms are much closer. We also notice that there is a larger variance in FEWA's result compared to RAW-UCB. This is not surprising because we had to drastically reduce the confidence bounds to make FEWA practical. It means that empirical FEWA filters arms based only on a handful of samples. This bet leads to both very good and very bad runs. Last, Figure 4.4 (bottom) shows that RAW-UCB outperforms FEWA at almost any time t, both on easy ($L = 4.24$) and difficult ($L = 0.233$) problems. The only round at which FEWA shows better performance than RAW-UCB is after the regret decay. It is because FEWA was less good at identifying the best arm in the first part of the game. Hence, just after the decay, it pulls more the other arm - which has become optimal.

In the following, we will compare RAW-UCB with wSWA. Notice that a similar comparison can hold for FEWA ($\alpha = 0.06$).

Results - Problem dependent performance and the impact of L. RAW-UCB with the best empirical tuning improves over wSWA on each problem (Figure 4.4 (top)). RAW-UCB

with the theoretical tuning recovers quite good performance as well.

In this setting, L has two different meanings. It is the maximum decay per round (noted as L in the theoretical section) and the gap between arms $\Delta_{2,h} = L/2$ (for any h). According to our problem-dependent bound in Theorem 4.3.3, the regret bound converges to $\mathcal{O}(KL)$ when L and $\Delta_{2,h}$ are large with respect to σ. It tends to show that setup where arms are well separated from each other are easy problems for FEWA and RAW-UCB. It is indeed confirmed in Figure 4.4 (top), where the regret of FEWA and RAW-UCB converges to $L/2$ when L is large.

Results - Worst-case improvement. In Figure 4.4 (top), the worst regret for any of the two versions of RAW-UCB is smaller than the worst regret of any of the three versions wSWA. Moreover, we remark that the regret at the round T has one maximum for the variation of L for RAW-UCB. This is not the case for wSWA where the regret increases again for large values of L.

It confirms our analysis. Indeed, Theorem 4.3.2 shows a larger regret rate than Proposition 4.1.6. Moreover, the analysis shows that the worst cases for RAW-UCB correspond to cases where the learner does $\mathcal{O}(T)$ mistakes of intermediate size $\mathcal{O}\left(\sqrt{K/T}\right)$ which corresponds to the single maximum in Figure 4.4 (top).

Results - Tuning and agnostic algorithms. Figure 4.4 (top) confirms that FEWA and RAW-UCB do not rely on the knowledge of L. Indeed, the optimal tuning is the same for all the 30 problems. By contrast, the performance of wSWA depends critically on the prior knowledge of L: each of the three displayed tunings is the best for a specific range of L.

Figure 4.4 (bottom) shows the advantage of anytime algorithms compared to the doubling trick. Indeed, the periodic restarts are quite expensive for wSWA.

Results - High-probability. We see that the variance of wSWA is quite large for intermediate values of L. It confirms the analysis of wSWA which shows two sources of the regret: the variance and the bias of the index. The regrets caused by variance has itself a large variance. Indeed, the sub-optimal arms are often correctly estimated, and hence not pulled by the index policy. It leads to many good runs of wSWA. However, there are still many runs on which there is a sufficient deviation in the indexes which leads to very large regret.

By contrast, the variance in the results is much more controlled by RAW-UCB and FEWA. Indeed, when the statistics of these algorithms are not significant enough they tend to explore which leads to less large deviation of the regret.

4.4.2 Simulated benchmark #2 (10 arms).

Algorithms. We display the same two versions of FEWA and RAW-UCB. We also show the three best algorithms presented in Subsection 4.1.4: two versions of wSWA with $\alpha \in$

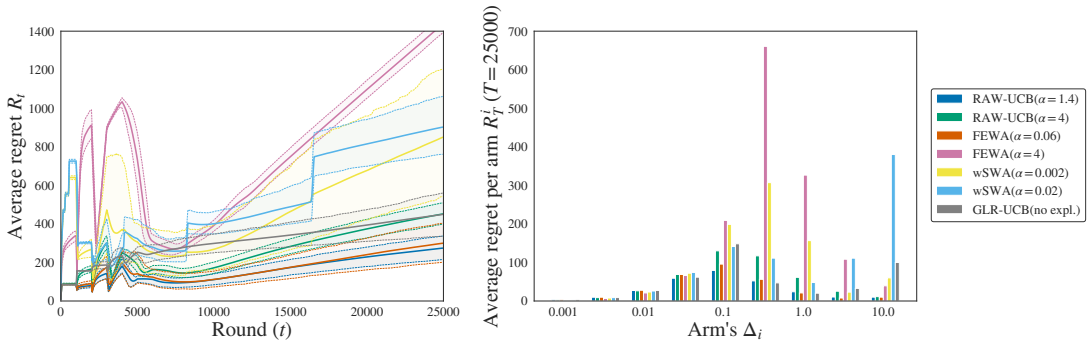


Figure 4.5: **Left:** Regret at the end of the game for different values of L. **Middle, Right:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region.

$\{0.002, 0.02\}$ and GLR-UCB with no exploration.

Results. The comparison between RAW-UCB, FEWA, and wSWA leads to a similar conclusion than for the two-arm bandit experiment. RAW-UCB and FEWA show superior performance, except for the theoretical tuning of FEWA which is too conservative.

In particular, these algorithms show a better adaptation to each arm's gap. Indeed, the regret per arm is more controlled, especially for large values of the gaps, on which wSWA suffers a large regret. There is also less deviation in the regret and we see the benefits of avoiding the doubling trick.

In the two-arm setup with a single decay, it is possible to find a value of α for which wSWA is correctly tuned for the specific decay. For instance, for $L \in [1,3]$, wSWA with $\alpha = 0.02$ has almost the same performance than RAW-UCB (Fig. 4.4). In the ten-arm setup with multiple decays, this is not possible anymore. Indeed, since there are several dropping values for each arm, there exists at least one arm on which the fixed window of wSWA is not correctly tuned. For instance, for $\alpha = 0.002$, wSWA suffers a large regret on the arm with $\Delta_i = 0.3$. For wSWA with $\alpha = 0.02$, the regret is large when $\Delta_i = 10$.

RAW-UCB and FEWA also improve over GLR-UCB when their confidence bounds are tuned. We recall that GLR-UCB is an algorithm that uses a classical UCB index with a change detection procedure. When the change-detection procedure triggers, it erases the history of the changing arm. Notice that the confidence bounds of the index of GLR-UCB are already well-tuned, as they use the same confidence bounds as the asymptotic optimal tuning of UCB. GLR-UCB shows sub-optimal performance on two arms $\Delta_i \in \{0.1, 10\}$. GLR-UCB suffers from the late restart for $\Delta_i = 0.1$. Indeed, the change-point is hard to detect, and the index of the sub-optimal value is positively biased while it has not restarted. For $\Delta_i = 10$, the large regret of GLR-UCB is due to an implementation artefact. Indeed, we used the fast implementation for the change detector (by default in (Besson 2018)). It speeds up the algorithm but it can delay the change-detection scheme (by 10 pulls in this case). This delay leads to large regret when the mistake associated with each arm is large (as it is the

case for $\Delta_i = 10$).

Policy	Running time (s)
FEWA ($\alpha = 0.06$)	91
FEWA ($\alpha = 4$)	780
RAW-UCB ($\alpha = 1.4$)	27
RAW-UCB ($\alpha = 4$)	25
wSWA ($\alpha = 0.002$)	1
wSWA ($\alpha = 0.02$)	1
GLR-UCB	46

Table 4.1: Average running time for the 10-arms experiment in seconds.

Running time. In Table 4.1, we display the running time for this experiment. The computational experiments were conducted using the Grid'5000 experimental testbed (Balouek et al. 2013). For meaningful comparison, all the algorithms run on the same "Grenoble/dahu" cluster (2 CPUs Intel Xeon Gold 6130, 16 cores/CPU, 192GB RAM, 223GB SSD, 447GB SSD, 3726GB HDD, 1 x 10Gb Ethernet, 1 x 100Gb Omni-Path).

RAW-UCB runs 25 times slower than wSWA. We will provide a computational analysis in the next section but we can already relate this increased running time with the higher number of statistics RAW-UCB update and compare at each round.

The α parameter of FEWA has a large impact on the running time. Indeed, the larger the α, the less aggressive are the filters, the longer it takes to reach the end of the filtering process. Yet, even when α is small, FEWA is slower than RAW-UCB. This is a consequence of the simplicity of the index policy over the filtering procedure. Indeed, in Python, we can use the fast C++ implementation of the scientific computing library Numpy to perform the most classical operations. Hence, for RAW-UCB, we only use the NumPy functions arg max and min to choose the next arm. For FEWA, the comparison part is more custom: we had to implement the while-loop at Line 10 with a Python loop, which is known to be quite slow. Notice that since the two algorithms use the same statistics we use the same function UPDATE in both algorithms.

GLR-UCB is slower than RAW-UCB. Notice that it is already a fast version of GLR-UCB which runs the change-detection subroutine sparsely (approximately 10 to 100 times faster than the original GLR-UCB).

We emphasize the better characteristics of RAW-UCB over FEWA: better bounds, better empirical performances, easier and faster implementation, a better agreement between theory and practice, closer to the classical UCB. For these reasons, we will focus our future empirical investigation on RAW-UCB.

4.5 Efficient algorithms

4.5.1 The numerical cost of adaptive windows

In the three last sections, we presented two adaptive windows algorithms that significantly improved over state-of-the-art algorithms, both theoretically and experimentally. Yet, our numerical experiments indicate that these improvements are computationally expensive. Indeed, at each round t, we store, update and compare $\mathcal{O}(t)$ statistics.

The full update of the statistics can be done at a worst case cost of $\mathcal{O}(t)$. Indeed, each statistics $\widehat{\mu}_i^h$ can be refreshed with a $\mathcal{O}(1)$ operation:

$$\widehat{\mu}_i^{h+1}(n+1) = \frac{h}{h+1}\widehat{\mu}_i^h(n) + \frac{1}{h+1}o_t\,.$$

The comparison part in both FEWA and RAW-UCB is also a $\mathcal{O}(t)$ operations. In FEWA, we do a scan based on $\widehat{\mu}_i^h$ for all $i \in \mathcal{K}_h$ with increasing h. Hence, the total number of unitary operations is in $\mathcal{O}(t)$ in the worst case, as it scales with the number of statistics. RAW-UCB computes one UCB for each of the $\mathcal{O}(t)$ statistics. For each arm, it selects the minimum UCB as the index, which can be done with complexity $\mathcal{O}(t)$. Finally, finding the largest index is a $\mathcal{O}(K)$ operation. Therefore, we can conclude,

Proposition 4.5.1 At any round t, FEWA and RAW-UCB have a $\mathcal{O}(t)$ worst-case complexity in time and memory.

R SWA (h) has a $\mathcal{O}(h)$ worst-case complexity in time and memory because the sliding-window mechanism needs to store and update $\mathcal{O}(h)$ statistics to always have the average of the h last sample ready. Hence, when it is optimally tuned for the minimax bound, SWA has a $\mathcal{O}\left(T^{2/3}\right)$ per round complexity. As often in non-stationary bandits, it may be possible to replace sliding window statistics with discounted statistics. Such modification often leads to a slightly worse theoretical regret rate with a much better $\mathcal{O}(K)$ complexity.

Hence, handling a large number of windows, which is the main strength of our algorithms to achieve a lower regret, is a significant drawback when it comes to design fast algorithms. Therefore, it is an open question whether one can enjoy the benefits of adaptive windows without suffering large time and space complexity.

4.5.2 The efficient update trick

We detail EFF_UPDATE, an update scheme to handle efficiently statistics of different windows. A similar yet different approach has appeared independently in the context of streaming mining (Bifet and Gavaldà 2007). EFF_UPDATE is built on two main ideas: *geometrically sparse* and *delayed* statistics.

First, at any time t we can avoid using $\left\{\widehat{\mu}_i^h\right\}_h$ for all possible windows h starting from 1 with an increment of 1. In fact, both statistics $\widehat{\mu}_i^h$ and constructed confidence levels $c(h,\delta_t)$ have very close value for successive h as h becomes large:

$$\widehat{\mu}_i^{h+1}(n) = \widehat{\mu}_i^h(n) + \mathscr{O}\left(\frac{\sigma+L}{h}\right),$$
$$c(h+1,\delta_t) = c(h,\delta_t) + \mathscr{O}\left(\frac{\sigma}{h^{3/2}}\right).$$

Hence, in both `FEWA` and `RAW-UCB`, we compute a lot of very similar quantities. Instead, we could use fewer statistics which are significantly different: $\left\{\widehat{\mu}_i^h(N_{i,t-1}^\pi)\right\}_{h\in H_{i,m}}$, where the window h is dispatched on a geometric grid,

$$H_{i,m}\left(N_{i,t-1}^\pi\right) \triangleq \left\{h_j \in \left\{1,\dots,N_{i,t-1}^\pi\right\} \mid h_{j+1} = \lceil m\cdot h_j\rceil \text{ and } h_1 = 1\right\} \quad \text{with } m>1.$$

When there is no confusion, we drop the dependency on $N_{i,t-1}^\pi$. This modification alone is not enough to reduce both the time and space complexity. Indeed, updating $\widehat{\mu}_i^h$ requires to replace the h-th last sample by the new one o_t. Hence, we need to store the t collected samples to be able to update any $\widehat{\mu}_i^h$ with $\mathscr{O}(1)$ complexity. Therefore, in `EFF_UPDATE`, we will use $\mathscr{O}(K\log(t))$ *delayed* statistics that we can update with $\mathscr{O}(K\log(t))$ space and time complexity.

`EFF_UPDATE` (Alg. 8) takes as input the new observation o_t that the learner gets at the N_i-th pull of arm i; the geometric window grid $H_{i,m}$ tuned with an hyperparameter $m>1$, and for each window h_j in this grid, three different numbers $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$, $p_i^{h_j}$, $n_i^{h_j}$. $\left\{\widehat{\mu}_{i,\texttt{eff}}^{h_j}\right\}_{i,h_j}$ represents the set of *current* statistics of window size h_j that will be used instead of $\left\{\widehat{\mu}_i^h\right\}_{i,h}$ in our efficient algorithms. We also store a pending statistic $p_i^{h_j}$ and a count $n_i^{h_j}$ which are used in the sparse update procedure of $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$. `EFF_UPDATE` outputs an updated set of statistics. The core of `EFF_UPDATE` is divided in four parts: 1) From Lines 1 to 6, we create new window's statistics at a logarithmic rate with respect to the growth of N_i; 2) From Lines 7 to 9, we update the statistics of window $h_1=1$; 3) From Lines 10 to 13, we update the other pending statistics and count; 4) From Lines 14 to 20, we eventually update $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ and refresh the corresponding pending statistic and count. The remaining details are quite technical. Thus, we first give the high-level properties that are ensured by the recursive usage of `EFF_UPDATE`. Then, we prove them by going through the algorithm line by line.

Algorithm 8 `EFF_UPDATE`

Require: o_t, $H_{i,m} \leftarrow \left\{h_j < \lceil m \cdot N_i \rceil \mid h_{j+1} = \lceil m \cdot h_j \rceil \text{ with } h_0 = 1\right\}$, $\left\{\left\{\widehat{\mu}_{i,\texttt{eff}}^{h_j}, p_i^{h_j}, n_i^{h_j}\right\}\right\}_{h_j \in H_{i,m}}$

if $N_i = \max(H_{i,m})$ **then** ▷ Create a new triplet with window $h_j = \lceil m \cdot N_i \rceil$
  $H_{i,m} \leftarrow H_{i,m} \cup \{\lceil m \cdot N_i \rceil\}$
  $p_i^{\lceil m \cdot N_i \rceil} = p_i^{N_i}$
  $n_i^{\lceil m \cdot N_i \rceil} \leftarrow n_i^{N_i}$
  $\widehat{\mu}_{i,\texttt{eff}}^{\lceil m \cdot N_i \rceil} \leftarrow \texttt{None}$
end if
$p_i^1 \leftarrow o_t$ ▷ Update the first triplet with o_t
$n_i^1 \leftarrow 1$
$\widehat{\mu}_{i,\texttt{eff}}^1 \leftarrow o_t$
for $h_j \in H_{i,m} \setminus \{1\}$ **do** ▷ Update the other pending statistics $p_i^{h_j}$ and $n_i^{h_j}$
  $p_i^{h_j} \leftarrow p_i^{h_j} + o_t$
  $n_i^{h_j} \leftarrow n_i^{h_j} + 1$
end for
for $h_j \in \textsc{Sort\_Desc}(H_{i,m} \setminus \{1\})$ **do**
  if $n_i^{h_j} = h_j$ **then**
    $\widehat{\mu}_{i,\texttt{eff}}^{h_j} \leftarrow p_i^{h_j}/h_j$ ▷ Replace the current statistic $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$
    $p_i^{h_j} = p_i^{h_{j-1}}$ ▷ Refresh the pending statistics
    $n_i^{h_j} \leftarrow n_i^{h_{j-1}}$
  end if
end for

Ensure: $\left\{\left\{\widehat{\mu}_{i,\texttt{eff}}^{h_j}, p_i^{h_j}, n_i^{h_j}\right\}\right\}_{h_j \in H_{i,m}}$

Proposition 4.5.2 $\left\{\left\{\widehat{\mu}_{i,\texttt{eff}}^{h_j}, p_i^{h_j}, n_i^{h_j}\right\}\right\}_{h_j \in H_{i,m}}$, constructed recursively with `EFF_UPDATE` with initial value $\left\{\left\{\widehat{\mu}_{i,\texttt{eff}}^1 : \texttt{None},\ p_i^1 : 0,\ n_i^1 : 0\right\}\right\}$ have the following properties:

1. $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ is the average of exactly h_j consecutive samples among the $2h_j - 1$ last ones.
2. The delay between two updates of $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ is in $\left\{\left\lceil \frac{m-1}{m} h_j \right\rceil, \dots, h_j - 1\right\}$.
3. $p_i^{h_j}$ is the sum of the $n_i^{h_j}$ last samples.
4. $n_i^{h_j} < h_j$ for $j \geq 1$. Also, $n_i^1 \leq 1$.
5. $\left\{n_i^{h_j}\right\}_{h_j}$ is an non-decreasing sequence with respect to h_j (or j).

Proof. The three last properties are trivially true at the initialization. Thus, we show by induction that they remain true after updates.

Proof of 3. At Lines 3 and 4, we create a new pending statistics and count by initializing them with other statistics and counts. Hence, because of the recursion hypothesis, all the pending statistics $p_i^{h_j}$ (including the created one) contains the sum of the $n_i^{h_j}$ *before last* pulls. At Lines 7 and 8, we update p_i^1 with the last sample and set n_i^1 to 1. At Lines 11 and 12, we add the last sample to $p_i^{h_j}$ (which was containing the before last samples) and increase the count by 1. Hence, at the end of Line 12, all the $p_i^{h_j}$ contains the sum of the last $n_i^{h_j}$ samples. Thus, refreshing $p_i^{h_j}$ and $n_i^{h_j}$ with $p_i^{h_{j-1}}$ and $n_i^{h_{j-1}}$ keeps this property true (Lines 17 and 18).

Proof of 4. For $j = 0$, n_i^1, which is equal to 0 at the initialization, is set at 1 at every update (Line 8). Hence, we have $n_i^{h_0} \leq h_0 = 1$. For $j \geq 1$, $n_i^{\lceil m \cdot N_i \rceil}$ is initialized at Line 4 with the value $n_i^{N_i} < N_i < \lceil m \cdot N_i \rceil$ by the induction hypothesis and because $m > 1$. Then, $n_i^{h_j} < h_j$ ($j \geq 1$) is increased by one at each update at Line 12. Hence, we now have $n_i^{h_j} \leq h_j$ for all $j \in H_{i,m}$. However, for $j \geq 1$, if $n_i^{h_j} = h_j$ (Line 15), it is replaced by the precedent count $n_i^{h_{j-1}} \leq h_{j-1} < h_j$ (Line 12). Thus, at the end of the update, we do have $n_i^{h_j} < h_j$ for $j \geq 1$.

Proof of 5. At Line 4, we create a new pending count corresponding to the largest h_j and we initialize it with the precedent largest count. At Lines 8 and 12, we set $n_i^1 = 1$ and increase all the other $n_i^{h_j}$ by one. This operation preserves the non-decreasing property of the ordered set. Last, at Line 18, we set few counts $n_i^{h_j}$ to the precedent value $n_i^{h_{j-1}}$- which also preserves the non-decreasing property of the ordered set.

Proof of 1 and 2. Thanks to Property 3, we know that $p_i^{h_j}$ is the sum of the $n_i^{h_j}$ last sample. It is still true at the end of Line 12 (see the proof). Then, at Line 16, and given the condition in Line 15, we set $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ with the average of the last h_j sample. Then, $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ is not updated untill the condition at Line 15 is fulfilled again.

$n_i^{h_j}$ is refreshed with a quantity larger or equal to 1 and smaller or equal to h_{j-1} at Line 18. Then, it is increased by one at each update. we know that $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ will be updated at least every $h_j - 1$, and at most every $h_j - h_{j-1}$ round. Hence, considering the worst possible delay we can conclude: $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ is the average of exactly h_j consecutive samples among the $2h_j - 1$ last ones. Last, considering that $h_{j-1} \leq h_j/m$, we conclude that the minimal delay is larger or equal to $\frac{m-1}{m} h_j$. ■

In Proposition 4.5.3, we show that `EFF_UPDATE` succeeds to drastically reduce the time and space complexity of the updates as soon as m is not too close to 1.

Proposition 4.5.3 After N_i updates, the time and space complexity of `EFF_UPDATE` scales with $\mathcal{O}(\min(\log_m N_i, N_i))$.

Proof. The time and space complexity scales with $|H_{i,m}|$. Indeed, there are $3|H_{i,m}|+2$ variables store in memory : $\left\{\{\widehat{\mu}_{i,\texttt{eff}}^{h_j}, p_i^{h_j}, n_i^{h_j}\}\right\}_{h_j\in H_{i,m}}$, o_t and N_i. Moreover, `EFF_UPDATE` does two FOR loops on $H_{i,m}$.

The size of $H_{i,m}$ is upper-bounded by $\mathcal{O}(\log_m N_i)$. Indeed, when the condition at Line 1 is fulfilled, $\max(H_{i,m})$ is replaced by $\lceil m\cdot N_i\rceil$. When it is not fulfilled, the maximum is not changed but N_i increases by one unit. Hence, we always have,

$$\max(H_{i,m}) \leq \lceil m\cdot N_i\rceil .$$

Moreover, when the condition is fulfilled, the maximum is replaced by,

$$\max(H_{i,m}) \leftarrow \lceil m\cdot \max(H_{i,m})\rceil .$$

Hence, when $H_{i,m}$ is initialized with $\{1\}$, we show recursively that,

$$\max(H_{i,m}) \geq m^{|H_{i,m}|-1}.$$

Combining the above equations, we have,

$$m^{|H_{i,m}|-1} \leq m\cdot N_i + 1.$$

Therefore,

$$|H_{i,m}| \leq \log_m(2N_i)+2.$$

This upper-bound diverges at finite N_i when $m\to 1$. However, the size of $H_{i,m}$ is increased one by one at Line 2. Hence, even if the condition at Line 1 is fulfilled at every round, $|H_{i,m}|\leq N_i+1$ ($H_{i,m}$ is initialized with $\{1\}$). ■

R When $m\leq 1+\frac{1}{N_i}$, $\left\{\widehat{\mu}_{i,\texttt{eff}}^{h_j}\right\}_{h_j\in H_{i,m}}$ (the outcome of `EFF_UPDATE`) is the same than the outcome of the classical update $\left\{\widehat{\mu}_i^h\right\}_{h\leq N_i}$ for the N_i first rounds. Yet, there is no free lunch, the complexity is $\mathcal{O}(N_i)$ in this regime (according to Proposition 4.5.3).

4.5.3 The delay in `EFF_UPDATE`

We have already emphasized that `EFF_UPDATE` is built on two ideas: geometrically sparse and delayed statistics. The geometrically sparse aspect is straightforward to understand and quantify: $h_{j+1}\sim m\cdot h_j$ up to the rounding.

In this Subsection, we provide a tight analysis of the delay between the updates. In fact, the important quantity is the normalized delay, that is, the delay divided by the window size. Indeed, each statistic of window h_j should represent the h_j last sample. A delay of 10 for $h_j = 10^6$ is very good as it succeeds to take into account most of the h_j last sample.

However, the same delay of 10 is a failure when $h_j = 1$, as the algorithm fails to give a good representation of the very last sample.

In Property 2, we show that the normalized delay cannot be larger than 100%. We also show that it is lower bounded by $\frac{m-1}{m}$. For small values of m, this amplitude can be large.

We use two tricks in the algorithm to reduce the delay. First, at Line 18, we refresh $p_i^{h_j}$ with $p_i^{h_{j-1}}$ which contains $n_i^{h_{j-1}} \in \{1,\dots,h_{j-1}\}$ samples. We could refresh $p_i^{h_j}$ and $n_i^{h_j}$ at 0 which would lead to 100% normalized delay. Instead, we use the variable available in the memory which contains the sum of h last sample, with the largest $h < h_j$. According to Properties 3, 4 and 5, this quantity is $p_i^{h_{j-1}}$.

Indeed, according to Property 3, $p_i^{h_{j'}}$ contains the $n_i^{h_{j'}}$ last samples. At the round of the update of $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$, $p_i^{h_{j'}}$ contains h_j samples before Line 17. According to Property 5, all the $j' > j$ have $n_i^{h_{j'}} > n_i^{h_j} = h_j$. According to Property 4, $n_i^{h_{j-1}} \leq h_{j-1} < h_j$. Moreover, for all $j' \leq j-1$, $n_i^{h_{j-1}} \geq n_i^{h_{j'}}$ (Property 5).

The second trick is to sort $H_{i,m}$ in the decreasing order at Line 14. If there are two synchronous consecutive updates of $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ and $\widehat{\mu}_{i,\texttt{eff}}^{h_{j+1}}$ at the same run of `EFF_UPDATE`, doing a backward loop guarantees to refresh $n_i^{h_{j+1}}$ with $n_i^{h_j} = h_j$ instead of a smaller value if we would do a forward loop.

In this Subsection, we show that these two tricks succeed to upper-bound the normalized delay by $\mathcal{O}\left(\frac{m-1}{m}\right)$ when m is an integer and when $m < 2$.

The integer case.

Proposition 4.5.4 When $m \in \mathbb{N} \setminus \{0,1\}$, $h_j = m^j$. Moreover, $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ is updated periodically with period $\omega_j = \frac{m-1}{m} h_j$ for $j \geq 1$ ($\omega_0 = 1$).

The main idea is simple: since any window h_j is a multiple of the lower order h_{j-k}, $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ is initialized and updated synchronously with all the lower order statistics. Hence, the pending statistic $p_i^{h_j}$ is refreshed with h_{j-1} sample, the largest possible number of sample in $p_i^{h_{j-1}}$. Hence, choosing integer values minimizes the delay (compared to the delay bounds we identify in Property 2). The proof is quite technical, and we delay it at the end of the Subsection.

We call $d_j(N_i) \in \{0,\dots,\omega_j - 1\}$, the number of pulls since the last update of statistic $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ after N_i pulls. We display in Figure 4.6 the normalized delay d_j/h_j after N_i pulls of each statistic. The updates are indeed periodic. We notice the strong synchronization in the updates: not only each period ω_j is at a m factor of the previous one, but the update of statistic j are at the same round as the updates of statistics $j' < j$.

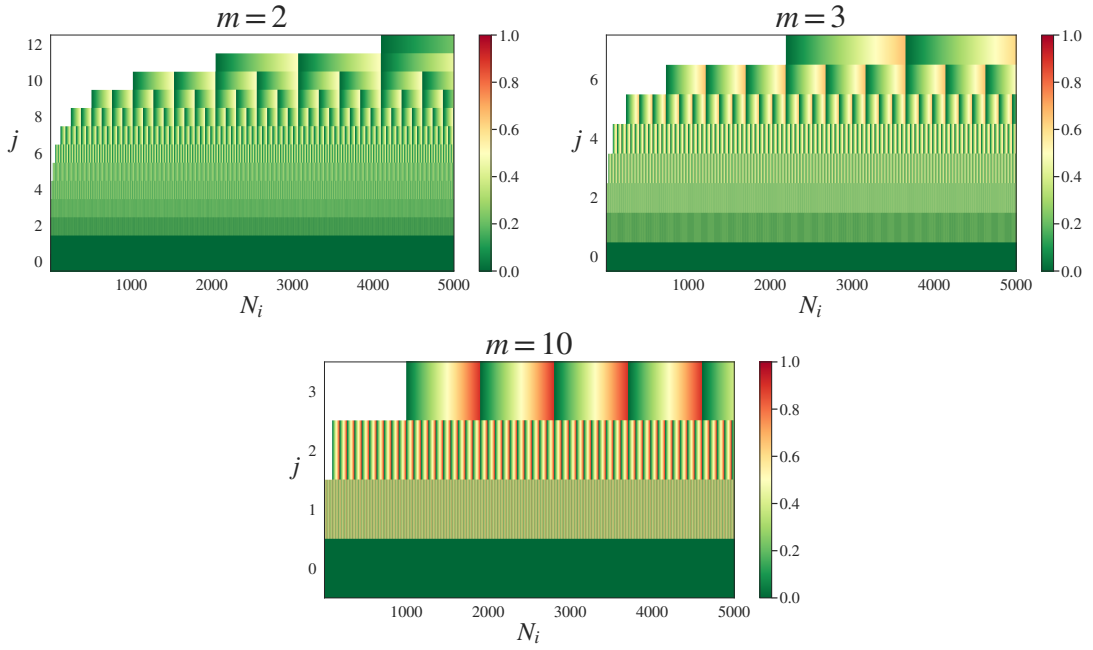


Figure 4.6: Normalized delay d_j/h_j after N_i pulls for each j-th statistic $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$. We display in white the rounds at which statistic j is not created yet.

However, for large values of m, the delay improvement is marginal. For $m = 10$, each statistic can be delayed by 90% their window size. Even, for $m = 2$ the normalized delay ω_j/h_j is 50%. The $m-1/m$ ratio would be very interesting for $m \to 1$.

The non integer case

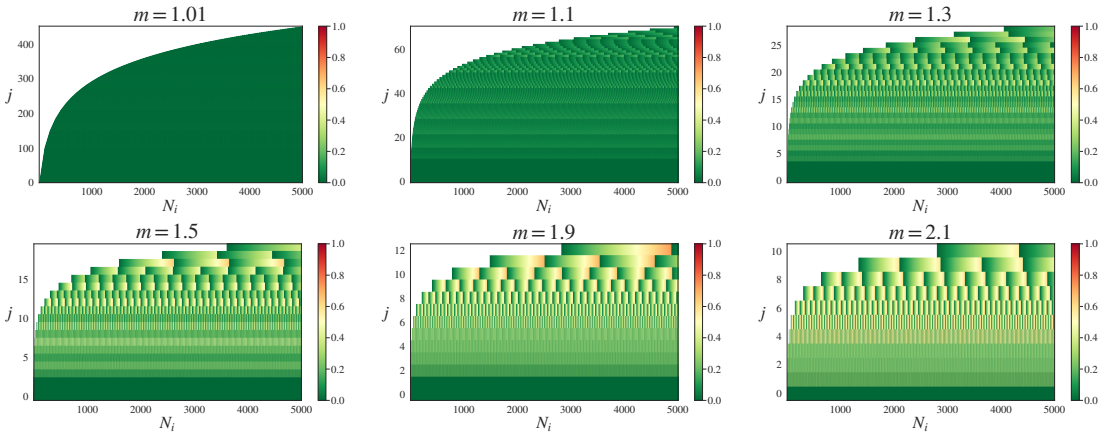


Figure 4.7: Normalized delay d_j/h_j after N_i pulls for each j-th statistic $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$. We display in white the rounds at which statistic j is not created yet.

In Figure 4.7, we display the delay for several non integer values. Compared to the integer case, the update of statistic j does not happen at the same round as the update of statistic $j' < j$. However, we notice that the updates are still periodic and the updating period ω_j is

a multiple of ω_{j-1}. We formalized this properties in Propositions 4.5.5 and 4.5.6 which we show at the end of the Subsection.

Proposition 4.5.5 For each statistic, the updates are periodic. Moreover, the update period ω_{j+1} is a multiple of period ω_j,

$$\omega_{j+1} = \omega_j \left(1 + \left\lfloor \frac{h_{j+1} - h_j - 1}{\omega_j} \right\rfloor\right).$$

Proposition 4.5.6 For $m < 2$, ω_{j+1} is either equal to ω_j or to $2 \cdot \omega_j$.

We notice that this weaker synchronization can lead to a larger normalized delay. Indeed, for $m = 2$ the normalized delay is bounded by 50% (Fig. 4.6 and Proposition 4.5.4) while for $m = 1.9$ and $m = 2.1$ some statistics are delayed by more than 70% (Fig 4.7). Yet, for $m \to 1$, the normalized delay seems to converge to 0. Indeed, we prove in Proposition 4.5.7 that the normalized period cannot exceed twice its minimal value $m-1/m$.

Proposition 4.5.7 For $m < 2$, either $\omega_j = 1$ or $\omega_j < \frac{2(m-1)}{m} h_j$.

Notice that when $\omega_j = 1$, there is zero delay in the updates (statistics are updated at every round). We investigate empirically whether this upper-bound is tight. We select ten thousand values of m uniformly at random between 1 and 2 and we add the value $m = 2$. For each value of m, we compute recursively all the h_j and ω_j (with Proposition 4.5.5) until $h_j > 10^{15}$. Then, we compute the ratio,

$$r_j \triangleq \frac{m\omega_j}{(m-1)\,h_j},$$

for any j such that $\omega_j \neq 1$. According to Proposition 4.5.7 and Property 2, this ratio always lies between 1 and 2.

In Figure 4.8, we display for each value of m the maximum, minimum, median and average of the sequence $\{r_j\}_j$. Notice that $h_j = 10^{15}$ is much larger than the horizon usually considered in bandits experiments, even to characterize asymptotic performance (Chapelle and Li 2011; Kaufmann et al. 2012a; Lattimore 2018). Hence, the displayed minimum and maximum are valid empirical bounds for real application.

For more than 90% of the values of m, the minimum is below 1.02, the maximum is larger than 1.95, and the median and mean are between 1.35 and 1.45. It shows that our theory is tight in general to characterize the best and worst possible normalized period.

There are deviations to this general case. First, when $m \to 2$, the ratio tends to 1 for all j. This is indeed the value when $m = 2$. When we compare $m = 1.9$ and $m = 2$ on Figures 4.6 and 4.7, we see that the normalized delay is drifting for $m = 1.9$: the updates are synchronous and the normalized delay is $\sim 50\%$ for the first statistics, but it becomes larger when j is increasing. We conjecture that the closer m is to 2, the slower is the drift.

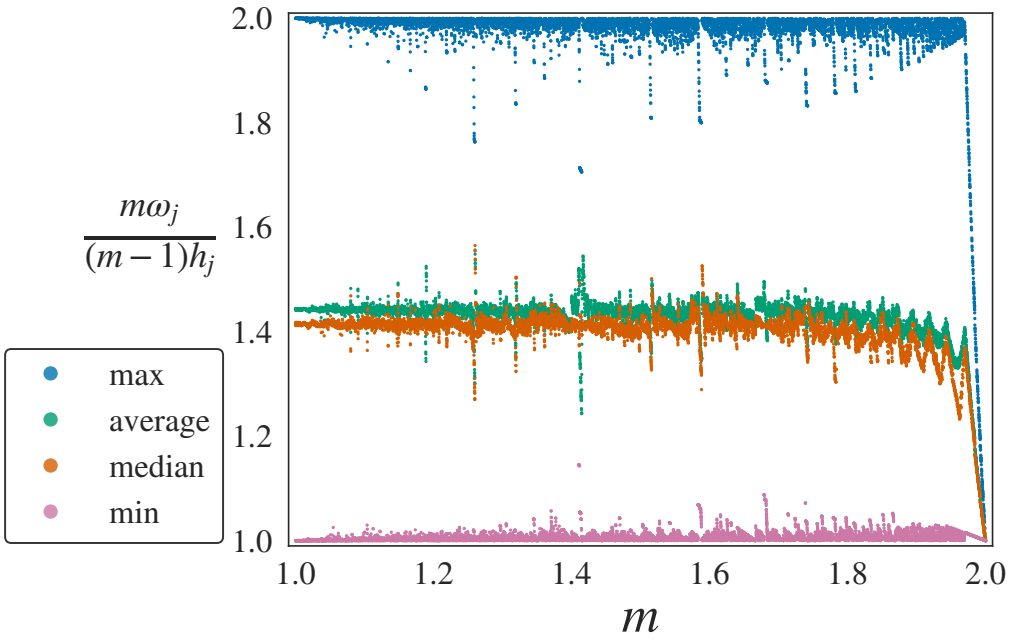


Figure 4.8: Impact of m on the minimum, maximum, average and median ratio among $\left\{m\omega_j/(m-1)h_j\right\}_j$.

Second, there are also local deviations (e.g near $m \sim 1.42$). They correspond to values of m such that m^k (with k a small integer) is a power of two. In that case, the ratios $\left\{r_j\right\}_j$ are cycling in the regime $h_j >> 1$ (*i.e.* when the rounding effect is negligible and $h_{j+1} \sim m \cdot h_j$) and take only k values up to small rounding perturbations. These values can either be quite good or quite bad. In fact, due to the rounding, these values are slowly drifting. We can try to control the drift by increasing or decreasing m very slightly to improve the median or the average delay for a given horizon.

As we can see on Figure 4.8, it is very sensitive to the exact value of m. For instance, with $\varepsilon = 1e^{-5}$, $m = (1-\varepsilon) \times 2^{1/3}$ has an average ratio of 1.29 while $m = (1+\varepsilon) \times 2^{1/3}$ has an average ratio of 1.55. Indeed, the normalized delay is not the same for a statistic which is updated just before the precedent one ($r_j \sim 1$) or for one which is updated just after ($r_j \sim 2$) . When the update of $p_i^{h_j}$ is just before, it is refreshed with almost h_{j-1} samples, which is the best possible value. When it is just after, it is refreshed with $\sim h_{j-2}$ which is close to the worse one. Due to this discontinuity, it is hard to take advantage of these local deviation.

To conclude, non-integer values for m leads to a larger ratio r_j than integer values (twice larger in the worst case, ~ 1.4 in average). However, the interesting quantity is the normalized period, which is equal to $(m-1)r_j/m$. For $m = 2$, the normalized period is 50% for all the statistics $j \geq 1$. In order to achieve a lower value in the worst case, one should choose $m \leq \frac{4}{3}$. If we target a lower value in average, one should choose $m \leq 1.56$. It shows that non-integer values are especially interesting when $m \to 1$. Yet, there is no free lunch:

the complexity of EFF_UPDATE scales with $\mathcal{O}(\log_m T)$ which diverges with $1/m-1$ when $m \to 1$.

Proofs

Proof of Proposition 4.5.4. When m has an integer value, we have,

$$h_j = \lceil m \cdot h_{j-1} \rceil = m \cdot h_{j-1} = \cdots = m^j \cdot h_0 = m^j.$$

For $j = 0$, $\widehat{\mu}^1_{i,\texttt{eff}}$ is updated at every update at Line 9. Hence, $\omega_0 = 1$. For $j = 1$, $h_1 = m$ is initialized after m pulls. At this round, we set $n_i^{h_1}$ to the value in $n_i^{h_0}$ which is equal to 1. Indeed, the first statistic is always up to date. Hence, the next update is after $m-1$ pulls. At this round, the pending statistics is again refreshed with p_0 which contains 1 sample and, recursively, we can conclude that $\widehat{\mu}^{h_1}_{i,\texttt{eff}}$ is updated every $\omega_1 = m-1 = \frac{m-1}{m} h_1$.

By induction, let j such that the statistic $j-1$ is updated periodically every $\omega_{j-1} = (m-1)\,m^{j-2}$ pulls from pull m^{j-1}. $\widehat{\mu}^{h_j}_{i,\texttt{eff}}$ is initialized after m^j pulls (Line 16) . It is synchronized with the m-th update of statistic $\widehat{\mu}^{h_{j-1}}_{i,\texttt{eff}}$. Indeed,

$$m^j = m \cdot m^{j-1} = m^{j-1} + m\omega_{j-1}.$$

Notice that we sort $H_{i,m}$ in the decreasing order at Line 14, hence $n_i^{h_j}$ is updated with $n_i^{h_{j-1}} = m^{j-1}$ before it is itself refreshed with $n_i^{h_{j-2}}$ (Line 18). Hence, $\widehat{\mu}^{h_j}_{i,\texttt{eff}}$ is updated for the first time after $\omega_j = h_j - n_i^{h_j} = m^j - m^{j-1} = (m-1)m^{j-1} = m\omega_{j-1}$ pulls, *i.e.* after $m^j + (m-1)m^{j-1}$ pulls of arm i. Again, this update is synchronized with the update of the lower order statistic:

$$m^j + \omega_j = m^j + (m-1)m^{j-1} = m^{j-1} + 2m\omega_{j-1}.$$

Hence, the pending statistic $p_i^{h_j}$ is again refreshed with $n_i^{h_{j-1}} = m^{j-1}$ sample. Recursively, we can repeat the very same argument and show that $\widehat{\mu}^{h_j}_{i,\texttt{eff}}$ is updated every $\omega_j = (m-1)\,m^{j-1}$ pulls from pull m^j. ■

Proof of Proposition 4.5.5. We will prove this property by induction on j. When $j = 0$, the updates happen at every round. Hence, $\omega_0 = 1$. Let j such that $\widehat{\mu}^{h_{j-1}}_{i,\texttt{eff}}$ is refreshed periodically with period ω_{j-1}. $\widehat{\mu}^{h_j}_{i,\texttt{eff}}$ is initialized after h_j pulls. At that round $p_i^{h_j}$ is initialized with the current value of $p_i^{h_{j-1}}$ which contains $n_i^{h_{j-1}}$ samples. Since statistic $j-1$ is updated with period ω_{j-1}, $n_i^{h_{j-1}}$ takes its value between $h_{j-1} - \omega_{j-1} + 1$ and h_{j-1}. At pull h_{j-1}, it was initialized with value $h_{j-1} - \omega_{j-1} + 1$. Then, it is increased by one at every pull and refresh at $h_{j-1} - \omega_{j-1} + 1$ when it reaches value h_j. Therefore, at pull h_j, we have,

$$n_i^{h_{j-1}}(h_j) = h_{j-1} - \omega_{j-1} + 1 + \left(h_j - h_{j-1} - 1 \mod \omega_{j-1}\right). \tag{4.37}$$

with $n_i^{h_{j-1}}(h)$, the value of $n_i^{h_{j-1}}$ at the end of the h-th pulls of arm i. The -1 is caused by the backward loop (Line 14: when $h_j - h_{j-1} \mod \omega_{j-1}$, the updates are synchronized such that it minimizes the delay (like in the integer case). The next update of $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ will happen in

$$\begin{aligned}
\omega_j &\triangleq h_j - n_i^{h_{j-1}}(h_j) \\
&= h_j - \left(h_{j-1} - \omega_{j-1} + 1 + \left(h_j - h_{j-1} - 1 \mod \omega_{j-1}\right)\right) \\
&= \omega_{j-1} + h_j - h_{j-1} - 1 - \left(h_j - h_{j-1} - 1 \mod \omega_{j-1}\right) \\
&= \omega_{j-1}\left(1 + \left\lfloor \frac{h_j - h_{j-1} - 1}{\omega_{j-1}} \right\rfloor\right)
\end{aligned}$$

The second line is justified by Equation 4.37. The last line uses $a - a \mod b = b\lfloor a/b \rfloor$. Hence, the first delay ω_j is a multiple of ω_{j-1}. Therefore, $n_i^{h_{j-1}}(h_j + \omega_j) = n_i^{h_{j-1}}(h_j)$ and $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ is refreshed with the same number of sample than its initialization, and the delay until the second update is $h_j - n_i^{h_{j-1}}(h_j + \omega_j) = n_i^{h_{j-1}}(h_j) = \omega_j$. Recursively, we show that $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ is updated periodically with period ω_j. ■

Proof of Proposition 4.5.6. By *reductio ad absurdum*, we consider the smallest $j \geq 1$ such that $\omega_{j+1} > 2 \cdot \omega_j$. A necessary and sufficient condition according to Proposition 4.5.5 is that

$$h_{j+1} - h_j - 1 \geq 2\omega_j. \tag{4.38}$$

When $1 < m < 2$, $h_0 = 1$, $\omega_0 = 1$ (as for any m), $h_1 = \lceil m \cdot h_0 \rceil = 2$, and $\omega_1 = 1$ (according to Prop 4.5.5). Hence, $j \geq 1$. Since $j \geq 1$ is the smallest value such that $\omega_{j+1} > 2 \cdot \omega_j$, we have that either $\omega_j = \omega_{j-1}$ or $\omega_j = 2 \cdot \omega_{j-1}$. If $\omega_j = \omega_{j-1}$, we have according to Proposition 4.5.5,

$$h_j - h_{j-1} - 1 < \omega_{j-1} = \omega_j.$$

If $\omega_j = 2 \cdot \omega_{j-1}$, we have with the same argument,

$$h_j - h_{j-1} - 1 < 2 \cdot \omega_{j-1} = \omega_j.$$

Since h_j, h_{j-1} and ω_j are integers, we have

$$h_j - h_{j-1} - 1 < \omega_j \implies h_j - h_{j-1} \leq \omega_j. \tag{4.39}$$

Using $h_{j+1} = \lceil m \cdot h_j \rceil$,

$$h_{j+1} - h_j - 1 = \lceil m \cdot h_j \rceil - \lceil m \cdot h_{j-1} \rceil - 1 \leq m\left(h_j - h_{j-1}\right) \tag{4.40}$$

Plugging Equations 4.38, 4.39 and 4.40,

$$m\left(h_j - h_{j-1}\right) \geq 2\left(h_j - h_{j-1}\right).$$

This is impossible for $m < 2$ and $h_j > h_{j-1}$ (which is the case when $m > 1$). Hence, we conclude that there exists no integer j such that $\omega_{j+1} > 2 \cdot \omega_j$.

■

Proof of Proposition 4.5.7. We want to upper bound the ratio ω_j/h_j for all j such that $\omega_j > 1$. When $m < 2$ we have either $\omega_j = \omega_{j-1}$ or $\omega_j = 2 \cdot \omega_{j-1}$ (Prop. 4.5.6). We first study the case where $\omega_j = 2 \cdot \omega_{j-1}$, *i.e.* (Prop; 4.5.5),

$$h_j - h_{j-1} - 1 \geq \omega_{j-1} = \omega_j/2.$$

Using $h_j = \lceil m \cdot h_{j-1} \rceil \implies h_{j-1} = \lfloor h_j/m \rfloor$,

$$h_j - h_{j-1} - 1 = h_j - \lfloor h_j/m \rfloor - 1 \leq \frac{m-1}{m} h_j.$$

Plugging the two last equations leads to,

$$\frac{\omega_j}{h_j} \leq \frac{2\,(m-1)}{m}.$$

We notice that $\left\{h_j\right\}_{j \in \mathbb{N}}$ is an increasing sequence. When $\omega_j = \omega_{j-1}$, we have $\omega_j/h_j < \omega_{j-1}/h_{j-1}$. Therefore, for any j such that $d_j > 1$ we can find the largest $j' \leq j$ such that $\omega_{j'} = 2 \cdot \omega_{j'-1}$ and compare,

$$\frac{\omega_j}{h_j} \leq \frac{\omega_j}{h'_j} = \frac{\omega'_j}{h'_j} \leq \frac{2\,(m-1)}{m}.$$

■

4.5.4 `EFF-FEWA` (π_{EF}) and `EFF-RAW-UCB` (π_{ER})

`EFF-FEWA` (π_{EF}) and `EFF-RAW-UCB` (π_{ER}) are the two efficient versions of our initial algorithms. With an hyperparameter $m > 1$, they use `EFF_UPDATE` instead of `UPDATE` (Lines 4 and 18 in `FEWA` and Lines 4 and 9 in `RAW-UCB`). Therefore, they use $\left\{\widehat{\mu}^{h_j}_{i,\mathtt{eff}}\right\}_{i,h_j \in H_{i,m}}$ instead of $\left\{\widehat{\mu}^{h}_{i}\right\}_{i,h \leq N_{i,t-1}}$.

More precisely, in `FEWA`, we replace the increment $h \leftarrow h+1$ by $h \leftarrow \lceil m \cdot h \rceil$ at Line 12. Hence, the next set is not called $\mathscr{K}_{h+1}$ but $\mathscr{K}_{\lceil m \cdot h \rceil}$ (Line 11 in `FEWA` and Line 6 in `FILTER`). Finally, at Lines 13 and 14, the condition is not $N_{i_t} = h$ but $N_{i_t} \leq h$. In the `FILTER` procedure, we also change $\widehat{\mu}^{h}_{i}$ by $\widehat{\mu}^{h}_{i,\mathtt{eff}}$ at Lines 2 and 4. In `RAW-UCB`, we only change the $h \leq N_i$ by $h_j \in H_{i,m}$ and $\widehat{\mu}^{h}_{i}$ by $\widehat{\mu}^{h_j}_{i,\mathtt{eff}}$ in the index computation at Line 7.

Proposition 4.5.8 At any round t, EFF-FEWA and EFF-RAW-UCB tuned with hyperparameter m have a $\mathcal{O}(K\log_m(t))$ worst-case time and space complexity.

Proof. For each arm, the algorithms use the statistics created and maintained by EFF_UPDATE plus a handful of variables (such as t). Hence, the space complexity is the sum of the complexities of EFF_UPDATE (see Prop. 4.5.3) for each arm, *i.e.*

$$\sum_{i\in\mathcal{K}} \mathcal{O}(\log_m(N_{i,T})) \leq \mathcal{O}(K\log_m(T)).$$

At every round t, the algorithms do one call of EFF_UPDATE, which costs at most $\mathcal{O}(\log t)$. For each of the $\mathcal{O}(K\log_m t)$, EFF-RAW-UCB computes one ucb with unit cost $\mathcal{O}(1)$. For each of the K arms, we find the minimum ucb among the $\mathcal{O}(\log_m t)$ ones. It costs $\mathcal{O}(K\log_m t)$ in total. Finally, we select the arm with the largest index, which costs $\mathcal{O}(K)$. Hence, the worst-case time complexity at any round t is $\mathcal{O}(K\log_m t)$.

EFF-FEWA uses the procedure FILTER at most for each existing window, *i.e.* $\mathcal{O}(\log_m(t))$. The inner time complexity of FILTER scales with $|\mathcal{K}_h| \leq K$. Therefore, in the worst case, the time complexity of EFF-FEWA at any round t is also bounded by $\mathcal{O}(K\log_m(t))$. ■

4.5.5 Regret analysis

In our analysis, the particularities of RAW-UCB and FEWA only appear in Proposition 4.2.1 and Corollary 4.3.1. We will derive analogous results for EFF-RAW-UCB and EFF-FEWA when $m = 2$. The upper-bounds will directly follow with no additional effort. We discuss the case $m \neq 2$ at the end of this Subsection.

A favorable event for efficiently updated adaptive windows

Proposition 4.5.9 For any round t and confidence $\delta_t \triangleq 2t^{-\alpha}$, let

$$\xi^{\alpha}_{t,m} \triangleq \left\{\forall i\in\mathcal{K},\ \forall n\leq t-1,\ \forall h_j \in H_{i,m}(n), \left|\widehat{\mu}^{h_j}_{i,\texttt{eff}}(t,\pi) - \overline{\mu}^{h_j}_{i,\texttt{eff}}(t,\pi)\right| \leq c(h_j,\delta_t)\right\}$$

be the event under which the estimates at a round t are all accurate up to $c(h,\delta_t) \triangleq \sqrt{2\sigma^2\log(2/\delta_t)/h}$. Then, for a policy π which pulls each arms once at the beginning, and for all $t > K$,

$$\mathbb{P}\left[\overline{\xi^{\alpha}_{t,2}}\right] \leq 3Kt\delta_t = 6Kt^{1-\alpha}.$$

(R) The probability of the unfavorable event $\overline{\xi^{\alpha}_{t,2}}$ scales with $\mathcal{O}\left(t^{1-\alpha}\right)$ compared to $\mathcal{O}\left(t^{2-\alpha}\right)$ for $\overline{\xi^{\alpha}_{t}}$ because the efficient algorithms construct less statistics. It means

that our theory will hold for a wider range of α. Yet, this benefits is only theoretical. The union bound in Proposition 4.2.1 is not tight because the different events share the same data. In practice, it leads to conservative tuning of the confidence bounds and one can decrease α to get better performance.

Proof. As in Propositions 4.1.5 and 4.2.1, we have to count the number of statistics that are required to hold in the confidence region. We call $u_j(n)$ the number of different values taken by variable $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ after t. According to Proposition 4.5.4, $u_0(n) = n \leq t$ because statistic 0 is created at the first round and updated at every round ($\omega_0 = 1$). For $m = 2$ and $j \geq 1$, $\widehat{\mu}_{i,\texttt{eff}}^{h_j}$ is created after $h_j = 2^j$ pulls and then updated every $\omega_j = 2^{j-1}$. Hence,

$$\forall j \geq 1 \text{ and } h_j \leq n, u_j(n) = 1 + \left\lfloor \frac{n - h_j}{\omega_j} \right\rfloor \leq \frac{n}{2^{j-1}} - 1 \leq \frac{n}{2^{j-1}}.$$

We do the union bound,

$$\begin{aligned}
\mathbb{P}\left[\overline{\xi_{t,2}^{\alpha}}\right] &\leq \sum_{i \in \mathscr{K}} \sum_{j=0}^{|H_{i,m}(N_{i,t})|} u_j(t)\delta_t \\
&\leq \sum_{i \in \mathscr{K}} \left(t + \sum_{j=1}^{|H_{i,m}(N_{i,t})|} \frac{t}{2^{j-1}} \right) \delta_t \\
&\leq 3Kt\delta_t.
\end{aligned}$$

■

Lemma 4.5.10 At any round t on favorable event $\xi_{t,2}^{\alpha}$, if arm i_t is selected by $\pi \in \{\pi_{\text{EF}}, \pi_{\text{ER}}\}$ tuned with $m = 2$, for any $h \leq N_{i,t-1}$, the average of its h last pulls cannot deviate significantly from the best available arm at that round, i.e.,

$$\overline{\mu}_{i_t}^{h}(t-1,\pi) \geq \max_{i \in \mathscr{K}} \mu_i(t, N_{i,t-1}) - \frac{C_\pi}{\sqrt{2\alpha}} c(h, \delta_t) \quad \text{with} \quad \begin{cases} C_{\pi_{\text{ER}}} = \frac{4\sqrt{\alpha}}{\sqrt{2}-1} \\ C_{\pi_{\text{EF}}} = \frac{8\sqrt{\alpha}}{\sqrt{2}-1} \end{cases}.$$

Proof. Like for Lemma 4.2.2 (see its proof), our proof is done in a more general rotting framework that can be used in the next chapter. We denote by $\overline{\mu}_i^{hh'}(t-1,\pi)$ and $\widehat{\mu}_i^{hh'}(t-1,\pi)$ the true mean and empirical average associated to the $h'-h$ samples between the h-th last one (included) and the h'-th last one (excluded). Let $j_h \in \mathbb{N}^\star$ such that: $2^{j_h} - 1 \leq h < 2^{j_h+1}$.

$$\overline{\mu}_{i_t}^{h}(t-1,\pi) \geq \overline{\mu}_{i_t}^{2^{j_h}-1}(t,\pi) = \sum_{j=0}^{j_h-1} \frac{2^j}{2^{j_h}-1} \overline{\mu}_{i_t}^{2^j 2^{j+1}}(t,\pi). \tag{4.41}$$

The inequality follows because the reward is decreasing and $h \geq 2^{j_h} - 1$. Then, we decompose the average in a weighted sum of averages of geometrically expanding windows. Since the reward is decreasing we have that,

$$\forall k \leq 2^j, \quad \overline{\mu}_{i_t}^{2^j 2^{j+1}}(t,\pi) \geq \overline{\mu}_{i_t}^{k:k+2^j}(t,\pi).$$

$\widehat{\mu}^{h_j}_{i_t,\texttt{eff}}$ contains 2^j samples among the $2^{j+1}-1$ last ones (see Proposition 4.5.2). Setting $k \leq 2^j$ to the current delay of the statistics $\widehat{\mu}^{h_j}_{i_t,\texttt{eff}}$ (see Point 2 in Proposition 4.5.2), we can write,

$$\overline{\mu}^{2^j 2^{j+1}}_{i_t}(t,\pi) \geq \overline{\mu}^{k:k+2^j}_{i_t}(t,\pi) = \overline{\mu}^{h_j}_{i_t,\texttt{eff}} \geq \widehat{\mu}^{h_j}_{i_t,\texttt{eff}} - c(2^j,\delta_t), \tag{4.42}$$

where we use that we are on $\xi^{\alpha}_{t,2}$ for the last inequality. Therefore, gathering Equations 4.41 and 4.42,

$$\overline{\mu}^{h}_{i_t}(t,\pi) \geq \sum_{j=0}^{j_h-1} \frac{2^j}{2^{j_h}-1}\left(\widehat{\mu}^{h_j}_{i_t,\texttt{eff}} - c(2^j,\delta_t)\right). \tag{4.43}$$

Now, we will use the mechanics of the two algorithms. On the first hand, for `EFF-RAW-UCB`, we make the index appear in the inequality,

$$\begin{aligned}
\overline{\mu}^{h}_{i_t}(t,\pi_{\mathrm{ER}}) &\geq \sum_{j=0}^{j_h-1} \frac{2^j}{2^{j_h}-1}\left(\widehat{\mu}^{h_j}_{i_t,\texttt{eff}} - c(2^j,\delta_t)\right) \\
&= \sum_{j=0}^{j_h-1} \frac{2^j}{2^{j_h}-1}\left(\widehat{\mu}^{h_j}_{i_t,\texttt{eff}} + c(2^j,\delta_t) - 2c(2^j,\delta_t)\right) \\
&\geq \min_{j\in H_{i_t,2}}\left(\widehat{\mu}^{h_j}_{i_t,\texttt{eff}} + c(2^j,\delta_t)\right) - 2\sum_{j=0}^{j_h-1}\frac{2^j}{2^{j_h}-1}c(2^j,\delta_t).
\end{aligned} \tag{4.44}$$

Then, we can relate the left part of the sum to the best current value $\mu_{i^\star_t}(t,N_{i^\star_t,t-1})$,

$$\min_{j\in H_{i_t,2}}\left(\widehat{\mu}^{h_j}_{i_t,\texttt{eff}} + c(2^j,\delta_t)\right) \geq \min_{j\in H_{i^\star_t,2}}\left(\widehat{\mu}^{h_j}_{i^\star_t,\texttt{eff}} + c(2^j,\delta_t)\right) \geq \overline{\mu}^{h_{\min}}_{i^\star_t,\texttt{eff}} \geq \mu_{i^\star_t}(t,N_{i^\star_t,t-1}). \tag{4.45}$$

where $h_{\min} \in \arg\min_{h_j\in H_{i^\star_t,2}}\left(\widehat{\mu}^{h_j}_{i^\star_t,\texttt{eff}} + c(h_j,\delta_t)\right)$.The first inequality follows because `EFF-RAW-UCB` selects the arm i_t with the largest index. In particular, the index of i_t is larger or equal to the index of $i^\star_t \in \arg\max_{i\in\mathcal{K}}\mu_i(t,N_{i^\star_t,t})$. The second inequality holds on $\xi^{\alpha}_{t,2}$. The third inequality uses the decreasing of the reward. Putting Equations 4.44 and 4.45, we get,

$$\overline{\mu}^{h}_{i_t}(t,\pi_{\mathrm{ER}}) \geq \mu_{i^\star_t}(t,N_{i^\star_t,t-1}) - 2\sum_{j=0}^{j_h-1}\frac{2^j}{2^{j_h}-1}c(2^j,\delta_t). \tag{4.46}$$

On the other hand, for `EFF-FEWA`, we know that the selected arm passes any filter of window $2^j \in H_{i_t,2}$. Therefore, with $i_{\max} \in \arg\max_{i\in\mathcal{K}_{h_j}}\overline{\mu}^{h_j}_{i,\texttt{eff}}$, we can write,

$$\begin{aligned}
\widehat{\mu}^{h_j}_{i_t,\texttt{eff}} &\geq \max_{i\in\mathcal{K}_{h_j}}\widehat{\mu}^{h_j}_{i,\texttt{eff}} - 2c\left(h_j,\delta_t\right) && \text{Filtering rule} \\
&\geq \widehat{\mu}^{h_j}_{i_{\max},\texttt{eff}} - 2c\left(h_j,\delta_t\right) && i_{\max}\in\mathcal{K}_{h_j} \\
&\geq \overline{\mu}^{h_j}_{i_{\max},\texttt{eff}} - 3c(h_j,\delta_t) && \text{on } \xi^{\alpha}_{t,2} \\
&= \max_{i\in\mathcal{K}_{h_j}}\overline{\mu}^{h_j}_{i,\texttt{eff}} - 3c\left(h_j,\delta_t\right).
\end{aligned} \tag{4.47}$$

We relate $\overline{\mu}^{h_j}_{i,\texttt{eff}}$ to the largest available value at the round t,

$$\max_{i\in\mathscr{K}_{h_j}} \overline{\mu}^{h_j}_{i,\texttt{eff}} \geq \max_{i\in\mathscr{K}_1} \overline{\mu}^{1}_{i,\texttt{eff}} = \max_{i\in\mathscr{K}} \overline{\mu}^{1}_{i,\texttt{eff}} \geq \overline{\mu}^{1}_{i^\star_t,\texttt{eff}} \geq \mu_{i^\star_t}(t, N_{i^\star_t,t-1}). \tag{4.48}$$

The last inequality follows from the decreasing of the reward and the before last from the definition of the maximum operator. The first one uses a similar argument than in Lemma 4.2.2: $\max_{i\in\mathscr{K}_{h_j}} \overline{\mu}^{h_j}_{i,\texttt{eff}}$ increases with h_j. Indeed, on $\xi^{\alpha}_{t,2}$,

$$i_j \in \arg\max_{i\in\mathscr{K}_{h_j}} \overline{\mu}^{h_j}_{i,\texttt{eff}} \subset \mathscr{K}_{h_{j+1}},$$

because it cannot be at more than two confidence bounds from the best empirical value during the filter h_j. Thus, we get,

$$\max_{i\in\mathscr{K}_{h_j}} \overline{\mu}^{h_j}_{i,\texttt{eff}} = \overline{\mu}^{h_j}_{i_j,\texttt{eff}} \leq \overline{\mu}^{h_{j+1}}_{i_j,\texttt{eff}} \leq \max_{i\in\mathscr{K}_{h_{j+1}}} \overline{\mu}^{h_{j+1}}_{i,\texttt{eff}}.$$

The first inequality follows because $\overline{\mu}^{h_{j+1}}_{i_j,\texttt{eff}}$ contains reward sample which are either in $\overline{\mu}^{h_j}_{i_j,\texttt{eff}}$ or are older than the ones in $\overline{\mu}^{h_j}_{i_j,\texttt{eff}}$. Indeed, when $m=2$, $\widehat{\mu}^{h_{j+1}}_{i,\texttt{eff}}$ is updated synchronously with $\widehat{\mu}^{h_j}_{i,\texttt{eff}}$ (see Figure 4.6 and its section on delay). Hence, at each update of $\widehat{\mu}^{h_{j+1}}_{i,\texttt{eff}}$, it contains all the samples of $\widehat{\mu}^{h_j}_{i,\texttt{eff}}$ and the 2^j precedent ones. Thus, because the reward is decreasing, we have $\overline{\mu}^{h_{j+1}}_{i_j,\texttt{eff}} \geq \overline{\mu}^{h_j}_{i_j,\texttt{eff}}$. The second inequality uses that $i_j \in \mathscr{K}_{h_{j+1}}$. Gathering Equations 4.43, 4.47 and 4.48, we get

$$\overline{\mu}^{h}_{i_t}(t,\pi_{\text{EF}}) \geq \mu_{i^\star_t}(t, N_{i^\star_t,t-1}) - 4\sum_{j=0}^{j_h-1} \frac{2^j}{2^{j_h}-1} c(2^j,\delta_t). \tag{4.49}$$

With few lines of algebra, we reduce the sum,

$$\begin{aligned}
\sum_{j=0}^{j_h-1} \frac{2^j}{2^{j_h}-1} c(2^j,\delta_t) &= \sum_{j=0}^{j_h-1} \frac{\sqrt{2}^j}{2^{j_h}-1} c(1,\delta_t) && c(2^j,\delta_t) = \frac{c(1,\delta_t)}{\sqrt{2^j}} \\
&= \frac{\sqrt{2}^{j_h}-1}{\left(\sqrt{2}-1\right)\left(2^{j_h}-1\right)} c(1,\delta_t) && \sum_{n=0}^{N} q^n = \frac{q^{N+1}-1}{q-1} \\
&= \frac{1}{\left(\sqrt{2}-1\right)\left(\sqrt{2}^{j_h}+1\right)} c(1,\delta_t) && a^2-1=(a-1)(a+1) \\
&\leq \frac{\sqrt{2}}{\left(\sqrt{2}-1\right)\sqrt{2^{j_h+1}}} c(1,\delta_t) && \sqrt{2^{j_h}}+1 \geq \frac{\sqrt{2^{j_h+1}}}{\sqrt{2}} \\
&\leq \frac{\sqrt{2}}{\left(\sqrt{2}-1\right)\sqrt{h}} c(1,\delta_t) && h \leq 2^{j_h+1} \\
&= \frac{\sqrt{2}}{\sqrt{2}-1} c(h,\delta_t). && \frac{c(1,\delta_t)}{\sqrt{h}} = c(h,\delta_t)
\end{aligned}$$

Plugging this last equation in Equations 4.46 and 4.49 leads to the final result,

$$\overline{\mu}_{i_t}^h(t,\pi) \geq \max_{i\in\mathscr{K}} \mu_i(t,N_{i,t-1}) - \frac{C_\pi}{\sqrt{2\alpha}} c(h,\delta_t) \quad \text{with} \begin{cases} C_{\pi_{\mathrm{ER}}} = \frac{4\sqrt{\alpha}}{\sqrt{2}-1} \\ C_{\pi_{\mathrm{EF}}} = \frac{8\sqrt{\alpha}}{\sqrt{2}-1} \end{cases} .$$

■

Using Proposition 4.5.9 and Lemma 4.5.10 instead of Prop. 4.2.1 and Corollary 4.3.1, we can obtain similar problem dependent and independent bounds than for `FEWA` and `RAW-UCB`. The proof directly follows from the precedent analysis.

Theorem 4.5.11 For any rotting bandit scenario with means $\{\mu_i\}_i \in \mathscr{L}_L^K$ and any time horizon T, $\pi \in \{\pi_{\mathrm{ER}}, \pi_{\mathrm{EF}}\}$ run with $\alpha \geq 4$ and $m = 2$ suffers an expected regret of

$$\mathbb{E}[R_T(\pi)] \leq C_\pi \sigma \sqrt{\log(T)} \left(\sqrt{KT} + K\right) + 6KL \text{ with } \begin{cases} C_{\pi_{\mathrm{ER}}} = \frac{4\sqrt{\alpha}}{\sqrt{2}-1} \\ C_{\pi_{\mathrm{EF}}} = \frac{8\sqrt{\alpha}}{\sqrt{2}-1} \end{cases} .$$

Theorem 4.5.12 For any rotting bandit scenario with means $\{\mu_i\}_i \in \mathscr{L}_L^K$ and any time horizon T, $\pi \in \{\pi_{\mathrm{ER}}, \pi_{\mathrm{EF}}\}$ run with $\alpha \geq 4$ and $m = 2$ suffers an expected regret of

$$\mathbb{E}[R_T(\pi)] \leq \sum_{i\in\mathscr{K}} \left(\frac{C_\pi^2 \sigma^2 \log(T)}{\Delta_{i,h_{i,T}^+ - 1}} + C_\pi \sigma \sqrt{\log(T)} + 6L \right)$$

$$\text{with} \begin{cases} C_{\pi_{\mathrm{ER}}} = \frac{4\sqrt{\alpha}}{\sqrt{2}-1} \\ C_{\pi_{\mathrm{EF}}} = \frac{8\sqrt{\alpha}}{\sqrt{2}-1} \\ \Delta_{i,h} \text{ and } h_{i,T}^+ \text{ defined in Equation 4.27.} \end{cases}$$

Among the differences, we notice that our theory holds for a larger range of $\alpha \geq 4$ but the constant C_π is $\frac{\sqrt{2}}{\sqrt{2}-1} \sim 3.4$ times larger than their original counter part. We will show empirically in the next Subsection that it is mostly a theoretical artifact due to the more complex analysis. For instance, to derive Lemma 4.5.10, we consider for simplicity that the statistics could be delayed up to 100% their window size while we show in Proposition 4.5.4 that the normalized period is at most 50%.

R **Can we adapt our theory for $m \neq 2$?** The case $m = 2$ is less technical. First, 2 is an integer, which avoids the messier analysis due to the ceil operator in $h_{j+1} = \lceil m \cdot h_j \rceil$. Moreover, in the proof of `EFF-FEWA` (Equation 4.48), we use the strong synchronicity in the update which is the case when m is an integer. Last the decomposition of $\widehat{\mu}_i^h$ (Equations 4.41 to 4.43) is simpler on the geometric grid of parameter 2. Yet, we believe that the proof could be adapted without major difficulties at least for `EFF-RAW-UCB` when $m < 2$ (which is the most interesting case).

4.5.6 Experimental Results

Simulated efficient benchmark

Setup. We study a two-arm rotting bandit similar to the one presented in Subsection 4.1.4 but with a longer horizon $T = 10^6$. Like in the previous setups, the noise is Gaussian. There are one constant arm with value 0 and one rotting arm which switches from 0.1 to -0.1 after $T/4$ pulls.

Algorithms. In Figure 4.9, we compare the performance of `RAW-UCB` with `EFF-RAW-UCB` for different values of m. We use the value $\alpha = 1.4$ which is the best empirical value we found in the previous setups. We also display the best of the 3 versions of `wSWA` that we already studied. We add the running time in Table 4.2.

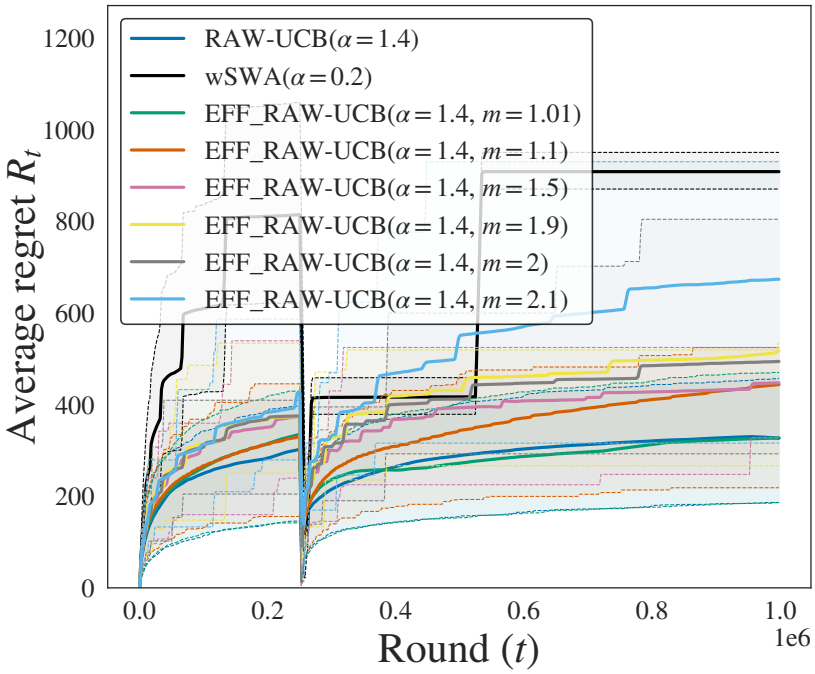


Figure 4.9: Regret across time. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region.

Results. Overall, the regret performance of `EFF-RAW-UCB` is up to 50% worse than the performance of `RAW-UCB`. The worst versions correspond to larger values of m. Yet, there are few counter-examples: $m = 2$ performs similarly to $m = 1.9$ and $m = 1.1$ performs similarly than $m = 1.5$ at the end of the game. We remark that there are discontinuities in the regret of the efficient algorithms. It is because the statistics are not updated at every round. Hence, when one statistic is updated, it can change the behavior of the algorithm for many rounds.

Policy	Running time (s)	comparison w/ RAW-UCB
RAW-UCB ($\alpha = 1.4$)	38837	100%
EFF-RAW-UCB ($\alpha = 1.4$, $m = 1.01$)	169	0.4 %
EFF-RAW-UCB ($\alpha = 1.4$, $m = 1.1$)	121	0.3 %
EFF-RAW-UCB ($\alpha = 1.4$, $m = 1.5$)	115	0.3 %
EFF-RAW-UCB ($\alpha = 1.4$, $m = 1.9$)	112	0.3 %
EFF-RAW-UCB ($\alpha = 1.4$, $m = 2$)	119	0.3 %
EFF-RAW-UCB ($\alpha = 1.4$, $m = 2.1$)	114	0.3 %
wSWA ($\alpha = 0.002$)	41	0.1 %
wSWA ($\alpha = 0.02$)	43	0.1 %
wSWA ($\alpha = 0.2$)	49	0.1 %

Table 4.2: Average running time and comparison with RAW-UCB for the efficient benchmark.

In terms of running time, the efficient trick drastically reduces the running time of EFF-RAW-UCB. While the theory suggests that there is no free lunch, we remark that setting a value very close to 1 does reduce the running time and recover very similar regret performance. Surprisingly, the running time are quite similar for $m \in [1.1, 2.1]$. The running time when $m = 1.01$ is only 48 seconds (+ 40%) larger than for $m = 1.1$ while there are 10 times more confidence intervals to compute. Hence, we believe that the UCBs computation time for $m \in [1.1, 2.1]$ is quite small compared to other fixed costs in the implementation (the reward generation, the $\log(t)$ computation, etc.). However, wSWA is still faster than EFF-RAW-UCB. It is surprising because its complexity is $\mathcal{O}\left(T^{2/3}\right)$, which is much larger than EFF-RAW-UCB's $\mathcal{O}(K \log_m T)$. Yet, in practice, wSWA computes $T^{2/3}$ sums while EFF-RAW-UCB computes $\mathcal{O}(K \log_m T)$ ucb indexes (with a $\sqrt{\cdot}$ and a log function).

We believe that we could speed up EFF-RAW-UCB with low-level implementation tricks. For instance, the profiling of the code indicates that the log function is very expensive. One could compute faster the $\log(t+1)$ from the previous value $\log(t)$. Yet, these low-level implementation tricks are not in the scope of this thesis.

Simulated benchmark #1 (2 arms) and #2 (10 arms).

Setup and Algorithms. We study the two benchmarks described in Subsection 4.1.4 and Section 4.4. In Figures 4.10 and 4.11, we compare RAW-UCB with EFF-RAW-UCB for two values of m $\{1.1, 2\}$.

Results. For the two values, we remark that EFF-RAW-UCB have a slightly worse performance than RAW-UCB (up to 50% for $m = 2$). It confirms our theoretical analysis which suggests that the performance of EFF-RAW-UCB is only at a constant factor of the performance of RAW-UCB. It also confirms that the smaller the m, the less regret we suffer.

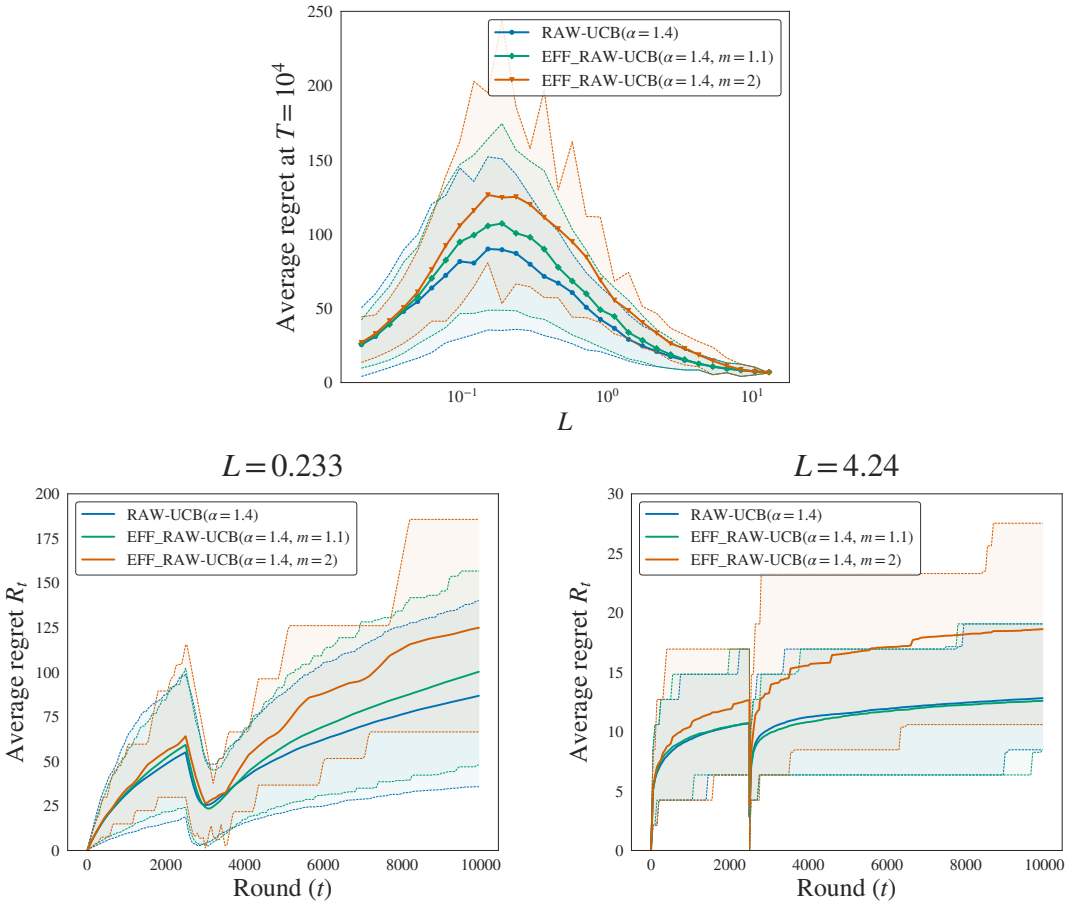


Figure 4.10: **Top:** Regret at the end of the game for different values of L. **Bottom:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region.

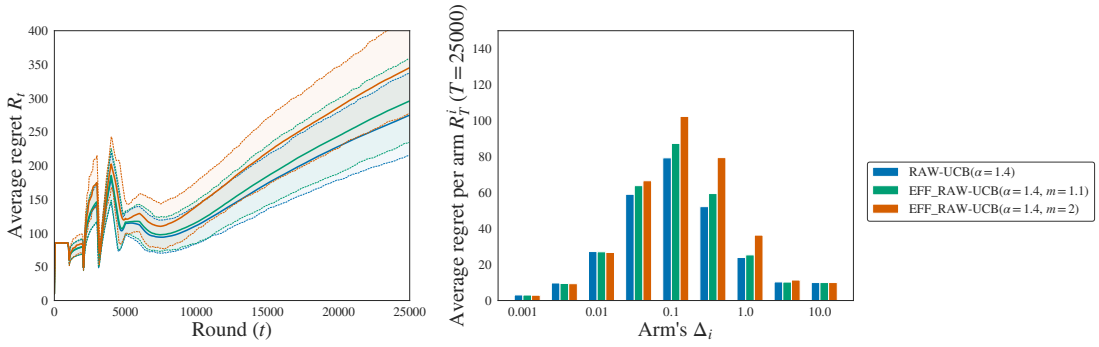


Figure 4.11: **Left:** Regret at the end of the game for different values of L. **Middle, Right:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region.

Notice that the two algorithms run in 3 seconds in average[3] versus 25 seconds for `RAW-UCB`

[3] 3.0 s for $m = 2$, 3.3 s for $m = 1.1$

and 1 second for wSWA. It shows that EFF-RAW-UCB effectively reduces the computation cost of RAW-UCB, even for a shorter horizon.

4.5.7 Conclusion

In this section, we provide a new update scheme which keeps $\mathcal{O}(\log_m t)$ averages with geometrical windows sequence of parameter m. These averages are updated with a delay which is proportional to the window size times $^{m-1}/_{m}$. We show that when we plug this efficient update scheme in our algorithms, we recover the same upper bounds as the original algorithms with a larger multiplicative constant. However, the computational complexity is considerably reduced from $\mathcal{O}(T)$ to $\mathcal{O}(K\log_m T)$. We also show that in practice we can recover almost the same performance as the classical algorithms but with a computational cost that is comparable with wSWA.

4.6 How harder are rotting bandits ?

In the last sections, we presented RAW-UCB, an algorithm which extends the results of UCB1 (Auer et al. 2002) on stationary bandits to the more general rotting bandits setup. Hence, we conclude that rotting bandits are not much harder than stationary ones.

Yet, UCB1 is only near asymptotic and minimax optimal. In Section 2.2, we explain that a better tuning of the confidence levels allows UCB variant to match the asymptotic and minimax rates for gaussian bandits.

This section investigates the impact of confidence levels tuning on RAW-UCB. How does it compare with UCB on stationary bandits? Does it improve the performance of RAW-UCB on our rotting benchmarks?

4.6.1 RAW-UCB++

We introduce RAW-UCB++, an algorithm which uses the RAW-UCB procedure (Alg.7) with a new index,

$$\text{ind}(i,t,\delta_{t,h}) \triangleq \min_{h\leq N_{i,t-1}} \left(\widehat{\mu}_i^h(N_{i,t-1}) + \sqrt{\frac{2\sigma^2 \log_+ (^2/\delta_{t,h})}{h}} \right)$$
$$\text{with} \quad \delta_{t,h} \triangleq \frac{2\left(^{Kh}/_{t}\right)^{\alpha}}{\left(1+\log_+ (^{t}/_{Kh})\right)^{\beta}}, \quad (4.50)$$

with $\log_+(\cdot) \triangleq \max(\log(\cdot), 0)$. The main difference with the index of RAW-UCB in Equation 4.25 is the more complex confidence level. First, we multiply our confidence level by Kh and replace log by $\log_+$. This is similar to the $\delta = ^{KN_{i,t}}/_{t}$ of MOSS-anytime (Degenne

and Perchet 2016). We replace $N_{i,t}$ - the number of pulls of arm i at the round t- by h, the number of sample in the associated average. Indeed, let us consider a two-arm bandit problem where the first arm has a much larger value $\mu_2 + 100\sigma$ than the second one (with value μ_2) at the beginning of the game. Hence, at the beginning of the game, $N_{1,t} \sim t$ because RAW-UCB++ can quickly identify arm 1 as the current best arm. After $\frac{T}{2}$ pulls, arm 1 abruptly decay to a value $\mu_2 + \sigma$. If we do not replace $N_{i,t}$ by h in the confidence levels in Equation 4.50, the exploration bonus would be canceled until the end of the game for all the UCB of arm 1 because $\frac{KN_{i,t}}{t} > 1$. Without the exploration bonus, there is a large enough probability that the index of arm 1 takes a value below μ_2. Indeed, since we take the minimum across indexes, if the first reward sample after the decay is below μ_2, then the meta-index will be below μ_2. In this case, RAW-UCB may pull arm 2 until the end of the game and suffer at least $\mathcal{O}(\sigma T)$ regret. Replacing $N_{i,t}$ by h restore the exploration bonus for arms which have recently decay.

Second, we add a logarithmic exploration inflation factor. Notice that we also divide t by Kh in the inner logarithm, as it is done for KL-UCB++ (Ménard and Garivier 2017). When the noise is not gaussian, the concentration results are slightly less tight and the asymptotic optimality proof often needs this factor. For instance, Cappé et al. (2013) use a factor $\log(t)^{-3}$ in their theory, but they recommend to not use it in practice. However, for RAW-UCB, we believe that extra-exploration is needed in practice. Indeed, we find our best experimental performance for $\alpha = 1.4$ which is larger than the asymptotic optimal tuning for UCB $\alpha = 1$ (Lattimore and Szepesvári 2020). In our theory in Section 4.3, we increase α by one compared to UCB1 to ensure that the t (instead of K) constructed statistics were into the confidence levels.

4.6.2 Experiments

Stationary Experiment

Setup. We consider a stationary bandits with two arms with $\mu_1 = 0$ and $\mu_2 = \Delta$. We consider two different values of $\Delta \in \{0.01, 1\}$. The rewards are then generated by applying a Gaussian i.i.d. noise $\mathcal{N}(0, \sigma = 1)$. We run the experiment with the horizon $T = 10^6$.

Algorithms. We consider UCB and MOSS-anytime (Degenne and Perchet 2016). We tune UCB with asymptotic optimal confidence level $\sqrt{2\log(t)/N_{i,t}}$ (Lattimore and Szepesvári 2020). For MOSS-anytime, we use $\sqrt{2\log\left(t/KN_{i,t}\right)/N_{i,t}}$, which corresponds to a tuning of its parameter $\alpha = 3$. We test RAW-UCB++ with many different values, but we display two different sets of values $\alpha = 1$ and $\beta = 3.5$ or $\alpha = 2$ and $\beta = 0$. These two sets of values give the most consistent performance on the two problems. We add RAW-UCB with $\alpha = 1.4$ for comparison. For RAW-UCB and RAW-UCB++, we use the efficient version with $m = 1.01$ which performs similarly than the classical algorithm Subsection 4.5.6.

Results. RAW-UCB++ seems to improve slightly the results compare to RAW-UCB. Yet, the improvement is not as significant than between MOSS-anytime and UCB. On the $\Delta = 1$

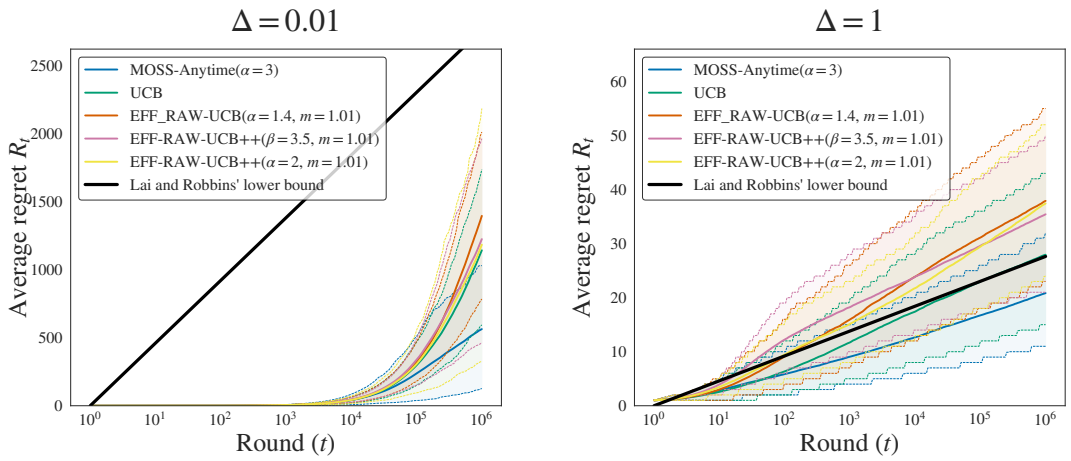


Figure 4.12: Stationary experiments

experiment, we see that the tuning with the logarithm ($\beta = 3.5$) seems to enjoy better asymptotic guarantee than the tuning with $\alpha = 2$. Yet, it is not clear if `RAW-UCB++` ($\beta = 3.5$) is asymptotic optimal with respect to the Lai and Robbin's lower bound. However, at finite horizon, the different parameters are quite close to each other.

Rotting Experiments #1 (2 arms) and #2 (10 arms).

Setup and Algorithms. We study the two benchmarks described in Subsection 4.1.4 and Section 4.4. In Figures 4.10 and 4.11, we compare `EFF-RAW-UCB++` ($\alpha = 2$, $m = 1.01$) with `EFF-RAW-UCB` ($\alpha = 1.4$, $m = 1.01$). The parameters α were selected according to previous experiments.

Results. `EFF-RAW-UCB++` performs slightly better than `EFF-RAW-UCB` for almost any experiments and at almost any rounds. A noticeable exception is when L is large in the two-arms experiment: the result of `EFF-RAW-UCB++` is slightly worse than for `EFF-RAW-UCB`. Overall, the results suggest that the aggressive confidence tuning technique of stationary bandits also improves the rotting adaptivity. Yet, we notice that the confidence levels with $\alpha = 2$ are less tight than the tuning of `MOSS-anytime` (which would correspond to $\alpha = 1$).

4.6.3 Towards a theoretical analysis of `RAW-UCB++`

Analyzing `UCB` with tight confidence levels in stationary bandits is already a challenging task (Degenne and Perchet 2016; Ménard and Garivier 2017; Lattimore 2018). The analysis of `RAW-UCB++` faces two additional difficulties: on the one hand, `RAW-UCB`'s meta-index is more complex than `UCB`'s; on the other hand, rotting bandits are more difficult to analyze than stationary ones.

First, we can ignore the second part of the problem and try to analyze `RAW-UCB++` on stationary problems. Tight analysis of `UCB` usually bound the number of pulls of the

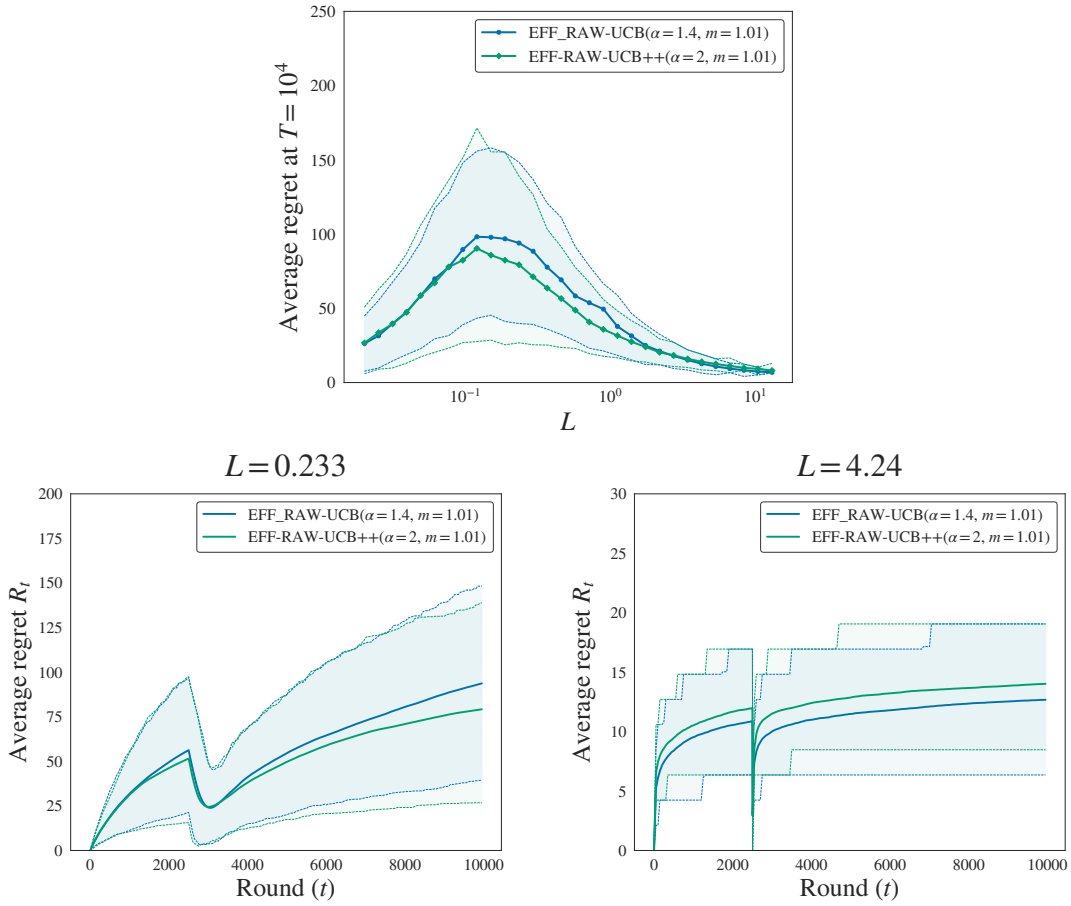


Figure 4.13: **Top:** Regret at the end of the game for different values of L. **Bottom:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region.

suboptimal arms. A classical trick is to set a threshold $\mu_\star - \varepsilon_i$ and notice that a necessary condition to pull a suboptimal arm i is that either the index of the optimal arm is below the threshold, or the index of the suboptimal arm is above the threshold,

$$N_{i,T} \leq \sum_{t=1}^{T} \mathbb{1}\left[\texttt{ind}\left(i_\star, t, \delta_{t,h}\right) < \mu_\star - \varepsilon_i\right] + \mathbb{1}\left[\texttt{ind}\left(i, t, \delta_{t,h}\right) > \mu_\star - \varepsilon_i\right].$$

The upper deviation of suboptimal arms' indexes is not more difficult to control for `RAW-UCB` than for `UCB`. Indeed, since we take the minimum across confidence bounds, the indexes of `RAW-UCB` are smaller than the indexes of `UCB` (when the confidence levels are the same).

Controlling the lower deviation of the optimal arm's index is more challenging. Indeed, at each round t, we have to control the probability that any ucb associated with any h last pulls after any $N_{i,t}$ pulls is below the threshold. Compared to `UCB` where there is

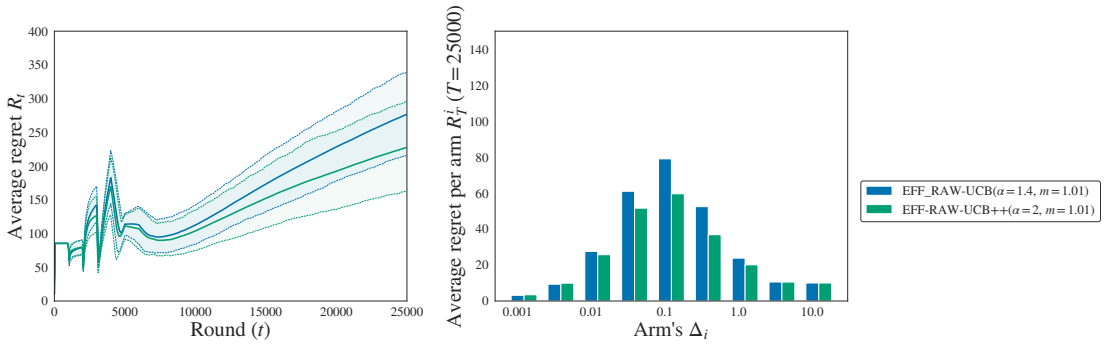


Figure 4.14: **Left:** Regret at the end of the game for different values of L. **Middle, Right:** Regret across time for two values of L. Average over 1000 runs. We highlight the $[10\%, 90\%]$ confidence region.

only a scan on the possible values of $N_{i,t}$, we have to handle a double scan on $N_{i,t}$ and h. In Section 4.3, we handle the multiple windows with a crude union bound which leads to a fairly large decrease of the confidence levels. A tighter analysis would probably require better statistical engineering than a union bound or a simple peeling argument. For instance, Maillard (2019) develop new concentration results for similar scan statistics for sequential change-point detectors (with some applications to bandits). The difficulty is that the quantity to be bounded is not a sub/super-martingale. Yet, it is quite uncertain that a tighter analysis is actually possible in our case. Indeed, the empirical tuning of `RAW-UCB` (resp. `RAW-UCB++`) increases α by 0.4 (resp. 1) compared to the confidence levels of `UCB` (resp. `MOSS-anytime`). This is comparable with the theoretical increase of 1 due to the union bound over all the possible windows.

4.7 Linear rotting bandits are impossible to learn

4.7.1 Linear bandits

In Section 2.5, we presented the contextual bandit, a line of work in which the learner is given a context on which depends the action's reward. The linear bandit is a special case where the reward is a noisy linear form of a context-action vector.

Unlike the classical multi-armed bandits, the feedback associated with one action can be used to learn the other actions' efficiency. In fact, the number of actions can be very large or even potentially infinite, as soon as the representation has a finite dimension.

This shared knowledge is interesting in the context of education. Indeed, a popular model to predict a student's answer is the Item Response Theory (Hambleton and Swaminathan 2013). It models the answer with a logistic function filled with a student and question-related parameters. In its most simple form, it can be seen as logistic regression, a generalized linear model. Generalized linear bandits were studied by (Filippi et al. 2010) as an extension of the linear bandit.

In the following, we will introduce formally the linear bandit. We will then discuss the possibility of extending our multi-armed rested rotting bandits result in the linear case.

Model

At each round, the learner chooses an action A_t in a fixed set of embedded actions $\mathscr{A} \subset \mathbb{R}^d$ and receives,

$$o_t \triangleq \langle \mu | A_t \rangle + \varepsilon_t \text{ with } \mathbb{E}[\varepsilon_t | \mathscr{H}_t] = 0 \text{ and } \forall \lambda \in \mathbb{R}, \mathbb{E}\left[e^{\lambda \varepsilon_t}\right] \leq e^{\frac{\sigma \lambda^2}{2}},$$

with $\mu \in \mathbb{R}^d$ a reward vector unknown to the learner. In this model, the learner might face a huge number of actions (or even infinite), but they only has d independent parameters to estimate to take the right decision. The goal of the learner is still to maximize the cumulative reward,

$$J_T(\pi) = \sum_{t=1}^{T} \langle \mu | \pi(t) \rangle.$$

The optimal oracle strategy selects $\pi^\star(t) = A^\star \triangleq \arg\max_{A_t \in \mathscr{A}} \langle \mu | A_t \rangle$. Thus, we can define the regret,

$$R_T(\pi) = J_T(\pi^\star) - J_T(\pi) = \langle \mu | \sum_{t=1}^{T} A^\star - \pi(t) \rangle.$$

4.7.2 Linear rested rotting bandits

We want to extend the rested rotting bandit to actions with a linear embedding. We introduce d non-increasing and L-Lipschitz functions $\mu_i : \mathbb{R}^+ \to \mathbb{R}$. While there were K reward functions defined on $\mathbb{N}$ in the rotting MAB model, we now have d functions defined on $\mathbb{R}^+$. Indeed, we now have d (and not K) reward parameters. Moreover, for the rested rotting MAB, the reward is evolving according to the number (in $\mathbb{N}$) of pulls of arm i. In the linear model, we cannot simply count the number of pulls along each direction because A_t has possibly components along all the directions. We will need to find a quantifier of the pulling intensity along direction i. This is not surprising that this quantifier will have value in $\mathbb{R}^+$ because we select direction i with intensity $A_{t,i} \in \mathbb{R}$. We suggest using,

$$N_{t,i} \triangleq \sum_{t'=1}^{t} A_{t',i}.$$

Hence, we define the feedback as,

$$o_t = \sum_{i \leq d} \int_{N_{t-1,i}}^{N_{t,i}} \mu_i(x) dx + \varepsilon_t = \int_{N_{t-1}}^{N_t} \langle \mu(n) | dn \rangle + \varepsilon_t,$$

with $\{\varepsilon_t\}$ a sequence of independent σ-subgaussian random variables. We also define the cumulative reward,

$$J_T(\pi) = \int_0^{N_T} \langle \mu(n) | dn \rangle.$$

Like in the rotting MAB model, the total reward depends only on the cumulative pulling intensity N_T. This property is very useful, as it allows to compare the performance of two policies only with their pulling differences (the overpulled/underpulled arms) and not with the specific order of the pulls, *i.e.*,

$$J_T(\pi_2) - J_T(\pi_1) = \int_{N_T^{\pi_1}}^{N_T^{\pi_2}} \langle \mu(n) | dn \rangle.$$

Last, we restrain the action vector to be in the positive quadrant, *i.e.* $\mathscr{A} \subset \mathbb{R}^{+d}$. Indeed, we want the pulling intensity to be non-decreasing with time t such that the reward associated to a vector $A \in \mathbb{R}^{+d}$ is non-increasing,

$$t_1 < t_2 \implies N_{t_1-1} \leq N_{t_2-1} \implies \int_{N_{t_1-1}}^{N_{t_1-1}+A} \langle \mu(n) | dn \rangle \geq \int_{N_{t_2-1}}^{N_{t_2-1}+A} \langle \mu(n) | dn \rangle.$$

This model correctly extends linear bandits and rested rotting bandits. On the first hand, when reward functions are constant, we recover the linear bandit model of Subsection 4.7.1. Indeed, we have,

$$o_t = \int_{N_{t-1}}^{N_t} \langle \mu | dn \rangle + \varepsilon_t = \langle \mu | N_t - N_{t-1} \rangle + \varepsilon_t = \langle \mu | A_t \rangle + \varepsilon_t,$$
$$J_T(\pi) = \int_0^{N_T} \langle \mu | dn \rangle = \langle \mu | N_T \rangle = \sum_{t=1}^{T} \langle \mu | A_t \rangle.$$

On the other hand, when the actions sets $\mathscr{A}$ are the d canonical basis vectors at every round, we recover the rested rotting bandits setting of Section 4.1. Indeed, if we call $\mu_i^{MAB}(n) \triangleq \int_n^{n+1} \mu_i(n) dn$, we have,

$$N_{t,i} = \sum_{t=1}^{T} \mathbb{1}(\pi(t) = i),$$
$$o_t = \int_{N_{t-1}}^{N_t} \langle \mu(n) | dn \rangle + \varepsilon_t = \mu_{i_t}^{MAB}(N_{t-1,i_t}) + \varepsilon_t,$$
$$J_T(\pi) = \int_0^{N_T} \langle \mu(n) | dn \rangle = \sum_{i \leq d} \sum_{n=0}^{N_{T,i}-1} \mu_i^{MAB}(n).$$

To conclude the argument, we notice that the constructed functions μ_i^{MAB} are in $\mathscr{L}_L$ (see Definition 4.1.1). Indeed,

$$\mu_i^{MAB}(n) - \mu_i^{MAB}(n+1) = \int_n^{n+1} (\mu_i(x) - \mu_i(x+1))\, dx \geq 0,$$
$$\mu_i^{MAB}(n) - \mu_i^{MAB}(n+1) = \int_n^{n+1} (\mu_i(x) - \mu_i(x+1))\, dx \leq L.$$

The first line follows by the non-increasing property of μ_i. The second line is justified because μ_i is L-lipschitz.

R We choose to measure the pulling intensity as $N_{t,i} \triangleq \sum_{t'=1}^{t} A_{t',i}$. In the (stationary) linear bandit problem, the number of pulls is replaced by the matrix $\sum_{t'=1}^{t} A_{t'} A_{t'}^{\intercal}$ which is used in the least-square regression and in the confidence ellipsoid computation. Hence, from an information-theoretic point of view, the natural identification is $N_{t,i} \triangleq \sum_{t'=1}^{t} A_{t',i}^2$ (which follows when the actions embeddings are the canonical basis vectors). Yet, our choice is motivated because we want a linear dependence between $N_{t,i}$ and the collected reward along direction i in the stationary case.

4.7.3 The offline problem

In the rested rotting bandits model, Heidari et al. (2016) show that the greedy oracle policy which selects the arm with the highest upcoming reward is anytime optimal. Unfortunately, this result does not hold in the linear rotting bandit model.

Proposition 4.7.1 Let's consider the simple $d = 2$ case. For any horizon $T \geq 2$, there exists an L-lipschitz reward vector function μ bounded in $[0, L]$, a fixed vector arms set $\mathscr{A}$, and a strategy π such that the greedy oracle π_O suffers,

$$J_T(\pi) - J_T(\pi_O) \geq \frac{L(T-2)}{8}.$$

We notice that the worst regret for reward values in $[0, L]$ and arms in $[0,1]^d$ is LT. Hence, Proposition 4.7.1 shows that π_O is unable to learn as its problem-independent performance is at a constant ratio of the worst possible rate. We will prove precisely this statement in Subsection 4.7.5 but we give here the main intuition. Let's consider a vector reward function such that the first direction decays from L to 0 in the middle of the game and the second one has a constant reward value $L/2$. In the MAB case - when arms are orthogonal - the greedy oracle π_O stops pulling the first direction when the associated reward decreases. However, when arms are not orthogonal (e.g. $(1,0)$ and $(1/2, 1/2)$), the greedy policy will collect quickly all the reward associated with the first direction by selecting the first arm. Then, it selects arm 2 which still pulls a fraction of the first direction, even though the reward has decayed. A better oracle strategy would be to notice that the good reward associated with the first direction will be gathered at the end of the game anyway, and focus on maximizing the reward associated with the other direction.

Notice that this better strategy needs to see at least up to the middle of the game to see that the reward will decay. Otherwise, it will not behave consistently on the problems on which the reward does not decay. Refining this argument, we can show in Proposition 4.7.2 that any short-sighted oracle - *i.e.* a policy that can only see the next T_O rewards for any combination of T_O arms - suffers a regret which scales at least with $T - T_O$. It means that the offline problem is a planning problem where we need full knowledge of the reward function and the horizon T.

Proposition 4.7.2 Let π a short-sighted policy which sees the future rewards associated to any combination of T_O arms. For $T \geq T_O + 23$, there exists a problem μ and a policy π' such that,

$$J_T(\pi') - J_T(\pi) \geq \frac{L(T - T_O)}{20}.$$

4.7.4 The noise-free online problem

A direct consequence of Proposition 4.7.2 is that any learning policy - a special case of short-sighted policies with $T_O = 0$ - suffers a worst-case linear regret, even in the absence of noise. Hence, we conclude that the rotting linear setup is not learnable.

Corollary 4.7.3 For any learning policy π and $T \geq 23$, there exists a problem μ and a policy π'

$$J_T(\pi') - J_T(\pi) \geq \frac{LT}{20}.$$

Again, we highlight the deep contrast with the MAB setting, where a simple greedy policy is guaranteed to make at most $K - 1$ mistakes. The key argument is that when a direction decreases we need to be able to stop pulling that direction and focus on other directions.

4.7.5 Proofs

Proof of Proposition 4.7.1. For $d = 2$, $\mathscr{A} = \{A_1, A_2\}$ with $A_1 = (1,0)^\intercal$ and $A_2 = (1/2, 1/2)^\intercal$. For any horizon $T \geq 2$, we consider the following L-lipschitz reward functions,

$$\mu_1(x) = \begin{cases} L & \text{if } x < \frac{T-1}{2} \\ L(1 - x + \frac{T-1}{2}) & \text{if } \frac{T-1}{2} \leq x \leq \frac{T+1}{2} \\ 0 & \text{else} \end{cases} \qquad \text{and} \qquad \mu_2(x) = \frac{L}{2}.$$

The greedy oracle strategy first selects A_1 for $\lfloor \frac{T}{2} \rfloor$ rounds and then A_2 for the remaining

$\lceil \frac{T}{2} \rceil$. Hence,

$$N_{T,1} = \left\lfloor \frac{T}{2} \right\rfloor + \frac{1}{2}\left\lceil \frac{T}{2} \right\rceil,$$
$$N_{T,2} = \frac{1}{2}\left\lceil \frac{T}{2} \right\rceil \geq \frac{T+1}{2}.$$

For $T \geq 2$, $N_{T,1} \geq \frac{T+1}{2}$, which means that the greedy policy collects all the $LT/2$ reward along the first direction. We also notice that $N_{T,2} \leq \frac{T+1}{4}$, which means that the total reward is bounded by,

$$J_T(\pi_O) = \int_0^{N_{T,1}} \mu_1(x)dx + \int_0^{N_{T,2}} \mu_2(x)dx \leq \frac{LT}{2} + \frac{L(T+1)}{8}.$$

We now consider the policy π_2 which always selects arm 2. At the end of the game, it gathers the reward,

$$J_T(\pi_2) = \int_0^{\frac{T}{2}} \mu_1(x)dx + \int_0^{\frac{T}{2}} \mu_2(x)dx = \frac{L(T-1/4)}{2} + \frac{LT}{4}.$$

Hence, we have that,

$$J_T(\pi_2) - J_T(\pi_O) \geq \frac{L(T-2)}{8}.$$

■

Proof of Proposition 4.7.2. We still consider $d = 2$, $\mathscr{A} = \{A_1, A_2\}$ with $A_1 = (1,0)^\intercal$ and $A_2 = (1/2, 1/2)^\intercal$. For any horizon T, we consider the (L-lipschitz) two reward vector functions μ^1 and μ^2,

$$\mu_1^1(x) = L \qquad \text{and } \mu_2^1(x) = \frac{L}{2},$$
$$\mu_1^2(x) = \begin{cases} L & \text{if } x < \frac{T+T_O-1}{2} \\ L(1-x+\frac{T+T_O-1}{2}) & \text{if } \frac{T+T_O-1}{2} \leq x \leq \frac{T+T_O+1}{2} \\ 0 & \text{else} \end{cases} \qquad \text{and } \mu_2^2(x) = \frac{L}{2}.$$

Notice that it is not possible to tell the difference between the two setups for the short-sighted oracle π before round t such that $N_{t,1} \geq \frac{T-T_O-1}{2}$. Notice that this round always exists because $N_{T,1} \geq \frac{T}{2}$ for any policy because the pulling intensity of the first direction is always greater than $1/2$.

On problem μ^1, we will compare π to π_1 which selects always arm 1. On problem μ^2, we will compare π to π_2 which selects always arm 2. We give the cumulative reward of the

different policies on the two problems,

$$J_T^1(\pi_1) = LT, \tag{4.51}$$
$$J_T^2(\pi_2) = \frac{L(T - 1/4)}{2} + \frac{LT}{4}, \tag{4.52}$$
$$J_T^1(\pi) = L\left(T - N_{T,2}^1\right), \tag{4.53}$$
$$J_T^2(\pi) \leq \frac{L\left(T + N_{T,2}^2\right)}{2}, \tag{4.54}$$

with $N_{T,2}^k$ the pulling intensity in direction 2 of π on problem k. The inequality upper-bounds the reward collected in direction 1 by its maximum $LT/2$. Since we cannot distinguish between the two problems until $N_{t,1} \geq \frac{T-T_O-1}{2}$, we will show that $N_{T,2}^1$ and $N_{T,2}^2$ are loosely related.

We call t_u the last round such that the two problems are indistinguishable. Therefore, we have that,

$$N_{t_u,1}^1 = N_{t_u,1}^2 \triangleq N_{t_u,1} \text{ and } N_{t_u,2}^1 = N_{t_u,2}^2 \triangleq N_{t_u,2}.$$

Problems are indistinguishable until,

$$N_{t_u,1} < \frac{T - T_O - 1}{2}.$$

We also know that the problems can be distinguished at the round $t_u + 1$, *i.e.*

$$\frac{T - T_O - 1}{2} \leq N_{t_u+1,1} \leq N_{t_u,1} + 1.$$

The second inequality is because $N_{t_u+1,1} - N_{t_u,1} \leq \max_{i \leq 2} A_{i,1} = 1$. We have bounded $N_{t_u,1}$ but we use $N_{T,2}^k$ in Equations 4.53 and 4.54. We will start by relating $N_{t_u,2}$ to $N_{t_u,1}$, and then we will relate $N_{T,2}^k$ to $N_{t_u,2}$. We call $n_{t,i}$, the number of pulls of arm i at the round t. We have that

$$N_{t,1} = n_{t,1} + \frac{n_{t,2}}{2},$$
$$N_{t,2} = \frac{n_{t,2}}{2},$$
$$n_{t,1} + n_{t,2} = t.$$

We can rewrite $N_{t,1} = t - N_{t,2}$. Hence, the above inequations on $N_{t_u,1}$ can be translated for $N_{t_u,2}$.

$$N_{t_u,2} > \frac{2t_u - T + T_O + 1}{2},$$
$$N_{t_u,2} \leq \frac{2t_u - T + T_O + 3}{2}. \tag{4.55}$$

We can now bound $N^k_{t_u,2}$ for all k,

$$\frac{2t_u - T + T_O + 1}{2} < N_{t_u,2} \leq N^k_{T,2} \leq \frac{T - t_u}{2} + N_{t_u,2} \leq \frac{t_u + T_O + 3}{2}.$$

The first and fourth inequality use Equation 4.55. The second uses that $t_u \leq T$ and that the pulling intensity can only grow through the rounds. The third inequality uses that with $N_{t_u,2}$ pulling intensity at the round t_u, the learner cannot reach more than $N_{t_u,2} + (T - t_u)\max_j A_{j,2} = N_{t_u,2} + \frac{T-t_u}{2}$ (by selecting arm 2 until the end of the game).

We can use Equations 4.51 and 4.53 (respectively 4.52 and 4.54) to lower-bound the difference of performance with respect to policy π_1 (respectively π_2) on problem 1 (respectively 2).

$$\begin{aligned}
J^1_T(\pi_1) - J^1_T(\pi) &= LN^1_{T,2} \geq \frac{L}{2}\left(2t_u - (T - T_O) + 1\right) \\
J^2_T(\pi_2) - J^2_T(\pi) &\geq \frac{L}{4}\left(T - 2N^2_{T,2} - 1/2\right) \geq \frac{L}{4}\left(T - t_u - T_O - 3.5\right).
\end{aligned}$$

The only quantity which depends on the algorithm π is $t_u \in \{1, \dots, T\}$. In a minimax perspective, we lower bound the maximum of these two bounds,

$$\max\left(J^1_T(\pi_1) - J^1_T(\pi), J^2_T(\pi_2) - J^2_T(\pi)\right) \geq \frac{L(T - T_O)}{10} - 1.15L.$$

We conclude the proof by noticing that when $T - T_O \geq 23$,

$$\frac{L(T - T_O)}{10} - 1.15L \geq \frac{L(T - T_O)}{20}.$$

■

5. The rotting assumption makes restless bandits easier

A non-rotting restless bandit. Who would say he is not tough?

5.1 Restless rotting bandits

5.1.1 Restless bandits model

Feedback loop

At each round t, an agent chooses an arm $i_t \in \mathcal{K} \triangleq \{1,...,K\}$ and receives a noisy reward o_t. The reward associated to each arm i is a σ^2-sub-Gaussian random variable with expected value of $\mu_i(t)$, which depends on the number of rounds t. Let $\mathcal{H}_t \triangleq \{\{i_s, o_s\}, \forall s < t\}$ be the sequence of arms pulled and rewards observed until round t, then

$$o_t \triangleq \mu_{i_t}(t) + \varepsilon_t \text{ with } \mathbb{E}[\varepsilon_t | \mathcal{H}_t] = 0 \text{ and } \forall \lambda \in \mathbb{R}, \mathbb{E}\left[e^{\lambda \varepsilon_t}\right] \leq e^{\frac{\sigma\lambda^2}{2}}. \tag{5.1}$$

Objective

We will only consider deterministic agents which output an arm i at each round t. Like in the previous chapter, we distinguish offline (or oracle) policies $\pi \in \Pi_O$ - which are functions which map the round t and the set of reward functions to arms - from online (or learning) policies $\pi \in \Pi_L$ - which are functions from the history of observations $\mathcal{H}_t$ at a round t to arms. For both types of policies, we often use the shorter notation $\pi(t)$, where

Chapter header: Fourth cover of *le Petit Journal*, 12th of july 1914.

the dependency on μ or $\mathscr{H}_t$ is implicit. The performance of a policy π is measured by the (conditionally expected) rewards accumulated over time,

$$J_T(\pi) \triangleq \sum_{t=1}^{T} \mu_{\pi(t)}(t). \tag{5.2}$$

Proposition 5.1.1 The characterization of the optimal oracle policies is straightforward,

$$\pi^\star \in \arg\max_{\pi \in \Pi_O} J_T(\pi) \iff \forall t \leq T, \pi^\star(t) \in \arg\max_{i \in \mathscr{K}} \mu_i(t).$$

In the following, we call $i_t^\star \in \arg\max_{i\in\mathscr{K}} \mu_i(t)$ one of the best arm at the round t, and $\mu_\star(t) \triangleq \max_{i\in\mathscr{K}} \mu_i(t)$ the corresponding best value.

Notice that there may be several optimal policies if, at a given round t, there are several arms with maximal value. However, all these policies get the same cumulative reward at every round; thus, the tie-break rule can be chosen arbitrary without impacting the performance. We set a policy $\pi^\star \in \arg\max_{\pi\in\Pi_O} J_T(\pi)$. Calling $J_T^\star = J_T(\pi^\star)$ the largest cumulative reward achievable, one can measure the regret of any policy (learning or oracle) compared to the optimal one,

$$R_T(\pi) \triangleq J_T^\star - J_T(\pi). \tag{5.3}$$

R Like in the rested setup, the regret is measured against the optimal oracle policy rather than a fixed-arm policy as it is a case in adversarial bandits. Moreover, for constant $\mu_i(t)$-s, the problem, and definition of regret reduce to the ones of stationary stochastic bandits (where the regret is measured against the best fixed-arm policy which is also the optimal oracle policy).

5.1.2 Piece-wise stationary bandits

Garivier and Moulines (2011) study the restless bandits case, where rewards are piece-wise stationary.

Assumption 5.1.2 Let V be a positive constant and Υ_T a positive integer. $\mu_i : \mathbb{N}^\star \to [-V, 0]$[1] are piecewise stationary non-increasing functions of the time t with at most $\Upsilon_T - 1$ breakpoints. Formally,

$$\sum_{t=1}^{T-1} \mathbb{1}\left(\exists i \in \mathscr{K}, \mu_i(t) \neq \mu_i(t+1)\right) \leq \Upsilon_T - 1.$$

We call $\{t_k\}_{k\leq \Upsilon-1}$ the set of breakpoints with $t_0 = 0$, μ_i^k the value of $\mu_i(t)$ for $t \in \{t_k+1, \dots, t_{k+1}\}$. We call $i_k^\star \in \arg\max_{i\in\mathscr{K}} \mu_i^k$ (one of) the best arm in batch k, $\mu_\star^k \in \max_{i\in\mathscr{K}} \mu_i^k$ the corresponding best value, and $\Delta_{i,k} \triangleq \mu_\star^k - \mu_i^k$ the gap to the best arm for arm i during batch k.

[1] We could choose any interval $[x, x+V]$. Yet, with the upcoming decreasing assumption, choosing $[-V, 0]$ instead of $[0, V]$ emphasizes that the learner cannot infer parameter V from the first pulls. Notice that we will never use that our rewards are negative in our analysis.

Lower Bounds

Proposition 5.1.3 — Auer et al. (2003). For any strategy π, there exists a K-armed piece-wise stationary bandit scenario with means $\{\mu_i(t)\}_{i,t}$ satisfying Assumption 5.1.2 such that,

$$\mathbb{E}\left[R_T(\pi)\right] \geq \frac{\sigma}{32}\sqrt{\Upsilon_T K T}\,.$$

This bound is not surprising as it shows that piece-wise stationary bandits with Υ_T change-points are at least as hard as Υ_T stationary problems with horizon $\frac{T}{\Upsilon_T}$ (Auer et al. 2003). We will show a slightly stronger result in Subsection 5.1.4.

Garivier and Moulines (2011) shows a self-bonding property of the regret. They build a problem μ' on which the reward function equals the reward on a stationary problem μ except on a period τ (see Figure 5.1). During this time span, the best arm of μ keeps its value while the worst arm *increases* to become optimal. The size of τ is chosen inversely proportional to the average pulling rate of the bad arm in μ. Indeed, the lower the pulling rate of the bad arm, the longer the adversary can increase its value in μ' without being noticeable by the learner (which can be quantified thanks to Lemma 5.1, Auer et al. (2003)). Since the pulling rate of the bad arm in μ is proportional to $R_T(\mu)$, we get a lower bound proportional to $\tau \sim \frac{T}{R_T(\mu)}$. We reproduce the version of the theorem in Lattimore and Szepesvári (2020).

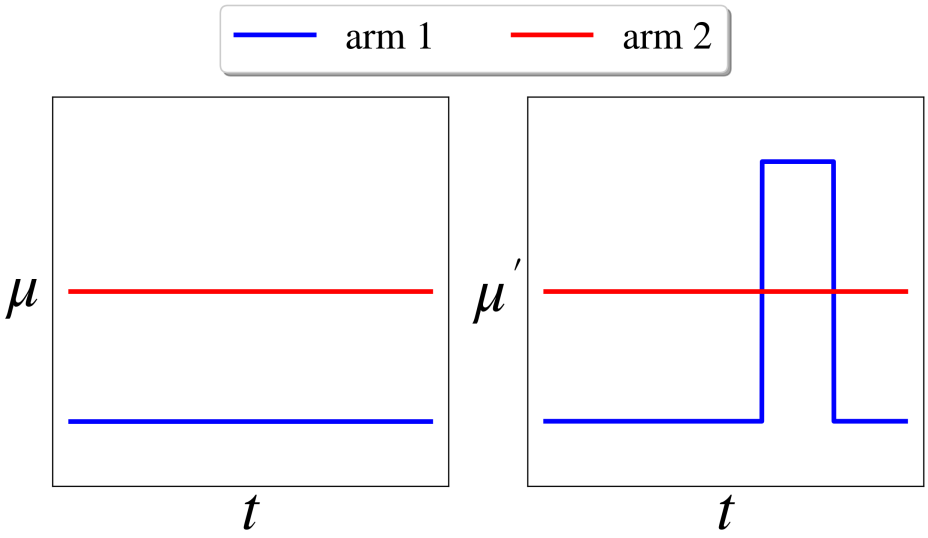


Figure 5.1: The reward functions μ and μ'. A policy with low regret on μ cannot achieve low regret on μ'.

Proposition 5.1.4 — Theorem 31.2, Lattimore and Szepesvári (2020). If a policy π performs (in expectation) a regret $\mathbb{E}[R_T(\pi,\mu)]$ on a 2-arm stationary instance μ, one can find a piece-wise stationary instance μ' with only two breakpoints such that, for a sufficiently long horizon T, the regret is lower bounded by

$$\mathbb{E}\left[R_T(\pi,\mu')\right] \geq \frac{T}{22\mathbb{E}\left[R_T(\pi,\mu)\right]}\,.$$

Corollary 5.1.5 Let π a minimax optimal policy on the piece-wise stationary setups. Then, for a sufficiently large horizon T, there exists a universal constant C such that for all the 2-arm stationary problems μ,

$$\mathbb{E}[R_T(\pi,\mu)] \geq C\sqrt{T}.$$

These results state that one cannot have simultaneously a near-optimal problem-dependent regret rate $\mathcal{O}(\log T)$ on stationary instances and the minimax optimal piece-wise stationary rate $\mathcal{O}\left(\sqrt{T}\right)$. It is very different from the stationary case (or even with the rested rotting bandits presented in the last section) where some algorithms are shown to perform optimally both problem-dependent and problem-independent wise (Ménard and Garivier 2017; Lattimore 2018).

Policies for piece-wise stationary bandits.

Softmax policies. For any sequence generated by an oblivious adversary, `Exp3.S` (Auer et al. 2003) - an extension of `Exp3`- is guaranteed to achieve $\mathcal{O}\left(\sqrt{K\Upsilon_T T \log(KT)}\right)$ regret against the best policy among the ones which change arms at most $\Upsilon_T - 1$ times. The bound holds in the special case where the adversary generates the reward with noisy piece-wise stationary functions. In that case, the pseudo-regret definition is equivalent to the piece-wise stochastic regret defined in Equation 5.3. Indeed, the optimal policy is included in the set of $\mathcal{O}\left((KT)^{\Upsilon_T}\right)$ policies with at most $\Upsilon_T - 1$ change of arms.

Passive forgetting policies. `D-UCB` (Kocsis and Szepesvári 2006) and `SW-UCB` (Garivier and Moulines 2011) are two ucb index policies which forget the older sample either by a discount factor or by a sliding window mechanism. The confidence interval increases when an arm has not been pulled for many rounds. When they are adequately tuned, these policies achieve respectively $\mathcal{O}\left(\sqrt{K\Upsilon_T T}\log T\right)$ and $\mathcal{O}\left(\sqrt{K\Upsilon_T T\log T}\right)$ minimax regret rate. While these policies do not improve the rate of `Exp3.S`, they are deterministic and more explainable.

Change-detection policies. Instead of throwing away old samples at a fixed pace, one could remove samples from the index only when they notice a change in the arm's mean. This is the spirit of the Change-Detection ucb algorithms. These algorithms have three components: an ucb index, a change-detection subroutine, and a fixed active exploration rate (either deterministic or random pulls). The active exploration rate is meant to detect the arms which change from suboptimal value to optimal ones (like in Figure 5.1). The optimal budget dedicated to active exploration scales with $\mathcal{O}\left(\sqrt{K\Upsilon_T T}\right)$.

`M-UCB` (Cao et al. 2019) uses a simple change detector which compares the average of the last $w/2$ samples with the average of the before last $w/2$ ones and check whether the difference is significant or not. The optimal tuning of the parameter w depends on the

value of Υ_T: if changes are large and frequent, one should choose a small value of w; if changes are small and sparse, one should choose a large value of w.

`CUSUM-UCB` (F. Liu et al. 2018) uses a change detector which constructs two random walks based on the upper and lower deviation of the new samples compared to the mean of the M first ones. If one of the random walks reaches a threshold h, then the change detector triggers. The random walks are negatively biased with a small value ε to prevent the natural deviation to trigger the change detector. Again, the optimal value of the parameters M, ε and h depends on the number of changes Υ_T.

`GLR-UCB` (Besson and Kaufmann 2019) uses the Gaussian Likelihood Ratio change detector. This change detector scans all the samples to detect any size of change on any period with high probability. The probability parameter only needs the knowledge of the horizon T to achieve near-optimal minimax bound. Mukherjee and Maillard (2019) introduces a very similar algorithm but study the assumption where all the arms change their value significantly at each breakpoint. With this assumption, they do not need active exploration and recover problem-dependent bound $\mathcal{O}(\log T)$.

On the theoretical side, the analysis often assumes that each change is large enough to be detected before the next change. Indeed, after the detection of the breakpoint, they use the analysis of `UCB` on each stationary batch. Before the change detection, they do not provide any non-trivial bound on the quality of the selected arm.

Agnostic policies. Auer et al. (2019) consider the problem with no assumption on the change-point detectability. They propose `AdSwitch`, which also uses a parameter-free change-detection subroutine but with an elimination policy: it pulls arms in a round-robin way in a refined set of good arms. Arms are excluded from this set when they demonstrate with high probability that they underperform. The bad arms are also actively explored with consecutive sampling: the algorithm selects at random an arm and a deviation size Δ and pulls the arm the right number of rounds to detect if there is a change of size Δ in the arm's value. Chen et al. (2019) extend this technique to the contextual bandits problem.

A previous attempt (Cheung et al. 2019) to solve this problem uses an expert aggregation bandit algorithm (e.g. `Exp4`) to select between different tuning of `SW-UCB`. Yet expert aggregation of bandit algorithm is problematic (Agarwal et al. 2017; Besson et al. 2018), and Cheung et al. (2019) has to run each copy by batch with full restart. This technique leads to a suboptimal rate $\widetilde{\mathcal{O}}\left(\sqrt{K \max\left(\Upsilon_T, \sqrt{T}\right) T}\right)$.

5.1.3 Variation budget bandits

Besbes et al. (2014) introduce the limited variation budget bandits, a restless setting where at each round Nature can modify the reward value of any arm but with a limited total variation budget V_T at the round T.

Assumption 5.1.6 $\mu_i : \mathbb{N}^\star \to [-V_T, 0]$ are functions of the time t with V_T a positive constant. Moreover, we have that

$$\sum_{t=1}^{T-1} \sup_{i \in \mathcal{K}} |\mu_i(t+1) - \mu_i(t)| \leq V_T \,. \tag{5.4}$$

Lower Bound

Proposition 5.1.7 — Besbes et al. (2014). For any strategy π, there exists a variation budget bandit scenario with means $\{\mu_i(t)\}_{i,t}$ satisfying Assumption 5.1.6 with a budget $V_T \geq \sigma\sqrt{\frac{K}{8T}}$ such that

$$\mathbb{E}\left[R_T(\pi)\right] \geq \frac{1}{16\sqrt{2}} \left(\sigma^2 V_T K T^2\right)^{1/3}.$$

In the next section, we prove a stronger statement, using only non-increasing reward functions. Yet, there is no additional difficulty. While the two Assumptions 5.1.2 and 5.1.6 leads to different regret rate (see Proposition 5.1.3), the proof (see e.g. Lemma 5.1.11 in the next subsection) shows that there is a strong similarity between the two problems, at least from a minimax perspective.

Policies for variation budget bandits. Most of the algorithms presented for the piecewise stationary case are also near-optimal for the variation budget case. Indeed, Besbes et al. (2014) show that `Exp3.S` also learns in the variation budget setup. They also present `Rexp3`, an algorithm based on `Exp3` with periodic restart which recovers a similar guarantee than `Exp3.S`. Cheung et al. (2019) and Russac et al. (2019) extend `SW-UCB` and `D-UCB` to the linear bandit setting with variation budget. Chen et al. (2019) proves that `AdSwitch` is also optimal in the variation budget setting. However, change-detection ucb algorithms are not proved to perform well in the variation budget setting. Indeed, their proofs use the proof of `UCB` on each stationary batch. In the variation budget setup, there is no stationary batch, which makes these algorithms harder to analyze.

5.1.4 The restless rotting assumption

Assumption 5.1.8 Reward functions $\{\mu_i\}_i$ are non-increasing with t.

We use this Assumption in conjunction with Assumption 5.1.2 or 5.1.6.

R With the rotting assumption, the variation budget assumption is very similar to the bounded assumption. Indeed, any set of decreasing functions $\mu_i : \mathbb{N}^\star \to [-V, 0]$ satisfies Equation 5.4 with $V_T = KV$. Reciprocally, any set of functions satisfying Equation 5.4 with $\mu_i(1) \in [-V_T, 0]$ are bounded in $[-2V_T, 0]$.

Lower bounds. We show that our additional decreasing assumption does not change the minimax rates of the two settings. This is an adaptation of the proof of Besbes et al. (2014) where we only use rotting functions.

Proposition 5.1.9 For any strategy π, there exists a rotting piece-wise stationary bandit scenario with means $\{\mu_i(t)\}_{i,t}$ satisfying Assumptions 5.1.2 and 5.1.8 with $\Upsilon_T \leq \left(\frac{32V^2T}{K\sigma^2}\right)^{1/3}$ such that,

$$\mathbb{E}\left[R_T(\pi)\right] \geq \frac{\sigma}{32}\sqrt{\Upsilon_T KT}\,.$$

Proposition 5.1.10 For any strategy π, there exists a rotting variation budget bandit scenario with means $\{\mu_i(t)\}_{i,t}$ satisfying Assumptions 5.1.6 and 5.1.8 with a budget $V_T \geq \sigma\sqrt{\frac{K}{8T}}$ such that,

$$\mathbb{E}\left[R_T(\pi)\right] \geq \frac{1}{16\sqrt{2}}\left(\sigma^2 V_T K T^2\right)^{1/3}.$$

The condition on Υ_T in Proposition 5.1.9 follows from the previous remark: if V is too small compared to Υ_T, then we have a budget constraint - with associated lower bound in Proposition 5.1.10 - rather than a breakpoint constraint.

Proof Our proof builds a set of rotting piece-wise stationary problems with an evenly spaced set of $\Upsilon - 1$ breakpoints. The adversary can choose the distance between arms $\Delta = \frac{1}{4}\sqrt{\frac{\sigma^2 K\Upsilon}{2T}}$ at the maximum such that the best arm is barely identifiable between two breakpoints (see Lemma 5.1, Auer et al. (2003)). At each breakpoint, each arm's value decreases by Δ or 2Δ. Even if the set of breakpoints would be known, the learner does not know which arm is the best on each stationary part. Hence, in the worst case, she suffers at least the sum of the minimax regret of Υ stationary bandits problems with horizon $\frac{T}{\Upsilon}$, *i.e.* $\mathcal{O}\left(\sqrt{K\Upsilon T}\right)$. In the piece-wise stationary setting, we can simply identify $\Upsilon = \Upsilon_T$. In the variation budget setting, the adversary has a constraint over $\Upsilon\Delta = \frac{1}{4}\sqrt{\frac{\sigma^2 K\Upsilon^3}{2T}} = \mathcal{O}(V_T)$. Hence, when the budget is limited, the adversary can choose up to $\Upsilon = \mathcal{O}\left(T^{1/3}\right)$ breakpoints such that the suboptimal arms are "sufficiently" far from the best one (*i.e* at Δ). This dependence on T leads to the increased regret rate of $\mathcal{O}\left(T^{2/3}\right)$.

Lemma 5.1.11 Let $\Upsilon \in \{1,\dots,T\}$ and $\left\{\tau_k \triangleq \left\lceil \frac{T}{\Upsilon}\right\rceil \text{ if } k \leq T \bmod \Upsilon \text{ else } \left\lfloor \frac{T}{\Upsilon}\right\rfloor\right\}_{k\leq\Upsilon}$. We call $t_k = \sum_{k'=1}^{k}\tau_{k'}$ and $t_0 = 0$. Consider a family of piece-wise stationary bandits indexed by a vector $i^\star \in (\{0\}\cup\mathscr{K})^\Upsilon$ as follows: arm i is a Gaussian distribution $\mathcal{N}(\mu_i(t),\sigma)$ such that

$$\forall k \in \{0,\dots,\Upsilon-1\}\,,\ \forall t \in \{t_{k-1}+1,\dots,t_k\}\,,\ \mu_i(t) = \begin{cases} -k\Delta \text{ if } i = i_k^\star \\ -(k+1)\Delta \text{ else.}\end{cases}$$

We denote by $\mathbb{E}_{i^\star}$ the expectation under the problem indexed by $i^\star$. Then, if $\Delta = \frac{1}{4}\sqrt{\frac{\sigma^2 K\Upsilon}{2T}}$, for any policy π :

$$\exists i^\star \in (\{0\} \cup \mathscr{K})^\Upsilon,\ \mathbb{E}_{i^\star}\big[R_T(\pi)\big] \geq \frac{\sqrt{\sigma^2 KT\Upsilon}}{32}.$$

Proof. Note that when $i_k^\star = 0$ then all the arms share the same means. We also define the vector $i^\star_{-k}$ equals to $i^\star$ with the coordinate k empty and for $i \in \mathscr{K}$ the vector $(i^\star_{-k}, i)$ as the vector where we fill the empty coordinate with i. We fix a policy π and we will lower bound its average regret on the bandits problem indexed by $i^\star \in \mathscr{K}^\Upsilon$

$$\begin{aligned}\frac{1}{K^\Upsilon}\sum_{i^\star\in\mathscr{K}^\Upsilon}\mathbb{E}_{i^\star}\big[R_T(\pi)\big] &= \frac{1}{K^\Upsilon}\sum_{i^\star\in\mathscr{K}^\Upsilon}\sum_{k=1}^{\Upsilon}\Delta\mathbb{E}_{i^\star}[\tau_k - N^k_{i^\star_k}] \\ &= \Delta\left(T - \frac{1}{K^\Upsilon}\sum_{i^\star\in\mathscr{K}^\Upsilon}\sum_{k=1}^{\Upsilon}\mathbb{E}_{i^\star}[N^k_{i^\star_k}]\right),\end{aligned}$$

where N_i^k is the number of pulls of arm i during epoch k. Thus we need to upper bound the following quantity

$$\frac{1}{K^\Upsilon}\sum_{i^\star\in\mathscr{K}^\Upsilon}\sum_{k=1}^{\Upsilon}\mathbb{E}_{i^\star}[N^k_{i^\star_k}] = \sum_{k=1}^{\Upsilon}\frac{1}{K^{\Upsilon-1}}\sum_{i^\star_{-k}\in\mathscr{K}^{\Upsilon-1}}\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}_{(i^\star_{-k},i)}[N_i^k].$$

Using the contraction of the entropy for the bounded random variable N_i^k/τ_k then the Pinsker inequality (see Garivier et al. (2019)) we get

$$2\left(\frac{1}{\tau_k K}\sum_{i=1}^{K}\mathbb{E}_{(i^\star_{-k},i)}[N_i^k] - \frac{1}{\tau_k K}\sum_{i=1}^{K}\mathbb{E}_{(i^\star_{-k},0)}[N_i^k]\right)^2 \leq \frac{1}{K}\sum_{i=1}^{K}\mathbb{E}_{(i^\star_{-k},0)}[N_i^k]\frac{\Delta^2}{2\sigma^2},$$

since problems $(i^\star_{-k}, i)$ and $(i^\star_{-k}, 0)$ differ only by a gap Δ on the arm i during epoch k. Thanks to the fact that $\sum_i N_i^k \leq \tau_k$ we get

$$\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}_{(i^\star_{-k},i)}[N_i^k] \leq \frac{\tau_k}{K} + \frac{\Delta}{2\sigma\sqrt{K}}\tau_k^{3/2}.$$

Putting all together we have for $K \geq 2$

$$\frac{1}{K^\Upsilon}\sum_{i^\star\in\mathscr{K}^\Upsilon}\mathbb{E}_{i^\star}\big[R_T(\pi)\big] \geq \left(\frac{T}{2} - \sum_{k=1}^{\Upsilon}\frac{\tau_k^{3/2}\Delta}{2\sigma\sqrt{K}}\right)\Delta.$$

We have $\tau_k = \left\lfloor \frac{T}{\Upsilon} \right\rfloor$ or $\tau_k = \left\lceil \frac{T}{\Upsilon} \right\rceil$ such that $\sum_{k=1}^{\Upsilon}\tau_k = T$. Hence, we have that $\tau_k \leq 2T/\Upsilon$ which leads to

$$\frac{1}{K^\Upsilon}\sum_{i^\star\in\mathscr{K}^\Upsilon}\mathbb{E}_{i^\star}\big[R_T(\pi)\big] \geq \left(\frac{1}{2}T - \frac{\sqrt{2}T^{3/2}\Delta}{\sigma\sqrt{K\Upsilon}}\right)\Delta.$$

Choosing $\Delta = \frac{1}{4}\sqrt{\frac{\sigma^2 K\Upsilon}{2T}}$, we get

$$\frac{1}{K^\Upsilon}\sum_{i^\star\in\mathscr{K}^\Upsilon}\mathbb{E}_{i^\star}\left[R_T(\pi)\right] \geq \frac{1}{4}\sqrt{\frac{\sigma^2 K\Upsilon}{2T}}\left(\frac{1}{4}T\right) \geq \frac{\sqrt{\sigma^2 KT\Upsilon}}{32}.$$

We can conclude by noticing that the average expected regret across the problem set is lesser or equal to the maximum across the same problem set. ■

Proof of Proposition 5.1.9. This result directly follows from Lemma 5.1.11 by choosing $\Upsilon = \Upsilon_T$. Indeed, the set of problems $\left\{i^\star \in (\{0\}\cup\mathscr{K})^{\Upsilon_T}\right\}$ satisfy Assumptions 5.1.2 and 5.1.8 as soon as $\Upsilon_T\Delta \leq V$, *i.e.* $\Upsilon_T \leq \left(\frac{32V^2T}{K\sigma^2}\right)^{1/3}$. ■

Proof of Proposition 5.1.10. We want to use Lemma 5.1.11 but we need to make the set of problems $\left\{i^\star \in (\{0\}\cup\mathscr{K})^{\Upsilon_T}\right\}$ comply with Assumption 5.1.6. First, the function are bounded by $-V_T$. Hence, we need :

$$\Upsilon\Delta \leq V_T. \tag{5.5}$$

Second, the total variation is bounded according to Equation 5.4. When t is not a breakpoint, the variation is null. At each breakpoint, the maximal variation across the arm is 2Δ. For $\Upsilon - 1$ breakpoint, we have that

$$2\Delta(\Upsilon - 1) \leq V_T. \tag{5.6}$$

Since $2\Delta(\Upsilon-1) \leq \frac{\sigma}{2}\sqrt{\frac{K}{2T}}\Upsilon^{3/2}$, we choose

$$\Upsilon = \min\left(\max\left(\left\lfloor 2\left(\frac{V_T^2 T}{K\sigma^2}\right)^{1/3}\right\rfloor, 1\right), T\right). \tag{5.7}$$

By construction, 5.7 satisfies 5.6. Moreover, when $\Upsilon > 1$, 5.6 is more restrictive than 5.5. For $\Upsilon = 1$, we simply assume $\Delta \leq V_T$, *i.e.* $V_T \geq \sigma\sqrt{\frac{K}{8T}}$.

Plugging 5.7 in Lemma 5.1.11 allows us to conclude

$$\mathbb{E}\left[R_T(\pi)\right] \geq \frac{1}{16\sqrt{2}}V_T^{1/3}\sigma^{2/3}K^{1/3}T^{2/3}.$$

■

5.2 Analysis of adaptive window policies on restless rotting bandits.

In Chapter 4, we presented four adaptive window policies (`FEWA`, `RAW-UCB`, `EFF-FEWA`, `EFF-RAW-UCB`). In this section, we will show that the exact same policies are able to match interesting upper bounds on the restless problems. The proof of the regret upper bounds in the rested case uses three main steps. First, we design one favorable event per round on which all the constructed statistics concentrate on a well-chosen confidence region, such that it holds with sufficiently high probability. This part does not use that we faced a rested non-stationary environment; it only uses the concentration of independent subgaussian variables which remains true in our restless problem due to Doob's optional skipping. Hence, we restate Propositions 4.2.1 and 4.5.9,

Proposition 5.2.1 We recall that, for any round t and confidence $\delta_t \triangleq 2t^{-\alpha}$, we define

$$\xi_t^\alpha \triangleq \left\{\forall i \in \mathscr{K},\ \forall n \leq t-1,\ \forall h \leq n, \left|\widehat{\mu}_i^h(t,\pi) - \overline{\mu}_i^h(t,\pi)\right| \leq c(h,\delta_t)\right\}$$
$$\xi_{t,m}^\alpha \triangleq \left\{\forall i \in \mathscr{K}, \forall n \leq t-1, \forall h_j \in H_{i,m}(n), \left|\widehat{\mu}_{i,\texttt{eff}}^{h_j}(t,\pi) - \overline{\mu}_{i,\texttt{eff}}^{h_j}(t,\pi)\right| \leq c(h_j,\delta_t)\right\}$$

with $c(h,\delta_t) \triangleq \sqrt{2\sigma^2 \log(2/\delta_t)/h}$. Then, for a policy π which pulls each arms once at the beginning, and for all $t > K$,

$$\mathbb{P}\left[\overline{\xi_t^\alpha}\right] \leq \frac{Kt^2\delta_t}{2} = Kt^{2-\alpha} \text{ and } \mathbb{P}\left[\overline{\xi_{t,m}^\alpha}\right] \leq 3Kt\delta_t = 6Kt^{1-\alpha}.$$

Then, we use the mechanics of the algorithms to relate the average past performance of the selected arm with the current best value of the arms. As we noticed in the proofs (see e.g. the proof of Lemma 4.2.2), we do not use the rested aspect of the problem. In fact, these results hold for a more general reward function $\mu_i(t,n)$ which is non-increasing with both t and n. Therefore, we also restate Lemmas 4.2.2, 4.2.3 and 4.5.10,

Lemma 5.2.2 At any round t on favorable event ξ_t^α (respectively, $\xi_{t,2}^\alpha$), if arm i_t is selected by $\pi \in \{\pi_{\mathrm{F}}, \pi_{\mathrm{R}}\}$ (respectively, $\pi \in \{\pi_{\mathrm{EF}}, \pi_{\mathrm{ER}}\}$ tuned with $m = 2$), for any $h \leq N_{i,t-1}$, the average of its h last pulls cannot deviate significantly from the best available arm at that round, i.e.,

$$\overline{\mu}_{i_t}^h(t,\pi) \geq \max_{i \in \mathscr{K}} \mu_i(t) - \frac{C_\pi}{\sqrt{2\alpha}} c(h,\delta_t) \quad \text{with} \quad \begin{cases} C_{\pi_{\mathrm{R}}} = 2\sqrt{2\alpha} \text{ and } C_{\pi_{\mathrm{ER}}} = \frac{4\sqrt{\alpha}}{\sqrt{2}-1} \\ C_{\pi_{\mathrm{F}}} = 4\sqrt{2\alpha} \text{ and } C_{\pi_{\mathrm{EF}}} = \frac{8\sqrt{\alpha}}{\sqrt{2}-1} \end{cases}.$$

Last, we use a specific rested regret decomposition to show that our algorithms are near-optimal both problem-dependent and problem-independent wise on rested rotting bandits. Unfortunately, this part cannot be used for the restless analysis. However, with a specific restless regret decomposition (see the proof in Subsection 5.2.1), we can show that our policies matches the two aforementioned lower bounds up to poly-logarithmic terms without any knowledge of the horizon T nor Υ_T or V_T.

Theorem 5.2.3 Let $\pi \in \{\pi_F, \pi_R\}$ tuned with $\alpha \geq 4$ or $\pi \in \{\pi_{EF}, \pi_{ER}\}$ tuned with $\alpha \geq 3$ and $m = 2$. For any piece-wise stationary bandit scenario with means $\{\mu_i(t)\}_{i,t}$ satisfying Assumptions 5.1.2 and 5.1.8 with $\Upsilon_T - 1$ change-points, π suffers an expected regret

$$\mathbb{E}[R_T(\pi)] \leq C_\pi \sigma \sqrt{\log T} \left(\sqrt{\Upsilon_T K T} + \Upsilon_T K\right) + 6KV.$$

Theorem 5.2.4 Let $\pi \in \{\pi_F, \pi_R\}$ tuned with $\alpha \geq 4$ or $\pi \in \{\pi_{EF}, \pi_{ER}\}$ tuned with $\alpha \geq 3$ and $m = 2$. For any variation budget bandit scenario with means $\{\mu_i(t)\}_{i,t}$ satisfying Assumptions 5.1.6 and 5.1.8 with variation budget V_T, π suffers an expected regret

$$\mathbb{E}[R_T(\pi)] \leq 4\left(C_\pi^2 \sigma^2 V_T K T^2 \log T\right)^{1/3} + 2\left(C_\pi \sigma V_T^2 K^2 T \sqrt{\log T}\right)^{1/3} + 6V_T K.$$

The remaining terms are of second-order when $KV_T \leq \mathcal{O}(T)$, which is a necessary condition for the problem to be learnable (see Proposition 5.1.10).

Are rotting restless bandits easier? Learning at the minimax rate without knowing Υ_T or V_T was achieved in the non-rotting setup by significantly more complex algorithms. For instance, Auer et al. (2019) use a combination of filtering on the set of potentially good arms, forced exploration planning on identified bad arms, and full restart of the algorithm when a change is detected. This algorithmic complexity has a performance cost, as `AdSwitch` is guaranteed to achieve 56 times the leading term in Theorem 5.2.3. Moreover, these algorithms rely on doubling trick when the horizon is unknown, which also has a regret cost compared to intrinsically anytime algorithms (Besson and Kaufmann 2018).

Yet, Proposition 5.1.9 and 5.1.10 show that the rotting assumption do not improve the minimax rate for the two considered setups. Interestingly both these lower bounds are matched by (tuned) `Exp3.S` (Auer et al. 2003), an algorithm originally designed for switching best arm in adversarial sequences of rewards. This is comparable to the fixed best arm world: adversarial and stochastic bandits share the same minimax rate which is matched in both setups by `Exp3`. The main interest of the stochastic assumption is to allow for *problem dependent analysis*. For the stochastic stationary bandits, it leads to a stronger $\mathcal{O}(\log(T))$ bounds. In the (non-rotting) piece-wise stationary setting, we argued in Subsection 5.1.2 that the learner has to maintain $\mathcal{O}\left(\sqrt{T}\right)$ exploratory pulls to shield against increase of currently suboptimal arm (see Proposition 5.1.4 and Corollary 5.1.5).

The decreasing Assumption 5.1.8 excludes the problems where suboptimal arms increases to become optimal from the set of possible problems. Theorem 5.2.5 shows that not only `RAW-UCB` is able to recover the $\mathcal{O}(\log(T))$ on stationary problems but also recovers the same rate on each batch of a rotting piece-wise stationary problem.

Theorem 5.2.5 Let $\pi \in \{\pi_{\mathrm{F}}, \pi_{\mathrm{R}}\}$ tuned with $\alpha \geq 4$ or $\pi \in \{\pi_{\mathrm{EF}}, \pi_{\mathrm{ER}}\}$ tuned with $\alpha \geq 3$ and $m = 2$. For any piece-wise stationary bandit scenario with means $\{\mu_i(t)\}_{i,t}$ satisfying Assumptions 5.1.2 and 5.1.8 with $\Upsilon_T - 1$ change-points, π suffers an expected regret

$$\mathbb{E}[R_T(\pi)] \leq \sum_{k=0}^{\Upsilon_T - 1} \sum_{i \in \mathscr{K}} \frac{C_\pi^2 \sigma^2 \log T}{\Delta_{i,k}} + C_\pi \sigma \Upsilon_T K \sqrt{\log T} + 6KV.$$

Notice that Mukherjee and Maillard (2019) use a different assumption to recover a similar problem-dependent bound. Indeed, they assume that all the arms change at the same time. In the counter-example displayed on Figure 5.1, it is important that the arm 1 changes its value while arm 2 is stationary. Indeed, in that case, the learner cannot infer the increase on arm 2 by sampling arm 1. Hence, the assumption of Mukherjee and Maillard (2019) excludes this counter-example of the set of possible problems. That is why they were able to provide a logarithmic problem-dependent bound.

5.2.1 Proofs

Sketch.

We start by separating the regret on the bad events $\overline{\xi_t^\alpha}$ from the good events ξ_t^α. According to Proposition 4.2.1, the bad events $\overline{\xi}_t$ have low probability for appropriate α. For $\alpha = 4$, they weigh at most $\mathscr{O}(KV)$ in the expected regret. On the good events, we write:

$$R_T(\pi) = \sum_{t=1}^{T} \mu_{i_t^\star}(t) - \overline{\mu}_{i_t}^{h_t}(t,\pi) + \overline{\mu}_{i_t}^{h_t}(t,\pi) - \mu_{i_t}(t). \tag{5.8}$$

Notice that Lemma 5.2.2 can bound the first difference for any h_t. When the reward is piece-wise stationary, we can select h_t such that we include all the pulls of arm i_t from the current stationary batch. If there is none, then it is the first pull of arm i_t in this batch. We handle these $\mathscr{O}(K\Upsilon_T)$ rounds separately (see Lemma 5.2.6). In the other cases, we note that the second difference is null because $\overline{\mu}_{i_t}^{h_t}(t,\pi) = \mu_{i_t}(t) = \mu_i^k$ by the piece-wise stationary assumption. The remaining of the proofs of Theorem 5.2.3 and 5.2.5 are then very similar to the analysis of Auer et al. 2002 on each stationary batch. Indeed, Lemma 5.2.2 is similar to the two confidence bounds guarantee of `UCB1`'s guarantee.

In the variation budget setting, there is no stationary batches. Hence, we cannot choose an h_t which cancels the second difference in Equation 5.8. Yet, we still decompose the rounds in Υ batches of equal length for the analysis. We choose h_t such that we include all the pulls of arm i_t from the current batch. For the sum of the first differences in Equation 5.8, there is no difference with the piece-wise stationary case and we can bound

$$\sum_{t=1}^{T} \mu_{i_t^\star}(t) - \overline{\mu}_{i_t}^{h_t}(t,\pi) \leq \widetilde{\mathscr{O}}\left(\sqrt{K\Upsilon T}\right). \tag{5.9}$$

We call $\Delta_i^k \triangleq \mu_i(t_k) - \mu_i(t_{k+1})$, the total variation of arm i in batch k. The sum of second differences in Equation 5.8 can be bounded as follows: on each batch of $T\Upsilon^{-1}$ rounds, each second difference is bounded by $\max_{i\in\mathscr{K}} \Delta_i^k$. When we sum over the batches, we get

$$\sum_{t=1}^{T} \overline{\mu}_{i_t}^{h_t}(t,\pi) - \mu_{i_t}(t) \leq \frac{T}{\Upsilon} \sum_{k=0}^{\Upsilon-1} \max_{i\in\mathscr{K}} \Delta_i^k \leq \frac{TV_T}{\Upsilon}. \tag{5.10}$$

Indeed, in the middle term, we have a maximum on the summed variation of arm i in batch k. On the right-hand side, we have V_T which bounds the sum over the rounds of maximal variation of the arms (see Equation 5.4). Thus, the right-hand side is larger because the maximum of sums is smaller than the sum of maximums. We can then choose $\Upsilon = \widetilde{\mathscr{O}}\left(T^{1/3}V_T^{2/3}K^{-1/3}\right)$ to minimise the sum of Equations 5.9 and 5.10. It leads to the leading term of our Theorem 5.2.4. Notice that we still have to handle the first pull of each arm in each batch. If we bound roughly each first pull by V_T, we would get $K\Upsilon V_T \sim \widetilde{\mathscr{O}}\left(V_T^{5/3}\right)$ which would be the leading term for large V_T. Our Lemma 5.2.6 is more careful such that it leads to a second order term when $KV_T \leq o(T)$.

Full proof

Lemma 5.2.6 — Bound on unfavorable events. Decomposition in unspecified batches. Bound on the first pull of each arm in each batch. Let an integer $\Upsilon \in \{1,\dots,T\}$.
Let $\mu_i : \mathbb{N}^\star \to [0,-V]$, the K decreasing reward functions.
Let $\{t_k \in \{1,\dots,T\} \,|t_k > t_{k-1}\}_{k\in\{1,\dots,\Upsilon-1\}}$ a set of $\Upsilon-1$ distinct rounds delimiting Υ batches. We set $t_0 = 0$ and $t_\Upsilon = T$.
We call $h_i^k \triangleq \sum_{t=t_k+1}^{t_{k+1}} \mathbb{1}(i_t = i)$ the number of pulls of arm i in batch k and $t_i^k(h)$ the time at which arm i is pulled for the h-th time since t_k+1. We also call $\mathscr{K}_k \triangleq \left\{i \in \mathscr{K} \,|h_i^k \geq 1\right\}$ the set of pulled arms in batch k.

Then, $\pi \in \{\pi_{\mathrm{R}}, \pi_{\mathrm{F}}\}$ run with $\alpha \geq 4$, or $\pi \in \{\pi_{\mathrm{ER}}, \pi_{\mathrm{EF}}\}$ run with $m = 2$ and $\alpha \geq 3$, suffers an expected regret of

$$\begin{aligned}\mathbb{E}[R_T(\pi)] \leq \mathbb{E}\left[\sum_{k=0}^{\Upsilon-1} \sum_{i\in\mathscr{K}_k} \sum_{t=t_k+1}^{t_{k+1}} \sum_{h=2}^{h_i^k} \mathbb{1}\left(t = t_i^k(h) \wedge \xi_t^\alpha\right)\left(\mu_\star(t) - \mu_i(t)\right)\right] \\ + C_\pi \sigma \Upsilon K \sqrt{\log T} + 6KV.\end{aligned}$$

Proof. We start by separating the favorable events from the unfavorable events:

$$R_T(\pi) = \underbrace{\sum_{t=1}^{T} \mathbb{1}(\xi_t^\alpha)(\mu_\star(t) - \mu_{i_t}(t))}_{R_T(\pi|\xi_t^\alpha)} + \underbrace{\sum_{t=1}^{T} \mathbb{1}\left(\overline{\xi_t^\alpha}\right)(\mu_\star(t) - \mu_{i_t}(t))}_{R_T(\pi|\overline{\xi_t^\alpha})}, \tag{5.11}$$

with $\mu_\star(t) \triangleq \max_{i\in\mathscr{K}} \mu_i(t)$. For $\alpha \geq 4$, we can bound the cost of the unfavorable events

thanks to Proposition 5.2.1,

$$\mathbb{E}\left[R_T(\pi|\overline{\xi_t^\alpha})\right] \leq \sum_{t=1}^{T} \mathbb{P}\left[\overline{\xi_t^\alpha}\right] V \leq \sum_{t=1}^{T} \frac{KV}{t^2} = \frac{KV\pi^2}{6} \leq 2KV. \tag{5.12}$$

On the favorable events, given any ordered set of $\Upsilon-1$ breakpoints $\{t_k\}$, we divide the horizon in Υ batches $\{t_k+1,\dots,t_{k+1}\}_{k\leq\Upsilon-1}$,

$$R_T(\pi|\xi_t^\alpha) \leq \sum_{k=0}^{\Upsilon-1}\sum_{t=t_k+1}^{t_{k+1}} \mathbb{1}\left(\xi_t^\alpha\right)\left(\mu_\star(t)-\mu_{i_t}(t)\right).$$

We define h_i^k the number of pulls of arm i in batch k, *i.e.* $h_i^k = \sum_{t=t_k+1}^{t_{k+1}} \mathbb{1}\left(i_t=i\right)$. We use $t_i^k(h)$ to designate the time at which arm i is pulled for the h-th time since t_k.

$$R_T(\pi|\xi_t^\alpha) \leq \sum_{k=0}^{\Upsilon-1}\sum_{t=t_k+1}^{t_{k+1}}\sum_{i\in\mathscr{K}_k}\sum_{h=1}^{h_i^k} \mathbb{1}\left(t_i^k(h)=t\wedge\xi_t^\alpha\right)\left(\mu_\star(t)-\mu_i(t)\right).$$

We split the regret on the first pulls of each batch,

$$\begin{aligned} R_T(\pi|\xi_t^\alpha) = &\underbrace{\sum_{k=0}^{\Upsilon-1}\sum_{t=t_k+1}^{t_{k+1}}\sum_{i\in\mathscr{K}_k} \mathbb{1}\left(t=t_i^k(1)\wedge\xi_t^\alpha\right)\left(\mu_\star(t)-\mu_i(t)\right)}_{FP} \\ &+\underbrace{\sum_{k=0}^{\Upsilon-1}\sum_{t=t_k+1}^{t_{k+1}}\sum_{i\in\mathscr{K}_k}\sum_{h=2}^{h_i^k} \mathbb{1}\left(t=t_i^k(h)\wedge\xi_t^\alpha\right)\left(\mu_\star(t)-\mu_i(t)\right)}_{OP}. \end{aligned} \tag{5.13}$$

Analysis of the first pulls. We call k_i^1, the index of the batch at which arm i is pulled for the first time (we assume that $T\geq K$). We call $\mathscr{K}_k^2 \triangleq \left\{i\in\mathscr{K}_k | k>k_i^1\right\}$, the set of arms pulled at least once during batch k and at least once in a batch before k. We split the regret due to the very first pull each arm from the other first pulls in each batch,

$$\begin{aligned} FP &= \sum_{k=0}^{\Upsilon-1}\sum_{i\in\mathscr{K}_k}\sum_{t=t_k+1}^{t_{k+1}} \mathbb{1}\left(t=t_i^k(1)\wedge\xi_t^\alpha\right)\left(\mu_\star(t)-\mu_i(t)\right) \\ &\leq \sum_{i\in\mathscr{K}}\left(0-\mu_i(t_i^{k_i^1}(1))\right) + \sum_{k=1}^{\Upsilon-1}\sum_{i\in\mathscr{K}_k^2}\sum_{t=t_k+1}^{t_{k+1}} \mathbb{1}\left(t=t_i^k(1)\wedge\xi_t^\alpha\right)\left(\mu_\star(t)-\mu_i(t)\right) \\ &= \sum_{i\in\mathscr{K}}\left(0-\mu_i(t_i^{k_i^1}(1))\right) \\ &\quad + \sum_{k=1}^{\Upsilon-1}\sum_{i\in\mathscr{K}_k^2}\sum_{t=t_k+1}^{t_{k+1}} \mathbb{1}\left(t=t_i^k(1)\wedge\xi_t^\alpha\right)\left(\mu_\star(t)-\overline{\mu}_i^1(t,\pi)+\overline{\mu}_i^1(t,\pi)-\mu_i(t)\right). \end{aligned}$$

The inequality is justified because $\mu_i(t) \leq 0$ for all t. In the last equation, we simply introduce $\overline{\mu}_i^1(t,\pi)$, the last pulled sample of arm i, which is well defined after the first pull of each arm. According to Lemma 5.2.2, the first difference is bounded on the high-probability event ξ_t^α,

$$\sum_{t=t_k+1}^{t_{k+1}} \mathbb{1}\left(t = t_i^k(1) \wedge \xi_t^\alpha\right)\left(\mu_\star(t) - \overline{\mu}_i^1(t,\pi)\right) \leq \frac{C_\pi}{\sqrt{2\alpha}} c(1, 2T^{-\alpha}) = C_\pi \sigma \sqrt{\log T}. \quad (5.14)$$

We will show that we can telescope the second sum. First, we notice that we can collapse the sum on t using $\mathbb{1}\left(t = t_i^k(1)\right)$. Moreover, ξ_t^α will not be needed: hence we can drop $\mathbb{1}\left(\xi_t^\alpha\right) \leq 1$.

$$\sum_{t=t_k+1}^{t_{k+1}} \mathbb{1}\left(t = t_i^k(1) \wedge \xi_t^\alpha\right)\left(\overline{\mu}_i^1(t,\pi) - \mu_i(t)\right) \leq \overline{\mu}_i^1(t_i^k(1),\pi) - \mu_i(t_i^k(1)). \quad (5.15)$$

For a given batch k on which arm i is pulled, the precedent reward sample has a mean $\overline{\mu}_i^1\left(t_i^k(1),\pi\right)$. This sample is the last pull of the last batch k' before k on which arm i is pulled. Hence, its mean is smaller than the mean of the first pull on this same batch k' because the reward is decreasing. Hence, the sum can telescope

$$\begin{aligned}
&\sum_{i\in\mathscr{K}}\left(0-\mu_i(t_i^{k_i^1}(1))\right) + \sum_{k=1}^{\Upsilon-1}\sum_{i\in\mathscr{K}_k^2}\sum_{t=t_k+1}^{t_{k+1}} \mathbb{1}\left(t = t_i^k(1)\wedge\xi_t^\alpha\right)\left(\overline{\mu}_i^1(t,\pi)-\mu_i(t)\right) \\
&\qquad \leq \sum_{i\in\mathscr{K}}\left\{0-\mu_i(t_i^{k_i^1}(1)) + \sum_{k=k_i^1+1}^{\Upsilon-1}\mathbb{1}\left(h_i^k\geq 1\right)\left(\overline{\mu}_i^1(t_i^k(1),\pi)-\mu_i(t_i^k(1))\right)\right\} \\
&\qquad \leq \sum_{i\in\mathscr{K}}\left(0-\mu_i(T)\right) \leq KV\,. \quad (5.16)
\end{aligned}$$

The first inequality uses the definition of $\mathscr{K}_k^2$ along with Equation 5.15. The second inequality follows from the telescoping argument presented above. The third inequality uses that $\mu_i(T) \geq -V$. Gathering Equation 5.14 and 5.16, we can bound the term FP (defined in Equation 5.13)

$$FP \leq KV + \sum_{k=1}^{\Upsilon-1}\sum_{i\in\mathscr{K}_k^2} C_\pi\sigma\sqrt{\log T} \leq KV + C_\pi\sigma\Upsilon K\sqrt{\log T}. \quad (5.17)$$

Conclusion. From Equation 5.11, we can bound the expected regret on the unfavorable events thanks to Equation 5.12. On the favorable events, we can split the rounds in batches on which we isolate the first pull of each arm on each batch thanks to Equation 5.13. Finally, we bound the regret due to these first pulls thanks to Equation 5.17, and for $\alpha \geq 4$,

$$\begin{aligned}
\mathbb{E}\left[R_T(\pi)\right] \leq \mathbb{E}&\left[\sum_{k=0}^{\Upsilon-1}\sum_{i\in\mathscr{K}_k}\sum_{t=t_k+1}^{t_{k+1}}\sum_{h=2}^{h_i^k}\mathbb{1}\left(t = t_i^k(h)\wedge\xi_t^\alpha\right)\left(\mu_\star(t)-\mu_i(t)\right)\right] \\
&+ C_\pi\sigma\Upsilon K\sqrt{\log T} + 3KV.
\end{aligned}$$

For the efficient algorithms, we can use the same proof with $\xi^{\alpha}_{t,2}$ and get for $\alpha \geq 3$,

$$\mathbb{E}\left[R_T(\pi)\right] \leq \mathbb{E}\left[\sum_{k=0}^{\Upsilon-1}\sum_{i\in\mathscr{K}_k}\sum_{t=t_k+1}^{t_{k+1}}\sum_{h=2}^{h_i^k}\mathbb{1}\left(t=t_i^k(h)\wedge\xi_t^{\alpha}\right)\left(\mu_\star(t)-\mu_i(t)\right)\right] + C_\pi\sigma\Upsilon K\sqrt{\log T}+6KV.$$

■

Lemma 5.2.7 — Analysis of the second pulls in each batch under the favorable events.. Let $\Delta_i^k \triangleq \mu_i(t_k+1)-\mu_i(t_{k+1})$, the decrement of arm i in batch k. For any arm i and any consecutive rounds $\{t_k+1,\dots,t_{k+1}\}$ such that i is pulled $h_i^k \geq 1$ times, the regret due to the pulls after the first one can be bounded under the favorable events,

$$\sum_{t=t_k+1}^{t_{k+1}}\sum_{h=2}^{h_i^k}\mathbb{1}\left(t=t_i^k(h)\wedge\xi_t^{\alpha}\right)\left(\mu_\star(t)-\mu_i(t)\right) \leq \left(h_i^k-1\right)\Delta_i^k+\sum_{h=2}^{h_i^k}\mathbb{1}\left(\xi^{\alpha}_{t_i^k(h)}\right)\left(\mu_\star(t_i^k(h))-\overline{\mu}_i^{h-1}(t_i^k(h),\pi)\right).$$

Proof. We call $\Delta_i(t,t') \triangleq \mu_i(t)-\mu_i(t')$ the variation of arm i between times t and t'. As a short notation, we refer to $\Delta_i^k \triangleq \Delta_i(t_k+1,t_{k+1})$ for the variation of arm i in batch k.

$$\forall h \leq h_i^k, \quad \mu_i(t_i^k(h)) \geq \mu_i(t_{k+1}) = \mu_i(t_k+1)-\Delta_i^k \geq \overline{\mu}_i^{h-1}(t_i^k(h),\pi)-\Delta_i^k. \tag{5.18}$$

The two inequalities are justified by the rewards decay. Indeed, any pull in batch k has a higher reward than the value of arm i at the end of the batch t_{k+1}. Moreover, the value at the beginning of the batch is higher that any average of h value in this batch. The middle equality follows from the definition of Δ_i^k.

Then, we plug Equation 5.18 in the left hand side of our claim,

$$\begin{aligned}
&\sum_{t=t_k+1}^{t_{k+1}}\sum_{h=2}^{h_i^k}\mathbb{1}\left(t=t_i^k(h)\wedge\xi_t^{\alpha}\right)\left(\mu_\star(t)-\mu_i(t)\right)\\
&\quad=\sum_{h=2}^{h_i^k}\mathbb{1}\left(\xi^{\alpha}_{t_i^k(h)}\right)\left(\mu_\star(t_i^k(h))-\mu_i(t_i^k(h))\right)\\
&\quad\leq\sum_{h=2}^{h_i^k}\mathbb{1}\left(\xi^{\alpha}_{t_i^k(h)}\right)\left(\mu_\star(t_i^k(h))-\overline{\mu}_i^{h-1}(t_i^k(h),\pi)+\Delta_i^k\right)\\
&\quad\leq\left(h_i^k-1\right)\Delta_i^k+\sum_{h=2}^{h_i^k}\mathbb{1}\left(\xi^{\alpha}_{t_i^k(h)}\right)\left(\mu_\star(t_i^k(h))-\overline{\mu}_i^{h-1}(t_i^k(h),\pi)\right).
\end{aligned}$$

The last inequality is justified by $\mathbb{1}\left(\xi^{\alpha}_{t_i^k(h)}\right)\leq 1$. ■

Piecewise stationary rotting bandits.

Let $\{t_k\}_{\{k\leq \Upsilon_T\}}$ be the set of breakpoints with $t_0=0$ and $t_{\Upsilon_T}=T$. For all $t\in\{t_k+1,\dots,t_{k+1}\}$, $\mu_i(t)=\mu_i^k$. We denote $i_k^\star\in\arg\max_{i\in\mathscr{K}}\mu_i^k$ (one of) the best arm(s) in batch k, and $\mu_\star^k\triangleq\max_{i\in\mathscr{K}}\mu_i^k$, the corresponding best value. We also call $\Delta_{i,k}\triangleq\mu_\star^k-\mu_i^k$ the gap between arm i and optimal arm in batch k.

Lemma 5.2.8 For an arm i and a stationary batch k, we call $h_{i,\xi}^k\triangleq\max\left(h\leq h_i^k \text{ s.t. } \xi^\alpha_{t_i^k(h)} \text{ holds}\right)$ the last pull of arm i in batch k under the favorable events (possibly 0). If $h_{i,\xi}^k\geq 1$, the regret due to the second pulls on the favorable events is bounded by,

$$\sum_{t=t_k+1}^{t_{k+1}}\sum_{h=2}^{h_i^k}\mathbb{1}\left(t=t_i^k(h)\wedge\xi_t^\alpha\right)\left(\mu_\star(t)-\mu_i(t)\right)\leq\left(h_{i,\xi}^k-1\right)\Delta_{i,k}\leq C_\pi\sigma\sqrt{\left(h_{i,\xi}^k-1\right)\log T}.$$

Proof. We apply Lemma 5.2.7 on each stationary batch. Hence, $\Delta_i^k=0$ and we can write,

$$\sum_{t=t_k+1}^{t_{k+1}}\sum_{h=2}^{h_i^k}\mathbb{1}\left(t=t_i^k(h)\wedge\xi_t^\alpha\right)\left(\mu_\star(t)-\mu_i(t)\right)\leq\sum_{h=2}^{h_i^k}\mathbb{1}\left(\xi^\alpha_{t_i^k(h)}\right)\left(\mu_\star(t_i^k(h))-\overline{\mu}_i^{h-1}(t_i^k(h),\pi)\right).$$

We notice that $\mu_\star(t_i^k(h))=\mu_\star^k$. We call $h_{i,\xi}^k\triangleq\max\left(h\leq h_i^k \text{ s.t. } \xi^\alpha_{t_i^k(h)} \text{ holds}\right)$. Hence,

$$\begin{aligned}
\sum_{h=2}^{h_i^k}\mathbb{1}\left(\xi^\alpha_{t_i^k(h)}\right)\left(\mu_\star(t_i^k(h))-\overline{\mu}_i^{h-1}(t_i^k(h),\pi)\right)&=\sum_{h=2}^{h_{i,\xi}^k}\mathbb{1}\left(\xi^\alpha_{t_i^k(h)}\right)\left(\mu_\star^k-\overline{\mu}_i^{h-1}(t_i^k(h),\pi)\right)\\
&\leq\sum_{h=2}^{h_{i,\xi}^k}\mu_\star^k-\overline{\mu}_i^{h-1}(t_i^k(h),\pi)\\
&=\sum_{h=2}^{h_{i,\xi}^k}\mu_\star^k-\mu_i^k\\
&=\left(h_{i,\xi}^k-1\right)\Delta_{i,k}.
\end{aligned}$$

The first equality follows from $\forall h>h_{i,\xi}^k$, $\mathbb{1}\left(\xi^\alpha_{t_i^k(h)}\right)=0$ by definition of $h_{i,\xi}^k$. The first inequality follows by dropping $\mathbb{1}\left(\xi^\alpha_{t_i^k(h)}\right)\leq 1$. The second equality uses that the function is stationary in batch k : $\forall h\leq h_{i,\xi}^k,\overline{\mu}_i^{h-1}(t_i^k(h),\pi)=\mu_i^k$. The last equality follows by definition of $\Delta_{i,k}$ (which does not depend on the summand index h).

Then, we apply Lemma 5.2.2 at time $t_i^k\left(h_{i,\xi}^k\right)$. By definition of $h_{i,\xi}^k$, $\mathbb{1}\left(\xi^\alpha_{t_i^k(h_{i,\xi}^k)}\right)=1$.

$$\left(h_{i,\xi}^k-1\right)\Delta_{i,k}\leq\frac{C_\pi}{\sqrt{2\alpha}}\left(h_{i,\xi}^k-1\right)c(h_{i,\xi}^k-1,2T^{-\alpha})=C_\pi\sigma\sqrt{\left(h_{i,\xi}^k-1\right)\log T}.\qquad\blacksquare$$

Theorem 5.2.3 Let $\pi \in \{\pi_{\mathrm{F}}, \pi_{\mathrm{R}}\}$ tuned with $\alpha \geq 4$ or $\pi \in \{\pi_{\mathrm{EF}}, \pi_{\mathrm{ER}}\}$ tuned with $\alpha \geq 3$ and $m = 2$. For any piece-wise stationary bandit scenario with means $\{\mu_i(t)\}_{i,t}$ satisfying Assumptions 5.1.2 and 5.1.8 with $\Upsilon_T - 1$ change-points, π suffers an expected regret

$$\mathbb{E}[R_T(\pi)] \leq C_\pi \sigma \sqrt{\log T} \left(\sqrt{\Upsilon_T K T} + \Upsilon_T K \right) + 6KV.$$

Proof. We apply Lemma 5.2.8,

$$\sum_{k=0}^{\Upsilon_T - 1} \sum_{i \in \mathscr{K}_k} \sum_{t=t_k+1}^{t_{k+1}} \sum_{h=2}^{h_i^k} \mathbb{1}\left(t = t_i^k(h) \wedge \xi_t^\alpha\right) \left(\mu_\star(t) - \mu_i(t)\right) \leq \sum_{k=0}^{\Upsilon_T - 1} \sum_{i \in \mathscr{K}_k} C_\pi \sigma \sqrt{h_{i,\xi}^k \log T}.$$

We notice that $\sum_{k=0}^{\Upsilon_T - 1} \sum_{i \in \mathscr{K}_k} h_{i,\xi}^k \leq T$. Hence, thanks to Jensen's inequality,

$$\sum_{k=0}^{\Upsilon_T - 1} \sum_{i \in \mathscr{K}_k} C_\pi \sigma \sqrt{h_{i,\xi}^k \log T} \leq C_\pi \sigma \sqrt{\Upsilon_T K T \log T}.$$

We use Lemma 5.2.6 with the last equation and conclude,

$$\mathbb{E}[R_T(\pi)] \leq C_\pi \sigma \sqrt{\log T} \left(\sqrt{\Upsilon_T K T} + \Upsilon_T K \right) + 6KV.$$

■

Theorem 5.2.5 Let $\pi \in \{\pi_{\mathrm{F}}, \pi_{\mathrm{R}}\}$ tuned with $\alpha \geq 4$ or $\pi \in \{\pi_{\mathrm{EF}}, \pi_{\mathrm{ER}}\}$ tuned with $\alpha \geq 3$ and $m = 2$. For any piece-wise stationary bandit scenario with means $\{\mu_i(t)\}_{i,t}$ satisfying Assumptions 5.1.2 and 5.1.8 with $\Upsilon_T - 1$ change-points, π suffers an expected regret

$$\mathbb{E}[R_T(\pi)] \leq \sum_{k=0}^{\Upsilon_T - 1} \sum_{i \in \mathscr{K}} \frac{C_\pi^2 \sigma^2 \log T}{\Delta_{i,k}} + C_\pi \sigma \Upsilon_T K \sqrt{\log T} + 6KV.$$

Proof. Let $\mathscr{K}_k \triangleq \{i \in \mathscr{K} | \Delta_{i,k} > 0\}$, the set of sub-optimal arms in batch k. We apply Lemma 5.2.8 to bound the number of wrong pull (under the favorable events) of arm $i \in \mathscr{K}_k$ during batch k,

$$\Delta_{i,k} \left(h_{i,\xi}^k - 1\right) \leq C_\pi \sigma \sqrt{\left(h_{i,\xi}^k - 1\right) \log T} \implies h_{i,\xi}^k \leq 1 + \frac{C_\pi^2 \sigma^2 \log T}{\Delta_{i,k}^2}.$$

Then, we apply Lemma 5.2.8 again to bound the regret due to second pulls of any sub-optimal arm $i \notin \arg\max_{i\in\mathscr{K}} \mu_i^k$ in any batch k,

$$\begin{aligned} OP(i,k) &\triangleq \sum_{t=t_k+1}^{t_{k+1}} \sum_{h=2}^{h_i^k} \mathbb{1}\left(t = t_i^k(h) \wedge \xi_t^\alpha\right)(\mu_\star(t) - \mu_i(t)) \\ &\leq C_\pi \sigma \sqrt{\left(h_{i,\xi}^k - 1\right)\log T} \\ &\leq \frac{C_\pi^2 \sigma^2 \log T}{\Delta_{i,k}}. \end{aligned}$$

We apply Lemma 5.2.6 on the set of $\Upsilon_T - 1$ breakpoints and we conclude thanks to the precedent equation,

$$\begin{aligned} \mathbb{E}[R_T(\pi)] &\leq \mathbb{E}\left[\sum_{k=0}^{\Upsilon_T-1} \sum_{i\in\mathscr{K}_k} OP(i,k)\right] + C_\pi \sigma \Upsilon_T K \sqrt{\log T} + 6KV \\ &\leq \sum_{k=0}^{\Upsilon_T-1} \sum_{i\in\mathscr{K}} \frac{C_\pi^2 \sigma^2 \log T}{\Delta_{i,k}} + C_\pi \sigma \Upsilon_T K \sqrt{\log T} + 6KV\,. \end{aligned}$$

■

Variation budget rotting bandits.

Theorem 5.2.4 Let $\pi \in \{\pi_{\mathrm{F}}, \pi_{\mathrm{R}}\}$ tuned with $\alpha \geq 4$ or $\pi \in \{\pi_{\mathrm{EF}}, \pi_{\mathrm{ER}}\}$ tuned with $\alpha \geq 3$ and $m = 2$. For any variation budget bandit scenario with means $\{\mu_i(t)\}_{i,t}$ satisfying Assumptions 5.1.6 and 5.1.8 with variation budget V_T, π suffers an expected regret

$$\mathbb{E}[R_T(\pi)] \leq 4\left(C_\pi^2 \sigma^2 V_T K T^2 \log T\right)^{1/3} + 2\left(C_\pi \sigma V_T^2 K^2 T \sqrt{\log T}\right)^{1/3} + 6V_T K.$$

Proof. Let $\Upsilon \in \{1, \dots, T\}$ a number of evenly spaced batches that we will specify later. We define the length of these batches $\left\{\tau_k \triangleq \left\lceil \frac{T}{\Upsilon} \right\rceil \text{ if } k \leq T \bmod \Upsilon \text{ else } \left\lfloor \frac{T}{\Upsilon} \right\rfloor\right\}_{k\leq\Upsilon}$. Note that $\sum_{k=1}^{\Upsilon} \tau_k = T$. Let $t_k = \sum_{k'=0}^{k} \tau_{k'}$ the last round of each batch and $t_0 = 0$. On each of these batches, we apply Lemma 5.2.7 for the set of arms which have been pulled in this batch,

$$\begin{aligned} \sum_{k=0}^{\Upsilon_T-1} \sum_{i\in\mathscr{K}_k} \sum_{t=t_k+1}^{t_{k+1}} \sum_{h=2}^{h_i^k} \mathbb{1}\left(t = t_i^k(h) \wedge \xi_t^\alpha\right)\left(\mu_\star(t) - \mu_i(t)\right) &\leq \sum_{k=0}^{\Upsilon-1} \sum_{i\in\mathscr{K}_k} \left(h_i^k - 1\right)\Delta_i^k \\ &+ \sum_{k=0}^{\Upsilon-1} \sum_{i\in\mathscr{K}_k} \sum_{h=2}^{h_i^k} \mathbb{1}\left(\xi_{t_i^k(h)}^\alpha\right)\left(\mu_\star(t_i^k(h)) - \overline{\mu}_i^{h-1}(t_i^k(h), \pi)\right). \quad (5.19) \end{aligned}$$

The first sums can be handled using Assumption 5.1.6 and the evenly spaced property of τ_k,

$$\sum_{k=0}^{\Upsilon-1} \sum_{i\in\mathscr{K}} \left(h_i^k - 1\right)\Delta_i^k \leq \sum_{k=0}^{\Upsilon-1} \max_{j\in\mathscr{K}} \Delta_j^k \sum_{i\in\mathscr{K}} \left(h_i^k - 1\right) = \sum_{k=0}^{\Upsilon-1} \max_{j\in\mathscr{K}} \Delta_j^k (\tau_k - K) \leq \frac{T}{\Upsilon} \sum_{k=0}^{\Upsilon-1} \max_{j\in\mathscr{K}} \Delta_j^k.$$

(5.20)

The first inequality is justified by definition of the maximum. The second equality states that the total number of pulls in batch k is τ_k. The third inequality uses that $\tau_k - K \leq \left\lceil \frac{T}{\Upsilon} \right\rceil - K \leq \left\lceil \frac{T}{\Upsilon} \right\rceil - K \leq \frac{T}{\Upsilon}$. Now, we need to relate $\max_{j \in \mathscr{K}} \Delta_j^k$ and V_T,

$$\sum_{k=0}^{\Upsilon-1} \max_{j\in\mathscr{K}} \Delta_j^k = \sum_{k=0}^{\Upsilon-1} \max_{j\in\mathscr{K}} \sum_{t=t_k+1}^{t_{k+1}-1} \Delta_j(t,t+1) \leq \sum_{k=0}^{\Upsilon-1} \sum_{t=t_k+1}^{t_{k+1}-1} \max_{j\in\mathscr{K}} \Delta_j(t,t+1) \leq \sum_{t=1}^{T} \max_{j\in\mathscr{K}} \Delta_j(t,t+1) \leq V_T. \tag{5.21}$$

The first inequality is justified because the maximum of a sum is smaller than the sum of the maximums. In the second inequality, we add positive terms which are the maximum of the decay among the arms at the boundary between batches. The last inequality is justified by Assumption 5.1.6. Therefore, we can bound the first sums using Equation 5.20 and 5.21,

$$\sum_{k=0}^{\Upsilon-1} \sum_{i\in\mathscr{K}} \left(h_i^k - 1\right) \Delta_i^k \leq \frac{V_T T}{\Upsilon}. \tag{5.22}$$

The second sums can be bounded using Lemma 5.2.2 on the high probability event $\xi^{\alpha}_{t_i^k(h)}$ and Jensen's inequality,

$$\begin{aligned}
\sum_{k=0}^{\Upsilon-1} \sum_{i\in\mathscr{K}_k} \sum_{h=2}^{h_i^k} \mathbb{1}\left(\xi^{\alpha}_{t_i^k(h)}\right) \left(\mu_\star(t_i^k(h)) - \overline{\mu}_i^{h-1}(t_i^k(h),\pi)\right) &\leq \sum_{k=0}^{\Upsilon-1} \sum_{i\in\mathscr{K}_k} \sum_{h=2}^{h_i^k} \frac{C_\pi c\left(h-1, 2T^{-\alpha}\right)}{\sqrt{2\alpha}} \\
&= \sum_{k=0}^{\Upsilon-1} \sum_{i\in\mathscr{K}_k} \sum_{h=2}^{h_i^k} C_\pi \sigma \sqrt{\frac{\log T}{h-1}} \\
&\leq \sum_{k=0}^{\Upsilon-1} \sum_{i\in\mathscr{K}_k} 2C_\pi \sigma \sqrt{h_i^k \log T} \\
&\leq 2C_\pi \sigma \sqrt{\Upsilon K T \log T}.
\end{aligned} \tag{5.23}$$

We remark that the bound in Eq. 5.22 is decreasing with Υ and the bound in Eq. 5.23 is increasing with Υ. We will choose Υ in order to minimize the sum of these two bounds (which will be our leading term). Therefore, we set,

$$\Upsilon \triangleq \left\lceil \left(\frac{V_T^2 T}{C_\pi^2 \sigma^2 K \log T} \right)^{1/3} \right\rceil. \tag{5.24}$$

We have that $\Upsilon \leq T$ when $V_T \leq C_\pi \sigma T \sqrt{K \log T}$. Moreover, we will use that $\Upsilon \leq 2\left(\frac{V_T^2 T}{C_\pi^2 \sigma^2 K \log T}\right)^{1/3}$ which is true when $V_T \geq \sqrt{\frac{C_\pi^2 \sigma^2 K \log T}{8T}}$.

Finally, we use Lemma 5.2.6 where we replace the inner sums thanks to Equations 5.19, 5.22 and 5.23. Then, we plug Υ set in 5.24 and conclude,

$$\begin{aligned}\mathbb{E}\left[R_T\left(\pi\right)\right] &\leq \frac{V_T T}{\Upsilon} + 2C_\pi\sigma\sqrt{\Upsilon KT\log T} + C_\pi\sigma\Upsilon K\sqrt{\log T} + 6V_T K \\ &\leq 4\left(C_\pi^2\sigma^2 V_T KT^2\log T\right)^{1/3} + 2\left(C_\pi\sigma V_T^2K^2T\sqrt{\log T}\right)^{1/3} + 6V_T K.\end{aligned}$$

When $V_T \leq \sqrt{\frac{C_\pi^2\sigma^2 K\log T}{8T}}$, the regret of any policy can be bounded ,

$$\begin{aligned}\mathbb{E}\left[R_T(\pi)\right] &\leq TV_T = V_T^{1/3}T^{2/3}V_T^{2/3}T^{1/3} \\ &\leq V_T^{1/3}T^{2/3}\left(\frac{C_\pi^2\sigma^2 K\log T}{8T}\right)^{1/3}T^{1/3} \\ &= \frac{1}{2}\left(C_\pi^2\sigma^2 V_T KT^2\log T\right)^{1/3} \\ &\leq 4\left(C_\pi^2\sigma^2 V_T KT^2\log T\right)^{1/3}.\end{aligned}$$

For completion, we also consider $V_T \geq C_\pi\sigma T\sqrt{K\log T}$. Yet, notice that in that case the leading term is $\mathscr{O}\left(KV_T\right)$. We start back from Lemma 5.2.6,

$$\begin{aligned}\mathbb{E}\left[R_T(\pi)\right] \leq \mathbb{E}&\left[\sum_{k=0}^{\Upsilon-1}\sum_{i\in\mathscr{K}_k}\sum_{t=t_k+1}^{t_{k+1}}\sum_{h=2}^{h_i^k}\mathbb{1}\left(t = t_i^k(h)\wedge\xi_t^\alpha\right)\left(\mu_\star(t)-\mu_i(t)\right)\right] \\ &+ C_\pi\sigma\Upsilon K\sqrt{\log T} + 6KV_T.\end{aligned}$$

In fact, this result can be slightly improved at no cost,

$$\begin{aligned}\mathbb{E}\left[R_T(\pi)\right] \leq \mathbb{E}&\left[\sum_{k=0}^{\Upsilon-1}\sum_{i\in\mathscr{K}_k}\sum_{t=t_k+1}^{t_{k+1}}\sum_{h=2}^{h_i^k}\mathbb{1}\left(t = t_i^k(h)\wedge\xi_t^\alpha\right)\left(\mu_\star(t)-\mu_i(t)\right)\right] \\ &+ C_\pi\sigma\min\left(\Upsilon K, T\right)\sqrt{\log T} + 6KV_T,\end{aligned}$$

because there are at most $\min\left(\Upsilon K, T\right)$ first pulls (see the proof of Lemma 5.2.6). Now, we choose $\Upsilon = T$. Hence, there is no second pulls and we have,

$$\mathbb{E}\left[R_T(\pi)\right] \leq C_\pi\sigma T\sqrt{\log T} + 6KV_T,$$

Now, we use that $C_\pi\sigma T\sqrt{\log T} \leq \frac{V_T}{\sqrt{K}} \leq KV_T$,

$$\begin{aligned}\mathbb{E}\left[R_T(\pi)\right] &\leq \left(C_\pi\sigma T\sqrt{\log T}\right)^{2/3}\left(C_\pi\sigma T\sqrt{\log T}\right)^{1/3} + 6KV_T \\ &\leq \left(C_\pi^2\sigma^2 V_T KT^2\log T\right)^{1/3} + 6KV_T \\ &\leq 4\left(C_\pi^2\sigma^2 V_T KT^2\log T\right)^{1/3} + 2\left(C_\pi\sigma V_T^2K^2T\sqrt{\log T}\right)^{1/3} + 6KV_T.\end{aligned}$$

■

5.3 Real-word data experiment on Yahoo! Front Page

R6A - Yahoo! Front page today module user click log dataset. This dataset was used for the Exploration and Exploitation Challenge[2] at ICML 2012 and inspired new algorithms. Among them, we mention the work of Tracà and Rudin (2015) who noticed the non-stationary trend and took advantage of it. Since then the dataset continues to be a benchmark[3] for non-stationary bandits (F. Liu et al. 2018; Cao et al. 2019). It contains the history of clicks on news articles of 45 million users in the first ten days of May 2009. We use three features in this dataset: *timestamp* (rounded every 5 minutes), *article_id*, and *click*.

A real decaying scenario. Every day, between 6 pm and 6 am EST (12 hours), we notice a decreasing trend in click probability. It suggests that people in the US read less and less news during the evening and night. For each day, we keep all the articles that have been recommended at every timestamp during the 12 hours. For these articles, we use a rolling average window of 30000 in order to estimate the probability of click for each article at each timestamp [4]. We use the real total traffic for each timestamp. We highlight that *we do not enforce any of our assumptions* to create reward functions to be aligned with our setup. In particular, we do not enforce them to be piecewise constant nor to be decreasing. At each round, the learner receives 10 reward samples in order to reduce the cost of computation.

Algorithms and Parameters. We include two versions of `FEWA` and `RAW-UCB`: with the theoretical tuning $\alpha = 4$; and with the empirical tuning $\alpha_{\mathrm{R}} = 1.4$ and $\alpha_{\mathrm{F}} = 0.06$. These two values were selected on the rested benchmark (c.f. Section 4.4). This benchmark has 30 different problems (for different L) but the best tuning of α is the same for all the considered problems. We replace `RAW-UCB` and `FEWA` with their efficient versions because of the longer horizon.

We also include `Exp3.S` (Auer et al. 2003) and `GLR-UCB` (Besson and Kaufmann 2019). For `Exp3.S`, we use the theoretical tuning which requires the knowledge of T and V_T. `GLR-UCB` has two parameters: a confidence level δ for its change-point detector and an active exploration rate α. We set α to zero. Indeed, the active exploration of change-detection algorithms is only useful in the increasing case (as argued by Cao et al. (2019)). We tune δ by its theoretical value, which requires the knowledge of T. Last, we only restart the history of the changed arm as our setup does not assume that all the rewards change simultaneously. For a fair comparison, we only use the subgaussian version of the algorithm. Indeed, KL-UCB indexes are expensive to compute. Instead, for all the confidence bound algorithms, we rather tune $\sigma^2 = 1$ in the rested benchmark and $\sigma^2 = 0.29$ in the restless benchmark (the variance of a binomial $\mathscr{B}(10, 0.03)$).

[2] http://explochallenge.inria.fr/

[3] As it allows for offline evaluations as the actions were samples uniformly.

[4] For each timestamp, we average the values given by rolling average. These values are close to each other because the number of click opportunities per article in the same timestamp is small compared to 30000.

We do not include SWA (Levine et al. 2017) which was shown to be less consistent than FEWA and RAW-UCB on rested rotting bandits. We do not include SW-UCB and D-UCB as they were shown to be unable to learn in the rested setting (Levine et al. 2017; Seznec et al. 2019). We also do not include CUSUM-UCB (F. Liu et al. 2018) and M-UCB (Cao et al. 2019), as 1) they were shown to under-perform against GLR-UCB (Besson and Kaufmann 2019); and 2) their change-detector is harder to tune.

Note that our goal is to compare algorithms with the same tuning in the rested and restless benchmark.

Results. We display the results for eight different days in Figure 5.2.We will comment day 2 and day 7. On day 2, there are several switches of optimal arms with many near-optimal ones: tracking the best arm is a "hard" problem. On day 7, one arm consistently dominates the others by far. Hence, it is an "easy" case where good algorithms should have a logarithmic regret rate. We also display the running time of each algorithm in Table 5.1.

Day	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
EFF-RAW-UCB ($\alpha=1.4, m=1.1$)	67	66	90	86	91	74	88	64	48
EFF-RAW-UCB ($\alpha=1.4, m=2$)	35	33	43	47	46	41	44	34	45
EFF-RAW-UCB ($\alpha=4, m=1.1$)	65	65	90	88	91	74	89	63	48
EFF-FEWA ($\alpha=0.06$)	143	175	223	159	183	115	193	116	165
EFF-FEWA ($\alpha=4$)	337	308	391	473	487	380	428	341	388
Exp3.S	56	53	67	77	75	69	71	55	78
GLR-UCB	560	613	683	2421	707	1529	957	971	4017

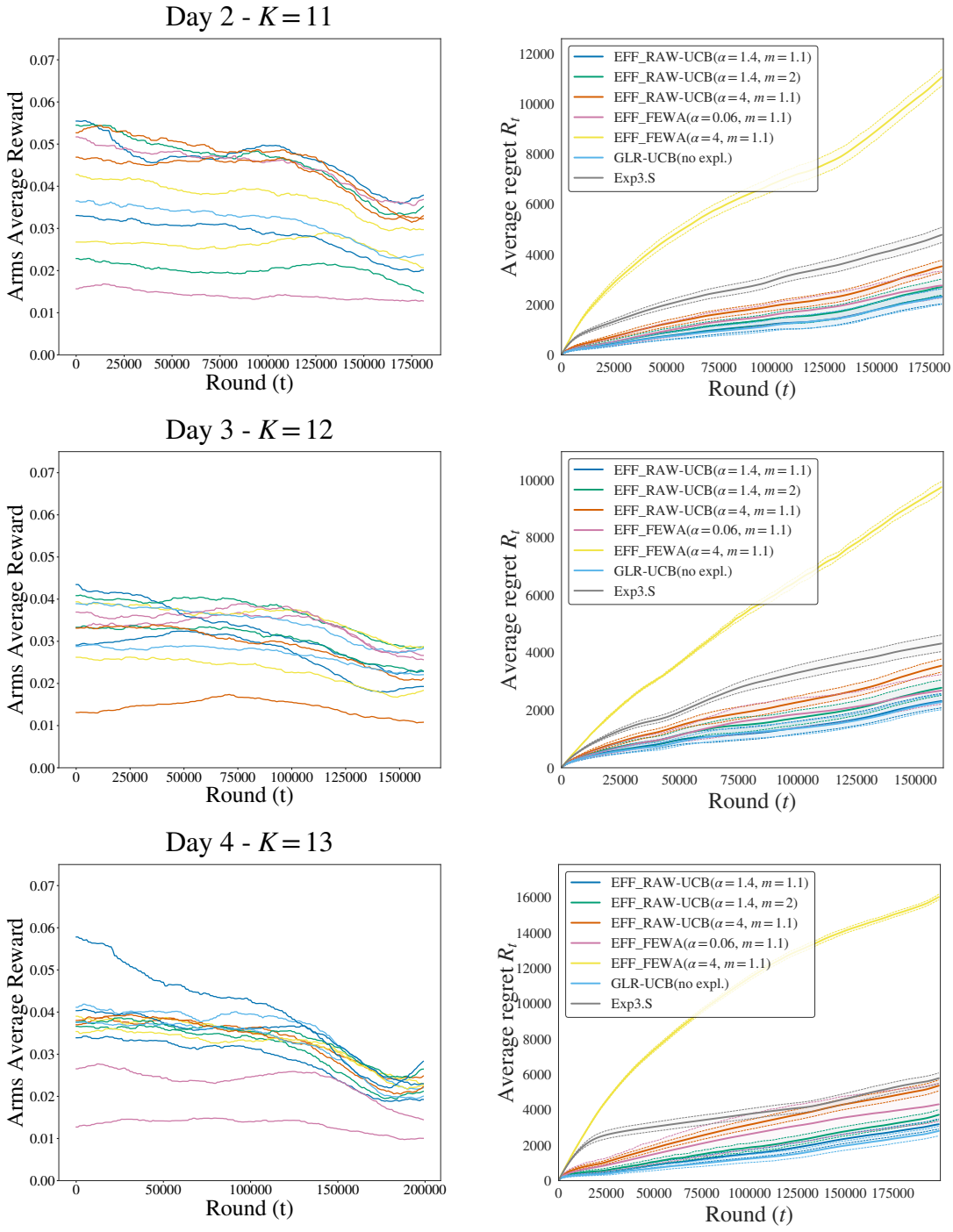
Table 5.1: Average computational time in seconds for each algorithm in each experiment.

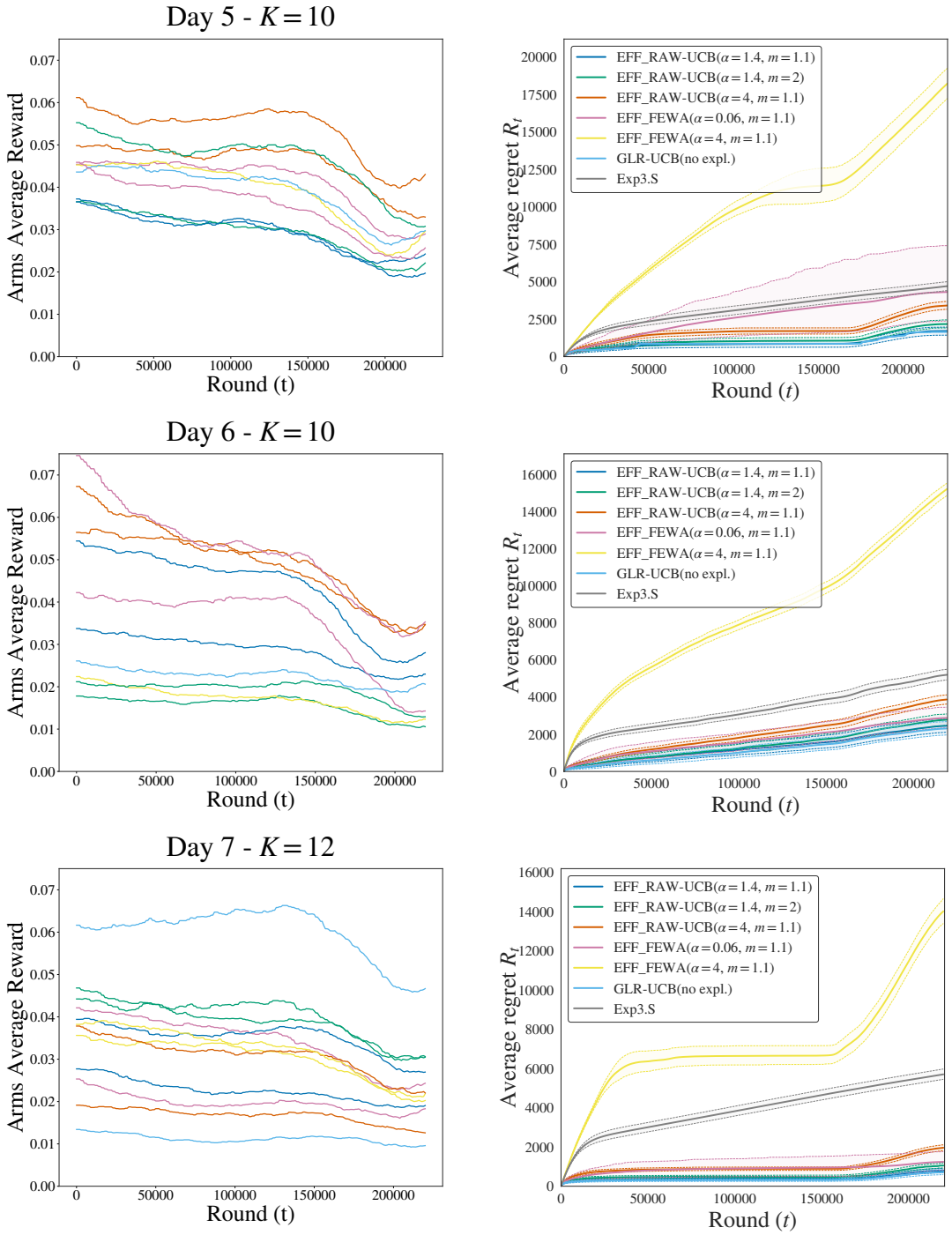
RAW-UCB **vs** FEWA**.** The two algorithms compute the same statistics and share most of their analysis. Yet, RAW-UCB consistently outperforms FEWA as it was the case on the rested benchmark. The difference between the two is even more significant in the restless case. Its theoretical tuning $\alpha = 4$ gets reasonable results, while theoretical FEWA is impractical. Finally, its empirical tuning $\alpha_{\mathrm{R}} = 1.4$ is similar to the asymptotic optimal tuning of UCB and shows good performance on both rested and restless problems. By contrast, FEWA with $\alpha_{\mathrm{F}} = 0.06$ shows worse performance with larger deviation on the restless benchmark.

RAW-UCB **vs** Exp3.S**.** Exp3.S has good performances on the restless benchmark, on which it has theoretical guarantees. Yet, it is consistently outperformed by RAW-UCB when we tune the confidence bounds. It is particularly true in easy instances, e.g. on day 7. Indeed, in these cases, we expect a logarithmic regret rate for RAW-UCB.

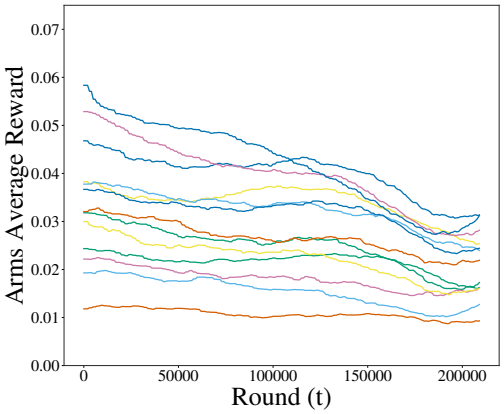
RAW-UCB **vs** GLR-UCB **(no active exploration).** On the restless benchmark, GLR-UCB shows similar results than RAW-UCB. Yet, we highlight that 1) GLR-UCB needs the knowledge

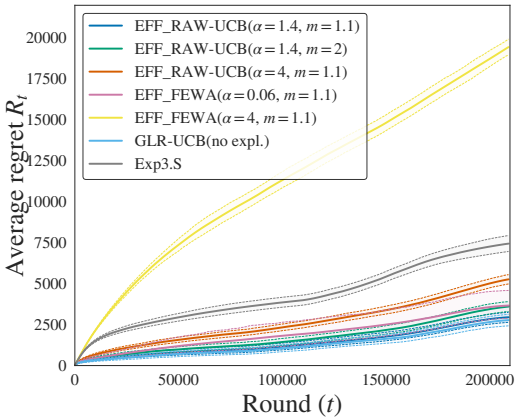
Figure 5.2: **Left:** reward functions from the Yahoo! dataset
Right: average regret of policies over 500 runs



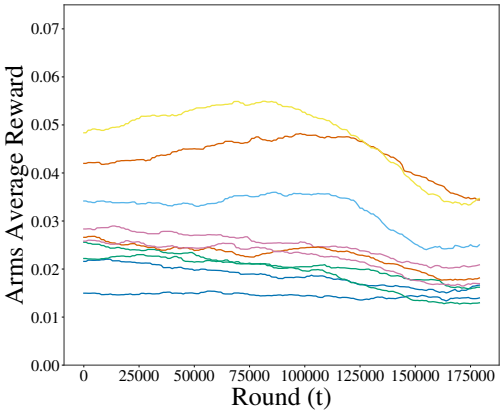

Day 5 - $K = 10$
Arms Average Reward
Round (t)
Average regret R_t
Round (t)
Day 6 - $K = 10$
Arms Average Reward
Round (t)
Average regret R_t
Round (t)
Day 7 - $K = 12$
Arms Average Reward
Round (t)
Average regret R_t
Round (t)
EFF_RAW-UCB($\alpha = 1.4$, $m = 1.1$)
EFF_RAW-UCB($\alpha = 1.4$, $m = 2$)
EFF_RAW-UCB($\alpha = 4$, $m = 1.1$)
EFF_FEWA($\alpha = 0.06$, $m = 1.1$)
EFF_FEWA($\alpha = 4$, $m = 1.1$)
GLR-UCB(no expl.)
Exp3.S

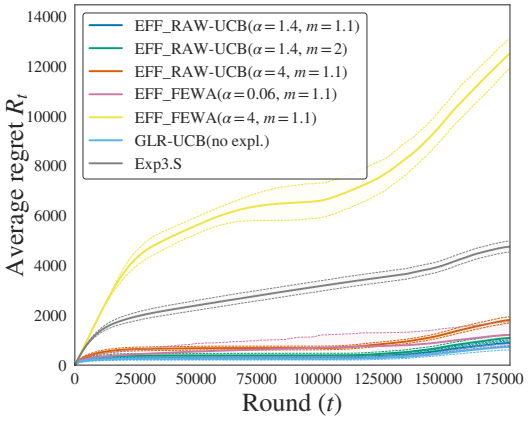
Day 8 - $K = 13$



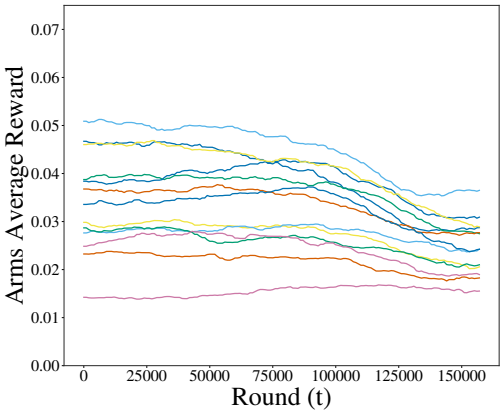


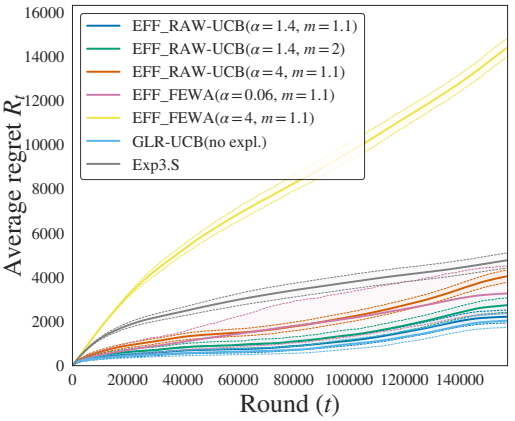
Day 9 - $K = 10$





Day 10 - $K = 13$





of the horizon to tune its change-detector; 2) we use an efficient version of RAW-UCB which runs ~ 10 times faster than GLR-UCB. In fact, the two algorithms are similar: they are UCB index policies, they recover logarithmic rate on easy restless rotting bandits problems and hence they would both suffer near-linear worst-case regret rate in the general restless setting (when active exploration is turned off for GLR-UCB). The main difference is that RAW-UCB scans its history to select its rotting UCB's window, while GLR-UCB scans its history to detect significant changes and restart.

5.4 Restless and rested rotting bandits

5.4.1 The general case

Assumption 5.4.1 For each arm i, any number of pulls n, and time t, the functions $\mu_i(t,\cdot)$ and $\mu_i(\cdot,n)$ are non-increasing.

In Section 5.2, we highlight that the main guarantee of our algorithms - Lemma 5.2.2 - holds in the general case of Assumption 5.4.1. Is it enough to show that our algorithms are near-optimal in this extended setup?

Like in Chapter 4 and 5, we define the regret with respect to the best oracle.

$$R_T(\pi,\mu) \triangleq \arg\max_{\pi_T^\star \in \Pi_O} J_T(\pi_T^\star,\mu) - J_T(\pi).$$

Like in the linear rested rotting bandits (Section 4.7), we can show that not only the greedy oracle suffers linear regret but no learning policy can get a sublinear regret rate in the worst-case.

Proposition 5.4.2 In the no noise setting ($\sigma = 0$), there exists a rotting 2-arms bandits problem (satisfying Assumption 5.4.1) with reward value in $[0,1]$, with one rested arm and one restless arm, and with at most one change-point before T each, such that the greedy oracle strategy π_O suffers a regret

$$R_T(\pi_O) \geq \left\lfloor \frac{T}{4} \right\rfloor.$$

Moreover, for any learning strategy π_S, there exists a rotting 2-arms bandits problem (satisfying Assumption 5.4.1) with reward value in $[0,1]$, with one rested arm and one restless arm, and with at most one change-point before T each, such that

$$R_T(\pi_S) \geq \left\lfloor \frac{T}{8} \right\rfloor.$$

Notice that the two reward functions of the constructed difficult problems are simple: either rested or restless, bounded, and with at most one break-point. If we consider a 2-arm setup with one rested arm and one restless arm, a good strategy may be to select the restless arm

even when its current value is the worst. Indeed, this value is only available now, while the good value of the rested arm will still be available in the future. Whether the restless rewards are interesting to the learner depends on the future behavior of the (currently best) rested arm. On the first hand, if it decays below the current value of the restless arm before T pulls, then the learner should profit from the restless reward available right now. On the other hand, if the rested arm stays optimal until the end of the game then the learner should ignore the restless arm and follows the greedy oracle strategy. However, the learner does not know in advance if (and how much) an arm will decay and any anticipation she makes will turn to be bad in the worst case. We formalize these ideas in the proof at the end of the section.

5.4.2 Rested rotting bandits with a restless envelope

Assumption 5.4.3 We consider the following reward functions,

$$\mu_i(t,n) = P(t)f_i(n) + S(t),$$

where $P : \mathbb{N}^* \to \mathbb{R}_+$, $\{f_i : \mathbb{N} \to \mathbb{R}\}_{i\in\mathscr{K}}$ and $S : \mathbb{N}^* \to \mathbb{R}$ are non-increasing functions.

Notice that all the arms have the same product P and sum S functions, the only difference is the rested evolution f_i. That is why we call this setup the rested rotting bandits with a restless envelope.

With this assumption, we can show that the greedy oracle is optimal.

Proposition 5.4.4 For any reward functions $\{\mu_i\}_{i\in\mathscr{K}}$ verifying Assumption 5.4.3 and any horizon T, $\pi_O \in \arg\max_{\pi\in\Pi_O} J_T(\pi)$.

We leave as an open problem to analyze the aforementioned algorithms in this setup. A first step would be to characterize the performance of the greedy bandit policy in the absence of noise (as we did for the rested problem, see Subsection 4.1.2). We may not recover the $\mathscr{O}(K)$ bound as in the rested setup. Indeed, the adversary can use the variation of P and S to trick the greedy bandit policy several times for each arm. Moreover, the order of the pull do matter in this problem: the cumulative reward is not a function of $\{N_{i,T}\}_{i\in\mathscr{K}}$ anymore.

5.4.3 Proofs

Proof of Proposition 5.4.2. Let μ^0 and μ^1, two decreasing 2-arms bandits problems such that:

$$\mu_1^0(t,n) = \mu_1(n) = 1 \text{ if } n < \frac{T}{2} \text{ else } 0,$$
$$\mu_1^1(t,n) = 1,$$
$$\mu_2^0(t,n) = \mu_2^1(t,n) = \mu_2(t) = 1/2 \text{ if } t < \frac{T}{2} \text{ else } 0.$$

Problem μ^1 only evolves according to time. Hence, the oracle greedy policy π_O is optimal for this problem and collects

$$J_T\left(\pi_O, \mu^1\right) = T. \tag{5.25}$$

On μ^0, π_O selects arm 1 during $\left\lfloor \frac{T}{2} \right\rfloor$ rounds and then both arms yield 0 reward. Thus, π_O collects

$$J_T\left(\pi_O, \mu^0\right) = \left\lfloor \frac{T}{2} \right\rfloor.$$

However, let π_0 the policy which selects arm 2 for $\left\lfloor \frac{T}{2} \right\rfloor$ rounds and arm 1 afterwards. Thus, π_0 collects

$$J_T\left(\pi_0, \mu^0\right) = (3/2)\left\lfloor \frac{T}{2} \right\rfloor. \tag{5.26}$$

Hence, we conclude the first part of our proposition,

$$R_T\left(\pi_O, \mu^0\right) = J_T\left(\pi_T^\star, \mu^0\right) - J_T\left(\pi_O, \mu^0\right) \geq J_T\left(\pi_0, \mu^0\right) - J_T\left(\pi_O, \mu^0\right) \geq \left\lfloor \frac{T}{4} \right\rfloor.$$

Now, we consider any learning policy π_S and we call $\mathbb{E}_j\left[N_{i,t}(\pi_S)\right]$ the (expected, if the policy is random) number of pulls of arm i at any round t by π_S on problem j. Note that the leaner will receive the same rewards for both problems until at least $\left\lfloor \frac{T}{2} \right\rfloor$. Therefore, we have that

$$\forall t \leq \left\lfloor \frac{T}{2} \right\rfloor, \pi\left(\mathscr{H}_t\left(\mu^0\right)\right) = \pi\left(\mathscr{H}_t\left(\mu^1\right)\right) \implies \mathbb{E}_0\left[N_{2,\left\lfloor \frac{T}{2} \right\rfloor}(\pi_S)\right] = \mathbb{E}_1\left[N_{2,\left\lfloor \frac{T}{2} \right\rfloor}(\pi_S)\right] \triangleq n_2.$$

On problem μ^1, π_S collects a reward of at most,

$$J_T\left(\pi_S, \mu^1\right) = \mathbb{E}_1[N_{1,T}(\pi_S)] + \frac{n_2}{2} = T - \mathbb{E}_1[N_{2,T}(\pi_S)] + \frac{n_2}{2} \leq T - \frac{n_2}{2}, \tag{5.27}$$

because $n_2 = \mathbb{E}_1\left[N_{2,\left\lfloor \frac{T}{2} \right\rfloor}(\pi_S)\right] \leq \mathbb{E}_1[N_{2,T}(\pi_S)]$. Using Equations 5.25 and 5.27, we can lower bound the regret of π_S,

$$R_T\left(\pi_S, \mu^1\right) = J_T\left(\pi_O, \mu^1\right) - J_T\left(\pi_S, \mu^1\right) \geq \frac{n_2}{2}.$$

On problem μ^0, π_S collects a reward of at most,

$$J_T\left(\pi_S, \mu^0\right) = \min\left(\mathbb{E}_1[N_{1,T}(\pi_S)], \left\lfloor \frac{T}{2} \right\rfloor\right) + \frac{n_2}{2} \leq \left\lfloor \frac{T}{2} \right\rfloor + \frac{n_2}{2}. \tag{5.28}$$

Using Equations 5.26 and 5.28, we can lower bound the regret of π_S,

$$R_T\left(\pi_S, \mu^0\right) = J_T\left(\pi_O, \mu^0\right) - J_T\left(\pi_S, \mu^0\right) \geq \frac{\lfloor T/2 \rfloor - n_2}{2}.$$

Hence, the worst case regret on the two setups is bounded by

$$R_T(\pi_S) \geq \max\left(\frac{n_2}{2}, \frac{\lfloor \frac{T}{2} \rfloor - n_2}{2}\right) \geq \left\lfloor \frac{T}{8} \right\rfloor.$$

■

Proof of Proposition 5.4.4. At any round t, we have,

$$\pi_{\mathrm{O}}(t) \in \underset{i \in \mathscr{K}}{\arg\max} \left(P(t) f_i(N_{i,t}) + S(t)\right) = \underset{i \in \mathscr{K}}{\arg\max} f_i\left(N_{i,t}\right).$$

Therefore, at round t, collects the t largest values of $\{f_i(n)\}_{i \in \mathscr{K}, n \leq T}$, *i.e.*

$$\forall i \in \mathscr{K},\ \forall n_i \geq N_{i,t},\ \mu_{\pi_{\mathrm{O}}(t)}\left(N_{\pi_{\mathrm{O}}(t),t}\right) \geq \mu_i\left(N_{i,t}\right) \geq \mu_i(n_i).$$

The first inequality is due to the selection rule of the policy; the second is due to the decreasing reward functions.

A direct consequence is that, at the round t, π_{O} selects the t-th largest value of $\{f_i(n)\}_{i \in \mathscr{K}, n \leq T}$. Hence, at the round T, it has selected the T largest value in the decreasing order. Since $P(t)$ is non-increasing and positive, an other policy which selects smaller values of $\{f_i(n)\}_{i \in \mathscr{K}, n \leq T}$, or the same values but in an other order, have a smaller or equal cumulative reward than π_{O}. ■

III Beyond rotting bandits

6 Master topics as soon as possible 163

6.1 Beyond rotting bandits: some motivations
6.2 Setup
6.3 Optimal Oracle: Focus on the largest under the threshold
6.4 What does random progression mean?
6.5 Learning Perspectives
6.6 Practical considerations for ITS applications

6. Master topics as soon as possible

Turn your head left and blink twice. You'll see a bandit in a POMDP.

6.1 Beyond rotting bandits: some motivations

Our motivation for studying the rested rotting bandits was the ability to target the least known topic. This educational strategy can be interesting before an exam, when we assume that all the topics should be at least understandable by the student. However, during the curriculum, targeting the most difficult subject can demotivate the student and could result in no learning (the wheel-spinning effect, Beck and Gong 2013).

`RAW-UCB` (or `FEWA`) keeps switching between topics either because the confidence intervals are reduced through the pulls or because the student gains proficiency on the topics. On Afterclasse, a student before the exam needs to study ~ 10 chapters divided into ~ 3 topics. It makes up to 30 potential arms. If the student answers two hundreds of exercises (which is a lot compared to the average student), `RAW-UCB` is barely different than round-robin.

Moreover, rotting bandits do not take into account the difficulty levels. If we consider that each difficulty level is a different arm, then `RAW-UCB` will focus on difficult questions before focusing on the easiest questions. Arguably, this is not a good educational strategy. If we consider that different difficulty levels are in the same arm, then we should select the difficulty uniformly at random to not bias the averages that `RAW-UCB` constructs. Another possibility is to choose the difficulty with a subroutine and correct the bias with specific computations (e.g. importance sampling).

Chapter header: *Fiat 500 - Reveal your darkside*, ©Marc da Cunha Lopes, Agence Leo Burnett.

For all these reasons, RAW-UCB is hard to test on students in a relevant educational scenario. However, RAW-UCB does not have only disadvantages: it is quite interesting to take educational decisions based on pessimistic estimates of the student's proficiencies. Indeed, if we stop learning a topic because its estimate is high enough, it is important to be sure that this estimate is not high just by chance.

In this chapter, we describe a setup where the goal is to validate topics as soon as possible. We show that, under relevant assumptions, the best thing to do is to first focus on the simplest topic and then switch to the more difficult ones promptly. In an online setting, we don't know which topic is the simplest, so we design an exploration strategy that outputs a topic among the easiest and then we focus on this arm until we are sure the topic is validated. This algorithm makes good use of the aforementioned pessimistic estimates to both select a simple topic and to be sure that the topic is validated at the end of the session. Finally, we discuss design improvements to switch our theoretical algorithm in a practical Intelligent Tutoring System (ITS).

6.2 Setup

We model the student-ITS interaction as a formal Partially Observable Markov Decision Process (POMDP).

State, actions and feedback The agent faces a set of K tasks. Each task i has a state $\mu_{i,t} \in \mathbb{R}$ at the round t with initial value $\mu_{i,1}$. At each round t, the agent selects a task i_t to allocate resource (e.g. time). He receives a noisy observation of its current state,

$$o_t \triangleq \mu_{i,t} + \varepsilon_t,$$

where $\{\varepsilon_t\}_{t \leq T}$ is an independent sequence of σ-subgaussian variables, *i.e.*

$$\mathbb{E}[\varepsilon_t | \mathscr{H}_t] = 0 \text{ and } \forall \lambda \in \mathbb{R}, \mathbb{E}\left[e^{\lambda \varepsilon_t}\right] \leq e^{\frac{\sigma \lambda^2}{2}},$$

with $\mathscr{H}_t \triangleq \{\{i_s, o_s\}\}_{s<t}$, the history of the agent at the beginning of the round t. We call $\boldsymbol{\mu_t} \triangleq \{\mu_{i,t}\}_{i \in \mathscr{K}}$.

In the context of Intelligent Tutoring Systems, a task is to learn a topic. The state is the average level of the student on that topic. The action is to give a student a question related to that topic, and the observation is the grade associated with the answer to that question.

Transition. Between consecutive rounds, the state $\boldsymbol{\mu_t}$ is randomly modified following transition probabilities which depend on the selected arm and the current state. It contrasts with the rotting bandits we studied so far where the evolution was deterministic. We

discuss the meaning of this random evolution concerning our ITS application at the end of the section. We use two assumptions that we already studied in the rested rotting bandits framework (Chapter 4): the rested and monotone evolution of the arms' states.

Assumption 6.2.1 The transitions are rested, which means that selecting a task only modifies the state of this particular task. Hence, we have that,

$$\mu_{i,t} = \mu_i(N_{i,t-1}),$$

with $\{\mu_i(n)\}_{n\in\mathbb{N}}$ a Markov Chain with transition operator $\mathscr{T}_i$ and $N_{i,t} \triangleq \sum_{s=1}^{t} \mathbb{I}\{i_s = i\}$.

Assumption 6.2.2 The state of a task can only increase with pulls. Hence, the transition operators $\{\mathscr{T}_i\}_{i\in\mathscr{K}}$ are triangular inferior.

Rotting bandits were considering non-increasing sequences of rewards while we consider now non-decreasing sequences of states. Yet, it can correspond to the same situation where the reward is the opposite of the state. This is not only a formal remark. It is indeed the case for Intelligent Tutoring System motivation: the student is progressing so the associated need to learn the topic is decreasing.

We now make two Assumptions on the transition operators $\{\mathscr{T}_i\}_{i\in\mathscr{K}}$.

Assumption 6.2.3 The transition operator is the same for all the tasks,

$$\forall i \in \mathscr{K}, \mathscr{T}_i = \mathscr{T}.$$

For a random variable X with density p, we call $F_p(z) \triangleq \mathbb{P}[X \leq z]$ the cumulative distribution function. We define the first-order stochastic dominance of a variable X (drawn with probability p_x) over a random variable Y (drawn with probability p_y),

$$X \succeq Y \iff p_x \succeq p_y \iff \forall z \in \mathbb{R},\ F_{p_x}(z) \leq F_{p_y}(z). \tag{6.1}$$

Assumption 6.2.4 The transition operator $\mathscr{T}$ is stochastically monotone, *i.e.* with $(\mathscr{T}\delta_x)(y) \triangleq \mathscr{T}(x,y)$,

$$\forall (x_1,x_2) \in \mathbb{R}^2, x_1 \leq x_2 \implies \mathscr{T}\delta_{x_1} \preceq \mathscr{T}\delta_{x_2}.$$

In other words, the larger the starting point, the larger the probability to reach any threshold at the next step. This assumption was first studied by Daley (1968). We restate their two main results,

Lemma 6.2.5 — Daley (1968). Assumption 6.2.4 is equivalent with

$$\forall (p,q), p \preceq q \implies \mathscr{T}p \preceq \mathscr{T}q.$$

Corollary 6.2.6 — Daley (1968). The larger the starting state, the larger the probability of reaching any threshold after a given number of steps $n \in \mathbb{N}$, *i.e.*,

$$\forall (x_1, x_2) \in \mathbb{R}^2, x_1 \leq x_2 \implies \mathscr{T}^n \delta_{x_1} \preceq \mathscr{T}^n \delta_{x_2}.$$

For Intelligent Tutoring Systems, Assumption 6.2.3 means that the student progresses in the same way for all the topics. It may not be true if the topics are completely different subjects (e.g. maths and history) but if it is two topics in the same chapter (e.g. Pythagore and Thales theorems), it is likely that the progression of the student will be similar. Assumption 6.2.4 assumes that the progression on the different topics is monotonic. If a student is quite good on a first topic and quite bad on another one, it is unlikely (yet possible) that after a single question on each topic, s/he masters the second one and not the first one.

Objective We consider a task as being completed when $\mu_{i,t} \geq \mu$ for a given threshold μ. We will consider two related objectives. First, the *simple* objective is to maximize the number of completed tasks after the horizon T,

$$r_T(\pi) \triangleq \sum_{i \in \mathscr{K}} \mathbb{1}\left[\mu_{i,T+1} \geq \mu\right].$$

With respect to this objective, we can define a reward for our POMDP associated to the transition from state x to y,

$$\rho(x, y) \triangleq \mathbb{1}\left[x < \mu \wedge y \geq \mu\right].$$

$r_T(\pi)$ is the sum of the reward,

$$r_T(\pi) = \sum_{t=1}^{T} \rho(\mu_{\pi(t),t}, \mu_{\pi(t),t+1}) + \sum_{i \in \mathscr{K}} \mathbb{1}\left[\mu_{i,1} \geq \mu\right]. \tag{6.2}$$

Notice that the second sum is the sum of the arms which are initially above the threshold: it does not depend on the agent's action. Second, the *cumulative* objective is to optimize,

$$J_T(\pi) = \sum_{t=1}^{T} r_t(\pi).$$

The reward at each round is $r_t(\pi)$. Thus, a validated task at the round t yields cumulatively a reward equal to the remaining number of rounds $T - t$. For Intelligent Tutoring Systems, a completed topic may trigger new teaching actions such as starting new topics. The sooner we can trigger these actions, the better it is. It suggests that it is not only important to master topics at the end of the studying session, but also to master them as fast as possible.

R For both objectives, the reward at any round t is a function of the state (current or previous) which is itself partially observable. One can hardly reconstruct the reward

at the round t from the unique observation sample at this same round. Indeed, if a student answers correctly to a question, we may have chosen a topic which is already mastered by the student (no reward for the action) or we may have chosen a topic which will be mastered very soon (good reward for the action). It contrasts with the cumulative reward in the multi-armed bandits paradigm, where the relationship between observation and reward is more straightforward.

Our objectives r_T and J_T are random quantities. In the following, we will aim at maximizing their expected values, where the expectation is on the random evolution of the Markov Chains, the random noise in the observation, and the potential randomization of the agent's strategy. In particular, we highlighted that Assumption 6.2.4 is a "smooth in probability" assumption. Hence, even if abrupt progression is possible, these paths will weigh little in the expected regret compared to smooth ones.

We give a consequence of Assumption 6.2.4 in terms of the number of rounds to reach the threshold μ,

Definition 6.2.1 Let $(\mu_i(n))_{n\in\mathbb{N}}$ a Markov chain with transition probabilities $\mathscr{T}$. We define the stopping time,

$$\tau_i \triangleq \min\{\tau \in \mathbb{N} \,|\, \mu_i(\tau) \geq \mu\},$$

the number of pulls to reach the threshold μ. We also define,

$$\tau_{i,t} \triangleq \max(\tau_i - N_{i,t-1}, 0).$$

the remaining number of pulls at a round t after $N_{i,t}$ pulls. We notice that $\tau_i = \tau_{i,0}$.

Let $(X_n)_{n\in\mathbb{N}}$ a Markov Chain with $X_0 = x$ a transition probabilities $\mathscr{T}$. We call,

$$\tau(x) \triangleq \min\{\tau \in \mathbb{N} \,|\, X_\tau \geq \mu\}.$$

Lemma 6.2.7 For any arm i, $k \in \mathbb{N}$ and $t \in \{1,\dots,T\}$,

$$\mathbb{P}[\tau_{i,t} = k|\mathscr{F}_t] = \mathbb{P}[\tau(\mu_{i,t}) = k|\mu_{i,t}].$$

Proof. It is equivalent to show that for all k,

$$\mathbb{P}[\tau_{i,t} \geq k|\mathscr{F}_t] = \mathbb{P}[\tau(\mu_{i,t}) \geq k|\mu_{i,t}].$$

Notice that $\mathbb{1}[\tau_{i,t} \geq k] \iff \mu_i(N_{i,t-1}+k) < \mu$. Hence,

$$\mathbb{P}[\tau_{i,t} \geq k|\mathscr{F}_t] = \mathbb{P}[\mu_i(N_{i,t-1}+k) < \mu|\mathscr{F}_t] = F_{\mathscr{T}^k\delta_{\mu_{i,t}}}(\mu).$$

We also have,

$$\mathbb{P}[\tau(\mu_{i,t}) \geq k|\mu_{i,t}] = \mathbb{P}[X_k < \mu|X_0 = \mu_{i,t}] = F_{\mathscr{T}^k\delta_{\mu_{i,t}}}(\mu).$$

■

Lemma 6.2.8 If $x \leq y$, $\tau(x) \succeq \tau(y)$.

It further implies $\mathbb{E}[\tau(x)] \geq \mathbb{E}[\tau(y)]$.

Proof. For any $x \in \mathbb{R}$,

$$\mathbb{P}[\tau(x) \geq n] = \mathbb{P}[X_{n-1} < \mu | X_0 = x] = F_{\mathscr{T}^{n-1}\delta_x}(\mu) \tag{6.3}$$

where the first equality is justified by the definition of $\tau(x)$ and Assumption 6.2.2. Using Corollary 6.2.6, we show the stochastic dominance,

$$x \leq y \implies \mathscr{T}^{n-1}\delta_x \preceq \mathscr{T}^{n-1}\delta_y \implies \tau(x) \succeq \tau(y).$$

The last implication uses that $\mathbb{P}[\tau(x) \geq n] = 1 - F_{\tau(x)}(n) \geq \mathbb{P}[\tau(y) \geq n] = 1 - F_{\tau(y)}(n)$, where the inequality comes from Equation 6.3. It implies that $F_{\tau(x)}(n) \leq F_{\tau(xy}(n)$ which is the definition of stochastic dominance.

For the expectation, we use the layer-cake representation together with Equation 6.3,

$$\mathbb{E}[\tau(x)] = \sum_{n=1}^{+\infty} \mathbb{P}[\tau(x) \geq n] = \sum_{n=0}^{+\infty} F_{\mathscr{T}^n\delta_x}(\mu).$$

Hence, if $x \leq y$, then $\forall n \in \mathbb{N}$, $F_{\mathscr{T}^n\delta_x}(\mu) \geq F_{\mathscr{T}^n\delta_y}(\mu)$. Therefore,

$$\mathbb{E}[\tau(x)] \geq \mathbb{E}[\tau(y)]$$

where we used Corollary 6.2.6. ■

6.3 Optimal Oracle: Focus on the largest under the threshold

6.3.1 The FLUT oracle

An oracle policy $\widetilde{\pi}$ is a policy which has access to the current and past values of all arms $\{\mu_{i,s}\}_{i\in\mathscr{K},s\leq t}$ and to the transition matrix $\mathscr{T}$. More precisely, we define the set of states at any round t $\boldsymbol{\mu_t} = \{\mu_{i,t}\}_{i\in\mathscr{K}}$ and the random variables known by an oracle at t,

$$\mathscr{F}_t = \left\{\{\boldsymbol{\mu_s}\}_{1\leq s\leq t}, \{i_s\}_{1\leq s\leq t-1}\right\}.$$

Notice that the oracle does not have access to the future of the Markov Chain, and can only make projections based on $\mathscr{T}$ and $\mathscr{F}_t$.

We define the sets of arms under and above the threshold before the round t:

$$\mathscr{K}_t^- \triangleq \{i \in \mathscr{K} | \mu_{i,t} < \mu\}$$
$$\mathscr{K}_t^+ \triangleq \{i \in \mathscr{K} | \mu_{i,t} \geq \mu\}.$$

We describe Focus on the Largest Under the Threshold (FLUT) in Algorithm 9, an oracle policy $\widetilde{\pi}^\star$ which selects at each round the arm with the largest state below the threshold μ.

Algorithm 9 Focus on the Largest Under the Threshold (FLUT or $\widetilde{\pi}^\star$)

Require: μ
1: **for** $t \leftarrow 1, 2, \dots$ **do**
2: RECEIVE $\boldsymbol{\mu_t} \leftarrow \{\mu_{i,t}\}_{i \in \mathscr{K}}$
3: $\mathscr{K}_t^- \leftarrow \{i \in \mathscr{K} | \mu_{i,t} < \mu\}$
4: **if** $\mathscr{K}_t^- \neq \{\}$ **then** PULL[a] $i_t \in \arg\max_{i \in \mathscr{K}_t^-} \mu_{i,t}$;
5: **else** PULL AT RANDOM $i_t \in \mathscr{K}$
6: **end if**
7: **end for**

[a] One can choose the tie break selection rule arbitrarily, e.g. by selecting the arm with the smallest index.

R We note that $\widetilde{\pi}^\star$ is an oracle policy which does not use the knowledge of $\mathscr{T}$. It is similar to the optimal oracle for rotting bandits. It is an interesting feature as one can hope to approximate $\mathscr{T}$ by simply estimating the state of the arms like in bandits, and without caring about the transitions.

6.3.2 Optimality

Theorem 6.3.1 For any oracle policy $\widetilde{\pi}$ and any round t,

$$r_t(\widetilde{\pi}^\star) \geq r_t(\widetilde{\pi}).$$

Corollary 6.3.2 $\widetilde{\pi}^\star$ maximizes $\mathbb{E}[r_t(\pi)]$ without the knowledge of the round t. Therefore, it maximizes $\mathbb{E}[J_T(\pi)]$ for any horizon T.

6.3.3 Proof of Theorem 6.3.1

Sketch

The proof is quite technical. We give here a sketch highlighting the main difficulties. In the spirit of the Bellman Equation (Bellman 1966), our proof shows recursively from the end that selecting the largest arm under the threshold is the best thing to do concerning $r_{t:T}$, the future reward collected from the round t. More precisely, "the best thing to do" means that $\widetilde{\pi}^\star$ maximizes $\mathbb{P}[r_{t:T}(\widetilde{\pi}) \geq r | \mathscr{F}_t]$ for any reward objective r.

The initialization at the last round is rather straightforward given our assumptions. Indeed, according to Assumption 6.2.3, all the arms have the same transition operator. Moreover, according to Assumption 6.2.4, the probability to reach any threshold in one step increases with the value of the starting point. Hence, following FLUT maximizes the probability to

reach the threshold for the selected arm. Because the transitions are rested, we cannot bring more than $r = 1$ arm above the threshold. Hence, FLUT maximizes $\mathbb{P}[r_{T:T}(\cdot) \geq 1 | \mathscr{F}_T] = \mathbb{P}[r_{T:T}(\cdot) = 1 | \mathscr{F}_T]$.

Then, we consider a round t such that FLUT is the best thing to do from $t+1$. Hence, we compare FLUT which policies which follow any rule at t and then FLUT from $t+1$. We split the possibilities in three: (1) either $i_t \in \mathscr{K}_t^+$, or (2) i_t is in the r largest value below the threshold at the round t, or (3) i_t is below this r-th value.

(1) Arguably, selecting an arm $i_t \in \mathscr{K}_t^+$ is totally useless because this arm is already above the threshold and the transitions are rested (Assumption 6.2.1).

(2) In order to pass r arms above the threshold until the end of the game, $\widetilde{\pi}^\star$ first selects repetitively the largest arm in $\mathscr{K}_t^-$ until it reaches the threshold, then the second largest, etc., until the r-th. Hence, $\mathbb{P}[r_{t:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_t]$ is equal to $\mathbb{P}\left[\sum_{x \in \mathscr{O}_r(\boldsymbol{\mu}_t)} \tau(x) \leq T - t + 1 \middle| \mathscr{F}_t\right]$, that is, the probability that the sum of the remaining pulls to reach the threshold for the r largest arms below the threshold[1] is smaller than the remaining rounds (Lemma 6.3.4). Since the transitions are rested and Markov, the order of the pulls does not matter: it is necessary to advance the r Markov chains to get at least r rewards. Hence, selecting any arm among the r largest values below the threshold and then follow $\widetilde{\pi}^\star$ from $t+1$ achieves the same $\mathbb{P}[r_{t:T}(\cdot) \geq r | \mathscr{F}_t]$ than $\widetilde{\pi}^\star$.

(3) Comparing FLUT with the case where we pull an arm below the r-th value of $\mathscr{K}_t^-$ is the most difficult part of the proof. However, it seems quite intuitive with our Assumption 6.2.4 that pulling an arm that is among the furthest to the threshold is not optimal.

According to Lemma 6.3.4 and Corollary 6.3.5, if we follow FLUT after $t+1$, the only thing that matter with respect to $\mathbb{P}[r_{t:T}(\cdot) \geq r | \mathscr{F}_t]$ is the r largest states of $\mathscr{K}_{t+1}^-$ (or the $r-1$ largest states of $\mathscr{K}_{t+1}^-$ if the arm that we select at the round t reaches the threshold). The larger are those states, the higher is $\mathbb{P}[r_{t+1:T}(\cdot) \geq r | \mathscr{F}_{t+1}]$. After the t-th round, we move two different values below the threshold if we follow FLUT or if we take an other arm. It is hard to compare these two states in terms of potential reward. The trick is to use the last result: $\widetilde{\pi}^\star$ performs the same than the policy which selects the r-th value of $\mathscr{K}_t^-$ (with respect to $\mathbb{P}[r_{t:T}(\cdot) \geq r | \mathscr{F}_t]$).

If we compare to this policy instead of FLUT, the r largest states of $\mathscr{K}_{t+1}^-$ are the $r-1$ largest states of $\mathscr{K}_t^-$ and an other value. If we pull an arm i_r with the r-th value below the threshold at the round t, then this other value is $\mu_{i_r,t+1}$. If we pull an arm i_t below $\mu_{i_r,t}$, then the other value is $\max(\mu_{i_r,t}, \mu_{i_t,t+1})$. We can compare the distributions associated to these two random variables, and see that the first one stochastically dominates the other one (thanks to Assumption 6.2.4).

In the two cases, $\mathbb{P}[r_{t:T}(\cdot) \geq r | \mathscr{F}_t]$ is the expectation of $\mathbb{P}[r_{t+1:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_{t+1}]$ over these random variables. Since $\mathbb{P}[r_{t+1:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_{t+1}]$ is non-decreasing with the r largest values

[1] $\mathscr{O}_r(\boldsymbol{\mu}_t)$ is the set of r largest values below the threshold at any round t.

in $\mathscr{K}_t^-$, we can show that $\mathbb{P}\left[r_{t:T}(\widetilde{\pi}^\star) \geq r|\mathscr{F}_t\right] \geq \mathbb{P}\left[r_{t:T}(\widetilde{\pi}) \geq r|\mathscr{F}_t\right]$ thanks to Lemma 6.3.6.

It concludes the induction as it shows that for any arm choice at the round t, $\widetilde{\pi}^\star$ maximizes $\mathbb{P}\left[r_{t:T}(\cdot) \geq r|\mathscr{F}_t\right]$ for any r.

Proof. **Introduction**

According to the definition of the first order stochastic dominance (Equation 6.1), we want to show that for all $r \in \mathbb{N}$ and for any oracle policy $\widetilde{\pi}$,

$$\mathbb{P}\left[r_T(\widetilde{\pi}^\star) \geq r|\mathscr{F}_1\right] \geq \mathbb{P}\left[r_T(\widetilde{\pi}) \geq r|\mathscr{F}_1\right].$$

$\mathscr{F}_1$ represents indeed the information available to the oracle at the beginning of the game. We recall the definition of $\rho(x,y) \triangleq \mathbb{1}\left[x < \mu \wedge y \geq \mu\right]$. We define,

$$r_{s:t}(\pi) = \sum_{t'=s}^{t} \rho\left(\mu_{\pi(t'),t'}, \mu_{\pi(t'),t'+1}\right).$$

Using Equation 6.2, we can write,

$$r_T(\widetilde{\pi}) = r_{1:T}(\widetilde{\pi}) + \sum_{i \in \mathscr{K}} \mathbb{1}\left[\mu_{i,1} \geq \mu\right].$$

Since the above sum does not depend on the policy $\widetilde{\pi}$, we will show recursively from the end $t = T$ that for all $r \in \mathbb{N}$ that,

$$\mathbb{P}\left[r_{t:T}(\widetilde{\pi}^\star) \geq r|\mathscr{F}_t\right] \geq \mathbb{P}\left[r_{t:T}(\widetilde{\pi}) \geq r|\mathscr{F}_t\right].$$

Last round

At the last round $t = T$, the $r_{T:T}$ is equal to 1 if the selected arm is above the threshold and else to 0. For $r > 1$ and $r = 0$, we have the trivial equalities,

$$\mathbb{P}\left[r_{T:T}(\widetilde{\pi}) \geq 0|\mathscr{F}_T\right] = 1,$$
$$\mathbb{P}\left[r_{T:T}(\widetilde{\pi}) \geq 2|\mathscr{F}_T\right] = 0.$$

For $r = 1$, if $\widetilde{\pi}(T) = i_T \in \mathscr{K}_T^+$, the probability of reaching μ with a new arm is null because the arm is already above the threshold. Hence,

$$\mathbb{P}\left[r_{T:T}(\widetilde{\pi}) \geq 1|\mathscr{F}_T \wedge i_T \in \mathscr{K}_T^+\right] = 0 \leq \mathbb{P}\left[r_{T:T}(\widetilde{\pi}^\star) \geq 1|\mathscr{F}_T\right].$$

If $\widetilde{\pi}(T) = i_T \in \mathscr{K}_T^-$, we can use Assumption 6.2.4,

$$\begin{aligned}\mathbb{P}\left[r_{T:T}(\widetilde{\pi}) \geq 1|\mathscr{F}_T \wedge i_T \in \mathscr{K}_T^-\right] &= \mathbb{P}\left[\mu_{i_T,T+1} \geq \mu|\mathscr{F}_T \wedge i_T \in \mathscr{K}_T^-\right] \\ &\leq \mathbb{P}\left[\mu_{i_T^\star,T+1} \geq \mu|\mathscr{F}_T \wedge i_T = i_T^\star\right] \\ &= \mathbb{P}\left[r_{T:T}(\widetilde{\pi}^\star) \geq 1|\mathscr{F}_T\right].\end{aligned}$$

Indeed, by definition of $i_T^\star$, $\mu_{i_T^\star,T} \geq \mu_{i_T,T}$ if $i_T \in \mathscr{K}_T^-$. Therefore, we do have for all r and any oracle policy $\widetilde{\pi}$,

$$\mathbb{P}\left[r_{T:T}(\widetilde{\pi}^\star) \geq r|\mathscr{F}_T\right] \geq \mathbb{P}\left[r_{T:T}(\widetilde{\pi}) \geq r|\mathscr{F}_T\right].$$

Backward induction

Now, we consider a round t such that, for any $\widetilde{\pi}$ and r,

$$\mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_{t+1}\right] \geq \mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}) \geq r | \mathscr{F}_{t+1}\right]. \tag{6.4}$$

We want to show that this relation is still true at the round t,

$$\mathbb{P}\left[r_{t:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_{t}\right] \geq \mathbb{P}\left[r_{t:T}(\widetilde{\pi}) \geq r | \mathscr{F}_{t}\right].$$

We consider the policy $\widetilde{\pi}_t$ which follows $\widetilde{\pi}^\star$ except at the round t where it follows $\widetilde{\pi}$. Thus, for any $r \in \mathbb{N}$,

$$\mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_{t+1}\right] = \mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_{t+1}\right] \geq \mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}) \geq r | \mathscr{F}_{t+1}\right]. \tag{6.5}$$

The first equality is justified by the fact that the two policies behave the same from $t+1$, hence they collect the same reward. The inequality follows from Equation 6.4.

$$\begin{aligned}
\mathbb{P}\left[r_{t:T}(\widetilde{\pi}) \geq r | \mathscr{F}_{t}\right] =& \mathbb{E}\left[\mathbb{1}\left[\mu_{i_t,t+1} \geq \mu\right] \mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}) \geq r-1 | \mathscr{F}_{t+1}\right] | \mathscr{F}_t \wedge i_t \sim \widetilde{\pi}(t)\right] \\
&+ \mathbb{E}\left[\mathbb{1}\left[\mu_{i_t,t+1} < \mu\right] \mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}) \geq r | \mathscr{F}_{t+1}\right] | \mathscr{F}_t \wedge i_t \sim \widetilde{\pi}(t)\right] \\
\leq& \mathbb{E}\left[\mathbb{1}\left[\mu_{i_t,t+1} \geq \mu\right] \mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}_t) \geq r-1 | \mathscr{F}_{t+1}\right] | \mathscr{F}_t \wedge i_t \sim \widetilde{\pi}(t)\right] \\
&+ \mathbb{E}\left[\mathbb{1}\left[\mu_{i_t,t+1} < \mu\right] \mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_{t+1}\right] | \mathscr{F}_t \wedge i_t \sim \widetilde{\pi}(t)\right] \\
=& \mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_{t}\right].
\end{aligned} \tag{6.6}$$

The inequality follows from Equation 6.5: following the $\widetilde{\pi}^\star$ (or equivalently $\widetilde{\pi}_t$) is optimal after round t. The equalities mean that either arm i_t reaches the threshold at the round t and we still need $r-1$ arms to reach the threshold after the round t, or arm i_t do not reach the threshold and we need r arms to reach the threshold after t. To conclude the proof, we need to show that,

$$\mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_{t}\right] \leq \mathbb{P}\left[r_{t:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_{t}\right].$$

We call $\mathscr{O}_r(\boldsymbol{\mu_t})$ the r largest states below the threshold at the round t (for $r \leq |\mathscr{K}_t^-|$). We call $\mu_t^r \triangleq \min \mathscr{O}_r(\boldsymbol{\mu_t})$, the r-th largest value below the threshold. We call $\boldsymbol{\mu_t^i}$, the set of states at t excluding the state of i. Hence, $\mathscr{O}_r\left(\boldsymbol{\mu_t^i}\right)$ is the set of the r largest states below the threshold excluding the state of i. We distinguish three cases: when $\mu_{i_t,t} \in]-\infty, \mu_t^r[$, $\mu_{i_t,t} \in [\mu_t^r, \mu[$ and $\mu_{i_t,t} \in [\mu, +\infty[$. We call $\mathscr{K}_t^r \triangleq \{i \in \mathscr{K} | \mu_{i,t} \in \mathscr{O}_r(\boldsymbol{\mu_t})\} \subset \mathscr{K}_t^-$ such that the three aforementioned cases corresponds to respectively $i_t \in \mathscr{K}_t^- \setminus \mathscr{K}_t^r$, $i_t \in \mathscr{K}_t^r$ and $i_t \in \mathscr{K}_t^+$.

Backward induction: the selected arm is in the r largest values below the threshold

We will start by considering the case $i_t \in \mathscr{K}_t^r$. It is equivalent to $\mu_{i_t,t-1} \in \mathscr{O}_r(\boldsymbol{\mu_t})$. Lemma 6.3.4 becomes,

$$\mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^r\right] = \mathbb{P}\left[\sum_{x \in \mathscr{O}_r(\boldsymbol{\mu_t})} \tau(x) \leq T-t+1 \middle| \mathscr{F}_t\right].$$

Notice that this expression is independent of $i_t \in \mathscr{K}_t^r$. Therefore, since $i_t^\star \in \mathscr{K}_t^r$ for any r, we have that,

$$\mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^r\right] = \mathbb{P}\left[r_{t:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_t\right]. \tag{6.7}$$

Backward induction: the selected arm is below the r-th largest value below the threshold

We consider the case $i_t \in \mathscr{K}_t^- \setminus \mathscr{K}_t^r$. Hence, $\mu_{i_t,t} \notin \mathscr{O}_{r-1}(\boldsymbol{\mu}_t)$, which implies $\mathscr{O}_{r-1}(\boldsymbol{\mu}_t^{i_t}) = \mathscr{O}_{r-1}(\boldsymbol{\mu}_t)$. Moreover, since the setup is rested, $\mathscr{O}_{r-1}(\boldsymbol{\mu}_{t+1}^{i_t}) = \mathscr{O}_{r-1}(\boldsymbol{\mu}_t^{i_t})$ if i_t is selected at the round t. Hence, $\mathscr{O}_{r-1}(\boldsymbol{\mu}_{t+1}^{i_t}) = \mathscr{O}_{r-1}(\boldsymbol{\mu}_t)$. Thus, we can rewrite Lemma 6.3.4,

$$\begin{aligned}
&\mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^- \setminus \mathscr{K}_t^r\right] \\
&\quad = \mathbb{P}\left[\tau(\max\left(\mu_t^r, \mu_{i_t,t+1}\right)) + \sum_{x \in \mathscr{O}_{r-1}(\boldsymbol{\mu}_t)} \tau(x) \leq T - t \,\middle|\, \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^- \setminus \mathscr{K}_t^r\right].
\end{aligned} \tag{6.8}$$

Let $i_r \in \mathscr{K}$, an arm with value μ_{t-1}^r at the beginning of the round t. We have that,

$$\begin{aligned}
\mathbb{P}\left[r_{t:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_t\right] &= \mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_t \wedge i_t = i_r\right] \\
&= \mathbb{P}\left[\tau(\mu_{i_r,t+1}) + \sum_{x \in \mathscr{O}_{r-1}(\boldsymbol{\mu}_t)} \tau(x) \leq T - t \,\middle|\, \mathscr{F}_t \wedge i_t = i_r\right].
\end{aligned} \tag{6.9}$$

The first equation follows from Equation 6.7. The second uses Lemma 6.3.4 with $\mu_{i_r,t+1} \geq \mu_{i_r,t} = \mu_t^r$. We also use that with the same argument $\mathscr{O}_{r-1}(\boldsymbol{\mu}_t^{i_r}) = \mathscr{O}_{r-1}(\boldsymbol{\mu}_t)$ because i_r corresponds to the r-th value below the threshold. Hence, both $\mathbb{P}\left[r_{t:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_t\right]$ and $\mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_t\right]$ can be written as the mean of the function,

$$f(y) = \mathbb{P}\left[\tau(y) + \sum_{x \in \mathscr{O}_{r-1}(\boldsymbol{\mu}_t)} \tau(x) \leq T - t \,\middle|\, \mathscr{F}_t\right],$$

according to different probability densities. Because f is non-decreasing (Corollary 6.3.5), we only have to show that the probability density associated to $\widetilde{\pi}^\star$ stochastically dominates the probability density associated $\widetilde{\pi}_t$ (Lemma 6.3.6). The probability density associated to $\widetilde{\pi}^\star$ in Equation 6.9 is,

$$p_\star(y) = \mathbb{P}\left[\mu_{i_r,t+1} = y | \mathscr{F}_t \wedge i_t = i_r\right] = \mathscr{T}\delta_{\mu_{i_r,t}}.$$

The probability density associated to arm i_t in Equation 6.8 is,

$$p_{i_t}(y) = \mathbb{P}\left[\max\left(\mu_{i_t,t+1}, \mu_t^r\right) = y | \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^- \setminus \mathscr{K}_t^r\right].$$

In order to prove the stochastic dominance, we want to show that $F_{p_\star} \leq F_{p_{i_t}}$. Notice that p_{i_t} is the rectified probability density of $\mu_{i_t,t+1}$, where the mass below μ_t^r is transferred at μ_t^r. Hence, we can write its CDF as,

$$F_{p_{i_t}}(x) = \begin{cases} 0, & \text{if } x < \mu_t^r \\ F_{\mathscr{T}\delta_{\mu_{i_t,t}}}(x), & \text{otherwise.} \end{cases} \tag{6.10}$$

For $x < \mu_t^r$,

$$F_{p_\star}(x) = 0 = F_{p_{i_t}}(x). \tag{6.11}$$

The first equality comes from Assumption 6.2.2: since the reward is non-decreasing we have $\mathbb{P}[\mu_{i_r,t+1} < \mu_{i_r,t} | \mathscr{F}_t \wedge i_t = i_r] = 0$. The second equality comes from Equation 6.10. For $x \geq \mu_t^r$,

$$\forall x \geq \mu_t^r, F_{p_\star}(x) = F_{\mathscr{T}\delta_{\mu_{i_r,t}}}(x) \leq F_{\mathscr{T}\delta_{\mu_{i_t,t}}}(x) = F_{p_{i_t}}(x). \tag{6.12}$$

where we use Assumption 6.2.4 and the fact that $\mu_{i_t,t} \leq \mu_t^r$. According to Equations 6.11 and 6.12, we do have $F_{p_\star}(x) \leq F_{p_{i_t}}(x)$ for all x which is the definition of stochastic dominance: $p_\star \succeq p_{i_t}$. Therefore, because f is non decreasing (see Corollary 6.3.5 and Lemma 6.3.6), we conclude,

$$\mathbb{P}[r_{t:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_t] = \mathbb{E}_{p_\star}[f] \geq \mathbb{E}_{p_{i_t}}[f] = \mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^- \setminus \mathscr{K}_t^r\right]. \tag{6.13}$$

Backward induction: the selected arm is above the threshold

We consider the case $\mu_{i_t,1} > \mu$. Intuitively, selecting such arm is useless, because it does not bring any new arm above or closer to the threshold. We write formally this argument,

$$\begin{aligned}
\mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^+\right] &= \mathbb{P}\left[\mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_{t+1}\right] | \mathscr{F}_t\right] \\
&= \mathbb{P}\left[\mathbb{P}\left[\sum_{x \in \mathscr{O}_r(\boldsymbol{\mu_{t+1}})} \tau(x) \leq T - t | \mathscr{F}_{t+1}\right] | \mathscr{F}_t\right] \\
&= \mathbb{P}\left[\sum_{x \in \mathscr{O}_r(\boldsymbol{\mu_t})} \tau(x) \leq T - t | \mathscr{F}_t\right] \\
&\leq \mathbb{P}\left[\sum_{x \in \mathscr{O}_r(\boldsymbol{\mu_t})} \tau(x) \leq T - t + 1 | \mathscr{F}_t\right] \\
&= \mathbb{P}[r_{t:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_t].
\end{aligned} \tag{6.14}$$

The first equation means that no arm goes above the threshold at the round t. The second equation follows from Lemma 6.3.4. The third equation follows because by the rested assumption all the arm $i \in \mathscr{K}_t^-$ keep their value between t and $t+1$. Hence, $\mathscr{O}_r(\boldsymbol{\mu_t}) = \mathscr{O}_r(\boldsymbol{\mu_{t+1}})$ for all r. The inequation follows because the event in the RHS probability include the event in the LHS probability. Finally, we use again Lemma 6.3.4.

Conclusion

Putting together Equations 6.7, 6.13 and 6.14, we can write,

$$\mathbb{P}\left[r_{t:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_t\right] \geq \mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_t \wedge i_t\right].$$

Hence, if we average the RHS on $i_t \sim \widetilde{\pi}_t(t)$ (notice that $\widetilde{\pi}_t(t)$ is a $\mathscr{F}_t$-measurable distribution by definition of $\mathscr{F}_t$, we have,

$$\mathbb{P}\left[r_{t:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_t\right] \geq \mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_t\right].$$

Now, we can use Equation 6.6 to conclude the induction,

$$\mathbb{P}\left[r_{1:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_t\right] \geq \mathbb{P}\left[r_{t:T}(\widetilde{\pi}) \geq r | \mathscr{F}_t\right].$$

Hence, using the induction,

$$\mathbb{P}\left[r_{1:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_1\right] \geq \mathbb{P}\left[r_{1:T}(\widetilde{\pi}) \geq r | \mathscr{F}_1\right].$$

This statement concludes the proof, as we noticed in the Introduction. ■

6.3.4 Technical Lemmas

Lemma 6.3.3 Let A a random variable $(\mathscr{F}_t \wedge i_t)$-measurable. Let i_t the selected arm by $\widetilde{\pi}$ at a round t. Then,

$$\mathbb{P}\left[\tau\left(\mu_{i_t,t+1}\right) \leq A | \mathscr{F}_t \wedge i_t\right] = \mathbb{P}\left[\tau\left(\mu_{i_t,t}\right) \leq A+1 | \mathscr{F}_t \wedge i_t\right].$$

Proof. According to Lemma 6.2.7,

$$\mathbb{P}\left[\tau\left(\mu_{i_t,t+1}\right) \leq A | \mathscr{F}_t \wedge i_t\right] = \mathbb{P}\left[\tau_{i_t,t+1} \leq A | \mathscr{F}_t \wedge i_t\right].$$

If arm i_t is selected at a round t,

$$\tau_{i_t,t+1} \triangleq \tau_{i_t} - N_{i_t,t+1} = \tau_{i_t} - (N_{i_t,t}+1) = \tau_{i_t,t} - 1.$$

Hence, we can write,

$$\begin{aligned}\mathbb{P}\left[\tau_{i_t,t+1} \leq A | \mathscr{F}_t \wedge i_t\right] &= \mathbb{P}\left[\tau_{i_t,t} \leq A+1 | \mathscr{F}_t \wedge i_t\right] \\ &= \mathbb{P}\left[\tau\left(\mu_{i_t,t}\right) \leq A+1 | \mathscr{F}_t \wedge i_t\right].\end{aligned}$$

■

Lemma 6.3.4 We define the number of arms which passes the threshold between rounds t and T (included) when we follow policy π,

$$r_{t:T}(\pi) \triangleq \sum_{i \in \mathscr{K}} \mathbb{1}\left[\mu_{i,T+1} \geq \mu \wedge \mu_{i,t} < \mu\right]. \tag{6.15}$$

Let $\widetilde{\pi}_t$ the policy which follows $\widetilde{\pi}^\star$ except at the round t where it uses any decision rule such that $i_t \in \mathscr{K}_t^-$. We call $\mathscr{O}_r(\boldsymbol{\mu_t})$, the set of the $r \in \{1,\dots,|\mathscr{K}_t^-|\}$ largest arm below the threshold at the round t. We call $\mu_t^r = \min \mathscr{O}_r(\boldsymbol{\mu_t})$, the r-th value below the threshold. We call $\mathscr{O}_{r-1}(\boldsymbol{\mu_t^{i_t}})$, the set of the $r-1$ largest values below μ at the round t excluding $\mu_{i_t,t}$. Then,

$$\begin{aligned}
&\mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^-\right] \\
&\qquad = \mathbb{P}\left[\tau(\max\left(\mu_{i_t,t+1}, \mu_t^r\right)) + \sum_{x \in \mathscr{O}_{r-1}(\boldsymbol{\mu_t^{i_t}})} \tau(x) \leq T - t \,\middle|\, \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^-\right].
\end{aligned}$$

Let $\mathscr{K}_t^r \triangleq \left\{i \in \mathscr{K} \,|\, \mu_{i,t-1} \in \mathscr{O}_r(\boldsymbol{\mu_t})\right\} \subset \mathscr{K}_t^-$, the set of arms below the threshold with a state larger or equal than μ_{t-1}^r. In the special case where $i_t \in \mathscr{K}_t^r$ (e.g. $\widetilde{\pi}^\star$), we have,

$$\mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^r\right] = \mathbb{P}\left[\sum_{x \in \mathscr{O}_r(\boldsymbol{\mu_t})} \tau(x) \leq T - t + 1 \,\middle|\, \mathscr{F}_t\right].$$

Proof. We will prove this claim by induction from $t = T$. For $r = 1$, we have,

$$\begin{aligned}
&\mathbb{P}\left[r_{T:T}(\widetilde{\pi}_T) \geq 1 | \mathscr{F}_T \wedge i_T \in \mathscr{K}_T^-\right] \\
&\qquad = \mathbb{P}\left[r_{T:T}(\widetilde{\pi}_T) = 1 | \mathscr{F}_T \wedge i_T \in \mathscr{K}_T^-\right] \\
&\qquad = \mathbb{P}\left[\mu_{i_T,T+1} \geq \mu | \mathscr{F}_T \wedge i_T \in \mathscr{K}_T^-\right] \\
&\qquad = \mathbb{P}\left[\tau\left(\max\left(\mu_{i_T,T+1}, \mu_T^1\right)\right) = 0 | \mathscr{F}_T \wedge i_T \in \mathscr{K}_T^-\right] \\
&\qquad = \mathbb{P}\left[\tau\left(\max\left(\mu_{i_T,T+1}, \mu_T^1\right)\right) + \sum_{x \in \mathscr{O}_0\left(\boldsymbol{\mu_T^{i_T}}\right)} \tau(x) \leq 0 \,\middle|\, \mathscr{F}_T \wedge i_T \in \mathscr{K}_T^-\right].
\end{aligned}$$

The first equality is justified because, by the rested Assumption 6.2.1, at most one arm can pass above the threshold during a single round. The only arm which can go above the threshold is the selected one, that is i_T, which leads to the second equation. The third equation uses that $\tau\left(\max\left(\mu_{i_T,T+1}, \mu_T^1\right)\right) = 0 \iff \max\left(\mu_{i_T,T+1}, \mu_T^1\right) \geq \mu \iff \mu_{i_T,T+1} \geq \mu$ because $\mu_T^1 < \mu$ by definition of μ_t^r. Last, we use that $\mathscr{O}_0\left(\boldsymbol{\mu_T^{i_T}}\right) = \{\}$ and that $\tau(\cdot) \geq 0$ by definition of τ.

For $\widetilde{\pi}^\star$, which is the special case where $i_T = i_T^\star \in \mathscr{K}_T^1$, we can write,

$$\begin{aligned}
\mathbb{P}\left[r_{T:T}(\widetilde{\pi}^\star) \geq 1 | \mathscr{F}_T\right] &= \mathbb{P}\left[r_{T:T}(\widetilde{\pi}_T) \geq 1 | \mathscr{F}_T \wedge i_T = i_T^\star\right] \\
&= \mathbb{P}\left[\sum_{x \in \mathscr{O}_1(\boldsymbol{\mu_{T+1}})} \tau(x) \leq 0 \,\middle|\, \mathscr{F}_T \wedge i_T = i_T^\star\right] \\
&= \mathbb{P}\left[\sum_{x \in \mathscr{O}_1(\boldsymbol{\mu_T})} \tau(x) \leq 1 \,\middle|\, \mathscr{F}_T\right].
\end{aligned}$$

The second equation follows from $\mathscr{O}_1\left(\boldsymbol{\mu_{T+1}}\right) = \left\{\mu_{i_T^\star,T}\right\}$. The third equation uses Lemma 6.3.3 since there is only one element in the sum.

Last, we notice that for $r > 1$,

$$\begin{aligned}
&\mathbb{P}\left[r_{T:T}(\widetilde{\pi}_T) > 1 | \mathscr{F}_T \wedge i_T \in \mathscr{K}_T^-\right] = 0,\\
&\mathbb{P}\left[\tau\left(\max\left(\mu_{i_T,T+1}, \mu_T^1\right)\right) + \sum_{x \in \mathscr{O}_{r-1}\left(\boldsymbol{\mu}_{\boldsymbol{T}}^{\boldsymbol{i_T}}\right)} \tau(x) \leq 0 \middle| \mathscr{F}_T \wedge i_T \in \mathscr{K}_T^-\right] = 0,\\
&\mathbb{P}\left[\sum_{x \in \mathscr{O}_r(\boldsymbol{\mu_T})} \tau(x) \leq 1 \middle| \mathscr{F}_T\right] = 0.
\end{aligned}$$

First, because $\widetilde{\pi}_T$ cannot bring more than one arm above the threshold in one round. The second and third equations follows because $r > 1$ and, for any r' and $\mathbf{X}$,

$$\sum_{x \in \mathscr{O}_{r'}(\mathbf{X})} \tau(x) \geq \left|\mathscr{O}_{r'}(\mathbf{X})\right| = r'.$$

Indeed, notice that $\tau(x) \geq 1$ when $x < \mu$, which is the case by definition of $\mathscr{O}_r(\cdot)$. Therefore, we have the desired equations for all $r \leq |\mathscr{K}_T^-|$ at the round T.

By induction, we assume a round t such that $\widetilde{\pi}^\star$ verifies for all $r \leq |\mathscr{K}_{t+1}^-|$,

$$\mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_{t+1}\right] = \mathbb{P}\left[\sum_{x \in \mathscr{O}_r(\boldsymbol{\mu_{t+1}})} \tau(x) \leq T - t \middle| \mathscr{F}_{t+1}\right].$$

Since $\widetilde{\pi}_t$ follows the oracle after the round t, we have,

$$\mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_{t+1}\right] = \mathbb{P}\left[\sum_{x \in \mathscr{O}_r(\boldsymbol{\mu_{t+1}})} \tau(x) \leq T - t \middle| \mathscr{F}_{t+1}\right]. \tag{6.16}$$

We decompose the probability at the round t on either arm i_t reaches the threshold at the round t or not,

$$\begin{aligned}
&\mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^-\right]\\
&\quad = \mathbb{E}\left[\mathbb{1}\left[\mu_{i_t,t+1} \geq \mu\right] \mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}_t) \geq r - 1 | \mathscr{F}_{t+1}\right] | \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^-\right]\\
&\qquad + \mathbb{E}\left[\mathbb{1}\left[\mu_{i_t,t+1} < \mu\right] \mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_{t+1}\right] | \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^-\right].
\end{aligned} \tag{6.17}$$

We start with the first term in the sum. When $\mu_{i_t,t+1} \geq \mu$, we can write,

$$\mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}_t) \geq r-1 | \mathscr{F}_{t+1}\right] \tag{6.18}$$

$$= \mathbb{P}\left[\sum_{x \in \mathscr{O}_{r-1}(\boldsymbol{\mu_{t+1}})} \tau(x) \leq T-t \middle| \mathscr{F}_{t+1}\right]$$

$$= \mathbb{P}\left[\tau(\max\left(\mu_{i_t,t+1}, \mu_t^r\right)) + \sum_{x \in \mathscr{O}_{r-1}(\boldsymbol{\mu_{t+1}})} \tau(x) \leq T-t \middle| \mathscr{F}_{t+1}\right]$$

$$= \mathbb{P}\left[\tau(\max\left(\mu_{i_t,t+1}, \mu_t^r\right)) + \sum_{x \in \mathscr{O}_{r-1}(\boldsymbol{\mu}_t^{\boldsymbol{i_t}})} \tau(x) \leq T-t \middle| \mathscr{F}_{t+1}\right]. \tag{6.19}$$

The first equality follows from Equation 6.16. The second equality follows because $\tau(\max \mu_{i_t,t+1}, \mu_t^r) = \tau(\mu_{i_t,t+1}) = 0$ when $\mu_{i_t,t+1} \geq \mu > \mu_t^r$. The last equality follows because since $\mu_{i_t,t+1} \geq \mu \implies \mu_{i_t,t+1} \notin \mathscr{O}_{r-1}(\boldsymbol{\mu_{t+1}})$. Hence, $\mathscr{O}_{r-1}(\boldsymbol{\mu_{t+1}}) = \mathscr{O}_{r-1}(\boldsymbol{\mu}_{\boldsymbol{t+1}}^{\boldsymbol{i_t}})$. Moreover, because the transitions are rested, $\mathscr{O}_{r-1}(\boldsymbol{\mu}_{\boldsymbol{t+1}}^{\boldsymbol{i_t}}) = \mathscr{O}_{r-1}(\boldsymbol{\mu}_t^{\boldsymbol{i_t}})$.

For the second term in the sum - when $\mu_{i_t,t+1} < \mu$ - we can write,

$$\mathbb{P}\left[r_{t+1:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_{t+1}\right]$$

$$= \mathbb{P}\left[\sum_{x \in \mathscr{O}_{r}(\boldsymbol{\mu_{t+1}})} \tau(x) \leq T-t \middle| \mathscr{F}_{t+1}\right]$$

$$= \mathbb{P}\left[\tau(\max\left(\mu_{i_t,t+1}, \mu_t^r\right)) + \sum_{x \in \mathscr{O}_{r-1}(\boldsymbol{\mu}_t^{\boldsymbol{i_t}})} \tau(x) \leq T-t \middle| \mathscr{F}_{t+1}\right]. \tag{6.20}$$

Again, we use Equation 6.16. Then, we cut $\mathscr{O}_r(\boldsymbol{\mu_{t+1}})$ in two: On the one hand, the $r-1$ largest values below the threshold excepted $\mu_{i_t,t}$, that is $\mathscr{O}_{r-1}(\boldsymbol{\mu}_{\boldsymbol{t+1}}^{\boldsymbol{i_t}})$. It is equal to $\mathscr{O}_{r-1}(\boldsymbol{\mu}_t^{\boldsymbol{i_t}})$ by the rested assumption. On the other hand, the remaining value which is $\mu_{i_t,t+1}$ if $\mu_{i_t,t+1} \geq \mu_{t+1}^r$, or else μ_{t+1}^r. In that second case, we have that $\mu_{t+1}^r > \mu_{i_t,t+1} \geq \mu_{i_t,t}$. Therefore, by the rested assumption, the r largest values below the threshold remain the same between t and $t+1$. Hence, we have that $\mu_{t+1}^r = \mu_t^r$.

Notice that Equations 6.19 and 6.20 leads to the same result, independently on whether $\mu_{i_t,t+1} \geq \mu$ or not. Hence, we can rewrite Equation 6.17 to conclude the first part of the Lemma,

$$\mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^-\right]$$

$$= \mathbb{P}\left[\tau(\max\left(\mu_{i_t,t+1}, \mu_t^r\right)) + \sum_{x \in \mathscr{O}_{r-1}(\boldsymbol{\mu}_t^{\boldsymbol{i_t}})} \tau(x) \leq T-t \middle| \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^-\right]. \tag{6.21}$$

Now, we look at the special case where $\widetilde{\pi}_t$ selects an arm in $\mathscr{K}_t^r$. This is for instance the

case of $\widetilde{\pi}^\star$. We can rewrite Equation 6.21,

$$\begin{aligned}
&\mathbb{P}\left[r_{t:T}(\widetilde{\pi}_t) \geq r | \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^r\right] \\
&\qquad = \mathbb{P}\left[\tau(\mu_{i_t,t+1}) + \sum_{x \in \mathscr{O}_{r-1}(\boldsymbol{\mu}_t^{i_t})} \tau(x) \leq T - t \,\middle|\, \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^r\right] \\
&\qquad = \mathbb{P}\left[\tau(\mu_{i_t,t}) + \sum_{x \in \mathscr{O}_{r-1}(\boldsymbol{\mu}_t^{i_t})} \tau(x) \leq T - t + 1 \,\middle|\, \mathscr{F}_t \wedge i_t \in \mathscr{K}_t^r\right] \\
&\qquad = \mathbb{P}\left[\sum_{x \in \mathscr{O}_r(\boldsymbol{\mu}_t)} \tau(x) \leq T - t + 1 \,\middle|\, \mathscr{F}_t\right].
\end{aligned}$$

The first equation follows from Equation 6.21 with $i_t \in \mathscr{K}_t^r \implies \mu_{i_t,t+1} \geq \mu_{i_t,t} > \mu_t^r$. The second equation follows by Lemma 6.3.3. Indeed, $T-(t+1)-\sum_{x \in \mathscr{O}_{r-1}(\boldsymbol{\mu}_t^{i_t})} \tau(x)$ is a $(\mathscr{F}_t \wedge i_t)$-measurable random variable. The last equation is justified by $\mu_{i_t,t} \in \mathscr{O}_r(\boldsymbol{\mu}_t) \implies \mathscr{O}_r(\boldsymbol{\mu}_t) = \mathscr{O}_{r-1}(\boldsymbol{\mu}_t^{i_t}) \cup \{\mu_{i_t,t}\}$. Last, we notice that $\sum_{x \in \mathscr{O}_r(\boldsymbol{\mu}_t)} \tau(x)$ is $\mathscr{F}_t$-measurable and does not depend on which $i_t \in \mathscr{K}_t^r$, thus we drop the i_t dependency. ■

Corollary 6.3.5 $\mathbb{P}\left[r_{t:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_t\right] = f(\mathscr{O}_r(\boldsymbol{\mu}_t))$, with $f(\mathbf{x}) \triangleq \mathbb{P}\left[\sum_{j=1}^r \tau(x_j) \leq T - t + 1\right]$ a non-decreasing function of its r variables.

Proof. According to Lemma 6.3.4,

$$\begin{aligned}
\mathbb{P}\left[r_{t:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_t\right] &= \mathbb{P}\left[\sum_{x \in \mathscr{O}_r(\boldsymbol{\mu}_t)} \tau(x) \leq T - t + 1 \,\middle|\, \mathscr{F}_t\right] \\
&= \mathbb{P}\left[\sum_{x \in \mathscr{O}_r(\boldsymbol{\mu}_t)} \tau(x) \leq T - t + 1 \,\middle|\, \mathscr{O}_r(\boldsymbol{\mu}_t)\right].
\end{aligned}$$

Indeed $\mathbb{1}\left[\sum_{x \in \mathscr{O}_r(\boldsymbol{\mu}_t)} \tau(x) \leq T - t + 1\right]$ is independent of $\mathscr{F}_t$ given $\mathscr{O}_r(\boldsymbol{\mu}_t)$. Hence, $\mathbb{P}\left[r_{t:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_t\right]$ is a function of the r largest states below μ. We study the CDF of the random variable $\sum_{x \in \mathscr{O}_r(\boldsymbol{\mu}_t)} \tau(x)$ given $\mathscr{O}_r(\boldsymbol{\mu}_t)$ that is,

$$f_m(x_1, \ldots, x_r) = \mathbb{P}\left[\sum_{j=1}^r \tau(x_j) \leq m\right].$$

Notice that $\mathbb{P}\left[r_{t:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_t\right] = f_{T-t+1}(\mathscr{O}_r(\boldsymbol{\mu}_t))$. We want to show that f_m is non-decreasing with each variable. Let's consider the i-th variable. We use that the probability of the sum of independent variables is the convolution of probabilities,

$$f_m(x_1, \ldots, x_r) = \sum_{k=0}^m \mathbb{P}\left[\sum_{j \neq i}^r \tau(x_j) = k\right] * \mathbb{P}\left[\tau(x_i) \leq m - k\right].$$

Hence, f_m is the sum of non-decreasing functions of x_i. Hence, $\mathbb{P}[r_{t:T}(\widetilde{\pi}^\star) \geq r | \mathscr{F}_t]$ is non decreasing with respect to any variable $\mu_t^i \in \mathscr{O}_r(\boldsymbol{\mu_t})$ (the others being fixed). ■

Lemma 6.3.6 Let f a non-decreasing function. Let X and Y two random variables with probability densities $\{p_x, p_y\}$ such that $p_x \succeq p_y$. Then,

$$\mathbb{E}[f(X)] \geq \mathbb{E}[f(Y)].$$

Proof. This is a standard result for stochastic dominance that we show for completion. The key argument is that stochastic dominance implies a monotone coupling between the two distributions. Indeed, let $X(z) = F_{p_x}^{-1}(z)$ and $Y(z) = F_{p_y}^{-1}(z)$ with z a random variable uniformly drawn in $[0,1]$. We do have that X and Y are drawn with respective probability p_x and p_y. Moreover, since $F_{p_x} \leq F_{p_y}$ (stochastic dominance), we can write that $X(z) \geq Y(z)$.

Now we consider the expectation of a non decreasing function f,

$$\mathbb{E}[f(X) - f(Y)] = \int_0^1 f(X(z)) - f(Y(z))dz \geq 0.$$

■

6.4 What does random progression mean?

One of the main differences with rotting bandits is that evolution is not deterministic anymore. From the formal point of view, it is an extension of the setup, as deterministic evolution is a special case of stochastic evolution. Yet, it is not clear what is the meaning of a random progression of the student. In this section, we give different interpretations of the transition operator $\mathscr{T}$ and we discuss the possibility to measure it and test our different assumptions.

Uncertainty is often modeled with classical tools from the probability theory. As noticed by Lavenant and Aït-Kaci (2019), this theory does not provide a meaning to what probabilities mean. In fact, it does not even provide a procedure to assign probabilities in practice. It is merely a theory of how we can compute together probabilities to determine other probabilities. Lavenant and Aït-Kaci (2019) review three ways to assign a probability: the classical one, the frequentist one, and the subjectivist one.

The classical conception uses the indifference principle, which assumes that there exist some base events - the issues - which are equiprobable, and hence, one should count the number of issues that realize an event and divide by the total number of issues to get its probability. A classical example is the throw of a dice where we assume that each outcome has a probability $1/6$. Notice that the characterization of what are the equiprobable issues does *not* come from the probability theory. For instance, we can assume that the 11 outcomes of the sum of two dices are equiprobable and accurately use the probability theory. Yet, such theory will lead to wrong predictions when we compare to what happens in the

real world [2]. The fact that the correct equiprobable issues for two dices are the product ensemble of the ensembles of issues of each dice comes from external considerations: the symmetry in the geometry of one dice, the chaotic movement of rolling dices which "compensates" the fact that the dices are thrown together, etc. Can we use this classical conception to assign our probabilities $\mathscr{T}$? The example of the sum of two dices tells us that it is not because we don't know that we should assume uniform probability. The power of the classical method comes from the potential power of the indifference principle for the specific setup. It is not because physicists have no idea about how atoms in a gas are dispatched that they can make accurate predictions. It is in fact because they have a very accurate idea - all the micro configurations of atoms in a gas are equally likely - that statistical physics can make accurate predictions.

The frequentist conception - arguably the most well known in the bandits' community - defines the probability empirically as the limit of the observed frequency of the outcome of a given protocol. Notice that this is a definition, not a Theorem (we refer to the discussion about the status of the law of large numbers with respect to the frequentist interpretation by Lavenant and Aït-Kaci (2019)). It is only the repetition of the protocol which gives a sense - and value - to a probability.

In our context, we do have a protocol: each incoming student on the website is a new realization of the protocol. In the frequentist interpretation, the transition probabilities $\mathscr{T}(x,x')$ should be interpreted as the fraction of students that reach level x' after one question on the topic where there were at level x. Hence, maximizing our objectives in *expectation* means that we maximize the objective on average across the population of students.

The frequentist interpretation is often believed to be the most scientific due to the elegant way it arranges facts, experimental setup, and theory. However, relating probabilistic models to objective reality is not always straightforward. Assuming that the states x and x' is observable (they are not) and that we do have many students at each level x (we don't, since there is an infinite number of x), we would be able to estimate $\mathscr{T}(x,x')$ by simply measuring the fraction of incoming students. If we want to actually test our assumptions on $\mathscr{T}$ for the frequentist interpretation, one should be able to evaluate the transition model under partial observability and continuous state space. This is not a straightforward operation (Shani et al. 2005). In fact, before the listed Assumptions 6.2.1 to 6.2.4, we assume that the future is independent of the past given the present (the Markov property). While this assumption is popular, it is rarely tested (Bickenbach and Bode 2001), and, again, partial observability and continuous state space make tests more challenging.

There also exist subjective interpretations of probabilities. For instance, Lavenant and Aït-Kaci (2019) advertise the interpretation of De Finetti (1972): probabilities are the amount an individual is ready to bet on an event if they are rewarded by one if the event occurs. This interpretation is *antirealistic* as a probability does not have to match the facts

[2] It is not clear what is a "wrong prediction" in an uncertain world. Indeed, when we try to relate probabilities to facts, the probability theory always says that facts are possible. Hence, to make a probabilistic theory testable (or refutable), it is philosophically necessary to interpret very likely / unlikely events as certain / uncertain. This is what Emile Borel calls the "loi unique du hasard" (unique law of chance).

in any way: there is no need for the gambler to be good. The interest of this interpretation is that we can recover the rules of probability calculus by assuming rational gambler. For instance, the fact that probabilities are normalized to one is a consequence that no one wants to accept a bet where he or she is sure to lose (for a well-chosen weighting scheme on the different bets). In this interpretation, probability theory is a way to enforce coherence in one's system of belief. For instance, our Theorem 6.3.1 states that if the bets we are ready to accept on the student progression satisfies our different assumptions then we should accept a better odds for the bet "policy $\widetilde{\pi}^\star$ will achieve at least r rewards" than for "any other policy $\widetilde{\pi}$ will achieve at least r rewards".

6.5 Learning Perspectives

6.5.1 Regret

Like in the previous chapters, we define the cumulative regret with respect to the optimal policy,

$$R_T^c(\pi) \triangleq \mathbb{E}\left[J_T(\widetilde{\pi}^\star)\right] - J_T(\pi).$$

We also define the simple regret,

$$R_T^s(\pi) \triangleq \mathbb{E}\left[r_T(\widetilde{\pi}^\star)\right] - r_T(\pi).$$

6.5.2 Counter-examples and a new learning assumption

We present two counter-examples which shows that the regret can scale linearly with T when the arms are allowed to stagnate.

Counter-example 1: Stagnating arms near the threshold

We consider a two-arm game with deterministic transitions: each pull add a quantity $\varepsilon > 0$ to the arm's state,

$$\forall i \in \mathscr{K},\ \forall n \in \{1,\dots,T\},\ \mu_i(n+1) = \mu_i(n) + \varepsilon.$$

It verifies all our Assumptions 6.2.1 to 6.2.4. We consider two sets of initial conditions,

$$\mu_1^1(0) = \mu, \qquad \mu_2^1(0) = \mu - \frac{2T\varepsilon}{3}, \tag{6.22}$$

$$\mu_1^2(0) = \mu - \frac{(T+1)\varepsilon}{3}, \qquad \mu_2^2(0) = \mu - \frac{2T\varepsilon}{3}. \tag{6.23}$$

On the problem 1 (Eq. 6.22), $\widetilde{\pi}^\star$ pulls arm 2 $\lceil 2T/3 \rceil$ times. Then, all the arms are above the threshold and $\widetilde{\pi}^\star$ plays randomly. Hence, $r_T(\widetilde{\pi}^\star) = 2$ and $J_T(\widetilde{\pi}^\star) = T + \lfloor T/3 \rfloor$. On the problem 2 (Eq. 6.23), $\widetilde{\pi}^\star$ pulls arm 1 $\lceil T+1/3 \rceil$ times and then arm 2 until the end of the game. Hence, $r_T(\widetilde{\pi}^\star) = 1$ and $J_T(\widetilde{\pi}^\star) = T - \lfloor T+1/3 \rfloor$.

We consider the case $\varepsilon \to 0$. If $N_{1,T} \geq \lceil {}^{T+1}/_{3} \rceil$ on problem 1, $R_T^c(\pi) = \lfloor {}^{T}/_{3} \rfloor$ and $R_T^s(\pi) = 1$. Moreover, if $N_{1,T} < \lceil {}^{T+1}/_{3} \rceil$ on problem 2, $R_T^c(\pi) \geq \lceil {}^{2T}/_{3} \rceil - \lceil {}^{T+1}/_{3} \rceil$ ($R_T^s(\pi)$ can take the value 0 or 1). For ε small enough, the two problems cannot be distinguished in the presence of noise ($\sigma > 0$) and hence, any algorithm would suffer linear regret in the worst-case.

Counter-example 2: Infinitely close arms with diverging behaviors

We consider a two-arm game with deterministic transitions:

$$\forall i \in \mathcal{K},\ \forall n \in \{1,\dots,T\},\ \mu_i(n+1) = f(\mu_i(n)) \text{ with } f(x) = \begin{cases} 0 & \text{if } x = 0 \\ x+\varepsilon & \text{if } x \leq \frac{3T\varepsilon}{4} \\ \mu & \text{otherwise} \end{cases}.$$

We consider the following initial states,

$$\mu_1(0) = 0, \qquad \mu_2(0) = \varepsilon.$$

Hence, arm 1 is stationary and arm 2 will need $\lceil \frac{3T}{4} \rceil - 1$ pulls to reach the threshold. Hence, $J_T(\widetilde{\pi}^\star) \sim {}^{T}/_{4}$ and $r_T(\widetilde{\pi}^\star) = 1$.

For ε small enough, the two arms cannot be distinguished with reasonable confidence before arm 2 reaches the threshold. Since we cannot pull both arms $\sim {}^{3T}/_{4}$, we cannot do much better than betting on one of the arms, and suffering $R_T(\pi) \sim {}^{T}/_{4}$ in half of the cases.

A new assumption

The two counter-examples show that stagnation is a problem for learning. We make a new assumption to limit this kind of behavior,

Assumption 6.5.1 Let $\varepsilon > 0$. Let $y \leq x + \varepsilon$. Then, $\mathscr{T}(x,y) = 0$.

Notice that this assumption is quite strong as it assumes that the selected state is always progressing by at least ε. Instead, we could assume that the state progresses by at least ε in expectation.

We hope to derive an ε-dependent lower bound by adapting the previous counter-examples. However, we can already state that small values of ε correspond to the hardest cases. For $\varepsilon \sim T^{-3/2}$, the worst-case regret is linear.

6.5.3 Focus on the Largest Under the Threshold with Exploration (FLUT-E)

Upper and lower confidence bounds on an increasing sequence

In Subsection 4.2.3, we use the fact that the rewards were decreasing in rotting bandits to compute an upper-confidence bound on the value of the next pull. Following the same idea, we use the increasing Assumption 6.2.2 to derive a lower-confidence bound on the value of the last pull at a round t, and an upper-confidence bound on the value of the first pull of each arm.

Estimators As in Subsection 4.1.3, we define the average of the last h observations of arm i at time t for learning policy π as

$$\widehat{\mu}_i^h(t,\pi) \triangleq \frac{1}{h}\sum_{s=1}^{t-1} \mathbb{1}\left(\pi(s)=i \wedge N_{i,s} > N_{i,t-1}-h\right) o_s,$$

and the average of the associated means as

$$\overline{\mu}_i^h(t,\pi) \triangleq \frac{1}{h}\sum_{s=1}^{t-1} \mathbb{1}\left(\pi(s)=i \wedge N_{i,s} > N_{i,t-1}-h\right) \mu_i(N_{i,s-1}).$$

Similarly, we also define the average of the first h observations of arm i at time t for policy π as,

$$\widehat{\mu}_i^{1:h}(t,\pi) \triangleq \frac{1}{h}\sum_{s=1}^{t-1} \mathbb{1}\left(\pi(s)=i \wedge N_{i,s} \leq h\right) o_s,$$

and the average of the associated means as

$$\overline{\mu}_i^{1:h}(t,\pi) \triangleq \frac{1}{h}\sum_{s=1}^{t-1} \mathbb{1}\left(\pi(s)=i \wedge N_{i,s} \leq h\right) \mu_i(N_{i,s-1}).$$

We recall that $c(h,\delta) = \sqrt{2\sigma^2 \log(2/\delta)/h}$. We define the ucb and lcb statistics,

$$\begin{aligned} \mathrm{ucb}(i,\delta) &= \min_{h \leq N_{i,t-1}} \widehat{\mu}_i^{1:h}(t,\pi) + c(h,\delta), \\ \mathrm{lcb}(i,\delta) &= \max_{h \leq N_{i,t-1}} \widehat{\mu}_i^{h}(t,\pi) - c(h,\delta). \end{aligned} \tag{6.24}$$

`FLUT-E` algorithm

We present the Focus on the Largest Under the Threshold with Exploration (`FLUT-E`) in Algorithm 10. During each phase p, `FLUT-E` (1) explore to find a good (hopefully, the best) arm below the threshold; and (2) focus on this arm until we are sure enough that its value is above the threshold.

At Line 9, the algorithm estimates $\mathscr{K}_t^-$. More precisely, it returns all the arms whose lcb on their last value is above μ. It corresponds to the arms for which we are not sufficiently sure that they are in $\mathscr{K}_t^+$. Since we want to stop pulling the arms in $\mathscr{K}_t^+$, it is important to be confident that they indeed reach the threshold. That is why we discard i_p if it is not in $\widehat{\mathscr{K}_t^-}$ (Line 11). We also increase the phase counter p by the number of arms which are detected above the threshold between $t-1$ and t (Line 10).

At Line 15, we select (if it exists) i_p, an arm whose lcb on the last value is at most at a distance Δ from the best ucb among arms in $\widehat{\mathscr{K}_t^-}$. By doing so, we guarantee that the selected arm is not too far from the best arm under the threshold selected by `FLUT`.

When this arm does not exist, we continue our round-robin exploration (Line 18). In practice (or maybe in theory), it may be interesting to filter out the arms for which we are sure that they are not among the best ones. For instance, we suggest restricting the round-robin exploration to the arms in $\left\{i \in \widehat{\mathscr{K}_t^-} \,|\, \mathrm{ucb}(i,\delta_T) \geq \max_{j \in \widehat{\mathscr{K}_t^-}} \mathrm{lcb}(j,\delta_T)\right\}$. Notice that when there are no arms in this set, then there is at least one candidate for arm i_p (Line 15).

Algorithm 10 Focus on the Largest Under the Threshold with Exploration (`FLUT-E`)

Require: μ, Δ, δ_T

1: $p \leftarrow 1$
2: $i_p \leftarrow$ `Null`
3: $\widehat{\mathscr{K}_0^-} \leftarrow \mathscr{K}$
4: **for** $t \leftarrow 1,2,\ldots,K$ **do** ▷ *Pull each arm once*
5: PULL $i_t \leftarrow t$; RECEIVE o_t
6: **end for**
7: **for** $t \leftarrow K+1, K+2, \ldots$ **do**
8: COMPUTE $\{\mathrm{lcb}(i,\delta_T), \mathrm{ucb}(i,\delta_T)\}_{i,\in\mathscr{K}}$ ▷ *Equation 6.24*
9: $\widehat{\mathscr{K}_t^-} \leftarrow \{i \in \mathscr{K} \,|\, \mathrm{lcb}(i,\delta_T) \leq \mu\}$
10: $p \leftarrow p + |\widehat{\mathscr{K}_{t-1}^-} \setminus \widehat{\mathscr{K}_t^-}|$
11: **if** i_p is not `Null` **and** $i_p \notin \widehat{\mathscr{K}_t^-}$ **then**
12: $i_p \leftarrow$ `Null`
13: **end if**
14: **if** i_p is `Null` **then**
15: $i_p \in \left\{i \in \widehat{\mathscr{K}_t^-} \,|\, \max_{j \in \widehat{\mathscr{K}_t^-}} \mathrm{ucb}(j,\delta_T) - \mathrm{lcb}(i,\delta_T) \leq \Delta\right\}$[a];
16: **end if**
17: **if** i_p is `Null` **then**
18: $i_t \in \arg\min_{i \in \widehat{\mathscr{K}_t^-}} N_{i,t}$
19: **else**
20: $i_t \leftarrow i_p$
21: **end if**
22: PULL i_t RECEIVE o_t
23: **end for**

[a] One can choose the tie break selection rule arbitrarily, e.g. by selecting the arm with the smallest index.

6.5.4 Regret upper bound perspectives

For simplicity, we consider the case where all the arms are below the threshold at the beginning. Without loss of generality, we assume that arms are ordered by their starting value.

Like in the previous chapters, we can design a high probability event such that our estimators are well concentrated. On this high probability event, we know that arms which

are not in $\widehat{\mathscr{K}_t}^-$ are above the threshold. It can be interesting to upper-bound the expected duration of the phases of FLUT-E compared to the ones of FLUT. The phase p of FLUT is simply the number of rounds it takes to reach the threshold from $\mu_p(0)$. There are three sources of overhead for the duration of the phase p of FLUT-E.

First, FLUT-E spends pulls in its exploration phase. The exploration stops when an arm i_p is found. Even in the case where the arms are near-stationary, the condition at Line 15 will be fulfilled when the confidence bound $c(N_{expl}, \delta_T)$ becomes comparable with Δ. (We conjecture : $4c(N_{expl}, \delta_T) \leq \Delta$). It gives an upper-bound on the number of exploration pulls N_{expl} and finally on the delay KN_{expl}.

Second, the arm i_p which is selected is not the best below the threshold as in FLUT. Yet, we conjecture that $\mu_{i_p}(N_{i,t}) \geq \mu_p(0) - \Delta$. With the Assumption 6.5.1, an imprecision of size Δ costs Δ/ε in number of pulls. Notice that Δ is a parameter of our algorithm, and we should tune its value to balance the two aforementioned costs.

Third, in the learning setup, there is a delay to detect when an arm is above the threshold. Thanks to our Assumption 6.5.1, the state cannot stay near μ for too long. After N_{detect} pulls, we conjecture that $\mu + N_{detect}\varepsilon/2 - 2c(N_{detect}, \delta_T) \leq \text{lcb}(i, \delta_T)$ due to this minimal increase. Hence, the condition at Line 11 will be necessary fulfilled when $N_{detect}\varepsilon$ has the same order of magnitude than $c(N_{detect}, \delta_T)$.

We are still quite far to get an upper bound on R_T^c (or R_T^s). Indeed, we need to characterize precisely how these overheads add together when we evaluate the total reward at round T.

6.6 Practical considerations for ITS applications

The fact that FLUT-E focuses on a given topic after the exploration phase is an interesting feature for ITS applications. Yet, the initial exploration phase may be very long if we try to learn from scratch for each student.

6.6.1 Including prior knowledge

If one topic is often easier than the others for students, we would like to use this prior information to speed up the exploration. Using Bayesian statistics instead of frequentist tools is a natural way to work with prior information (see Subsection 2.2.3 for the stationary bandits case).

How can we learn the prior? Knowledge Tracing (Desmarais and Baker 2012) is the application of (often online) supervised learning to the prediction of the student's answer given the question and past interactions. Wilson et al. (2016) design shallow models which outperform deep networks (Piech et al. 2015; Khajah et al. 2016; Xiong et al. 2016) in their experiments. These models use some variations of a classical student model - the item response theory - with a Bayesian learning method. In its simplest form, Item Response

Theory (Hambleton and Swaminathan 2013) associates to each student a proficiency θ_s and to each exercise a difficulty d_i such that the difference $\theta_s - d_i$ is fed in a logistic model to output the probability of success. Wilson et al. (2016) add a hierarchical Bayesian structure: each item difficulty have a prior which depends on the topic difficulty which is a parameter drawn from an uninformative Gaussian prior.

We could replace the frequentist confidence levels in `FLUT-E` by Bayesian credible intervals on the parameters of a similar model. Indeed, we can estimate credible intervals with MCMC sampling, which is often used in Bayesian learning (Andrieu et al. 2003). This approach would incorporate prior knowledge (learned from the other students' data) and enable shared knowledge between arms.

6.6.2 The exercises population is finite

On Afterclasse, there are roughly 20 questions per couple topic-difficulty. Notice that our confidence band is quite large for this number of samples: $c(h = 20, \delta_t = 10\%) \sim 0.16$. Hence, even if a student answers the 20 questions correctly, its lcb on the topic will be smaller than 0.85. It is a problem if the targeted μ is above 0.85. We suggest using the ratio of answered questions as a multiplicative factor in front of the confidence band in the lcb / ucb definition (Equation 6.24). Hence, when a window h includes all the questions, the associated confidence level becomes the empirical average (no uncertainty).

6.6.3 Tuning Δ with ε

We have already noticed that Δ should be tuned theoretically according to ε: the smaller the ε, the more accurate we need to be in the exploration phase. However, in practice, we do not know ε. We can estimate from data the average progression of students per question, but (1) it is not the minimum progression, and (2) it is not student-specific.

We suggest to overestimate ε (and, hence, Δ) at the beginning of the game. It would reduce the exploration phase and with the usage of prior information, it is even possible that the algorithm directly starts to exploit an arm. We can decrease the value of Δ if the student starts to wheel-spin, that is when a phase lasts for too long. We believe that for an ITS application it is indeed better to take a guess and start focusing on a topic, and explore only if the student shows unexpected difficulties.

6.6.4 Managing difficulty with a Zone of Proximal Development

In Section 6.1, we mention that the Rotting Bandits framework can hardly take different difficulties into account. In the current framework, we can use `FLUT-E` together with the Zone of Proximal Development paradigm (Luckin 2001; Clement et al. 2015). The arms are the different topic-difficulty pairs, but we locked the advanced difficulties at the beginning. We unlock them when the student validates the easier difficulty associated with that topic, that is when the easier arm is not in $\widehat{\mathscr{K}_t^-}$.

IV

References

References 191

References

– Tu n'as même pas appris le métier de tailleur ? dit-elle.
– Jamais, répondit K.
– Quelle est ta profession ?
– Arpenteur.
– Qu'est-ce là ?
K. le lui expliqua, l'explication la fit bâiller.

Franz Kafka, Le Château, *Dernier Chapitre*.

Abbasi-yadkori, Yasin, Dávid Pál, and Csaba Szepesvári (2011). "Improved Algorithms for Linear Stochastic Bandits". In: *Advances in Neural Information Processing Systems 24*. Edited by J Shawe-Taylor, R S Zemel, P L Bartlett, F Pereira, and K Q Weinberger. Curran Associates, Inc., pages 2312–2320. URL: http://papers.nips.cc/paper/4417-improved-algorithms-for-linear-stochastic-bandits.pdf (cited on page 43).

Abe, Naoki and Philip M Long (1999). "Associative Reinforcement Learning Using Linear Probabilistic Concepts". In: *Proceedings of the Sixteenth International Conference on Machine Learning*. ICML '99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pages 3–11. ISBN: 1558606122 (cited on page 43).

Agarwal, Alekh, Haipeng Luo, Behnam Neyshabur, and Robert E Schapire (2017). "Corralling a band of bandit algorithms". In: *Conference on Learning Theory*. PMLR, pages 12–38 (cited on page 135).

Agrawal, Shipra and Navin Goyal (2013). "Further Optimal Regret Bounds for Thompson Sampling". In: *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*. Edited by Carlos M Carvalho and Pradeep Ravikumar. Volume 31. Proceedings of Machine Learning Research. Scottsdale, Arizona, USA: PMLR, pages 99–107. URL: http://proceedings.mlr.press/v31/agrawal13a.html (cited on page 38).

Andrieu, Christophe, Nando De Freitas, Arnaud Doucet, and Michael I Jordan (2003). "An introduction to MCMC for machine learning". In: *Machine learning* 50.1-2, pages 5–43 (cited on pages 38, 187).

Astrom, Karl J (1965). "Optimal control of Markov processes with incomplete state information". In: *Journal of mathematical analysis and applications* 10.1, pages 174–205 (cited on page 44).

Audibert, Jean-Yves and Sébastien Bubeck (2009). "Minimax policies for adversarial and stochastic bandits". In: *Proceedings of the Conference on Learning Theory (COLT), 2009*, pages 217–226. URL: https://hal-enpc.archives-ouvertes.fr/hal-00834882 (cited on pages 37, 39, 85).

— (June 2010). "Best Arm Identification in Multi-Armed Bandits". In: *COLT - 23th Conference on Learning Theory - 2010*. Haifa, Israel, 13 p. URL: https://hal-enpc.archives-ouvertes.fr/hal-00654404 (cited on page 48).

Audibert, Jean-Yves, Rémi Munos, and Csaba Szepesvári (Apr. 2009). "Exploration-exploitation tradeoff using variance estimates in multi-armed bandits". In: *Theoretical Computer Science* 410.19, pages 1876–1902. ISSN: 03043975. DOI: 10.1016/j.tcs.2009.01.016 (cited on page 36).

Auer, Peter (2002). "Using Confidence Bounds for Exploitation-Exploration Trade-offs". In: *Journal of Machine Learning Research* 3.Nov, pages 397–422. ISSN: ISSN 1533-7928 (cited on page 43).

Auer, Peter, Nicolò Cesa-Bianchi, and Paul Fischer (May 2002). "Finite-time analysis of the multiarmed bandit problem". In: *Machine Learning* 47.2-3, pages 235–256. ISSN: 08856125. DOI: 10.1023/A:1013689704352 (cited on pages 38, 61, 86, 118, 142).

Auer, Peter, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire (Jan. 2003). "The nonstochastic multiarmed bandit problem". In: *SIAM Journal on Computing* 32.1, pages 48–77. ISSN: 00975397. DOI: 10.1137/S0097539701398375 (cited on pages 35, 39, 40, 42, 50, 74, 75, 133, 134, 137, 141, 152).

Auer, Peter, Pratik Gajane, and Ronald Ortner (2019). "Adaptively Tracking the Best Bandit Arm with an Unknown Number of Distribution Changes". In: *Proceedings of the Thirty-Second Conference on Learning Theory*. Edited by Alina Beygelzimer and Daniel Hsu. Volume 99. Proceedings of Machine Learning Research. Phoenix, USA: PMLR, pages 138–158. URL: http://proceedings.mlr.press/v99/auer19a.html (cited on pages 58, 135, 141).

Balouek, Daniel, Alexandra Carpen Amarie, Ghislain Charrier, Frédéric Desprez, Emmanuel Jeannot, Emmanuel Jeanvoine, Adrien Lèbre, David Margery, Nicolas Niclausse, Lucas Nussbaum, Olivier Richard, Christian Perez, Flavien Quesnel, Cyril Rohr, and Luc Sarzyniec (2013). "Adding Virtualization Capabilities to the Grid'5000 Testbed". In: *Communications in Computer and Information Science*. Volume 367 CCIS. Springer Verlag, pages 3–20. ISBN: 9783319045184. DOI: 10.1007/978-3-319-04519-1_1 (cited on page 97).

Beck, Joseph E and Yue Gong (2013). "Wheel-spinning: Students who fail to master a skill". In: *International conference on artificial intelligence in education*. Springer, pages 431–440 (cited on page 163).

Bellman, Richard (1966). "Dynamic programming". In: *Science* 153.3731, pages 34–37 (cited on pages 44, 169).

Besbes, Omar, Yonatan Gur, and Assaf Zeevi (2014). "Stochastic Multi-Armed-Bandit Problem with Non-stationary Rewards". In: *Advances in Neural Information Processing Systems 27*. Edited by Z Ghahramani, M Welling, C Cortes, N D Lawrence, and K Q Weinberger. Curran Associates, Inc., pages 199–207. arXiv: 1405.3316. URL:

http://papers.nips.cc/paper/5378-stochastic-multi-armed-bandit-problem-with-non-stationary-rewards.pdf (cited on pages 43, 58, 135–137).

Besson, Lilian (2018). *SMPyBandits: an Open-Source Research Framework for Single and Multi-Players Multi-Arms Bandits (MAB) Algorithms in Python*. Online at: \url{GitHub.com/SMPyBandits/SMPyBandits}. URL: https://github.com/SMPyBandits/SMPyBandits/ (cited on page 96).

Besson, Lilian and Emilie Kaufmann (Mar. 2018). "What Doubling Tricks Can and Can't Do for Multi-Armed Bandits". In: arXiv: 1803.06971. URL: http://arxiv.org/abs/1803.06971 (cited on pages 71, 141).

— (Feb. 2019). "The Generalized Likelihood Ratio Test meets klUCB: an Improved Algorithm for Piece-Wise Non-Stationary Bandits". In: arXiv: 1902.01575. URL: http://arxiv.org/abs/1902.01575 (cited on pages 58, 74, 135, 152, 153).

Besson, Lilian, Emilie Kaufmann, and Christophe Moy (2018). "Aggregation of multi-armed bandits learning algorithms for opportunistic spectrum access". In: *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, pages 1–6 (cited on page 135).

Bickenbach, Frank and Eckhardt Bode (2001). "Markov or Not Markov? This Should Be a Question". In: *Regional Science and Urban Economics*. ISSN: 0166-0462 (cited on page 181).

Bifet, Albert and Ricard Gavaldà (2007). "Learning from time-changing data with adaptive windowing". In: *Proceedings of the 7th SIAM International Conference on Data Mining*, pages 443–448. ISBN: 9780898716306. DOI: 10.1137/1.9781611972771.42 (cited on page 98).

Bouneffouf, Djallel and Raphael Féraud (Sept. 2016). "Multi-armed bandit problem with known trend". In: *Neurocomputing* 205, pages 16–21. ISSN: 0925-2312. DOI: 10.1016/J.NEUCOM.2016.02.052. URL: https://www.sciencedirect.com/science/article/pii/S092523121600299X (cited on page 42).

Bubeck, Sébastien and Nicolo Cesa-Bianchi (2012). "Regret analysis of stochastic and nonstochastic multi-armed bandit problems". In: *arXiv preprint arXiv:1204.5721* (cited on page 51).

Burnetas, Apostolos N. and Michael N. Katehakis (June 1996). "Optimal adaptive policies for sequential allocation problems". In: *Advances in Applied Mathematics* 17.2, pages 122–142. ISSN: 01968858. DOI: 10.1006/aama.1996.0007 (cited on page 35).

Cannan, Edwin (Mar. 1892). "The Origin of the Law of Diminishing Returns, 1813-15". In: *The Economic Journal* 2.5, page 53. ISSN: 00130133. DOI: 10.2307/2955940. URL: https://www.jstor.org/stable/2955940 (cited on page 57).

Cao, Yang, Zheng Wen, Branislav Kveton, and Yao Xie (2019). "Nearly Optimal Adaptive Procedure with Change Detection for Piecewise-Stationary Bandit". In: *Proceedings of Machine Learning Research*. Edited by Kamalika Chaudhuri and Masashi Sugiyama. Volume 89. Proceedings of Machine Learning Research. PMLR, pages 418–427. URL: http://proceedings.mlr.press/v89/cao19a.html (cited on pages 58, 134, 152, 153).

Cappé, Olivier, Aurélien Garivier, Odalric Ambrym Maillard, Rémi Munos, and Gilles Stoltz (2013). "Kullback–leibler upper confidence bounds for optimal sequential allocation". In: *Annals of Statistics* 41.3, pages 1516–1541. ISSN: 00905364. DOI: 10.1214/13-AOS1119. arXiv: 1210.1136 (cited on pages 36, 119).

Carpentier, Alexandra and Andrea Locatelli (2016). "Tight (Lower) Bounds for the Fixed Budget Best Arm Identification Bandit Problem". In: *29th Annual Conference on Learning Theory*. Edited by Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir. Volume 49. Proceedings of Machine Learning Research. Columbia University, New York, New York, USA: PMLR, pages 590–604. URL: http://proceedings.mlr.press/v49/carpentier16.html (cited on page 48).

Cella, Leonardo and Nicolo Cesa-Bianchi (2020). "Stochastic Bandits with Delay-Dependent Payoffs". In: edited by Silvia Chiappa and Roberto Calandra. Volume 108. Proceedings of Machine Learning Research. Online: PMLR, pages 1168–1177. URL: http://proceedings.mlr.press/v108/cella20a.html (cited on page 43).

Chapelle, Olivier and Lihong Li (2011). "An Empirical Evaluation of Thompson Sampling". In: *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, pages 2249–2257 (cited on pages 42, 49, 105).

Chen, Yifang, Chung-Wei Lee, Haipeng Luo, and Chen-Yu Wei (2019). "A New Algorithm for Non-stationary Contextual Bandits: Efficient, Optimal and Parameter-free". In: *Proceedings of the Thirty-Second Conference on Learning Theory*. Edited by Alina Beygelzimer and Daniel Hsu. Volume 99. Proceedings of Machine Learning Research. Phoenix, USA: PMLR, pages 696–726. URL: http://proceedings.mlr.press/v99/chen19b.html (cited on pages 58, 135, 136).

Cheung, Wang Chi, David Simchi-Levi, and Ruihao Zhu (2019). "Learning to Optimize under Non-Stationarity". In: *Proceedings of Machine Learning Research*. Edited by Kamalika Chaudhuri and Masashi Sugiyama. Volume 89. Proceedings of Machine Learning Research. PMLR, pages 1079–1087. URL: http://proceedings.mlr.press/v89/cheung19b.html (cited on pages 58, 135, 136).

Chow, Yuan Shih and Henry. Teicher (1997). *Probability theory : independence, interchangeability, martingales*. Springer, page 488. ISBN: 9780387982281. URL: https://www.springer.com/gp/book/9780387982281 (cited on page 68).

Clement, Benjamin, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes (2015). "Multi-Armed Bandits for Intelligent Tutoring Systems". In: *Journal of Educational Data Mining* 7.2 (cited on pages 34, 50–52, 187).

Daley, Daryl J (1968). "Stochastically monotone Markov chains". In: *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 10.4, pages 305–317 (cited on pages 165, 166).

De Finetti, B (1972). *Probability, Induction and Statistics: The Art of Guessing*. WILEY SERIES in PROBABILITY and STATISTICS: PROBABILITY and STATISTICS SECTION Series. J. Wiley. ISBN: 9780471201403. URL: https://books.google.fr/books?id=hENg7qRPOPYC (cited on page 181).

Degenne, Rémy and Vianney Perchet (2016). "Anytime optimal algorithms in stochastic multi-armed bandits". In: *Proceedings of The 33rd International Conference on Machine Learning*. Edited by Maria Florina Balcan and Kilian Q Weinberger. Volume 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, pages 1587–1595. URL: http://proceedings.mlr.press/v48/degenne16.html (cited on pages 37, 118–120).

Desmarais, Michel C and Baker (2012). "A review of recent advances in learner and skill modeling in intelligent learning environments". In: *User Modeling and User-Adapted Interaction* 22.1-2, pages 9–38 (cited on page 186).

Erraqabi, Akram, Alessandro Lazaric, Michal Valko, Emma Brunskill, and Yun-En Liu (2017). "Trading off Rewards and Errors in Multi-Armed Bandits". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Edited by Aarti Singh and Jerry Zhu. Volume 54. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, pages 709–717. URL: http://proceedings.mlr.press/v54/erraqabi17a.html (cited on page 48).

Filippi, Sarah, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári (2010). "Parametric Bandits: The Generalized Linear Case". In: *Advances in Neural Information Processing Systems 23*. Edited by J D Lafferty, C K I Williams, J Shawe-Taylor, R S Zemel, and A Culotta. Curran Associates, Inc., pages 586–594. URL: http://papers.nips.cc/paper/4166-parametric-bandits-the-generalized-linear-case.pdf (cited on pages 43, 122).

Gabillon, Victor, Mohammad Ghavamzadeh, and Alessandro Lazaric (2012). "Best arm identification: A unified approach to fixed budget and fixed confidence". In: *Advances in Neural Information Processing Systems*. Volume 4, pages 3212–3220. ISBN: 9781627480031 (cited on page 48).

Garivier, Aurélien, Hédi Hadiji, Pierre Ménard, and Gilles Stoltz (2018). *KL-UCB-switch: Optimal regret bounds for stochastic bandits from both a distribution-dependent and a distribution-free viewpoints*. arXiv: 1805.05071 (cited on page 37).

Garivier, Aurélien and Emilie Kaufmann (2016). "Optimal Best Arm Identification with Fixed Confidence". In: *29th Annual Conference on Learning Theory*. Edited by Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir. Volume 49. Proceedings of Machine Learning Research. Columbia University, New York, New York, USA: PMLR, pages 998–1027. URL: http://proceedings.mlr.press/v49/garivier16a.html (cited on page 48).

Garivier, Aurélien, Pierre Ménard, Laurent Rossi, and Pierre Menard (Nov. 2017). "Thresholding Bandit for Dose-ranging: The Impact of Monotonicity". In: arXiv: 1711.04454. URL: http://arxiv.org/abs/1711.04454 (cited on page 48).

Garivier, Aurélien, Pierre Ménard, and Gilles Stoltz (2019). "Explore first, exploit next: The true shape of regret in bandit problems". In: *Mathematics of Operations Research* 44.2, pages 377–399. ISSN: 15265471. DOI: 10.1287/moor.2017.0928. arXiv: 1602.07182 (cited on page 138).

Garivier, Aurélien and Eric Moulines (2011). "On upper-confidence bound policies for switching bandit problems". In: *Proceedings of the 22nd International Conference on Algorithmic Learning Theory (ALT), 2011, Espoo, Finland.* Volume 6925 LNAI. Springer, Berlin, Heidelberg, pages 174–188. ISBN: 9783642244117. DOI: 10.1007/978-3-642-24412-4_16 (cited on pages 22, 43, 51, 58, 66, 74, 132–134).

Gottlieb, Jacqueline, Pierre-Yves Oudeyer, Manuel Lopes, and Adrien Baranes (2013). "Information-seeking, curiosity, and attention: computational and neural mechanisms". In: *Trends in Cognitive Sciences* 20, pages 1–9 (cited on page 50).

Hambleton, Ronald K and Hariharan Swaminathan (2013). *Item response theory: Principles and applications*. Springer Science & Business Media (cited on pages 122, 187).

Heidari, Hoda, Michael Kearns, and Aaron Roth (2016). "Tight Policy Regret Bounds for Improving and Decaying Bandits". In: *Proceedings of the International Joint*

Conference on Artificial Intelligence (IJCAI), pages 1562–1570 (cited on pages 17, 21, 42, 57, 61–63, 65, 66, 75, 125).

Howard, Ronald A (1960). "Dynamic programming and markov processes." In: (cited on page 44).

Immorlica, Nicole and Robert Kleinberg (Nov. 2018). "Recharging bandits". In: *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*. Volume 2018-Octob. IEEE Computer Society, pages 309–319. ISBN: 9781538642306. DOI: 10.1109/FOCS.2018.00037 (cited on pages 43, 57).

Jaksch, Thomas, Peter Auer, and Ronald Ortner (2009). "Near-optimal regret bounds for reinforcement learning". In: *Advances in neural information processing systems*, pages 89–96 (cited on page 44).

Jin, Tianyuan, Pan Xu, Jieming Shi, Xiaokui Xiao, and Quanquan Gu (Mar. 2020). "MOTS: Minimax Optimal Thompson Sampling". In: arXiv: 2003.01803. URL: http://arxiv.org/abs/2003.01803 (cited on page 38).

Käser, Tanja, Severin Klingler, and Markus Gross (2016). "When to stop? Towards universal instructional policies". In: *Proceedings of the sixth international conference on learning analytics & knowledge*, pages 289–298 (cited on pages 52, 53).

Kaufmann, Emilie, Olivier Cappe, and Aurelien Garivier (2012a). "On Bayesian Upper Confidence Bounds for Bandit Problems". In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*. Edited by Neil D Lawrence and Mark Girolami. Volume 22. Proceedings of Machine Learning Research. La Palma, Canary Islands: PMLR, pages 592–600. URL: http://proceedings.mlr.press/v22/kaufmann12.html (cited on pages 38, 105).

Kaufmann, Emilie, Olivier Cappé, and Aurélien Garivier (2016). "On the Complexity of Best-Arm Identification in Multi-Armed Bandit Models". In: *Journal of Machine Learning Research* 17.1, pages 1–42. ISSN: 1533-7928. URL: http://jmlr.org/papers/v17/kaufman16a.html (cited on page 48).

Kaufmann, Emilie, Nathaniel Korda, and Rémi Munos (Oct. 2012b). "Thompson sampling: An asymptotically optimal finite-time analysis". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Volume 7568 LNAI. Springer, Berlin, Heidelberg, pages 199–213. ISBN: 9783642341052. DOI: 10.1007/978-3-642-34106-9_18 (cited on page 38).

Khajah, Mohammad, Robert V. Lindsey, and Michael C. Mozer (2016). "How deep is knowledge tracing?" In: *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016*. arXiv: 1604.02416 (cited on page 186).

Kocsis, Levente and Csaba Szepesvári (2006). "Discounted ucb". In: *2nd PASCAL Challenges Workshop*. Volume 2 (cited on pages 51, 134).

Komiyama, Junpei and Tao Qin (2014). "Time-Decaying Bandits for Non-stationary Systems". In: *Web and Internet Economics (WINE)*. Edited by Tie-Yan Liu, Qi Qi, and Yinyu Ye. Cham: Springer International Publishing, pages 460–466. ISBN: 978-3-319-13129-0 (cited on page 57).

Lai, Tze Leung (1987). "Adaptive Treatment Allocation and the Multi-Armed Bandit Problem". In: *Annals of Statistics* 15.3, pages 1091–1114. ISSN: 0090-5364. DOI: 10.1214/AOS/1176350495. URL: https://projecteuclid.org/euclid.aos/1176350495 (cited on page 37).

Lai, Tze Leung and Herbert Robbins (Mar. 1985). "Asymptotically efficient adaptive allocation rules". In: *Advances in Applied Mathematics* 6.1, pages 4–22. ISSN: 10902074. DOI: 10.1016/0196-8858(85)90002-8. URL: https://www.sciencedirect.com/science/article/pii/0196885885900028 (cited on pages 35, 36, 74, 85, 86).

Lan, Andrew S and Richard G Baraniuk (2016). "A contextual bandits framework for personalized learning action selection". In: *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016*, pages 424–429. URL: https://pdfs.semanticscholar.org/a19e/4e14c424597df1d11f2cf99d09452b3da25b.pdf (cited on page 50).

Lattimore, Tor (Jan. 2018). "Refining the Confidence Level for Optimistic Bandit Strategies". In: *J. Mach. Learn. Res.* 19.1, pages 765–796. ISSN: 1532-4435 (cited on pages 37, 38, 105, 120, 134).

Lattimore, Tor and Csaba Szepesvari (2017). "The End of Optimism? An Asymptotic Analysis of Finite-Armed Linear Bandits". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Edited by Aarti Singh and Jerry Zhu. Volume 54. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, pages 728–737. URL: http://proceedings.mlr.press/v54/lattimore17a.html (cited on page 43).

Lattimore, Tor and Csaba Szepesvári (June 2020). *Bandit Algorithms*. Cambridge University Press UK. ISBN: 1108486827. URL: https://tor-lattimore.com/downloads/book/book.pdf (cited on pages 37, 87, 119, 133).

Lavenant, Hugo and Hassan Aït-Kaci (2019). *How can we assign a probability to an event?* (Cited on pages 180, 181).

Levine, Nir, Koby Crammer, and Shie Mannor (2017). "Rotting Bandits". In: *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 3074–3083. arXiv: 1702.07274 (cited on pages 17, 21, 42, 57, 61, 66–68, 70–72, 75, 76, 84, 85, 153).

Lindsey, Robert V, Michael C Mozer, William J Huggins, and Harold Pashler (2013). "Optimizing instructional policies". In: *Advances in neural information processing systems*, pages 2778–2786 (cited on page 50).

Liu, Fang, Joohyun Lee, and Ness Shroff (2018). "A Change-Detection Based Framework for Piecewise-Stationary Multi-Armed Bandit Problem". In: URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16939 (cited on pages 58, 135, 152, 153).

Liu, Yun-En, Travis Mandel, Emma Brunskill, and Zoran Popovic (2014). "Trading Off Scientific Knowledge and User Learning with Multi-Armed Bandits". In: *EDM* (cited on page 48).

Locatelli, Andrea, Maurilio Gutzeit, and Alexandra Carpentier (2016). "An optimal algorithm for the Thresholding Bandit Problem". In: *Proceedings of The 33rd International Conference on Machine Learning*. Edited by Maria Florina Balcan and Kilian Q Weinberger. Volume 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, pages 1690–1698. URL: http://proceedings.mlr.press/v48/locatelli16.html (cited on page 48).

Louëdec, Jonathan, Laurent Rossi, Max Chevalier, Aurélien Garivier, and Josiane Mothe (2016). "Algorithme de bandit et obsolescence : un modèle pour la recommandation (regular paper)". In: *Conférence francophone sur l'Apprentissage Automatique, Mar-*

seille, 05/07/2016-07/07/2016. http://www.lif.univ-mrs.fr: Laboratoire d'Informatique Fondamentale de Marseille, (en ligne). URL: http://www.irit.fr/publis/SIG/2016%7B%5C_%7DCAP%7B%5C_%7DLRCGM.pdf%20-%20http://oatao.univ-toulouse.fr/17130/ (cited on page 57).

Luckin, Rosemary (2001). "Designing children's software to ensure productive interactivity through collaboration in the Zone of Proximal Development (ZPD)". In: *Information Technology in Childhood Education Annual* 2001.1, pages 57–85 (cited on pages 50, 187).

Maillard, Odalric-Ambrym (2019). "Sequential change-point detection: Laplace concentration of scan statistics and non-asymptotic delay bounds". In: (cited on page 122).

McMahan, H. Brendan (2011). "Follow-the-regularized-leader and mirror descent: Equivalence theorems and L1 regularization". In: *Journal of Machine Learning Research*. Volume 15, pages 525–533 (cited on page 42).

Melesko, Jaroslav and Vitalij Novickij (2019). "Computer Adaptive Testing Using Upper-Confidence Bound Algorithm for Formative Assessment". In: *Applied Sciences* 9.20, page 4303 (cited on page 53).

Ménard, Pierre and Aurélien Garivier (2017). "A minimax and asymptotically optimal algorithm for stochastic bandits". In: *Proceedings of the 28th International Conference on Algorithmic Learning Theory*. Edited by Steve Hanneke and Lev Reyzin. Volume 76. Proceedings of Machine Learning Research. Kyoto University, Kyoto, Japan: PMLR, pages 223–237. URL: http://proceedings.mlr.press/v76/m%7B%5C'%7Be%7D%7Dnard17a.html (cited on pages 37, 119, 120, 134).

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller (2013). "Playing atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602* (cited on page 45).

Mu, Tong, Shuhan Wang, Erik Andersen, and Emma Brunskill (2018). "Combining adaptivity with progression ordering for intelligent tutoring systems". In: *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, pages 1–4 (cited on page 52).

Mukherjee, Subhojyoti and Odalric-Ambrym Maillard (May 2019). "Distribution-dependent and Time-uniform Bounds for Piecewise i.i.d Bandits". In: arXiv: 1905.13159. URL: http://arxiv.org/abs/1905.13159 (cited on pages 135, 142).

Mukherjee, Subhojyoti, Naveen Kolar Purushothama, Nandan Sudarsanam, and Balaraman Ravindran (2017). "Thresholding Bandits with Augmented UCB". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. IJCAI'17. AAAI Press, pages 2515–2521. ISBN: 9780999241103 (cited on page 48).

Neu, Gergely (2015). "Explore no more: Improved high-probability regret bounds for non-stochastic bandits". In: *Advances in Neural Information Processing Systems*, pages 3168–3176 (cited on page 51).

Papadimitriou, Christos H and John N Tsitsiklis (1994). "The complexity of optimal queueing network control". In: *Proceedings of IEEE 9th Annual Conference on Structure in Complexity Theory*. IEEE, pages 318–322 (cited on page 43).

Piech, Chris, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas Guibas, and Jascha Sohl-Dickstein (2015). "Deep knowledge tracing". In: *Advances in Neural Information Processing Systems*. arXiv: 1506.05908 (cited on page 186).

Pike-Burke, Ciara (2019). "Sequential decision problems in online education". PhD thesis. Lancaster University. DOI: 10.17635/lancaster/thesis/604 (cited on pages 34, 52).

Pike-Burke, Ciara and Steffen Grunewalder (2019). "Recovering Bandits". In: *Advances in Neural Information Processing Systems 32*. Edited by H. Wallach and H. Larochelle and A. Beygelzimer and F. d'Alché-Buc and E. Fox and R. Garnett. Curran Associates, Inc., pages 14122–14131. URL: http://papers.nips.cc/paper/9561-recovering-bandits.pdf (cited on pages 43, 57).

Rafferty, Anna N, Emma Brunskill, Thomas L Griffiths, and Patrick Shafto (2016). "Faster teaching via pomdp planning". In: *Cognitive science* 40.6, pages 1290–1332 (cited on page 54).

Rollinson, Joseph and Emma Brunskill (2015). "From Predictive Models to Instructional Policies." In: *International Educational Data Mining Society* (cited on page 52).

Russac, Yoan, Claire Vernade, and Olivier Cappé (2019). "Weighted Linear Bandits for Non-Stationary Environments". In: *Advances in Neural Information Processing Systems 32*. Edited by H Wallach, H Larochelle, A Beygelzimer, F d\textquotesingle Alché-Buc, E Fox, and R Garnett. Curran Associates, Inc., pages 12040–12049. URL: http://papers.nips.cc/paper/9372-weighted-linear-bandits-for-non-stationary-environments.pdf (cited on pages 58, 136).

Seznec, Julien, Andrea Locatelli, Alexandra Carpentier, Alessandro Lazaric, and Michal Valko (2019). "Rotting bandits are no harder than stochastic ones". In: *Proceedings of Machine Learning Research, The 22nd International Conference on Artificial Intelligence and Statistics, 16-18 April 2019*. Edited by Kamalika Chaudhuri and Masashi Sugiyama. Volume 89. Proceedings of Machine Learning Research. PMLR, pages 2564–2572. URL: http://proceedings.mlr.press/v89/seznec19a.html (cited on pages 57, 153).

Seznec, Julien, Pierre Menard, Alessandro Lazaric, and Michal Valko (2020). "A single algorithm for both restless and rested rotting bandits". In: *International Conference on Artificial Intelligence and Statistics*, pages 3784–3794 (cited on pages 57, 58).

Shani, Guy, Ronen I Brafman, and Solomon E Shimony (2005). "Model-Based Online Learning of POMDPs". In: *Machine Learning: ECML 2005*. Edited by João Gama, Rui Camacho, Pavel B Brazdil, Alípio Mário Jorge, and Luís Torgo. Berlin, Heidelberg: Springer Berlin Heidelberg, pages 353–364. ISBN: 978-3-540-31692-3 (cited on page 181).

Silver, David, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. (2016). "Mastering the game of Go with deep neural networks and tree search". In: *nature* 529.7587, pages 484–489 (cited on page 46).

Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. (2017). "Mastering the game of go without human knowledge". In: *nature* 550.7676, pages 354–359 (cited on page 46).

Sutton, Richard S and Andrew G Barto (2018). *Reinforcement Learning: An Introduction*. Second. The MIT Press. URL: http://incompleteideas.net/book/the-book-2nd.html (cited on page 44).

Teng, S, J Li, L P Ting, K Chuang, and H Liu (Nov. 2018). "Interactive Unknowns Recommendation in E-Learning Systems". In: *2018 IEEE International Conference on Data Mining (ICDM)*, pages 497–506. DOI: 10.1109/ICDM.2018.00065 (cited on page 53).

Tewari, Ambuj and Susan A Murphy (2017). "From Ads to Interventions: Contextual Bandits in Mobile Health". In: *Mobile Health: Sensors, Analytic Methods, and Applications*. Edited by James M Rehg, Susan A Murphy, and Santosh Kumar. Cham: Springer International Publishing, pages 495–517. ISBN: 978-3-319-51394-2. DOI: 10.1007/978-3-319-51394-2_25. URL: https://doi.org/10.1007/978-3-319-51394-2%7B%5C_%7D25 (cited on page 43).

Thompson, William R (1933). "On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples". In: *Biometrika* 25.3/4, pages 285–294. ISSN: 00063444. DOI: 10.1093/biomet/25.3-4.285. URL: http://www.jstor.org/stable/2332286 (cited on pages 33, 38, 42).

Tracà, Stefano and Cynthia Rudin (May 2015). "Regulating Greed Over Time". In: arXiv: 1505.05629. URL: http://arxiv.org/abs/1505.05629 (cited on pages 34, 42, 152).

Tsallis, Constantino (July 1988). "Possible generalization of Boltzmann-Gibbs statistics". In: *Journal of Statistical Physics* 52.1-2, pages 479–487. ISSN: 00224715. DOI: 10.1007/BF01016429 (cited on page 42).

Valko, Michal, Nathaniel Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini (2013). "Finite-time analysis of kernelised contextual bandits". In: *arXiv preprint arXiv:1309.6869* (cited on page 43).

Valko, Michal, Rémi Munos, Branislav Kveton, and Tomáš Kocák (2014). "Spectral bandits for smooth graph functions". In: *International Conference on Machine Learning*, pages 46–54 (cited on page 43).

Ventola, C Lee (2015a). "The antibiotic resistance crisis: part 1: causes and threats". In: *Pharmacy and therapeutics* 40.4, page 277 (cited on page 57).

— (2015b). "The antibiotic resistance crisis: part 2: management strategies and new agents". In: *Pharmacy and Therapeutics* 40.5, page 344 (cited on page 57).

Villar, Sofía S., Jack Bowden, and James Wason (2015). "Multi-armed bandit models for the optimal design of clinical trials: Benefits and challenges". In: *Statistical Science* 30.2, pages 199–215. ISSN: 08834237. DOI: 10.1214/14-STS504 (cited on page 34).

Vinyals, Oriol, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojciech M Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, et al. (2019). "Alphastar: Mastering the real-time strategy game starcraft II". In: *DeepMind blog*, page 2 (cited on page 46).

Warlop, Romain, Alessandro Lazaric, and Jérémie Mary (2018). "Fighting Boredom in Recommender Systems with Linear Reinforcement Learning". In: *Advances in Neural Information Processing Systems 31*. Edited by S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett. Curran Associates, Inc., pages 1757–1768. URL: http://papers.nips.cc/paper/7447-fighting-boredom-in-recommender-systems-with-linear-reinforcement-learning.pdf (cited on page 57).

Watkins, Christopher J. C. H. (1989). "Learning from delayed rewards". In: (cited on page 45).

Watkins, Christopher J. C. H. and Peter Dayan (1992). “Q-learning”. In: *Machine learning* 8.3-4, pages 279–292 (cited on page 45).

Whittle, P. (1980). “Multi-Armed Bandits and the Gittins Index”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 42, pages 143–149. DOI: 10.2307/2984953. URL: https://www.jstor.org/stable/2984953 (cited on page 42).

— (1988). “Restless bandits: activity allocation in a changing world”. In: *Journal of Applied Probability* 25.A, pages 287–298. ISSN: 0021-9002. DOI: 10.2307/3214163. URL: https://www.jstor.org/stable/3214163 (cited on pages 42, 43).

Wilson, Kevin H., Yan Karklin, Bojian Han, and Chaitanya Ekanadham (Apr. 2016). “Back to the Basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation”. In: arXiv: 1604.02336. URL: http://arxiv.org/abs/1604.02336 (cited on pages 186, 187).

Xiong, Xiaolu, Siyuan Zhao, Eric G. Van Inwegen, and Joseph E. Beck (2016). “Going deeper with deep knowledge tracing”. In: *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016* (cited on page 186).

Xu, Jie, Tianwei Xing, and Mihaela Van Der Schaar (2016). “Personalized course sequence recommendations”. In: *IEEE Transactions on Signal Processing* 64.20, pages 5340–5352 (cited on page 50).

Yarats, Denis, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus (2019). “Improving sample efficiency in model-free reinforcement learning from images”. In: *arXiv preprint arXiv:1910.01741* (cited on page 46).

Yu, Yang (2018). “Towards Sample Efficient Reinforcement Learning.” In: *IJCAI*, pages 5739–5743 (cited on page 46).

Zimmert, Julian and Yevgeny Seldin (July 2018). “Tsallis-INF: An Optimal Algorithm for Stochastic and Adversarial Bandits”. In: arXiv: 1807.07623. URL: http://arxiv.org/abs/1807.07623 (cited on pages 39, 40, 42).