Thèse de doctorat

Dynamics and learning in online allocation problems

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à l'École nationale de la statistique et de l'administration
économique

École doctorale n°574 École doctorale de mathématiques
Hadamard (EDMH)
Spécialité de doctorat: Mathématiques appliquées

Thèse présentée et soutenue à Palaiseau, le 11/12/2025, par

Maria Cherifa

Composition du Jury :

Arnak Dalalyan
Professeur, ENSAE, CREST   Président du jury

Pascal Moyal
Professeur des Universités, Université de Lorraine
(Institut Élie Cartan)   Rapporteur

Bruno Gaujal
Directeur de recherche (INRIA),Université Grenoble
Alpes (Laboratoire d'informatique LIG)   Rapporteur

Matthieu Jonckheere
Directeur de recherche CNRS, LAAS CNRS   Examinateur

Vianney Perchet
Professeur, CREST, ENSAE   Directeur de thèse

Clément Calauzènes
Researcher, Criteo AI Lab   Co-directeur de thèse

Abstract

The thesis analyzes how decision-making systems evolve, learn, and adapt in uncertain environments. It focuses on matching problems on bipartite graphs, which model real-time decision-making scenarios such as organ allocation, appointment scheduling, volunteer coordination, ride-sharing, and online advertising. Particular attention is given to the latter, where matchings between users and advertisers are carried out through large-scale automated auctions. The work lies at the intersection of online matching and the multi-armed bandit problem. The first concerns sequential, irrevocable decisions, while the second captures the trade-off between exploration and exploitation in uncertain environments. Combining these perspectives provides a framework for understanding situations in which digital systems must learn and adapt simultaneously. The first contribution introduces an online matching model with budget refills, inspired by advertising settings where advertisers' resources can be replenished over time. Two settings—adversarial and stochastic—are analyzed, and theoretical bounds are established on the efficiency of standard algorithms, highlighting the impact of refill frequency and structure on competitiveness. The second contribution studies online matching on graphs generated by a Stochastic Block Model, representing heterogeneous communities of users or advertisers. When compatibility probabilities are unknown, the problem becomes a bandit setting. An algorithm combining Explore-Then-Commit and Balance is proposed to estimate these probabilities and maximize expected matching size, with theoretical guarantees on convergence and competitiveness. The final contribution addresses the optimization of collective gain in online advertising auctions through a structured bandit approach. Each decision selects a coalition of advertisers participating in a second-price auction. Leveraging the structure of the reward function, the thesis proposes greedy algorithms that balance exploration and exploitation. Regret analysis and numerical experiments demonstrate their efficiency and robustness.

Keywords: online matching, bipartite graphs, random graphs, stochastic approximations, multi-arm bandit algorithms.

Résumé

La thèse étudie la manière dont les systèmes décisionnels évoluent, apprennent et s'adaptent dans des environnements incertains. Elle se concentre sur les problèmes d'appariement sur graphes bipartis, qui modélisent des décisions prises en temps réel sans connaissance complète de l'avenir, comme dans l'attribution d'organes, la gestion de rendez-vous, la coordination de bénévoles, la mobilité ou la publicité en ligne. Une attention particulière est portée à ce dernier domaine, où les appariements entre utilisateurs et annonceurs sont réalisés par des enchères automatiques de grande échelle. Le travail se situe à l'interface du matching en ligne et du problème du bandit multi-bras : le premier décrit des décisions séquentielles irréversibles, tandis que le second formalise le compromis entre exploration et exploitation dans l'apprentissage sous incertitude. Leur combinaison permet d'analyser des situations réalistes dans lesquelles les systèmes numériques doivent simultanément apprendre et s'adapter dans un environnement dynamique et partiellement observé.La première contribution introduit un modèle d'appariement en ligne avec recharge de budgets, inspiré des mécanismes publicitaires où les ressources des annonceurs peuvent être renouvelées. Deux cadres — adversarial et stochastique — sont étudiés, et des bornes théoriques sont établies sur l'efficacité d'algorithmes classiques, montrant l'impact de la fréquence et de la structure des rechargements. La seconde contribution analyse l'appariement en ligne sur des graphes issus d'un modèle à blocs stochastiques, reflétant la présence de communautés d'utilisateurs ou d'annonceurs. Lorsque les probabilités de compatibilité sont inconnues, le problème devient un bandit : un algorithme combinant Explore-Then-Commit et Balance est proposé pour estimer ces probabilités et maximiser la taille attendue du matching, avec des garanties de convergence et de compétitivité. La troisième contribution porte sur l'optimisation du gain collectif dans les enchères publicitaires en ligne via des bandits structurés. Chaque décision sélectionne une coalition d'annonceurs dans une enchère au second prix. En exploitant la structure de la récompense, des algorithmes gloutons sont proposés pour équilibrer exploration et exploitation. L'analyse du regret, soutenue par des bornes de concentration et des expériences numériques, montre leur efficacité et leur robustesse.

Mots-clés : appariement en ligne, graphes bipartis, graphes aléatoires, approximations stochastiques, algorithmes de bandits multi-bras.

Acknowledgments (Remerciements)

Cette thèse est le fruit de plusieurs années traversées d'élans, de doutes, de découvertes, de découragement parfois, de petites victoires qui redonnent confiance, et surtout de rencontres lumineuses. Aucun mot ne pourra vraiment traduire ce que ces années ont représenté pour moi, mais je veux, avec sincérité, dire merci à celles et ceux qui ont marqué ce chemin. Rien de ce que j'ai accompli n'aurait été possible sans vous.

À mes directeurs de thèse, je dois bien plus que des remerciements. Clément, tu as été mon premier repère scientifique tout au long de la thèse, celui qui m'a accueillie chez Criteo avec une simplicité et une chaleur qui ont immédiatement dissipé mes appréhensions. Merci pour la liberté que tu m'as laissée dès le premier jour, et pour cette façon que tu avais de sentir immédiatement quand quelque chose n'était pas clair pour moi. Merci d'avoir eu cette patience constante, de m'avoir appris à être rigoureuse, à clarifier une idée, à écrire proprement une preuve, à présenter un résultat avec assurance. Tu ne t'es jamais lassé d'expliquer, et tu l'as fait sans jamais me faire sentir que je posais trop de questions. Merci aussi pour ces moments où tu m'as poussée, parfois sans même le vouloir : une seule de tes questions pouvait suffire à me faire réfléchir pendant des jours, et me faire évoluer autant. Rien de tout cela n'aurait été possible sans toi. Vianney, merci pour ton accueil dans ton équipe, pour ta disponibilité même quand ton emploi du temps débordait, pour ta franchise qui pouvait parfois me faire peur mais qui m'a tant fait avancer, et pour cette manière unique que tu as d'alléger chaque réunion, de rendre un problème difficile soudain un peu plus simple. Vous formez, toi et Clément, une très bonne équipe.

Je tiens également à remercier chaleureusement les membres de mon jury. Je vous suis profondément reconnaissante pour le temps consacré à la relecture de mon manuscrit, ainsi que pour l'attention et le soin apportés à l'évaluation de mes travaux. Pascal, merci pour l'intérêt porté à mes recherches depuis nos échanges au CIRM. Merci également d'avoir accepté la tâche de rapporteur, pour la qualité du rapport, la finesse des remarques, la pertinence des observations, ainsi que pour des compliments particulièrement encourageants. Bruno, merci d'avoir accepté de rapporter cette thèse, et pour un rapport à la fois clair, approfondi et constructif, accompagné de commentaires très motivants. Matthieu, merci d'avoir accepté d'être examinateur malgré une demande formulée en toute dernière minute, et d'avoir répondu présent avec autant de disponibilité et de bienveillance. Enfin, Arnak, merci d'avoir accepté d'être examinateur et président du jury. Merci également de m'avoir fait découvrir l'Arménie lors de la summer school de 2023 — un pays qui m'a profondément marquée et que je continue de recommander autour de moi. Merci aussi pour ton accueil chaleureux au CREST, pour ta gentillesse, ton sourire constant et ta réactivité, même face à mes demandes les plus urgentes. J'espère sincèrement avoir l'occasion de revenir bientôt en Arménie, peut-être à l'occasion d'une prochaine conférence.

Au MAP5, ce laboratoire qui a été bien plus qu'un simple lieu de travail pour moi, je souhaite adresser quelques mots très particuliers à celles et ceux qui l'ont rendu si précieux. Antoine C., merci pour ton accueil chaleureux, ta flexibilité et la confiance que tu m'as accordée en me permettant de rejoindre le laboratoire malgré un contrat un peu atypique. Ton sourire constant et ton énergie communicative contribuent énormément à l'atmosphère du MAP5. Merci aussi pour ton aide — y compris lors de mes péripéties administratives — et pour ta disponibilité. J'ai adoré collaborer avec toi en TD de statistiques pour les L2. Georges K., merci pour ta gentillesse, ta souplesse et la facilité avec laquelle tu m'as toujours mise à l'aise dans mes choix de TD. Merci également pour ton aide lors de mes soucis administratifs. Aux éphémères du labo, et en particulier aux membres du bureau 725 C1, merci pour votre accueil si chaleureux. Éloi, merci pour ta gentillesse et ton sourire dès le premier jour ; avec Adélie, tu fais clairement partie des personnes qui portent le 725 C1 et, plus largement, le MAP5. Merci pour nos discussions — mathématiques ou non — et pour les restaurants chinois que tu m'as fait découvrir avec Xuwen : vous formez un couple adorable. Merci aussi de m'avoir accueillie chez toi pendant les périodes intenses de rédaction ; tu as rendu mes journées au MAP5 tellement plus légères. Alex, thank you for your constant smile, for all our coffee and bubble tea breaks during the writing period, and for being such a comforting presence during the most stressful moments of the PhD. Thank you as well for the gym sessions — hopefully I'll manage to go more often and not only run! And good luck for your defense scheduled just one day after mine! clearly, our PhDs agreed to cross the finish line together. Adélie, merci pour ton accueil dès le premier jour, même si tu es ensuite partie en Suède. L'énergie que tu apportes au labo est incroyable, et j'espère qu'on pourra très bientôt recourir ensemble. Lucie, merci pour ton accueil, pour la Parisienne que nous avons courue ensemble, et pour toutes nos petites discussions. Elles m'ont apporté énormément de soutien dans les moments les plus difficiles de la thèse. Rayane, mon voisin de bureau, merci pour ta bonne humeur, tes rires et tes taquineries avec Thomas. Vous rendez le 725 C1 hilarant. Guillaume, merci pour ta bienveillance et pour toutes nos discussions sur la recherche et ses défis. Thomas, même si tu ne fais pas officiellement partie du 725 C1, tu en es clairement un membre de cœur. Merci pour ton sourire constant, ta sympathie et vos chamailleries avec Rayane qui égayaient nos après-midi. Et bon courage pour ta soutenance, prévue à peine un jour après la mienne — on ne pouvait pas faire plus synchronisé ! Sylvain, merci pour la compote de poire, pour nos échanges matinaux, et pour m'avoir si souvent prêté ton badge — que j'oubliais, malheureusement, beaucoup trop souvent. Bernardin, même si nous nous sommes rencontrés tardivement, j'ai beaucoup apprécié ta sympathie et nos discussions. Je passe maintenant au bureau 725 A1. Ivan, figure incontournable du labo et du 725 A1, merci d'avoir essayé de m'apprendre le tarot, pour ta gentillesse, ton sourire, et pour tous les moments passés chez Éloi pendant la rédaction. Profite bien de ton séjour en Argentine, puis en Autriche ! Eyal, même si tu as fait le bon move en déménageant au 8 étage, dans ma tête tu restes toujours associé au 725 A1. Merci pour ton sourire constant, et ton ouverture d'esprit. Merci aussi pour tes idées lumineuses pour les interrogations de MC2. Sagbo, merci pour ton sourire à chacune de nos rencontres et nos discussions toujours agréables. Oumayma, je garde un excellent souvenir de notre collaboration

en TD de MC2 ; même si nous avons manqué de temps pour mieux nous connaître, j'ai beaucoup apprécié nos petits moments partagés. Et je termine avec l'un des plus beaux bureaux de doctorants du MAP5 : celui du 8 étage. Clémence, merci pour ta positivité, tes compliments et ta générosité ; tu proposais toujours ton aide spontanément. Merci pour ta relecture minutieuse de mes slides et pour tes retours constructifs. J'espère que nous organiserons bientôt d'autres séances de running avec les runners du MAP5. Beatriz, merci pour ton énergie chaleureuse, ta délicatesse et ta gentillesse. Nos discussions m'ont souvent redonné un vrai coup de boost pendant mes journées au MAP5. J'espère rejoindre ton running club très bientôt. Perla, merci pour ton sourire constant et ton enthousiasme ; même si nous n'avons pas souvent eu l'occasion d'échanger, chacune de nos discussions a été un plaisir. Enfin, aux anciens stagiaires devenus pour la plupart doctorants — Raphaël, César, Pierre, Tess, et sûrement d'autres — je vous souhaite beaucoup de courage et de très belles réussites : je suis certaine que vous allez tout déchirer. Et pour celles et ceux que je n'ai pas mentionnés, je suis désolée : sachez que vous êtes formidables.

A particular thank you goes to Gayane. We first met during the summer school in Armenia and crossed paths again later in Marseille, sharing a room that quickly became the starting point of a genuine friendship. Thank you for your kindness, for always checking in on me, and for offering your help so naturally. Now that we live in the same city, we truly should make an effort to see each other more often.

Je voudrais aussi remercier Christophe Vignat. Je n'oublierai jamais tes cours d'optimisation en L3 : ils ont marqué mon parcours, et je suis très heureuse que nous ayons gardé le contact depuis. Merci pour les stages que j'ai pu obtenir grâce à tes recommandations, pour les nombreuses lettres que tu as écrites en ma faveur, pour tes messages de nouvelles et pour tes invitations à venir présenter mon parcours à tes étudiants de M2. Je suis vraiment reconnaissante que nos échanges aient perduré toutes ces années.

Cette thèse n'aurait jamais vu le jour sans le soutien et l'affection indéfectibles de ma famille. À ma chère maman, je voudrais adresser les mots les plus tendres. Merci pour ton soutien constant, pour tes appels quotidiens, ta douceur, ta bienveillance, ton sourire qui apaise tout, et cette capacité unique que tu as de rendre chaque difficulté plus légère. Je ne serais pas la personne que je suis aujourd'hui sans toi. Tu es, et tu resteras toujours, mon pilier, la personne la plus importante à mes yeux. Ta force tranquille, ton courage, ton intelligence, ta générosité et ce regard plein d'amour m'ont guidée à chaque étape. Et, objectivement, tu es l'une des plus belles personnes que je connaisse. J'espère pouvoir, un jour, te rendre ne serait-ce qu'une part de la lumière que tu m'as donnée. À mon père, parti trop tôt mais jamais absent : depuis 2016, tu m'accompagnes autrement. Tes rires, tes blagues, ta légèreté, tes encouragements, et surtout ton amour pour les études et la science sont restés ancrés en moi. C'est toi qui as nourri mon désir d'apprendre, de comprendre et d'avancer. Je te dédie cette thèse. J'espère qu'elle te rendrait fier. À ma grand-mère, dont les cours de littérature arabe et la présence constante ont accompagné toute mon enfance : tu as profondément influencé la personne que je suis aujourd'hui. À ma tante Faiza et à son mari, merci pour votre soutien constant et nos discussions

toujours si enrichissantes. Merci, Faiza, pour les cours de français de mon enfance, pour la culture générale que tu m'as transmise et pour l'ouverture d'esprit que tu as contribué à façonner en moi. Une grande part de ma curiosité et de mon rapport au savoir vient directement de toi. Tonton Momo, merci pour ton soutien, tes conseils — notamment dans le domaine de la recherche — et pour m'avoir appris à rédiger mes premières lettres. Ton aide m'a accompagnée bien plus loin que tu ne le crois. À ma cousine Sarah, merci pour ta disponibilité, ton soutien et ta présence à chaque fois que j'avais besoin d'aide. Je suis tellement heureuse de t'avoir dans ma vie. Je remercie aussi ma tante Wassila et toute sa famille pour les moments formidables partagés ensemble. Merci, Wassila, pour tes messages réguliers pour prendre de mes nouvelles : ils comptaient bien plus que tu ne peux l'imaginer. Enfin, une pensée reconnaissante pour mes oncles — Chaouki, Salim, Amin et Redouane — pour leur présence, leur gentillesse et leur bienveillance, chacun à leur manière.

Et puis, il y a mes amis. Medina, depuis l'école primaire jusqu'à Paris : tu es l'une des plus belles constantes de ma vie. Merci pour ton écoute, nos sorties, nos voyages, et cette présence réconfortante qui m'accompagne depuis toujours et continue de m'apporter tant de bien au quotidien. Sylia et Yasmine, nous avons traversé ensemble tant d'étapes : des éclats de rire, des voyages, des moments de joie immense et d'autres plus difficiles. Je suis profondément reconnaissante de vous avoir dans ma vie, et encore plus heureuse de constater que les années n'ont fait que renforcer notre lien. Sabrina, rencontrée en pleine thèse grâce à Walid : je suis tellement heureuse que nos chemins se soient croisés. En si peu de temps, nous avons partagé une quantité incroyable de souvenirs — voyages, rires, sorties, courses, et mille petits instants qui ont illuminé ces années. J'espère de tout cœur que cela continuera longtemps. Enfin, à ceux qui ont traversé ma vie un jour, et qui se reconnaîtront en lisant ces lignes : même si nos chemins se sont séparés, parfois dans la douleur, parfois sans que je m'y attende, merci d'avoir été présents, ne serait-ce qu'un temps. Votre passage a compté.

Contents

1

Introduction (en français)

Le problème d'appariement (matching) n'est pas une idée récente. Bien avant les algorithmes, ou même l'existence des ordinateurs, les sociétés humaines avaient déjà développé leurs propres méthodes pour relier les besoins aux ressources, les demandeurs aux fournisseurs et les individus aux opportunités. Dans les anciens systèmes d'apprentissage, les anciens choisissaient les jeunes à former en s'appuyant sur leur intuition, leur lignée ou leur potentiel. Dans les communautés traditionnelles, les entremetteurs arrangeaient les mariages en équilibrant soigneusement le statut social, la compatibilité et les intérêts familiaux. Dans les villages, les enfants étaient confiés à des enseignants à travers des décisions communautaires, souvent influencées par l'ordre d'arrivée ou l'urgence des besoins. Ces formes d'appariement étaient locales, personnelles, et exigeaient beaucoup de temps. Elles reposaient en grande partie sur la connaissance, la mémoire et les relations humaines. L'entremetteur, l'enseignant, le maître d'apprentissage : ils n'étaient pas de simples participants au système, **ils étaient le système**. Les décisions étaient prises avec soin, laissant place à la négociation, aux hésitations et au jugement humain. Si une meilleure solution apparaissait plus tard, la communauté pouvait réajuster ses choix.

Aujourd'hui, l'appariement est devenu presque invisible, mais il n'en est pas moins essentiel. Chaque fois que nous ouvrons une application, recherchons un produit, commandons un trajet, faisons défiler un profil sur une application de rencontre ou postulons à un emploi, un système numérique tente de nous apparier avec quelque chose ou quelqu'un. Cela se fait à une vitesse imperceptible et à une échelle difficile à concevoir. Ces systèmes opèrent à l'échelle mondiale, en continu, et presque toujours de manière automatique. Mais ce qui a le plus changé, ce n'est pas seulement la vitesse ou l'échelle : c'est le moment de la décision. Dans les systèmes numériques modernes, les personnes et les ressources n'arrivent plus de manière ordonnée, ni en attendant leur tour. Elles apparaissent de façon imprévisible, en temps réel. Et les décisions ne peuvent plus être repoussées en attendant d'avoir une vision complète.

Il n'y a pas de bouton « pause » pour collecter d'abord toutes les données, puis faire son choix par la suite. Les appariements doivent se faire immédiatement, à partir des informations partielles disponibles à cet instant. Il est impossible de savoir quel utilisateur demandera un trajet dans les cinq secondes à venir, ni si un donneur de rein plus compatible se manifestera demain. Le système doit choisir dans l'instant, ou laisser passer l'opportunité. Et surtout, une fois l'appariement effectué, il est souvent **définitif**. Un chauffeur démarre. Une publicité est diffusée. Un poste est pourvu. Une place dans un cours est occupée. L'avenir est dès lors déterminé par les choix du présent. **C'est cette irréversibilité, combinée à l'incertitude** sur ce qui peut encore survenir, qui rend l'appariement en ligne fondamentalement différent — et infiniment plus complexe — des formes traditionnelles d'appariement qui l'ont précédé. Pourtant, malgré cette complexité, l'appariement en ligne est devenu une pierre angulaire discrète de la vie moderne. Il alimente les services que nous utilisons au quotidien, régule l'accès à des ressources limitées, et détermine souvent qui obtient quoi, quand et comment. Concevoir de meilleurs systèmes d'appariement en ligne n'est donc pas seulement une affaire d'algorithmes : c'est une manière de façonner l'équité, l'efficacité et les opportunités dans une société numérique.

Pour mieux comprendre l'importance et la diversité de ces systèmes d'appariement modernes, il est utile d'observer comment ils interviennent dans des domaines très variés de notre vie. Qu'il s'agisse de la santé publique, de l'éducation, de l'emploi, du logement, de la mobilité ou même des relations humaines, ces systèmes sont omniprésents. Dans ce qui suit, nous présentons plusieurs exemples concrets, issus de contextes très différents mais unis par une même logique : celle de prendre des décisions d'appariement en temps réel, avec des informations partielles, et souvent sans retour en arrière possible. Ces contextes d'application illustrent, chacun à sa manière, comment ces décisions invisibles façonnent en profondeur l'accès aux ressources, l'organisation des services, et même certaines dimensions de la vie sociale.

1.1   Dons d'organe : un appariement pour sauver des vies

Le don d'organes est un acte de solidarité vitale, qui permet de sauver ou d'améliorer la vie de milliers de patients dans chaque pays. On distingue principalement deux types de dons : le don post-mortem, lorsqu'un organe est prélevé sur une personne décédée (avec son consentement ou celui de ses proches), pour être attribué à un patient en attente ; et le don de son vivant, plus rare, qui concerne essentiellement le rein ou une partie du foie. Dans les deux cas, le facteur temps est décisif : une fois un organe disponible, il faut agir vite pour identifier un receveur compatible et organiser la transplantation, car chaque greffe est une opportunité souvent unique pour un patient.

Dans le cas des greffes de rein, il arrive fréquemment qu'un patient ait un proche

1

volontaire pour faire don d'un rein. Mais il est possible que cette personne ne soit pas biologiquement compatible. Plutôt que de renoncer, les hôpitaux peuvent alors proposer un échange croisé : si un autre couple patient-donneur est dans la même situation, mais avec une compatibilité croisée, un échange devient possible. Ce principe peut s'étendre à plusieurs couples pour former une chaîne de dons. Dans certains cas, cette chaîne débute grâce à un donneur altruiste, une personne qui donne un rein sans destinataire désigné. Cela déclenche ce qu'on appelle, dans le milieu médical, une « greffe domino » : son rein est attribué à un premier patient, dont le donneur initial devient disponible pour une autre personne, et ainsi de suite. Une seule décision peut ainsi permettre de débloquer plusieurs greffes successives.

Ce système repose sur une logistique complexe. De nouvelles paires patient-donneur s'inscrivent au fil du temps. Certaines sont immédiatement compatibles avec des chaînes en construction, d'autres doivent attendre plusieurs semaines, voire plusieurs mois. Un dilemme se pose à chaque instant : faut-il saisir une opportunité de greffe dès qu'une compatibilité est identifiée, même si cela ne concerne qu'un petit nombre de patients ? Ou faut-il attendre, dans l'espoir que d'autres paires rejoignent le système et permettent d'allonger la chaîne, augmentant ainsi le nombre total de greffes ? Agir trop tôt, c'est potentiellement sacrifier une opportunité plus large ; attendre trop longtemps, c'est risquer la dégradation de l'état de santé des patients ou le désistement de certains donneurs, ce qui pourrait annuler toute la chaîne envisagée. Chaque greffe réussie repose donc sur un fragile équilibre entre urgence médicale et espoir de configurations futures. Ce processus met en lumière une réalité essentielle : les décisions sont prises au fur et à mesure que les possibilités apparaissent, avec une connaissance partielle de l'avenir, et rarement la possibilité de revenir en arrière. Une fois qu'un donneur est engagé dans une greffe, il ne peut plus participer à une autre. Chaque correspondance modifie les options suivantes. Ces décisions, souvent invisibles pour le grand public, sont pourtant cruciales. Leur enchaînement détermine la capacité du système à sauver des vies. C'est précisément ce caractère séquentiel, irréversible et incertain des décisions — prises une par une, au fil du temps — qui rend la gestion du don d'organes si complexe et délicate dans le milieu hospitalier.

1.2 Systèmes de rendez-vous médicaux : qui obtient le créneau ?

Dans les hôpitaux comme dans les cabinets médicaux, la prise de rendez-vous se fait de plus en plus via des plateformes en ligne, telles que Doctolib. Certains patients y réservent des consultations de routine, d'autres recherchent un créneau en urgence. Chaque jour, de nouveaux patients rejoignent la plateforme, à des moments imprévisibles et avec des besoins très variés. Le système doit donc attribuer les créneaux disponibles au fur et à mesure, sans connaître à l'avance tous les profils ni l'ensemble des demandes à venir. Faut-il remplir le planning dès que possible pour éviter que les médecins restent inactifs, ou conserver des créneaux pour d'éventuelles

urgences ? Que faire lorsqu'un patient ne se présente pas à son rendez-vous ? Peut-on proposer sa place à quelqu'un d'autre à la dernière minute ? Et comment garantir une certaine équité, notamment pour les personnes âgées, précaires ou peu familières avec le numérique, qui risquent d'être défavorisées par ce type de système ? Dans ce contexte, chaque décision d'attribution a un impact direct sur l'accès aux soins. Un système mal conçu peut entraîner des retards de prise en charge, des créneaux perdus ou une saturation des services d'urgence. Ce qui rend le problème particulièrement complexe, c'est que toutes ces décisions doivent être prises en temps réel, avec une information partielle et, bien souvent, sans possibilité de revenir en arrière.

1.3 Coordination des bénévoles : aider là où le besoin est le plus urgent

Dans des contextes humanitaires — qu'il s'agisse de l'accueil de réfugiés, de la gestion de catastrophes naturelles ou de crises sanitaires — des organisations comme les Nations Unies, des ONG internationales ou des structures locales doivent coordonner, en temps réel, des milliers de bénévoles. Des plateformes sont utilisées pour recenser les personnes prêtes à intervenir : médecins, traducteurs, logisticiens, chauffeurs ou simples volontaires disponibles. En parallèle, les centres d'accueil, les cliniques de terrain ou les antennes locales publient des besoins urgents, qui évoluent en permanence selon la situation sur le terrain : « une infirmière est attendue dans un hôpital mobile dans une heure », « un colis alimentaire doit être livré avant le couvre-feu » ou encore « un traducteur parlant arabe doit intervenir dans un centre d'hébergement d'ici la fin de la journée ». Mais ni les volontaires ni les demandes ne se présentent de façon prévisible. Ils apparaissent progressivement, au fil du temps. Un bénévole peut se déclarer disponible quelques minutes avant ou après l'arrivée d'un besoin crucial. La plateforme doit alors prendre une décision rapide : faut-il l'affecter immédiatement à une mission actuelle, ou attendre, en espérant une demande plus urgente ou mieux adaptée ? Ces choix doivent être faits avec une information partielle, et bien souvent, sans possibilité de retour en arrière. Pour des organisations comme les Nations Unies ou Médecins Sans Frontières, il ne s'agit pas simplement d'un problème logistique. C'est une question d'impact concret. Un mauvais appariement peut entraîner des retards dans l'aide, une mauvaise répartition des ressources humaines, voire des situations où des personnes ne sont pas prises en charge à temps. À l'inverse, un système réactif, efficace, bien pensé peut sauver des vies, étendre la portée de l'intervention et limiter les souffrances.

1.4 Réservation de trajet : un appariement en mouvement

Dans les grandes villes contemporaines, les plateformes de VTC doivent prendre des décisions rapides et irréversibles pour affecter les chauffeurs aux passagers à mesure que les demandes affluent. Aux heures de pointe, la pression est intense : des milliers d'utilisateurs sollicitent simultanément une course, tandis que les chauffeurs circulent, terminent une course ou se repositionnent. La plateforme doit alors estimer en temps réel qui est disponible, où, et pour quelle course — sans connaître les requêtes à venir. Prenons l'exemple d'un matin ordinaire. Un passager demande une course depuis le centre-ville vers la périphérie. En quelques secondes, la plateforme analyse le trafic, les temps d'arrivée et la disponibilité des chauffeurs. Une course est confirmée. Tout semble fluide, mais cette décision, une fois prise, ne peut être annulée. Or, quelques instants plus tard, une autre demande arrive, possiblement plus pertinente — mais le chauffeur est déjà engagé. La plateforme doit donc ajuster sa stratégie en continu, en tenant compte des décisions passées et de ressources de plus en plus contraintes. Décider trop tard nuit à l'expérience utilisateur, mais agir trop vite mène à des affectations sous-optimales. Chaque décision individuelle semble anodine, mais leur accumulation quotidienne influence profondément la qualité du service : temps d'attente, rentabilité, trafic, et équilibre géographique. L'enjeu est donc de concevoir des algorithmes capables d'anticiper, de s'adapter, et de maintenir de bonnes performances malgré l'incertitude et l'irréversibilité des choix.

1.5 Publicité en ligne : enchères en temps réel et à grande vitesse

Aujourd'hui, notre navigation sur Internet est étroitement liée à la publicité. Chaque fois qu'un utilisateur effectue une recherche — par exemple « la meilleure paire de chaussures pour la course à pied » —, qu'il consulte une page web ou qu'il fait défiler son fil d'actualité sur les réseaux sociaux, une décision est prise en une fraction de seconde : quelle publicité afficher, pour quel utilisateur et à quel moment ? Ce choix, loin d'être aléatoire, repose sur un processus automatique extrêmement rapide, dans lequel des annonceurs — qu'il s'agisse de marques, de commerçants en ligne ou de grandes plateformes — participent à une enchère en temps réel. Dès qu'une opportunité d'affichage se présente, la plateforme évalue instantanément plusieurs éléments : quel est le profil de l'utilisateur ? Quels annonceurs sont intéressés par cette audience ? Combien sont-ils prêts à payer ? Et surtout, quelle est la probabilité que cette personne clique sur l'annonce ou effectue un achat ? En quelques millisecondes, une vente aux enchères algorithmique est exécutée, et l'annonce la plus « rentable » selon ces critères s'affiche à l'écran. Mais ces décisions sont loin d'être simples. Les annonceurs disposent de budgets limités : ils ne peuvent pas se permettre de viser tous les utilisateurs, à tout moment. Ils cherchent à investir au bon endroit, au bon

1

moment — auprès d'utilisateurs jugés suffisamment pertinents, c'est-à-dire suscep-tibles d'acheter, de s'inscrire ou d'interagir avec la marque. Dès lors, un dilemme se pose : faut-il miser maintenant sur cet utilisateur ou attendre une opportunité future potentiellement plus avantageuse ? Une décision mal calibrée peut gaspiller une part du budget ou faire passer à côté d'une opportunité importante. Ce type de choix se reproduit des milliards de fois par jour, dans un environnement instable, compétitif et personnalisé. Dans ce cadre, le système chargé de faire correspondre annonceurs et utilisateurs doit prendre des décisions instantanées, avec une connais-sance incomplète de l'avenir : il ne sait pas quels utilisateurs apparaîtront plus tard, ni quelles opportunités publicitaires se présenteront ensuite. C'est précisément cette incertitude sur le futur, combinée à l'irréversibilité des décisions et aux contraintes budgétaires, qui rend la tâche particulièrement complexe.

1.6   Inscription aux cours : le dilemme du premier arrivé

Dans le contexte universitaire, les étudiants sont souvent en concurrence pour ac-céder à des cours très demandés. Lors des périodes d'inscription — en début d'an-née ou de semestre —, des milliers d'entre eux se connectent simultanément sur les plateformes d'inscription pour tenter d'obtenir une place. Un étudiant peut réus-sir à s'inscrire simplement parce qu'il a été plus rapide, tandis qu'un autre, arrivé quelques secondes plus tard, peut se retrouver exclu, même si le cours en question est essentiel à l'obtention de son diplôme.

Cette situation soulève une question délicate : faut-il attribuer les places selon le principe du « premier arrivé, premier servi » ? Ou faudrait-il plutôt tenir compte de critères comme l'année d'étude, le niveau de priorité académique ou le parcours individuel de l'étudiant ? Ce dilemme est d'autant plus complexe que, dans la plupart des cas, une fois qu'un cours est complet, il devient difficile de réorganiser les inscriptions sans générer d'insatisfaction ou de confusion. Chaque affectation devient, en pratique, une décision quasi irréversible. Si un étudiant plus prioritaire se manifeste après coup, il devient alors souvent trop tard pour intervenir sans perturber l'équilibre général.

Ce type de situation constitue une forme discrète mais réelle de mise en corre-spondance en ligne, où les décisions doivent être prises au fil du temps, sans visibilité sur les inscriptions futures. Le défi pour les établissements est donc de concevoir des systèmes d'attribution robustes, capables de gérer la dynamique séquentielle du processus : les étudiants ne s'inscrivent pas tous en même temps, et les places sont en nombre limité. Il ne s'agit pas seulement d'optimiser l'affectation pour qu'elle soit rapide ou efficace, mais aussi de garantir qu'elle soit équitable, afin de ne pas compromettre les parcours académiques des étudiants. L'enjeu dépasse l'organisa-tion administrative : il touche à l'égalité des chances et à l'équité dans l'accès à la formation.

1.7 Applications de rencontre : trouver le bon swipe

Dans l'univers des applications de rencontre, la mise en relation est au cœur même du fonctionnement de ces plateformes. Chaque jour, des millions d'utilisateurs se connectent dans l'espoir de rencontrer quelqu'un avec qui créer un lien, une histoire ou simplement un moment de connexion. Sur des applications comme Tinder, Bumble ou Hinge, les profils sont présentés un par un, et chaque utilisateur exprime son intérêt par un simple geste : glisser à droite ou à gauche. Mais derrière ce geste simple, une série de décisions se joue en continu : quels profils vous montrer, dans quel ordre, et à qui afficher le vôtre en retour. Ce processus s'inscrit dans un environnement qui n'est pas figé. De nouveaux utilisateurs s'inscrivent chaque jour, changent de localisation, modifient leur description ou ajustent leurs préférences. Face à cette instabilité permanente, les plateformes doivent constamment s'adapter. D'un côté, elles doivent proposer des profils intéressants pour maintenir l'attention et fidéliser les utilisateurs ; de l'autre, elles font face à un dilemme permanent : faut-il vous proposer immédiatement un profil prometteur ou attendre l'arrivée potentielle d'une personne encore plus compatible ? Faut-il vous montrer à un utilisateur actif à l'instant, ou à quelqu'un qui, selon ses habitudes, serait plus susceptible de vous répondre ? Ces décisions sont prises sans connaître l'avenir — ni les prochaines connexions ni les réactions possibles. Et souvent, une fois un profil passé ou ignoré, il est difficile, voire impossible, de revenir en arrière. Mais au-delà de l'aspect technique ou algorithmique, ces plateformes jouent aussi un rôle social profond. Elles influencent les comportements, les attentes, et parfois même les normes relationnelles. Une rencontre réussie peut déboucher sur une relation durable ; à l'inverse, une mauvaise expérience peut créer de la déception, de la frustration, voire conduire à quitter la plateforme. Cela pose des questions essentielles : qui est mis en avant ? À quel rythme les profils sont-ils proposés ? Quels critères déterminent réellement les compatibilités ? Ce n'est pas uniquement une affaire de données ou de performances algorithmiques. C'est un mécanisme où les émotions, les préférences individuelles et les décisions prises à l'instant T influencent silencieusement l'avenir des utilisateurs.

Tous ces exemples, bien qu'issus de domaines très différents, révèlent une réalité commune : le problème d'appariement est omniprésent, et ses enjeux sont loin d'être simples. Derrière chaque mise en relation se cachent des décisions complexes, souvent prises en temps réel, dans des environnements contraints et instables. L'efficacité, l'équité, voire l'impact social de ces systèmes dépendent des algorithmes utilisés pour gérer les appariements. Comprendre leur fonctionnement, évaluer leurs performances et trouver des moyens de les améliorer constitue donc un enjeu à la fois technique et sociétal majeur.

Dans cette thèse, nous concentrons notre analyse plus précisément sur le cas de la publicité en ligne. Ce domaine, où l'appariement s'effectue entre utilisateurs et annonceurs via des enchères algorithmiques à grande échelle, constitue un ter-

rain d'étude très riche. Nous cherchons à mieux comprendre les mécanismes liés à ces systèmes et à analyser leurs performances dans différents scénarios et sous diverses contraintes. Pour étudier ce problème, nous procédons par étapes. Nous commençons par modéliser l'appariement d'un point de vue mathématique, en nous appuyant sur des outils issus des probabilités et des statistiques. Cette première modélisation, plus simple et générique, nous permet d'évaluer les performances de plusieurs algorithmes standards. Nous introduisons ensuite des contraintes inspirées des systèmes réels : limitations de budget, évolution budgétaire au cours du temps, contraintes liées aux profils des utilisateurs, etc., et analysons par la suite comment les algorithmes réagissent à ces nouvelles conditions. Dans un second temps, nous considérons un modèle mathématique plus riche, plus proche des dynamiques observées sur les plateformes réelles, et poursuivons l'évaluation des algorithmes d'appariement dans ce cadre. Enfin, nous proposons une conception plus formelle des mécanismes d'enchères en ligne, ainsi que de nouveaux algorithmes d'appariement adaptés à ce contexte, accompagnés d'une analyse théorique de leurs performances.

2

Contexte (en français)

Contents

L'objectif de ce chapitre est de fournir un aperçu des concepts fondamentaux nécessaires à la compréhension de la formulation mathématique des problèmes de matching (appariement) ainsi que des algorithmes standards utilisés pour construire des matchings maximaux. Nous commençons par introduire la notion de graphe, qui constitue la structure de base pour modéliser les relations et les attributions entre entités. Différents types de graphes sont présentés, suivis d'une définition formelle du matching. Nous passons ensuite au cadre en ligne, où plusieurs modèles de matching en ligne sont discutés, ainsi que la notion de ratio de compétitivité, utilisée pour évaluer la performance des algorithmes de matching. Sous un autre angle, nous introduisons le problème du bandit manchot, un modèle central dans la prise de décision séquentielle. Bien que sa structure diffère de celle du matching sur graphe, il fait face des défis similaires : prendre des décisions avec une information partielle,

s'adapter au fil du temps, et optimiser les résultats dans un contexte d'incertitude — des principes que l'on retrouve également dans de nombreuses applications de matching en ligne. Dans la suite de ce chapitre, nous présentons les principes de base du modèle du bandit, ainsi que quelques algorithmes standards utilisés pour le résoudre.

2.1   Matching et sa version en ligne

Dans cette section, nous commençons par définir ce qu'est un graphe, ainsi que plusieurs types importants de graphes couramment utilisés pour modéliser des systèmes du monde réel. Les graphes offrent un cadre mathématique puissant pour représenter les relations entre entités — telles que les utilisateurs et les ressources, ou les agents et les tâches —, ce qui en fait un outil central dans l'étude des problèmes de matching. Nous définissons ensuite formellement la notion de matching sur un graphe, qui correspond à l'idée d'attribuer des paires compatibles sans conflit. Sur cette base, nous introduisons la version en ligne du matching, un concept clé pour modéliser des scénarios du monde réel (voir Chapters 1 and 4). Enfin, nous présentons plusieurs algorithmes classiques conçus pour le matching en ligne et discutons des outils théoriques permettant d'évaluer leur performance.

2.1.1   Définition d'un graphe

Définition 1. *Un graphe est un couple $G = (V, E)$, où V est un ensemble de nœuds (ou sommets), et E est un ensemble d'arêtes qui sont des paires non ordonnées $\{v_1, v_2\} \in V^2$ de nœuds. Si les arêtes ont une direction — c'est-à-dire qu'elles vont d'un sommet vers un autre — on parle alors de graphe orienté ; sinon, le graphe est non orienté. Un nœud peut ne faire partie d'aucune arête ; on le qualifie alors d'isolé. Lorsqu'une arête $\{v_1, v_2\}$ existe, les nœuds v_1 et v_2 sont dits adjacents.*

En fonction de la structure de l'ensemble des sommets V et de celle des arêtes E, les graphes peuvent prendre différentes formes. Nous présentons ci-après plusieurs types de graphes particulièrement importants dans le cadre de notre étude.

2.1.2   Types de graphes

Définition 2. *Un graphe biparti $G = (U, V, E)$ est un graphe dont les sommets peuvent être divisés en deux ensembles disjoints U et V ($U \cap V = \varnothing$), tels que chaque arête relie un sommet de U à un sommet de V, et qu'aucune arête n'existe entre des sommets d'un même ensemble, c'est-à-dire, $E \subseteq U \times V$.*

2

Dans le reste de ce manuscrit, nous concentrons notre analyse exclusivement sur les graphes bipartis, car ils offrent un cadre naturel pour modéliser les scénarios qui nous intéressent. Les graphes bipartis se divisent généralement en deux grandes catégories : les graphes bipartis déterministes, dans lesquels la structure des arêtes et la répartition des sommets entre les deux ensembles sont fixées ; et les graphes bipartis aléatoires, dans lesquels ces éléments sont définis de manière stochastique. Dans la définition ci-dessous, nous introduisons formellement la notion de graphe biparti aléatoire.

Définition 3. *Soient U et V deux ensembles disjoints de sommets, avec $|U| = n$, $|V| = m$, et $\Omega = 2^{U \times V}$ l'ensemble de toutes les parties de $U \times V$, c'est-à-dire l'ensemble de tous les sous-ensembles possibles d'arêtes entre U et V. Soit $(\Omega, \mathcal{F}, \mathbb{P})$ un espace probabilisé, où $\mathcal{F}$ est l'ensemble des parties de Ω (la tribu engendrée) et $\mathbb{P} : \mathcal{F} \to [0,1]$ une mesure de probabilité définie sur les sous-ensembles d'arêtes. Alors, un graphe aléatoire biparti est une variable aléatoire,*

$$G : (\Omega, \mathcal{F}, \mathbb{P}) \to \mathcal{G}_{n,m},$$

où $\mathcal{G}_{n,m}$ désigne l'ensemble de tous les graphes bipartis dont les ensembles de sommets U et V sont fixés.

Ayant précédemment mentionné notre intérêt pour les graphes bipartis en raison de leur adéquation à la modélisation des problèmes étudiés, nous portons notre attention à présent vers des modèles spécifiques de graphes bipartis aléatoires. Ces modèles se distinguent par la manière dont les arêtes sont générées et par la structure des ensembles de sommets. Nous commençons par le modèle de graphe biparti d'Erdős–Rényi, un cadre standard et largement utilisé de graphes aléatoires.

Définition 4. *Soient U et V deux ensembles disjoints de sommets, avec $|U| = n$, $|V| = m$, et $\Omega = 2^{U \times V}$ l'ensemble de toutes les parties de $U \times V$, c'est-à-dire l'ensemble de tous les sous-ensembles possibles d'arêtes entre U et V. Soit $(\Omega, \mathcal{F}, \mathbb{P}_p)$ un espace probabilisé, où $\mathcal{F}$ est l'ensemble des parties de Ω et $\mathbb{P}_p$ est la mesure de probabilité définie sur Ω telle que chaque arête $(u, v) \in U \times V$ est incluse indépendamment avec une probabilité $p \in [0, 1]$, c'est-à-dire que pour tout $S \subseteq U \times V$, $\mathbb{P}(\{S\}) = p^{|S|}(1-p)^{|U \times V| - |S|}$. Alors, le graphe biparti aléatoire d'Erdős–Rényi est la variable aléatoire,*

$$G : (\Omega, \mathcal{F}, \mathbb{P}_p) \to \mathcal{G}_{n,m},$$

où $\mathcal{G}_{n,m}$ désigne l'ensemble de tous les graphes bipartis ayant pour ensembles de sommets fixés U et V.

Bien que le modèle d'Erdős–Rényi repose sur des probabilités d'arêtes uniformes et indépendantes entre toutes les paires de sommets, il ne permet pas de représenter la structure en communautés fréquemment observée dans les problèmes du monde réel. Une généralisation de ce modèle, appelée modèle à blocs stochastiques (SBM, pour Stochastic Block Model), regroupe les sommets en communautés et autorise des probabilités d'arêtes variables entre les groupes. Cela permet de modéliser des interactions structurées.

Définition 5. *Soient U et V deux ensembles disjoints de sommets, avec $|U| = n$, $|V| = m$, et $\Omega = 2^{U \times V}$ l'ensemble de tous les sous-ensembles d'arêtes de $U \times V$. Supposons que chaque nœud $u \in U$ soit assigné à une classe $c(u) \in C$, et que chaque $v \in V$ soit assigné à une classe $c(v) \in D$. Soit $P \in [0,1]^{C \times D}$ une matrice de probabilités de connexion, où $P_{i,j}$ représente la probabilité qu'une arête existe entre un nœud $u \in U$ tel que $c(u) = i$ et un nœud $v \in V$ tel que $c(v) = j$.*

Soit $(\Omega, \mathcal{F}, \mathbb{P}_{sbm})$ un espace probabilisé, où $\mathcal{F}$ est l'ensemble des parties de Ω, et $\mathbb{P}_{sbm}$ est la mesure de probabilité définie sur Ω telle que les arêtes sont incluses de manière indépendante, c'est-à-dire que pour tout $S \subseteq U \times V$, $\mathbb{P}_{sbm}(\{S\}) = \prod_{(u,v) \in S} P_{c(u),c(v)} \prod_{(u,v) \notin S} (1 - P_{c(u),c(v)})$. Alors, le modèle à blocs stochastiques biparti est une variable aléatoire,

$$G : (\Omega, \mathcal{F}, \mathbb{P}_{sbm}) \to \mathcal{G}_{n,m},$$

où $\mathcal{G}_{n,m}$ désigne l'ensemble de tous les graphes bipartis ayant pour ensembles de sommets fixés U et V.

Plusieurs autres modèles de graphes aléatoires ont été proposés dans la littérature de la théorie des graphes [19, 52, 36], afin de représenter diverses propriétés structurelles et probabilistes des réseaux. Parmi ceux-ci, on trouve le modèle de configuration, qui permet de fixer à l'avance les distributions de degrés ; les graphes aléatoires géométriques, qui intègrent des contraintes spatiales ; les graphes aléatoires uniformes, où les graphes sont échantillonnés de manière uniforme à partir d'une classe fixée ; ou encore les arbres de Galton–Watson, couramment utilisés pour modéliser des processus de branchement, ainsi que de nombreux autres types de graphes aléatoires. Dans ce travail, nous limitons notre étude aux modèles de graphes bipartis aléatoires les plus pertinents pour les problèmes abordés dans la suite du manuscrit, à savoir les modèles d'Erdős–Rényi et les modèles à blocs stochastiques.

2.1.3   Un matching sur un graphe

Maintenant que nous avons défini les graphes bipartis et présenté plusieurs catégories de modèles correspondants, nous nous intéressons à une notion structurelle importante : le matching (ou appariement). Celle-ci représente l'idée d'une affectation par paires entre les deux ensembles de sommets et joue un rôle central dans de nombreuses applications concrètes. Dans ce qui suit, nous en donnons une définition formelle dans un graphe biparti.

Définition 6. *Soit $G = (U, V, E)$ un graphe biparti. Pour chaque sommet $u \in U$, on définit son degré dans G, noté $\deg_G(u)$, comme le nombre d'arêtes dans E qui sont incidentes à u, et pour chaque sommet $v \in V$, $\deg_G(v) = 1$, bien que v puisse bien sûr avoir de nombreux voisins dans le graphe. À chaque sommet $u \in U$, on associe un entier naturel $c(u) \in \mathbb{N}$, appelé sa capacité de matching, satisfaisant $c(u) \leq \deg_G(u)$. Un matching dans G est un sous-ensemble d'arêtes $M \subseteq E$ tel que*

chaque sommet $u \in U$ est incident à au plus $c(u)$ arêtes dans M, et chaque sommet $v \in V$ est incident à au plus une arête dans M.

Remarque 1. *Dans la définition classique d'un matching dans un graphe biparti, chaque sommet de U comme de V ne peut être apparié qu'à un seul sommet de l'ensemble opposé.*
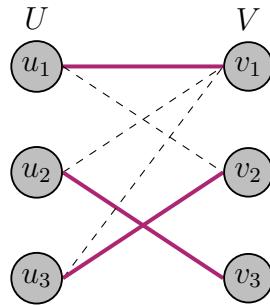
Figure 2.1: Matching dans un graphe biparti

2.1.4 Matching en ligne

La version hors ligne du problème de matching a été largement étudiée [42, 19, 59, 18, 101]. Dans ce cadre, la structure complète du graphe est connue à l'avance, et l'objectif est de calculer un matching optimal en se basant sur une information complète. Cependant, comme discuté précédemment dans Chapters 1 and 4, de nombreux scénarios réels correspondent à des situations où la structure complète du graphe n'est pas disponible au début du processus. Pour modéliser ces dynamiques en ligne, le concept de matching biparti en ligne a été introduit dans [63], où un côté du graphe biparti reste fixe tandis que l'autre est révélé de manière séquentielle. Pour mieux formaliser ce cadre, nous définissons la notion de graphe biparti en ligne, dans lequel la nature dynamique du problème est explicitement représentée par l'arrivée progressive d'une partie du graphe au fil du temps.

Définition 7. *Soit $G = (U, V, E)$ un graphe biparti. Dans le cadre du matching biparti en ligne, l'ensemble U est connu à l'avance et reste fixe tout au long du processus, tandis que l'ensemble V est révélé séquentiellement au fil du temps. À chaque pas de temps $t \in \{1, \ldots, |V|\}$, un sommet $v_t \in V$ est révélé, accompagné de l'ensemble de ses voisins dans U. L'algorithme doit alors prendre une décision irrévocable, celle d'apparier ou non v_t à un voisin disponible dans U.*

Ici, la notion de temps est modélisée par la révélation séquentielle des sommets de V, avec l'arrivée d'un sommet à chaque étape.
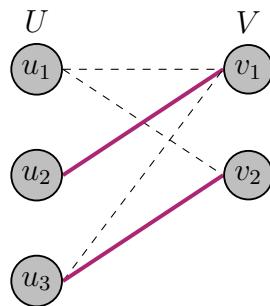
Figure 2.2: Matching en ligne dans un graphe biparti

Dans le cadre en ligne, un matching sur un graphe biparti conserve la définition hors ligne donnée en Définition 6. Cependant, dans ce contexte, le graphe est révélé de manière séquentielle, et les décisions doivent être prises sans connaissance des sommets à venir. À mesure que de nouveaux sommets de V sont dévoilés au fil du temps, le matching évolue en conséquence. Tout au long de ce processus, les algorithmes de matching en ligne tentent de construire en temps réel, un matching aussi large que possible, malgré l'information partielle dont ils disposent à chaque étape.

2.1.4.1   le ratio de compétitivité

Comme discuté précédemment, dans le problème de matching en ligne, lorsqu'un sommet de V arrive, un algorithme en ligne tente de l'apparier à l'un de ses voisins dans U, en suivant certaines règles de décision. Une fois la décision prise, elle est irrévocable. Cela modélise des situations pratiques dans lesquelles les décisions ne peuvent pas être annulées : par exemple, lorsqu'une publicité est affichée à un utilisateur ou qu'un profil est présenté sur une application de rencontre, le choix est fait et ne peut être modifié. Pour évaluer la performance de tels algorithmes en ligne, une mesure standard appelée ratio de compétitivité a été introduite dans la littérature sur le matching en ligne [81]. Ce ratio reflète la perte induite par le fonctionnement en ligne — sans connaissance complète des données futures — par rapport à un algorithme hors ligne optimal, qui connaît l'ensemble du graphe à l'avance. De manière générale, il est défini comme le rapport entre la taille espérée du matching produit par l'algorithme en ligne et celle de l'optimum hors ligne. Plus formellement :

Définition 8. *Soit $\mathcal{G}$ une famille de graphes bipartis, et* ALG(G) *la taille de le matching construit par un algorithme en ligne* ALG *sur un graphe $G \in \mathcal{G}$, et* OPT(G) *la taille de le matching construit par l'algorithme hors ligne optimal* OPT*. On dit alors qu'un algorithme* ALG *atteint un ratio de compétitivité α sur la famille $\mathcal{G}$ s'il existe une constante c telle que, pour tout $G \in \mathcal{G}$,*

$$\mathbb{E}[\text{ALG}(G)] \geq \alpha\mathbb{E}[\text{OPT}(G)] + c,$$

où les espérances sont prises par rapport à l'aléa éventuel dans les algorithmes. Deux types de garanties compétitives sont couramment étudiées. Dans le premier cas, le graphe G st tiré aléatoirement selon une distribution donnée sur un ensemble de graphes (par exemple, un modèle de graphe aléatoire), et le ratio de compétitivité est évalué en espérance par rapport à cette source d'aléa. Dans le second cas, une analyse dans le pire cas est effectuée sur l'ensemble des graphes $G \in \mathcal{G}$, et la borne doit alors être satisfaite uniformément pour tous les graphes de la famille.

Le ratio de compétitivité, noté `CR`, est une quantité comprise entre 0 et 1. Une valeur élevée de `CR` indique une meilleure performance de l'algorithme considéré. Deux facteurs principaux influencent la valeur de `CR` : le choix de l'algorithme lui-même et la classe de graphes $\mathcal{G}$ sur laquelle l'algorithme est évalué.

2.1.4.2 Cadres et algorithmes de matching en ligne biparti

Dans la dernière partie de cette section, nous présentons les cadres standards généralement considérés dans l'étude du problème de matching en ligne [81]. Ceux-ci incluent, en particulier, les hypothèses classiques formulées sur la famille de graphes $\mathcal{G}$, ainsi que les algorithmes proposés dans la littérature pour chacun de ces contexte.

Le cadre adversarial. Dans ce cadre, la famille $\mathcal{G}$ peut être n'importe quelle famille de graphes bipartis, ce qui signifie que, pour tout $G \in \mathcal{G}$, les sommets de V peuvent arriver dans un ordre arbitraire. Le ratio de compétitivité d'un algorithme, dans ce cas, est évalué sur le graphe pour lequel il obtient les moins bons résultats.

De manière étonnante, même dans ce cadre difficile, une garantie de 1/2 sur le `CR` peut être obtenue avec la stratégie classique `Greedy`.

Algorithm 1: Algorithme `Greedy`

---
1 Pour $t = 1$ jusqu'à $|V|$ **faire**
2   Apparier v_t à un voisin libre choisi uniformément au hasard.
3 Fin pour

Le résultat suivant montre que le ratio de compétitivité de `Greedy` est borné inférieurement par 1/2. Ce résultat s'applique à `Greedy` ainsi qu'à tout algorithme qui choisit une correspondance dès qu'elle est disponible.

Théorème 1. *Dans le cadre adversarial,*

$$\mathtt{CR}(\mathtt{Greedy}) \geq \frac{1}{2}.$$

Preuve 1. *Considérons le cas illustré dans la Figure 2.3, où l'algorithme `Greedy` échoue à matcher le sommet v_2, bien que celui-ci soit matché dans le matching maximal du graphe final. Cela ne peut se produire que si le sommet de U qui*

aurait été matché à v_2 dans la solution optimale a déjà été matché par `Greedy` *à un autre sommet, disons v_1, arrivé plus tôt. Ainsi, pour chaque événement "manqué" (c'est-à-dire un sommet que* `Greedy` *ne parvient pas à matcher mais qui l'est dans le matching maximal), il existe au moins un événement de "match" (un sommet effectivement matché par* `Greedy`*). Cela implique que le nombre total de matchings dans la solution optimale est au plus le double du nombre de matchings effectués par* `Greedy`*. Autrement dit, la taille du matching produit par* `Greedy` *est au moins égale à la moitié de celle du matching optimal. Par conséquent, le ratio de compétitivité de* `Greedy` *est minoré par $\frac{1}{2}$.*
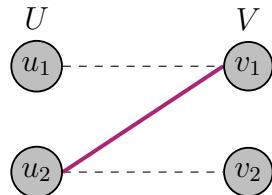
Figure 2.3: Une instance difficile pour `Greedy`

$\square$

On peut aller plus loin et montrer que cette borne inférieure est atteinte, comme suit :

Proposition 1. *Dans le cadre adversarial,*

$$\mathtt{CR}(\mathtt{Greedy}) = \frac{1}{2}.$$

Preuve 2. *Considérons un graphe G_n illustré dans la Figure 2.4. Il comporte $2n$ sommets de chaque côté, répartis comme suit : $U = U_1 \cup U_2$ et $V = V_1 \cup V_2$, avec $U_1 = \{u_1, \ldots, u_n\}$ et $U_2 = \{u_{n+1}, \ldots, u_{2n}\}$, et de manière similaire pour V. L'ensemble des arêtes est défini par $E = \{(u_i, v_i), \forall i \in [2n], (u_{n+i}, v_j) \, \forall (i,j) \in [n]^2\}$. Le matching maximal possible est de taille $2n$. Si l'on applique l'algorithme `Greedy` sur G_n, on observe que tout sommet $v_i \in V_1$ choisira, avec une probabilité supérieure à $\frac{n-i+1}{n-i+2}$, un partenaire dans U_2, car il ne possède qu'un seul voisin dans U_1, et qu'à l'instant i, au plus $i-1$ sommets ont déjà été matchés dans U_2. Cela implique qu'en espérance, $n - o(n)$ sommets de V_1 seront matchés à un sommet $u_j \in U_2$. Ainsi,*

$$\lim_{n \to \infty} \frac{\mathbb{E}[\mathtt{Greedy}(G_n)]}{2n} = \frac{1}{2}.$$
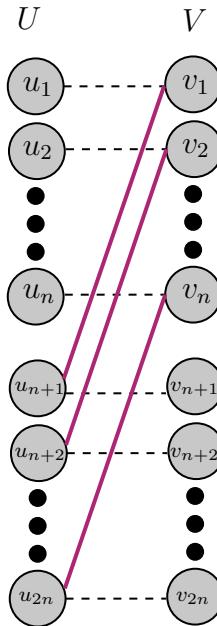
Figure 2.4: Une instance difficile pour `Greedy`

$\square$

Comme présenté dans [81], un autre algorithme peut offrir de meilleures garanties que `Greedy` dans le cas adversarial : l'algorithme `Ranking`, qui repose sur le principe de la randomisation corrélée. L'algorithme `Ranking` fonctionne de la manière suivante : au début du processus, une permutation aléatoire π est tirée, et chaque sommet $u_i \in U$ se voit attribuer un rang $\pi(i)$. Lorsqu'un sommet $v \in V$ arrive, il est matché à l'un de ses voisins dans U ayant le rang le plus faible.

Algorithm 2: L'algorithme `Ranking`

---
1 Tirer une permutation aléatoire π.
2 Pour $i = 1$ jusqu'à $|U|$ **faire**
3   Affecter à u_i un rang $\pi(i)$.
4 Pour $t = 1$ jusqu'à $|V|$ **faire**
5   Matcher v_t à un voisin disponible ayant le rang le plus faible.

En revenant à l'exemple présenté dans la Figure 2.4, on peut comprendre pourquoi l'algorithme `Ranking` obtient de meilleures performances que `Greedy`. Lorsqu'un sommet $v \in V_1 \cup V_2$ arrive, `Greedy` a tendance à le matcher à un voisin de degré élevé dans U, ce qui introduit un biais : `Greedy` match souvent trop tôt les sommets de degré élevé. Or, une stratégie plus efficace consisterait à préserver ces sommets de haut degré comme options de secours pour les arrivées futures ayant moins de voisins. En attribuant des rangs aux sommets de U, `Ranking` corrige ce biais. Lorsqu'un sommet $v \in V_1 \cup V_2$ arrive, `Ranking` le match au voisin libre ayant le rang le plus bas. En conséquence, au fil du temps et en espérance, `Ranking` parvient à matcher plus de sommets que `Greedy` sur cette instance.

Theorem 1. *Dans le cadre adversarial,*

$$\text{CR}(\texttt{Ranking}) \geq 1 - \frac{1}{e} \simeq 0.63,$$

où e est le nombre d'Euler.

Ce résultat a été établi à l'aide de plusieurs techniques de preuve différentes. Une preuve directe basée sur la correspondance entre les événements "manqué" et les événements de "match", est proposée dans [16]. Une autre approche repose sur une analyse primal-duale ; voir [35] pour plus de détails. Dans l'article de référence ayant introduit l'algorithme `Ranking` [63], les auteurs ont démontré que la borne inférieure sur son ratio de compétitivité est atteinte, en utilisant une famille d'instances appelées graphes triangulaires supérieurs (upper-triangular graphs). En s'appuyant sur cet exemple et en appliquant le lemme de Yao, ils ont également montré qu'aucun algorithme randomisé ne peut atteindre un ratio de compétitivité supérieur à $1 - \frac{1}{e}$. Pour plus de détails, voir [63, 81].

Ordre Aléatoire. Dans ce cadre, le graphe peut toujours être un graphe biparti quelconque, mais contrairement au modèle adversarial, les sommets de V arrivent dans un ordre aléatoire. Dans ce modèle d'arrivée aléatoire, les algorithmes `Greedy` et `Ranking` obtiennent de meilleures performances que dans le cadre adversarial. Plus précisément, `Greedy` atteint un ratio de compétitivité de $1 - \frac{1}{e}$. Cela s'explique par le fait qu'avec un ordre d'arrivée aléatoire, `Greedy` se comporte de manière similaire à `Ranking`, et partage la même borne inférieure de $1 - \frac{1}{e}$ sur le `CR`. Quant à l'algorithme `Ranking`, son ratio de compétitivité exact dans le modèle d'arrivée aléatoire reste un problème ouvert. Toutefois, la meilleure borne inférieure connue, d'environ 0.696, a été obtenue dans [76].

Known i.i.d. Dans les applications réelles, on dispose souvent d'une certaine information sur la structure du graphe, ce qui rend les modèles adversariaux ou à ordre aléatoire très restrictifs. Pour mieux refléter ces scénarios pratiques, un nouveau cadre, appelé modèle Known i.i.d., a été introduit dans [40]. Dans ce cadre, l'algorithme connaît l'ensemble des sommets U ainsi qu'une distribution sur les types possibles de sommets dans V, où un type spécifie l'ensemble des voisins dans U. Grâce à cette information supplémentaire, un nouvel algorithme, appelé `Suggested − Matching`, a été proposé dans [40]. L'idée est simple : l'algorithme pré-calcule un matching parfait sur le graphe de base (dans lequel V est remplacé par l'ensemble des types), et s'en sert pour orienter les décisions en ligne. Lorsqu'un sommet de V arrive, l'algorithme lui assigne un voisin unique, déterminé à l'avance à partir du matching sur le graphe de base, avec lequel il tente de réaliser une correspondance. L'objectif est ainsi de reproduire, autant que possible, le matching parfait calculé hors ligne, mais dans un cadre en ligne. Comme montré dans [40], cet algorithme atteint un ratio de compétitivité de $1 - \frac{1}{e}$. Cependant, bien qu'efficace, cette approche reste sous-optimale, car tout type apparaissant une seconde fois (ou

plus) reste non matché. Cette limitation a conduit à une série d'améliorations. La première avancée a introduit des algorithmes avec deux choix : au lieu de s'appuyer sur un seul matching suggéré dans le graphe de base pour guider les décisions, l'algorithme utilise deux matchings, ce qui permet d'élever le `CR` à $0,706$ [40, 77, 58]. Plus récemment, un algorithme utilisant plus de deux choix a été proposé. Celui-ci repose sur la résolution d'un programme linéaire, combiné à des techniques supplémentaires de résolution en ligne (Online Resolution Scheme, OCS) [55], permettant d'avoir un ratio de compétitivité de $0,716$.

2.1.5 Le $b-$matching problème

Comme nous l'avons vu tout au long de ce chapitre, le problème du matching en ligne fait l'objet d'un vaste champ de recherche, témoignant à la fois de sa richesse théorique et de sa pertinence pratique. De nombreuses généralisations de ce problème ont été proposées — certaines pour mieux modéliser des applications réelles, d'autres pour en saisir la complexité dans des contextes plus élaborés. L'une de ces généralisations est le problème de b-matching, dans lequel, au lieu de supposer que les sommets de U ne peuvent être appariés qu'une seule fois, on autorise chaque sommet à être apparié jusqu'à b fois, où $b \in \mathbb{N}^*$ représente le budget associé à chaque sommet de U. On peut interpréter ces budgets comme les degrés des sommets de U. Cette généralisation est particulièrement pertinente dans le cadre de la publicité en ligne, où les annonceurs (les sommets de U) disposent généralement de budgets de campagne limitant le nombre de fois où leurs annonces peuvent être affichées aux utilisateurs (les sommets de V arrivant en ligne). Pour traiter ce problème, un algorithme appelé `Balance` a été introduit dans [62]. Il fonctionne comme suit :
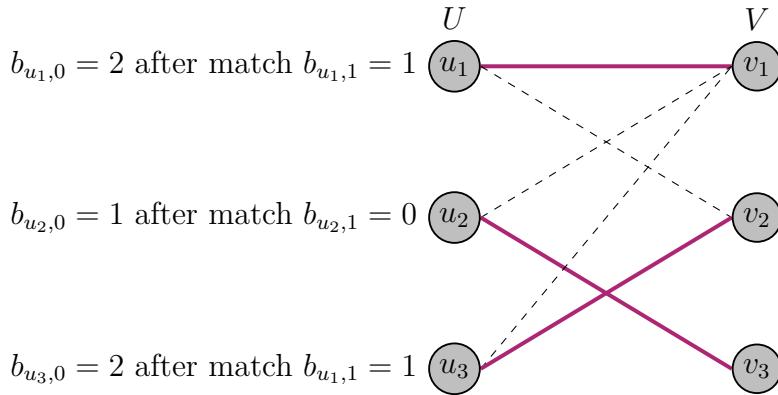
Algorithm 3: L'algorithme `Balance`

---
1 Pour $t = 1, \ldots, |V|$ **faire**
2   Matcher v_t à un voisin ayant le budget restant le plus élevé.
3 Fin Pour

Pour mieux comprendre le fonctionnement de l'algorithme `Balance`, appliquons-le à l'exemple illustré dans la Figure 2.5. Nous avons trois sommets dans U : u_1 avec un budget initial $b_{u_1,0} = 2$, u_2 avec $b_{u_2,0} = 1$, et u_3 avec $b_{u_3,0} = 2$. Lorsque v_1 arrive, `Balance` peut le matcher à l'un de ses voisins — supposons qu'il choisisse u_1. Après ce matching, le budget de u_1 devient $b_{u_1,1} = 1$. Ensuite, v_2 arrive avec deux voisins : u_1, dont le budget est maintenant $b_{u_1,1} = 1$, et u_3, avec $b_{u_3,1} = 2$. Comme `Balance` privilégie le voisin ayant le budget restant le plus élevé, il choisira de matcher v_2 avec u_3. Enfin, lorsque v_3 arrive, il n'a qu'un seul voisin, u_2, donc `Balance` n'a qu'un seul choix possible.

$b_{u_1,0} = 2$ after match $b_{u_1,1} = 1$

$b_{u_2,0} = 1$ after match $b_{u_2,1} = 0$

$b_{u_3,0} = 2$ after match $b_{u_1,1} = 1$

Figure 2.5: Fonctionnement de l'algorithme `Balance`

On peut observer que l'algorithme `Balance` prend des décisions légèrement plus stratégiques que `Greedy`, dans la mesure où il privilégie systématiquement les voisins ayant le budget restant le plus élevé, répartissant ainsi progressivement la charge entre les sommets de U — ce qui correspond précisément à l'idée derrière son nom. L'analyse de `Balance` dans le cadre adversarial a été présentée pour la première fois dans [62], où les auteurs ont démontré un ratio de compétitivité paramétré par b, le budget des sommets de U, donné par : $1 - \frac{1}{(1+1/b)^b}$. Ce résultat est obtenu à l'aide de la construction d'un graphe biparti en ligne soigneusement conçu, suivie de l'analyse du matching produit par `Balance`. Ils ont également montré que `Balance` est optimal parmi tous les algorithmes déterministes dans le cadre adversarial. Une généralisation supplémentaire, plus proche des applications pratiques, du problème de b-matching a été proposée dans [6]. Dans ce cadre, les budgets des sommets de U sont fixes mais non égaux. La performance de `Balance` est alors caractérisée en fonction du plus petit budget, noté $b_{\min} = \min_{u \in U} b_u$, où b_u est le budget du sommet $u \in U$. Le ratio de compétitivité correspondant est donné par : $1 - \frac{1}{(1+1/b_{\min})^{b_{\min}}}$.

2.2   Le problème de bandit multi-bras

Dans sa forme classique, le problème de matching en ligne consiste à choisir, au fur et à mesure de l'arrivée des sommets d'un côté du graphe, une séquence de décisions maximisant la taille finale du matching. Dans ce cadre, même si la structure du graphe — c'est-à-dire l'ensemble des voisins de chaque sommet entrant — est révélée au moment de l'arrivée du sommet, une hypothèse essentielle demeure : le matching est purement combinatoire. Autrement dit, les arêtes n'ont pas de valeur associée, ou, si elles en ont une, elle est connue à l'avance. Cependant, dans de nombreuses applications réelles — comme les systèmes de recommandation, les plateformes de commerce en ligne ou l'allocation dynamique de ressources — cette hypothèse est trop restrictive. Dans ces contextes, le choix d'un voisin génère une récompense, dont la valeur n'est pas connue avant d'être observée. Le problème n'est alors plus seulement de construire un matching de grande taille, mais d'apprendre quelles arêtes rapportent réellement, tout en optimisant les récompenses

2

cumulées au fil du temps. Cette dimension supplémentaire transforme profondément la nature du problème : l'algorithme ne connaît plus les conséquences de ses décisions et doit continuellement arbitrer entre exploiter les arêtes déjà identifiées comme profitables et en explorer d'autres pour améliorer son estimation. C'est précisément ce dilemme — apprendre tout en sélectionnant les meilleures options — qui mène naturellement au cadre des bandits manchots. Inspiré des machines à sous, ce problème met en scène un agent confronté à plusieurs choix possibles (les « bras »), chacun produisant une récompense aléatoire, inconnue à l'avance. À chaque étape, l'agent doit choisir un bras afin de maximiser la récompense cumulée, tout en gérant le délicat compromis entre exploitation (continuer à choisir l'option la plus prometteuse) et exploration (tester des alternatives) [72]. Dans la suite de ce chapitre, nous introduisons la définition formelle du problème du bandit manchot dans le cadre stochastique. Nous présentons ensuite la notion de regret, qui joue un rôle analogue au ratio de compétitivité dans les problèmes de matching. Enfin, nous détaillons quelques algorithmes classiques utilisés pour résoudre ce type de problème.

2.2.1 Définition du problème de bandit

Un problème de bandit manchot à plusieurs bras (multi-armed bandit, MAB) est un problème de prise de décision séquentielle, défini par un ensemble fini d'actions $\mathcal{A} = \{1, \ldots, K\}$, appelées bras dans la littérature sur les bandits. Chaque bras $i = 1, \ldots, K$ est associé à une suite de récompenses inconnues $X_{i,1}, X_{i,2}, \ldots, X_{i,K}$ dans l'intervalle $[0, 1]$. À chaque étape, le joueur prend une décision en sélectionnant un bras k_t dans $\mathcal{A}$ et observe une récompense $X_{k_t,t}$. L'objectif du joueur est de choisir la meilleure séquence d'actions au fil du temps afin de maximiser la récompense cumulée. Il existe plusieurs variantes du MAB, chacune reposant sur des hypothèses différentes concernant la structure des récompenses et leur génération. Dans le cadre adversarial, les récompenses sont déterminées par un adversaire et peuvent varier en fonction des actions passées de l'algorithme. Le cadre linéaire, quant à lui, suppose que chaque bras est associé à un vecteur de caractéristiques, et que la récompense espérée est une fonction linéaire de ces caractéristiques. Enfin, dans le cadre stochastique, qui est celui que nous étudions dans ce chapitre, chaque bras est associé à une distribution de probabilité fixe mais inconnue, à partir de laquelle les récompenses sont tirées de manière indépendante à chaque étape. Pour plus de détails sur ces différents cadres, voir [72]. Le problème stochastique de bandit manchot peut être résumé comme suit : à chaque étape temporelle $t = 1, \ldots, T$,

- Le joueur choisit un bras $k_t \in \mathcal{A}$.

- Sachant k_t, l'environnement génère une récompense $X_{k_t,t} \sim \nu_{k_t}$.

- Le joueur n'observe que la récompense $X_{k_t,t}$ à chaque instant.

De plus, dans le cadre stochastique, on suppose généralement que les récompenses générées par chaque bras sont indépendantes et identiquement distribuées (i.i.d.).

2.2.2 Le regret

La notion de regret peut être introduite de plusieurs manières équivalentes. En réalité, dire que l'objectif du joueur est de maximiser la récompense cumulée au fil du temps revient à dire que son but est de minimiser le regret cumulé, défini comme suit :

$$R(T) = \max_{k=1,\ldots,K} \sum_{t=1}^{T} X_{k,t} - \sum_{t=1}^{T} X_{k_t,t}.$$

Ici, $X_{k,t}$ désigne la récompense obtenue en jouant le bras k au temps t, et k_t est le bras choisi par le joueur à l'étape t. On définit également $\mu_k = \mathbb{E}[X_{k,t}]$ comme la récompense espérée du bras k, et on note $\mu^* \in \arg\max_{k=1,\ldots,K} \mu_k$ la meilleure récompense espérée. Dans le cadre stochastique, on s'intéresse souvent à une quantité appelée pseudo-regret, qui correspond à une comparaison avec le meilleur bras en espérance, plutôt qu'avec le bras optimal sur la suite effective des récompenses observées.

Définition 9. *Le pseudo regret est défini par,*

$$\tilde{R}(T) = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{k_t}\right].$$

On peut remarquer que le pseudo-regret est majoré par le regret espéré $\mathbb{E}[R(T)]$. Il existe une autre formulation du pseudo-regret, largement utilisée pour analyser un algorithme de bandit. Celle-ci repose sur les deux quantités suivantes :

$$\Delta_k = \mu^* - \mu_k, \qquad \text{et } N_k(t) = \sum_{s=1}^{t} \mathbb{1}_{\{k_t=k\}}.$$

où Δ_k désigne l'écart de sous-optimalité du bras k, c'est-à-dire la différence entre la récompense moyenne du meilleur bras μ^* et celle du bras k. $N_k(t)$ représente le nombre de fois où le bras k a été sélectionné par le joueur jusqu'au temps t. Avec ces quantités, le pseudo-regret peut être exprimé de manière équivalente comme suit :

$$\tilde{R}(T) = \sum_{k=1}^{K} \Delta_k \mathbb{E}[N_k(t)].$$

2.2.3 Les algorithmes de bandit standards

Comme mentionné précédemment, l'objectif d'un joueur est de maximiser la récompense cumulée au fil du temps. Pour cela, il doit trouver un équilibre entre exploration — c'est-à-dire recueillir des informations sur les bras — et exploitation — utiliser ces informations pour sélectionner le meilleur bras. Ce compromis est connu dans la littérature sur les bandits sous le nom de dilemme exploration-exploitation.

2

Une stratégie classique pour traiter ce dilemme est l'approche Explore-Then-Commit (ETC). Elle consiste à effectuer une phase d'exploration de longueur mK, au cours de laquelle chaque bras est tiré $m \geq 1$ fois. À l'issue de cette phase, le joueur exploite le bras ayant obtenu la meilleure récompense empirique.

Algorithm 4: L'algorithme `ETC`

Input: $m \geq 1$, paramètre

1 À l'étape t, chosir l'action,

$$k_t = \begin{cases} (t \bmod K) + 1 & \text{if } t \leq mK, \\ \arg\max_k \hat{\mu}_k(mK) & \text{if } t > mK. \end{cases}$$

Où $\hat{\mu}_k(t) = \frac{1}{N_k(t)} \sum_{s=1}^{t} \mathbb{1}_{\{k_s=k\}} X_{k,s}$. D'après [72, 43], `ETC` atteint la performance suivante :

Théorème 2. *Si* $1 \leq m \leq T/K$, *alors,*

$$\tilde{R}(T) \leq m \sum_{k=1}^{K} \Delta_k + (T - mK) \sum_{k=1}^{K} \Delta_l e^{-m\Delta_k^2}.$$

Une démonstration détaillée de ce résultat peut être trouvée dans [72, 43]. La borne sur le regret donnée dans la Théorème 2 met en évidence le compromis fondamental entre exploration et exploitation. Plus précisément, lorsque m est grand, l'algorithme consacre plus de temps à l'exploration, ce qui augmente le premier terme $\sum_{k=1}^{K} \Delta_k$ et, par conséquent, accroît le regret. À l'inverse, si m est trop petit, il y a une forte probabilité de sélectionner un bras sous-optimal lors de la phase d'exploitation, ce qui rend le second terme dominant et peut conduire à un regret important. Ainsi, le choix de m doit équilibrer ces deux effets et dépend à la fois de l'écart de sous-optimalité et de l'horizon temporel. Si l'horizon T est parfois connu à l'avance, il est rarement réaliste de supposer que l'on connaît l'écart de sous-optimalité. Néanmoins, on peut montrer qu'il existe un choix de m, dépendant de T, qui permet d'obtenir un regret indépendant du problème, d'ordre $\mathcal{O}(T^{2/3})$.

Bien que `ETC` adopte une stratégie simple pour équilibrer exploration et exploitation, il présente plusieurs limitations : une sensibilité au choix de la durée d'exploration, qui induit un risque de s'engager trop tôt ou trop tard ; la nécessité de connaître l'horizon T à l'avance pour calibrer m ; et une séparation rigide entre les phases d'exploration et d'exploitation, qui rend l'algorithme peu flexible si de nouvelles informations sont collectées au cours du processus. Pour surmonter ces limites, des stratégies plus adaptatives ont été proposées. L'une des approches les plus étudiées est celle de la borne supérieure de confiance (Upper Confidence Bound, `UCB`), fondée sur le principe d'optimisme. Plus précisément, pour chaque bras k, l'algorithme construit un intervalle de confiance sur sa récompense espérée à partir des observations passées. Il agit ensuite de manière optimiste, en supposant que la meilleure récompense possible dans cet intervalle est la vraie récompense, et

choisit en conséquence le prochain bras à tirer — c'est-à-dire celui ayant la borne supérieure de confiance la plus élevée.

Algorithm 5: L'algorithme `UCB`

Input: Pour les tours $t = 1, \ldots, K$, tirer le bras $k_t = t$.

1 **Pour** $t = K + 1, \ldots, T$ **faire**

2   Choisir

$$k_t \in \underset{k \in \{1, \ldots, K\}}{\arg\max} \left(\hat{\mu}_k(N_k(t-1)) + \sqrt{\frac{2\log(t)}{N_k(t-1)}} \right).$$

3   Observer la récompense et mettre à jour les bornes supérieures de confiance

---

D'après [72, 43], `UCB` atteint la performance suivante :

Théorème 3. *Pour n'importe quel horizon T,*

$$\tilde{R}(T) \le 3 \sum_{i=1}^{K} \Delta_i + \sum_{i : \Delta_i > 0} \frac{16\log(T)}{\Delta_i}.$$

La démonstration de ce résultat peut être consultée dans [72, 43]. La Théorème 3 montre que l'algorithme `UCB` atteint une borne sur le regret qui croît logarithmiquement avec T, ce qui représente une amélioration significative par rapport à la stratégie `ETC`. De plus, en utilisant le fait que le regret induit par le tirage du bras k ne peut pas dépasser $T\Delta_k$, cette borne dépendante de la distribution peut être transformée en une borne indépendante de la distribution d'ordre $\mathcal{O}(\sqrt{TK\log(T)})$. Ce taux est quasi optimal, la borne inférieure minimax étant de l'ordre de $\mathcal{O}(\sqrt{TK})$. Le facteur logarithmique supplémentaire peut d'ailleurs être éliminé en utilisant des algorithmes plus sophistiqués, comme la stratégie minimax optimale (Minimax Optimal Strategy).

Au-delà des algorithmes présentés dans cette section, de nombreuses autres stratégies ont été développées pour résoudre le problème du bandit manchot. Parmi elles, on trouve par exemple l'approche ϵ-greedy, qui effectue une exploration aléatoire avec une probabilité fixe, ou encore le Thompson Sampling, qui repose sur des principes bayésiens pour équilibrer exploration et exploitation. Par ailleurs, comme brièvement évoqué au début de cette section, le cadre des bandits a été étendu à plusieurs variantes pour traiter des applications réelles plus complexes, telles que les bandits linéaires, les bandits adversariaux, les bandits contextuels, ou encore les bandits combinatoires. Chacune de ces variantes introduit des défis supplémentaires et nécessite des algorithmes plus élaborés. Ce chapitre se concentre toutefois sur le problème canonique du bandit stochastique et sur les algorithmes les plus largement étudiés, qui constituent une base solide pour comprendre des modèles et techniques plus avancés.

3

Contributions (en français)

Contents

3.1   Aperçu de la thèse

L'objectif principal de cette thèse porte sur le matching (appariement) en ligne dans les graphes bipartis. Comme introduit dans Chapter 1, ce type de problème trouve de nombreuses applications dans des domaines variés. Dans cette thèse, l'accent est mis plus particulièrement sur la publicité en ligne. Dans ce cadre, le premier projet s'inspire des interactions entre utilisateurs et annonceurs sur les plateformes publicitaires. Alors qu'une grande partie de la littérature existante sur le matching en ligne dans le domaine de la publicité simplifie ou assouplit certaines contraintes clés, ce travail prend en compte une contrainte à la fois fondamentale et réaliste, omniprésente dans la publicité en ligne : les annonceurs disposent de budgets qui limitent le nombre de fois où leurs annonces peuvent être affichées aux utilisateurs. Ces budgets ne sont pas statiques : ils évoluent dans le temps. Par

exemple, des annonceurs, comme des marques de vêtements, peuvent augmenter leur budget lors de périodes de soldes, telles que le Black Friday, afin de maximiser leurs profits, puis le réduire par la suite. Ce projet vise donc à intégrer de telles contraintes budgétaires dynamiques dans le cadre classique du matching en ligne, et à analyser les performances de différents algorithmes dans ces conditions plus réalistes. Pour mieux comprendre l'impact de cette dynamique budgétaire sur les performances des algorithmes, une partie de l'analyse repose sur le modèle classique de graphe aléatoire d'Erdős–Rényi. Ce cadre théorique, bien que simplifié, permet de développer des intuitions utiles sur la manière dont l'évolution des budgets peut influencer les mécanismes de matching.

Bien que cette étude initiale ait permis de modéliser la dynamique des budgets, elle s'appuie sur des graphes bipartis relativement simples. Dans un contexte plus réaliste, notamment en publicité en ligne, les réseaux reliant les utilisateurs aux annonceurs sont nettement plus complexes. Ils présentent souvent des structures communautaires, où les nœuds sont regroupés en clusters, et les interactions sont régies par des affinités et préférences à l'échelle des groupes. Par exemple, les utilisateurs peuvent être regroupés selon leur âge, leurs centres d'intérêt ou leurs habitudes de navigation sur Internet, tandis que les annonceurs peuvent être classés selon leurs audiences cibles ou le type de publicité qu'ils diffusent. Étudier cette structure plus riche nécessite des modèles de graphes bipartis plus sophistiqués. Parmi ceux-ci, le modèle à blocs stochastiques (Stochastic Block Model, SBM) est particulièrement adapté, car il regroupe explicitement les nœuds des deux côtés du graphe en classes et définit les probabilités d'interaction en fonction de la compatibilité entre les classes. Ces considérations motivent le deuxième projet de cette thèse, qui se concentre sur le matching en ligne dans les modèles de blocs stochastiques. Ce projet a deux objectifs principaux : premièrement, comprendre comment les algorithmes classiques, tels que `Greedy` et `Balance`, se comportent lorsque le graphe sous-jacent présente une structure communautaire ; deuxièmement, intégrer une dimension d'apprentissage en supposant que les probabilités définissant le modèle de blocs stochastiques sont inconnues à l'avance et doivent être estimées au fil du temps. Cette hypothèse reflète un cadre plus réaliste, dans lequel la plateforme doit simultanément apprendre la structure des interactions entre utilisateurs et annonceurs, tout en prenant des décisions de matching en ligne. Pour traiter cette question, nous formulons le problème comme un problème de bandits manchots (multi-armed bandit) et proposons un algorithme basé sur la stratégie Explore-Then-Commit (`ETC`), combinée à l'algorithme `Balance`, afin d'estimer les probabilités et de construire le matching le plus large possible.

Au-delà des interactions entre utilisateurs et annonceurs, une composante cruciale de la publicité en ligne repose sur les mécanismes d'enchères, dans lesquels les annonceurs sont en concurrence en temps réel pour obtenir la possibilité d'afficher leurs publicités. Cette problématique a motivé le troisième projet de cette thèse, qui porte sur l'étude des enchères en ligne. Plus précisément, nous considérons un cadre dans lequel les impressions publicitaires (emplacements disponibles en ligne pour les annonces) sont vendues via des enchères au second prix, et le décideur doit sélectionner un sous-ensemble de campagnes à faire participer à chaque tour d'enchère.

Nous formulons ce problème comme un problème de bandits manchots structurés, où chaque bras correspond au choix du nombre d'annonceurs de la coalition sélectionnés pour participer à un tour donné. Le résultat de chaque enchère fournit un retour sous forme de récompense, indiquant si la coalition a remporté l'impression ainsi que le prix payé. Une observation clé de ce travail est que la récompense possède une structure particulière, que l'on peut exploiter pour améliorer l'efficacité de l'apprentissage. En nous appuyant sur cette propriété, nous avons développé des algorithmes qui en tirent parti afin d'équilibrer exploration et exploitation, et d'optimiser le gain cumulé de la coalition de campagnes au fil du temps.

Les sections suivantes présentent plus en détail les contributions de chacun de ces projets.

3.2 Online matching with budget refills

Dans ce projet [29], nous considérons un graphe biparti $G = (U, V, E)$, composé de deux ensembles de nœuds $U = \{1, \ldots, n\}$ et $V = \{1, \ldots, T\}$, pour $T, n \in \mathbb{N}^*$, ainsi qu'un ensemble d'arêtes $E \subseteq U \times V$. Les nœuds de U sont connus à l'avance, tandis que ceux de V sont découverts séquentiellement. Chaque nœud $u \in U$ dispose d'un budget $b_{u,t} \in \mathbb{N}$, qui évolue dynamiquement selon un processus qui sera précisé ultérieurement. Nous étudions deux cadres : le cadre adversarial, dans lequel le graphe G est déterministe et les budgets des nœuds de U évoluent selon une dynamique déterministe ; et un cadre stochastique, dans lequel le graphe est aléatoire et les dynamiques de budget sont régies par un processus stochastique.

Dans les sections suivantes, les principaux résultats obtenus pour chacun de ces cadres sont présentés.

3.2.1 Le cadre adversarial

Considérons un graphe biparti $G \in \mathcal{G}_{T,m}$, où $\mathcal{G}_{T,m}$ est une famille de graphes définie comme suit :

$$\mathcal{G}_{T,m} = \Big\{ (U, V, E, (\eta_{u,t})_{u \in U, t \in V}) \ : \ \forall t \in V, \ \exists u_t \in U \text{ tel que } \eta_{u_t,t} = \mathbf{1}_{\{t \equiv 0 \ (\mathrm{mod} \ m)\}}$$
$$\text{et } (u_t, t) \in E \Big\}.$$

où m est un paramètre que nous préciserons ultérieurement. Dans le cadre adversarial, il est essentiel d'imposer certaines restrictions sur le pouvoir de l'adversaire, afin d'éviter que le pire cas ne réduise le problème à un graphe sans refills effectifs. Nous adoptons donc les hypothèses suivantes : le calendrier de rechargement est fixe et connu à l'avance, et chaque nœud $t \in V$ a au moins un voisin dans U.

Le matching en ligne généré par un algorithme `ALG`, sur un graphe $G \in \mathcal{G}_{T,m}$, est un sous-ensemble d'arêtes qui peut être représenté par une matrice binaire $\mathbf{x} \in$

$\{0, 1\}^{n \times T}$, et qui doit satisfaire les contraintes suivantes : seules les arêtes de E peuvent être sélectionnées pour l'appariement, chaque nœud de V ne peut être apparié qu'au plus une fois, et un nœud de U ne peut être apparié au temps t que si son budget est strictement positif à cet instant.

Dans ce modèle, l'évolution du budget de chaque $u \in U$ dépend du fait que l'arête $(u, t) \in E$ soit incluse dans le matching en ligne, ce qui est indiqué par la variable binaire $x_{u,t} \in 0, 1$, ainsi que du rechargement, correspondant à l'ajout d'une unité tous les m pas de temps. Intuitivement, le rechargement modélise la régénération périodique de la capacité d'un nœud à participer à de nouveaux matchings. Formellement, cette dynamique peut être exprimée comme suit :

$$\forall u \in U, b_{u,t} = b_{u,t-1} - x_{u,t} + \mathbb{1}_{\{t \bmod m = 0\}}, \quad \text{and } b_{u,0} = b_0 \text{ pour } b_0 \geq 1.$$

$\mathtt{OPT}$ désigne le plus grand matching possible a posteriori, avec une matrice associée $\mathbf{x}^*$. Dans ce cas, le ratio de compétitivité est défini par

$$\mathtt{CR}^{\mathrm{adv}}(\mathtt{ALG}, \mathcal{G}_{T,m}) = \min_{G \in \mathcal{G}_{T,m}} \frac{\mathtt{ALG}(G)}{\mathtt{OPT}(G)}.$$

où $\mathtt{ALG}(G) = \sum_{u \in U} \sum_{t=1}^{T} x_{u,t}$ et $\mathtt{OPT}(G) = \sum_{u \in U} \sum_{t=1}^{T} x_{u,t}^*$ représentent les tailles du matching généré respectivement par $\mathtt{ALG}$ et $\mathtt{OPT}$.

Dans ce cadre, nous concentrons notre analyse sur l'algorithme $\mathtt{Balance}$, conçu à l'origine pour les environnements adversariaux. Notre objectif est de comprendre l'effet des budgets évolutifs sur le processus de matching. Pour cela, nous étudions les performances de $\mathtt{Balance}$ dans deux régimes distincts : un régime où les rechargements de budget sont peu fréquents, spécifiquement lorsque $m = \omega(\sqrt{T})$, et un autre où ils sont fréquents, c'est-à-dire lorsque $m = o(\sqrt{T})$. Le premier résultat de ce projet concerne le premier cas, et est énoncé ci-dessous.

Théorème 4. *En supposant que les budgets initiaux sont $b_{1,0} = b_{2,0} = \cdots = b_{n,0} = b_0 \geq 1$. Si $m = \omega(\sqrt{T})$ et $b_0(b_0 + 1)^{b_0} \leq m$, alors,*

$$\sup_{\mathtt{ALG}:deterministic} \mathtt{CR}^{\mathrm{adv}}(\mathtt{ALG}, \mathcal{G}_{T,m}) \leq 1 - \frac{1}{\left(1 + \frac{1}{b_0}\right)^{b_0}} + o_T(1).$$

La démonstration de ce résultat s'appuie sur les travaux de [62] ; plus de détails sont fournis dans Section 7.A.1. L'intuition principale de ce théorème est que, lorsque les rechargements de budget sont relativement rares, le pire cas est arbitrairement proche de celui du b_0-matching (plus de détails sur la définition du $b-$matching sont donnés dans Chapter 2). Cela est intuitif : dans un tel régime, la majorité des matchings ont lieu au début, épuisant les budgets initiaux, tandis que les rechargements occasionnels — étant trop espacés — ont peu d'impact sur le

3

résultat global. Par conséquent, ces rechargements rares n'améliorent pas significativement le nombre total de matchings.

En revanche, lorsque les rechargements de budget sont fréquents, la situation change de manière significative. Contrairement au scénario précédent, où le ratio de compétitivité CR était borné par $1 - \frac{1}{e}$, la borne s'améliore ici pour atteindre environ 0.73.

Théorème 5. *En supposant que les budgets initiaux sont $b_{1,0} = b_{2,0} = \cdots = b_{n,0} = b_0 \geq 1$. Pour $m = o(\sqrt{T})$ et $mb_0 = o(T)$, alors,*

$$\text{CR}^{\text{adv}}(\texttt{Balance}, \mathcal{G}_{T,m}) \leq \underbrace{1 - \frac{(1-\alpha)}{e^{(1-\alpha)}}}_{} \simeq 0.73325\ldots + o_{m,T}(1).$$

où α est défini par $\frac{1}{2} = \int_0^\alpha \frac{xe^x}{1-x}\mathrm{d}x$.

La démonstration complète est fournie dans Section 7.A.2. Elle repose sur la construction d'un graphe adversarial exploitant l'incapacité de l'algorithme à prédire quels nœuds resteront disponibles. Initialement, tous les nœuds de U sont connectés aux nœuds entrants de manière à ce que Balance distribue les matchings de façon uniforme et accumule du budget via des rechargements fréquents. Durant cette période, Balance et l'algorithme optimal hors ligne OPT ont un comportement identique. La différence apparaît par la suite, lorsque l'adversaire supprime la majorité des nœuds de U, en particulier ceux où Balance avait accumulé du budget. Contrairement à Balance, qui ne peut pas anticiper quels nœuds seront supprimés, l'algorithme optimal hors ligne dispose d'une connaissance parfaite a priori des nœuds éliminés. Il peut donc allouer le budget uniquement aux nœuds qui resteront disponibles, garantissant qu'aucun budget n'est gaspillé sur des nœuds supprimés. Cette suppression systématique contraint Balance à gaspiller une grande partie de son budget accumulé, l'empêchant de transformer les rechargements fréquents en un plus grand nombre de matchings. Par conséquent, même si les rechargements rendent le problème moins contraint, le ratio de compétitivité ne peut pas dépasser environ 0.73.

Le dernier résultat de cette section montre qu'aucun algorithme déterministe ne peut surpasser Balance dans ce cadre.

Théorème 6. *En supposant que les budgets initiaux sont $b_{1,0} = b_{2,0} = \cdots = b_{n,0} = b_0 \geq 1$. Pour $m = o(\sqrt{T})$,*

$$\sup_{\texttt{ALG}} \mathbb{E}\left[\text{CR}^{\text{adv}}(\texttt{ALG}, \mathcal{G}_{T,m})\right] \leq \text{CR}^{\text{adv}}(\texttt{Balance}, G^{th.2}) + o_T(1).$$

où l'espérance est prise par rapport à l'aléa de ALG.

L'intuition derrière ce résultat est que maintenir les budgets équilibrés entre les nœuds disponibles de U est la meilleure stratégie possible pour un algorithme dans le graphe adversarial utilisé dans la démonstration du Théorème 6. En effet,

l'adversaire supprime les nœuds de U un par un, en commençant par ceux ayant le budget le plus élevé, sans jamais rendre à nouveau disponible un nœud déjà supprimé.

3.2.2 Le cadre stochastique

Le problème de matching en ligne avec rechargement de budget dans le cadre stochastique est étudié selon le cadre suivant :

1. Le graphe aléatoire est un modèle d'Erdős–Rényi $G(n, T, p)$, c'est-à-dire un graphe biparti avec n sommets d'un côté, T de l'autre, et chaque arête potentielle $(u, t) \in U \times V$ apparaît indépendamment avec une probabilité p.

2. Le régime considéré est le régime parcimonieux (sparse), dans le sens où $p = \frac{a}{n}$ avec $a > 0$. Ce choix est motivé par les applications à la publicité en ligne, où le nombre d'utilisateurs dépasse largement celui des campagnes publicitaires, et seule une petite portion des utilisateurs est éligible à participer.

3. La séquence de rechargements est une réalisation de variables aléatoires de Bernoulli indépendantes, de paramètre β/n, pour un certain $\beta > 0$.

Comme souligné précédemment, chaque nœud $u \in U$ est associé à un budget $b_{u,t} \in \mathbb{N}$. Nous ajoutons l'hypothèse supplémentaire selon laquelle le budget maximal par nœud est borné par un certain $K \in \mathbb{N}^*$, de sorte que la dynamique des budgets s'exprime désormais comme suit :

$$b_{u,t} = \min(K, b_{u,t-1} - x_{u,t} + \eta_{u,t}), \qquad \text{avec } b_{u,0} = b_0 \geq 1.$$

Imposer une borne supérieure K sur les budgets est motivé par deux considérations : cela reflète des contraintes pratiques dans les applications réelles (comme la publicité en ligne), et cela simplifie l'analyse en réduisant le problème au suivi d'un nombre fini d'états budgétaires dans le temps.

Comme défini précédemment, un matching en ligne sur G généré par un algorithme ALG est un sous-ensemble d'arêtes, représenté par une matrice binaire $\mathbf{x} \in \{0, 1\}^{n \times T}$, et doit satisfaire les mêmes contraintes introduites dans le cadre adversarial.

Dans le cadre stochastique, la performance d'un algorithme peut être mesurée soit par la taille espérée de matching qu'il produit, soit par le ratio entre les tailles espérées des matchings produits par ALG et OPT. Formellement, les différentes quantités que nous considérerons sont :

$$\texttt{CR}^{\text{sto}}(\texttt{ALG}, \mathcal{D}) = \frac{\mathbb{E}_{G \sim \mathcal{D}}[\texttt{ALG}(G)]}{\mathbb{E}_{G \sim \mathcal{D}}[\texttt{OPT}(G)]}, \quad \text{ou la taille du matching} = \mathbb{E}_{G \sim \mathcal{D}}[\texttt{ALG}(G)].$$

où $\mathcal{D}$ désigne la distribution du graphe considéré ainsi que celle des rechargements. Bien que les définitions de $\mathtt{ALG}(G)$ et $\mathtt{OPT}(G)$ restent inchangées, nous expliciterons la dépendance à l'horizon temporel T si nécessaire en écrivant $\mathtt{ALG}(G, T)$ et $\mathtt{OPT}(G, T)$.

Le premier résultat de cette section identifie la taille asymptotique du matching généré par l'algorithme $\mathtt{Greedy}$ sur le modèle biparti d'Erdős–Rényi avec rechargement de budgets. Il établit qu'avec forte probabilité, la taille du matching généré par $\mathtt{Greedy}$ est proche de la solution d'un système d'équations différentielles ordinaires.

Théorème 7. *Avec probabilité* $1 - \mathcal{O}\left(n^{1/4} \exp(-a^3 n^{1/4})\right)$, *la taille du matching créé par* $\mathtt{Greedy}$, *notée* $\mathtt{Greedy}(G, T)$ *satisfait :*

$$\mathtt{Greedy}(G, T) = nh(T/n) + \mathcal{O}(n^{3/4}),$$

et,

$$\frac{\mathbb{E}[\mathtt{Greedy}(G, T)]}{n} - h(T/n) \underset{n \to +\infty}{\to} 0.$$

où $h(\tau)$ *est solution de l'équation suivante,*

$$\dot{h}(\tau) = 1 - e^{-a(1 - z_0(\tau))}, \qquad \frac{1}{n} \le \tau \le \frac{T}{n}.$$

et $z_0(\tau)$ *satisfait le système suivant :*

$$\begin{cases} \dot{z}_0(\tau) = -z_0(\tau)\beta + \frac{z_1(\tau)}{1 - z_0(\tau)}(1 - e^{-a + az_0(\tau)}) & \textit{for } k = 0, \\ \dot{z}_k(\tau) = (z_{k-1}(\tau) - z_k(\tau))\beta + (z_{k+1}(\tau) - z_k(\tau))\frac{1 - e^{-a + az_0(\tau)}}{1 - z_0(\tau)} & \textit{for } 1 \le k \le K - 1, \\ \dot{z}_k(\tau) = \beta\, z_{k-1}(\tau) - z_k(\tau)\frac{1 - e^{-a(1 - z_0(\tau))}}{1 - z_0(\tau)} & \textit{for } k = K, \\ \sum_{k=0}^{K} z_k(\tau) = 1. \end{cases}$$

$$(3.1)$$

La démonstration de ce résultat repose sur le suivi de deux processus stochastiques couplés : le nombre cumulé de matchings au fil du temps, et l'évolution des nœuds de U ayant un budget k. On démontre que ces processus satisfont les hypothèses de [98], ce qui permet d'établir que la dynamique discrète et aléatoire du système converge, avec haute probabilité, vers la trajectoire déterministe définie par Equation (3.1).

Une fois cette convergence établie, l'étape suivante consiste à comprendre le comportement asymptotique du système différentiel lui-même. Bien qu'il soit généralement difficile d'obtenir une solution explicite, selon la littérature sur les équations différentielles, une approche alternative consiste à étudier si le système converge vers un état stationnaire stable, et à relier le comportement limite de $\mathtt{Greedy}(G, T)$ à cet équilibre. Nous analysons cela dans deux cas : $K = 1$, où le système est réduit à deux équations, et l'état stationnaire est montré comme étant exponentiellement stable ; et le cas général $K \ge 1$, où nous prouvons que la solution stationnaire est asymptotiquement stable.

Corollaire 1. *Pour $K \geq 1$, avec probabilité d'au moins $1 - 2\exp(-a^2 n^{\frac{3}{2}}/8T)$,*

$$|\mathtt{Greedy}(G, T) - nh^*(T/n)| \leq o(T).$$

et,

$$\frac{\mathbb{E}[\mathtt{Greedy}(G, T)]}{n} - h^*(T/n) \underset{n \to +\infty}{\to} 0.$$

avec $h^*(x) = \int_{1/n}^{x}(1 - e^{-a(1-z_0^*)})\mathrm{d}\tau = \left(x - \frac{1}{n}\right)(1 - e^{-a(1-z_0^*)})$, et z_0^* est une unique solution de $\sum_{k=0}^{K} z_0^* \left(\frac{\beta}{g(z_0^*)}\right)^k = 1$ avec $g(z_0^*) = \frac{1 - e^{-a(1-z_0^*)}}{1 - z_0^*}$.

Pour $K = 1$, avec probabilité d'au moins $1 - 2\exp(-a^2 n^{\frac{3}{2}}/8T)$,

$$\left|\mathbb{E}[\mathtt{Greedy}(G, T)] - T(1 - e^{-a(1-z_0^*)})\right| \leq c\frac{T}{(\log(T))^{3/4}} = o(T).$$

où $z_0^* = \frac{1}{\beta} - \frac{1}{a}W\left(\frac{a}{\beta}e^{-a\left(1-\frac{1}{\beta}\right)}\right)$, avec $W(\cdot)$ la fonction Lambert, et c une constante universelle.

La différence entre ces deux cas réside dans la nature de la stabilité : lorsque $K = 1$, le système présente une stabilité exponentielle, ce qui entraîne une convergence rapide vers l'état stationnaire. En revanche, pour $K \geq 1$, la stabilité n'est que asymptotique, ce qui ne garantit pas une convergence aussi rapide.

Du côté du ratio de compétitivité $\mathtt{CR}$, le dernier résultat principal de cette section montre que le ratio de compétitivité de l'algorithme $\mathtt{Greedy}$ dans ce cadre converge vers 1 lorsque T, K et n deviennent grands.

Théorème 8. *Pour tout $\alpha, \beta > 0$, le ratio de competitivité tend vers 1, lorsque T, K, n tendent vers l'infini :*

$$\lim_{K,n \to +\infty} \lim_{T \to +\infty} \mathtt{CR}^{\mathrm{sto}}(\mathtt{Greedy}, \mathcal{D}) = 1.$$

Cette conclusion repose sur le développement d'une borne inférieure sur $\mathtt{CR}^{\mathrm{sto}}$, qui dépend de $(z_0^*, \ldots, z_K^*)$, la solution stationnaire de Equation (3.1). Cette borne est obtenue par un calcul exact de la taille du matching réalisé par l'algorithme $\mathtt{Greedy}$, ainsi que par une borne supérieure sur la taille du matching généré par $\mathtt{OPT}$. À mesure que T, K et n croissent, l'écart entre $\mathtt{Greedy}$ et l'algorithme optimal devient négligeable, ce qui fait converger le ratio de compétitivité vers 1.

3.3 Online matching on stochastic block model

Nous étudions le problème de matching biparti sur un modèle à blocs stochastiques (SBM) [30], où le graphe biparti $G = (U, V, E)$ est défini par un ensemble de nœuds

hors ligne $U = [n] = \{1, \ldots, n\}$ et un ensemble de nœuds en ligne $V = [T] = \{1, \ldots, T\}$, où $T, n \in \mathbb{N}^*$. Les arêtes $(u, t) \in E$ sont incluses de manière indépendante en fonction des classes latentes. Chaque nœud hors ligne $u \in U$ et chaque nœud en ligne $t \in V$ se voit attribuer une classe issue respectivement des ensembles $\mathcal{C} = [C]$ et $\mathcal{D} = [D]$, avec $D, C \in \mathbb{N}^*$, tirées indépendamment selon les distributions μ et ν. Étant données les classes, l'arête (u, t) apparaît dans E avec une probabilité $p_{u,t} = p\big(c(u), d(t)\big) \in [0, 1]$, où $p = (p(c, d))_{c,d \in \mathcal{C} \times \mathcal{D}}$ est une matrice d'affinité entre classes. Dans ce travail, nous nous concentrons sur le *régime parcimonieux* (sparse regime). Plus précisément, nous supposons l'existence d'une matrice $a = (a_{c,d}) \in \mathbb{R}+^{C \times D}$ telle que $p(c, d) = \frac{a_{c,d}}{n}$, avec $a_{c,d} \leq a$ pour tout $c \in \mathcal{C}$, $d \in \mathcal{D}$, où $a \in (0, n)$. Cette hypothèse garantit que les degrés espérés des nœuds hors ligne sont bornés lorsque $n \to \infty$, ce qui reflète des contraintes réalistes du monde réel.

Pour formaliser le processus, on pose $T = \alpha n$ pour un certain $\alpha > 0$, et on note b_c la proportion de nœuds hors ligne appartenant à la classe c. Pour un algorithme de matching donné ALG, on définit la variable booléenne $m_u(t)$ comme étant égale à 1 si, et seulement si, le sommet u a été apparié par ALG avant l'arrivée du sommet t (sinon $m_u(t) = 0$). On note également $\mathcal{N}_c := \{u \in U, \ c(u) = c\}$ l'ensemble des nœuds de classe $c \in \mathcal{C}$, et $\mathcal{M}_c(t) := \{u \in \mathcal{N}_c, m_u(t) = 1\}$ l'ensemble des sommets de classe c appariés avant l'arrivée du sommet $t \in V$ (on note $M_c(t)$ sa cardinalité), et enfin $M(t) := \sum_{c \in \mathcal{C}} M_c(t)$ la taille totale du matching construit jusqu'à l'instant t. Enfin, nous noterons e_i le i-ème vecteur de base de $\mathbb{R}^C$.

Selon que les probabilités de compatibilité $p_{u,t}$ soient connues à l'avance ou doivent être apprises au fil du temps, différentes approches algorithmiques et analyses théoriques seront développées dans les sections suivantes.

3.3.1 $p_{u,t}$ **connus**

Le premier résultat de cette partie concerne les performances de l'algorithme Myopic, qui prend des décisions purement gloutonnes (greedy), sans tenter d'anticiper la disponibilité future. Lorsqu'un sommet $t \in V$ arrive, l'algorithme sélectionne une classe $c_t \in \mathcal{C}$ selon une distribution de probabilité fixe, puis tente d'apparier le sommet à un nœud disponible appartenant à cette classe. Cette sélection est faite sans vérifier au préalable si des nœuds de cette classe sont effectivement disponibles à l'instant t.

Pour comprendre comment cette stratégie simple se comporte dans le modèle à blocs stochastiques, nous fournissons une approximation de la taille espérée du matching qu'elle produit. Plus précisément, nous montrons que l'évolution du matching sous Myopic suit de près la solution d'une équation différentielle qui capture la dynamique du processus au cours du temps.

Théorème 9. *Soit $y_c : [0, \alpha] \to \mathbb{R}$, la solution du système d'équations différentielles*

suivant :

$$\begin{cases} \dot{y}_c(s) & = \sum_{d=1}^{D} \left(1 - e^{-a_{c,d}(b_c - y_c(s))}\right) Q^*(c,d), \\ y_c(0) & = 0. \end{cases} \tag{3.2}$$

Alors, pour chaque classe $c \in \mathcal{C}$, la taille du matching $M_c(t)$ produite par `Myopic` *satisfait, pour tout $t \in [T]$*

$$\left| \frac{M_c(t)}{n} - y_c(t/n) \right| \leq \frac{3L_c e^{\alpha L_c}}{n^{1/3}}, \;\; where \; L_c = \sum_{d=1}^{D} a_{c,d} Q^*(c,d).$$

avec une probabilité au moins $1 - 2Ce^{-n^{1/3}L_c^2/8\alpha}$. De plus, pour $c \in [C]$, on a $y_c(t) = \tilde{y}_c(t) - e_c(t)$, où $e_c(0) = 0$, et $\tilde{y}_c(t) = b_c - b_c \exp\left(-tL_c\right)$, et e_c satisfait,

$$e_c(t) \leq \frac{J_c}{L_c}(1 - e^{-L_c t}).$$

où $J_c = \frac{b_c^2}{2} \sum_{d=1}^{D} a_{c,d}^2 Q^*(c,d)$.

Ce résultat montre que la taille du matching réalisé par `Myopic` dans ce modèle peut être approximée par la solution du système 3.2. En raison de la complexité du système, obtenir une solution explicite de 3.2 n'est pas envisageable. Nous dérivons donc une solution approchée avec une borne d'erreur contrôlée, garantissant que l'approximation reste précise.

Le second résultat de cette partie repose sur les limitations de `Myopic` et prend en compte la disponibilité des nœuds. Nous étudions ici `Ex-ante Balance`, qui choisit une classe c maximisant la probabilité qu'au moins un nœud non apparié soit disponible et connecté au nœud entrant (l'algorithme détaillé se trouve dans Algorithm 12). Le résultat suivant montre que la taille du matching générée par `Ex-ante Balance` est, avec forte probabilité, proche d'une inclusion différentielle décrite dans Section 8.A.

Théorème 10. *Soit m l'unique solution de l'inclusion différentielle suivante :*

$$\dot{m} \in F(m) := conv\left\{ f_{c,b_c}(m_c)e_c \; ; \; c \in \arg\max_{k \in [C]} f_{k,b_k}(m_k) \right\},$$

qui est l'enveloppe convexe des applications

$$f_{c,b_c}(x) = \sum_{d=1}^{D} (1 - e^{-a_{c,d}(b_c - x)})\nu(d).$$

Alors, le matching construit par `Balance` *satisfait, pour tout $t \in [T]$ and $c \in \mathcal{C}$,*

avec une probabilité au moins $1 - \frac{b\alpha}{N\epsilon^2}$,

$$\left|\frac{M_c(t)}{n} - m_c(t/n)\right| \le \min(\alpha, e^{L\alpha}/\sqrt{2L})\sqrt{A_{\alpha,c}/n + \delta_c B_{\alpha,c} + \epsilon C_{\alpha,c}},$$

où les constantes sont définies comme suit : $L = \max_{c \in [C]} \sum_{d=1}^{D} a_{c,d}\nu(d)$, $\delta_c = \frac{1}{n}\sum_{d=1}^{D} \frac{a_{c,d}}{e}\nu(d)$, $K_\alpha = (c\alpha + \epsilon)e^{c\alpha}/c$, ϵ *défini dans Lemma 35 et c dans Lemma 31,* $U_c = \sum_{d=1}^{D}(1 - e^{-a_{c,d}b_c})\nu(d)$, $A_{\alpha,c} = U_c(U_c^2 + \frac{14U_c}{3} + 2K_\alpha)$, $B_{\alpha,c} = 2U_c^2 + 4L\delta_c + 12K_\alpha$, $C_{\alpha,c} = 2U_c^2 + 4L\epsilon + 8K_\alpha$.

Ici, le système converge vers une inclusion différentielle plutôt qu'une équation différentielle ordinaire, en raison des discontinuités dans la règle de décision de l'algorithme. Contrairement à `Myopic`, dont les décisions de matchings sont lipschitziennes et bien approchées par une ODE, `Ex-ante Balance` sélectionne dynamiquement la classe maximisant la probabilité de succès. Cela introduit des changements abrupts dans l'évolution du système, brisant les hypothèses de continuité nécessaires à la convergence vers une ODE.

Pour traiter ce problème, nous caractérisons le comportement limite de l'algorithme `Ex-ante Balance` via une inclusion différentielle, plus adaptée à ce contexte, car elle capture l'ensemble des trajectoires possibles du système. Bien qu'il soit généralement difficile d'obtenir une solution explicite d'une inclusion différentielle, nous exploitons ici le fait que `Ex-ante Balance` équilibre implicitement les probabilités de matching entre les classes. En nous appuyant sur cette structure, nous dérivons une expression explicite du comportement limite. Des détails supplémentaires sont fournis dans Chapter 8.

3.3.2 $p_{u,t}$ inconnus

Dans ce cas, nous étudions le cadre où les probabilités de connexion $a_{c,d}$ sont inconnues et doivent être estimées en ligne. Cela transforme le problème en un cadre de bandits, où chaque classe $c \in \mathcal{C}$ peut être vue comme un bras. Lorsqu'une classe c_t est sélectionnée à l'instant t, une récompense de Bernoulli est observée — indiquant si un matching réussi a eu lieu entre le nœud entrant et un nœud disponible dans la classe c_t. Contrairement au cadre classique des bandits, la récompense espérée associée à un bras n'est pas stationnaire : elle dépend de la dynamique du système, notamment de la disponibilité des nœuds dans chaque classe. En conséquence, il n'existe pas de notion de "meilleur bras", ce qui rend le problème plus complexe. Une solution consiste à introduire l'algorithme `ETC − balance`, qui combine une stratégie d'exploration fixe de type Explore-Then-Commit (ETC) avec la règle de sélection `Ex-ante Balance`. La performance de `ETC − balance` est évaluée en terme de regret, défini comme l'écart entre le nombre cumulé des matchings obtenus par un oracle connaissant parfaitement les probabilités et celui obtenu par `ETC − balance`. Lorsque la phase d'exploration dure $T_{\text{explore}} = T^{\frac{q+3}{4}}$ étapes, pour tout $0 < q < 1$, le regret satisfait la borne suivante :

Théorème 11. *Soit* $R(T) = \sum_{i \in \mathcal{C}} M_i(T) - \hat{M}_i(T)$ *le regret de* `ETC − balance`. *Supposons que la phase d'exploration dure* $T_{explore} = T^{\frac{q+3}{4}}$, *for some* $0 < q < 1$. *Alors, le regret vérifie :*

$$R(T) = \mathcal{O}(T^{\frac{q+3}{4}}).$$

3.4 Optimizing the coalition gain in Online Auctions with Greedy Structured Bandits

Dans ce travail [15], nous considérons T impressions publicitaires vendues via des *enchères au second prix* (voir [69] pour plus de détails). Lors de l'enchère $t \in [T]$, chaque participant (enchérisseur) soumet une enchère pour l'impression en fonction de sa propre valeur (stochastique) et de celle-ci. L'enchérisseur ayant la mise la plus élevée remporte l'objet et paie un prix égal à la deuxième plus haute enchère. Le *décideur* gère $N \in \mathbb{N}^*$ campagnes publicitaires formant une *coalition*. À l'instant t, deux groupes d'enchérisseurs participent à l'enchère : (1) $n_t \in [N]$ enchérisseurs issus de la coalition, choisis par le décideur *ex ante* — c'est-à-dire sans connaître les valeurs réalisées des enchérisseurs —, et (2) $p \in \mathbb{N}^*$ autres enchérisseurs, que nous appelons la *concurrence*. Lorsqu'un enchérisseur de la coalition remporte l'enchère, le décideur observe la valeur réalisée du gagnant (également appelée *enchère gagnante*). Dans ce projet, nous supposons que tous les enchérisseurs sont identiques, leurs valeurs étant des échantillons i.i.d. d'une distribution supportée sur $[0, 1]$, dont la fonction de répartition est notée F. Sous cette hypothèse, la récompense espérée du décideur à l'instant t est donnée par $r(n_t)$, où r est défini par :

$$r : n \in [N] \mapsto r(n) \coloneqq \mathbb{E}_{\mathbf{v}=(v_i)_{i \in [n+p]} \sim F \times \cdots \times F}\left[(\mathbf{v}_{(1)} - \mathbf{v}_{(2)}) \mathbb{1}\left\{ \arg\max_{i \in [n+p]} v_i \in [n] \right\} \right]. \quad (3.3)$$

où $\mathbf{v}_{(1)}$ et $\mathbf{v}_{(2)}$ désignent respectivement les première et deuxième plus grandes valeurs de $\mathbf{v}$, et $[n]$ est utilisé pour abréger $1, \ldots, n$. Ce cadre conduit naturellement à un problème de type bandit manchot (multi-armed bandit problem), dans lequel le décideur choisit séquentiellement des *bras*, $n_1, \ldots, n_T \in [N]$, et cherche à minimiser son *regret cumulé espéré* $\mathcal{R}(T)$, défini par :

$$\mathcal{R}(T) = \sum_{t \le T} r(n^*) - r(n_t), \quad \text{avec} \quad n^* = \arg\max_{n \in [N]} r(n). \quad (3.4)$$

En utilisant des techniques issues des statistiques d'ordre, l'équation Equation (3.3) peut être réécrite de manière équivalente comme suit :

$$n \in [N] \mapsto r(n) = n \int_0^1 F^{p+n-1}(x) - F^{p+n}(x) \mathrm{d}x. \quad (3.5)$$

Cette expression sous forme d'intégrale met en évidence que la fonction de récompense définie peut être unimodale pour certains choix de F. Dans la suite de ce

travail, nous nous concentrons sur les distributions assurant cette unimodalité.

En considérant cette structure, la première étape de ce projet a consisté à estimer
r. D'après Equation (3.5), on remarque qu'estimer $F^{p+n-1}(x)$ et $F^{p+n}(x)$ suffit pour
approximer $r(n)$, ce qui nous amène à construire $\hat{r}_k(n)$, un estimateur de $r(n)$ à partir
d'échantillons provenant de n'importe quel bras k, en utilisant les puissances de la
fonction de répartition empirique, définie par :

$$\widehat{r}_k(n) = n \int_0^1 \left(\widetilde{F}_{k+p}^{n+p-1}(x) - \widetilde{F}_{k+p}^{n+p}(x) \right) \mathrm{d}x, \tag{3.6}$$

où $\widetilde{F}_{k+p}^{\ell} : x \mapsto (\widehat{F}_{k+p}(x))^{\frac{\ell}{k+p}}, ; \widehat{F}_{k+p} : x \mapsto \frac{1}{m_k} \sum_{j=1}^{m_k} \mathbb{1}\{w_{k,j} \leq x\}$ est la fonction de
répartition empirique de $\overline{W_k}$, avec $\overline{W_k} = (w_{k,1}, \ldots, w_{k,m_k})$ représentant les valeurs
observées après avoir joué le bras k et remporté m_k enchères. En supposant l'uni-
modalité, et en restreignant l'ensemble des bras utilisés pour estimer la récompense,
on obtient le résultat de concentration suivant :

Théorème 12. *Soit $n \in [N]$ et $k \in \mathcal{V}(n)$. Soit $\widehat{r}_k(n)$ défini selon Equation* (3.6)
*à partir de m_k échantillons collectés via k. Alors, il existe des constantes $\beta_{k,n}$
(dépendant de n, k, p) et $\xi_{k,n,F}$ (dépendant aussi de F) telles que, avec probabilité au
moins $1 - \delta$,*

$$|\widehat{r}_k(n) - r(n)| \leq \beta_{k,n} \sqrt{\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}} + n \times \xi_{k,n,F} \left(\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k} \right)^{\frac{n+p-1}{k+p}}. \tag{3.7}$$

*De plus, ces constantes admettent des bornes universelles pour tout n, k, p, F. Par
exemple, si $m_k \geq 4$, alors on a $\beta_{k,n} \leq 33$ et $\gamma_{k,n,F} \leq 100$.*

Avec ce résultat de concentration en main, nous proposons deux stratégies pour
résoudre le problème précédemment défini : la première stratégie est Local-Greedy
(`LG`), une adaptation naturelle d'une politique standard dans les bandits unimodaux,
`OSUB` [31]. L'idée principale de `OSUB` est de jouer une stratégie de type `UCB` localement
autour d'un bras de référence, et de se rapprocher progressivement du bras optimal
$n^\star$ en déplaçant ce bras de référence dans sa direction. Avec `LG`, ce principe est
adapté pour exploiter efficacement la structure du problème : à chaque tour t,
`LG` définit un bras de référence ℓ_t, appelé *leader*, mais joue de manière *gloutonne*
dans le *voisinage* $\mathcal{V}(\ell_t)$, à l'aide d'estimations simples basées uniquement sur les
échantillons collectés via ℓ_t. En outre, une *condition d'échantillonnage*, implémentée
via un paramètre $\alpha \in (0, 1)$, est utilisée pour garantir une bonne concentration des
estimateurs. La seconde stratégie, appelée `GG`, repose sur une idée intuitive : elle joue
`LG` uniquement si elle peut identifier, avec haute probabilité, dans quel segment de la
fonction de récompense se trouve le meilleur bras. Pour cela, `GG` utilise une procédure
de type Successive Elimination [39] sur un *sous-ensemble* de bras formant une *grille
de référence* notée $\mathcal{S}$, qui sert à discrétiser grossièrement l'espace des actions afin
de localiser efficacement la région contenant le bras optimal. Une fois cette région

identifiée, `GG` applique `LG` dans ce segment. Cette approche hybride combine une exploration globale grossière avec une exploitation locale fine, permettant d'identifier efficacement et de manière fiable les bras les plus performants.

Pour ces deux stratégies, on obtient les résultats suivants sur le regret :

Théorème 13. *Soit* $\Delta := \min_{n \in [N-1]} |r(n+1) - r(n)|$. *Sous l'hypothèse d'uni-modalité et avec* $\alpha = (\log_{3/2} N + 1)^{-1}$, *le regret de* `LG` *est borné par une **constante dépendante du problème**: il existe des constantes* $(C_n)_{n \in [N] \setminus \{n^\star\}}$, *chacune vérifiant* $C_n = \widetilde{\mathcal{O}}_N \left(\frac{\Delta_n}{\Delta^2} \right)$, *tel que* $\mathcal{R}_T \leq \sum_{n \in [N] \setminus n^\star} C_n$.

De plus, si l'ensemble des bras forme un unique voisinage d'estimation, c'est-à-dire si $\forall n \in [N] : \mathcal{V}(n) \supset [N]$, *alors chaque constante* C_n *peut être raffinée en* $\widetilde{\mathcal{O}}_n (\Delta_n^{-1})$, *ce qui donne* $\mathcal{R}_T = \widetilde{\mathcal{O}}(\sqrt{NT})$, *résultat qui reste valable même si la fonction de récompense n'est pas unimodale.*

Supposons que `GG` *soit paramétré avec un niveau de confiance* $\delta_t = \frac{1}{N^2 t^3}$ *et* $\alpha = 1/4$. *Alors, pour tout* $T \in \mathbb{N}$, *on a :*

$$
\mathcal{R}_T = \widetilde{\mathcal{O}}_N \left(\sum_{n \in \mathcal{B}^\star} \frac{1}{\Delta_n} + \sum_{n \in \mathcal{S}} \frac{\log(T)}{\Delta_n} \wedge \Delta_n \left(\frac{\mathbb{1}\{n < n^\star\}}{\Delta_{v_l(n^\star)}^2} + \frac{\mathbb{1}\{n > n^\star\}}{\Delta_{v_r(n^\star)}^2} \right) \right) .
$$

De plus, on a $\mathcal{R}_T = \widetilde{\mathcal{O}} \left(\sqrt{(K + |\mathcal{B}^\star|)T} \right)$, *avec* $K = \lfloor \log_{3/2}(N) \rfloor$, *et* $\mathcal{B}^*$ *désigne le segment optimal de la grille de référence.*

Alors que les approches classiques de bandits donnent des bornes dépendant de T, les algorithmes `GG` et `LG` présentés ici fournissent des bornes constantes dépendant du problème. En outre, `GG` et `LG` évitent une dépendance quadratique en N pour les grands T grâce aux nouvelles bornes de concentration. Globalement, bien que `GG` offre les meilleures garanties théoriques, `LG` bénéficie de meilleures constantes, ce qui le rend plus adapté à la plupart des applications pratiques (voir la discussion à la fin de Section 9.3 et les résultats expérimentaux en Section 9.D).

4

Introduction (in english)

The concept of matching is not a recent idea. Long before algorithms, or even the invention of computers, human societies developed their own methods to connect needs with resources, seekers with providers, and individuals with opportunities. In ancient learning systems, elders selected young people to train based on their intuition, lineage, or skills. In traditional communities, matchmakers arranged marriages by balancing social status, compatibility, and family interests. In villages, children were assigned to tutors or teachers through communal decisions, often influenced by order of arrival or the urgency of need. These forms of matching were local, personal, and time-consuming. They relied heavily on knowledge, memory, and human relationships. The matchmaker, the teacher, the master — they were not just parts of the system: **they were the system**. Decisions were made carefully, leaving room for negotiation, hesitation, and human judgment. If a better solution came along later, the community could adjust its choices.

Today, matching is nearly invisible, yet more essential than ever. Every time we open an app, search for a product, order a ride, scroll through a profile on a dating app, or apply for a job, a digital system is at work, trying to connect us with something or someone. This happens at an imperceptible speed. These systems operate globally, continuously, and almost always automatically. But what has changed the most is not just the speed or the scale — **it's the timing of the decision**. In modern digital systems, people and resources do not arrive in neat order, waiting their turn. They appear randomly, in real time. And decisions can no longer be delayed until a complete picture emerges. There is no "pause" button to collect all the data first and make a decision later. Matches must be made immediately, based on partial information available at that moment. It is impossible to know who will request a ride in five seconds, or whether a more compatible kidney donor will register tomorrow. The system must choose now — or miss the opportunity. And above all, once a match is made, it is often **final**. A

driver starts their trip. An ad is displayed. A job is filled. A class spot is taken. The future is constrained by choices made in the present. It is this **irreversibility, combined with uncertainty** about what might still happen, that makes online matching fundamentally different — and vastly more complex — than traditional forms of matching that came before it. Yet despite this complexity, online matching has become a quiet cornerstone of modern life. It powers the services we use daily, governs access to resources, and increasingly determines who gets what, when, and how. Designing better online matching systems, therefore, is not just a matter of algorithms: it is a way to shape fairness, efficiency, and opportunity in a digital society.

To grasp the importance and diversity of these online matching systems, it is useful to examine how they operate across various domains in our lives. Whether it is public health, education, employment, housing, mobility, or even personal relationships, these systems are ubiquitous. In the following, we introduce some concrete examples from different contexts, yet all governed by the same underlying logic: making matching decisions in real time, with partial information and often without the possibility of going back. These applications highlight how seemingly invisible decisions deeply shape access to resources, the organization of services, and even key dimensions of social life.

4.1   Organ Donation: Matching to Save Lives

Organ donation is a vital act of solidarity that transforms the lives of thousands of patients in every country. There are two main types of donation: post-mortem donation, where an organ is harvested from a deceased person (with their consent or that of their relatives) and assigned to a patient on the waiting list; and living donation, which is less common and generally involves a kidney or a portion of the liver. In both cases, timing is a critical factor: once an organ becomes available, medical teams must act quickly to identify a compatible recipient and organize the transplant, as each graft often represents a unique opportunity for a patient.

In kidney transplantation, it is common for a patient to have a relative or friend willing to donate a kidney. However, this person may not be biologically compatible. Instead of abandoning the donation, hospitals can suggest a paired exchange: if another donor-recipient pair is in the same situation but with cross-compatibility, a mutual exchange can occur. This principle can extend to several pairs, forming a donation chain. In some cases, this chain begins with an altruistic donor—someone who donates a kidney without a designated recipient. In medical settings, this is referred to as a "domino transplant": the altruistic donor's kidney goes to the first patient, whose original donor becomes available for another patient, and so on. A single act of generosity can thus trigger a cascade of life-saving transplants.

This system relies on complex logistics. New patient-recipient pairs register over time. Some are immediately compatible with ongoing chains, while others must

wait several weeks or even months. A dilemma arises at every moment: should a transplant opportunity be seized as soon as compatibility is found, even if it only benefits a small group of patients? Or should one wait in the hope that additional pairs will enter the system, allowing the chain to grow and ultimately increase the total number of transplants? Acting too early can sacrifice a broader opportunity; waiting too long risks a deterioration in patients' health or the withdrawal of certain donors, which could cancel the entire chain procedure. Each successful transplant therefore depends on a fragile balance between medical urgency and the hope of future configurations. This process shows a fundamental reality: decisions are made as opportunities arise, with only partial knowledge of the future and rarely the possibility to reverse course. Once donors are committed to transplants, they can no longer be part of other chains. Each match changes the set of future possibilities. These decisions, often invisible to the general public, are nevertheless critical. Their sequence directly determines the system's ability to save lives. It is precisely this sequential, irreversible, and uncertain nature of decision-making — taken one by one, over time — that makes the management of organ donation chains so complex and so vital.

4.2 Medical Appointment Systems: Who Gets the Slot?

In hospitals or private medical centers, appointment scheduling is increasingly managed through online platforms such as Doctolib. While some patients use them to book routine checkups, others seek urgent appointments. Every day, new patients join the system at unpredictable times and with diverse medical needs. The system must assign available time slots dynamically, without knowing future demand or patient profiles. Should the schedule be filled as quickly as possible to avoid doctors being idle, or should some slots be held back for potential emergencies? What happens if a patient does not show up—can their slot be reassigned at the last minute? And how can fairness be preserved, particularly for elderly, vulnerable, or digitally inexperienced patients who may be disadvantaged by such systems? In this context, each scheduling decision directly affects access to medical care. An inefficient system can lead to delayed treatment, wasted appointments, or overcrowded emergency rooms. What makes this problem complex is that all of these decisions must be made in real time, with partial information, and often no opportunity to backtrack.

4.3   Volunteer Coordination: Helping Where the Need is Most Urgent

In humanitarian contexts — whether it involves welcoming refugees, managing natural disasters, or health crises — organizations such as the United Nations, international NGOs, and local associations must coordinate thousands of volunteers in real time. Digital platforms are used to register individuals ready to help, such as doctors, translators, logisticians, drivers or simply available volunteers. At the same time, field clinics, shelters, and local offices post urgent and constantly evolving needs based on urgent on-the-ground situations: "a nurse is needed at a mobile hospital within an hour," "a food package must be delivered before curfew," or "an Arabic-speaking translator is needed at a shelter by the end of the day." However, neither the volunteers nor the requests are known in advance. Both arrive gradually, over time. A volunteer might become available just minutes before—or after—a critical need arises. The platform must act fast: should the volunteer be assigned immediately to a current request, or wait in the hope of a more urgent or better-suited mission? These decisions are made under uncertainty and are often irreversible. For organizations like the United Nations or Doctors Without Borders, this is not just a logistical issue—it is a matter of real-world impact. A wrong match can lead to delayed aid, inefficient distribution of human resources, or even cases where people do not receive help in time. Conversely, a responsive and well-designed system can save lives, extend the reach of humanitarian intervention, and reduce suffering.

4.4   Ride-hailing: Matching in Motion

In modern cities, ride-hailing platforms must make fast and irreversible decisions to assign drivers to passengers as requests arrive in real time. During peak hours, the pressure is intense: thousands of users simultaneously request rides, while drivers are navigating traffic, completing trips, or repositioning themselves. The platform must estimate in real time who is available, where they are, and which assignment makes the most sense — all without knowing what the next requests will be. Consider a typical morning. A passenger requests a ride from the city center to the outskirts. Within seconds, the platform analyzes traffic conditions, estimated arrival times, and driver availability. A ride is confirmed. Everything appears seamless, but the decision, once made, cannot be reversed. Just moments later, another request may come in — perhaps more suitable — but the driver is already assigned. The platform must then continuously adjust its strategy, taking into account past decisions and increasingly limited resources. Delaying a decision risks frustrating users, while deciding too quickly can lead to suboptimal matches. Each individual decision may seem insignificant, but its cumulative effect shapes the overall quality of service: passenger wait times, driver profitability, traffic flow, and the geographic balance of supply. The challenge, therefore, is to design algorithms that can anticipate, adapt,

and maintain strong performance in the face of constant uncertainty and irreversible choices.

4.5   Online Advertising: Real-Time, High-Speed Auctions

Nowadays, much of our online browsing is closely related to advertising. Every time a user searches for something online — like "best running shoes" — or visits a webpage, or scrolls through a social media feed, a decision is made in a fraction of a second: Which ad should be displayed for which user, and at what moment? This choice is far from random. It relies on an ultra-fast, automated process in which advertisers—from global brands to independent merchants—participate in a real-time auction. As soon as an opportunity to display an ad arises, the platform rapidly evaluates several factors: What is the user profile? Which advertisers are interested in this audience? How much are they willing to pay? And most importantly, what is the likelihood that the user will click on the ad or make a purchase? Within milliseconds, an algorithmic auction is executed, and the ad deemed most "profitable" based on these criteria appears on-screen. However, these decisions are far from trivial. Advertisers have limited budgets: they cannot afford to target all users at all times. They aim to invest in the right place at the right time—reaching users considered sufficiently relevant, i.e., likely to buy, sign up, or interact with the brand. Consequently, a dilemma arises: should they bid now on this user, or wait for a potentially better one later? A poorly designed decision can waste part of the budget or miss out on an important opportunity. These decisions happen billions of times per day in a fast-moving and highly competitive environment. In this setting, the systems responsible for matching advertisers to users have to make instantaneous decisions with incomplete knowledge of the future. Platforms do not know which users will appear later, nor which advertising opportunities will come next. It is precisely this uncertainty about the future, combined with the irreversibility of decisions and budget constraints, that makes the task so complex.

4.6   Course Enrollment: The First-Come Dilemma

At universities, students often compete for access to highly popular courses. During registration periods — whether at the beginning of the academic year or semester — thousands of students log into enrollment platforms at the same time, trying to secure a spot. One student may succeed simply by being quicker, while another, who connects just seconds later, may be locked out—even if the course is essential for their graduation. This situation raises an important question: Should course seats be allocated on a first-come, first-served basis? Or should other criteria be taken into account, such as year of study, academic priority, or a student's individual curriculum? The issue becomes even more complicated when a course fills up, as

reorganizing enrollments without generating dissatisfaction or confusion is difficult. Each assignment is, in practice, an almost irreversible decision. If a higher-priority student appears later, it is often too late to change everything without disrupting the overall balance. This kind of situation is a subtle yet real form of online matching, where decisions must be made over time, without any visibility into future registrations. The challenge for institutions is therefore to design robust allocation systems capable of handling the sequential nature of the problem. The goal is not only to optimize speed or efficiency but also to ensure fairness, so that students' academic progress is not compromised. The issue goes beyond administrative logistics — it concerns equal opportunity and fairness in educational access.

4.7 Dating Apps: Finding the Right Swipe

In the world of dating apps, matching lies at the very heart of how these systems operate. Every day, millions of users log on hoping to meet someone with whom they can create a connection, a relationship, or simply share a moment. On apps like Tinder, Bumble, or Hinge, profiles show up one by one, and each user expresses interest or not with a simple gesture: swiping right or left. But behind this simplicity, there is a stream of decisions being: which profile to show you, in what order, and to whom your own profile should be shown in return. This process unfolds in a constantly changing environment. New users sign up every day, change their location, update their bios, or adjust their preferences. With this constant instability, these platforms must continually adapt. On the one hand, they need to present engaging profiles to capture attention and maintain user activity; on the other, they face a constant dilemma: should they show you a promising profile right now, or wait for someone more compatible to appear later? Should they show your profile to someone currently active, or to someone who is more likely to respond? Beyond the technical or algorithmic aspect, these platforms play a deeper social role. They influence behaviors, shape expectations, and sometimes even redefine relationship norms. A successful match can lead to a long-term relationship; a poor experience can result in frustration, disillusionment, or users abandoning the platform entirely. This raises important questions: Who gets shown more? How frequently do profiles appear? What truly determines compatibility? It is not just a matter of data or algorithmic performance. It is a system in which emotions, individual preferences, and split-second decisions silently shape the future of each user.

All these examples, though drawn from very different contexts, reveal a common reality: the matching problem is everywhere, and its challenges are far from simple. Behind each pairing lie complex and delicate decisions, made in real time and within evolving, constrained environments. The efficiency, fairness, and even the social impact of these systems depend on the algorithms that manage the matching. Understanding how these algorithms work, evaluating their performance, and finding ways to improve them is therefore a major technical and societal challenge.

In this thesis, we focus specifically on the case of online advertising. This domain, where matching takes place between users and advertisers through large-scale algorithmic auctions, offers a rich area for study. We aim to better understand the mechanisms behind these online systems and to analyze their performance under various settings and constraints. To study this problem, we proceed step by step. We begin by modeling the matching problem from a mathematical perspective, using tools from probability theory and statistics. This initial model—simpler and more generic—allows us to evaluate the performance of several standard algorithms. We then introduce constraints inspired by real-world systems—such as budget limitations, time-varying budgets, and restrictions on user profiles—and analyze how algorithms adapt and perform under these additional constraints.

In the second phase, we consider a richer mathematical model, one that more closely reflects the dynamics observed in real platforms, and we evaluate the performance of matching algorithms within this framework. Finally, we propose a formal design of online auction mechanisms, along with new matching algorithms tailored to this setting, and provide a theoretical analysis of their performance.

4

4

5

Background (in english)

Contents

The goal of this chapter is to provide an overview of the foundational concepts needed to understand the mathematical formulation of matching problems and the standard algorithms used to build maximum matchings. We begin by introducing the notion of graphs, which are the basic structures for modeling relationships and assignments between entities. Different types of graphs are presented, followed by a formal definition of matching. We then move to the online setting, where several frameworks for modeling online matching problems are discussed, along with the notion of the competitive ratio, which is used to evaluate the performance of matching algorithms. From another perspective, we introduce the multi-armed bandit problem, a central model in sequential decision-making. Though different in structure from graph matching, it addresses related challenges: making decisions

with partial information, adapting over time, and optimizing outcomes under uncertainty—principles that also arise in many online matching applications. In the rest of this chapter, we present the basic principles of the bandit model and some standard algorithms used to solve this problem.

5.1 Matching and its online version

In this section, we start by defining graphs and presenting several important types commonly used in modeling real-world systems. Graphs provide a powerful mathematical framework for representing relationships between entities—such as users and resources, or agents and tasks—making them central to the study of matching problems. We then formally define the notion of a matching on a graph, which captures the idea of assigning compatible pairs without conflicts. Building on this, we introduce the online version of matching, a key concept for modeling real-world scenarios (see Chapters 1 and 4). Finally, we present several classical algorithms designed for online matching and discuss the theoretical tools used to evaluate their performance.

5.1.1 Definition of a graph

Definition 1. *A graph is a pair $G = (V, E)$, where V is a set of nodes (vertices), and E is a set of edges which are unordered pairs $\{v_1, v_2\} \in V^2$ of nodes. If the edges have directions—i.e., they go from one vertex to another—we speak of a directed graph; otherwise, the graph is undirected. A node may belong to no edge, and is called isolated. When an edge v_1, v_2 exists, the nodes v_1 and v_2 are called adjacent.*

Depending on the structure of the set V and the edge set E, graphs can take different forms. In what follows, we introduce several types of graphs that are particularly relevant to our study.

5.1.2 Types of graphs

Definition 2. *A bipartite graph $G = (U, V, E)$ is a graph whose vertices can be divided into two disjoint sets U and V ($U \cap V = \varnothing$), such that every edge connects a vertex in U to a vertex in V, and no edge exists between nodes within the same set, $E \subseteq U \times V$.*

In the rest of this manuscript, we focus exclusively on bipartite graphs, as they provide a natural framework for modeling the scenarios of interest. Bipartite graphs typically fall into two main categories: deterministic bipartite graphs, in which the edge structure and the assignment of nodes across the two sets are fixed; and random

bipartite graphs, in which these elements are defined stochastically. In the definition below, we formally introduce the concept of a random bipartite graph.

Definition 3. *Let U, V be disjoint sets of vertices, with $|U| = n, |V| = m$ and $\Omega = 2^{U \times V}$ be the set of all edge subsets of $U \times V$. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space where $\mathcal{F}$ is the power set of Ω and $\mathbb{P} : \mathcal{F} \to [0, 1]$ is a probability measure over edge subsets. Then, a random bipartite graph is the random variable,*

$$G : (\Omega, \mathcal{F}, \mathbb{P}) \to \mathcal{G}_{n,m},$$

where $\mathcal{G}_{n,m}$ is the set of all bipartite graphs with fixed vertex sets U and V.

Having previously mentioned our focus on bipartite graphs due to their suitability for modeling the problems at hand, we now turn to specific random bipartite graph models. These models differ in how edges are generated and in the structure of the vertex sets. We begin with the Erdős–Rényi bipartite graph model, a standard and widely used random graph framework.

Definition 4. *Let U, V be a disjoint sets of vertices, with $|U| = n, |V| = m$ and $\Omega = 2^{U \times V}$ be the set of all edge subsets of $U \times V$. Let $(\Omega, \mathcal{F}, \mathbb{P}_p)$ be a probability space where $\mathcal{F}$ is the power set of Ω and $\mathbb{P}_p$ is the probability measure defined on Ω such that each edge $(u, v) \in U \times V$ is included independently with probability $p \in [0, 1]$, that is, for each $S \subseteq U \times V$, $\mathbb{P}_p(\{S\}) = p^{|S|}(1 - p)^{|U \times V| - |S|}$. Then, the Erdős–Rényi bipartite random graph is the random variable,*

$$G : (\Omega, \mathcal{F}, \mathbb{P}_p) \to \mathcal{G}_{n,m},$$

where $\mathcal{G}_{n,m}$ is the set of all bipartite graphs with fixed vertex sets U and V.

While the Erdős–Rényi model assumes uniform and independent edge probabilities across all vertex pairs, it does not capture the community structure often present in real-world problems. A generalization of the Erdős–Rényi model, known as the stochastic block model (SBM), groups vertices into communities and allows edge probabilities to vary between groups. This enables the modeling of structured interactions.

Definition 5. *Let U, V be a disjoint sets of vertices, with $|U| = n, |V| = m$ and $\Omega = 2^{U \times V}$ be the set of all edge subsets of $U \times V$, suppose that each node $u \in U$ is assigned to a class $c(u) \in C$ and $v \in V$ is assigned to $c(v) \in D$. Let $P \in [0, 1]^{C \times V}$ be a matrix of connection probabilities, where $P_{i,j}$ gives the probability of an edge between any $u \in U$ with $c(u) = i$ and $v \in V$ with $c(v) = j$. Let $(\Omega, \mathcal{F}, \mathbb{P}_{sbm})$ be a probability space where $\mathcal{F}$ is the power set of Ω and $\mathbb{P}_{sbm}$ is the probability measure defined on Ω such such that edges are included independently, that is, for each $S \subseteq U \times V$, $\mathbb{P}_{sbm}(\{S\}) = \prod_{(u,v) \in S} P_{c(u),c(v)} \prod_{(u,v) \notin S} (1 - P_{c(u),c(v)})$. Then, the stochastic block model bipartite graph is the random variable,*

$$G : (\Omega, \mathcal{F}, \mathbb{P}_{sbm}) \to \mathcal{G}_{n,m},$$

where $\mathcal{G}_{n,m}$ is the set of all bipartite graphs with fixed vertex sets U and V.

Many other random graph models havebeen proposed in the graph theory literature [19, 52, 36] to capture various structural and probabilistic features of networks. These include the configuration model, which allows for predefined degree distributions; geometric random graphs, which incorporate spatial constraints; uniform random graphs, where graphs are sampled uniformly from a fixed class; and Galton–Watson trees, commonly used to represent branching processes, among many other random graphs. In this work, we restrict our attention to random bipartite graph models that are most relevant to the problems studied in the remainder of the manuscript, namely the Erdős–Rényi model and stochastic block models.

5.1.3 Matching on a graph

Having defined bipartite graphs and introduced several classes of bipartite graph models, we now turn to an important structural concept: the notion of a *matching*. A matching formalizes the idea of pairwise assignments between the two vertex sets and plays a central role in many real-world applications. In what follows, we present a formal definition of a matching in a bipartite graph.

Definition 6. *Let $G = (U, V, E)$ be a bipartite graph, for each vertex $u \in U$, we define its degree in G, denoted $\deg_G(u)$, as the number of edges in E that are incident to u, and for each vertex $v \in V, \deg_G(v) = 1$, although v may of course have many neighbors in the graph. To each vertex $u \in U$, we associate a non-negative integer $c(u) \in \mathbb{N}$, called its* matching capacity, *satisfying, $c(u) \leq \deg_G(u)$. A matching in G is a subset of edges $M \subseteq E$ such that each vertex $u \in U$ is incident to at most $c(u)$ edges in M, and each vertex $v \in V$ is incident to at most one edge in M.*

Remark 1. *In the standard definition of a matching in a bipartite graph, each vertex in both U and V is allowed to be matched to at most one vertex from the opposite set.*
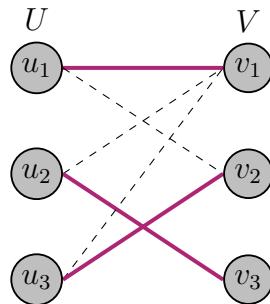
Figure 5.1: Matching on a bipartite graph

5.1.4 Online matching

The offline version of the matching problem has been widely studied [42, 19, 59, 18, 101]. In this setting, the entire graph is known in advance, and the objective is to

compute an optimal matching based on complete information. However, as discussed earlier in Chapters 1 and 4, many real-world scenarios involve situations where the full structure of the graph is not available at the beginning of the process. To model these online dynamics, the concept of online bipartite matching was introduced in [63], where one side of the bipartite graph remains fixed while the other is revealed incrementally. To understand this setting more formally, we define the notion of an online bipartite graph, where the dynamic nature of the problem is explicitly encoded in the arrival of one part of the graph over time:

Definition 7. *Let $G = (U, V, E)$ be a bipartite graph. In the online bipartite setting, the set U is known in advance and remains fixed throughout the process, while the set V is revealed sequentially over time. At each time step $t \in \{1, \ldots, |V|\}$, a vertex $v_t \in V$ is revealed along with the set of its neighbors in U. The algorithm must then make an irrevocable decision, such as whether to match v_t to an available neighbor in U.*

Here, the notion of time is modeled by the sequential revelation of vertices in V, with one vertex arriving at each time step.
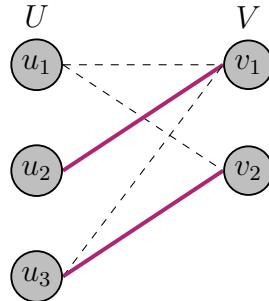
Figure 5.2: Online bipartite graph

As defined earlier, a matching on an online bipartite graph is a subset of edges such that no two edges share a common vertex. However, in the online context, the graph is revealed sequentially, and decisions must be made without knowledge of future arrivals. As new vertex in V is unveiled over time, the matching evolves accordingly. Throughout this process, online matching algorithms attempt to build the matching in real time—one that is as large as possible, despite operating with only partial information about the graph at each time step.

5.1.4.1 The competitive ratio

As previously discussed, in the online matching problem, when a vertex in V arrives, an online algorithm attempts to match it to one of its neighbors in U based on certain decision-making rules. Once this decision is made, it is irrevocable. This models practical situations where decisions cannot be reversed: for instance, once an ad is shown to a user or a profile is displayed on a dating app, the choice is made and

cannot be undone. To measure the performance of such online algorithms, a standard metric called the competitive ratio was introduced in the literature on online matching [81]. This ratio captures the loss incurred by operating online—without full knowledge of future inputs—compared to an optimal offline algorithm that knows the entire graph in advance. Broadly speaking, it is defined as the ratio between the expected size of the matching produced by the online algorithm and that produced by the offline optimum. More formally,

Definition 8. *Let $\mathcal{G}$ be a family of bipartite graphs, and $\mathtt{ALG}(G)$ the matching size built by an online algorithm $\mathtt{ALG}$ on $G \in \mathcal{G}$, and and $\mathtt{OPT}(G)$ the matching size built by the optimal offline algorithm $\mathtt{OPT}$. We say then that an algorithm $\mathtt{ALG}$ reaches a competitive ratio α over the family $\mathcal{G}$, if there exists a constant c such that for any $G \in \mathcal{G}$,*

$$\mathbb{E}[\mathtt{ALG}(G)] \geq \alpha \mathbb{E}[\mathtt{OPT}(G)] + c,$$

where the expectations are taken with respect to the possible randomness in the algorithms. Two types of competitive guarantees are commonly studied. In the first, G itself is sampled from a given distribution over graphs (e.g., a random graph model), and the competitive ratio is evaluated in expectation over this randomness. In the second, worst-case analysis is performed over all $G \in \mathcal{G}$, and the bound must hold uniformly for every graph in the family.

The competitive ratio, denoted $\mathtt{CR}$, is a quantity between 0 and 1. A higher $\mathtt{CR}$ indicates better performance of the algorithm under consideration. Two main factors influence the value of $\mathtt{CR}$: the choice of the algorithm itself and the class of graphs $\mathcal{G}$ over which the algorithm is evaluated.

5.1.4.2 Online bipartite matching settings and algorithms

In the final part of this section, we present standard frameworks considered in the study of the online matching problem [81]. These typically include the classical assumptions made about the graph family $\mathcal{G}$, along with the algorithms proposed in the literature for each context.

Adversarial setting. In the adversarial framework, the family $\mathcal{G}$ can be any family of bipartite graphs, meaning that for any $G \in \mathcal{G}$, the vertices in V may arrive in an arbitrary order. The competitive ratio of an algorithm in this case is evaluated on the graph where it performs the worst.

Surprisingly, even in this challenging setting, a guarantee of $1/2$ on the $\mathtt{CR}$ can be achieved with the standard $\mathtt{Greedy}$ strategy.

Algorithm 6: $\mathtt{Greedy}$ policy

---
1 for $t = 1, \ldots, |V|$ **do**
2 | Match v_t to a free neighbor chosen uniformly at random.

The next result, shows that the competitive ratio of `Greedy` has a lower bound of $1/2$, this result is true for `Greedy` and for all algorithms picking any match as soon as one is available.

Theorem 2. *In the adversarial setting,*

$$\texttt{CR(Greedy)} \geq \frac{1}{2}.$$

Proof. Consider the case shown in Figure 5.3, where `Greedy` fails to match vertex v_2, even though v_2 is matched in the maximum matching of the final graph. This can only happen if the vertex in U that would have been matched to v_2 in the optimal solution has already been matched by `Greedy` to another vertex, say v_1, that arrived earlier. Thus, for any "miss" event (i.e., `Greedy` fails to match a vertex that is matched in the maximum matching), there is at least one "match" event (i.e., `Greedy` matches a vertex). This implies that the total number of matches in the optimal solution is at most twice the number of matches made by `Greedy`. Equivalently, the matching produced by `Greedy` is at least half the size of the optimal matching. Hence, the competitive ratio of `Greedy` has a lower bound of $\frac{1}{2}$.
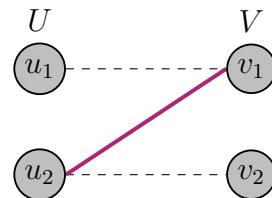
Figure 5.3: A difficult instance for `Greedy`

□

We can go further and prove that this lower bound is tight as follows,

Proposition 2. *In the adversarial setting,*

$$\texttt{CR(Greedy)} = \frac{1}{2}.$$

Proof. Consider a graph G_n illustrated in Figure 5.4. There are $2n$ vertices on either side of G_n, which are split as follows: $U = U_1 \cup U_2$ and $V = V_1 \cup V_2$ with $U_1 = \{u_1, \ldots, u_n\}$ and $U_2 = \{u_{n+1}, \ldots, u_{2n}\}$, and similarly for V. The set of edges is defined by $E = \{(u_i, v_i), \forall i \in [2n], (u_{n+i}, v_j) \; \forall (i, j) \in [n]^2\}$. The largest possible matching is of size $2n$. If `Greedy` is performed on G_n, we can see that any node $v_i \in V_1$ will pick its match in U_2 with probability greater than $\frac{n-i+1}{n-i+2}$ as it only has one neighbor in U_1, and at time step i, at most $i - 1$ vertices have already been

matched in U_2. This implies that, in expectation, $n - o(n)$ vertices of V_1 are matched with a vertex $u_j \in U_2$. Thus,

$$\lim_{n \to \infty} \frac{\mathbb{E}[\texttt{Greedy}(G_n)]}{2n} = \frac{1}{2}.$$
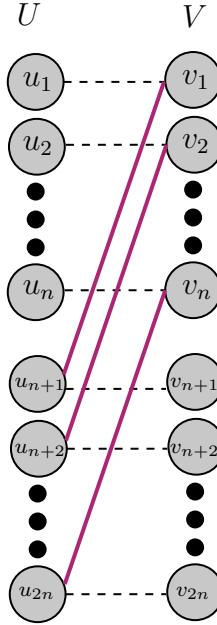
Figure 5.4: A difficult instance for `Greedy`

$\square$

As presented in [81], another algorithm can have better guarantees than `Greedy` in the adversarial case: the `Ranking` algorithm, which relies on the principle of "correlated randomness." The `Ranking` algorithm works as follows: at the beginning of the process, a random permutation π is drawn, and each vertex $u_i \in U$ is assigned a rank $\pi(i)$. When a vertex $v \in V$ arrives, it is matched to its free neighbor that has the lowest rank.

Algorithm 7: `Ranking` policy

---

1 Draw a random permutation π.
2 for $i = 1, \ldots, |U|$ **do**
3 $\quad$ Assign to u_i a rank $\pi(i)$.
4 for $t = 1, \ldots, |V|$ **do**
5 $\quad$ Match v_t to its lowest ranked free neighbor.

Taking a look back at the example presented in Figure 5.4, we can understand why `Ranking` achieves better performance than `Greedy`. When a vertex $v \in V_1 \cup V_2$ arrives, `Greedy` tends to match it to a high-degree neighbor in U, introducing a bias:

`Greedy` often matches high-degree vertices too early. However, a better policy would preserve high-degree vertices as fallback options for future arrivals that have fewer neighbors. By assigning ranks to nodes in U, `Ranking` corrects this bias. When a vertex $v \in V_1 \cup V_2$ arrives, `Ranking` matches it to the lowest-ranked free neighbor. As a result, over time and in expectation, `Ranking` matches more vertices than `Greedy` on this instance.

Theorem 3. *In the adversarial framework,*

$$\mathtt{CR}(\mathtt{Ranking}) \geq 1 - \frac{1}{e} \simeq 0.63.$$

Here e is the Euler number.

This result has been established through several different proof techniques. A direct proof, based on mapping "miss" events to "match" events, is given in [16]. Another line of proof uses primal-dual analysis; see [35] for details. In the original paper that introduced the `Ranking` algorithm, [63] demonstrated that the lower bound on its competitive ratio is tight, using a family of instances known as upper-triangular graphs. Building on this example and applying Yao's lemma, they further showed that no randomized algorithm can achieve a competitive ratio better than $1 - \frac{1}{e}$. For full details, see [63, 81].

Random order. In this setting, the graph can still be any bipartite graph, but unlike the adversarial framework, the vertices in V arrive in random order. Under this random arrival model, both `Greedy` and `Ranking` achieve higher performance than in the adversarial setting. Specifically, `Greedy` achieves a `CR` of $1 - \frac{1}{e}$. This is because, with random order, `Greedy` behaves similarly to `Ranking` and, in fact, shares the same lower bound of $1 - \frac{1}{e}$ on the `CR`. As the `Ranking` algorithm, its exact competitive ratio in the random arrival setting remains an open problem. However, the best known lower bound, approximately 0.696 was obtained in [76].

Known i.i.d. In real-world applications, we often have some information about the graph structure, making the adversarial and random-order models too restrictive. To better capture such practical scenarios, a new framework called the Known i.i.d. model was introduced in [40]. In this setting, the algorithm knows the set of nodes U as well as a distribution over possible types of nodes in V, where a type specifies the set of neighbors in U. With this additional information, a new algorithm was proposed in [40] called the `Suggested − Matching` algorithm. The idea is simple: the algorithm precomputes a perfect matching on the base graph (where V is replaced by the set of types) and uses it to guide online decisions. When a vertex arrives, it is assigned exactly one predetermined neighbor to try to match with. In doing so, the algorithm attempts to reconstruct the perfect matching from the base graph as closely as possible in the online setting. In [40], it was shown that this algorithm achieves a competitive ratio of $1 - \frac{1}{e}$. However, while effective, this approach is still suboptimal, as any type arriving for the second time or more is left unmatched.

This limitation led to a series of improvements. The first advancement introduced algorithms with two choices: instead of focusing on one suggested matching in the base graph to guide decision-making, the algorithm uses two matchings for this purpose, raising the CR to 0.706 [40, 77, 58]. More recently, an algorithm with more than two choices was proposed. It relies on the solution of a linear program, with additional Online Resolution Scheme (OCS) techniques [55], pushing the competitive ratio further to 0.716.

5.1.5 The $b-$matching problem

As seen throughout this chapter, there is a vast line of research dedicated to the problem of online matching, reflecting both its theoretical depth and practical relevance. Many generalizations of this problem have been introduced—some to better model real-world applications, and others to capture the problem's complexity in more complex settings. One such generalization is the b-matching problem, where, instead of assuming that nodes in U can be matched only once, we allow each node to be matched up to b times, where $b \in \mathbb{N}^*$ is the budget of each node in U. We can think of these budgets as the degrees of nodes in U. This generalization is particularly relevant in online advertising scenarios, where advertisers (nodes in U) typically have campaign budgets that limit how many times their ads can be shown to users (arriving nodes in V). To address this problem, an algorithm called `Balance` was introduced in [62]. It operates as follows:

Algorithm 8: `Balance` policy

---

1 for $t = 1, \ldots, |V|$ **do**
2 $\quad$ Match v_t to a neighbor with highest remaining budget;

To better understand how `Balance` works, let us apply it to the example in Figure 5.5. We have three nodes in U: u_1 with initial budget $b_{u_1,0} = 2$, u_2 with $b_{u_2,0} = 1$, and u_3 with $b_{u_3,0} = 2$. When v_1 arrives, `Balance` can match it to one of its neighbors—let us say u_1. After this match, the budget of u_1 becomes $b_{u_1,1} = 1$. Next, v_2 arrives with two neighbors: u_1, now with $b_{u_1,1} = 1$, and u_3 with $b_{u_3,1} = 2$. Since `Balance` prefers the neighbor with the highest remaining budget, it will choose u_3 to match with v_2. Finally, when v_3 arrives, it has only one neighbor, u_2, so there is only one possible choice for `Balance`.

$b_{u_1,0} = 2$ after match $b_{u_1,1} = 1$ U V

$b_{u_2,0} = 1$ after match $b_{u_2,1} = 0$

$b_{u_3,0} = 2$ after match $b_{u_1,1} = 1$
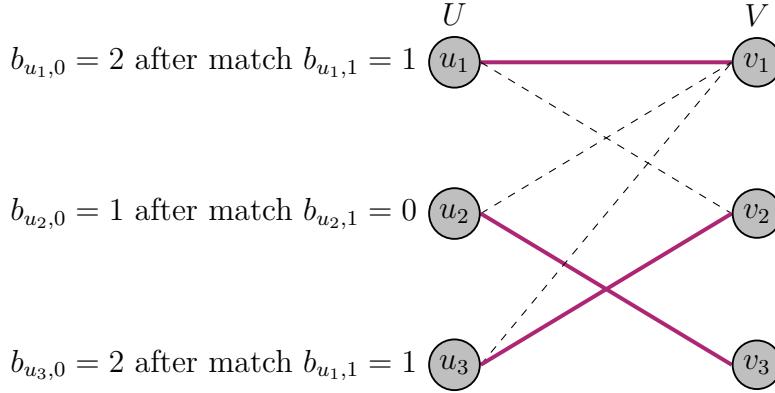
Figure 5.5: `Balance` algorithm process

One can observe that `Balance` makes slightly more strategic decisions than `Greedy`, as it consistently favors neighbors with the highest remaining budget, gradually balancing the load across nodes in U—which is precisely the idea behind its name. The analysis of `Balance` in the adversarial setting was first presented in [62], where they proved a competitive ratio that is parameterized by b, the budget of nodes in U, given by $1 - \frac{1}{(1+1/b)^b}$. This result is obtained by constructing a carefully designed online bipartite graph and analyzing the matching produced by `Balance`. They also showed that `Balance` is optimal among all deterministic algorithms in the adversarial setting. A further generalization, more aligned with practical applications, of the b-matching problem was proposed in [6]. In this setting, all the budgets of nodes in U are fixed but not equal. The performance of `Balance` is then characterized in terms of the smallest budget $b_{\min} = \min_{u \in U} b_u$, where b_u is the budget of node $u \in U$. The `CR` is given by $1 - \frac{1}{(1+1/b_{\min})^{b_{\min}}}$.

5.2 Multi-arm bandit problem

While the classical online matching problem focuses on choosing the best sequence of decisions as vertices on one side of the graph arrive over time, it usually assumes that the underlying graph structure—namely, the set of potential neighbors for each arriving vertex—is known in advance. This modeling choice is well-suited to many applications, as discussed in Chapters 1 and 4. However, in a wide range of real-world scenarios such as recommendation systems, online marketplaces, or resource allocation, this classical framework overlooks a crucial element: although the neighbors of an arriving vertex are known, the reward associated with allocating it to each of these neighbors is not known beforehand. The algorithm only discovers these rewards as decisions are made. In such settings, the objective is no longer simply to construct a large matching or to maximize a competitive ratio based solely on combinatorial constraints. Instead, the goal becomes to learn and maximize the cumulative reward over time, despite initially lacking information about the value of each possible action. This change of objective introduces an additional layer of

uncertainty: the algorithm must now balance learning the reward structure with making good immediate decisions. This naturally leads to the multi-armed bandit problem, whose origins lie in the analogy with slot machines. A player, faced with several machines (or arms), must choose which one to play, each pull generating a reward unknown in advance. The goal is to maximize the cumulative reward over time. This requires navigating the fundamental trade-off between exploitation—selecting the arm that has yielded high rewards so far—and exploration—testing less-sampled arms in search of potentially better rewards [72]. In the remainder of this chapter, we formally introduce the multi-armed bandit problem, focusing on the stochastic setting. We then define the notion of regret, which plays a role analogous to the competitive ratio in online matching. Finally, we present several standard algorithms used in the bandit framework.

5.2.1   Definition of bandit problem

A multi-armed bandit problem (MAB) is a sequential decision-making problem, defined by a finite set of actions $\Theta = \{1, \ldots, K\}$, called arms in the bandit literature. Each arm $i = 1, \ldots, K$ has a sequence of unknown rewards $X_{i,1}, X_{i,2}, \ldots, X_{i,K}$ in $[0, 1]$. At each round, the player makes a decision by choosing an arm k_t in Θ and observes a reward $X_{k_t,t}$. The goal of the player is to choose the best sequence of actions over time to maximize the cumulative reward.

There are many frameworks within the MAB paradigm, each depending on the structure of the reward and different modeling assumptions. Among them are the adversarial, linear, and stochastic settings. In the adversarial framework, rewards are generated by an adversary and may vary in response to the algorithm's past actions. The linear setting assumes that each arm is associated with a feature vector and that the expected reward is a linear function of these features. For more details on other frameworks, see [72]. In this chapter, we focus on the stochastic setting, where each arm has a reward drawn independently from a fixed but unknown probability distribution. The stochastic MAB can be summarized as follows: at each time step $t = 1, \ldots, T$,

- The player chooses an arm $k_t \in \Theta$.

- Knowing k_t, the environment draws a reward $X_{k_t,t} \sim \nu_{k_t}$.

- The player observes only the reward $X_{k_t,t}$ at each time-step.

Moreover, in the stochastic setting, the rewards generated by each arm are typically assumed to be independent and identically distributed (i.i.d.).

5.2.2 The regret

The notion of regret can be introduced in several equivalent ways. In fact, stating that the goal of the player is to maximize cumulative reward over time is equivalent to saying that the objective is to minimize the cumulative regret, defined as,

$$R(T) = \max_{k=1,\dots,K} \sum_{t=1}^{T} X_{k,t} - \sum_{t=1}^{T} X_{k_t,t}.$$

Here, $X_{k,t}$ denotes the reward of arm k at time t, and k_t is the arm chosen by the player at round t. We also define $\mu_k = \mathbb{E}[X_{k,t}]$ as the expected reward of arm k, and let $\mu^* \in \arg\max_{k=1,\dots,K} \mu_k$ be the best expected reward. In the stochastic setting, it is common to focus on a related quantity known as the pseudo-regret, which corresponds to competing with the best arm in expectation rather than the optimal arm on the sequence of realized rewards.

Definition 9. *The pseudo regret is defined as,*

$$\tilde{R}(T) = T\mu^* - \mathbb{E}\left[\sum_{t=1}^{T} \mu_{k_t}\right].$$

We can notice that the pseudo-regret is upper-bounded by the expected regret $\mathbb{E}[R(T)]$. There exists another formulation of the pseudo regret, widely used to analyze a bandit algorithm. It is based on the following two quantities:

$$\Delta_k = \mu^* - \mu_k, \qquad \text{and} \qquad N_k(t) = \sum_{s=1}^{t} \mathbb{1}_{\{k_t=k\}}.$$

where Δ_k denotes the suboptimal gap of arm k, which is the difference between the mean reward of the best arm μ^* and the mean reward of arm k. $N_k(t)$ denotes the number of times the arm was pulled by the player up to time t. With theses quantities in hand, the pseudo-regret can equivalently be expressed as,

$$\tilde{R}(T) = \sum_{k=1}^{K} \Delta_k \mathbb{E}[N_k(t)].$$

5.2.3 Some standard bandit algorithms

As introduced previously, the goal of a player is to maximize the cumulative reward over time. To do so, they must balance the trade-off between exploration—collecting information about the arms—and exploitation—using that information to select the best arm. This trade-off is known in the bandit literature as the exploration-exploitation dilemma. One standard strategy to address this dilemma is the Explore-Then-Commit (ETC) approach. It consists in performing an exploration phase of

length mK, in which each arm is pulled $m \geq 1$ times. After this phase, it exploits the arm with the best empirical reward.

Algorithm 9: ETC policy

Input: $m \geq 1$ sampling parameter.

1 In round t choose action

$$
k_t = \begin{cases} (t \bmod K) + 1 & \text{if } t \leq mK, \\ \arg\max_k \hat{\mu}_k(mK) & \text{if } t > mK. \end{cases}
$$

Where $\hat{\mu}_k(t) = \frac{1}{N_k(t)} \sum_{s=1}^{t} \mathbb{1}_{\{k_s=k\}} X_{k,s}$. Based on [72, 43], ETC acheives the following performance,

Theorem 4. *If* $1 \leq m \leq T/K$, *then,*

$$
\tilde{R}(T) \leq m \sum_{k=1}^{K} \Delta_k + (T - mK) \sum_{k=1}^{K} \Delta_k e^{-m\Delta_k^2}.
$$

A detailed proof of this result can be found in [72, 43]. The regret bound in Theorem 4 highlights the fundamental trade-off between exploration and exploitation. More specifically, when m is large, the algorithm spends more time exploring, which increases the first term $\sum_{k=1}^{K} \Delta_k$ and thereby increases the regret. Conversely, for small m, there is a high probability of selecting a suboptimal arm during exploitation, causing the second term to dominate and potentially lead to large regret. Thus, the choice of m must balance these effects and depends on both the suboptimality gap and the horizon. While the horizon T is sometimes known in advance, it is rarely realistic to assume knowledge of the suboptimality gap. Nonetheless, one can show that there exists a choice of m that depends on T and leads to the problem-independent regret of order $\mathcal{O}(T^{2/3})$.

While ETC uses a simple strategy to balance exploration and exploitation, it suffers from several limitations: sensitivity to the choice of the exploration length, which induces a risk of committing too early or too late; the requirement of knowing the time horizon T in advance to calibrate m; and the separation between the exploration and exploitation phases, which makes the algorithm inflexible if more information is gathered later in the process. To overcome these limitations, more adaptive strategies have been proposed. One widely studied approach is the Upper Confidence Bound (UCB), which is based on the optimism principle. More precisely, for each arm k, it builds a confidence interval on its expected reward based on past observations. Then it acts *optimistically*, as if the best possible rewards within the confidence interval are the true rewards, and chooses the next arm to pull accordingly—meaning that it pulls the arm with the highest upper confidence bound.

Algorithm 10: `UCB` policy

Input: For rounds $t = 1, \ldots, K$, pull arm $k_t = t$.

1 for $t = K + 1, \ldots, T$ **do**

2   Choose $k_t \in \arg\max_{k \in \{1, \ldots, K\}} \left(\hat{\mu}_k(N_k(t-1)) + \sqrt{\frac{2\log(t)}{N_k(t-1)}} \right)$.

3   Observe the reward and update the upper confidence bounds.

Based on [72, 43], `UCB` acheives the following performance,

Theorem 5. *For any time horizon T,*

$$\tilde{R}_T \le 3 \sum_{i=1}^{K} \Delta_i + \sum_{i:\Delta_i > 0} \frac{16\log(T)}{\Delta_i}.$$

The proof of this result can be found in [72, 43]. Theorem 5 shows that `UCB` achieves a regret bound that scales logarithmically with T, a significant improvement over the `ETC` strategy. Moreover, using the fact that the regret incurred from pulling arm k cannot be larger than $T\Delta_k$, this distribution-dependent upper bound can be transformed into a distribution-free bound of order $\mathcal{O}(\sqrt{TK\log(T)})$. This rate is nearly optimal, as the minimax lower bound is $\mathcal{O}(\sqrt{TK})$. The extra logarithmic factor can, in fact, be removed by using more refined algorithms, such as the Minimax Optimal Strategy.

Beyond the algorithms presented in this section, many other strategies have been developed to address the multi-armed bandit problem, including ϵ-greedy, which explores randomly with a fixed probability, and Thompson Sampling, which uses Bayesian principles to balance exploration and exploitation. Furthermore, as briefly discussed at the beginning of the section, the bandit framework has been extended in several settings to address more complex real-world applications, such as linear bandits, adversarial bandits, contextual bandits, and combinatorial bandits. Each of these variants introduces additional challenges and requires more complex algorithms. However, in this chapter, we have focused on the canonical stochastic bandit problem and the most widely studied algorithms, which together provide a solid foundation for understanding more advanced models and techniques.

5

6

Contributions (in english)

Contents

6.1 Overview of the thesis

The main focus of this thesis is online matching in bipartite graphs. As introduced in Chapters 1 and 4, such problems have a wide range of applications across diverse domains. In this work, the emphasis is specifically on online advertising. The first project was inspired by the interaction between users and advertisers. While much of the existing literature on online matching in advertising simplifies or relaxes key constraints, this work addresses a fundamental and realistic constraint that is ubiquitous in online advertising: advertisers operate under budgets that limit how many times their ads can be displayed to the users. Crucially, these budgets are not static; they evolve over time. For instance, advertisers such as clothing brands may replenish their budgets during sales periods like Black Friday to maximize their profits and then scale it back afterward. This project aimed to incorporate such

dynamic budget constraints into the standard online matching framework and to analyze how different algorithms perform under these more realistic conditions. To build foundational insights, part of the analysis focused on the classical Erdős–Rényi random graph model, offering a simplified yet informative setting for understanding how budget evolution impacts algorithmic performance.

Although this initial study captured budget dynamics, it relied on relatively simple bipartite graphs. In real-world applications, particularly in advertising, the networks that connect users to advertisers are significantly more complex. They often exhibit community structures, where nodes are grouped into clusters and interactions are governed by group-level affinities and preferences. For example, users might be clustered by age, interests, or browsing habits, while advertisers can be grouped by their target audiences or the types of ads they promote. Capturing this richer structure requires more sophisticated bipartite graph models. Among these, the stochastic block model is especially well suited, as it explicitly groups nodes from both sides of the graph into classes and defines interaction probabilities based on class compatibility. These considerations motivated the second project of this thesis, which focused on online matching on stochastic block models. This project had two main objectives: first, to understand how standard algorithms such as `Greedy` and `Balance` perform when the underlying graph exhibits community structure; and second, to incorporate a learning feature by assuming that the probabilities defining the stochastic block model are not known in advance and must be estimated over time. This assumption reflects a more realistic setting in which the platform must simultaneously learn the structure of user-advertiser interactions and make online matching decisions. To tackle this challenge, we reformulate the problem as a multi-armed bandit problem and propose an algorithm based on an Explore-Then-Commit `ETC` strategy and `Balance` policy to learn the probabilities and build the largest matching possible.

Beyond the interactions between users and advertisers, a crucial component of online advertising is related to auction mechanisms, where advertisers compete in real time to secure opportunities to display their ads. This motivated the third project of this thesis, which focused on the study of online auctions. Specifically, we considered the framework in which ad impressions (online spots for ads) are sold through second-price auctions, and the decision-maker must select a subset of campaigns to participate in each auction round. We framed this problem as a structured multi-armed bandit setting, where each arm corresponds to the choice of how many bidders from the coalition are selected to participate in a given auction round. The outcome of each auction provides feedback in the form of a reward, reflecting whether the coalition won the impression and the associated price paid. A key insight from this work is that the expected reward function over the arms exhibits a particular structure, which can be exploited to improve learning efficiency. Building on this observation, we developed algorithms that leverage this property to balance exploration and exploitation, and optimize the cumulative gain of the coalition of campaigns over time. In the next sections, we present in more detail the contributions of each of these projects.

6.2 Online matching with budget refills

In this work [29], we consider a bipartite graph $G = (U, V, E)$, composed of two sets of nodes $U = \{1, \dots, n\}$ and $V = \{1, \dots, T\}$ for $T, n \in \mathbb{N}^*$ and a set of edges $E \subseteq U \times V$. Nodes in U are known beforehand, while nodes in V are discovered sequentially. Each node $u \in U$ has a budget $b_{u,t} \in \mathbb{N}$ that evolves dynamically according to a process to be specified later. We study two frameworks: the adversarial framework, in which the graph G is deterministic and the budgets of nodes in U evolve according to a deterministic dynamics, and a stochastic framework where the graph is random and the budget dynamics are governed by a stochastic process. The following sections present the main results for each framework.

6.2.1 The adversarial framework

Let $G \in \mathcal{G}_{T,m}$ be a bipartite graph, where $\mathcal{G}_{T,m}$ denotes the family of graphs defined below,

$$\mathcal{G}_{T,m} = \Big\{ (U, V, E, (\eta_{u,t})_{u \in U, t \in V}) \ : \ \forall t \in V, \ \exists u_t \in U \text{ such that } \eta_{u_t,t} = \mathbf{1}_{\{t \equiv 0 \pmod{m}\}}$$
$$\text{and } (u_t, t) \in E \Big\}.$$

m is a parameter that we will specify later. In the adversarial setting, it is essential to impose certain restrictions on the power of the adversary, in order to prevent the worst-case scenario from reducing the problem to a graph without effective refills. We therefore adopt the following assumptions: the refill schedule is fixed and known in advance, and each node $t \in V$ has at least one neighbor in U.

The online matching generated by an algorithm `ALG`, on $G \in \mathcal{G}_{T,m}$, is a subset of edges that can be represented by a binary matrix $\mathbf{x} \in \{0, 1\}^{n \times T}$ that must satisfy the following constraints: only edges in E can be selected for matching; each node in V can only be matched at most once; and a node in U can only be matched at time t if its budget is strictly positive at that moment.

In this model, the budget evolution for each $u \in U$ depends on whether the edge $(u, t) \in E$ is included in the online matching, which is indicated by the binary variable $x_{u,t} \in \{0, 1\}$, as well as the refill which is the addition of one unit every m time steps. Intuitively, the refill models the periodic regeneration of a node's capacity to participate in new matchings. Formally, this dynamic can be expressed as follows,

$$\forall u \in U, b_{u,t} = b_{u,t-1} - x_{u,t} + \mathbb{1}_{\{t \bmod m = 0\}}, \quad \text{and } b_{u,0} = b_0 \text{ for some } b_0 \geq 1.$$

`OPT` is the largest possible matching in hindsight, with matrix $\mathbf{x}^*$. In this case

the competitive ratio is defined by

$$\mathtt{CR}^{\mathrm{adv}}(\mathtt{ALG}, \mathcal{G}_{T,m}) = \min_{G \in \mathcal{G}_{T,m}} \frac{\mathtt{ALG}(G)}{\mathtt{OPT}(G)}.$$

where $\mathtt{ALG}(G) = \sum_{u \in U} \sum_{t=1}^{T}, x_{u,t}$, and $\mathtt{OPT}(G) = \sum_{u \in U} \sum_{t=1}^{T}, x_{u,t}^{*}$, are the sizes of the matching generated by $\mathtt{ALG}$ and $\mathtt{OPT}$ respectively.

In this framework, we focus on the analysis of $\mathtt{Balance}$ algorithm, as it was initially designed for adversarial settings. Our goal is to understand the effect of the refills on the matching process. To this end, we study the performance of $\mathtt{Balance}$ in two distinct regimes: one where refills are infrequent, specifically when $m = \omega(\sqrt{T})$, and another where refills are frequent, $m = o(\sqrt{T})$. The first result of this project addresses the former case and is stated below.

Theorem 6. *Assuming the initial budgets are $b_{1,0} = b_{2,0} = \cdots = b_{n,0} = b_0 \geq 1$. If $m = \omega(\sqrt{T})$ and $b_0(b_0 + 1)^{b_0} \leq m$, then,*

$$\sup_{\mathtt{ALG}: deterministic} \mathtt{CR}^{\mathrm{adv}}(\mathtt{ALG}, \mathcal{G}_{T,m}) \leq 1 - \frac{1}{\left(1 + \frac{1}{b_0}\right)^{b_0}} + o_T(1).$$

The proof of this result relies on the findings in [62], and the complete details are provided in Section 7.A.1. The main insight of this theorem is that when refills are relatively infrequent, the worst case is arbitrarily close to that of the b_0-matching (additional details regarding the definition of the b_0-matching can be found in Chapter 5). This is intuitive: in such a regime, most of the matching occurs early on, depleting the initial budgets, while the occasional refills—being too sparse—have little impact on the overall outcome. As a result, these rare refills do not significantly improve the total number of matches.

In contrast, when refills are frequent, the situation changes significantly. Unlike the previous scenario, where the $\mathtt{CR}$ was bounded by $1 - \frac{1}{e}$, here the bound improves to approximatly 0.73.

Theorem 7. *Assuming the initial budgets are $b_{1,0} = b_{2,0} = \cdots = b_{n,0} = b_0 \geq 1$. For $m = o(\sqrt{T})$ and $mb_0 = o(T)$, then,*

$$\mathtt{CR}^{\mathrm{adv}}(\mathtt{Balance}, \mathcal{G}_{T,m}) \leq \underbrace{1 - \frac{(1-\alpha)}{e^{(1-\alpha)}}}_{\simeq 0.73325...} + o_{m,T}(1).$$

where α is defined by $\frac{1}{2} = \int_0^\alpha \frac{xe^x}{1-x}\mathrm{d}x$. The bound is reached for the graph defined in the proof.

The full proof is provided in Section 7.A.2. It relies on building an adversarial graph to exploit the algorithm's inability to predict which nodes will remain available. Initially, all nodes in U are connected to incoming nodes so that $\mathtt{Balance}$

distributes matches evenly and accumulates budget through frequent refills. During this period, `Balance` and the optimal offline algorithm `OPT` perform identically. The difference arises later, when the adversary removes most of the nodes in U, specifically, those where `Balance` had stored budget. Unlike `Balance`, which cannot anticipate which nodes will survive, the optimal offline algorithm has perfect knowledge of the eliminated nodes. Therefore, it can allocate the budget exclusively to nodes that remain available, ensuring that no budget is lost on those that are removed at the time of their removal. This systematic removal of nodes forces `Balance` to waste much of its accumulated budget, preventing it from converting frequent refills into proportionally more matches. As a result, even though refills make the problem less constrained, the competitive ratio still cannot exceed roughly 0.73.

The last result of this section shows that no deterministic algorithm can achieve better performance than `Balance` in this setting.

Theorem 8. *Assuming the initial budgets are $b_{1,0} = b_{2,0} = \cdots = b_{n,0} = b_0 \geq 1$. For $m = o(\sqrt{T})$,*

$$\sup_{\texttt{ALG}} \mathbb{E}\left[\texttt{CR}^{\mathrm{adv}}(\texttt{ALG}, \mathcal{G}_{T,m})\right] \leq \texttt{CR}^{\mathrm{adv}}(\texttt{Balance}, G^{th.2}) + o_T(1).$$

where the expectation is taken over the randomness from `ALG`.

The intuition behind this result is that keeping budgets equalized among the currently available U-nodes is the best choice an algorithm can adopt in the adversarial graph used for the proof of Theorem 17. This is because the adversary removes U-nodes one after another, starting with those that have the highest budget and never providing again a U-node already removed.

6.2.2 The stochastic framework

The online matching problem with refills of the budgets in the stochastic setting is studied in the following framework:

1. The random graph is a standard Erdős–Rényi model $G(n, T, p)$, i.e., a bipartite graph with n vertices on one side, T on the other, where each potential edge $(u, t) \in U \times V$ occurs independently with probability p.

2. The considered regime is the sparse one, in the sense that $p = \frac{a}{n}$ with $a > 0$. This setup is motivated by online advertising, where the number of users greatly exceeds the number of ad campaigns, and only a small subset of users is eligible to participate.

3. The sequence of refills is a realization of independent Bernoulli random variables with parameter β/n, for some $\beta > 0$.

As emphasized previously, each node $u \in U$ is associated with a budget $b_{u,t} \in \mathbb{N}$. We add the additional assumption that the maximum budget per node is capped at some $K \in \mathbb{N}^*$ so that the budget dynamics are now expressed as:

$$b_{u,t} = \min(K, b_{u,t-1} - x_{u,t} + \eta_{u,t}), \qquad \text{with } b_{u,0} = b_0 \geq 1 \,.$$

Capping the maximum budget at K is motivated by the following considerations: it reflects practical constraints in real-world applications like online advertising, and it simplifies the analysis by reducing the problem to tracking a finite number of budget states over time.

As previously defined, an online *matching* on G generated by an algorithm ALG is a subset of edges, represented by a binary matrix $\mathbf{x} \in \{0,1\}^{n \times T}$, and must satisfy the same constraints introduced in the adversarial setting. In the stochastic setting, the performance of an algorithm can be measured either by the expected size of matching it creates or by the ratio between expected matching sizes of ALG and OPT. Formally, the different quantities we shall consider are

$$\mathrm{CR}^{\mathrm{sto}}(\mathtt{ALG}, \mathcal{D}) = \frac{\mathbb{E}_{G \sim \mathcal{D}}[\mathtt{ALG}(G)]}{\mathbb{E}_{G \sim \mathcal{D}}[\mathtt{OPT}(G)]}, \qquad \text{or the matching size} = \mathbb{E}_{G \sim \mathcal{D}}[\mathtt{ALG}(G)].$$

$\mathcal{D}$ denotes the distribution of the considered graph and the refill process. While the definitions of $\mathtt{ALG}(G)$ and $\mathtt{OPT}(G)$ remain the same, we will make the dependence on the time horizon T explicit when needed by writing $\mathtt{ALG}(G, T)$ and $\mathtt{OPT}(G, T)$.

The first result of this section identifies the asymptotic size of the matching generated by Greedy on the bipartite Erdős–Rényi model with budget refills. It establishes that, with high probability, the size of the matching generated by Greedy is close to the solution of a system of ordinary differential equations.

Theorem 9. *With probability* $1 - \mathcal{O}\left(n^{1/4} \exp(-a^3 n^{1/4})\right)$, *the matching size created by* Greedy *denoted by* $\mathtt{Greedy}(G, T)$ *satisfies,*

$$\mathtt{Greedy}(G, T) = nh(T/n) + \mathcal{O}(n^{3/4}).$$

and,

$$\frac{\mathbb{E}[\mathtt{Greedy}(G, T)]}{n} - h(T/n) \underset{n \to +\infty}{\to} 0$$

where $h(\tau)$ *is solution of the following equation,*

$$\dot{h}(\tau) = 1 - e^{-a(1 - z_0(\tau))}, \qquad \frac{1}{n} \leq \tau \leq \frac{T}{n}.$$

and $z_0(\tau)$ satisfies the following system,

$$
\begin{cases}
\dot{z}_0(\tau) = -z_0(\tau)\beta + \frac{z_1(\tau)}{1-z_0(\tau)}(1 - e^{-a+az_0(\tau)}) & \text{for } k = 0, \\
\dot{z}_k(\tau) = (z_{k-1}(\tau) - z_k(\tau))\beta + (z_{k+1}(\tau) - z_k(\tau))\frac{1-e^{-a+az_0(\tau)}}{1-z_0(\tau)} & \text{for } 1 \leq k \leq K-1, \\
\dot{z}_k(\tau) = \beta\, z_{k-1}(\tau) - z_k(\tau)\frac{1-e^{-a(1-z_0(\tau))}}{1-z_0(\tau)} & \text{for } k = K, \\
\sum_{k=0}^{K} z_k(\tau) = 1.
\end{cases}
$$

$$(6.1)$$

The proof of this result relies on tracking two coupled stochastic processes: the cumulative number of matches over time and the evolution of the number of the nodes in U with budget k. It is proved that these processes satisfy the assumptions of [98], which allows one to establish that the discrete, random dynamics of the system converge, with high probability, to the deterministic trajectory defined by Equation (6.1).

Once this convergence is established, the next step is to understand the long-term behaviour of the differential system itself. Although obtaining a closed-form solution is generally intractable according to the literature on differential equations, an alternative approach is to examine whether the system converges to a stable stationary state and to relate the limiting behavior of $\texttt{Greedy}(G,T)$ to this equilibrium. We analyze this in two settings: $K = 1$, where the system is reduced to two equations and the stationary state is shown to be exponentially stable, and the general case $K \geq 1$, where we prove that the stationary solution is asymptotically stable.

Corollary 1. *For $K \geq 1$, with probability at least $1 - 2\exp(-a^2 n^{\frac{3}{2}}/8T)$,*

$$
|\texttt{Greedy}(G,T) - nh^*(T/n)| \leq o(T).
$$

and,

$$
\frac{\mathbb{E}[\texttt{Greedy}(G,T)]}{n} - h^*(T/n) \underset{n \to +\infty}{\to} 0
$$

with $h^*(x) = \int_{1/n}^{x}(1 - e^{-a(1-z_0^*)})\mathrm{d}\tau = \left(x - \frac{1}{n}\right)\left(1 - e^{-a(1-z_0^*)}\right)$, and z_0^* is the unique solution of $\sum_{k=0}^{K} z_0^* \left(\frac{\beta}{g(z_0^*)}\right)^k = 1$ with $g(z_0^*) = \frac{1-e^{-a(1-z_0^*)}}{1-z_0^*}$.

For $K = 1$, with probability at least $1 - 2\exp(-a^2 n^{\frac{3}{2}}/8T)$,

$$
\left|\mathbb{E}[\texttt{Greedy}(G,T)] - T(1 - e^{-a(1-z_0^*)})\right| \leq c\frac{T}{(\log(T))^{3/4}} = o(T).
$$

where $z_0^* = \frac{1}{\beta} - \frac{1}{a}W\left(\frac{a}{\beta}e^{-a\left(1-\frac{1}{\beta}\right)}\right)$, with $W(\cdot)$ the Lambert function, and c is some universal constant.

The key difference between these cases lies in the nature of the stability: when $K = 1$, the system exhibits exponential stability, leading to rapid convergence to

the stationary state. In contrast, for $K \geq 1$, the stability is only asymptotic, which does not guarantee such a fast convergence.

On the CR side, the final main result of this section demonstrates that the competitive ratio of Greedy in this setting converges to 1 as T, K and n grow significantly.

Theorem 10. *For any $\alpha, \beta > 0$, the competitive ratio tends to 1, as T, K, n approach infinity, as*

$$\lim_{K,n \to +\infty} \lim_{T \to +\infty} \mathrm{CR}^{\mathrm{sto}}(\texttt{Greedy}, \mathcal{D}) = 1.$$

This conclusion is based on establishing a lower bound on $\mathrm{CR}^{\mathrm{sto}}$, which depends on $(z_0^*, \ldots, z_K^*)$ the stationary solution of Equation (6.1). The bound is derived through an exact calculation of the matching size achieved by the Greedy algorithm and an upper bound on the matching size generated by OPT. As T, K, and n grow significantly, the gap between Greedy and the optimal algorithm vanishes, leading the competitive ratio to converge to 1.

6.3 Online matching on stochastic block model

We study the online bipartite matching problem on a stochastic block model (SBM) [30], where the bipartite graph $G = (U, V, E)$ is defined by a set of offline nodes $U = [n] = \{1, \ldots, n\}$ and a set of online nodes $V = [T] = \{1, \ldots, T\}$, where $T, n \in \mathbb{N}^*$. The edges $(u, t) \in E$ are included independently based on latent class labels. Each offline node $u \in U$ and online node $t \in [T]$ is assigned a class from sets $\mathcal{C} = [C]$ and $\mathcal{D} = [D]$, where $C, D \in \mathbb{N}^*$, drawn independently from distributions μ and ν respectively. Given the class labels, the edge (n, t) appears in $\mathcal{E}$ with probability $p_{u,t} = p(c(u), d(t)) \in [0, 1]$, where $p = (p(c, d))_{c,d \in \mathcal{C} \times \mathcal{D}}$ is a class-to-class affinity matrix. In this work, we focus on the *sparse regime*. More precisely, we assume the existence of a non-negative matrix $a = (a_{c,d}) \in \mathbb{R}_+^{C \times D}$ such that $p(c, d) = \frac{a_{c,d}}{N}$ with $a_{c,d} \leq a$ for all $c \in \mathcal{C}$, $d \in \mathcal{D}$, where $a \in (0, n)$. This ensures that offline nodes have bounded expected degrees as $n \to \infty$, reflecting real-world constraints.

To formalize the process, we let $T = \alpha n$ for some $\alpha > 0$, and use b_c to denote the proportion of offline nodes in class c. For a given matching algorithm ALG, we define the Boolean variable $m_u(t)$ to be equal to 1 if, and only if, the vertex u has been included in the matching by ALG before the vertex t arrives (otherwise $m_u(t) = 0$). Additionally, we denote by $\mathcal{N}_c := \{u \in U, c(u) = c\}$ the set of nodes of class $c \in \mathcal{C}$, and $\mathcal{M}_c(t) = \{u \in \mathcal{N}_c, m_u(t) = 1\}$ the set of matched vertices of class c before seeing vertex $t \in V$ (we denote by $M_c(t)$ its cardinality) and by $M(t) := \sum_{c \in \mathcal{C}} M_c(t)$ the size of the matching constructed so far. Finally, we shall denote by e_i the i-th basis vector of $\mathbb{R}^C$.

Depending on whether the compatibility probabilities $p_{u,t}$ are known in advance or must be learned over time, different algorithmic approaches and theoretical results are developed in the following sections.

6.3.1 $p_{u,t}$ known

The first result of this part concerns the performance of the `Myopic` algorithm, which makes purely greedy decisions without attempting to look ahead or anticipate future availability. When a vertex $t \in V$ arrives, the algorithm selects a class $c_t \in \mathcal{C}$ according to a fixed probability distribution, and then attempts to match it to an available node within that class. The selection is made without verifying whether any nodes in the chosen class are actually available at time t.

To understand how this simple strategy performs in the stochastic block model, we provide an approximation for the expected matching size it produces. More precisely, we show that the evolution of the matching under `Myopic` closely follows the solution of a differential equation that captures the dynamics of the process over time.

Theorem 11. *Let $y_c : [0, \alpha] \to \mathbb{R}$ be the solution of the following ODE*

$$\begin{cases} \dot{y}_c(s) & = \sum_{d=1}^{D} \left(1 - e^{-a_{c,d}(b_c - y_c(s))}\right) Q^*(c,d), \\ y_c(0) & = 0. \end{cases} \tag{6.2}$$

Then, for each class $c \in \mathcal{C}$, the matching size $M_c(t)$ produced by `Myopic` *satisfies, for all $t \in [T]$*

$$\left| \frac{M_c(t)}{n} - y_c(t/n) \right| \leq \frac{3 L_c e^{\alpha L_c}}{n^{1/3}}, \quad \text{where } L_c = \sum_{d=1}^{D} a_{c,d} Q^*(c,d),$$

with probability at least $1 - 2Ce^{-n^{1/3}L_c^2/8\alpha}$. Moreover, for $c \in [C]$, $y_c(t) = \tilde{y}_c(t) - e_c(t)$, where $e_c(0) = 0$, and $\tilde{y}_c(t) = b_c - b_c \exp(-tL_c)$, and e_c satisfies,

$$e_c(t) \leq \frac{J_c}{L_c}(1 - e^{-L_c t}).$$

Where $J_c = \frac{b_c^2}{2} \sum_{d=1}^{D} a_{c,d}^2 Q^*(c,d)$.

This result shows that the matching size achieved by `Myopic` in this model can be approximated by the solution of Equation (6.2). Due to the complexity of the system, obtaining a closed-form solution for 6.2 is not tractable. Instead, we derive an approximate solution with a controlled error bound, ensuring that the approximation remains accurate.

The second result of this part builds on the limitations of `Myopic` and takes into account node availability. Here, we study `Ex-ante Balance`, which chooses a

class c that maximizes the probability that at least one unmatched node is available and connected to the arriving node (the detailed algorithm can be found in Algorithm 12). The following result shows that the matching size created by `Ex-ante Balance` is, with high probability, close to the solution of a differential inclusion Section 8.A.

Theorem 12. *Let m be the unique solution of the differential inclusion*

$$\dot{m} \in F(m) := conv\left\{ f_{c,b_c}(m_c)e_c \; ; \; c \in \arg\max_{k \in [C]} f_{k,b_k}(m_k) \right\},$$

which is the convex hull of the mappings

$$f_{c,b_c}(x) = \sum_{d=1}^{D}(1 - e^{-a_{c,d}(b_c - x)})\nu(d).$$

Then the matching built by `Balance` satisfies for all $t \in [T]$ and $c \in \mathcal{C}$, with probability at least $1 - \frac{b\alpha}{N\epsilon^2}$,

$$\left| \frac{M_c(t)}{n} - m_c(t/n) \right| \leq \min(\alpha, e^{L\alpha}/\sqrt{2L})\sqrt{A_{\alpha,c}/N + \delta_c B_{\alpha,c} + \epsilon C_{\alpha,c}}, \qquad (6.3)$$

The different constants in are defined by $L = \max_{c \in [C]} \sum_{d=1}^{D} a_{c,d}\nu(d)$, $\delta_c = \frac{\sum_{d=1}^{D} \frac{a_{c,d}}{e}\nu(d)}{N}$, $K_\alpha = (c\alpha + \epsilon)e^{c\alpha}/c$, ϵ as defined in Lemma 35 and c in Lemma 31, $U_c = \sum_{d=1}^{D}(1 - e^{-a_{c,d}b_c})\nu(d)$, $A_{\alpha,c} = U_c(U_c^2 + \frac{14U_c}{3} + 2K_\alpha)$, $B_{\alpha,c} = 2U_c^2 + 4L\delta_c + 12K_\alpha$, $C_{\alpha,c} = 2U_c^2 + 4L\epsilon + 8K_\alpha$.

Here, the convergence is toward a differential inclusion rather than an ordinary differential equation, due to the discontinuities in the decision rule of the algorithm. Unlike `Myopic`, whose matching decisions are Lipschitz-continuous and can be well approximated by an ODE, `Ex-ante Balance` dynamically selects the class that maximizes the probability of a match. This introduces abrupt changes in the system's evolution, breaking the continuity assumptions required for standard ODE convergence. To handle this complexity, we instead characterize the limiting behavior of `Ex-ante Balance` using a differential inclusion, which is more suitable for this context as it captures the possible trajectories of the system. Although finding a closed-form solution to a differential inclusion is generally difficult, we exploit the fact that `Ex-ante Balance` implicitly balances match probabilities across classes. Leveraging this structure, we derive an explicit closed-form expression for the limiting behavior. Further details can be found in Chapter 8.

6.3.2 $p_{u,t}$ **unknown**

In this case, we study the setting in which the connection probability parameters $a_{c,d}$ are unknown and must be estimated online. This transforms the problem into a

bandit setting, where each class $c \in \mathcal{C}$ can be viewed as an arm. When a class c_t is
selected at time t, a Bernoulli reward is observed—indicating whether a successful
match occurred between the arriving node and an available node in class c_t. Unlike
the classical multi-armed bandit setting, the expected reward associated with an
arm is not stationary: it depends on the dynamics of the system, in particular
on the availability of nodes in each class. As a result, there is no notion of a
"best arm", which makes the problem more complex. One solution is to introduce
the algorithm $\texttt{ETC} - \texttt{balance}$, which combines a fixed exploration strategy of the
Explore-Then-Commit (ETC) type with the selection rule $\texttt{Ex-ante Balance}$. The
performance of $\texttt{ETC} - \texttt{balance}$ is evaluated in terms of regret, defined as the gap
between the cumulative matches achieved by an oracle with full knowledge of the
probabilities and those achieved by $\texttt{ETC} - \texttt{balance}$. When the exploration phase
lasts for $T_{\text{explore}} = T^{\frac{q+3}{4}}$ steps, for any $0 < q < 1$, the regret satisfies the following
bound:

Theorem 13. *Let $R(T) = \sum_{i \in \mathcal{C}} M_i(T) - \hat{M}_i(T)$ denote the regret of $\texttt{ETC} - \texttt{balance}$.*
Suppose the exploration phase lasts for $T_{explore} = T^{\frac{q+3}{4}}$, for some $0 < q < 1$. Then
the regret satisfies
$$R(T) = \mathcal{O}(T^{\frac{q+3}{4}}).$$

6.4 Optimizing the coalition gain in Online Auctions with Greedy Structured Bandits

In this work [15], we consider T ad impressions sold through *second-price auctions*
(see [69] for a survey). At auction $t \in [T]$, each participant (bidder) bids on the
item based on their own (stochastic) value for the item. The highest bidder wins
the item and pays a price equal to the second-highest bid. The *decision maker*
runs $N \in \mathbb{N}^*$ advertising campaigns forming a *coalition*. At time t, two groups of
bidders participate: (1) $n_t \in [N]$ bidders from the coalition chosen by the decision
maker *ex-ante* –, without knowing the realization of the bidders' values – and (2)
$p \in \mathbb{N}^*$ other bidders, that we call the *competition*. When a bidder from the coalition
wins the auction, the decision maker observes the realized value of the winner (also
called *winning bid*). In this work, we consider that all bidders are identical, their
values are i.i.d samples from a distribution supported on $[0, 1]$, whose cumulative
distribution function is denoted by F. Under this assumption, the expected reward
of the decision maker at time t is given by $r(n_t)$, where r is defined by,

$$r : n \in [N] \mapsto r(n) \coloneqq \mathbb{E}_{\mathbf{v}=(v_i)_{i \in [n+p]} \sim F \times \cdots \times F}\left[(\mathbf{v}_{(1)} - \mathbf{v}_{(2)}) \mathbb{1}\left\{ \arg\max_{i \in [n+p]} v_i \in [n] \right\} \right]. \quad (6.4)$$

where $\mathbf{v}_{(1)}$ and $\mathbf{v}_{(2)}$ are respectively the first and second maximum of $\mathbf{v}$, and $[n]$ is used
to abbreviate $\{1, \ldots, n\}$. This setup naturally leads to a multi-armed bandit (MAB)
problem, where the decision maker chooses *arms* $n_1, \ldots, n_T \in [N]$ sequentially and

aims to minimize its cumulative *expected regret* $\mathcal{R}(T)$ defined by

$$\mathcal{R}(T) = \sum_{t \leq T} r(n^*) - r(n_t) \, , \quad \text{with} \quad n^* = \arg\max_{n \in [N]} r(n). \qquad (6.5)$$

Using order statistics techniques Equation (6.4) can be equivalently written as:

$$n \in [N] \mapsto r(n) = n \int_0^1 F^{p+n-1}(x) - F^{p+n}(x) \mathrm{d}x. \qquad (6.6)$$

This integral expression highlights that the reward function defined in Equation (6.6) can be unimodal for some choices of F. In the remainder of this work, we focus on distributions that ensure unimodality.

Leveraging this structure, the first step in this project was to estimate r. From Equation (6.6), it appears that estimating both $F^{p+n-1}(x)$, and $F^{p+n}(x)$ is sufficient to estimate $r(n)$, thus we built $\hat{r}_k(n)$, an estimator of $r(n)$ using samples drawn from any arm k and based on powers of the empirical CDF, defined by,

$$\widehat{r}_k(n) = n \int_0^1 \left(\widetilde{F}_{k+p}^{n+p-1}(x) - \widetilde{F}_{k+p}^{n+p}(x) \right) \mathrm{d}x \, . \qquad (6.7)$$

where $\widetilde{F}_{k+p}^{\ell} : x \mapsto (\widehat{F}_{k+p}(x))^{\frac{\ell}{k+p}}$, $\widehat{F}_{k+p} : x \mapsto \frac{1}{m_k} \sum_{j=1}^{m_k} \mathbb{1}\{w_{k,j} \leq x\}$ (emp. c.d.f. of $\overline{W}_k$), and $\overline{W}_k = (w_{k,1}, \ldots, w_{k,m_k})$ the feedback gathered after playing arm k and winning the auction m_k times. Based on the unimodality assumption, and restricting the range of arms that can be used to estimate the reward, we get the following concentration result,

Theorem 14. *Consider any $n \in [N]$ and $k \in \mathcal{V}(n)$. Let $\widehat{r}_k(n)$ be defined according to (9.5) from m_k samples collected by k. Then, there exists some constants $\beta_{k,n}$ (depending on n, k, p) and $\xi_{k,n,F}$ (additionally depending on F) such that, with probability $1 - \delta$,*

$$|\widehat{r}_k(n) - r(n)| \leq \beta_{k,n} \sqrt{\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}} + n \times \xi_{k,n,F} \left(\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k} \right)^{\frac{n+p-1}{k+p}} . \qquad (6.8)$$

Furthermore, the constants admit universal upper bounds for any n, k, p, F. For instance if $m_k \geq 4$ it holds that $\beta_{k,n} \leq 33$ and $\gamma_{k,n,F} \leq 100$.

With this concentration result in hand, we propose two strategies to tackle the problem defined earlier: The first strategy is Local-Greedy (`LG`), which is a natural adaptation of a standard policy in unimodal bandits, `OSUB` [31]. The main idea of `OSUB` is to play `UCB` locally around a reference arm, and eventually reach the optimal arm $n^\star$ by gradually moving the reference arm in its direction. With `LG`, we adapt this principle to efficiently exploit the structure of the problem considered: at each round t, `LG` defines a reference arm ℓ_t, called the *leader*, but plays *greedily* in

the *neighborhood* $\mathcal{V}(\ell_t)$, based on simple power estimates computed using samples from ℓ_t only. In addition, a *sampling requirement*, implemented by a parameter $\alpha \in (0,1)$, is used to ensure good concentration of these estimates. The second strategy is GG is based on a very intuitive idea: it plays a Local-Greedy strategy only if it can identify, with high probability, which segment of the reward function contains the best arm. To implement this idea, GG uses a Successive Elimination procedure [39] on a *subset* of arms forming a *reference grid*, denoted by $\mathcal{S}$, which serves as a coarse discretization of the action space to efficiently localize the region containing the optimal arm. By operating over this subset, GG quickly narrows down the most promising region and then applies Local-Greedy within that segment. This hybrid approach combines coarse global exploration with focused local exploitation, enabling efficient and reliable identification of high-performing arms.

For these strategies, we obtain the following results on the regret,

Theorem 15. *Let $\Delta := \min_{n \in [N-1]} |r(n+1) - r(n)|$ (worst local gap). Under unimodality assumption and with $\alpha = (\log_{3/2} N + 1)^{-1}$, the regret of LG is upper bounded by a **problem-dependent constant**: there exists $(C_n)_{n \in [N] \setminus \{n^\star\}}$, each satisfying $C_n = \widetilde{\mathcal{O}}_N \left(\frac{\Delta_n}{\Delta^2} \right)$, such that $\mathcal{R}_T \leq \sum_{n \in [N] \setminus n^\star} C_n$.*

Additionally, if the arm set forms a single estimation neighborhood, that is $\forall n \in [N] : \mathcal{V}(n) \supset [N]$, then each constant C_n can be refined to $\widetilde{\mathcal{O}}_n(\Delta_n^{-1})$, providing $\mathcal{R}_T = \widetilde{\mathcal{O}}(\sqrt{NT})$, which holds even when the reward function is not unimodal.

Suppose that GG is tuned with confidence level $\delta_t = \frac{1}{N^2 t^3}$, and $\alpha = 1/4$. Then, for any $T \in \mathbb{N}$ it holds that

$$\mathcal{R}_T = \widetilde{\mathcal{O}}_N \left(\sum_{n \in \mathcal{B}^\star} \frac{1}{\Delta_n} + \sum_{n \in \mathcal{S}} \frac{\log(T)}{\Delta_n} \wedge \Delta_n \left(\frac{\mathbb{1}\{n < n^\star\}}{\Delta_{v_l(n^\star)}^2} + \frac{\mathbb{1}\{n > n^\star\}}{\Delta_{v_r(n^\star)}^2} \right) \right) .$$

Additionally, it holds that $\mathcal{R}_T = \widetilde{\mathcal{O}} \left(\sqrt{(K + |\mathcal{B}^\star|)T} \right)$, for $K = \lfloor \log_{3/2}(N) \rfloor$.

While traditional bandit approaches yield problem-dependent bounds depending on T, the algorithms GG and LG presented in this work have constant problem-dependent bounds. Furthermore, GG and LG avoid a quadratic dependency in N for large T thanks to new concentration bounds. Overall, while GG has the best theoretical guarantees, LG has better constants and is therefore better suited for most practical applications (see the discussion at the end of Section 9.3 and experimental results in Section 9.D).

In the following chapters, we present each of the three projects in detail. Each chapter is based on work that has been submitted to major conferences or already published in peer-reviewed venues.

6

7
Online matching with budget refills

This chapter is based on [29], which was presented at the *Matchup Workshop 2024* and is currently under review for publication at *ICMLCN 2026*.

Inspired by sequential budgeted allocation problems, we explore the online matching problem with budget refills. Specifically, we consider an online bipartite graph $G = (U, V, E)$, where the nodes in V are discovered sequentially and the nodes in U are known beforehand. Each $u \in U$ is endowed with a budget $b_{u,t} \in \mathbb{N}$ that dynamically evolves over time. Unlike the canonical setting, where budgets are fixed, many practical applications involve periodic budget refills. This added dynamic introduces a richer and more complex problem, which we investigate here. Intuitively, adding extra budgets in U seems to ease the matching task, and our results support this intuition. In fact, for the stochastic framework considered, where we analyze the matching size built by the `Greedy` algorithm on an Erdős–Rényi random graph, we show that the matching size generated by `Greedy` converges with high probability to the solution of an explicit system of ordinary differential equations (ODEs). Moreover, under specific conditions, the competitive ratio (the performance measure of the algorithm) can even tend to 1. For the adversarial part, where the graph considered is deterministic and the algorithm used is `Balance`, the b-matching bound holds when refills are scarce. However, when refills are regular, our results suggest a potential improvement in the algorithm's performance. In both cases, the `Balance` algorithm manages to reach the performance of the upper bound on the adversarial graphs considered.

Contents

7.1 Introduction

Finding matchings in bipartite graphs is a fundamental problem that lies at the intersection of graph theory [46, 101], network theory, and combinatorial optimization [74, 90], with far-reaching implications across a wide range of practical applications, particularly in operations research, where it is often referred to as the assignment problem (see also [48]). Specifically, a bipartite graph, denoted as $G = (U, V, E)$, consists of two distinct sets of nodes, U and V, and a set of edges $E \subseteq U \times V$. These graphs provide a natural representation of systems in which entities from one set are paired with entities from the other, modeling relationships, dependencies, or allocations. Solving the matching problem involves identifying optimal pairings between nodes in the two sets while satisfying specific constraints.

The standard online matching problem. Recent real-world applications, particularly in online advertising, have generated significant interest in the online variant of the matching problem (see [81]). In this version, the graph $G = (U, V, E)$ consists of two sets of nodes $U = [n] := \{1, \ldots, n\}$ and $V = [T] := \{1, \ldots, T\}$ for $n, T \in \mathbb{N}^*$ and set of edges $E \subseteq U \times V$. The nodes in U are known beforehand, while the nodes in V arrive sequentially along with their associated edges. Throughout this section, we adopt the standard convention in online matching whereby each element $t \in V$ denotes both a node of the graph and the time step at which that node arrives. This identification is classical in the literature and reflects the fact that one node is revealed per time step. Each node $u \in U$ has a budget $b_{u,t} \geq 0$, with an initial budget $b_{u,0} = 1$ for all $u \in U$.

When a node $t \in V$ is observed, an online algorithm must decide whether to match it with a node $u \in U$ such that $(u, t) \in E$ and u have not yet been matched (i.e. $b_{u,t} = 1$). Once a matching decision is made, it cannot be changed, making the process irreversible. The budget of each node $u \in U$ at time $t \in V$ evolves according to the following dynamics:

$$b_{u,t} = \begin{cases} b_{u,t-1} - 1 & \text{if } u \text{ is matched to } t, \\ b_{u,t-1} & \text{otherwise.} \end{cases}$$

Building on the foundational online matching problem, several generalizations have been introduced, including the b-matching problem [61, 67]. In this extension, nodes in U are assigned an initial budget greater than 1 (i.e., $b_{u,0} > 1$ for all $u \in U$). This modification allows a vertex $u \in U$ to be matched multiple times, up to its remaining budget. A vertex $t \in V$ can only be matched to a vertex $u \in U$ if u has at least one unit of budget left, as studied for instance in [62, 6, 7]. To address both the standard online matching problem and its extensions, several online algorithms have been proposed. For example, the `Greedy` algorithm matches each arriving vertex from V to any available neighbor in U, while the `Balance` algorithm pairs an arriving vertex with the neighbor in U with the highest remaining budget. The performance of these algorithms is evaluated through their *competitive ratio*, defined as the ratio between the number of edges in the matching produced by the online algorithm and the number of edges in an optimal offline b-matching. This generalizes the classical definition used in standard online matching (see [81, 40]).

The online matching with budget refills setting. Motivated by the dynamic nature of online advertising, we model the scenario where U represents the pool of campaigns or ads available to advertisers, and the nodes in V correspond to the advertising slots that arrive sequentially. Each slot has varying eligibility for a subset of campaigns, determined by factors such as geographic location, browsing history, and other relevant features. The main objective of the advertiser is to maximize the number of ads displayed. In practice, campaigns are not displayed only once but come with a predetermined budget for impressions; for example, a particular

ad may be shown no more than 10,000 times per day. This budget can evolve over time, with the possibility of allocating additional resources to certain campaigns at periodic intervals, thereby motivating the model explored in this work.

Formally, we consider the same graph $G = (U, V, E)$ as in the standard matching problem. However, unlike the b-matching setting, an additional layer of complexity arises due to the dynamic nature of the budget for each node $u \in U$ at time $t \in V$. The budget, denoted $b_{u,t} \in \mathbb{N}$, decreases whenever u is matched but can also be replenished according to a replenishment process governed by $\eta_{u,t} \in \mathbb{N}$. In this work, we focus on a simple replenishment model in which budgets are refilled at a constant average rate over time. This dynamic introduces significant challenges in analyzing budget evolution and its impact on matching performance.

$$b_{u,t} = \begin{cases} b_{u,t-1} - 1 + \eta_{u,t} & \text{if } u \text{ is matched to } t, \\ b_{u,t-1} + \eta_{u,t} & \text{otherwise.} \end{cases}$$

The budget refills. As mentioned above, the concept of budget refills in online matching is inspired by online advertising, where advertisers often replenish their budgets during ad campaigns to sustain or increase their exposure and, ultimately, their revenue. These dynamics allow advertisers to adapt and optimize their spending based on campaign performance over time. In this work, we focus on two distinct settings for budget refills. The first setting is stochastic, reflecting the uncertainty and variability often observed in real-world scenarios. Here, the refills follow a probabilistic model governed by a Bernoulli random variable with parameter $\frac{\beta}{n}$, where $\beta > 0$ and n represents the number of nodes in U:

$$\eta_{u,t} \sim \mathcal{B}(\beta/n).$$

The choice of a parameter that depends on n arises from our focus on the sparse regime of the graph in this case. By applying refill dynamics to the considered graph, we aim to explore how budget evolution affects the matching process under a straightforward refill mechanism. This approach prioritizes clarity and avoids the added complexity of more sophisticated random variables, which could obscure key insights into the dynamics at play. The second setting involves an adversarial context, where both the graph and the refill dynamics are deterministic. In this scenario, the graph structure is chosen adversarially, meaning the edges are designed in a way that could potentially hinder the matching process. However, the refills themselves are not adversarial; they follow a predefined deterministic evolution, adding a unit of budget every fixed number of time steps m:

$$\eta_{u,t} = \mathbb{1}_{\{t \bmod m = 0\}}.$$

This framework allows us to analyze the interaction between predictable budget replenishment and a challenging graph structure. Despite their inherent differences, the stochastic and adversarial settings exhibit noteworthy similarities. In

the stochastic case, budget evolution is modeled using Bernoulli random variables with an expected value of β/n. Likewise, in the adversarial setting, budget increments occur deterministically at fixed intervals, leading to a refill frequency of $1/m$, where m is determined based on n. By considering both stochastic and adversarial perspectives, we gain a broader understanding of budget refill dynamics and their implications for online matching and decision-making, particularly in the context of online advertising.

The contributions. Our main contributions, addressing both the stochastic and adversarial frameworks, are summarized as follows:

- In the adversarial case with relatively many refills (i.e., m is negligible compared to $\sqrt{T}$), we show that the initial budgets have no significant impact on the asymptotic competitive ratio. We also derive an upper bound for the competitive ratio of `Balance`, demonstrating that it is optimal for the specific graph used in our proof. This bound, which applies to any algorithm, is given by $1 - \frac{1-\alpha}{e^{1-\alpha}}$, where $\alpha \simeq 0.603$.

- In the adversarial case, with relatively few refills, i.e., when m is of order (larger than) $\sqrt{T}$, we demonstrate that refills have a negligible effect on the competitive ratio of the `Balance` algorithm, which matches that of the b_0-matching problem (i.e., the problem without refills). Interestingly, the refill frequency $1/m$ does not appear in the competitive ratio. Stated otherwise, the dominating effect is the initialization of the budgets.

- In the stochastic framework, we analyze the asymptotic performance of the `Greedy` algorithm on the Erdős–Rényi model. Our approach demonstrates that, with high probability, the discrete and random matching process closely follows the continuous and deterministic solution of a system of ordinary differential equations. Furthermore, we analyze the stability of the stationary solution of this system to derive a closed-form expression for the performance of `Greedy`. Finally, in terms of competitive ratio, we establish a lower bound on the competitive ratio that depends on different parameters of the problem. Notably, this lower bound converges to 1 as these parameters approach infinity, indicating near-optimal performance in such limiting cases.

7.2 The adversarial framework

The initial strand of research focused on online matching (without refills) within the adversarial framework, where the algorithm is evaluated on the worst possible instance and vertex arrival order. Notably, the `Greedy` algorithm, which matches incoming vertices with any available neighbors, achieves a competitive ratio of $1/2$ in the worst-case scenario. However, its performance improves, reaching a $1 - 1/e$ competitive ratio when incoming vertices arrive in a random order, as highlighted in

[47]. Another significant contribution is the `Ranking` algorithm introduced in [64], which is worst-case optimal, consistently achieving at least $1 - 1/e$ on any instance [64, 35, 16]. Moreover, it exhibits superior performance in scenarios featuring random vertex arrivals [76]. Beyond traditional online matching, the b-matching problem assigns fixed budgets $b \in \mathbb{N}^*$ to nodes in U, as pioneered in [62]. In this context, [62] introduced the deterministic `Balance` algorithm, which matches a new vertex in V with a neighbor in U that has the highest remaining budget. They proved that `Balance` achieves an optimal competitive ratio of $1 - (1/(1 + 1/b)^b)$, tending towards $1 - 1/e$ as b grows. Furthermore, [6] explored a broader setting where nodes within U possess varying budgets b_u. Using primal-dual methods, they showed that the `Balance` algorithm achieves a competitive ratio of $(1/(1 + 1/b_{\min})^{b_{\min}})$, where $b_{\min} = \min_{u \in U} b_u$.

7.2.1 Model

To study the online matching problem in the adversarial setting with budget refills, it is essential to impose some restrictions on the adversary's power. Therefore, we adopt the following assumptions regarding the model being considered:

1. The sequence of refills $(\eta_{u,t})_{u \in U, t \in V}$ is a parameter of the problem, set and known in advance to a refill of one unit every m time steps.

2. Every node $t \in V$ has at least one neighbor in U.

If the sequence of refills $(\eta_{u,t})_{u \in U, t \in V}$ were to be chosen in an adversarial fashion, the adversary would simply set it to 0, reducing the problem to the classical b-matching problem. Moreover, to prevent this reduction, we assume that each node $t \in V$ has at least one neighbor in U. Furthermore, the choice of a refill of one unit every m time steps comes from the motivating application of advertising, where advertisers usually renew their budgets monthly or quarterly. Additionally, considering a constant value for refills provides a clear and simple setting to disentangle the asymptotic effect of the refills from the initialization of budgets.

Formally, the subset of graphs from which the oblivious adversary can choose is the following,

$$\mathcal{G}_{T,m} = \Big\{ (U, V, E, (\eta_{u,t})_{u \in U, t \in V}) \ : \ \forall t \in V, \ \exists u_t \in U \text{ such that } \eta_{u_t,t} = \mathbf{1}_{\{t \equiv 0 \ (\mathrm{mod} \ m)\}}$$
$$\text{and } (u_t, t) \in E \Big\}.$$

The budget evolution for each $u \in U$ is dependent on whether the edge $(u, t) \in E$ is included in the online matching, which is indicated by the binary variable $x_{u,t} \in \{0, 1\}$, as well as whether t is a multiple of m. Formally, this dynamic can be

expressed as follows:

$$\forall u \in U, \quad b_{u,t} = b_{u,t-1} - x_{u,t} + \mathbb{1}_{\{t \bmod m = 0\}}, \qquad \text{and} \qquad b_{u,0} = b_0 \text{ for some } b_0 \geq 1.$$

As a consequence, the online *matching* on $G \in \mathcal{G}_{T,m}$ generated by an algorithm ALG is the subset of edges that can be represented by a binary matrix $\mathbf{x} \in \{0,1\}^{n \times T}$ that must satisfy the following constraints:

1. $\forall (u,t) \in U \times V, \ (u,t) \notin E \Rightarrow x_{u,t} = 0$ (only edges in E can be matched).

2. $\forall t \in V, \ \sum_{u \in U} x_{u,t} \leq 1$ (V-nodes can only be matched once).

3. $\forall (u,t) \in U \times V, \ b_{u,t-1} < 1 \Rightarrow x_{u,t} = 0$ (U-nodes need positive budget to be matched).

In online bipartite matching problems, the performance of an algorithm ALG is evaluated by its competitive ratio, which is the ratio between the size of the matching ALG has created and the largest possible matching in hindsight, also referred to as OPT with matrix $\mathbf{x}^*$. The rationale is that the optimal matching of some deterministic graph G can be arbitrarily small. Hence, the constructed matching size alone does not provide any good insight on the "quality" of an algorithm in the adversarial case. Formally, in the adversarial framework, the objective of the algorithm is to obtain the highest worst-case competitive ratio $\text{CR}^{\text{adv}}(\text{ALG}, \mathcal{G}_{T,m})$, defined as follows:

$$\text{CR}^{\text{adv}}(\text{ALG}, \mathcal{G}_{T,m}) = \min_{G \in \mathcal{G}_{T,m}} \frac{\text{ALG}(G)}{\text{OPT}(G)}.$$

where $\text{ALG}(G) = \sum_{u \in U} \sum_{t=1}^{T} x_{u,t}$, and $\text{OPT}(G) = \sum_{u \in U} \sum_{t=1}^{T} x^*_{u,t}$, are the sizes of the matching generated by ALG and OPT respectively.

As previously highlighted, our analysis focuses on evaluating the Balance algorithm within the mentioned model. We aim to dissect the impact of the initial budget b_0 and the refill process by parameterizing our results with T, which is both the finite horizon and the size of V. This choice slightly limits the adversary's power, as it cannot impact the length of the horizon T by simply providing no edge for an arbitrary number of time steps.

7.2.2 Main results

7.2.2.1 Regime $m = \omega(\sqrt{T})$

Recall that the notation $m = \omega(\sqrt{T})$ means that m grows strictly faster than $\sqrt{T}$ as $T \to \infty$, i.e., $\frac{m}{\sqrt{T}} \longrightarrow \infty$ as $T \to \infty$. Intuitively, in the regime $m = \omega(\sqrt{T})$,

the performance within a single period of length m can have a dominant influence on the overall CR. While this is clearly the case when $m = T$, it also holds true even for $m = \omega(T)$.

Theorem 16. *Assuming the initial budgets are $b_{1,0} = b_{2,0} = \cdots = b_{n,0} = b_0 \geq 1$. If $m = \omega(\sqrt{T})$ and $b_0(b_0 + 1)^{b_0} \leq m$, then,*

$$\sup_{\text{ALG}:\textit{deterministic}} \text{CR}^{\text{adv}}(\text{ALG}, \mathcal{G}_{T,m}) \leq 1 - \frac{1}{\left(1 + \frac{1}{b_0}\right)^{b_0}} + o_T(1).$$

The bound is reached for the graph defined in the proof.

Sketch of proof. The complete proof is provided in section 7.A.1. It relies on using $d = \left\lfloor \frac{m}{|V_K|} \right\rfloor$ duplicates of the graph $G_K = (U_K, V_K, E_K)$ presented in [62], where the size depends on b_0. More precisely, it uses d copies of G_K at the beginning of the process and during the remaining time $T - m$ only one node $\tilde{u}$ from U is connected with all the remaining edges in V (see 7.1a for illustration). Then, the number of edges matched by ALG and OPT during these $T - m$ last steps is the same, denoted γ_T which is at most $\left\lfloor \frac{T}{m} \right\rfloor$ as it relies on the refills of $\tilde{u}$ only (see section 7.A.1 for more details). Thus,

$$\text{CR}^{\text{adv}}(\text{ALG}, \mathcal{G}_{T,m}) \leq \frac{d\text{ALG}(G_K) + \gamma_T}{d\text{OPT}(G_K) + \gamma_T}.$$

Since $\gamma_T = o(\sqrt{T})$, we can conclude that,

$$\text{CR}^{\text{adv}}(\text{ALG}, \mathcal{G}_{T,m}) \leq 1 - \frac{1}{\left(1 + \frac{1}{b_0}\right)^{b_0}} + o_T(1).$$

$\square$

7.2.2.2   Regime $m = o(\sqrt{T})$

In the regime $m = o(\sqrt{T})$, where the refills dominate the initialization, the upper bound on the CR is weaker. Unlike the previous scenario where it was bounded by $1 - \frac{1}{e} \approx 0.63$, it is now bounded only by 0.73. The following theorems establish this upper bound for **Balance**, and demonstrate that no algorithm can achieve significantly better performance.

Theorem 17. *Assuming the initial budgets are $b_{1,0} = b_{2,0} = \cdots = b_{n,0} = b_0 \geq 1$. For $m = o(\sqrt{T})$ and $mb_0 = o(T)$, then,*

$$\text{CR}^{\text{adv}}(\texttt{Balance}, \mathcal{G}_{T,m}) \leq \underbrace{1 - \frac{(1-\alpha)}{e^{(1-\alpha)}}}_{\simeq 0.73325...} + o_{m,T}(1).$$
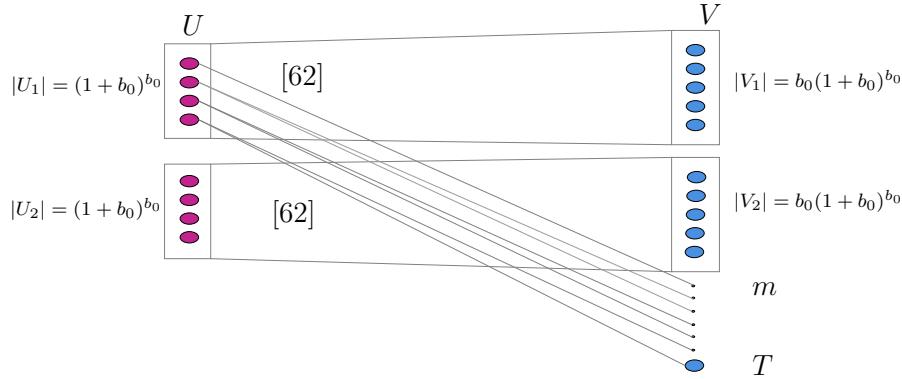
where α is defined by $\frac{1}{2} = \int_0^{\alpha} \frac{xe^x}{1-x}\mathrm{d}x$. The bound is reached for the graph defined in the proof.

Sketch of proof. The full proof is provided in section 7.A.2. It relies on building an adversarial graph $G^{\text{th.2}} = (U, V, E)$ with the following structure (see 7.1b for an illustration): initially, for a period of size $t_0 \simeq \frac{T}{e}$, the size of U exceeds m ($|U| \simeq 2m - 1$), allowing the algorithm to accumulate a significant amount of budget. During this period, `Balance` and `OPT` consistently match nodes and accumulate an equal amount of budget on U, but it is not distributed in the same way. At time t_0, the adversary removes all but $m - 1$ nodes from U (starting with those with the highest budgets). Specifically, when the adversary eliminates nodes, it has no impact on `OPT` because `OPT` has perfect hindsight knowledge of the eliminated nodes. Therefore, it can allocate the budget exclusively to nodes that remain available, ensuring that no budget is lost on eliminated nodes at the time of their removal. However, `Balance` remains unaware of which nodes will be eliminated. Consequently, at the time of elimination, the nodes still have some budget. Subsequently, the remaining nodes are removed one by one, at a rate that depends on m, carefully designed for `OPT` to widen the gap with `Balance` as much as possible.
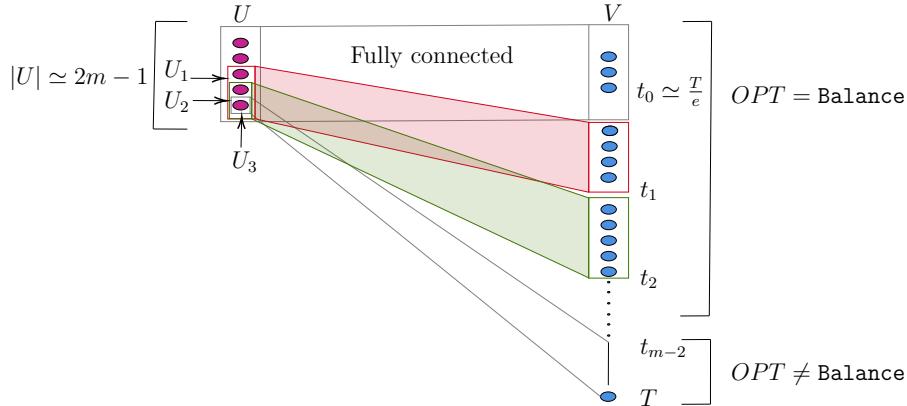
ALG vs OPT over time: As previously mentioned, up to time t_{m-2}, both `Balance` and `OPT` have the same performance. It is only between t_{m-2} and T that distinctions arise. Hence, the crucial step lies in determining the remaining budget of `Balance` at time t_{m-2} denoted $P_{t_{m-2}}$. To accomplish this, it is necessary to compute the values of t_i and then analyze how the remaining budget of `Balance` evolves over time.

Intuition behind the choice of "t_i": Since the main difference between `Balance` and `OPT` lies in the fact that `OPT` knows the eliminated nodes beforehand, one important quantity to track is the t_i which represents the time taken to deplete the budget of node u_i by consistently avoiding matches with u_i before t_{i-1} and then matching it at every time step between t_{i-1} and t_i (a strategy employed by `OPT`). t_i is determined by the following recurrence relationship:

$$t_{i+1} \approx b_0 - 1 + t_i + \frac{b_0 + t_i}{m - 1}.$$

(a) The graph used for the proof of theorem 16

(b) The graph $G^{\text{th.}2}$ used for the proof of theorem 17

by solving it we get,

$$t_i \approx \left(1 + \frac{1}{m-1}\right)^i (t_0 + mb_0 - m + 1) - mb_0 + m - 1.$$

Intuition about the value of remaining budget: The remaining budget at time t_i follows the following recurrence,

$$P_{t_i} \approx \Big(\underbrace{P_{t_{i-1}}}_{\text{the remaining budget at time } t_{i-1}} + \underbrace{(n-i)(t_i - t_{i-1})/m}_{\text{the refills received between time } t_i \text{ and } t_{i-1}}$$

$$- \underbrace{(t_i - t_{i-1})}_{\text{number of nodes matched}} \Big) \frac{n-i-1}{n-i}.$$

The expression $\frac{n-i-1}{n-i}$ represents the ratio of the number of nodes at time t_i to the number of nodes at time t_{i-1}.

Therefore, the crux of the proof lies in examining the dynamics and rate of evolution of P_{t_i} and handling the technicalities related to the approximations of the

floor and ceil functions involved in the construction of the different quantities of the problem.

$\square$

7.2.2.3 No algorithm can beat `Balance`

Theorem 18. *Assuming the initial budgets are* $b_{1,0} = b_{2,0} = \cdots = b_{n,0} = b_0 \geq 1$, *for* $m = o(\sqrt{T})$,

$$\sup_{\texttt{ALG}} \mathbb{E}\left[\texttt{CR}^{\mathrm{adv}}(\texttt{ALG}, \mathcal{G}_{T,m})\right] \leq \texttt{CR}^{\mathrm{adv}}(\texttt{Balance}, G^{th.2}) + o_T(1).$$

where the expectation is taken over the randomness from `ALG`.

The proof is provided in section 7.A.3.

Sketch of proof. The intuition is that keeping budgets equalized between the currently available U-nodes is the best choice an algorithm can make in the adversarial graph used for the proof of theorem 17. This is because the adversary removes U-nodes one after the other, beginning with those with the highest budget and never providing again a U-node already removed. $\square$

7.3 The stochastic framework

Another line of research in online matching has focused on the stochastic version of the problem. A classical framework is the known i.i.d. model, which assumes the existence of a probability distribution governing the types of arriving vertices, drawn independently and identically at each iteration. When this distribution is known, algorithms achieving significantly better competitive ratios than `Ranking` have been designed [77, 58, 26, 55]. The current best competitive ratio in this setting is approximately 0.711. Despite its versatility, the i.i.d. model remains somewhat idealized: algorithms tailored to this framework often fail to exploit additional structural information about the graph, and their guarantees—being worst-case over all input distributions—do not always reflect their average performance in practice. As highlighted in [23], in many real-world or average-case scenarios, simple greedy strategies match or even outperform state-of-the-art algorithms specifically designed for the i.i.d. setting. This has motivated the development of new stochastic input models that more accurately capture practical applications such as online advertising.

A related line of work investigates the performance of standard online algorithms on specific families of random graphs, thereby incorporating partial knowledge about the graph structure. One classical model is the Erdős–Rényi bipartite graph considered in [79], where each potential edge in $U \times V$ appears independently with

some probability. Of particular interest is the sparse regime, in which each vertex of U has an expected degree that does not grow with the size of V, corresponding to a connection probability of order c/n. Even analyzing the most basic `Greedy` algorithm in these settings turns out to be highly nontrivial and has led to several important insights [79, 22, 37].

Beyond Erdős–Rényi graphs, a more expressive stochastic model is the configuration model (CM), which allows the degree distributions of vertices to be prescribed. The asymptotic structure of optimal matchings on infinite CM graphs was characterized in the seminal work [20]. Later developments analyzed the asymptotic performance of simple online algorithms: the greedy matching size on bipartite configuration model graphs was studied in [8], while the greedy and degree-greedy algorithms on general (non-bipartite) CM graphs were investigated in [9]. In a related direction, the asymptotic size of the randomized greedy matching on regular graphs was established in [44].

7.3.1   Model

The online matching problem with refills of the budgets in the stochastic setting is studied in the following framework:

1. The random graph is a standard Erdős–Rényi model $G(n, T, p)$, i.e., a bipartite graph with n vertices on one side, T on the other side and each potential edge $(u, t) \in U \times V$ occurs independently with probability p.

2. The regime considered is the sparse one, in the sense that $p = \frac{a}{n}$ with $a > 0$. This setup is motivated by online advertising, where the number of users greatly exceeds the number of ad campaigns, and only a small subset of users are eligible to participate.

3. The sequence of refills $(\eta_{u,t})_{u \in U, t \in V}$ is a realization of independent Bernoulli random variable of parameter β/n, for some $\beta > 0$.

As emphasized previously, each node $u \in U$ is associated with a budget $b_{u,t} \in \mathbb{N}$. We add the additional assumption that the maximum budget per node is capped at some $K \in \mathbb{N}^*$ so that the budget dynamics are now expressed as follows,

$$b_{u,t} = \min(K, b_{u,t-1} - x_{u,t} + \eta_{u,t}), \qquad \text{with } b_{u,0} = b_0 \geq 1 \qquad (7.1)$$

The reasons behind capping the maximal budget to K are three-fold. First, in many applications in mind, the budget is capped (either by one, which corresponds to an idle/active state, or by a large number as in the online advertisement motivating example). Second, with the algorithm and the random graph considered, the budget will follow a negatively biased (and non-homogeneous) random walk, so that the maximal budget is sub-linear with arbitrarily high probability (hence this restriction is actually without loss of generality in the random model considered). Third, this

capping induces a finite number of quantities to track through time (namely the current proportion of vertices with this or that budget), which greatly simplifies the analysis.

As defined previously, an online *matching* on G generated by an algorithm $\mathtt{ALG}$ is a subset of edges, represented by a binary matrix $\mathbf{x} \in \{0, 1\}^{n \times T}$, satisfying Items 1 to 3.

The performance of an algorithm in the stochastic setting can be either measured by the size of the expected matching size it creates or by the ratio between expected matching sizes of $\mathtt{ALG}$ and $\mathtt{OPT}$. Formally, the different quantities we shall consider are

$$\mathtt{CR}^{\mathrm{sto}}(\mathtt{ALG}, \mathcal{D}) = \frac{\mathbb{E}_{G \sim \mathcal{D}}[\mathtt{ALG}(G)]}{\mathbb{E}_{G \sim \mathcal{D}}[\mathtt{OPT}(G)]}, \qquad \text{or the matching size} = \mathbb{E}_{G \sim \mathcal{D}}[\mathtt{ALG}(G)].$$

where we denote by $\mathcal{D}$ the distribution of graph considered and the refills, $\mathtt{ALG}(G) = \sum_{u \in U} \sum_{t=1}^{T} x_{u,t}$, and $\mathtt{OPT}(G) = \sum_{u \in U} \sum_{t=1}^{T} x_{u,t}^*$ are the sizes of the matching generated by $\mathtt{ALG}$ and $\mathtt{OPT}$ respectively. Although the dependence on T is implicit in $\mathtt{ALG}(G)$ and $\mathtt{OPT}(G)$, we explicitly write $\mathtt{ALG}(G, T)$ and $\mathtt{OPT}(G, T)$ to emphasize their evolution over time, which is essential for the analysis of the underlying dynamic process.

7.3.2 Main results

Our first main theorem, stated below, identifies the asymptotic size of the matching generated by $\mathtt{Greedy}$ on the bipartite Erdős-Rényi model with budget refills. The result shows that with high probability, the size of the matching generated by $\mathtt{Greedy}$ is close to the solution of a system of ordinary differential equations.

Theorem 19. *With probability* $1 - \mathcal{O}\left(n^{1/4} \exp(-a^3 n^{1/4})\right)$*, the matching size created by* $\mathtt{Greedy}$ *denoted by* $\mathtt{Greedy}(G, T)$ *satisfies,*

$$\mathtt{Greedy}(G, T) = nh(T/n) + \mathcal{O}(n^{3/4}).$$

and,

$$\frac{\mathbb{E}[\mathtt{Greedy}(G, T)]}{n} - h(T/n) \underset{n \to +\infty}{\to} 0.$$

where $h(\tau)$ *is solution of the following equation,*

$$\dot{h}(\tau) = 1 - e^{-a(1 - z_0(\tau))}, \qquad \frac{1}{n} \leq \tau \leq \frac{T}{n}.$$

and $z_0(\tau)$ satisfies the following system,

$$
\begin{cases}
\dot{z}_0(\tau) = -z_0(\tau)\beta + \frac{z_1(\tau)}{1-z_0(\tau)}(1 - e^{-a+az_0(\tau)}) & \text{for } k = 0, \\
\dot{z}_k(\tau) = (z_{k-1}(\tau) - z_k(\tau))\beta + (z_{k+1}(\tau) - z_k(\tau))\frac{1-e^{-a+az_0(\tau)}}{1-z_0(\tau)} & \text{for } 1 \le k \le K-1, \\
\dot{z}_k(\tau) = \beta\, z_{k-1}(\tau) - z_k(\tau)\frac{1-e^{-a(1-z_0(\tau))}}{1-z_0(\tau)} & \text{for } k = K, \\
\sum_{k=0}^{K} z_k(\tau) = 1.
\end{cases}
$$

$$(7.2)$$

Remark 2. *The system Equation (7.2) admits a unique solution, since it satisfies the Cauchy–Lipschitz conditions.*

Sketch of proof. For $0 \le k \le K$, $t \in [T]$, let $U_k(t) = \{u \in U : b_{u,t} = k\}$ be the set of nodes with budget equals to k at time t and $Y_k(t) = |U_k(t)|$ the total number of nodes with budget equals k in U. The expectation of the one-step change of the variable $\texttt{Greedy}(G,t)$ can be expressed as,

$$
\mathbb{E}\left[\texttt{Greedy}(G,t+1) - \texttt{Greedy}(G,t)|\texttt{Greedy}(G,t)\right] = 1 - \left(1 - \frac{a}{n}\right)^{\sum_{k\ge 1} Y_k(t)}
$$

$$
= 1 - \left(1 - \frac{a}{n}\right)^{n - Y_0(t)}.
$$

As the evolution of $\texttt{Greedy}(G,t)$ depends on Y_0, an analysis of the process $\mathbf{Y}(t) = (Y_k(t))_{k\ge 0}$ is necessary. The dynamics of this process is described by the following system:

$$
\begin{cases}
\mathbb{E}\left[\Delta_0(t)|\mathbf{Y}(t)\right] = -Y_0(t)\left[\frac{\beta}{n}(1 - p\,\Sigma(t))\right] + Y_1(t)(1 - \frac{\beta}{n})p\,\Sigma(t), \\
\mathbb{E}\left[\Delta_1(t)|\mathbf{Y}(t)\right] = -Y_1(t)\left[\frac{\beta}{n}(1 - p\,\Sigma(t)) + (1 - \frac{\beta}{n})p\,\Sigma(t)\right] + Y_2(t)(1 - \frac{\beta}{n})p\,\Sigma(t) \\
Y_0(t)\frac{\beta}{n}, \\
\mathbb{E}\left[\Delta_k(t)|\mathbf{Y}(t)\right] = \frac{\beta}{n}(1 - p\,\Sigma(t))\left[Y_{k-1}(t) - Y_k(t)\right] + \left[Y_{k+1}(t) - Y_k(t)\right](1 - \frac{\beta}{n})p\,\Sigma(t).
\end{cases}
$$

where $\forall k \ge 0$, $\Delta_k(t) = Y_k(t+1) - Y_k(t)$, and $\Sigma(t) = \frac{1}{p(n-Y_0(t))}(1 - (1-p)^{(n-Y_0(t))})$.

After establishing the evolution of these processes, the main idea behind the proof of theorem 19 (postponed to section 7.B.1) is to show that $\texttt{Greedy}(G,T)$ is closely related to the solution of some ordinary differential equations (this is sometimes called the differential equation method [98, 95, 38] or stochastic approximations [87]). $\qquad\square$

Building on the fact that, with high probability, $\texttt{Greedy}(G,T)$ is close to $nh(T/n)$, a function depending on $z_0(T/n)$, the solution of eq. (7.2), the objective is to solve this system to obtain an exact approximation of the matching size created by $\texttt{Greedy}$ on the Erdős–Rényi model. However, finding a closed-form solution of the system of differential equations eq. (7.2) is quite challenging. To address this complexity, one approach is to explore the system's stationary solution and examine its stability. This means determining whether the solution to eq. (7.2) converges to this

stationary state and then showing that $\mathtt{Greedy}(G,T)$ converges towards a function depending on the stationary solution of eq. (7.2).

More precisely, corollary 2 shows that for $K \geq 1$, $\mathtt{Greedy}(G,T)$ converges with high probability to nh^*, a function of z_0^*, the stationary solution of eq. (7.2). Additionally, as $n \to +\infty$, $\frac{\mathbb{E}(\mathtt{Greedy}(G,T))}{n}$ converges to $h^*(\psi)$. Furthermore, corollary 3 demonstrates, at a specified rate, the convergence of $\mathtt{Greedy}(G,T)$ to $nh^*(T/n)$ with high probability, and also, for $n \to +\infty$, $\frac{\mathbb{E}(\mathtt{Greedy}(G,T))}{n}$ converges to $h^*(T/n)$. The key distinction between these results lies in the nature of the convergence of $z_0(t)$ to z_0^*: in corollary 2, $z_0(t)$ asymptotically converges to z_0^*, whereas in corollary 3, the convergence is exponential.

Corollary 2. *For $K \geq 1$, with probability at least $1 - 2\exp(-a^2 n^{\frac{3}{2}}/8T)$,*

$$|\mathtt{Greedy}(G,T) - nh^*(T/n)| \leq o(T),$$

and,

$$\frac{\mathbb{E}[\mathtt{Greedy}(G,T)]}{n} - h^*(T/n) \underset{n \to +\infty}{\to} 0,$$

with $h^*(x) = \int_{1/n}^{x}(1 - e^{-a(1-z_0^*)})\mathrm{d}\tau = \left(x - \frac{1}{n}\right)\left(1 - e^{-a(1-z_0^*)}\right)$, and z_0^* is the unique solution of $\sum_{k=0}^{K} z_0^* \left(\frac{\beta}{g(z_0^*)}\right)^k = 1$ with $g(z_0^*) = \frac{1 - e^{-a(1-z_0^*)}}{1 - z_0^*}$.

Section 7.B.2 contains the detailed proof.

Sketch of proof.   The first step is to compute $\bar{S}_{z_0^*}$, the unique stationary solution of eq. (7.2). Then, to demonstrate that $\bar{S}_{z_0^*}$ is an asymptotically stable stationary solution, we rely on matrix perturbation theory. Once stability is established, we further prove that the matching size converges to a function that depends on $\bar{S}_{z_0^*}$. □

Corollary 3. *For $K = 1$, with probability at least $1 - 2\exp(-a^2 n^{\frac{3}{2}}/8T)$,*

$$\left|\mathbb{E}[\mathtt{Greedy}(G,T)] - T(1 - e^{-a(1-z_0^*)})\right| \leq c\frac{T}{(\log(T))^{3/4}} = o(T).$$

where $z_0^* = \frac{1}{\beta} - \frac{1}{a}W\left(\frac{a}{\beta}e^{-a\left(1-\frac{1}{\beta}\right)}\right)$, with $W(\cdot)$ the Lambert function, and c is some universal constant.

The proof is postponed to section 7.B.3.

Sketch of proof.   For $K = 1$, eq. (7.2) is reduced to a system of two equations. Firstly, we compute $S_{z_0^*}^1$, the stationary solution of the reduced system. Then, we prove that $S_{z_0^*}^1$ is an exponentially stable stationary solution using the perturbation method. Once the exponential stability is established, we further get that the matching size converges to a function depending only on $S_{z_0^*}^1$. □

The final main result of this section is summarized as follows. First, we establish a lower bound on $\mathrm{CR}^{\mathrm{sto}}$, which depends on $(z_0^*, \ldots, z_K^*)$ the stationary solution of eq. (7.2). This lower bound is derived through an exact calculation of the matching size achieved by the `Greedy` algorithm and an upper bound on the matching size generated by `OPT`. Subsequently, we demonstrate that the competitive ratio converges to 1 as T, K and n grows significantly.

Proposition 3. *For $T, K, n, b_0, \beta \in \mathbb{N}^*$,*

$$\mathrm{CR}^{\mathrm{sto}}(\texttt{Greedy}, \mathcal{D}) \geq \frac{Tg(z_0^*)(1 - z_0^*) + nb_0 - n\left(\frac{\beta}{g(z_0^*) - \beta} - \frac{(K+1)\beta^{K+1}}{g(z_0^*)^{K+1} - \beta^{K+1}}\right)}{nb_0 + \beta T}$$
$$+ \mathcal{O}_{K,\beta}(T^{-1/4}).$$

where $\sum_{k=0}^{K} z_0^* \left(\frac{\beta}{g(z_0^*)}\right)^k = 1$ with $g(z_0^*) = \frac{1 - e^{-a(1-z_0^*)}}{1 - z_0^*}$ as defined in corollary 2.

The proof is presented in section 7.B.4.

Sketch of proof. Initially, we express $\texttt{Greedy}(G, T)$ as a function of $T, z_0(t), \beta, a$. Then, we use an upper bound on $\texttt{OPT}(G, T)$, which is not very tight as it only takes into account the initial budget and the refills. Subsequently, we approximate $\texttt{Greedy}(G, T)$ by a function that depends on the stationary solution $\bar{S}_{z_0^*}$. It is noteworthy that in this context, the matching size $\texttt{Greedy}(G, T)$ aligns with that of theorem 19 through the integration of the system eq. (7.2), up to negligible terms. □

From proposition 3, the next result shows that when K, T, n goes to infinity, the competitive ratio approaches 1.

Theorem 20. *For any $\alpha, \beta > 0$, the competitive ratio tends to 1, as T, K, n approach infinity, as*

$$\lim_{K,n \to +\infty} \lim_{T \to +\infty} \mathrm{CR}^{\mathrm{sto}}(\texttt{Greedy}, \mathcal{D}) = 1.$$

The proof is in section 7.B.5.

Sketch of proof. The proof relies on calculating z_0^* as K approaches infinity. Subsequently, as T approaches infinity, the limit of $\mathrm{CR}^{\mathrm{sto}}(\texttt{Greedy}, \mathcal{D})$ is shown to be $g(z_0^*)(1 - z_0^*)/\beta$. Finally, as K tends towards infinity, the limit converges to 1, with the assurance that this convergence happens with high probability as n tends to infinity. □

Conclusion

We study online matching on a bipartite graph $G = (U, V, E)$ with dynamic budget refills on nodes in U. Two settings are considered: *stochastic*, where refills follow a

Bernoulli process in Erdős–Rényi graphs, and *adversarial*, with deterministic graphs and refill patterns. In the stochastic case, `Greedy` performs well under periodic refills, with its competitive ratio `CR` approaching 1. In the adversarial setting, infrequent refills have little impact on `Balance`, aligning its performance with b-matching. Frequent refills, however, yield better upper bounds on `CR`. Establishing lower bounds is challenging due to dynamic budgets. A naive lower bound of $1 - \frac{1}{e}$ is obtained by duplicating nodes in a variant of `Ranking`, leaving a gap with the upper bound $1 - \frac{1-\alpha}{e^{1-\alpha}}$ when $m = o(\sqrt{T})$.

7

Appendix 7

7.A   Adversarial Case

7.A.1   Proof of theorem 16

Theorem 16. *Assuming the initial budgets are $b_{1,0} = b_{2,0} = \cdots = b_{n,0} = b_0 \geq 1$. If $m = \omega(\sqrt{T})$ and $b_0(b_0 + 1)^{b_0} \leq m$, then,*

$$\sup_{\texttt{ALG}:deterministic} \mathrm{CR}^{\mathrm{adv}}(\texttt{ALG}, \mathcal{G}_{T,m}) \leq 1 - \frac{1}{\left(1 + \frac{1}{b_0}\right)^{b_0}} + o_T(1).$$

The bound is reached for the graph defined in the proof.

Proof. Let $b_0, m, T \in \mathbb{N}^*$ such that $m \geq kb_0$ where $k \triangleq (1 + b_0)^{b_0}$ and $m \leq T$. The bipartite graph of size (k, kb_0) used in [62, Sec. 2, Thm. 5] is denoted (U_0, V_0, E_0). To put the emphasis on which set of nodes the edges are defined on, E_0 will actually be denoted $E_0(U_0, V_0)$ as the structure of edges will be used on different subsets of nodes of the final graph. The graph $G = (U, [T], E)$ with $U = \{u_1, \ldots, u_n\}$ of size $n \in \mathbb{N}^*$ is built as follows. Intuitively, the first period of length m is implementing copies of E_0 on disjoint nodes, then one remaining node in U is the only neighbor of all following time steps. Denoting $j = \left\lfloor \frac{m}{kb_0} \right\rfloor$,

$$E = \left(\bigcup_{i=1}^{j} E_0(U_i, V_i) \right) \cup \left(\{\tilde{u}\} \times [\![jkb_0 + 1, T]\!] \right) \tag{7.3}$$

where $U_i = \{u_l : l \in [\![(i-1)k + 1, ik]\!]\}$, $V_i = [\![(i-1)mkb_0 + 1, ikb_0]\!]$ and $\tilde{u}$ is chosen to be a node of U_1 which has been depleted of its initial budget during V_1 (there is at least one).

For each $i \in [j]$, on each subset V_i of time steps, as per [62, Proof of Thm. 5], **ALG** matches at most $b_0(b_0 + 1)^{b_0} - b_0^{b_0+1}$ edges, while **OPT** manages to match $b_0(b_0 + 1)^{b_0}$ edges. After time jkb_0, both **ALG** and **OPT** match the same number of edges γ_T which is at most the sum of refills obtained by $\tilde{u}$ – i.e. $\left\lfloor \frac{T}{m} \right\rfloor$ – (its initial budget is used

during period V_1). In the end,

$$\text{CR}^{\text{adv}}(\text{ALG}, \mathcal{G}_{T,m}) \leq \frac{j(kb_0 - b_0^{b_0+1}) + \gamma_T}{jkb_0 + \gamma_T} \tag{7.4}$$

$$\leq \frac{j(kb_0 - b_0^{b_0+1})}{jkb_0} + o_T(1) \quad \text{as } \gamma_T = o(\sqrt{T}) \text{ and } jkb_0 = \omega(\sqrt{T}) \tag{7.5}$$

$$= 1 - \frac{1}{\left(1 + \frac{1}{b_0}\right)^{b_0}} + o_T(1) \qquad \text{def. of } k = (1 + b_0)^{b_0} \tag{7.6}$$

Similarly, it is straightforward to show that `Balance` achieves the lower bound of the b-matching problem on each of the duplicates of $E_0(U_i, V_i)$, as these sub-graphs are disjoint.
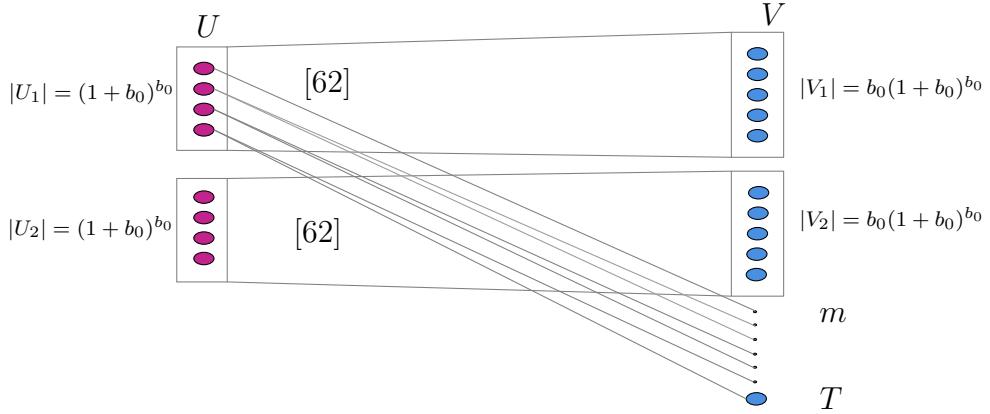
Figure 7.2: The graph used for the proof of theorem 16

□

7.A.2 Proof of theorem 17

Theorem 17. *Assuming the initial budgets are $b_{1,0} = b_{2,0} = \cdots = b_{n,0} = b_0 \geq 1$. For $m = o(\sqrt{T})$ and $mb_0 = o(T)$, then,*

$$\text{CR}^{\text{adv}}(\text{Balance}, \mathcal{G}_{T,m}) \leq \underbrace{1 - \frac{(1-\alpha)}{e^{(1-\alpha)}}}_{\simeq 0.73325...} + o_{m,T}(1).$$

where α is defined by $\frac{1}{2} = \int_0^\alpha \frac{xe^x}{1-x}\mathrm{d}x$. The bound is reached for the graph defined in the proof.

We provide a slightly more detailed result here.

Theorem 21. *Assuming the initial budgets are* $b_{1,0} = b_{2,0} = \cdots = b_{n,0} = b_0 \geq 1$. *For* $m = o(\sqrt{T})$ *and* $mb_0 = o(T)$, *then,*

$$
\mathtt{CR}^{\mathrm{adv}}(\mathtt{Balance}, \mathcal{G}_{T,m}) \leq 1 - \frac{mb_0 + t_0}{e(mb_0 + 2t_0)} - \frac{1}{e} \int_0^\alpha \frac{x^2 e^x}{1 - x} \mathrm{d}x
$$
$$
+ \frac{mb_0}{t_0} \left(1 - \frac{1}{e} + \frac{1}{e} \int_0^\alpha \frac{x(\alpha - x)e^x}{1 - x} \mathrm{d}x \right) + o_{m,T}(1).
$$

where α *is defined as follows* $\int_0^\alpha \frac{xe^x}{1-x} \mathrm{d}x = 1 - \frac{t_0}{mb_0 + 2t_0}$. *The upper bound is reached for the graph defined in the proof.*

The proof is organized as follows:

1. Definition of the adversarial graph.

2. Decomposition of $\mathtt{Balance}(G^{\mathrm{th.2}})$.

3. Several lemmas to treat each term of the decomposition.

Definition of the adversarial graph for $\mathtt{Balance}$. For $b_0, m, T \in \mathbb{N}^*$ such that $m \leq T$, the number of U-nodes is set to $n = m - 1 + \max \left(\left\lceil \frac{t_0}{b_0 + \lfloor \frac{t_0}{m} \rfloor} \right\rceil, \left\lceil \frac{m \lfloor \frac{t_0}{m} \rfloor}{b_0 + \lfloor \frac{t_0}{m} \rfloor - 1} \right\rceil \right)$ (Note that when $b_0 \ll \frac{t_0}{m}$, then $n \simeq 2m - 1$). The graph $G^{\mathrm{th.2}} = (U, V, E)$ is defined as follows,

$$
\begin{cases}
U = [n], \\
V = [T], \\
E = (U \times [t_0]) \cup (U_1 \times [\![t_0 + 1, t_1]\!]) \cup \cdots \cup (U_{m-1} \times [\![t_{m-2} + 1, t_{m-1}]\!]) \\
\cup (U_{m-1} \times [\![t_{m-1} + 1, T]\!]).
\end{cases}
$$

where,

- U_1 is the subset of the $m - 1$ node with the lowest budget at time t_0, i.e.

$$
U_1 \overset{\mathrm{unif}}{\sim} \{ A \subseteq U : |A| = m - 1, \forall u \in A, u' \in U \setminus A, b_{u,t_0} \leq b_{u',t_0} \}.
$$

- for any $i > 1$, U_i is built be removing the node with the lowest budget at time t_{i-1} from U_{i-1} – i.e. $U_i = U_{i-1} \setminus \{u_i\}$ where

$$
u_i \overset{\mathrm{unif}}{\sim} \underset{u \in U_{i-1}}{\arg\min} \, b_{u, t_{i-1}}.
$$

- for any $i \geq 1$, $t_i = \inf\{t > t_{i-1} : b_0 + \lfloor \frac{t}{m} \rfloor = (t - t_{i-1})\}$. *Intuition:* t_i *is the time it takes to take the budget of* u_i *to 0 by never matching* u_i *before* t_{i-1} *and matching it at every time step between* t_{i-1} *and* t_i *(which is what* $\mathtt{OPT}$ *does).*

- t_0 is chosen such that $T - t_{m-1} = o(T)$ (it is possible as proven in lemma 1)
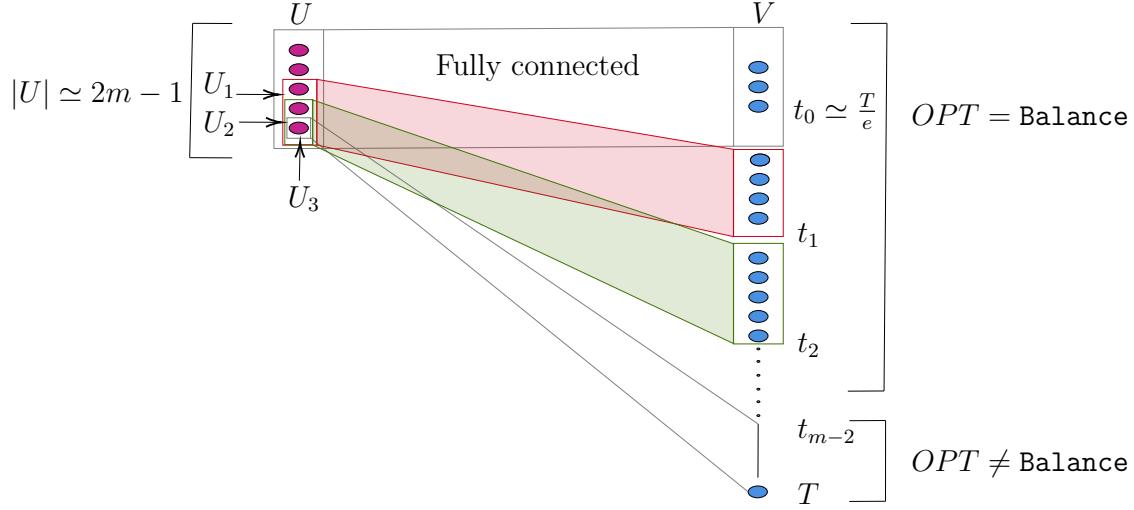
Figure 7.3: The graph $G^{\text{th.2}}$ used for the proof of theorem 17

Proof. The objective is to compute the performance of `Balance` and `OPT` on the graph $G^{\text{Th. 2}}$ defined above to obtain a bound on the `CR`.

Performance of `OPT`. Before time t_0, `OPT` can use nodes from $U \setminus U_1$ to match a each time step: $|U \setminus U_1| = \max\left(\left\lceil \frac{t_0}{b_0 + \left\lfloor \frac{t_0}{m} \right\rfloor} \right\rceil, \left\lceil \frac{m \left\lfloor \frac{t_0}{m} \right\rfloor}{b_0 + \left\lfloor \frac{t_0}{m} \right\rfloor - 1} \right\rceil \right)$, which ensures that the total budget of nodes in $U \setminus U_1$ over the period $[t_0]$ is at least t_0 (accounting for the last refill that cannot necessarily be fully used). As a remark, if $b_0 = 1$, this simplifies to $|U \setminus U_1| = m$: with a refill every m timesteps, m nodes suffice to match at every time step. Thus, at time t_0, `OPT` never matched any node from U_1. Then, by induction and definition of t_i, $\text{OPT}(G^{\text{th.2}}) = t_{m-1}$.

Performance of `ALG`.

$$\text{Balance}(G^{\text{th.2}}) = \underbrace{\sum_{t=1}^{t_0} \sum_{u \in U} x_{u,t}}_{\triangleq A_0} + \sum_{i=1}^{m-1} \underbrace{\sum_{t=t_{i-1}+1}^{t_i} \sum_{u \in U_i} x_{u,t}}_{\triangleq A_i} \tag{7.7}$$

$$= A_0 + \sum_{i=1}^{m-1} A_i \tag{7.8}$$

$$= A_0 + \sum_{i=1}^{m-1} B_{t_{i-1}}^{(i)} - B_{t_i}^{(i)} + (m-i)\left(\left\lfloor \frac{t_i}{m} \right\rfloor - \left\lfloor \frac{t_{i-1}}{m} \right\rfloor \right) \quad \text{(by induction)} \tag{7.9}$$

$$\tag{7.10}$$

where $B_t^{(i)} = \sum_{u \in U_i} b_{u,t}$,

$$= A_0 + \sum_{i=1}^{m-1} B_{t_{i-1}}^{(i)} - B_{t_i}^{(i)} + \sum_{i=1}^{m-1} (m-i) \left\lfloor \frac{t_i}{m} \right\rfloor - \sum_{i=0}^{m-2} (m-i-1) \left\lfloor \frac{t_i}{m} \right\rfloor \tag{7.11}$$

$$= A_0 + \left\lfloor \frac{t_{m-1}}{m} \right\rfloor - (m-1) \left\lfloor \frac{t_0}{m} \right\rfloor + \sum_{i=1}^{m-1} B_{t_{i-1}}^{(i)} - B_{t_i}^{(i)} + \sum_{i=1}^{m-2} \left\lfloor \frac{t_i}{m} \right\rfloor \tag{7.12}$$

$$= A_0 - (m-1) \left\lfloor \frac{t_0}{m} \right\rfloor + \sum_{i=1}^{m-1} B_{t_{i-1}}^{(i)} - B_{t_i}^{(i)} + \sum_{i=1}^{m-1} \left\lfloor \frac{t_i}{m} \right\rfloor \tag{7.13}$$

$$= A_0 - (m-1) \left\lfloor \frac{t_0}{m} \right\rfloor + \sum_{i=1}^{m-1} B_{t_{i-1}}^{(i)} - B_{t_{i-1}}^{(i-1)} + B_{t_{i-1}}^{(i-1)} - B_{t_i}^{(i)} + \sum_{i=1}^{m-1} \left\lfloor \frac{t_i}{m} \right\rfloor \tag{7.14}$$

$$= A_0 - (m-1) \left\lfloor \frac{t_0}{m} \right\rfloor + B_{t_0}^{(0)} - B_{t_{m-1}}^{(m-1)} + \sum_{i=1}^{m-1} B_{t_{i-1}}^{(i)} - B_{t_{i-1}}^{(i-1)} + \sum_{i=1}^{m-1} \left\lfloor \frac{t_i}{m} \right\rfloor \tag{7.15}$$

$$= A_0 - (m-1) \left\lfloor \frac{t_0}{m} \right\rfloor + B_{t_0}^{(0)} - B_{t_{m-1}}^{(m-1)} + \sum_{i=0}^{m-2} B_{t_i}^{(i+1)} - B_{t_i}^{(i)} + \sum_{i=1}^{m-1} \left\lfloor \frac{t_i}{m} \right\rfloor \tag{7.16}$$

$$= A_0 - (m-1) \left\lfloor \frac{t_0}{m} \right\rfloor + B_{t_0}^{(1)} - B_{t_{m-1}}^{(m-1)} + \sum_{i=1}^{m-2} B_{t_i}^{(i+1)} - B_{t_i}^{(i)} + \sum_{i=1}^{m-1} \left\lfloor \frac{t_i}{m} \right\rfloor \tag{7.17}$$

$$= \underbrace{A_0 - (m-1) \left\lfloor \frac{t_0}{m} \right\rfloor}_{\triangleq Q_1} + \underbrace{B_{t_0}^{(1)}}_{\triangleq Q_2} - \underbrace{B_{t_{m-1}}^{(m-1)}}_{\triangleq Q_3} - \underbrace{\sum_{i=1}^{m-2} \left\lceil \frac{B_{t_i}^{(i)}}{m-i} \right\rceil}_{\triangleq Q_4} + \underbrace{\sum_{i=1}^{m-1} \left\lfloor \frac{t_i}{m} \right\rfloor}_{\triangleq Q_5}. \tag{7.18}$$

where the last equality comes from $B_{t_i}^{(i+1)} = B_{t_i}^{(i)} - \left\lceil \frac{B_{t_i}^{(i)}}{m-i} \right\rceil$ which in turn comes from the definition of U_{i+1} (the adversary removes the node with most budget) combined with lemma 2 (`Balance` equalizes budget among available nodes).

The following lemma proves that $T = t_{m-1} + o(T)$,

Lemma 1. *For $t_0 \leq \frac{T}{e}$, $T = t_{m-1} + o(T)$.*

Proof. According to lemma 6,

$$\tilde{t}_{m-1} = \left(1 + \frac{1}{m-1}\right)^{m-1}(t_0 + mb_0 - m + 1) - mb_0 + m - 1.$$

Putting everything together gives,

$$
\begin{aligned}
T - t_{m-1} &= T - t_{m-1} - \tilde{t}_{m-1} + \tilde{t}_{m-1} \\
&\leq T + \left(1 + \frac{1}{m-1}\right)^{m-1}(t_0 + mb_0 - m + 1) - mb_0 + m - 1 \\
&\leq T + e(t_0 + mb_0) - mb_0.
\end{aligned}
$$

choosing t_0 such that $t_0 = \lfloor T/e \rfloor$ along with the fact that $m = o(\sqrt{T})$, implies that $T - t_{m-1} = o(T)$. □

Computation of the CR.

$$
\begin{aligned}
&\mathrm{CR}^{\mathrm{adv}}(\texttt{Balance}, \mathcal{G}_{T,m}) \\
&= \frac{Q_1 + Q_2 - Q_3 - Q_4 + Q_5}{t_{m-1} + o(T)} \\
&= \frac{\mathcal{O}\left(\frac{t_0}{m}\right) + Q_2 - Q_3 - Q_4 + Q_5}{t_{m-1} + o(T)} \qquad\qquad\qquad \text{as } A_0 = t_0 \\
&= \frac{\mathcal{O}\left(\frac{t_0}{m}\right) + (m-1)\left(b_0 + \lfloor\frac{t_0}{m}\rfloor - \lfloor\frac{t_0}{n}\rfloor\right) + \mathcal{O}(m) - Q_3 - Q_4 + Q_5}{t_{m-1} + o(T)} \qquad \text{lemma 4} \\
&= \frac{\mathcal{O}\left(\frac{t_0}{m}\right) + (m-1)\left(b_0 + \lfloor\frac{t_0}{m}\rfloor - \lfloor\frac{t_0}{n}\rfloor\right) + \mathcal{O}(m) - Q_4 + Q_5}{t_{m-1} + o(T)} \\
&= \frac{1}{t_{m-1} + o(T)}\left(\mathcal{O}\left(\frac{t_0}{m}\right) + (m-1)\left(b_0 + \left\lfloor\frac{t_0}{m}\right\rfloor - \left\lfloor\frac{t_0}{n}\right\rfloor\right) - \lfloor\alpha^* m\rfloor\left\lceil\frac{B_{t_1}^{(1)}}{m}\right\rceil \right. \\
&\qquad\qquad \left. + \bar{t}_0 \int_{\frac{1}{m}}^{\alpha^*} g_m(x)\mathrm{d}x + \frac{g_m(\alpha^*) - g_m(1/m)}{m} + \mathcal{O}(m^2) + Q_5\right) \qquad \text{lemma 13} \\
&= \frac{1}{t_{m-1} + o(T)}\left(\mathcal{O}\left(\frac{t_0}{m}\right) + (m-1)\left(b_0 + \left\lfloor\frac{t_0}{m}\right\rfloor - \left\lfloor\frac{t_0}{n}\right\rfloor\right) - \lfloor\alpha^* m\rfloor\left\lceil\frac{B_{t_1}^{(1)}}{m}\right\rceil \right. \\
&\qquad\qquad + \bar{t}_0 \int_{\frac{1}{m}}^{\alpha^*} g_m(x)\mathrm{d}x + \frac{g_m(\alpha^*) - g_m(1/m)}{m} + \mathcal{O}(m^2) \\
&\qquad\qquad \left. + \left(\left(\left(1 + \frac{1}{m-1}\right)^{m-1} - 1 + \frac{1}{m}\right)t_0 + \mathfrak{B}(m, b_0)\right)\right). \qquad \text{lemma 9}
\end{aligned}
$$

where $\alpha^* \in (1/m, 1)$ the solution of

$$\frac{\bar{t}_0}{m} \int_{\frac{1}{m}}^{\alpha^*} \frac{z}{1-z} e^z \mathrm{d}z - m\alpha^* - Y_1 = 0,$$

and

$$g_m(x) = \frac{x(\alpha^* - x)}{1 - x} \left(1 + \frac{1}{m-1}\right)^{mx}.$$

thus $\mathfrak{B}(m, b_0) \leq (e - 2)mb_0 + b_0$. $\qquad\square$

7.A.2.1 Proof of lemma 2

The following lemma states that U-nodes that were available exactly at the same time steps in the past should have the same budget within one unit when the algorithm is `Balance`.

Lemma 2. *Let $W \subseteq U$ such that $\forall s \leq t \in V,\ (\exists u \in W, (u, s) \in E) \Rightarrow (\forall u \in W, (u, s) \in E)$. For the algorithm* `Balance`,

$$\exists \beta_t \in \mathbb{N}, \forall u \in W, \exists z_{u,t} \in \{0, 1\}, \ s.t. \ b_{u,t} = \beta_t + z_{u,t} \quad and \quad \sum_{u' \in W} z_{u',t} < |W|$$

Proof. We first focus on the first part of the result. Let $t \in \mathbb{N}^*$ and $W \subseteq U$ such that $\forall s \leq t \in V,\ (\exists u \in W, (u, s) \in E) \Rightarrow (\forall u \in W, (u, s) \in E)$. We need to prove that using the `Balance` algorithm implies that the budgets of the nodes at time t differ only by one. We will prove it by recursion using the following hypothesis,

$$K(i) : \exists \beta_i \in \mathbb{N}, \forall u \in W, b_{u,i} = \beta_i + z_{u,i} \quad \text{with} \quad z_{u,i} \in \{0, 1\} \quad \text{and} \quad \sum_{u' \in W} z_{u',i} < |W|.$$

By assumption, $\forall u \in W\ b_{u,0} = b_0$, which means that $K(0)$ holds.

At time i, `Balance` chooses $u_i \in \arg\max_{u \in U:(u,i) \in E} b_{u,i}$. If $u_i \notin W$, the result is direct from $K(i-1)$ with $\beta_i = \beta_{i-1} + \mathbb{1}_{\{i \bmod m = 0\}}$. Otherwise, there are two cases when $u_i \in W$.

Case $\forall u \in W, b_{u,i-1} = \beta_{i-1}$. Then, by choosing $\beta_i = \beta_{i-1} - 1 + \mathbb{1}_{\{i \bmod m=0\}}$, we have $b_{u_i,i} = \beta_i$ and for any $u' \in W \setminus \{u_i\}, b_{u',i} = b_{u,i-1} + \mathbb{1}_{\{i \bmod m=0\}}$.

Case $\exists u, u' \in W, b_{u,i-1} \neq b_{u',i-1}$. Then, by choosing $\beta_i = \beta_{i-1} + \mathbb{1}_{\{i \bmod m=0\}}$, we have $b_{u_i,i} = \beta_i$ and $\forall u \in W \setminus \{u_i\}, b_{u,i} = b_{u,i-1} + \mathbb{1}_{\{i \bmod m=0\}}$.

In both cases, $K(i)$ holds.

$\qquad\square$

7.A.2.2 Proof of lemma 4

During the phase i, between t_{i-1} and t_i, the graph is fully-connected to U_i. Thus, $\sum_{u \in U_i} b_{u,t}$ follows the following dynamic Z_t.

Given $k, m, t, j \in \mathbb{N}^*$, the dynamic of interest is

$$Z_t = Z_{t-1} - \mathbb{1}_{\{Z_{t-1} \geq 1\}} + k\mathbb{1}_{\{t \bmod m = j\}}.$$

where $k = |U_i|$, and j accounts for the fact that a phase begins at a time t_{i-1} that is not necessarily a multiple of m.

Lemma 3. *For $k, m, t, Z_0, j \in \mathbb{N}^*$,*

$$Z_t = \begin{cases} (Z_0 - t)_+ + k\mathbb{1}_{\{t=j\}} & \text{if } t \leq j, \\ g(Z_j, k, t-j, m) & \text{if } j < t \leq j + t^*, \\ f(k, m, t-j, \tilde{t}) & \text{if } j + \tilde{t} \leq t. \end{cases}$$

$$t^* = \begin{cases} Z_j + k \left\lceil \frac{Z_j+1-m}{m-k} \right\rceil & \text{if } m > k, \\ Z_j & \text{if } m \leq k \text{ and } Z_j < m, \\ +\infty & \text{otherwise.} \end{cases} \quad \text{and } \tilde{t} = m \left\lceil \frac{t^*}{m} \right\rceil,$$

and $f(k, m, t, \tilde{t}) = \left(\mathbb{1}_{\{k<m\}}(k - (t \bmod m))_+ + \mathbb{1}_{\{k \geq m\}} \left(k \left(1 + \left\lfloor \frac{t-\tilde{t}}{m} \right\rfloor \right) - (t - \tilde{t}) \right) \right)$
and $g(Z_j, k, t, m) = \left(Z_j + k \left\lfloor \frac{t}{m} \right\rfloor - t \right).$

Proof. **First, the case when j = 0.**
For $k, m, t, Z_0 \in \mathbb{N}^*$, t^* is defined to be the first time at which Z_t reaches 0.

$$t^* = \min_{t \in \mathbb{N}^*} t \quad \text{s.t. } Z_t = 0.$$

which value is given by lemma 5.

For any $t \leq t^*$, $\mathbb{1}_{\{[Z_{t-1}>0]\}} = 1$, thus, by recursion,

$$Z_t = Z_0 - t + k \sum_{t'=1}^{t} \mathbb{1}_{\{[t' \bmod m = 0]\}}$$

$$= Z_0 - t + k \left\lfloor \frac{t}{m} \right\rfloor.$$

For any $t^* \leq t < \tilde{t}$, $t \bmod m \neq 0$ and thus $Z_t = 0$ (by recursion starting at $Z_{t^*} = 0$).

For any $t > \tilde{t}$, the analysis is split between the case $k \geq m$ and $k < m$. In both cases $Z_{\tilde{t}} = k$ and we denote $t = \tilde{t} + \Delta t$.

First, if $k \geq m$, similarly as before, we get $Z_{\tilde{t}+\Delta t} = (Z_{\tilde{t}} - \Delta t + k \lfloor \frac{\Delta t}{m} \rfloor)$: for $k \geq m$, it is always true that $Z_{\tilde{t}+\Delta t-1} > 0$ which gives the result by recursion.

Second, if $k < m$, the result is proved by recursion.

Recursion hypothesis – $H(t) = Z_{\tilde{t}+\Delta t} = (k - (t \bmod m))_+$ and by definition of $\tilde{t}$, $H(\tilde{t})$ holds as $Z_{\tilde{t}} = k$.

$$
\begin{aligned}
Z_{t+1} &= Z_t - \mathbb{1}_{\{Z_t > 0\}} + k \mathbb{1}_{\{t+1 \bmod m = 0\}} \\
&= (k - (\tilde{t} + \Delta t) \bmod m)_+ - \mathbb{1}_{\{(k - (\tilde{t}+\Delta t) \bmod m))_+ > 0\}} + k \mathbb{1}_{\{(\tilde{t}+\Delta t+1) \bmod m = 0\}} \\
&= (k - \Delta t \bmod m)_+ - \mathbb{1}_{\{(k - \Delta t \bmod m) > 0\}} + k \mathbb{1}_{\{(\Delta t+1) \bmod m = 0\}}.
\end{aligned}
$$

- **If** $(\Delta t + 1) \bmod m = 0$**,** we necessarily have $Z_{\tilde{t}+\Delta t} = 0$ (as $k \leq m - 1$). Thus $Z_{\tilde{t}+\Delta t+1} = k = (k - (\tilde{t} + \Delta t + 1) \bmod m)_+$.

- **If** $(\Delta t + 1) \bmod m \neq 0$ **and** $Z_{\tilde{t}+\Delta t} > 0$**,** we have $Z_{\tilde{t}+\Delta t+1} = Z_{\tilde{t}+\Delta t} - 1$ which gives the result,

- **If** $(\Delta t + 1) \bmod m \neq 0$ **and** $Z_{\tilde{t}+\Delta t} = 0$**,** we have $Z_{\tilde{t}+\Delta t+1} = Z_{\tilde{t}+\Delta t}$ which gives the result.

Second, the general case when $0 \leq j < m$.
Let $\tilde{Z}_t = Z_{t+j}$, $t_j^* = \min_{t \in \mathbb{N}^*} t$ s.t. $\tilde{Z}_t = 0$ and $\tilde{t}_j = m \lfloor \frac{t_j^*}{m} \rfloor$. Using the result proved above for $j = 0$ gives for any $t > j$

$$
\begin{aligned}
\tilde{Z}_{t-j} &= g(\tilde{Z}_0, k, t - j, m) \mathbb{1}_{\{t-j \leq t_j^*\}} + \mathbb{1}_{\{t-j \geq \tilde{t}_j\}} f(k, m, t - j, \tilde{t}) \\
&\Leftrightarrow Z_t = g(Z_j, k, t - j, m) \mathbb{1}_{\{t-j \leq t^*\}} + \mathbb{1}_{\{t-j \geq \tilde{t}\}} f(k, m, t - j, \tilde{t}),
\end{aligned}
$$

and for any $t \leq j$,

$$
Z_t = (Z_0 - t)_+ + k \mathbb{1}_{\{t=j\}}.
$$

$\square$

Lemma 4. $B_{t_0}^{(1)} = (m - 1) \left(b_0 + \lfloor \frac{t_0}{m} \rfloor - \lfloor \frac{t_0}{n} \rfloor \right) + (m - 1 - (t_0 \bmod n)_+)$.

Proof. By application of lemma 3, $B_{t_0}^{(0)} = nb_0 + n \lfloor \frac{t_0}{m} \rfloor - t_0$. By application of lemma 2,

$$
\begin{aligned}
B_{t_0}^{(1)} &= (m - 1) \left\lfloor \frac{nb_0 + n \lfloor \frac{t_0}{m} \rfloor - t_0}{n} \right\rfloor + (m - 1 - (t_0 \bmod n))_+ \\
&= (m - 1) \left(b_0 + \left\lfloor \frac{t_0}{m} \right\rfloor - \left\lfloor \frac{t_0}{n} \right\rfloor \right) + (m - 1 - (t_0 \bmod n))_+.
\end{aligned}
$$

□

7.A.2.3 Proof of lemma 9

This section is organized as follows:

1. A characterization of t_i by a recursive equation.

2. The introduction of $\tilde{t}_i$ (to approximate t_i).

3. The quantification of the approximation error between t_i and $\tilde{t}_i$.

4. A closed-form computation of $\sum_{i=1}^{m-1} \tilde{t}_i$.

5. The final result.

A characterization of t_i. The following result allows to characterize the sequence of t_i by a recursive equation.

Lemma 5. *For $a, b \geq 0, m, c \geq 2$, if $t^* = \inf\{t \in \mathbb{N}^* : b + c \lfloor \frac{t}{m} \rfloor = (t - a)\}$ then,*

$$t^* = \begin{cases} a + b + c \left\lceil \frac{a+b+1-m}{m-c} \right\rceil & \text{if } m > c, \\ a + b & \text{if } m \leq c \text{ and } a + b < m, \\ +\infty & \text{otherwise.} \end{cases}$$

Proof. First,

$$t^* = \min_{t \in \mathbb{N}^*} t \quad \text{s.t. } b + c \left\lfloor \frac{t}{m} \right\rfloor - (t - a) = 0 \tag{7.19}$$

$$= \min_{t \in \mathbb{N}^*} t \quad \text{s.t. } b + a + (c - m) \left\lfloor \frac{t}{m} \right\rfloor - \left\{ \frac{t}{m} \right\} = 0 \qquad (\{\cdot\} \text{ denotes the fractional part}) \tag{7.20}$$

$$= \min_{\substack{k \in \mathbb{N} \\ j \in [0...m-1]}} km + j \quad \text{s.t. } (a + b) - j + (c - m)k = 0 \text{ and } km + j > 0 \tag{7.21}$$

$$= \min_{k \in \mathbb{N}} a + b + ck \quad \text{s.t. } (a + b) + 1 - m \leq (m - c)k \leq a + b \text{ and } a + b + k > 0. \tag{7.22}$$

Going from eq. (7.20) to eq. (7.21) is done by using the Euclidean division of t by m as $t = km + j$. As eq. (7.22) is linear in k with positive coefficients, it is minimized at the lowest feasible value of k which is

$$k^* = \begin{cases} \left\lceil \frac{a+b+1-m}{m-c} \right\rceil & \text{if } m > c, \\ 0 & \text{if } m \leq c \text{ and } a + b < m, \\ +\infty & \text{otherwise.} \end{cases}$$

The result follows by using the fact that $t^* = k^*m + j^*$ where $j^* = a + b + (1 - m)k^*$.   □

Corollary 4. $\forall i \in \mathbb{N}, t_{i+1} = b_0 - 1 + t_i + \left\lceil \frac{b_0 + t_i}{m-1} \right\rceil$.

Proof. Direct application of lemma 5 from the definition of t_i.   □

Introduction of $\tilde{t}_i$ to approximate t_i.   To obtain a sequence $\tilde{t}_i$ close to t_i with a closed form, the intuition is to "remove" the fractional part and solve the arithmetic-geometric equation.

$$\forall i \in \mathbb{N}^*, \tilde{t}_{i+1} = b_0 - 1 + \tilde{t}_i + \frac{b_0 + \tilde{t}_i}{m-1}, \tag{7.23}$$

$$\tilde{t}_0 = t_0. \tag{7.24}$$

The intuitive justification is that we are in the regime $m = o(\sqrt{T})$, thus the error introduced by ignoring a term of order $\left\{ \frac{b_0 + t_i}{m-1} \right\}$ is small (especially if $t_1 = \Theta(T)$). Now, given that $\tilde{t}_i$ follows an arithmetic-geometric equation, it admits a closed-form expression:

Lemma 6. *For any $i \in \mathbb{N}$,*

$$\tilde{t}_i = \left(1 + \frac{1}{m-1} \right)^i (t_0 + mb_0 - m + 1) - mb_0 + m - 1. \tag{7.25}$$

Proof. Let i be in $\mathbb{N}$.

$$\tilde{t}_{i+1} = b_0 - 1 + \tilde{t}_i + \frac{b_0 + \tilde{t}_i}{m-1}$$

$$\Leftrightarrow \tilde{t}_{i+1} = \frac{m}{m-1}\tilde{t}_i + \frac{m}{m-1}b_0 - 1$$

$$\Leftrightarrow \tilde{t}_{i+1} + mb_0 - m + 1 = \frac{m}{m-1}\tilde{t}_i + \frac{m}{m-1}b_0 - 1 + mb_0 - m + 1$$

$$\Leftrightarrow \tilde{t}_{i+1} + mb_0 - m + 1 = \frac{m}{m-1}\left(\tilde{t}_i + b_0 + (m-1)b_0 - m + 1 \right)$$

$$\Leftrightarrow \tilde{t}_{i+1} + mb_0 - m + 1 = \frac{m}{m-1}\left(\tilde{t}_i + mb_0 - m + 1 \right)$$

$$\Leftrightarrow \tilde{t}_{i+1} = -mb_0 + m - 1 + \left(\frac{m}{m-1} \right)^i (\tilde{t}_0 + mb_0 - m + 1).$$

□

Quantification of the approximation error.

Lemma 7. $\forall i \in \mathbb{N}^*, t_i - \tilde{t}_i < (m-1)\left(\left(1 + \frac{1}{m-1} \right)^i - 1 \right).$

Proof.

$$t_i - \tilde{t}_i = t_{i-1} - \tilde{t}_{i-1} + \left\lceil \frac{t_{i-1} + b_0}{m-1} \right\rceil - \frac{\tilde{t}_{i-1} + b_0}{m-1} \qquad \text{eq. (7.23) and corollary 4}$$

$$= t_{i-1} - \tilde{t}_{i-1} + \frac{t_{i-1} - \tilde{t}_{i-1}}{m-1} + \left\lceil \frac{t_{i-1} + b_0}{m-1} \right\rceil - \frac{t_{i-1} + b_0}{m-1}$$

$$= (t_{i-1} - \tilde{t}_{i-1})\left(1 + \frac{1}{m-1}\right) + \left\lceil \frac{t_{i-1} + b_0}{m-1} \right\rceil - \frac{t_{i-1} + b_0}{m-1}.$$

Thus, by induction, using that $t_0 - \tilde{t}_0 = 0$ (by definition).

$$t_i - \tilde{t}_i < (t_{i-1} - \tilde{t}_{i-1})\left(1 + \frac{1}{m-1}\right) + 1 \tag{7.26}$$

$$< \left(1 + \frac{1}{m-1}\right)^i (m-1) + 1 - m. \tag{7.27}$$

$\square$

Closed-form computation of $\sum_{i=1}^{m-1} \tilde{t}_i$.

Lemma 8. $\sum_{i=0}^{m-1} \tilde{t}_i = \left(\left(1 + \frac{1}{m-1}\right)^{m-1} - 1 + \frac{1}{m}\right) mt_0 + \mathfrak{A}(m, b_0),$ *where*

$$\mathfrak{A}(m, b_0) = m(mb_0 - m + 1)\left(\left(1 + \frac{1}{m-1}\right)^{m-1} - 2 + \frac{1}{m}\right).$$

Proof.

$$\sum_{i=0}^{m-1} \tilde{t}_i = -m(mb_0 - m + 1) + (t_0 + mb_0 - m + 1)\sum_{i=0}^{m-1}\left(1 + \frac{1}{m-1}\right)^i$$

$$= -m(mb_0 - m + 1) - m + (t_0 + mb_0 - m + 1)\frac{\left(1 + \frac{1}{m-1}\right)^m - 1}{1 + \frac{1}{m-1} - 1}$$

$$= -m(mb_0 - m + 1) - m + (t_0 + mb_0 - m + 1)(m-1)\left(\left(1 + \frac{1}{m-1}\right)^m - 1\right)$$

$$= \left(\left(1 + \frac{1}{m-1}\right)^{m-1} - 1 + \frac{1}{m}\right)mt_0$$

$$+ \underbrace{m(mb_0 - m + 1)\left(\left(1 + \frac{1}{m-1}\right)^{m-1} - 2 + \frac{1}{m}\right)}_{\triangleq \mathfrak{A}(m, b_0)}.$$

$\square$

Putting everything together.

Lemma 9. $\sum_{i=1}^{m-1} \left\lfloor \frac{t_i}{m} \right\rfloor = \left(\left(1 + \frac{1}{m-1}\right)^{m-1} - 1 + \frac{1}{m} \right) t_0 + \mathfrak{B}(m, b_0)$ *where,* $\mathfrak{B}(m, b_0) < (e-2)mb_0 + b_0$

Proof.

$$\sum_{i=1}^{m-1} \left\lfloor \frac{t_i}{m} \right\rfloor = \sum_{i=1}^{m-1} \frac{\tilde{t}_i}{m} + \underbrace{\sum_{i=1}^{m-1} \frac{t_i - \tilde{t}_i}{m} - \sum_{i=1}^{m-1} \left\{ \frac{\tilde{t}_i}{m} \right\}}_{\triangleq \mathfrak{E}(m)}.$$

where

$$\mathfrak{E}(m) \leq \sum_{i=1}^{m-1} \frac{t_i - \tilde{t}_i}{m}$$

$$< \frac{m-1}{m} \sum_{i=1}^{m-1} \left(\left(1 + \frac{1}{m-1}\right)^i - 1 \right)$$

$$< \frac{m-1}{m} \left(\frac{\left(1 + \frac{1}{m-1}\right)^m - 1}{1 + \frac{1}{m-1} - 1} - m \right)$$

$$< \frac{(m-1)}{m} \left((m-1) \left(\left(1 + \frac{1}{m-1}\right)^m - 1 \right) - m \right)$$

$$< \frac{(m-1)}{m} \left(m \left(1 + \frac{1}{m-1}\right)^{m-1} + 1 - 2m \right)$$

$$< (m-1) \left(\left(1 + \frac{1}{m-1}\right)^{m-1} - 2 \right) + \frac{m-1}{m}.$$

Using lemma 8 gives

$$\sum_{i=1}^{m-1} \left\lfloor \frac{t_i}{m} \right\rfloor = \left(\left(1 + \frac{1}{m-1}\right)^{m-1} - 1 + \frac{1}{m} \right) t_0 + \underbrace{\mathfrak{E}(m) + \mathfrak{A}(\mathfrak{m}, \mathfrak{b_o})}_{\triangleq \mathfrak{B}(m, b_0)}.$$

where

$$\mathfrak{B}(m, b_0) < \left(\left(1 + \frac{1}{m-1}\right)^{m-1} - 2 + \frac{1}{m} \right) mb_0.$$

$\square$

7.A.2.4 Proof of lemma 13

The objective of this subsection is the compute $\sum_{i=1}^{m-1} \left\lceil \frac{B_{t_i}^{(i)}}{m-i} \right\rceil$. This section is organised as follows:

1. The introduction of Y_i to approximate.

2. The bounding of Y_i.

3. The quantification of the approximation error between Y_i and $\left\lceil \frac{B_{t_i}^{(i)}}{m-i} \right\rceil$.

4. The final result.

Introduction of Y_i. For any $i \geq 1$, Y_i is defined by the following recursion:

$$Y_1 = \left\lceil \frac{B_{t_1}^{(1)}}{m-1} \right\rceil, \tag{7.28}$$

$$Y_{i+1} = Y_i - (\tilde{t}_{i+1} - \tilde{t}_i)\left(\frac{1}{m-i-1} - \frac{1}{m}\right) + 1. \tag{7.29}$$

where $\tilde{t}_i$ is the approximate time dynamic defined in eq. (7.23).

The bounding of Y_i .

Lemma 10. *For $1 \leq i < m-1$,*

$$Y_1 + i - 1 - \frac{\bar{t}_0}{m}g((i+1)/m) \leq Y_i \leq Y_1 + i - 1 - \frac{\bar{t}_0}{m}g(i/m),$$

where $g(z) = \int_{\frac{1}{m}}^{z} \frac{x}{1-x}\exp(x)dx$, and $\bar{t}_0 = t_0 - mb_0 - m + 1$.

Proof. By definition of Y_i, it holds

$$Y_i = Y_1 + i - 1 - \sum_{k=2}^{i}(\tilde{t}_k - \tilde{t}_{k-1})\left(\frac{1}{m-k} - \frac{1}{m}\right)$$

$$= Y_1 + i - 1 - \sum_{k=2}^{i}(\tilde{t}_k - \tilde{t}_{k-1})\frac{k}{m(m-k)}$$

$$= Y_1 + i - 1 - \frac{\bar{t}_0}{m}\sum_{k=2}^{i}\left(1 + \frac{1}{m-1}\right)^k \left(1 - \frac{m-1}{m}\right)\frac{k}{(m-k)} \quad \text{by } eq. \text{ (7.25)}$$

$$= Y_1 + i - 1 - \frac{\bar{t}_0}{m^2}\sum_{k=2}^{i}\frac{k}{m-k}\left(1 + \frac{1}{m-1}\right)^k.$$

where $\bar{t}_0 = t_0 - mb_0 - m + 1$. Moreover, since $(1 + \frac{1}{m-1}) \geq \exp(\frac{1}{m})$, this gives

$$\exp(\frac{k}{m}) \leq (1 + \frac{1}{m-1})^k \leq \exp(\frac{k}{m-1}) \leq \exp(\frac{k}{m})(1 + \frac{2}{m}),$$

Since the function $x \mapsto \frac{x}{1-x}\exp(x)$ is increasing on $\mathbb{R}_+$, we get that (for $i < m - 1$)

$$\underbrace{\int_{\frac{1}{m}}^{\frac{i}{m}} \frac{x}{1-x}\exp(x)dx}_{\triangleq g(i/m)} \leq \underbrace{\frac{1}{m}\sum_{k=2}^{i} \frac{k}{m-k}\exp(\frac{k}{m})}_{A} \leq \int_{\frac{2}{m}}^{\frac{i+1}{m}} \frac{x}{1-x}\exp(x)dx,$$

Or equivalently, for $i < m - 1$,

$$Y_1 + i - 1 - \frac{\bar{t}_0}{m}g((i+1)/m) \leq Y_i \leq Y_1 + i - 1 - \frac{\bar{t}_0}{m}g(i/m).$$

$\square$

7

Quantification of approximation error. The sequence Y_i only approximates well $\left\lceil \frac{B_{t_i}^{(i)}}{m-i} \right\rceil$ as long as it stays positive.

Lemma 11. $\left\lceil \frac{B_{t_i}^{(i)}}{m-i} \right\rceil - Y_i \leq \mathcal{O}(i).$

Proof. By application of lemma 3 on $B_t^{(i)}$

$$B_{t_i}^{(i)} = B_{t_{i-1}}^{(i)} - \left\lceil \frac{B_{t_{i-1}}^{(i)}}{m-i+1} \right\rceil + (m-i)\left(1 + \left\lfloor \frac{t_i}{m} \right\rfloor - \left\lceil \frac{t_{i-1}}{m} \right\rceil\right) - (t_i - t_{i-1}).$$

Thus, using the definition of Y_i in eq. (7.28),

$$\left\lceil \frac{B_{t_i}^{(i)}}{m-i} \right\rceil - Y_i = \frac{B_{t_i}^{(i)}}{m-i} - Y_i + \mathcal{O}(1)$$

$$= \frac{1}{m-i}\left(B_{t_{i-1}}^{(i)} - \left\lceil \frac{B_{t_{i-1}}^{(i)}}{m-i+1} \right\rceil\right) + \left(1 + \left\lfloor \frac{t_i}{m} \right\rfloor - \left\lceil \frac{t_{i-1}}{m} \right\rceil\right) - \frac{t_i - t_{i-1}}{m-i} - Y_i$$

$$+ \mathcal{O}(1)$$

$$= (t_i - t_{i-1})\left(\frac{1}{m} - \frac{1}{m-i}\right) - \left(Y_{i-1} - (\tilde{t}_i - \tilde{t}_{i-1})\left(\frac{1}{m} - \frac{1}{m-i}\right) + 1\right)$$

$$+ \frac{B_{t_{i-1}}^{(i)}}{m-i+1} + 1 + \mathcal{O}(1)$$

$$= \left\lceil \frac{B_{t_{i-1}}^{(i)}}{m-i+1} \right\rceil - Y_{i-1} + (t_i - t_{i-1} - \tilde{t}_i + \tilde{t}_{i-1})\left(\frac{1}{m} - \frac{1}{m-i}\right) + \mathcal{O}(1).$$

By induction

$$\left\lceil \frac{B_{t_i}^{(i)}}{m-i} \right\rceil - Y_i = \sum_{j=2}^{i}(t_j - t_{j-1} - \tilde{t}_j + \tilde{t}_{j-1})\left(\frac{1}{m} - \frac{1}{m-j}\right) + \mathcal{O}(i) \qquad (7.30)$$

$$= \sum_{j=2}^{i}\frac{(t_j - t_{j-1}) - (\tilde{t}_j - \tilde{t}_{j+1})}{m} - \sum_{j=2}^{i}\frac{t_j - t_{j-1} - \tilde{t}_j + \tilde{t}_{j-1}}{m-j} + \mathcal{O}(i)$$

$$(7.31)$$

$$= \frac{(t_i - t_1) - (\tilde{t}_i - \tilde{t}_1)}{m} - \sum_{j=2}^{i}\frac{\left((t_{j-1} - \tilde{t}_{j-1})\left(1 + \frac{1}{m-1}\right) + 1\right) + \tilde{t}_{j-1} - t_{j-1}}{m-j}$$

$$+ \mathcal{O}(i) \qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{by eq. (7.26)}$$

$$(7.32)$$

$$= \frac{t_i - t_1 - \tilde{t}_i + \tilde{t}_1}{m} - \sum_{j=2}^{i}\frac{t_{j-1} - \tilde{t}_{j-1} + 1}{(m-j)(m-1)} + \mathcal{O}(i) \qquad (7.33)$$

$$\leq \mathcal{O}(i) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{by lemma 7.}$$

$$(7.34)$$

□

Lemma 12. $i^* = \lfloor \alpha^* m \rfloor$.

Proof. The objective is to find i^* such that

$$Y_1 + i^* - 1 - \frac{\bar{t}_0}{m}g((i^* + 1)/m) < 0 \leq Y_1 + i^* - 1 - \frac{\bar{t}_0}{m}g(i^*/m). \qquad (7.35)$$

Let define $\alpha^* \in (1/m, 1)$ the solution of

$$\frac{\bar{t}_0}{m}\int_{\frac{1}{m}}^{\alpha}\frac{z}{1-z}e^z\mathrm{d}z - m\alpha^* - Y_1 = 0\,,$$

then $i^* = \lfloor \alpha^* m \rfloor$ satisfies eq. (7.35). □

Lemma 13.

$$\sum_{i=1}^{m-1}\left\lceil \frac{B_i^{(i)}}{m-i} \right\rceil = \lfloor \alpha^* m \rfloor \left\lceil \frac{B_{t_1}^{(1)}}{m} \right\rceil - \bar{t}_0\int_{\frac{1}{m}}^{\alpha^*}g(x)\mathrm{d}x + \frac{g(\alpha^*) - g(1/m)}{m} + \mathcal{O}(m^2).$$

where $\alpha^* \in (1/m, 1)$ the solution of

$$\frac{\bar{t}_0}{m}\int_{\frac{1}{m}}^{\alpha}\frac{z}{1-z}e^z\mathrm{d}z - m\alpha^* - Y_1 = 0.$$

and

$$g(x) = \frac{x(\alpha^* - x)}{1 - x}\left(1 + \frac{1}{m-1}\right)^{mx}$$

.

Proof. Let define $i^* = \lfloor \alpha^* m \rfloor$. Then,

$$\sum_{i=1}^{m-1} \left\lceil \frac{B_i^{(i)}}{m-i} \right\rceil$$

$$= \sum_{i=1}^{i^*} Y_i + \sum_{i=1}^{i^*} \left\lceil \frac{B_i^{(i)}}{m-i} \right\rceil - Y_i + \sum_{i=i^*+1}^{m-1} \left\lceil \frac{B_i^{(i)}}{m-i} \right\rceil$$

$$= \sum_{i=1}^{i^*} Y_i + \sum_{i=1}^{i^*} \left(\frac{t_i - t_1 - \tilde{t}_i + \tilde{t}_1}{m} - \sum_{j=2}^{i} \frac{t_{j-1} - \tilde{t}_{j-1}}{(m-j)(m-1)} + \mathcal{O}(i) \right)$$

$$+ (m-1)(m-1-i^*) - \sum_{i=i^*+1}^{m-1} (t_i \bmod m)$$

$$= \sum_{i=1}^{i^*} Y_i + i^* \frac{\tilde{t}_1 - t_1}{m} + \sum_{i=1}^{i^*} \frac{t_i - \tilde{t}_i}{m} - \sum_{i=1}^{i^*} \sum_{j=2}^{i} \frac{t_{j-1} - \tilde{t}_{j-1}}{(m-j)(m-1)} + \mathcal{O}((i^*)^2)$$

$$+ (m-1)(m-1-i^*) - \sum_{i=i^*+1}^{m-1} (t_i \bmod m)$$

$$= i^* \frac{\tilde{t}_1 - t_1}{m} + \sum_{i=1}^{i^*} \frac{t_i - \tilde{t}_i}{m} - \sum_{i=1}^{i^*} \sum_{j=2}^{i} \frac{1}{(m-j)} \left(\left(1 + \frac{1}{m-1} \right)^{j-1} - 1 \right)$$

$$+ \sum_{i=1}^{i^*} Y_i + (m-1)(m-1-i^*) + \mathcal{O}(m^2) \quad \text{by } i^* = \lfloor \alpha^* m \rfloor \text{ and eq. (7.27)}.$$

$$= \sum_{i=1}^{i^*} Y_i + i^* \frac{\tilde{t}_1 - t_1}{m} + \sum_{i=1}^{i^*} \frac{t_i - \tilde{t}_i}{m} - \frac{1}{(m-1)} \sum_{i=1}^{i^*} \sum_{j=2}^{i} \left(\left(1 + \frac{1}{m-1} \right)^{j-1} - 1 \right)$$

$$+ (m-1)(m-1-i^*) + \mathcal{O}(m^2)$$

$$= \sum_{i=1}^{i^*} Y_i + i^* \frac{\tilde{t}_1 - t_1}{m} + \frac{i^*(i^*+1)}{2(m-1)} - \frac{1}{(m-1)} \sum_{i=1}^{i^*} \sum_{j=2}^{i} \left(\left(1 + \frac{1}{m-1} \right)^{j-1} \right)$$

$$- \frac{i^*}{m-1} + (m-1)(m-1-i^*) + \sum_{i=1}^{i^*} \frac{t_i - \tilde{t}_i}{m} + \mathcal{O}(m^2)$$

$$= \sum_{i=1}^{i^*} Y_i + i^* \frac{\tilde{t}_1 - t_1}{m} + \sum_{i=1}^{i^*} \frac{t_i - \tilde{t}_i}{m} + \frac{i^*(i^*+1)}{2(m-i^*)} + (m-1)(m-1-i^*) + \mathcal{O}(m^2)$$

$$= \sum_{i=1}^{i^*} Y_i + \mathcal{O}(m^2) \qquad \text{by lemma 7 and } i^* = \lfloor \alpha^* m \rfloor.$$

$$= \sum_{i=1}^{i^*} \left(Y_1 + i - 1 - \frac{\bar{t}_0}{m^2} \sum_{k=2}^{i} \frac{k}{m-k} \left(1 + \frac{1}{m-1} \right)^k \right) + \mathcal{O}(m^2) \text{ by eq. (7.30)}$$

(7.36)

$$= i^* Y_1 + \frac{i^*(i^*+1)}{2} - i^* - \frac{\bar{t}_0}{m^2} \sum_{i=1}^{i^*} \sum_{k=2}^{i} \frac{k}{m-k} \left(1 + \frac{1}{m-1} \right)^k + \mathcal{O}(m^2) \quad (7.37)$$

$$= i^* Y_1 - \frac{\bar{t}_0}{m^2} \sum_{i=1}^{i^*} \sum_{k=2}^{i} \frac{k}{m-k} \left(1 + \frac{1}{m-1} \right)^k + \mathcal{O}(m^2) \quad \text{by } i^* = \lfloor \alpha^* m \rfloor \quad (7.38)$$

$$= i^* Y_1 - \frac{\bar{t}_0}{m^2} \sum_{k=2}^{i^*} \frac{(i^*-k)k}{m-k} \left(1 + \frac{1}{m-1} \right)^k + \mathcal{O}(m^2) \quad (7.39)$$

$$= i^* Y_1 - \frac{\bar{t}_0}{m} \sum_{k=2}^{i^*} \frac{(\frac{i^*}{m} - \frac{k}{m})\frac{k}{m}}{1 - \frac{k}{m}} \left(\left(1 + \frac{1}{m-1} \right)^m \right)^{\frac{k}{m}} + \mathcal{O}(m^2)$$

$$= i^* Y_1 - \frac{\bar{t}_0}{m} \sum_{k=2}^{i^*} g\left(\frac{k}{m} \right) + \mathcal{O}(m^2) \quad (7.40)$$

$$\text{with } g(x) = \frac{x(\alpha^* - x)}{1 - x} \left(1 + \frac{1}{m-1} \right)^{mx} \quad (7.41)$$

$$= i^* Y_1 - \frac{\bar{t}_0}{m} \sum_{k=2}^{i^*} g\left(\frac{k}{m} \right) + \mathcal{O}(m^2) \quad (7.42)$$

$$= i^* \left\lceil \frac{B_{t_1}^{(1)}}{m-1} \right\rceil - \bar{t}_0 \int_{\frac{1}{m}}^{\alpha^*} g(x)\, dx + \frac{g(\alpha^*) - g(1/m)}{m} + \mathcal{O}(m^2). \quad (7.43)$$

Moving from eq. (7.42) to eq. (7.43) arises from approximating a Riemann sum by an integral. $\qquad \square$

7.A.3 Proof of theorem 18

Theorem 18. *Assuming the initial budgets are $b_{1,0} = b_{2,0} = \cdots = b_{n,0} = b_0 \geq 1$, for $m = o(\sqrt{T})$,*

$$\sup_{\text{ALG}} \mathbb{E}\left[\text{CR}^{\text{adv}}(\text{ALG}, \mathcal{G}_{T,m}) \right] \leq \text{CR}^{\text{adv}}(\text{Balance}, G^{th.2}) + o_T(1).$$

where the expectation is taken over the randomness from ALG*.*

Proof. The proof is based on the adversarial graph design defined in section 7.A.2 and organized in two steps:

1. Showing that the sequence of total budget decreases at least as fast as the one of Balance.

2. Using this in eq. (7.9) to show that $\texttt{ALG}(G^{\text{th.2}}) \leq \texttt{Balance}(G^{\text{th.2}}) + o(T)$.

$\texttt{ALG}$ is assumed to be any matching algorithm, potentially randomized. The matching built by $\texttt{ALG}$ is denoted $\mathbf{x}$, the graph $G^{\text{th.2}}$ is adversarially defined based on $\mathbf{x}$ as in section 7.A.2 and we use the rest of the notation defined there. Note that only the choice of nodes in $U_1, \ldots, U_{m-1}$ differs from the graph adversarial to $\texttt{ALG}$, not the other quantities such as t_i.

For any $i \leq m-1$ and $t \in [T]$, the total budget of $\texttt{ALG}$ is denoted $B_t^{(i)} = \mathbb{E}\left[\sum_{u \in U_i} x_{u,t}\right]$ where the expectation is taken over the randomness of $\texttt{ALG}$. We denote $B_t^{(i),\texttt{bal}}$ for the sequence generated by $\texttt{Balance}$ in section 7.A.2 for comparison.

Dynamic of $B_{t_i}^{(i)}$. First, at time t_0, by application of lemma 3, $B_{t_0}^{(0)} = B_{t_0}^{(0),\texttt{bal}}$. The key element of the proof is to use that

$$\forall y \in \mathbb{R}_+^n, \forall k \leq n, \sum_{j=1}^{k} y_{(n-j)} \leq \frac{k}{n} \|y\|_1 \quad \text{where } y_{(j)} \geq 0 \text{ is the } i^{th} \text{ largest coordinate of } y.$$

$$(7.44)$$

Thus, by applying eq. (7.44) on $b_{\cdot, t_0}$, after the adversary removes $n - m + 1$ nodes to build U_1, we have $B_{t_0}^{(1)} \leq B_{t_0}^{(1),\texttt{bal}} + m - 1$. The term $r_0 = m - 1$ comes from the fact $\texttt{Balance}$ does not exactly equalize the budgets (see lemma 2) and a randomized balance algorithm could do it more accurately in expectation. By induction, using at each step lemma 3 and eq. (7.44), we obtain that $\forall i \leq m-1, B_{t_i}^{(i)} \leq B_{t_i}^{(i),\texttt{bal}} + i(m-i)$

Showing that $\texttt{ALG}(G^{\textbf{th.2}}) \leq \texttt{Balance}(G^{\textbf{th.2}}) + o(T)$. Denoting i_t the phase of the graph to which time step t belongs, it is possible to show that,

$$\texttt{ALG}(G^{\text{th.2}}) = t^* + \sum_{i=i_{t^*}+1}^{m-1} (m-i)\left(\left\lfloor \frac{t_i}{m} \right\rfloor - \left\lfloor \frac{t_{i-1}}{m} \right\rfloor\right) + \mathcal{O}(m)$$

$$\text{where } t^* = \max\{t \in [T] : B_t^{(i_t)} \geq 1\}.$$

Noting that, $B_t^{(i_t),\texttt{bal}} \leq B_t^{(i_t)} + i_t(m - i_t)$, leads to $\texttt{ALG}(G^{\text{Th.2}}) \leq \texttt{Balance}(G^{\text{Th.2}}) + m^2$, which gives,

$$\texttt{CR}^{\text{adv}}(\texttt{ALG}, \mathcal{G}_{T,m}) \leq \texttt{CR}(\texttt{ALG}, G^{\text{Th.2}}) \leq \texttt{CR}(\texttt{Balance}, G^{\text{Th.2}}) + o_T(1),$$

where $\texttt{CR}(\texttt{ALG}, G^{\text{th.2}}) = \frac{\texttt{ALG}(G^{\text{th.2}})}{\texttt{OPT}(G^{\text{th.2}})}$.

$\square$

7.B   Stochastic Case

7.B.1   Proof of theorem 19

Theorem 19. *With probability* $1 - \mathcal{O}\left(n^{1/4} \exp(-a^3 n^{1/4})\right)$, *the matching size created by* Greedy *denoted by* Greedy(G, T) *satisfies,*

$$\text{Greedy}(G, T) = nh(T/n) + \mathcal{O}(n^{3/4}).$$

and,

$$\frac{\mathbb{E}[\text{Greedy}(G, T)]}{n} - h(T/n) \underset{n \to +\infty}{\to} 0.$$

where $h(\tau)$ *is solution of the following equation,*

$$\dot{h}(\tau) = 1 - e^{-a(1 - z_0(\tau))}, \qquad \frac{1}{n} \leq \tau \leq \frac{T}{n}.$$

and $z_0(\tau)$ *satisfies the following system,*

$$
\begin{cases}
\dot{z}_0(\tau) = -z_0(\tau)\beta + \frac{z_1(\tau)}{1 - z_0(\tau)}(1 - e^{-a + az_0(\tau)}) & \text{for } k = 0, \\
\dot{z}_k(\tau) = (z_{k-1}(\tau) - z_k(\tau))\beta + (z_{k+1}(\tau) - z_k(\tau))\frac{1 - e^{-a + az_0(\tau)}}{1 - z_0(\tau)} & \text{for } 1 \leq k \leq K - 1, \\
\dot{z}_k(\tau) = \beta z_{k-1}(\tau) - z_k(\tau)\frac{1 - e^{-a(1 - z_0(\tau))}}{1 - z_0(\tau)} & \text{for } k = K, \\
\sum_{k=0}^{K} z_k(\tau) = 1.
\end{cases}
$$

(7.2)

The proof of theorem 19 is based on Wormald's theorem introduced in [97, 98] and is organized as follows:

1. Definition of the evolution of $(Y_k(t))_{k \geq 0}$ and Greedy(G, t).

2. Proving that $(Y_k(t))_{k \geq 0}$ and Greedy(G, t) satisfy the hypotheses of the Wormald theorem.

3. Application of Wormald theorem on $(Y_k(t))_{k \geq 0}$ and Greedy(G, t).

Recall that for all $u \in U$, $b_{u,t} \in \mathbb{N}$ is given by,

$$b_{u,t} = \min(K, b_{u,t-1} - x_{u,t} + \eta_t), \qquad \text{with } b_{u,0} = b_0 \geq 1 \quad \text{and } K \in \mathbb{N}. \qquad (7.45)$$

Here η_t is a realization of a Bernoulli random variable with parameter $\frac{\beta}{n}$ denoted $\mathcal{B}\left(\frac{\beta}{n}\right)$.

First, let us introduce some notations, for $k \in \mathbb{N}$ and $t \in [T]$,

- $U_k(t) = \{u \in U : b_{u,t} = k\}$ is the set of nodes with budget k.

- $Y_k(t) = |U_k(t)|$ is the number of nodes with budget equals to k.

- $\texttt{Greedy}(G, T) = \sum_{u \in U} \sum_{t=1}^{T} x_{u,t}$ is the matching size.

- $C(t) = \sum_{k \geq 1} Y_k(t) = n - Y_0(t)$ is the total number of nodes with budget at least equals to 1.

In order to apply Wormald's theorem, it is necessary to track the evolution of $\texttt{Greedy}(G, T)$. To achieve this, we must precisely quantify the one-step change in $\texttt{Greedy}(G, t)$ for all $t \in [T]$. This crucial step is addressed in the forthcoming lemma,

Lemma 14. *For $t \in [T]$, the expectation of the one-step change of $\texttt{Greedy}(G, t)$ is given by,*

$$\mathbb{E}\left[\texttt{Greedy}(G, t+1) - \texttt{Greedy}(G, t) | \texttt{Greedy}(G, t)\right] = 1 - \left(1 - \frac{a}{n}\right)^{\sum_{k \geq 1} Y_k(t)}$$

$$= 1 - \left(1 - \frac{a}{n}\right)^{n - Y_0(t)}.$$

Proof. For $t \in [T]$, the matching size at time $t + 1$ is defined as follows,

$$\texttt{Greedy}(G, t+1) = \texttt{Greedy}(G, t) + \mathbb{1}_{\{x_{u,t+1}=1, \ u \in U_k(t+1)\}}.$$

Moving to conditional expectation gives,

$$\mathbb{E}\left[\texttt{Greedy}(G, t+1) - \texttt{Greedy}(G, t) | \texttt{Greedy}(G, t)\right] = \mathbb{P}\left(x_{u,t+1} = 1, \ u \in U_k | \texttt{Greedy}(G, t)\right)$$

$$= 1 - \left(1 - \frac{a}{n}\right)^{C(t)}$$

$$= 1 - \left(1 - \frac{a}{n}\right)^{n - Y_0(t)}.$$

$\square$

Since the evolution of $\texttt{Greedy}(G, t)$ depends on Y_0, we need to quantify the evolution of $Y_k(t), \forall k \in \mathbb{N}, t \in [T]$. This is done in the subsequent lemma,

Lemma 15. *For $t \in [T]$, denoting $\Sigma(t) = \frac{1}{pC(t)}(1 - (1-p)^{C(t)})$, the expectation of the one-step change of Y_k, when matching is built using $\texttt{Greedy}$ algorithm is given by,*

$$\begin{cases} \mathbb{E}\left[\Delta_0(t)|\mathbf{Y}(t)\right] = -Y_0(t)\left[\frac{\beta}{n}(1-p\,\Sigma(t))\right] + Y_1(t)(1-\frac{\beta}{n})p\,\Sigma(t), \\ \mathbb{E}\left[\Delta_1(t)|\mathbf{Y}(t)\right] = -Y_1(t)\left[\frac{\beta}{n}(1-p\,\Sigma(t)) + (1-\frac{\beta}{n})p\,\Sigma(t)\right] + Y_0(t)\frac{\beta}{n} \\ +Y_2(t)(1-\frac{\beta}{n})p\,\Sigma(t), \\ \mathbb{E}\left[\Delta_k(t)|\mathbf{Y}(t)\right] = \frac{\beta}{n}(1-p\,\Sigma(t))\left[Y_{k-1}(t) - Y_k(t)\right] + \left[Y_{k+1}(t) - Y_k(t)\right](1-\frac{\beta}{n})p\,\Sigma(t) \\ \forall k > 1. \end{cases}$$

$$(7.46)$$

where $\forall k \geq 0,\ \Delta_k(t) = Y_k(t+1) - Y_k(t).$

Proof. For $t \in [T]$, the evolution of the number of nodes with budget $k \in \mathbb{N}$ can be formulated as,

$$Y_k(t+1) = Y_k(t) - \sum_{u\in U_k(t)} \mathbb{1}_{\{\{\eta_t=1\}\cap\{x_{u,t}=0\}\cup\{\{x_{u,t}=1\}\cap\{\eta_t=0\}\}\}} + \sum_{u\in U_{k-1}(t)} \mathbb{1}_{\{\{\eta_t=1\}\cap\{x_{u,t}=0\}\}}$$

$$+ \sum_{u\in U_{k+1}(t)} \mathbb{1}_{\{\{x_{u,t}=1\}\cap\{\eta_t=0\}\}}.$$

$$(7.47)$$

We are interested in the conditional expectation of eq. (7.47) denoted by $E(t) = \mathbb{E}\left[Y_k(t+1) - Y_k(t)|\mathbf{Y}(t)\right]$ where $\mathbf{Y}(t) = (Y_k(t))_{k\geq 0},$

$$E(t) \hspace{9cm} (7.48)$$

$$= -\sum_{u\in U_k(t)} \mathbb{P}\left(\{\{\eta_t=1\}\cap\{x_{u,t}=0\}\}\cup\{\{x_{u,t}=1\}\cap\{\eta_t=0\}\}|\mathbf{Y}(t)\right)$$

$$+\sum_{u\in U_{k-1}(t)} \mathbb{P}\left(\{\{\eta_t=1\}\cap\{x_{u,t}=0\}\}\Big|\mathbf{Y}(t)\right)$$

$$+\sum_{u\in U_{k+1}(t)} \mathbb{P}\left(\{\{x_{u,t}=1\}\cap\{\eta_t=0\}\}\Big|\mathbf{Y}(t)\right) \hspace{3cm} (7.49)$$

$$= -\sum_{u\in U_k(t)} \mathbb{P}\left(\{\{\eta_t=1\}\cap\{x_{u,t}=0\}\}|\mathbf{Y}(t)\right) - \sum_{u\in U_k(t)} \mathbb{P}\left(\{\{\eta_t=0\}\cap\{x_{u,t}=1\}\}|\mathbf{Y}(t)\right)$$

$$+\sum_{u\in U_{k-1}(t)} \mathbb{P}\left(\{\{\eta_t=1\}\cap\{x_{u,t}=0\}\}\Big|\mathbf{Y}(t)\right)$$

$$+\sum_{u\in U_{k+1}(t)} \mathbb{P}\left(\{\{x_{u,t}=1\}\cap\{\eta_t=0\}\}\Big|\mathbf{Y}(t)\right) \hspace{3cm} (7.50)$$

$$= -\sum_{u\in U_k(t)} \frac{\beta}{n}\mathbb{P}\left(\{x_{u,t}=0\}|\mathbf{Y}(t)\right) - \sum_{u\in U_k(t)} (1-\frac{\beta}{n})\mathbb{P}\left(\{x_{u,t}=1\}|\mathbf{Y}(t)\right)$$

$$+\sum_{u\in U_{k-1}(t)} \frac{\beta}{n}\mathbb{P}\left(\{x_{u,t}=0\}\Big|\mathbf{Y}(t)\right) + \sum_{u\in U_{k+1}(t)} (1-\frac{\beta}{n})\mathbb{P}\left(\{x_{u,t}=1\}|\mathbf{Y}(t)\right). \hspace{1cm} (7.51)$$

Moving from eq. (7.49) to eq. (7.50) and then from eq. (7.50) to eq. (7.51) is

done using independence.

To get the final expression of $E(t)$, we need to compute $\mathbb{P}\left[\{x_{u,t} = 1\}\middle|\mathbf{Y}(t)\right]$. By using Bayes formula we can see that,

$$\mathbb{P}\left[\{x_{u,t} = 1\}\middle|\mathbf{Y}(t)\right] = \mathbb{P}\left[\{(u,t) \in G\}\middle|\mathbf{Y}(t)\right] \mathbb{P}\left[\{x_{u,t} = 1\}\middle|\mathbf{Y}(t), (u,t) \in G\right]$$
$$= p\,\mathbb{P}\left[\{x_{u,t} = 1\}\middle|\mathbf{Y}(t), (u,t) \in G\right].$$

Now, let us compute $\mathbb{P}\left[\{x_{u,t} = 1\}\middle|\mathbf{Y}(t), (u,t) \in G\right]$,

$$\mathbb{P}\left[\{x_{u,t} = 1\}\middle|\mathbf{Y}(t), (u,t) \in G\right] = \sum_{i=1}^{C(t)} \mathbb{P}\left[\{x_{c,t} = 1, c \in [i], B_c(t) \geq 1\}\middle|\mathbf{Y}(t), (u,t) \in G\right]$$

$$= \sum_{i=1}^{C(t)} \mathbb{P}\left[\{x_{c,t} = 1\}\middle|c \in [i], B_c(t) \geq 1, \mathbf{Y}(t), (u,t) \in G\right]$$

$$\mathbb{P}\left[c \in [i], B_c(t) \geq 1\middle|\mathbf{Y}(t), (u,t) \in G\right]$$

$$= \sum_{i=1}^{C(t)} \frac{1}{i}\mathbb{P}\left[c \in [i], B_c(t) \geq 1\middle|\mathbf{Y}(t), (u,t) \in G\right]$$

$$= \sum_{i=1}^{C(t)} \frac{1}{i}\binom{C(t)-1}{i-1} p^{i-1}(1-p)^{C(t)-i}$$

$$= \underbrace{\frac{1}{pC(t)}(1 - (1-p)^{C(t)})}_{\Sigma(t)}.$$

Thus, we get

$$\mathbb{P}\left[\{x_{u,t} = 1\}\middle|\mathbf{Y}(t)\right] = \frac{1}{C(t)}(1 - (1-p)^{C(t)}).$$

Due to `Greedy` algorithm, here the choice of u inside the probabilities doesn't depend on U_k, so putting everything together in $E(t)$, and distinguishing cases where $k = 0, k = 1$ and $k \geq 2$, we get,

$$\begin{cases} \mathbb{E}\left[\Delta_0(t)|\mathbf{Y}(t)\right] = -Y_0(t)\left[\frac{\beta}{n}(1 - p\,\Sigma(t))\right] + Y_1(t)(1 - \frac{\beta}{n})p\,\Sigma(t), \\ \mathbb{E}\left[\Delta_1(t)|\mathbf{Y}(t)\right] = -Y_1(t)\left[\frac{\beta}{n}(1 - p\,\Sigma(t)) + (1 - \frac{\beta}{n})p\,\Sigma(t)\right] + Y_0(t)\frac{\beta}{n} \\ +Y_2(t)(1 - \frac{\beta}{n})p\,\Sigma(t), \\ \mathbb{E}\left[\Delta_k(t)|\mathbf{Y}(t)\right] = \frac{\beta}{n}(1 - p\,\Sigma(t))\left[Y_{k-1}(t) - Y_k(t)\right] + \left[Y_{k+1}(t) - Y_k(t)\right](1 - \frac{\beta}{n})p\,\Sigma(t) \\ \forall k > 1. \end{cases}$$

$$(7.52)$$

where $\forall k \geq 0, \ \Delta_k(t) = Y_k(t+1) - Y_k(t)$.

$\square$

Before establishing the hypotheses of Wormald's theorem, we introduce the following technical lemma,

Lemma 16. *For $n > 0$, $a \leq n/2$ and $0 \leq w \leq 1$,*

$$0 \leq e^{-aw} - \left(1 - \frac{a}{n}\right)^{nw} \leq \frac{a}{ne}.$$

Proof. Using the following inequalities: $1 - x \geq e^{-x-x^2}$ for $x \leq \frac{1}{2}$ and $1 - x \leq e^{-x}$ for $x \geq 0$, we obtain $e^{-aw}\left(1 - \frac{a^2 w}{n}\right) \leq \left(1 - \frac{a}{n}\right)^{nw} \leq e^{-aw}$. The result follows by rearranging terms and using that $awe^{-aw} \leq 1/e$.

$\square$

To apply Wormald's theorem [98] in our model, three key hypotheses need to be met: the boundedness hypothesis, the Lipschitz hypothesis, and the trend hypothesis. These hypotheses will be established in the following lemmas for both $\mathtt{Greedy}(G, t)$ and $Y_k(t)$.

Lemma 17. $\forall k \geq 0$, *let* $-\epsilon < \tau < \frac{T}{n} + \epsilon$ *and* $-\epsilon < z_k < 1 + \epsilon$ *where* $\epsilon > 0$. *The functions $f_k(\tau)$ and $j_0(\tau)$ defined as follows,*

$$f_k(\tau) = \begin{cases} -z_0(\tau)\beta + \frac{z_1(\tau)}{1-z_0(\tau)}(1 - e^{-a+az_0(\tau)}) & \text{for } k = 0, \\ (z_{k-1}(\tau) - z_k(\tau))\beta + (z_{k+1}(\tau) - z_k(\tau))\frac{1-e^{-a+az_0(\tau)}}{1-z_0(\tau)} & \text{for } k \geq 0. \end{cases}$$
$$j_0(\tau) = 1 - e^{-a(1-z_0(\tau))}$$

are Lipschitz with a constant $L = (\beta + a)(1 + \epsilon)$ and $L' = ae^{a\epsilon}$ respectively.

Proof. The proof is done for $k = 0$ and remains the same for $k \geq 1$. Let $-\epsilon < z_0 < 1 + \epsilon$, $-\epsilon < z_1 < 1 + \epsilon$, $-\epsilon < \tau < \frac{T}{n} + \epsilon$ and $-\epsilon < \tau' < \frac{T}{n} + \epsilon$.

$|f_0(\tau) - f_0(\tau')|$

$= \left| -z_0(\tau)\beta + \frac{z_1(\tau)}{1 - z_0(\tau)}(1 - e^{-a+az_0(\tau)}) + z_0(\tau')\beta - \frac{z_1(\tau')}{1 - z_0(\tau')}(1 - e^{-a+az_0(\tau')}) \right|$

$\leq \beta \left| z_0(\tau') - z_0(\tau) \right| + \left| \frac{z_1(\tau)}{1 - z_0(\tau)}(1 - e^{-a+az_0(\tau)}) + \frac{z_1(\tau')}{1 - z_0(\tau')}(1 - e^{-a+az_0(\tau')}) \right|$

$\leq \beta \left| z_0(\tau') - z_0(\tau) \right| + a \left| z_1(\tau) + z_1(\tau') \right| \quad \text{using}(1 - e^{-ax} \leq ax)z.$

Thus we get,
$$|f_0(\tau) - f_0(\tau')| \leq (\beta + a + 2)(1 + \epsilon)|\tau - \tau'|.$$

Therefore, we proved that f_0 is L-Lipschitz with $L = (\beta + a + 2)(1 + \epsilon)$. Now, let us proceed to prove that j_0 is Lipschitz,

$$
\begin{aligned}
|j_0(\tau) - j_0(\tau')| &= |e^{-a(1-z_0(\tau))} - e^{-a(1-z_0(\tau'))}| \\
&= e^{-a(1-z_0(\tau))}|1 - e^{a(z_0(\tau')-z_0(\tau))}| \\
&\leq e^{-a(1-z_0(\tau))}a|z_0(\tau') - z_0(\tau)| \qquad \text{using } 1 - e^{ax} \leq -ax \\
&\leq e^{a\epsilon}a|\tau - \tau'|.
\end{aligned}
$$

Hence j_0 is L'-Lipschitz with $L' = e^{a\epsilon}a$.

$\square$

The next lemma proves the trend hypothesis,

Lemma 18. *For $t \in [T]$ the functions $f_k\left(\frac{t}{n}, \frac{Y_0(t)}{n}, \ldots, \frac{Y_K(t)}{n}\right)$ and $j(\frac{t}{n}, \frac{Y_0(t)}{n})$ are given by,*

$$
f_k =
$$
$$
\begin{cases}
-\frac{Y_0(t)\beta}{n}(1 - \frac{1}{n-Y_0(t)}(1 - e^{-a(1-\frac{Y_0(t)}{n})})) + \frac{Y_1(t)(n-a)}{n}\frac{1}{n-Y_0(t)}(1 - e^{-a(1-\frac{Y_0(t)}{n})}) & \text{for } k = 0, \\[1.5ex]
\frac{-Y_1(t)}{n}\left[a(1 - \frac{1}{n-Y_0(t)}(1 - e^{-a(1-\frac{Y_0(t)}{n})})) + \frac{(n-a)}{n-Y_0(t)}(1 - e^{-a(1-\frac{Y_0(t)}{n})})\right] & \\[1ex]
+\frac{Y_2(t)(n-a)}{n(n-Y_0(t))}(1 - e^{-a(1-\frac{Y_0(t)}{n})}) + \frac{Y_0(t)a}{n} & \text{for } k = 1, \\[1.5ex]
\frac{-Y_k(t)}{n}\left[a(1 - \frac{1}{n-Y_0(t)}(1 - e^{-a(1-\frac{Y_0(t)}{n})})) + \frac{(n-a)}{n-Y_0(t)}(1 - e^{-a(1-\frac{Y_0(t)}{n})})\right] & \\[1ex]
+\frac{Y_{k+1}(t)(n-a)}{n(n-Y_0(t))}(1 - e^{-a(1-\frac{Y_0(t)}{n})}) + \frac{Y_{k-1}(t)a}{n}(1 - \frac{1}{n-Y_0(t)}(1 - e^{-a(1-\frac{Y_0(t)}{n})})) & \text{for } k < K, \\[1.5ex]
\frac{Y_{k-1}(t)\beta}{n}(1 - \frac{1}{n-Y_0(t)}(1 - e^{-a(1-\frac{Y_0(t)}{n})})) - \frac{Y_k(t)(n-a)}{n}\frac{1}{n-Y_0(t)}(1 - e^{-a(1-\frac{Y_0(t)}{n})}) & \text{for } k = K.
\end{cases}
$$
$$
j = 1 - e^{-a(1-\frac{Y_0(t)}{n})}.
$$

and we have for all $k \geq 1$,

$$
\left|\mathbb{E}(\texttt{Greedy}(G, t+1) - \texttt{Greedy}(G, t)|\texttt{Greedy}(G, t)) - j\left(\frac{t}{n}, \frac{Y_0(t)}{n}\right)\right| \leq \frac{a}{en}. \quad (7.53)
$$

$$
\left|\mathbb{E}(Y_k(t+1) - Y_k(t)|\mathbf{Y}(t)) - f_k\left(\frac{t}{n}, \frac{Y_0(t)}{n}, \ldots, \frac{Y_k(t)}{n}, \ldots\right)\right| \leq \frac{a}{en}. \quad (7.54)
$$

Proof. Let's prove eq. (7.54) for $k = 0$ (the proof is the same for $k \geq 1$),

$$
\begin{aligned}
M_0 &= \left|\mathbb{E}(Y_0(t+1) - Y_0(t)|\mathbf{Y}(t)) - f_0\left(\frac{t}{n}, \frac{Y_0(t)}{n}, \frac{Y_1(t)}{n}\right)\right| \\
M_0 &\leq \left|\frac{-Y_0(t)\beta}{n(n - Y_0(t))}(1 - (1 - \frac{a}{n})^{n-Y_0(t)} - 1 + e^{-a(1-\frac{Y_0(t)}{n})})\right| \\
&\quad + \left|\frac{Y_1(t)(n - \beta)}{n(n - Y_0(t))}(1 - (1 - \frac{a}{n})^{n-Y_0(t)} - 1 + e^{-a(1-\frac{Y_0(t)}{n})})\right| \\
&\leq \frac{a}{ne} \qquad \text{(using lemma 27).}
\end{aligned}
$$

Let's now prove eq. (7.53),

$$P = \left| \mathbb{E}(\texttt{Greedy}(G, t+1) - \texttt{Greedy}(G, t)|\texttt{Greedy}(G, t)) - j\left(\frac{t}{n}, \frac{Y_0(t)}{n}\right) \right|$$

$$P = \left| e^{-a(1-\frac{Y_0(t)}{n})} - ((1 - \frac{a}{n})^{n-Y_0(t)}) \right|$$

$$\leq \frac{a}{ne} \qquad \text{(using lemma 27)}.$$

$\square$

The following lemma shows the Boundness hypothesis,

Lemma 19. *For* $t \in [T]$, $k \geq 0$,

$$|\texttt{Greedy}(G, t+1) - \texttt{Greedy}(G, t)| \leq \beta',$$
$$|Y_k(t+1) - Y_k(t)| \leq \beta.$$

with $\beta, \beta' > 0$.

Proof. For $t \in [T]$ and $k \geq 0$,

$$|\texttt{Greedy}(G, t+1) - \texttt{Greedy}(G, t)| = \mathbb{1}_{\{\{x_{u,t+1}=1, u \in U_k(t+1)\}\}} \leq 1.$$

Hence we have $\beta' = 1$.

As seen previously, $Y_k(t)$ is the number of nodes with budget equals to k at time t. So, by the nature of the matching process,

$$|Y_k(t+1) - Y_k(t)| \leq 1.$$

Hence $\beta = 1$.

$\square$

In the following lemma we approximate with high probability $Y_k(t), \forall k \geq 0, t \in [T]$ by the solution of a system of differential equations,

Lemma 20. *With probability* $1 - \mathcal{O}\left(n^{1/4}\exp(-a^3 n^{1/4})\right)$,

$$Y_k(T) = nz_k(T/n) + \mathcal{O}(n^{3/4}) \quad \text{for } k \geq 0.$$

$\forall \; \tau \in [\frac{1}{n}, \frac{T}{n}]$, $(z_0, \dots, z_K)$ *is the solution of the following system,*

$$\begin{cases} \dot{z}_0(\tau) = -z_0(\tau)\beta + \frac{z_1(\tau)}{1-z_0(\tau)}(1 - e^{-a+az_0(\tau)}) & \text{for } k = 0, \\ \dot{z}_k(\tau) = (z_{k-1}(\tau) - z_k(\tau))\beta + (z_{k+1}(\tau) - z_k(\tau))\frac{1-e^{-a+az_0(\tau)}}{1-z_0(\tau)} & \text{for } 1 \leq k \leq K-1, \\ \dot{z}_k(\tau) = \beta z_{k-1}(\tau) - z_k(\tau)\frac{1-e^{-a(1-z_0(\tau))}}{1-z_0(\tau)} & \text{for } k = K, \\ \sum_{k=0}^{K} z_k(\tau) = 1. \end{cases}$$

$$(7.55)$$

Proof. For $\frac{1}{n} \leq \tau \leq \frac{T}{n}$, let us consider the normalized random variable $Z_k(\tau) = \frac{Y_k(\tau n)}{n}$ and $\mathbf{Z}(\tau) = (Z_k(\tau))_{k \geq 0}$. The conditional expectation of the one-step change of $Z_k(\tau)$ for different values of k is given by,

- For $k = 0$,

$$\frac{\mathbb{E}\left[Z_0(\tau + \frac{1}{n}) - Z_0(\tau)|\mathbf{Z}(\tau)\right]}{1/n}$$

$$= \frac{\mathbb{E}\left[Y_0(\tau n + 1)/n - Y_0(\tau n)/n|\mathbf{Y}(\tau n)/n\right]}{1/n}$$

$$= \frac{\mathbb{E}\left[-Z_0(\tau)\left[\frac{\beta}{n}(1 - \frac{1}{n - nZ_0(\tau)}(1 - (1-p)^{n-nZ_0(\tau)}))\right]\right]}{1/n}$$

$$+ \frac{\mathbb{E}\left[Z_1(\tau)(1 - \frac{\beta}{n})\frac{1}{n - nZ_0(\tau)}(1 - (1-p)^{n-nZ_0(\tau)})\right]}{1/n},$$

when $n \to +\infty$, we get,

$$\dot{z}_0(\tau) = -z_0(\tau)\beta + \frac{z_1(\tau)}{1 - z_0(\tau)}(1 - e^{-a+az_0(\tau)}).$$

- For $k = 1$,

$$\frac{\mathbb{E}\left[Z_1(\tau + \frac{1}{n}) - Z_1(\tau)|\mathbf{Z}(\tau)\right]}{1/n}$$

$$= \frac{\mathbb{E}\left[Y_1(\tau n + 1)/n - Y_1(\tau n)/n|\mathbf{Y}(\tau n)/n \, \forall k \geq 1\right]}{1/n}$$

$$= \frac{\mathbb{E}\left[-Z_1(\tau)\left[\frac{\beta}{n}(1 - \frac{1}{n - nZ_0(\tau)}(1 - (1-p)^{n-nZ_0(\tau)}))\right]\right]}{1/n}$$

$$+ \frac{\mathbb{E}\left[-Z_1(\tau)(1 - \frac{\beta}{n})\frac{1}{n - nZ_0(\tau)}(1 - (1-p)^{n-nZ_0(\tau)}))\right]}{1/n}$$

$$+ \frac{\mathbb{E}\left[Z_0(\tau)\frac{\beta}{n} + Z_2(\tau)(1 - \frac{\beta}{n})\frac{1}{n - nZ_0(\tau)}(1 - (1-p)^{n-nZ_0(\tau)})\right]}{1/n},$$

when $n \to +\infty$ we get,

$$\dot{z}_1(\tau) = \beta(z_0(\tau) - z_1(\tau)) + (z_2(\tau) - z_1(\tau))\frac{1 - e^{-a+az_0(\tau)}}{1 - z_0(\tau)}.$$

- $k \geq 2$,

$$\frac{\mathbb{E}\left[Z_k(\tau + \frac{1}{n}) - Z_k(\tau)|\mathbf{Z}(t)\right]}{1/n}$$

$$= \frac{\mathbb{E}\left[Y_k(\tau\,n + 1)/n - Y_k(\tau\,n)/n|\mathbf{Y}(\tau\,n)/n\right]}{1/n}$$

$$= \frac{\mathbb{E}\left[-Z_k(\tau)\left[\frac{\beta}{n}(1 - \frac{1}{n-nZ_0(\tau)}(1 - (1-p)^{n-nZ_0(\tau)}))\right]\right]}{1/n}$$

$$+ \frac{\mathbb{E}\left[-Z_k(\tau)(1 - \frac{\beta}{n})\frac{1}{n-nZ_0(\tau)}(1 - (1-p)^{n-nZ_0(\tau)}))\right]}{1/n}$$

$$+ \frac{\mathbb{E}\left[Z_{k-1}(\tau)\frac{\beta}{n}(1 - \frac{1}{n-nZ_0(\tau)}(1 - (1-p)^{n-nZ_0(\tau)}))\right]}{1/n}$$

$$+ \frac{\mathbb{E}\left[Z_{k+1}(\tau)(1 - \frac{\beta}{n})\frac{1}{n-nZ_0(\tau)}(1 - (1-p)^{n-nZ_0(\tau)}))\right]}{1/n}.$$

when $n \to +\infty$ we get,

$$\dot{z}_k(\tau) = (z_{k-1}(\tau) - z_k(\tau))\beta + (z_{k+1}(\tau) - z_k(\tau))\frac{1 - e^{-a+az_0(\tau)}}{1 - z_0(\tau)}.$$

Applying the Wormald theorem [97, 98], with the domain D defined by $-\epsilon < \tau < \frac{T}{n} + \epsilon$, $-\epsilon < z_k < 1 + \epsilon$, for $\epsilon > 0$. And taking $\beta = 1$ for the boundeness hypothesis (see lemma 19), $\Lambda_1 = a/(e\,n)$ for the trend hypothesis (see lemma 18). The Lipschitz hypothesis is satisfied with Lipschitz constant $L = (\beta + a)(1 + \epsilon)$ (see lemma 17). Setting $\lambda = a\,n^{-1/4}$, the Wormald theorem gives with probability $1 - \mathcal{O}\left(n^{1/4}\exp(-a^3 n^{1/4})\right)$,

$$Y_k(T) = nz_k(T/n) + \mathcal{O}(n^{3/4}) \quad \text{for } k \geq 0,$$

with $(z_0, \ldots, z_K)$ the solution of the following system,

$$\begin{cases} \dot{z}_0(\tau) = -z_0(\tau)\beta + \frac{z_1(\tau)}{1-z_0(\tau)}(1 - e^{-a+az_0(\tau)}) & \text{for } k = 0, \\ \dot{z}_k(\tau) = (z_{k-1}(\tau) - z_k(\tau))\beta + (z_{k+1}(\tau) - z_k(\tau))\frac{1-e^{-a+az_0(\tau)}}{1-z_0(\tau)} & \text{for } 1 \leq k \leq K-1, \\ \dot{z}_k(\tau) = \beta\,z_{k-1}(\tau) - z_k(\tau)\frac{1-e^{-a(1-z_0(\tau))}}{1-z_0(\tau)} & \text{for } k = K, \\ \sum_{k=0}^{K} z_k(\tau) = 1. \end{cases}$$

$\square$

Now we have all the tools to prove theorem 19.

Proof. For $\frac{1}{n} \leq \tau \leq \frac{T}{n}$, let us consider the normalized random variable $H(\tau) =$

$\frac{\texttt{Greedy}(G,\tau\,n)}{n}$, the conditional expectation of the one-step change of $H(\tau)$ is given by,

$$\frac{\mathbb{E}\left[H\left(\tau+\frac{1}{n}\right)-H(\tau)\big|H(\tau)\right]}{1/n} =$$

$$\frac{\mathbb{E}\left[\texttt{Greedy}(G,\tau\,n+1)/n-\texttt{Greedy}(G,\tau\,n)/n|\texttt{Greedy}(G,\tau\,n)/n\right]}{1/n}$$

$$= 1-(1-\frac{a}{n})^{n-n\,Z_0(\tau)}.$$

when $n\to+\infty$ we get,

$$\dot{h}(\tau) = 1 - e^{-a(1-z_0(\tau))}.$$

Applying Wormald theorem [97, 98], we choose the domain D defined by $-\epsilon < \tau < \frac{T}{n}+\epsilon$, $-\epsilon < z_0 < 1+\epsilon$ for $\epsilon > 0$. We have $\beta' = 1$ for the boundedness hypothesis (see lemma 19), $\delta = a/(e\,n)$ for the trend hypothesis (lemma 18). The Lipschitz hypothesis is satisfied with Lipschitz constant $L' = ae^{a\epsilon}$. Setting $\lambda = a\,n^{-1/4}$, the Wormald theorem gives with probability $1 - \mathcal{O}\left(n^{1/4}\exp(-a^3 n^{1/4})\right)$,

$$\texttt{Greedy}(G,T) = nh(T/n) + \mathcal{O}(n^{3/4})$$

where $h(\tau)$ is solution of the following equation,

$$\dot{h}(\tau) = 1 - e^{-a(1-z_0(\tau))}, \qquad 1/n \le \tau \le T/n$$

and $z_0(\tau)$ as defined in the following system,

$$\begin{cases} \dot{z}_0(\tau) = -z_0(\tau)\beta + \frac{z_1(\tau)}{1-z_0(\tau)}\left(1-e^{-a+az_0(\tau)}\right) & \text{for } k=0 \\ \dot{z}_k(\tau) = (z_{k-1}(\tau)-z_k(\tau))\beta + (z_{k+1}(\tau)-z_k(\tau))\frac{1-e^{-a+az_0(\tau)}}{1-z_0(\tau)} & \text{for } 1 \le k \le K-1 \\ \dot{z}_k(\tau) = \beta\,z_{k-1}(\tau) - z_k(\tau)\frac{1-e^{-a(1-z_0(\tau))}}{1-z_0(\tau)} & \text{for } k=K \\ \sum_{k=0}^{K} z_k(\tau) = 1 \end{cases}$$

Since $\texttt{Greedy}(G,T)$ is bounded and thus uniformly integrable, so convergence in probability implies convergence in mean:

$$\lim_{n\to+\infty} \frac{\mathbb{E}[\texttt{Greedy}(G,T)]}{n} = h(T/n).$$

$\square$

The next theorem applies an improved version of the Wormald theorem (see [95]) on $\texttt{Greedy}(G,T)$,

Theorem 22. *With probability at least $1 - 2e^{-a^2 n^{\frac{3}{2}}/8T}$ we have,*

$$\max_{1\le t\le T} |\texttt{Greedy}(G,t) - nh(t/n)| \le 3e^{L'T/n}an^{3/4}.$$

with $L' = ae^{a\epsilon}$ and $\epsilon > 0$, here $h(\tau)$ is solution of the following equation,

$$\dot{h}(\tau) = 1 - e^{-a(1-z_0(\tau))} \qquad 1/n \le \tau \le T/n.$$

and $z_0(\tau)$ is defined by the following system,

$$\begin{cases} \dot{z}_0(\tau) = -z_0(\tau)\beta + \frac{z_1(\tau)}{1-z_0(\tau)}(1 - e^{-a+az_0(\tau)}) & \text{for } k = 0, \\ \dot{z}_k(\tau) = (z_{k-1}(\tau) - z_k(\tau))\beta + (z_{k+1}(\tau) - z_k(\tau))\frac{1-e^{-a+az_0(\tau)}}{1-z_0(\tau)} & \text{for } 1 \le k \le K-1, \\ \dot{z}_k(\tau) = \beta\, z_{k-1}(\tau) - z_k(\tau)\frac{1-e^{-a(1-z_0(\tau))}}{1-z_0(\tau)} & \text{for } k = K, \\ \sum_{k=0}^{K} z_k(\tau) = 1. \end{cases}$$

$$(7.56)$$

Proof. Using the normalized random variable $H(\tau) = \frac{\texttt{Greedy}(G, \tau\, n)}{n}$ with $\frac{1}{n} \le \tau \le \frac{T}{n}$, let us compute the conditional expectation of the one-step change of $H(\tau)$,

$$\frac{\mathbb{E}\left[H\left(\tau + \frac{1}{n}\right) - H(\tau)\big|H(\tau)\right]}{1/n}$$

$$= \frac{\mathbb{E}\left[\texttt{Greedy}(G, \tau\, n + 1)/n - \texttt{Greedy}(G, \tau\, n)/n|\texttt{Greedy}(G, \tau\, n)/n\right]}{1/n}$$

$$= 1 - (1 - \frac{a}{n})^{n-n\, Z_0(\tau)}.$$

when $n \to +\infty$ we get,
$$\dot{h}(\tau) = 1 - e^{-a(1-z_0(\tau))}.$$

Applying the non-asymptotic version of the Wormald theorem [95], we choose the domain D defined by $-\epsilon < \tau < \frac{T}{n} + \epsilon$, $-\epsilon < z_0 < 1 + \epsilon$ for $\epsilon > 0$. We have $\beta' = 1$ (lemma 19), $\delta = a/(e\, n)$ for the trend hypothesis(lemma 18). The Lipschitz hypothesis is satisfied with Lipschitz constant $L' = ae^{a\epsilon}$(lemma 17). Setting $\lambda = a\, n^{-1/4}$ we have with probability at least $1 - 2e^{-a^2 n^{\frac{3}{2}}/8T}$,

$$\max_{1 \le t \le T} |\texttt{Greedy}(G, t) - nh(t/n)| \le 3e^{L'T/n}an^{3/4}.$$

$\square$

7.B.2   Proof of corollary 2

Corollary 2. *For $K \ge 1$, with probability at least $1 - 2\exp(-a^2 n^{\frac{3}{2}}/8T)$,*

$$|\texttt{Greedy}(G, T) - nh^*(T/n)| \le o(T),$$

and,

$$\frac{\mathbb{E}[\texttt{Greedy}(G, T)]}{n} - h^*(T/n) \underset{n \to +\infty}{\to} 0,$$

with $h^*(x) = \int_{1/n}^{x}(1 - e^{-a(1-z_0^*)})\mathrm{d}\tau = \left(x - \frac{1}{n}\right)\left(1 - e^{-a(1-z_0^*)}\right)$, and z_0^* is the unique
solution of $\sum_{k=0}^{K} z_0^* \left(\frac{\beta}{g(z_0^*)}\right)^k = 1$ with $g(z_0^*) = \frac{1-e^{-a(1-z_0^*)}}{1-z_0^*}$.

The proof is organized as follows:

1. Finding the stationary solution of eq. (7.55).

2. Proving that the stationary solution is asymptotically stable.

3. Proving that $\mathtt{Greedy}(G, T)$ converges to a function depending on the stationary solution.

The next result gives a general form for the stationary solution of eq. (7.55),

Lemma 21. *For $\frac{1}{n} \leq \tau \leq \frac{T}{n}$, let $\bar{S}_{z_0^*} = (z_0^*, \ldots, z_k^*, \ldots, z_K^*)$ be the stationary solution
of the system,*

$$
\begin{cases}
\dot{z}_0(\tau) = -z_0(\tau)\beta + \frac{z_1(\tau)}{1-z_0(\tau)}(1 - e^{-a+az_0(\tau)}) & \text{for } k = 0, \\
\dot{z}_k(\tau) = (z_{k-1}(\tau) - z_k(\tau))\beta + (z_{k+1}(\tau) - z_k(\tau))\frac{1-e^{-a+az_0(\tau)}}{1-z_0(\tau)} & \text{for } 1 \leq k \leq K-1, \\
\dot{z}_k(\tau) = \beta z_{k-1}(\tau) - z_k(\tau)\frac{1-e^{-a(1-z_0(\tau))}}{1-z_0(\tau)} & \text{for } k = K, \\
\sum_{k=0}^{K} z_k(\tau) = 1.
\end{cases}
$$

(7.57)

$\bar{S}_{z_0^*}$ *is unique and satisfies,*

$$
z_k^* = z_0^* \left(\frac{\beta}{g(z_0^*)}\right)^k \quad \text{for } 0 \leq k \leq K, \tag{7.58}
$$

where $g(z_0^*) = \frac{1-e^{-a(1-z_0^*)}}{1-z_0^*}$.

Proof. eq. (7.58) is proved by recurrence. For the uniqueness, according to eq. (7.57),
$\bar{S}_{z_0^*}$ satisfies $\sum_{k=1}^{K} z_k^* = 1$, using eq. (7.58) we get that $P(z_0^*) = \sum_{k=1}^{K} z_0^* \left(\frac{\beta}{g(z_0^*)}\right)^k - 1$.
$P(0) = -1$ and $\lim_{z_0 \to 1} P(z_0) > 0$. Moreover P is continuous and monotonic, this
implies that $P(z_0) = 0$ has a unique solution. Thus, $\bar{S}_{z_0^*}$ is unique.

$\square$

Remark 3. *Given that z_k^* follows a geometric progression, for convergence, it's
essential that $\left|\frac{\beta}{g(z_0^*)}\right| \leq 1$. Therefore, we will proceed with the remaining proofs under
this assumption.*

The following lemma shows that $\bar{S}_{z_0^*}$ is an asymptotically stable stationary solution of eq. (7.57).

Theorem 23. $\bar{S}_{z_0^*}$ *is a an asymptotically stable stationary solution of eq. (7.57).*

Proof. Let $Z = \begin{pmatrix} z_0(t) \\ \vdots \\ z_K(t) \end{pmatrix}$, eq. (7.55) can be seen as $\dot{Z} = F(Z)$, where,

$$F(z_0(t), \ldots, z_K(t)) = \left(-\beta z_0(t) + z_1(t)g(z_0(t)), \ldots, \beta z_{k-1}(t) - z_k(t)\frac{1 - e^{-a(1-z_0(t))}}{1 - z_0(t)} \right).$$

The Jacobian of F at $\bar{S}_{z_0^*}$ is then given by,

$$DF(\bar{S}_{z_0^*}) =$$
$$\begin{pmatrix} -\beta + z_1^* g'(z_0^*) & g(z_0^*) & 0 & \ldots & \ldots & 0 \\ \beta + (z_2^* - z_1^*)g'(z_0^*) & -\beta - g(z_0^*) & g(z_0^*) & 0 & \ldots & 0 \\ (z_3^* - z_2^*)g'(z_0^*) & \beta & -\beta - g(z_0^*) & g(z_0^*) & 0.. & 0 \\ \vdots & 0 & \beta & -\beta - g(z_0^*) & g(z_0^*) & ..0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -z_K^* g'(z_0^*) & \ldots & \ldots & 0 & \beta & -g(z_0^*) \end{pmatrix}.$$

Since proving that $\bar{S}_{z_0^*}$ is asymptotically stable is equivalent to proving that the eigenvalues of $DF(\bar{S}_{z_0^*})$ are non-positives [94]. We achieve this using the perturbation method. To do so, we shall write,

$$DF(\bar{S}_{z_0^*}) = M + uv^\top.$$

where $v^\top = (1, 0, \ldots, 0)$ and,

$$u^\top = g'(z_0^*)(z_1^*, z_2^* - z_1^*, \ldots, -z_K^*)$$
$$= z_1^* g'(z_0^*) \left(1, \frac{\beta}{g(z_0^*)} - 1, (\frac{\beta}{g(z_0^*)} - 1)\frac{\beta}{g(z_0^*)}, (\frac{\beta}{g(z_0^*)} - 1)\left(\frac{\beta}{g(z_0^*)}\right)^2, \ldots \right).$$

and M is the matrix with $g(z_0^*)$ above the diagonal, β below it and its diagonal is $(-\beta, -\beta - g(z_0^*), \ldots, -\beta - g(z_0^*), -g(z_0^*))$.

$$M = \begin{pmatrix} -\beta & g(z_0^*) & 0 & \ldots & \ldots & 0 \\ \beta & -\beta - g(z_0^*) & g(z_0^*) & 0 & \ldots & 0 \\ 0 & \beta & -\beta - g(z_0^*) & g(z_0^*) & 0.. & 0 \\ \vdots & 0 & \beta & -\beta - g(z_0^*) & g(z_0^*) & ..0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & \ldots & 0 & \beta & -g(z_0^*) \end{pmatrix}.$$

Let us denote by $\Pi_M(\lambda)$ the characteristic polynomial of M, as a function of λ, so that

$$\Pi_{M+uv^\top}(\lambda) = \Pi_M(\lambda) \left(1 + v^\top (M - \lambda I)^{-1} u \right).$$

which implies that eigenvalues of $M + uv^\top$ are either eigenvalues of M or solutions of

$$1 + v^\top (M - \lambda I)^{-1} u = 0.$$

Since 0 is an eigenvalue of $M + uv^\top$, we aim at proving that $1 + v^\top (M - \lambda I)^{-1} u = 0$ has K non-positives solutions.

We now claim that the eigenvalues of M are $\mu_j := -\beta - g(z_0^*) + 2\sqrt{\beta g(z_0^*)} \cos(\frac{j\pi}{k+1})$ for $j \in [k]$ and 0. This is a consequence of standard computations along with Theorem 2.2 of [70]. We also denote by ω_j the eigenvectors of M associated to μ_j (and ω_0 to 0) and by P the matrix whose columns are $\omega_0, \omega_1, \dots$. As a consequence,

$$q(\lambda) = 1 + v^\top (M - \lambda I)^{-1} u = 1 + v^\top P \mathrm{diag}\left(\frac{1}{\mu_j - \lambda}\right) P^{-1} u$$

$$= 1 + \left(\frac{\omega_{0,1}}{-\lambda}, \dots, \frac{\omega_{j,1}}{\mu_j - \lambda}\right) P^{-1} u.$$

Since we can take any eigenvectors in P, we can assume that $\omega_{j,1} \geq 0$ hence it remains to prove that $P^{-1} u$ is a vector with non-negative coordinates. Notice that this vector is the vector of u in the basis formed by the eigenvectors of M.

The computations of ω_j are quite standard, and they yield, denoting $\theta = \sqrt{\frac{\beta}{g(z_0^*)}}$, for $1 \leq j \leq K$,

$$\omega_j = \left(\theta \sin(\frac{j\pi}{K+1}), \dots, (-\theta)^{t+1} \sin(\frac{(t+1)j\pi}{K+1}) + (-\theta)^t \sin(\frac{tj\pi}{K+1}), \dots, \right.$$
$$\left. (-\theta)^{K+1} \sin(\frac{(K+1)j\pi}{K+1}) + (-\theta)^K \sin(\frac{tK\pi}{K+1}) \right).$$

As a consequence, u and all the eigenvectors ω_j are orthogonal to the vectors of ones, which indicates that $u = \sum_j \alpha_j \omega_j$ for some scalar α_j. The objective is to prove that they are all positive

The exact forms of u and ω_j give, after a few lines of algebra,

$$\sum_{j=1}^{K} \alpha_j \sin(m\frac{j\pi}{K+1}) = (-\theta)^m, \qquad \forall m \in [K].$$

This system can be rewritten using the Chebyshev polynomials of second kind (denoted by U_n) as,

$$\sum_{j=1}^{K} \alpha_j U_{m-1}(\cos(\frac{j\pi}{K+1})) \sin(\frac{j\pi}{K+1}) = (-\theta)^m, \qquad \forall m \in [K].$$

Hence in a more compact matrix way it can be seen as,

$$W \begin{pmatrix} \alpha_1 \sin(\frac{\pi}{K+1}) \\ \vdots \\ \alpha_K \sin(\frac{K\pi}{K+1}) \end{pmatrix} = -\vec{\theta},$$

where $-\vec{\theta} = \left((-\theta)^j \right)_{j \in [K]}$ and W is the matrix whose j-th column is

$$(U_0(\cos(\frac{j\pi}{K+1})), \dots, U_{K-1}(\frac{j\pi}{K+1}))^\top.$$

We introduce now the following polynomial,

$$P_m(X) = \gamma_m \frac{U_k(X)}{X - \cos(\frac{m\pi}{K+1})} = \sum_{j=1}^{K} \beta_{j,m} U_{j-1}(X)$$

where

$$\gamma_m = \frac{1}{\Pi_{j \neq m} (\cos(\frac{m\pi}{K+1}) - \cos(\frac{j\pi}{K+1}))2^K} = \frac{1}{2^K} \frac{1}{\Pi_{j \neq m} - 2\sin(\frac{m+j}{2}\frac{\pi}{K+1})\sin(\frac{m-j}{2}\frac{\pi}{K+1})}.$$

so that the sign of γ_m is $(-1)^{m-1}$, $P_m(\cos(\frac{j\pi}{K+1})) = 0$ for all $j \neq m$, and $P_m(\cos(\frac{m\pi}{K+1})) = 1$. We get that

$$\alpha_m = \frac{1}{\sin(\frac{m\pi}{K+1})} \sum_{j=1}^{K} \beta_{j,m}(-\theta)^j.$$

Using the fact that, by definition of P_m,

$$U_K(X) = \sum_{j=1}^{K} \frac{1}{\gamma_m} \beta_{j,m} U_{j-1}(X)(X - \cos(\frac{m\pi}{K+1})).$$

and the property of Chebyshev polynomial,

$$U_k(X) = 2X U_{k-2}(X) - U_{k-3},$$

we can identify the coefficients β_j that satisfy a linear recurrence of order 2 and are defined by

$$\beta_j = 2\gamma_m \frac{\sin(\frac{(K-j+1)m\pi}{K+1})}{\sin(\frac{m\pi}{K+1})}.$$

It remains to compute $\alpha_m = \sum \beta_j(-\theta)^j$, and standard computations yield that

$$\alpha_m = -2\gamma_m(-1)^m \frac{(1 - \theta^2)}{1 + 2\cos(\frac{m\pi}{K+1})\theta + \theta^2} \geq 0.$$

Thus, we have proved that $P^{-1}u$ is a vector with non-negative coordinates.

Consequently, $q(\lambda)$ is an increasing function of λ. As a result, $M + uv^\top$ has K eigenvalues of negative real part and one eigenvalue equals to zero. From this, we can conclude that $\bar{S}_{z_0^*}$ is an asymptotically stationary solution of eq. (7.57).

$\square$

7

Given the previous results, we can prove corollary 2,

Proof. Let $h^*(T/n) = \int_{1/n}^{T/n}(1 - e^{-a(1-z_0^*)})\mathrm{d}\tau = \frac{(T-1)(1-e^{-a(1-z_0^*)})}{n}$, we have,

$$
\begin{aligned}
&|\texttt{Greedy}(G,T) - nh^*(T/n)| \\
&= |\texttt{Greedy}(G,T) - nh(T/n) + nh(T/n) - nh^*(T/n)| \\
&\leq \underbrace{\max_{1 \leq t \leq T}\left(|\texttt{Greedy}(G,t) - nh(t/n)|\right.}_{\leq 3e^{L'T/n}an^{3/4} \text{ by theorem 22}} + |nh(T/n) - nh^*(T/n)|).
\end{aligned}
$$

Let's focus on $D = |nh(T/n) - nh^*(T/n)|$, for $1 \leq T' < T$ and $\delta > 0$,

$$
\begin{aligned}
D &= |nh(T/n) - nh(T'/n + \delta) + nh(T'/n + \delta) - nh^*(T/n)| \\
&= |nh(T/n) - nh(T'/n + \delta) - nh^*(T/n) + nh^*(T'/n + \delta) + nh(T'/n + \delta) \\
&\quad - nh^*(T'/n + \delta)| \\
&= \left| n\int_{T'/n+\delta}^{T/n}(e^{-a(1-z_0^*)} - e^{-a(1-z_0(t))})\mathrm{d}t + n\int_{1/n}^{T'/n+\delta}(e^{-a(1-z_0^*)} - e^{-a(1-z_0(t))})\mathrm{d}t \right| \\
&= \left| ne^{-a(1-z_0^*)}\int_{T'/n+\delta}^{T/n}(1 - e^{a(z_0(t)-z_0^*)})\mathrm{d}t + ne^{-a(1-z_0^*)}\int_{1/n}^{T'/n+\delta}(1 - e^{a(z_0(t)-z_0^*)})\mathrm{d}t \right| \\
&\leq ne^{-a(1-z_0^*)}\left| \int_{T'/n+\delta}^{T/n}(-a(z_0(t) - z_0^*))\mathrm{d}t + \int_{1/n}^{T'/n+\delta}(-a(z_0(t) - z_0^*))\mathrm{d}t \right|
\end{aligned}
$$

using $(1 - e^{ax} \leq -ax)$

$$
\leq ne^{-a(1-z_0^*)}a\int_{T'/n+\delta}^{T/n}|z_0(t) - z_0^*|\mathrm{d}t + ne^{-a(1-z_0^*)}a\int_{1/n}^{T'/n+\delta}|z_0(t) - z_0^*|\mathrm{d}t
$$

$$\leq ne^{-a(1-z_0^*)}a \int_{T'/n+\delta}^{T/n} \epsilon \mathrm{d}t + ne^{-a(1-z_0^*)}a \int_{1/n}^{T'/n+\delta} |z_0(t) - z_0^*| \mathrm{d}t$$

using $(theorem\ 23)$

$$\leq ne^{-a(1-z_0^*)}a\epsilon \left(\frac{T - T'}{n} - \delta \right) + ne^{-a(1-z_0^*)}a \int_{1/n}^{T'/n+\delta} 1 \mathrm{d}t$$

using $0 \leq z_0 \leq 1$ and $0 \leq z_0^* \leq 1$

$$\leq ne^{-a(1-z_0^*)}a\epsilon \left(\frac{T}{n} \right) + ne^{-a(1-z_0^*)}a \left(\frac{T' - 1}{n} + \delta \right)(1 - \epsilon)$$

$$\leq ne^{-a(1-z_0^*)}a\epsilon \left(\frac{T}{n} \right) + ne^{-a(1-z_0^*)}a \underbrace{\left(\frac{T' - 1}{n} + \delta \right)}_{\leq \frac{T}{n^2}}$$

$$\leq 2e^{-a(1-z_0^*)}a \left(\frac{T}{n} \right) \qquad \text{choosing } (\epsilon = \frac{1}{n}).$$

Thus,

$$|\texttt{Greedy}(G, T) - nh^*(T/n)| \leq 3e^{L'T/n}an^{3/4} + 2e^{-a(1-z_0^*)}a\frac{T}{n}.$$

with $L' = ae^{a\gamma}$ where $\gamma > 0$. Taking $n = cT$ with $c < 1$, we can see that $|\texttt{Greedy}(G, T) - nh^*(T/n)| \leq o(T)$,

$\square$

7.B.3 Proof of Corollary 3

Corollary 3. *For $K = 1$, with probability at least $1 - 2\exp(-a^2 n^{\frac{3}{2}}/8T)$,*

$$\left| \mathbb{E}[\texttt{Greedy}(G, T)] - T(1 - e^{-a(1-z_0^*)}) \right| \leq c\frac{T}{(\log(T))^{3/4}} = o(T).$$

where $z_0^* = \frac{1}{\beta} - \frac{1}{a}W\left(\frac{a}{\beta}e^{-a\left(1 - \frac{1}{\beta}\right)} \right)$, with $W(\cdot)$ the Lambert function, and c is some universal constant.

The proof is organized as follows:

1. Finding the stationary solution of eq. (7.55) for $K = 1$.

2. Proving that the stationary solution is exponentially stable.

3. Applying an improved version of the Wormald theorem on $\texttt{Greedy}(G, T)$.

4. Proving that $\texttt{Greedy}(G, T)$ converges to a function depending on the stationary solution.

Intuitively $K = 1$ means that the maximum budget reached by each node in U is equal to 1. From a technical aspect, supposing $K = 1$ reduces eq. (7.57) to a system of 2 equations as follows, for $t \in [\frac{1}{n}, \frac{T}{n}]$

$$\begin{cases} \dot{z}_0(t) & = -\beta\, z_0(t) + \frac{z_1(t)}{1-z_0(t)}(1 - e^{-a(1-z_0(t))}), \\ \dot{z}_1(t) & = \beta z_0(t) - \frac{z_1(t)}{1-z_0(t)}(1 - e^{-a(1-z_0(t))}), \\ z_0(t) + z_1(t) & = 1. \end{cases} \qquad (7.59)$$

By simplifying eq. (7.59), we reduce the system to the following equation,

$$\dot{z}_0(t) = -\beta\, z_0(t) + 1 - e^{-a(1-z_0(t))}. \qquad (7.60)$$

The following lemma computes the unique stationary solution of eq. (7.60),

Lemma 22. *The stationary solution of eq. (7.60) is unique and is given for $\beta, a > 0$ by,*

$$z_0^* = \frac{1}{\beta} - \frac{1}{a}W\left(\frac{a}{\beta}e^{-a\left(1-\frac{1}{\beta}\right)}\right).$$

where W is the Lambert function.

Proof. Let's define $G(z_0) = -\beta\, z_0 + 1 - e^{-a(1-z_0)}$, the stationary solution of eq. (7.60) is the solution of $G(z_0^*) = 0$ (the homogeneous equation) with $a > 0$ and $\beta > 0$,

$$G(z_0^*) = 0 \iff 1 - e^{-a(1-z_0^*)} = \beta z_0^* \iff e^{a\left(\frac{1}{\beta}-z_0^*\right)}a\left(\frac{1}{\beta} - z_0^*\right) = \frac{a}{\beta}e^{-a\left(1-\frac{1}{\beta}\right)}$$

$$\iff a\left(\frac{1}{\beta} - z_0^*\right) = W\left(\frac{a}{\beta}e^{-a\left(1-\frac{1}{\beta}\right)}\right).$$

Where W is the Lambert function. So, by rearranging the terms in the last equation, the solution of $G(z_0^*) = 0$ is given by,

$$z_0^* = \frac{1}{\beta} - \frac{1}{a}W\left(\frac{a}{\beta}e^{-a\left(1-\frac{1}{\beta}\right)}\right).$$

Let's prove the uniqueness of the stationary solution, $G(0) = 1 - e^{-a} > 0$ and $G(1) = -\beta < 0$ and we have $\forall\, 0 \leq z_0 \leq 1$, $\frac{dG(z_0)}{dz_0} = -(\beta + ae^{-a}e^{-a(1-z_0)}) < 0$. Thus $G(z_0) = 0$ has a unique solution.

$\square$

The following theorem proves that z_0^* is exponentially stable, meaning that $\forall t \in [\frac{1}{n}, \frac{T}{n}]$, $z_0(t)$ converges to z_0^* with an exponential rate.

Theorem 24. *For any $f_0 \geq 0$ and $t \in [\frac{1}{n}, \frac{T}{n}]$, consider the ordinary differential equation (ODE),*

$$\begin{cases} \dot{z}_0(t) & = \text{-}\beta z_0(t) + 1 - e^{-a(1-z_0(t))}, \\ z_0(1/n) & = f_0. \end{cases}$$

Thus, it implies that $z_0(t)$ converges to z_0^* exponentially.

Proof. The idea here is to prove that for any perturbation that we add to z_0^*, this perturbation tends to 0 when t tends to $+\infty$ exponentially.

Let's consider $\epsilon(t) : \mathbb{R} \to \mathbb{R}$, a pertubation of the stationary solution z_0^*,

$$\dot{\epsilon}(t) = -\beta \, z_0^* - \beta \, \epsilon(t) + 1 - e^{-a(1-z_0^*(t)-\epsilon(t))}$$
$$\dot{\epsilon}(t) = -1 + e^{-a(1-z_0^*)} - \beta \, \epsilon(t) + 1 - e^{-a(1-z_0^*-\epsilon(t))}$$
$$\dot{\epsilon}(t) = e^{-a(1-z_0^*)} \left(1 - e^{a\epsilon(t)}\right) - \beta \, \epsilon(t)$$
$$\dot{\epsilon}(t) \leq -a\epsilon(t) \, e^{-a(1-z_0^*)} - \beta \, \epsilon(t) \qquad \text{using that } \ (1 - e^{a\epsilon(t)} \leq -a\epsilon(t))$$
$$\dot{\epsilon}(t) \leq \epsilon(t) \underbrace{\left(-a \, e^{-a(1-z_0^*)} - \beta\right)}_{\leq 0}.$$

Integrating the last equation , we get,

$$\ln\left(|\epsilon(t)|\right) - \ln(|\epsilon(0)|) \leq \left(-a \, e^{-a(1-z_0^*)} - \beta\right) t \tag{7.61}$$
$$|\epsilon(t)| \leq |\epsilon(0)| \, \exp\left(-t \left(a \, e^{-a(1-z_0^*)} + \beta\right)\right) \tag{7.62}$$
$$|\epsilon(t)| \leq |f_0 - z_0^*| \exp\left(-t \left(a e^{-a\left(1-\frac{1}{\beta}+\frac{1}{a}W\left(\frac{a}{\beta}e^{-a(1-\frac{1}{\beta})}\right)\right)} + \beta\right)\right) \tag{7.63}$$
$$|\epsilon(t)| \leq |f_0 - z_0^*| \exp\left(-t\beta \left(1 + W(e^{-a(1-\frac{1}{\beta})})\right)\right). \tag{7.64}$$

Moving from eq. (7.63) to eq. (7.64) is done using $\exp(-W(x)) = W(x)/x$.

Thus $\lim\limits_{t\to+\infty} \epsilon(t) = 0$ exponentially with the following rate $\omega = \beta \left(1 + W(e^{-a(1-\frac{1}{\beta})})\right)$.

$\square$

Lemma 23. $S_{z_0^*}^1 = (z_0^*, z_1^*) = (z_0^*, z_0^* \frac{\beta}{g(z_0^*)})$ *is an exponentially stable stationary solution of eq.* (7.59).

Proof. According to lemma 22, z_0^* is a stationary solution of eq. (7.60), this implies that $S_{z_0^*}^1 = (z_0^*, z_1^*) = (z_0^*, z_0^* \frac{\beta}{g(z_0^*)})$ is a stationary solution of eq. (7.59). As previously demonstrated, $\forall t \in [\frac{1}{n}, \frac{T}{n}]$, $z_0(t)$ converges to z_0^* exponentially. This implies that eq. (7.60) possesses an exponentially stable stationary solution. Given that eq. (7.60) is a reduced version of eq. (7.59), we can conclude that $S_{z_0^*}^1$ is an exponentially stable stationary solution for eq. (7.59). $\square$

With all the essential elements assembled, we are now ready to establish the proof for corollary 3

Proof. Let $h^*(T/n) = \int_{1/n}^{T/n}(1 - e^{-a(1-z_0^*)})\mathrm{d}\tau = \frac{(T-1)(1-e^{-a(1-z_0^*)})}{n}$, we have,

$$|\mathtt{Greedy}(G, T) - nh^*(T/n)| = |\mathtt{Greedy}(G, T) - nh(T/n) + nh(T/n)|$$

$$\leq \underbrace{\max_{1\leq t\leq T}(|\mathtt{Greedy}(G, t) - nh(t/n)|)}_{\leq 3e^{L'T/n}an^{3/4} \text{ by theorem 22}} + |nh(T/n) - nh^*(T/n)|$$

Let's focus on $A = |nh(T/n) - nh^*(T/n)|$,

$$A = \left| n\int_{1/n}^{T/n}(1 - e^{-a(1-z_0(\tau))})\mathrm{d}\tau - n\int_{1/n}^{T/n}(1 - e^{-a(1-z_0^*)})\mathrm{d}\tau \right| \tag{7.65}$$

$$= \left| n\int_{1/n}^{T/n}(e^{-a(1-z_0^*)} - e^{-a(1-z_0(\tau))})\mathrm{d}\tau \right| \tag{7.66}$$

$$= \left| ne^{-a(1-z_0^*)}\int_{1/n}^{T/n}(1 - e^{a(z_0(\tau)-z_0^*)})\mathrm{d}\tau \right| \tag{7.67}$$

$$\leq \left| ne^{-a(1-z_0^*)}\int_{1/n}^{T/n} -a(z_0(\tau) - z_0^*)\mathrm{d}\tau \right| \quad (\text{using } 1 - e^{ax} \leq -ax) \tag{7.68}$$

$$\leq nae^{-a(1-z_0^*)}\int_{1/n}^{T/n}|z_0(\tau) - z_0^*|\mathrm{d}\tau \tag{7.69}$$

$$\leq nae^{-a(1-z_0^*)}|f_0 - z_0^*|\int_{1/n}^{T/n}\exp\left(-\tau\beta\left(1 + W(e^{-a(1-\frac{1}{\beta})})\right)\right)\mathrm{d}\tau \tag{7.70}$$

$$\leq \frac{nae^{-a(1-z_0^*)}|f_0 - z_0^*|}{\beta\left(1 + W(e^{-a(1-\frac{1}{\beta})})\right)}\left(e^{-\frac{\beta}{n}\left(1 + W(e^{-a(1-\frac{1}{\beta})})\right)} - e^{-\frac{T}{n}\beta\left(1 + W(e^{-a(1-\frac{1}{\beta})})\right)}\right). \tag{7.71}$$

Moving from eq. (7.69) to eq. (7.70) is done using theorem 24. Thus we have,

$$|\mathtt{Greedy}(G, T) - nh^*(T/n)| \leq 3e^{L'T/n}an^{3/4} + \frac{nae^{-a(1-z_0^*)}|f_0 - z_0^*|}{P}\left(e^{-\frac{1}{n}P} - e^{-\frac{T}{n}P}\right)$$

with $L' = ae^{a\epsilon}$ and $P = \beta\left(1 + W(e^{-a(1-\frac{1}{\beta})})\right)$, $\epsilon > 0$.

Now let us focus on $A = 3e^{L'T/n}an^{3/4} + \frac{nae^{-a(1-z_0^*)}|f_0-z_0^*|}{P}\left(e^{-\frac{1}{n}P} - e^{-\frac{T}{n}P}\right)$ and

considering that $n = \frac{T}{\alpha \log(T)}$ with $\alpha > 0$, we get,

$$
\begin{aligned}
A &\leq 3e^{L'T/n}an^{3/4} + \frac{nae^{-a(1-z_0^*)}|f_0 - z_0^*|}{P}\left(1 - e^{-\frac{T}{n}P}\right)\\
&= \frac{ae^{-a(1-z_0^*)}|f_0 - z_0^*|}{P}\left(\frac{T}{\alpha \log(T)} - \frac{T^{1-\alpha P}}{\alpha \log(T)}\right)\\
&\quad + 3a\frac{T^{\frac{3}{4}+\alpha L'}}{(\alpha \log(T))^{\frac{3}{4}}}\\
&\leq \frac{ae^{-a(1-z_0^*)}|f_0 - z_0^*|}{P}\frac{T}{\alpha \log(T)} + 3a\frac{T^{\frac{3}{4}+\alpha L'}}{(\alpha \log(T))^{\frac{3}{4}}}.
\end{aligned}
$$

Taking $\alpha = \frac{1}{4(P+L')}$, with $z_0^* = \frac{1}{\beta} - \frac{1}{a}W\left(\frac{a}{\beta}e^{-a(1-\frac{1}{\beta})}\right)$ and using the fact that $e^{-W(x)} = \frac{W(x)}{x}$, we get,

$$
\begin{aligned}
A &\leq \frac{\beta W\left(e^{-a(1-\frac{1}{\beta})}\right)}{\alpha P}|f_0 - z_0^*|\frac{T}{\log(T)} + 3a(4(L'+P))^{\frac{3}{4}}\frac{T^{\frac{3L'+4P}{4(P+L')}}}{(\log(T))^{\frac{3}{4}}}\\
&= c_1\frac{T}{\log(T)} + c_2\frac{T^{\omega'}}{(\log(T))^{\frac{3}{4}}}\\
&\leq c\frac{T}{(\log(T))^{3/4}}.
\end{aligned}
$$

where $c_1 = \frac{4\beta(P+L')W\left(e^{-a(1-\frac{1}{\beta})}\right)}{P}$, $c_2 = 3a(4(L'+P))^{\frac{3}{4}}$, $c = c_1 + c_2$, $\omega = \frac{3P+4L'}{4(L'+P)}$ and $\omega' = \frac{3L'+4P}{4(P+L')}$.

$\square$

7.B.4 Proof of proposition 3

Proposition 3. *For $T, K, n, b_0, \beta \in \mathbb{N}^*$,*

$$
\begin{aligned}
\mathrm{CR}^{\mathrm{sto}}(\mathtt{Greedy}, \mathcal{D}) &\geq \frac{Tg(z_0^*)(1-z_0^*) + nb_0 - n\left(\frac{\beta}{g(z_0^*)-\beta} - \frac{(K+1)\beta^{K+1}}{g(z_0^*)^{K+1}-\beta^{K+1}}\right)}{nb_0 + \beta T}\\
&\quad + \mathcal{O}_{K,\beta}(T^{-1/4}).
\end{aligned}
$$

where $\sum_{k=0}^{K} z_0^*\left(\frac{\beta}{g(z_0^*)}\right)^k = 1$ with $g(z_0^*) = \frac{1-e^{-a(1-z_0^*)}}{1-z_0^*}$ as defined in corollary 2.

Proof. According to eq. (7.45), we have that for all $u \in U$,

$$
b_{u,t} = \min(K, b_{u,t-1} - x_{u,t} + \eta_t), \qquad \text{with } b_{u,0} = b_0 \geq 1.
$$

Which gives,

$$\texttt{Greedy}(G, T) = \sum_{u \in U} \sum_{t=1}^{T} x_{u,t}$$

$$= nb_0 + \underbrace{\sum_{u \in U} \sum_{t=1}^{T} \mathbb{E}[\eta_t]}_{A_1} - \underbrace{\sum_{u \in U} b_{u,T}}_{A_2} - \underbrace{\sum_{u \in U} \sum_{t=1}^{T} \mathbb{E}[\mathbb{1}_{\{b_{u,t}=K\}} \mathbb{1}_{\{\eta_t=1\}}]}_{A_3}.$$

According to lemma 20, we have w.h.p $\forall k \geq 0, t \in [T], Y_k(t) = nz_k(t/n) + \mathcal{O}(n^{3/4})$, let us then compute A_1, A_2 and A_3,

$$A_1 = \sum_{u \in U} \sum_{t=1}^{T} \mathbb{E}[\eta_t] = \beta\, T,$$

$$A_2 = \sum_{u \in U} b_{u,T} = n \sum_{k=1}^{K} k z_k(T/n) + \mathcal{O}\left(\frac{K(K+1)}{2} n^{3/4}\right),$$

$$A_3 = \beta \sum_{t=1}^{T} z_K(t/n) + \mathcal{O}(\beta T n^{-1/4}).$$

Using the following upper bound on $\texttt{OPT}(G, T) \leq nb_0 + \beta T$ and $n = \mathcal{O}(T)$, we get that,

$$\texttt{CR}^{\text{sto}}(\texttt{Greedy}, \mathcal{D}) \geq \frac{nb_0 + \beta T - n \sum_{k=1}^{K} k z_k(T/n) - \beta \sum_{t=1}^{T} z_K(t/n)}{nb_0 + \beta T} + \mathcal{O}(T^{-1/4}).$$

According theorem 23, $\forall \tau \in [\frac{1}{n}, \frac{T}{n}], (z_0(\tau), \dots, z_K(\tau))$ converges to $\bar{S}_{z_0^*}$ asymptotically, this implies that,

$$\texttt{CR}^{\text{sto}}(\texttt{Greedy}, \mathcal{D}) \geq \frac{nb_0 + \beta T - \beta T z_0^* \left(\frac{\beta}{g(z_0^*)}\right)^K - nz_0^* \sum_{k=1}^{K} k \left(\frac{\beta}{g(z_0^*)}\right)^k}{nb_0 + \beta T} + \mathcal{O}(T^{-1/4}) \tag{7.72}$$

From $\sum_{k=0}^{K} z_0^* \left(\frac{\beta}{g(z_0^*)}\right)^k = 1$, we have that $\left(\frac{\beta}{g(z_0^*)}\right)^K = \frac{g(z_0^*)}{\beta} - \frac{1}{z_0^*}\left(\frac{g(z_0^*)}{\beta} - 1\right)$, this

gives,

$$\text{CR}^{\text{sto}}(\text{Greedy}, \mathcal{D}) \geq \frac{nb_0 + \beta T - \beta T z_0^* \left(\frac{\beta}{g(z_0^*)}\right)^K - nz_0^* \sum_{k=1}^{K} k \left(\frac{\beta}{g(z_0^*)}\right)^k}{nb_0 + \beta T} + \mathcal{O}(T^{-1/4})$$

$$\geq \frac{nb_0 + \beta T - \beta T z_0^* \left(\frac{g(z_0^*)}{\beta} - \frac{1}{z_0^*}\left(\frac{g(z_0^*)}{\beta} - 1\right)\right) - nz_0^* \sum_{k=1}^{K} k \left(\frac{\beta}{g(z_0^*)}\right)^k}{nb_0 + \beta T}$$

$$+ \mathcal{O}(T^{-1/4})$$

$$\geq \frac{nb_0 + g(z_0^*)T(1 - z_0^*) - nz_0^* \sum_{k=1}^{K} k \left(\frac{\beta}{g(z_0^*)}\right)^k}{nb_0 + \beta T} + \mathcal{O}(T^{-1/4}).$$

Using $1 - \left(\frac{\beta}{g(z_0^*)}\right)^{K+1} = \frac{1}{z_0^*}\left(1 - \frac{\beta}{g(z_0^*)}\right)$ and $\sum_{k=1}^{K} kx^k = x\frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{1-x^{K+1}}{1-x}\right)$ with $x = \frac{\beta}{g(z_0^*)}$,

$$\text{CR}^{\text{sto}}(\text{Greedy}, \mathcal{D}) \geq \frac{nb_0 + g(z_0^*)T(1 - z_0^*) - nz_0^* \sum_{k=1}^{K} k \left(\frac{\beta}{g(z_0^*)}\right)^k}{nb_0 + \beta T} + \mathcal{O}(T^{-1/4})$$

$$\geq \frac{nb_0 + g(z_0^*)T(1 - z_0^*) - n\left(\frac{\beta}{g(z_0^*)-\beta} - \frac{(K+1)\beta^{K+1}}{g(z_0^*)^{K+1}-\beta^{K+1}}\right)}{nb_0 + \beta T} + \mathcal{O}(T^{-1/4}).$$

$$\square$$

7.B.5 Proof of theorem 20

Theorem 20. *For any $\alpha, \beta > 0$, the competitive ratio tends to 1, as T, K, n approach infinity, as*

$$\lim_{K,n\to+\infty} \lim_{T\to+\infty} \text{CR}^{\text{sto}}(\text{Greedy}, \mathcal{D}) = 1.$$

Proof.

$$\lim_{T\to+\infty} \text{CR}^{\text{sto}}(\text{Greedy}, \mathcal{D}) = \frac{g(z_0^*)(1 - z_0^*)}{\beta}.$$

When $K \to \infty$, z_0^* satisfies $\sum_{k=0}^{+\infty} z_0^* \left(\frac{\beta}{g(z_0^*)}\right)^k = 1$, this gives,

$$\sum_{k=0}^{+\infty} z_0^* \left(\frac{\beta}{g(z_0^*)}\right)^k = z_0^* \frac{1}{1 - \frac{\beta}{g(z_0^*)}} = 1 \implies 1 - e^{-a(1-z_0^*)} = \beta,$$

which leads to $z_0^* = 1 + \frac{ln(1-\beta)}{a}$.

Thus,

$$\lim_{K,n\to+\infty} \lim_{T\to+\infty} \mathrm{CR}^{\mathrm{sto}}(\mathtt{Greedy}, \mathcal{D}) = \lim_{K,n\to+\infty} \frac{g(z_0^*)(1-z_0^*)}{\beta} = 1.$$

$\square$

8

Online matching on stochastic block model

8

While online bipartite matching has gained significant attention in recent years, existing analyses in stochastic settings fail to capture the performance of algorithms on heterogeneous graphs, such as those incorporating inter-group affinities or other social network structures. In this work, we address this gap by studying online bipartite matching within the stochastic block model (SBM). A fixed set of offline nodes is matched to a stream of online arrivals, with connections governed probabilistically by latent class memberships. We analyze two natural algorithms: a `Myopic` policy that greedily matches each arrival to the most compatible class, and the `Ex-ante Balance` algorithm, which accounts for both compatibility and remaining capacity. For the `Myopic` algorithm, we prove that the size of the matching converges, with high probability, to the solution of an ordinary differential equation (ODE), for which we provide a tractable approximation along with explicit error bounds. For the `Ex-ante Balance` algorithm, we demonstrate the convergence of the matching size to a differential inclusion and derive an explicit limiting solution. Lastly, we explore the impact of estimating the connection probabilities between classes online, which introduces an exploration–exploitation trade-off.

Contents

8.1   Introduction

Matching in bipartite graphs is a central problem that lies at the crossroads of graph theory [46, 100], network science, and combinatorial optimization [74, 90]. A bipartite graph $G = (U, V, E)$ consists of two disjoint sets of nodes, U and V, together with a set of edges $E \subseteq U \times V$ linking elements across the two sets. Such graphs naturally represent systems in which entities from one group must be paired with entities from another—examples include assigning tasks to agents, matching products with consumers, or displaying ads to users. A *matching* is a subset of edges

with no shared endpoints, guaranteeing that each entity participates in at most one pairing. The key challenge is to compute matchings that are optimal with respect to resource constraints and some objective function, such as maximizing coverage or utility. This problem is of great practical importance, notably in operations research, where it is closely related to the classical assignment problem [48].

Recent real-world applications, particularly in online advertising, ride-hailing platforms, and real-time job allocation, have attracted significant interest in the online version of the matching problem [81]. In this setting, nodes in U (e.g., advertisers or servers) are fixed and known in advance, while nodes in V (e.g., users or requests) arrive sequentially. When a new node $t \in V$ arrives, the algorithm must decide on the spot whether to match it to an available node $u \in U$, such that $(u,t) \in E$, with the constraint that each node can be matched at most once. These decisions are irrevocable, making the problem both practically and theoretically challenging. The central goal is to design algorithms that construct a maximum matching—that is, one that covers as many nodes as possible. Typically, such algorithms are analyzed in one of two ways: by approximating the size of the resulting matching, or by evaluating their competitive ratio, defined as the worst-case ratio between the size of the algorithm's matching and that of an optimal matching computed with full knowledge of the graph in advance.

Online matching has been explored through several theoretical lenses, most notably the adversarial and stochastic frameworks [81]. In the adversarial setting, the graph and arrival sequence are designed to be worst-case, providing robust but conservative performance guarantees. In contrast, stochastic models assume randomness in either the graph structure or the arrival process [24], allowing for more realistic analyses and often stronger guarantees. Within this stochastic line of work, much attention has been given to Erdős–Rényi-type models [79], where edges are included independently with a fixed probability. While such models are analytically tractable and provide valuable insights, they fail to capture complex patterns like community structure, heterogeneity, and group-based interactions observed in real-world systems [53].

To address these limitations, the *stochastic block model* (SBM) has emerged as a powerful alternative [53, 3]. SBM introduces latent classes (or communities) and allows edge probabilities to depend on class membership, thereby capturing structured heterogeneity—such as homophily (the tendency of similar nodes to connect more frequently) [80], core-periphery patterns (where a densely connected "core" group links to many others while "periphery" nodes have fewer connections) [21], or inter-group affinities (specific preferences or tendencies for nodes in one group to connect with nodes in another group) [5]. This makes SBM particularly well-suited for modeling interactions in social networks, recommendation engines, and online marketplaces, where the likelihood of a match depends not just on individual attributes but also on group-level dynamics [65]. In the context of online bipartite matching, this leads naturally to the *bipartite stochastic block model*, where one partition consists of fixed agents and the other of arriving users, both belonging to latent classes that govern connection probabilities [68].

We study online matching in the bipartite stochastic block model, focusing on the sparse regime, where the average degree of each node remains bounded as the system grows. This regime is particularly relevant in practice, as real-world platforms typically feature users or items that interact with only a small subset of the population. Sparsity not only reflects these empirical network structures but also introduces significant analytical challenges. Formally, we consider a bipartite graph $G = (U, V, E)$, where U is a fixed set of nodes and V is a set of nodes that arrive sequentially. Each node $u \in U$ is independently assigned a class $c(u) \in \mathcal{C}$ according to a distribution $\mu(u)$, and each arriving node $t \in V$ is independently assigned a class $d(t) \in \mathcal{D}$ with distribution $\nu(t)$. Conditional on these class assignments, an edge between n and t is present independently with probability $p_{u,t}$, which depends on their respective classes. A defining feature of this model, motivated by real-world constraints, is that the set of edges incident to each arriving node is not known in advance. Instead, when a node $t \in V$ arrives, the algorithm observes only then which edges exist between t and the nodes in U. This assumption captures a key operational reality in many systems, where information about potential interactions is revealed only upon arrival or activation of a new entity. For example, in on-line job platforms, the compatibility between a newly posted job (a node in V) and existing freelancers (nodes in U) becomes clear only when the job description is published. The system cannot precompute all possible matches due to computational constraints and the dynamic, user-driven nature of postings. In recommendation systems, user preferences are inferred from behavior observed at login, and only then can the platform determine which content is relevant — effectively modeling the appearance of edges at the moment of interaction. Given this online nature of information revelation, an algorithm must operate under uncertainty: it observes each vertex in V sequentially and must decide, upon arrival, whether and how to match it to an available node in U, without knowledge of future arrivals or their connections. In this work, we introduce and analyze two natural algorithms designed for this setting:

- `Myopic`: Upon the arrival of a node $t \in V$, the algorithm chooses a compatible class $c^* \in \mathcal{C}$ and attempts to match t with an available node from this class.

- `Ex-ante Balance`: This algorithm selects the class with the highest probability of a successful match, considering both compatibility and current availability.

We focus on the `Myopic` and `Ex-ante Balance` algorithms because they are simple, practical, and reflect decision-making heuristics commonly used in real-world systems. Notably, while the `Myopic` algorithm has been studied in stochastic models, the `Ex-ante Balance` algorithm—despite its widespread use—has only been analyzed in adversarial settings. Its theoretical performance in structured stochastic environments, such as the bipartite stochastic block model, remains unexplored. These algorithms are appealing not only for their practical relevance but also for their interpretability and ease of implementation, which is particularly valuable in

applications like online marketplaces, content recommendation, or allocation systems. We first analyze them under the assumption that the compatibility probabilities $p_{u,t}$ between user and item classes are known. This setting provides a tractable analytical framework, allowing us to rigorously characterize the fluid-limit behavior of both algorithms. While this fully informed setting offers key insights, it often does not reflect the realities faced by many real-world systems. Motivated by these practical constraints, we then turn to the more realistic case where the probabilities $p_{u,t}$ are not known a priori. In many applications—such as recommendation engines or online platforms—the interaction propensities between user and item types must be inferred over time through observed outcomes. This naturally gives rise to a bandit setting, where the algorithm must estimate the unknown affinities $p_{u,t}$ from binary feedback (indicating whether a match succeeded or failed), while simultaneously making irrevocable matching decisions. This introduces an exploration-exploitation trade-off that is absent in the known-parameter regime. To address this challenge, we propose and analyze a bandit version of the `Ex-ante Balance` algorithm, which learns class affinities dynamically while aiming to preserve strong matching performance.

Our two main contributions, correspond to the two settings considered:

- **When $p_{u,t}$ are known:** We provide a fluid-limit analysis of both `Myopic` and `Ex-ante Balance` algorithms in the sparse bipartite stochastic block model. Specifically,

  - We prove that the matching size obtained by the `Myopic` algorithm is, with high probability, close to the solution of a specific ordinary differential equation. Due to the complexity of solving this equation in closed form, we derive a tractable approximation and show that the resulting error remains small.

  - For the `Ex-ante Balance` algorithm, we prove that the matching size converges with high probability to a solution of a differential inclusion—a generalization of ODEs that captures the algorithm's discontinuous decision rules. To our knowledge, this is the first use of differential inclusions to analyze online matching problems.

  - We extend this analysis to a generalized version of `Ex-ante Balance`, where the decision rule incorporates both connection probabilities and real-time availability, and show that its performance similarly converges to a well-defined differential inclusion.

- **When $p_{u,t}$ are unknown:** We study $\mathtt{ETC - balance}$ algorithm a bandit extension of the `Ex-ante Balance` algorithm, where the compatibility probabilities $p_{u,t}$ are unknown and must be learned over time. In this setting, the algorithm receives binary feedback for each matching attempt and must estimate the latent affinities between classes while making sequential, irrevocable decisions. We analyze the regret of this learning-based algorithm and prove that it is of order $\mathcal{O}(T^{\frac{q+3}{4}})$ for $0 < q < 1$ using stochastic approximations and differential inclusions tools.

Related works

Online bipartite matching has been extensively studied, particularly in adversarial and stochastic models ([56, 81] for a survey). In the adversarial setting, the `Greedy` algorithm guarantees a $1/2$ competitive ratio, improving to $1-1/e$ under random arrivals [47]. The `Ranking` algorithm achieves the optimal $1-1/e$ bound in this setting and performs even better with random arrivals [64, 35, 76]. In contrast, stochastic models assume known distributions over vertex types, often in the i.i.d. setting. This allows improved performance, with algorithms reaching competitive ratios up to 0.711 [77, 58, 26, 55]. However, the i.i.d. model overlooks graph structure, and in many practical or average-case settings, simple heuristics can match or outperform these algorithms [23]. This has motivated the study of stochastic input models that better reflect real-world graphs. Consequently, another line of work focuses on applying online algorithms to specific random graph families. A foundational example is online matching in Erdős–Rényi graphs, particularly in the sparse regime where each edge exists independently with probability c/n [79, 22, 37]. Even for simple strategies like `Greedy`, analysis in this setting is nontrivial and yields valuable insights. The configuration model further generalizes this approach by prescribing degree distributions for vertices [85, 1]. Additional generalizations of Erdős–Rényi have introduced dynamic elements—for instance, models where node degrees evolve over time to reflect changing environments or behaviors [29]. The stochastic block model (SBM), a structured extension of Erdős–Rényi that captures community structure, has also been studied in the online setting. In the dense regime, [92] analyze max-weight policies, and give necessary and sufficient conditions to achieve perfect matchings infinitely often. In the general SBM, [25] extend lower bounds from the Erdős–Rényi case, demonstrating that the 0.837 bound from [79] remains tight when communities have equal expected degrees. They propose several efficient heuristics for online matching in SBMs, which perform well empirically, but none are proven to achieve asymptotic optimality. However, all these works focus primarily on dense regimes or heuristics without theoretical guarantees. In particular, the sparse regime of SBM, which is highly relevant for many real-world applications where connections are scarce and structured, has received limited attention. From another perspective, bandit algorithms offer a general framework for decision-making under uncertainty, where limited feedback guides sequential choices. These models focus on balancing between exploration and exploitation and have found broad application in online learning and resource allocation [72, 91]. While conceptually distinct, they share core challenges with online matching and offer complementary insights.

8.2   Model

We consider the online bipartite matching problem, where the nodes on one side arrive sequentially, with an additional graph structure given by a *stochastic block model*. The latter is defined by a bipartite graph $G = (U, V, E)$, where $U = [n] :=$ $\{1, \ldots, n\}$ is the set of "offline" nodes, $V = [T]$ is the set of "online" nodes, where

$n, T \in \mathbb{N}^*$, and $E \subset U \times V$ is the set of edges. This underlying graph is random, in the sense that each edge $(u,t) \in U \times V$ belongs to E independently with some probability. The block model assumes that each node belongs to a latent class: we denote by $\mathcal{C} := [C]$ the set of classes on the offline side, and by $\mathcal{D} := [D]$ the set of classes on the online side. Each offline node $u \in U$ is assigned a class $c(u) \in \mathcal{C}$, and each online node $t \in V$ is assigned a class $d(t) \in \mathcal{D}$. These assignments are drawn independently: nodes on the offline side are sampled from a distribution μ over $\mathcal{C}$, and nodes on the online side are sampled from a distribution ν over $\mathcal{D}$. Given the class labels, the edge (u,t) appears in E with probability $p_{u,t} = p\big(c(u), d(t)\big) \in [0,1]$, where $p = (p(c,d))_{c,d \in \mathcal{C} \times \mathcal{D}}$ is a class-to-class affinity matrix. In this work, we focus on the *sparse regime*, where the underlying graph remains sparse as the number of offline nodes n grows. More precisely, we assume the existence of a non-negative matrix $a = (a_{c,d}) \in \mathbb{R}_+^{C \times D}$ such that $p(c,d) = \frac{a_{c,d}}{n}$. Moreover, we assume that $a_{c,d} \leq a$ for all $c \in \mathcal{C}$, $d \in \mathcal{D}$, where $a \in (0,n)$ is a fixed constant. This choice ensures that each offline node has a bounded expected degree, even as $n \to \infty$, which reflects realistic constraints in large-scale platforms where individual users or items interact with only a limited number of others.

As mentioned in the introduction, in the online matching problem, an algorithm ALG observes sequentially the vertices in V and constructs on the fly a matching (i.e., a subset of edges such that any vertex belongs to at most one of them) irrevocably: after seeing the vertex $t \in V$, it can decide to add irrevocably an edge $(u,t) \in E$ to the current matching if t does not belong to an edge of the matching yet.

We shall now introduce some notations. We denote by b_c the proportion of nodes of U of class $c \in \mathcal{C}$. We also assume that there exists some scaling factor $\alpha > 0$ such that $T = \alpha n$ (as we will consider asymptotic results when n is large). We will also define the Boolean variable $m_u(t)$ equal to 1 if, and only if, the vertex u has been included in the matching by ALG before the vertex t arrives (otherwise $m_u(t) = 0$). Additionally, we denote by $\mathcal{N}_c := \{u \in U, c(u) = c\}$ the set of nodes of class $c \in \mathcal{C}$, by $\mathcal{M}_c(t) = \{u \in \mathcal{N}_c, m_u(t) = 1\}$ the set of vertices of class c that are already matched before seeing vertex $t \in V$ (we denote by $M_c(t)$ its cardinality) and by $M(t) := \sum_{c \in \mathcal{C}} M_c(t)$ the size of the matching constructed so far. We also denote by $\mathcal{F}_c(t) = \{u \in \mathcal{N}_c \backslash \mathcal{M}_c(t), (u,t) \in \mathcal{E}\}$ the set of vertices of class c that are not matched so far (thus free), but such that (u,t) belongs to E (they are the "available neighbors" of t of class c), and by $C(t) = \{c \in [C] | \mathcal{F}_c(t) \neq \varnothing\}$ the set of classes that have available nodes at time $t \in [T]$. Finally, we shall denote by e_i the i-th basis vector of $\mathbb{R}^C$.

8.3 Known compatibility probabilities

8.3.1 A warm-up: Myopic algorithm

In this section, we present a simple yet foundational algorithm, Myopic, which serves as a baseline for more sophisticated algorithms presented later. This algorithm is

designed to make fast, greedy decisions for matching, without attempting to look ahead or anticipate future availability. Specifically, when a new vertex $t \in V$ arrives (e.g., a request or a user), the policy *selects a class $c_t \in \mathcal{C}$* according to a fixed probability distribution, and then *attempts to match t* to an available node within that class. The selection is made without verifying beforehand whether any nodes in the chosen class are actually available at time t. As a result, `Myopic` is computationally simple and immediate in its decisions, but it may sometimes fail to make a match due to resource unavailability. Despite its reactive nature, `Myopic` is carefully designed to respect class-specific budget constraints and to maximize the expected long-term success rate of matches. The key component of the algorithm is the computation of a probability matrix $Q^*(c, d)$, which solves the following optimization problem:

$$Q^* \in \arg\max_Q \sum_{c,d} Q(c, d) p(c, d),$$

$$\text{s.t} \quad \sum_d Q(c, d)\nu(d) = b_c, \ \forall c \in \mathcal{C} \quad \text{and} \quad \sum_c Q(c, d)\nu(d) = \nu(d), \forall d \in \mathcal{D}.$$

In particular, Q^* represents the optimal transport plan that maps the distribution ν to the budget vector b, minimizing the transport cost with respect to $-p(c, d)$. This matrix can be computed efficiently using the Hungarian algorithm, with a computational complexity of $\mathcal{O}(CD(C + D))$. Notably, the special case where there is only one class (i.e., $C = D = 1$) reduces to the setting studied in [79].

Algorithm 11: `Myopic` policy

Output: Updated matching $M(t)$

1 Compute the optimal transport plan Q^*.

2 for $t \in [T]$ **do**

3   Choose $c_t \in \mathcal{C}$ at random with probability $Q^*(c_t, d_t)/\nu(d_t)$.

4   **if** $\mathcal{F}_{c_t}(t) = \varnothing$ **then**

5 $M(t) = M(t - 1)$.

6   **else**

7 $M(t) = M(t - 1) \cup \{(n_t, t)\}$ for $n_t \sim \text{unif}(\mathcal{F}_{c_t}(t))$.

In order to understand the behavior of the `Myopic` policy in large-scale matching markets, we consider a deterministic approximation via an ordinary differential equation (ODE). The guiding intuition is that, as the number of agents grows (nodes in the graph), stochastic variability in the system averages out, and the evolution of the system can be captured by a smooth deterministic trajectory. However, the ODE associated with the dynamics of the `Myopic` policy Equation (8.1) is nonlinear and does not generally admit a closed-form solution, especially due to the complex structure of the underlying graph encoded in the parameters $a_{c,d}$ and $Q^*(c, d)$. This makes direct analysis challenging. In certain structured settings, however, the ODE becomes more tractable. For instance, when $a_{c,d} = c$ for all (c, d) which corresponds to the Erdős–Rényi model, the ODE simplifies and admits a closed-form solution, as

shown in [79]. This illustrates how the structure of the graph can significantly affect the tractability of the analysis and motivates the need for approximation techniques in the more general, heterogeneous case.

To address this, we introduce a simplified surrogate ODE, which approximates the behavior of the original system while being analytically tractable. This auxiliary equation has a closed-form solution $\tilde{y}_c(t)$, allowing us to extract structural insights such as convergence rates and equilibrium behavior. Crucially, we rigorously control the error between the true ODE solution $y_c(t)$ and the approximate solution $\tilde{y}_c(t)$ by bounding it with an explicit error term $e_c(t)$.

The next theorem formalizes this two-step approximation strategy. It shows that:

- The normalized matching size produced by the `Myopic` policy closely follows the ODE solution $y_c(t)$ with high probability.

- The ODE solution is well-approximated by the simpler function $\tilde{y}_c(t)$ with a small and explicitly controlled error.

Theorem 25. *Let $y_c : [0, \alpha] \to \mathbb{R}$ be the solution of the following ODE*

$$\begin{cases} \dot{y}_c(s) &= \sum_{d=1}^{D} \left(1 - e^{-a_{c,d}(b_c - y_c(s))}\right) Q^*(c, d), \\ y_c(0) &= 0. \end{cases} \tag{8.1}$$

Then, for each class $c \in \mathcal{C}$, the matching size $M_c(t)$ produced by `Myopic` satisfies, for all $t \in [T]$

$$\left| \frac{M_c(t)}{n} - y_c(t/n) \right| \leq \frac{3L_c e^{\alpha L_c}}{n^{1/3}}, \quad \text{where } L_c = \sum_{d=1}^{D} a_{c,d} Q^*(c, d),$$

with probability at least $1 - 2Ce^{-n^{1/3}L_c^2/8\alpha}$. Moreover, for $c \in [C]$, $y_c(t) = \tilde{y}_c(t) - e_c(t)$, where $e_c(0) = 0$, and $\tilde{y}_c(t) = b_c - b_c \exp(-tL_c)$, and e_c satisfies,

$$e_c(t) \leq \frac{J_c}{L_c}(1 - e^{-L_c t}).$$

Where $J_c = \frac{b_c^2}{2} \sum_{d=1}^{D} a_{c,d}^2 Q^*(c, d)$.

Remark 4. *The existence and uniqueness of the solution to Equation (8.1) follow directly from the Cauchy–Lipschitz (Picard–Lindelöf) theorem, since the vector field satisfies the required regularity assumptions.*

The full proof of Theorem 25 is in Section 8.B.1.

8.3.2 Ex-ante Balance

The `Ex-ante Balance` algorithm builds on the limitations of the `Myopic` policy by trying to take into account node availability. When a node of class $c(t)$ arrives, the `Myopic` policy selects a compatible class c purely based on potential compatibility between the classes—without considering whether any unmatched nodes from that class are actually available at that moment. In contrast, `Ex-ante Balance` takes a more informed approach: it chooses the class c that maximizes the probability that at least one unmatched node is available and connected to the arriving node. This probability is given by $1 - \left(1 - \frac{a_{c,j}}{n}\right)^{nb_c - M_c(t)}$, which captures the likelihood of an edge existing under a stochastic block model, where edge probabilities between classes are governed by parameters $a_{c,j}$.

Algorithm 12: `Ex-ante Balance`

   Output: Updated matching $M(t)$.

1   **for** $t \in [T]$ **do**

2   Choose $c_t = \arg\max_{c \in [1,C]} \sum_{j=1}^{D} (1 - (1 - \frac{a_{c,j}}{n})^{nb_c - M_c(t)})\nu(j)$.

3   **if** $\mathcal{F}_{c_t}(t) = \varnothing$ **then**

4 $M(t) = M(t-1)$.

5   **else**

6 $M(t) = M(t-1) \cup \{(u_t, t)\}$ for $u_t \sim \mathrm{unif}(\mathcal{F}_{c_t}(t))$.

A key distinction between the `Myopic` and `Ex-ante Balance` policies lies in how they respond to the evolving state of the system—and this difference has important implications for their large-scale behavior. The `Myopic` approach selects a class based solely on static compatibility between node types, leading to smooth dynamics. As the system scales, the randomness introduced by node arrivals averages out, and the evolution of the system can be accurately described by an ordinary differential equation (ODE). This continuous, deterministic approximation leverages the fact that the `Myopic` policy's decision rule is smooth and Lipschitz-continuous. In contrast, the `Ex-ante Balance` policy takes a more strategic decision by selecting the class that maximizes the actual match probability. This introduces discontinuities into the process—small variations in the system state can lead to abrupt changes in the selected class. As a result, the system no longer evolves smoothly, and the assumptions required for ODE convergence break down.

To address this, we turn to the framework of differential inclusions, a generalization of ODEs designed to handle such discontinuous dynamics. Rather than prescribing a single trajectory, a differential inclusion allows the system's evolution to follow a set of possible directions at each point, capturing the non-smooth transitions in behavior driven by abrupt changes in decision rules. The following theorem formalizes this connection by proving that, with high probability, the normalized matching sizes produced by `Ex-ante Balance` converge to a solution of a differential inclusion. For an introduction to differential inclusions and their relevance in this context, see Section 8.A.

Theorem 26. *Let m be the unique solution of the differential inclusion*

$$\dot{m} \in F(m) := conv\left\{ f_{c,b_c}(m_c)e_c \; ; \; c \in \arg\max_{k\in[C]} f_{k,b_k}(m_k) \right\},$$

which is the convex hull of the mappings

$$f_{c,b_c}(x) = \sum_{d=1}^{D}(1 - e^{-a_{c,d}(b_c-x)})\nu(d).$$

Then the matching built by **Ex-ante Balance** *satisfies for all $t \in [T]$ and $c \in \mathcal{C}$, with probability at least $1 - \frac{b\alpha}{n\epsilon^2}$,*

$$\left| \frac{M_c(t)}{n} - m_c(t/n) \right| \leq \min(\alpha, e^{L\alpha}/\sqrt{2L})\sqrt{A_{\alpha,c}/n + \delta_c B_{\alpha,c} + \epsilon C_{\alpha,c}},$$

The different constants in are defined by $L = \max_{c\in[C]} \sum_{d=1}^{D} a_{c,d}\nu(d)$, $\delta_c = \frac{\sum_{d=1}^{D} \frac{a_{c,d}}{e}\nu(d)}{n}$, $K_\alpha = (c\alpha + \epsilon)e^{c\alpha}/c$, ϵ as defined in Lemma 35 and c in Lemma 31, $U_c = \sum_{d=1}^{D}(1 - e^{-a_{c,d}b_c})\nu(d)$, $A_{\alpha,c} = U_c(U_c^2 + \frac{14U_c}{3} + 2K_\alpha)$, $B_{\alpha,c} = 2U_c^2 + 4L\delta_c + 12K_\alpha$, $C_{\alpha,c} = 2U_c^2 + 4L\epsilon + 8K_\alpha$.

As `Ex-ante Balance` algorithm always picks the class with the highest probability of connection, which decreases in case of match, it tends to progressively equalize these probabilities across the classes in $\mathcal{C}$. Once two classes have their probabilities (almost) equalized, these stay equal over time by decreasing at the same rate. This generates phases, in which the k "first" classes (with highest initial probability) have their probabilities equalized, decreasing at the same rate, up until reaching the probability of the $(k+1)^{\text{th}}$ class. This phasing allows to obtain an explicit formula for m. The remainder of the section introduces the exact formula and some intuitions of how it is built, while the formal proof is deferred to Section 8.C.2.

In the large-n limit, the marginal probability that the next online node is connected to at least one node in class $c \in \mathcal{C}$, given a vector $\boldsymbol{\beta} \in \mathbb{R}_+^C$ representing the proportions of available nodes in each class, is given by: $f_{c,\beta_c}(z) = \sum_{d\in\mathcal{D}}\left(1 - e^{-a_{c,d}(\beta_c-z)}\right)\nu(d)$. W.l.o.g., we assume for the analysis that the elements of $\mathcal{C}$ are ordered by decreasing marginal probability of receiving at least one edge at the initial time step – i.e. $f_{1,b_1}(0) \geq f_{2,b_2}(0) \geq \cdots \geq f_{C,b_C}(0)$. Additionally, as $a_{c,d} > 0$, f_{c,β_c} is strictly decreasing and thus invertible, with f_{c,β_c}^{-1} also strictly decreasing.

During phase k, the $C - k$ last classes are not selected at all, while the k first classes have their probabilities equalized, decreasing at the same rate. Given the budgets $\boldsymbol{\beta} \in \mathbb{R}_+^C$ at the beginning of the phase, the number of nodes matched during the phase, in each of the k classes, at time t, ends up evolving following $\mu_{k,\boldsymbol{\beta}}(t)$

defined as the solution of the following separable ODE

$$\begin{cases} \frac{\mathrm{d}\mu_{k,\boldsymbol{\beta}}}{F_{k,\boldsymbol{\beta}}(\mu_{k,\boldsymbol{\beta}})} = \mathrm{d}t \\ \mu_{k,\boldsymbol{\beta}}(0) = 0 \end{cases} \qquad \text{where} \qquad F_{k,\boldsymbol{\beta}} = \left(\sum_{c=1}^{k} f_{c,\beta_c}^{-1} \right)^{-1}.$$

Note that, as $\frac{\mathrm{d}\mu_{k,\boldsymbol{\beta}}}{\mathrm{d}t} > 0$, $\mu_{k,\boldsymbol{\beta}}$ is strictly increasing, positive and invertible.

To assemble the phases, and provide the full expression of m, two sequences need to be defined. First, the sequence of time-steps $(t_k)_{k \in [C]}$ defines the phases. Then $\left(\boldsymbol{\beta}^{(k)} \right)_{k \in [C]}$ describes the proportion of available nodes per class at the start of each phase. They are defined as follows:

$$\forall c, k \in [C], \ \boldsymbol{\beta}_c^{(k)} = \begin{cases} b_c & \text{if } k \leq c, \\ \boldsymbol{\beta}_c^{(k-1)} - f_{c,\boldsymbol{\beta}^{(k-1)}}^{-1} \left(f_{k,\boldsymbol{\beta}^{(1)}}(0) \right) & \text{otherwise.} \end{cases}$$

$$\forall k \in [C+1], \ t_k = \begin{cases} 0 & \text{if } k = 1, \\ \min \left(T, t_{k-1} + \mu_{k-1,\boldsymbol{\beta}^{(k-1)}}^{-1} \left(F_{k-1,\boldsymbol{\beta}^{(k-1)}}^{-1} \left(f_{k,\boldsymbol{\beta}^{(1)}}(0) \right) \right) \right) & \text{otherwise.} \end{cases}$$

Theorem 27. *For any $c \in \mathcal{C}$, the function*

$$m_c^* : t \mapsto (b_c - \boldsymbol{\beta}_c^{(k_t)}) + \left(f_{c,\boldsymbol{\beta}^{(k_t)}}^{-1} (F_{k_t,\boldsymbol{\beta}^{(k_t)}}(\mu_{k_t,\boldsymbol{\beta}^{(k_t)}}(t - t_{k_t}))) \right)_+.$$

where $k_t = \max\{k \in [C] : t > t_k\}$, is the unique solution of the differential inclusion defined in Theorem 26.

Figure 8.1 illustrates the accuracy of $m_c^*(t/n)$ to estimate $\frac{M_c(t)}{n}$ in the sparse regime.

8.3.3 Ex-post Balance

While the `Ex-ante Balance` policy improves upon the purely compatibility-driven `Myopic` approach by taking into account the probability that a class still has available nodes, it operates under an expected-availability principle: it selects the class that is most likely to contain unmatched nodes, but without checking whether any of these available nodes are actual neighbors of the newly arriving vertex. As the reviewer points out, the algorithm indeed selects a class whose expected availability is positive, but this does not guarantee that the specific available nodes in that class are compatible with the current arrival. Consequently, `Ex-ante Balance` may still choose a class in which none of the actually available nodes can be matched to the incoming vertex, leading to wasted opportunities. To address this limitation, we introduce the `Ex-post Balance` policy. This variant strengthens the decision

rule by checking, before selecting a class, that there is at least one available node within that class that is also a neighbor of the arriving vertex. In other words, `Ex-post Balance` does not rely solely on expected availability but verifies real-time compatibility between the incoming vertex and the remaining nodes. This additional condition prevents the algorithm from targeting classes where no feasible match is possible. We show that the matching size produced by `Ex-post Balance` converges with high probability to the solution of a differential inclusion. Unlike the inclusion in Theorem 26, the limiting system for `Ex-post Balance` explicitly accounts for these real-time feasibility constraints. Full details and proofs are provided in Section 8.C.3.

8.3.4 Balance algorithm with smoothing

We previously examined the `Ex-ante Balance` algorithm, which introduces discontinuities due to its strategy of always selecting the class with the highest matching probability. This non-smooth behavior causes the system to converge not to a standard differential equation, but rather to a differential inclusion, which captures the set-valued dynamics induced by the discontinuities. While this is not inherently worse, it leads to a different type of analysis. As an alternative, we consider a smoothed version of `Ex-ante Balance`, denoted by `Smoothbalance`, where the max function is replaced with a smooth approximation. This smoothing alters the convergence behavior, leading to a differential equation instead of a differential inclusion, and enables a different—though not necessarily better—analytical perspective.

Algorithm 13: `Smoothbalance`

   Output: Updated matching $M(t)$.
1 Node $t \in [T]$ arrives, its class $d(t) \in [D]$ is revealed
2 Each class $c \in [C]$ is chosen with probability $q_{c,d(t)}$
3 The policy chooses a class c_t^* with probability $q_{c_t^*,d(t)}$
4 if $W_{c_t^*}(t) = \varnothing$ **then**
5 | $M(t) = M(t-1)$.
6 else if $W_{c_t^*}(t) = \{w_1, \dots, w_L\}$ *and* $\exists 1 \le j \le L$ *such that* $(w_j, t) \in E$ **then**
7 | $M(t) = M(t-1) \cup (w_j, t)$.
8 else
9 | $M(t) = M(t-1)$.

For $\xi > 0$, $q_{c,d(t)} = \dfrac{\left(1 - \left(1 - \frac{a_{c,d(t)}}{n}\right)^{nb_c - M_c(t)}\right)^{\frac{1}{\xi}}}{\sum_{j \in [C]} \left(1 - \left(1 - \frac{a_{j,d(t)}}{n}\right)^{nb_j - M_j(t)}\right)^{\frac{1}{\xi}}}.$

Theorem 28. *With probability* $1 - \mathcal{O}(n^{1/4} e^{-n^{1/4} \left(\sum_{d=1}^{D} a_{c,d}\right)^3})$, *the number of matched nodes in the class* $c \in [C]$ *satisfies,*

$$M_c(T) = n z_c(T/n) + \mathcal{O}(n^{3/4}).$$

where z_c with $z_i(0) = 0$ and $0 \le \tau \le T/n$, is the solution of,

$$\dot{z}_c(\tau) = \sum_{d=1}^{D} \left(1 - e^{-a_{c,d}(b_c - z_c(\tau))}\right)^{1+\frac{1}{\xi}} \frac{1}{\sum_{k=1}^{C} \left(1 - e^{-a_{k,d}(b_k - z_k(\tau))}\right)^{\frac{1}{\xi}}} \nu(d). \qquad (8.2)$$

The proof of Theorem 28 can be found in Section 8.C.4.

The stationary solution. Solving Equation (8.2) directly is challenging due to its non-linearity and the coupled nature of the terms across classes. Instead of seeking a closed-form solution, we focus on analyzing the stationary points of the system and studying their stability. By showing that the solution converges to a stable stationary state, we can infer that the matching process concentrates around a function determined by this equilibrium. This approach allows us to characterize the long-term behavior of the system without having to explicitly solve the full dynamic equation.

Let $y_c(\tau) = b_c - s_c(\tau)$, Equation (8.31) becomes,

$$\dot{y}_c(\tau) = -\sum_{d=1}^{D} \left(1 - e^{-a_{c,d}y_i(\tau)}\right)^{1+\frac{1}{\xi}} \frac{1}{\sum_{k=1}^{C} \left(1 - e^{-a_{k,d}y_k(\tau)}\right)^{\frac{1}{\xi}}} \nu(d), \qquad (8.3)$$

with $y_c(0) = b_c$

Lemma 24. *Equation* (8.3) *has a unique stable stationary solution denoted* $y_c^* = 0$.

The proof of Lemma 24 can be found in Section 8.C.4.

We have established that the stationary solution of the differential equation is stable, meaning that trajectories starting close to the equilibrium will remain close and eventually converge to it. However, we have not shown exponential stability, so we lack precise information about the rate of convergence. In particular, without exponential stability, we cannot guarantee how fast the system approaches the stationary state. Despite this, the stability result is sufficient to analyze the concentration of M_c, as it allows us to conclude that the matching process will, in the long run, remain close to a function governed by the stationary solution, even if the convergence may be slow.

8.4   Unknown compatibility probabilities

In many real-world applications, the underlying parameters of the graph—such as the connection probabilities between node classes or the distribution of arriving node types—are not known *a priori*. These parameters may be shaped by latent

variables, evolve over time, or be inferred only through noisy and partial observations. Consequently, algorithms must operate under uncertainty and learn these parameters dynamically in order to make effective matching decisions. In this section, we study the setting in which the connection probabilities $a_{c,d}$ are unknown and must be estimated online. This transforms the problem into a bandit setting, where each class $c \in \mathcal{C}$ can be viewed as an arm, and upon choosing class c_t at time t, a Bernoulli reward is observed—indicating whether a successful match occurred between the arriving node and an available node in class c_t.

8.4.1   ETC − balance

Algorithm 14 presents the ETC − balance policy, which combines a fixed-duration Explore-Then-Commit (ETC) strategy with the Ex-ante Balance rule for class selection. The algorithm proceeds in two phases. During the exploration phase, which lasts for a fixed number of rounds T_{explore}, arriving nodes are matched by selecting classes uniformly at random. This allows the algorithm to collect data on the outcomes of triplets (c, d, m), where $c \in \mathcal{C}$, $d \in \mathcal{D}$, and m is the current matching size of the class c. We let $T_{c,d,m}$ denote the number of times the triplet (c, d, m) has been observed. After this phase, the algorithm enters the commitment phase and uses the collected data to estimate the match failure probabilities. Specifically, it estimates $D_{c,d}(m) = \left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - m}$ using an estimator $\hat{D}_{c,d}(m)$, whose form and concentration properties are given in Lemma 44. These estimates are then incorporated into the Ex-ante Balance rule to select the class c that maximizes the expected success probability, weighted by the type distribution $\nu(d)$.

8.4.2   Regret

For each class $i \in \mathcal{C}$, let $M_i(t)$ and $\hat{M}_i(t)$ denote the number of matches made by the Ex-ante Balance and ETC − balance algorithms, respectively, up to time t. We define the regret of ETC − balance as the total difference in matching performance across all classes when compared to Ex-ante Balance, which has full knowledge of the match probabilities. In other words, the regret quantifies the cumulative loss incurred by ETC − balance due to not knowing the match probabilities in advance. The following result shows that this regret grows at most on the order of $\mathcal{O}(n^{(q+3)/4})$ for some $0 < q < 1$.

Theorem 29. *Let $R(T) = \sum_{i \in \mathcal{C}} M_i(T) - \hat{M}_i(T)$ denote the regret of* ETC − balance. *Suppose the exploration phase lasts for $T_{\text{explore}} = T^{\frac{q+3}{4}}$, for some $0 < q < 1$. Then the regret satisfies*

$$R(T) = \mathcal{O}(T^{\frac{q+3}{4}}).$$

Algorithm 14: `ETC − balance`

Input: T, $\nu(d)$, n, T_{explore}, confidence level δ
Output: Matching $M(t)$.

1 Init: $\mathcal{M}(0) = \varnothing$, $M_c(0) = 0$, $T_{c,d,m}(0) = 0$, $\hat{D}_{c,d}(m) = 1$.

2 for $t = 1$ **to** T **do**

3   Node t of type $d(t)$ arrives.

4   **if** $t \leq T_{explore}$ **then**

5   Select c_t uniformly at random.

6   **else**

7   **foreach** $c \in [C]$ **do**

8   Let $m = M_c(t-1)$, $s(m) = \frac{b_c}{b_c - m/n}$

9   Compute $\hat{D}_{c,d}(m)$

10   Choose $c_t = \arg\max_c \sum_d 1 - \hat{D}_{c,d}(M_c(t-1))\nu(d)$.

11   **if** $\mathcal{F}_{c_t}(t) \neq \varnothing$ **then**

12   Match (u_t, t) for $u_t \sim \text{unif}(\mathcal{F}_{c_t}(t))$, update $\mathcal{M}(t)$, $M_{c_t}(t)$, and $Y_{c_t,d(t)}(t) = 1$.

13   **else**

14   No match, $Y_{c_t,d(t)}(t) = 0$.

15   Update $T_{c_t,d(t),M_{c_t}(t)} \leftarrow T_{c_t,d(t),M_{c_t}(t)} + 1$.

8
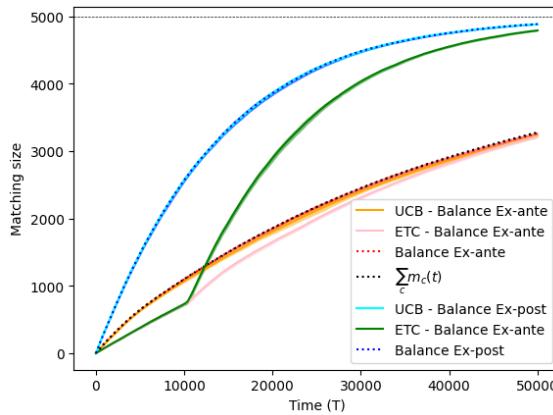
Figure 8.1: Illustration of the matching size for the different methods. $T = 50000$, $n = 5000$, $C = 5$, $D = 6$ and simulations are averaged over 20 trajectories. **1.** As expected the *ex-post* versions of the algorithms perform better than their *ex-ante* counterparts. **2.** The fluid limit m^* (dark) is an accurate estimator of the actual empirical trajectory of `Ex-ante Balance` (red). **3.** A UCB version of balance can be built using the confidence set from Lemma 44 and performs empirically better than ETC. Yet, its analysis remains an open question.

Conclusion

We studied online bipartite matching within the Stochastic Block Model (SBM), capturing structured heterogeneity in real-world networks through class-dependent connection probabilities. We analyzed two main algorithms under known probabilities: the Myopic policy, which is simple but limited by ignoring availability, and the `Ex-ante Balance` algorithm, which accounts for compatibility and capacity, and is shown to converge to a differential inclusion. When the probabilities are unknown, we introduced `ETC − balance`, a bandit-based extension that learns affinities over time and achieves sublinear regret. Simulations confirm that UCB-based methods outperform `ETC − balance`, while `Ex-ante Balance` under known probabilities closely matches actual outcomes, validating its effectiveness. A promising future research direction is to analyze UCB in this setting using differential inclusion tools, to better understand its asymptotic behavior and theoretical guarantees.

8

Appendix 8

8.A   Differential inclusions

This section aims to introduce the fundamental concepts of differential inclusions. Unlike ordinary differential equations (ODE), where the derivative of the unknown function is determined by a single-valued map, differential inclusions generalize this by allowing the derivative to lie within a set-valued map. This broader framework is well-suited for modeling dynamical systems that involve uncertainty, discontinuities, or control constraints.

8.A.1   Set-Valued Maps

Let $F : \mathbb{R}^n \to \mathbb{R}^n$ denote a set-valued map, i.e a mapping which assigns to each point $x \in \mathbb{R}^n$ a subset $F(x) \subseteq \mathbb{R}^n$. We consider in general that $F(x)$ is nonempty for all $x \in \mathbb{R}^n$. The notation $\langle x, y \rangle$ denotes the standard inner product on $\mathbb{R}^d$, and the norm is given by $\|x\| = \sqrt{\langle x, x \rangle}$. For a set $A \subset \mathbb{R}^d$, we define its norm as $\|A\| = \sup_{x \in A} \|x\|$.

A set-valued map F is said to be:

- **Upper semicontinuous (u.s.c.):** A set-valued map $F : \mathbb{R}^n \to \mathbb{R}^n$ is upper semicontinuous at a point $y \in \mathbb{R}^n$ if for every sequence $y^{(n)} \to y$ and any sequence $x_n \in F(y^{(n)})$ such that $x_n \to x$, it holds that $x \in F(y)$.

- **Locally bounded** if for every compact set $K \subset \mathbb{R}^n$, there exists a constant M such that,
$$\sup_{x \in K} \sup_{v \in F(x)} \|v\| \le M.$$

- **Measurable** if its graph is measurable in the product sigma-algebra on $\mathbb{R}^n \times \mathbb{R}^n$.

- **One sided Lipschitz** with constant L if for all $z, \tilde{z} \in \mathbb{R}^n$ and for $z \in F(y), \tilde{z} \in F(\tilde{y})$,
$$\langle y - \tilde{y}, z - \tilde{z} \rangle \le L \|y - \tilde{y}\|^2.$$

8.A.2   Definition of differential inclusions

Definition 10. *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a set valued and $T > 0$. A differential inclusion is defined as,*

$$\dot{x}(t) \in F(x(t)) \qquad \text{for } t \in [0, T]. \tag{8.4}$$

together with an initial condition,

$$x(0) = x_0.$$

Let $I \subseteq \mathbb{R}$, a function $x : I \to \mathbb{R}^n$ is a solution to the differential inclusion defined in Equation (8.26) with initial condition $x(0) = x_0$ if there exists a function $\phi : I \to \mathbb{R}^n$ such that:

- For all $t \in I$: $x(t) = x(0) + \int_0^t \phi(s)\mathrm{d}s$.

- For almost every $t \in I$ $\phi(t) \in F(x(t))$.

8.A.3 Existence and uniqueness of the solution

Proposition 4. *([41, 10, 71])*

- *If F is upper semicontinuous and if there exists c such that $\|F(y)\| \leq c(1+\|y\|)$ then for any initial condition x_0, $\dot{x} \in F(x)$ has at least one solution on $[0, +\infty)$ with $x(0) = x_0$.*

- *If F is one-sided Lipschitz, then for all $T > 0$ there exists at most one solution of $\dot{x} \in F(x)$ on $[0, T]$.*

- *If F is upper semicontinuous and one-sided Lipschitz then for $T > 0$, $\dot{x} \in F(x)$ has a unique solution on $[0, T]$.*

8.B Myopic algorithm

This section is organized into two parts. In the first part, we analyze the `Myopic` algorithm. In the second part, we show that, under certain assumptions, the model reduces to the Erdős–Rényi case studied in [79]. We begin by considering the following `Myopic` algorithm, as introduced in the main paper:

Algorithm 15: `Myopic` policy

Output: Updated matching $M(t)$

1 Compute the optimal transport plan Q^*

2 for $t \in [T]$ **do**

3 $\quad$ Choose $c_t \in \mathcal{C}$ at random with probability $Q^*(c_t, d_t)/\nu(d_t)$.

4 $\quad$ **if** $\mathcal{F}_{c_t}(t) = \varnothing$ **then**

5 $\quad\quad$ $M(t) = M(t-1)$.

6 $\quad$ **else**

7 $\quad\quad$ $M(t) = M(t-1) \cup \{(u_t, t)\}$ for $u_t \sim \mathrm{unif}(\mathcal{F}_{c_t}(t))$.

8.B.1   Proof of Theorem 25

Theorem 25. *Let $y_c : [0, \alpha] \to \mathbb{R}$ be the solution of the following ODE*

$$\begin{cases} \dot{y}_c(s) & = \sum_{d=1}^{D} \left(1 - e^{-a_{c,d}(b_c - y_c(s))}\right) Q^*(c, d), \\ y_c(0) & = 0. \end{cases} \tag{8.1}$$

Then, for each class $c \in \mathcal{C}$, the matching size $M_c(t)$ produced by Myopic *satisfies, for all $t \in [T]$*

$$\left| \frac{M_c(t)}{n} - y_c(t/n) \right| \leq \frac{3 L_c e^{\alpha L_c}}{n^{1/3}}, \quad \text{where } L_c = \sum_{d=1}^{D} a_{c,d} Q^*(c, d),$$

with probability at least $1 - 2Ce^{-n^{1/3} L_c^2 / 8\alpha}$. Moreover, for $c \in [C]$, $y_c(t) = \tilde{y}_c(t) - e_c(t)$, where $e_c(0) = 0$, and $\tilde{y}_c(t) = b_c - b_c \exp(-t L_c)$, and e_c satisfies,

$$e_c(t) \leq \frac{J_c}{L_c}(1 - e^{-L_c t}).$$

Where $J_c = \frac{b_c^2}{2} \sum_{d=1}^{D} a_{c,d}^2 Q^*(c, d)$.

The proof of Theorem 25 is based on the Wormald theorem [95, 97] and is structured as follows:

- We first define the evolution of $M_c(t)$, the size of the matching constructed by Myopic in class $c \in [C]$ at time $t \in [T]$.

- We then verify that $M_c(t)$ satisfies the conditions required to apply the Wormald theorem [95, 97].

- We apply the Wormald theorem [95, 97] to analyze the behavior of $M_c(t)$.

- Next, we construct an approximate solution to the differential equation that serves as a continuous approximation of $M_c(t)$.

- Finally, we derive an explicit bound on the error between the true solution and its approximation.

Let $\mathbf{M}(t) = (M_1(t), \ldots, M_C(t))$ denote the vector of matching sizes in each class $c \in [C]$ constructed by the Myopic algorithm. For each class $c \in [C]$, the matching size evolves according to the following dynamics:

$$M_c(t + 1) = M_c(t) + \mathbb{1}_{\{\exists u \in \backslash_c(t) \text{ s.t } c_t^* = c \text{ and } m_u(t+1) = 1\}} \tag{8.5}$$

Here, c_t^* represents the class selected by Myopic at time t.

The first step is to compute the expected one-step change in $M_c(t)$, the size of the matching constructed by `Myopic`. This is formalized in the following lemma.

Lemma 25. *For $t \in [T]$, $c \in [C]$ and for any $B = (b_1, \ldots, b_C)$, the expectation of the one-step change of $M_c(t)$, when matching is constructed using `Myopic` is given by,*

$$\mathbb{E}[M_c(t+1) - M_c(t)|\mathbf{M}(t), B] = F_c(t, M_1(t), \ldots, M_C(t)).$$

where $F_c(t, M_1(t), \ldots, M_C(t)) = \sum_{d=1}^{D} \left(1 - \left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - M_c(t)}\right) Q^*(c, d)$.

Proof. Moving to conditional expectation gives,

$$\mathbb{E}[M_c(t+1) - M_c(t)|\mathbf{M}(t), B]$$
$$= \mathbb{E}[\mathbb{1}_{\{\exists u \in \mathcal{N}_c(t), c_t^* = c, m_u(t+1) = 1\}}|\mathbf{M}(t), B]$$
$$= \mathbb{P}(\exists u \in \mathcal{N}_c(t), c_t^* = c, m_u(t+1) = 1|\mathbf{M}(t), B)$$
$$= \sum_{d=1}^{D} \mathbb{P}(\exists u \in \mathcal{N}_c(t), c_t^* = c, m_u(t+1) = 1|\mathbf{M}(t), B, d(t+1) = d)\nu(d)$$
$$= \sum_{d=1}^{D} \mathbb{P}(\exists u \in \mathcal{N}_c(t), m_u(t+1) = 1|\mathbf{M}(t), B, d(t+1) = d, c_t^* = c)\nu(d)$$
$$\mathbb{P}(c_t^* = c|\mathbf{M}(t), B, d(t+1) = d)$$
$$= \sum_{d=1}^{D} \left(1 - (1 - p(c, d))^{nb_c - M_c(t)}\right) \nu(d)Q^*(c, d)/\nu(d)$$
$$= \sum_{d=1}^{D} \left(1 - \left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - M_c(t)}\right) Q^*(c, d).$$

$\square$

The following lemma establishes the Lipschitz continuity of the function $f_c(x) = \sum_{d=1}^{D}(1 - e^{-a_{c,d}(nb_c - x)})Q^*(c, d)$.

Lemma 26. *For $x \le nb_c$ and $c \in [C]$, the function f_c is L_c-Lipschitz with $L_c = \sum_{d=1}^{D} a_{c,d}Q^*(c, d)$.*

Proof. For $x \le nb_c$, the function f_c is a sum of continuous and differentiable functions on $\mathbb{R}$. Its derivative is given by

$$|f_c'(x)| = \sum_{d=1}^{D} e^{-a_{c,d}(nb_c - x)}a_{c,d}Q^*(c, d) \le \sum_{d=1}^{D} a_{c,d}Q^*(c, d).$$

Since f_c is differentiable, the Mean Value Theorem implies that for any $x, y \in \mathbb{R}$ with $x, y \le nb_c$, there exists $\xi \in (x, y)$ such that

$$|f_c(x) - f_c(y)| = |f_c'(\xi)| \, |x - y| \le L_c|x - y|,$$

where the Lipschitz constant is defined by $L_c = \sum_{d=1}^{D} a_{c,d} Q^*(c,d)$. $\qquad \square$

We have the following technical result,

Lemma 27. *For $n > 0$, $a \leq n/2$ and $0 \leq w \leq 1$,*

$$0 \leq e^{-aw} - \left(1 - \frac{a}{n}\right)^{nw} \leq \frac{a}{ne}.$$

Proof. Using the following inequalities: $1 - x \geq e^{-x-x^2}$ for $x \leq \frac{1}{2}$ and $1 - x \leq e^{-x}$ for $x \geq 0$, we obtain $e^{-aw}\left(1 - \frac{a^2 w}{n}\right) \leq \left(1 - \frac{a}{n}\right)^{nw} \leq e^{-aw}$. The result follows by rearranging terms and using that $awe^{-aw} \leq 1/e$. $\qquad \square$

The next lemma bounds the distance between F_c defined in Lemma 25 and f_c,

Lemma 28. *For $c \in [C]$,*

$$|F_c(t, M_1(t), \dots, M_C(t)) - f_c(M_c(t))| \leq \sum_{d=1}^{D} \frac{a_{c,d}}{ne} Q^*(c,d). \tag{8.6}$$

Proof. For $c \in [C]$,

$$|F_c(t, M_1(t), \dots, M_C(t)) - f_c(M_c(t))|$$

$$= \left| \sum_{d=1}^{D} e^{-a_{c,d}(nb_c - M_c(t))} Q^*(c,d) - \left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - M_c(t)} Q^*(c,d) \right|$$

$$\leq \sum_{d=1}^{D} \frac{a_{c,d}}{ne} Q^*(c,d) \qquad (Lemma\ 27).$$

$\qquad \square$

Now we are ready to prove Theorem 25.

Proof. From Lemma 25, we have

$$\mathbb{E}[M_c(t+1) - M_c(t)|\mathbf{M}(t), B] = \sum_{d=1}^{D} \left(1 - \left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - M_c(t)}\right) Q^*(c,d).$$

Let $Y_c(s) = \frac{M_c(sn)}{n}$ denote the normalized matching size in class $c \in [C]$, and define the vector $\mathbf{Y}(s) = (Y_1(s), \dots, Y_C(s))$ for $0 \leq s \leq T/n$. Then, we obtain:

$$\frac{\mathbb{E}[Y_c(s + 1/n) - Y_c(s)|\mathbf{Y}(s), B]}{1/n} = \sum_{d=1}^{D} \left(1 - \left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - nY_c(s)}\right) Q^*(c,d).$$

As $n \to \infty$, we find:

$$\text{for} \quad s \in \left[\frac{T}{n}\right] \quad \dot{y}_c(s) = \sum_{d=1}^{D} (1 - e^{-a_{c,d}(b_c - y_c(s))}) Q^*(c,d), \text{and} \quad y_c(0) = 0.$$

Applying Wormald's theorem [95], and considering the domain $0 \le y_s \le 1$ with $\beta = 1$ (by the nature of the matching process), we define the Lipschitz constant as $L_c = \sum_{d=1}^{D} a_{c,d} Q^*(c,d)$ (see Lemma 26). Additionally, we set $\delta = \sum_{d=1}^{D} \frac{a_{c,d}}{ne} Q^*(c,d)$ as in Lemma 28, which gives $\lambda = n^{-1/3} \sum_{d=1}^{D} a_{c,d} Q^*(c,d)$. Therefore, with probability at least $1 - 2Ce^{-n^{1/3}L_c^2/8\alpha}$ the approximation holds.

$$|M_c(t) - ny_c(t/n)| \le 3e^{L_c\alpha} n^{2/3} L_c$$

Where y_c satisfies for $s \in [T/n]$,

$$\dot{y}_c(s) = \sum_{d=1}^{D} \left(1 - e^{-a_{c,d}(b_c - y_c(s))}\right) Q^*(c,d) \tag{8.7}$$

$$y_c(0) = 0$$

Now we want to approximate the solution of Equation (8.7). To simplify the analysis, we introduce the change of variable $z_c(t) = y_c(t) - b_c$ where $t \in [T/n]$. Under this transformation, Equation (8.7) becomes:

$$\begin{cases} \dot{z}_c(t) &= \sum_{d=1}^{D} \left(1 - e^{a_{c,d} z_c(t)}\right) Q^*(c,d) \\ z_c(0) &= -b_c \end{cases} \tag{8.8}$$

The right-hand side of Equation (8.8) involves terms of the form $1 - e^{a_{c,d} z_c}$. Since $-b_c \le z_c \le 0$, we can use the Taylor expansion:

$$1 - e^{a_{c,d} z_c} = -a_{c,d} z_c + \mathcal{O}(z_c^2).$$

Consider now $\tilde{z}_c$ be the solution of the following linearized differential equation with initial condition $\tilde{z}_c(0) = -b_c$ and $t \in [T/n]$:

$$\tilde{z}_c'(t) = -\tilde{z}_c(t) \sum_{d=1}^{D} a_{c,d} Q^*(c,d). \tag{8.9}$$

The solution to Equation (8.9) is explicitly given by:

$$\tilde{z}_c(t) = -b_c \exp\left(-t \sum_{d=1}^{D} a_{c,d} Q^*(c,d)\right).$$

With $\tilde{z}_c$ in hand, we are ready to bound the error between the exact solution of Equation (8.8) and its approximation $\tilde{z}_c$. Let $e_c = \tilde{z}_c - z_c$ be the approximation error, its derivative for $t \in [T/n]$ is given by,

$$e_c'(t) = \tilde{z}_c'(t) - z_c'(t) \tag{8.10}$$

$$= \sum_{d=1}^{D}(-a_{c,d}\tilde{z}_c(t) - 1 + e^{a_{c,d}z_c(t)})Q^*(c,d). \tag{8.11}$$

Since $z_c \in [-b_c, 0]$, we have the inequality

$$1 - e^{a_{c,d}z_c(t)} \leq -a_{c,d}z_c(t) \quad \text{for all } d = 1, \ldots, D.$$

This implies,

$$\dot{z}_c(t) = \sum_{d=1}^{D}\left(1 - e^{a_{c,d}z_c(t)}\right)Q^*(c,d) \leq -z_c(t)\sum_{d=1}^{D}a_{c,d}Q^*(c,d).$$

This upper bound matches the right-hand side of the linearized system Equation (8.9) evaluated at $z_c(t)$. Since both solutions share the same initial condition, $z_c(0) = \tilde{z}_c(0) = -b_c$, we can apply the comparison principle [66, Chapter 3, Lemma 3.4].

It follows that

$$z_c(t) \leq \tilde{z}_c(t) \quad \text{for all } t \in [T/n],$$

which implies that the approximation error

$$e_c(t) := \tilde{z}_c(t) - z_c(t) \geq 0.$$

For all $t \in [T/n]$, using Taylor expansion of order 2,

$$e_c'(t) \leq \sum_{d=1}^{D}\left(-a_{c,d}\tilde{z}_c(t) + a_{c,d}z_c(t) + \frac{a_{c,d}^2 z_c^2(t)}{2}\right)Q^*(c,d) \tag{8.12}$$

$$e_c'(t) \leq \sum_{d=1}^{D}\left(-a_{c,d}e_c(t) + \frac{a_{c,d}^2 b_c^2}{2}\right)Q^*(c,d) \qquad (\text{using } z_c \in [-b_c, 0]) \tag{8.13}$$

$$e_c'(t) \leq -e_c(t)L_c + J_c. \tag{8.14}$$

We multiply Equation (8.14) by $e^{L_c t}$,

$$e_c'(t)e^{L_c t} + L_c e^{L_c t}e_c(t) \leq J_c e^{L_c t}.$$

Integrating both sides,

$$e^{tL_c}e_c(t) - e_c(0) \leq J_c\frac{e^{L_c t} - 1}{L_c}$$

Thus,

$$e_c(t) \leq \frac{J_c}{L_c}(1 - e^{-L_c t}) + e(0)e^{-L_c t} \tag{8.15}$$

$$e_c(t) \leq \frac{J_c}{L_c}(1 - e^{-L_c t}) \tag{8.16}$$

Thus, for $t \in [0, \frac{T}{n}]$,

$$z_c(t) = \tilde{z}_c(t) - e_c(t).$$

where $e_c(t)$ satisfies Equation (8.16), with $L_c = \sum_{d=1}^{D} a_{c,d} Q^*(c,d)$ and $J_c = \sum_{d=1}^{D} \frac{a_{c,d} b_c^2}{2} Q^*(c,d)$.

Thus replacing $z_c(t)$ by $y_c(t) - b_c$ we get the final result. $\qquad \square$

8.B.2 Recovering the Erdős–Rényi case

Consider the special case where the connection probability depends only on the class c, that is, $p(c,d) = \frac{a_c}{n}$. In this setting, the graph structure intoduced in Section 8.2 simplifies to an Erdős–Rényi random graph [17, 59]. Under this assumption, Equation (8.7) reduces to:

$$\dot{z}_c(t) = \left(1 - e^{-a_c z_c(t)}\right) \sum_{d=1}^{D} Q^*(c,d),$$

$$\frac{-a_c \dot{z}_c(t)\, e^{-a_c z_c(t)}}{e^{-a_c z_c(t)} - 1} = -a_c \sum_{d=1}^{D} Q^*(c,d).$$

Integrating both sides with respect to time yields:

$$\ln\left|e^{-a_c z_c(t)} - 1\right| - \ln\left|e^{-a_c b_c} - 1\right| = -a_c t \sum_{d=1}^{D} Q^*(c,d).$$

Solving for $z_c(t)$, we obtain the closed-form expression:

$$z_c(t) = -\frac{1}{a_c} \ln\left(1 + \left(e^{-a_c b_c} - 1\right) e^{-a_c t \sum_{d=1}^{D} Q^*(c,d)}\right). \tag{8.17}$$

This shows that when the model is reduced to the Erdős–Rényi setting, we obtain an exact solution to the corresponding differential equation. Consequently, with high probability, the size of the matching in each class $c \in [C]$ concentrates around z_c as given in Equation (8.17). These results are in close agreement with those found in [79], which also examined the dynamics of matching in Erdős–Rényi graphs and observed similar asymptotic behavior.

8.C Balance algorithm

This section is organized into two parts. First, we analyze the case where matching is performed using the `Ex-ante Balance` algorithm, as defined in the main paper. Next, we extend the analysis to the `Ex-post Balance` algorithm.

8.C.1 Ex-ante Balance

We consider the following `Ex-ante Balance` algorithm.

Algorithm 16: `Ex-ante Balance`

Output: Updated matching $M(t)$.
1 **for** $t \in [T]$ **do**
2   Choose $c_t = \arg\max_{c \in [1,C]} \sum_{d=1}^{D} (1 - (1 - \frac{a_{c,j}}{n})^{nb_c - M_c(t)}) \nu(d)$.
3   **if** $\mathcal{F}_{c_t}(t) = \varnothing$ **then**
4        $M(t) = M(t-1)$.
5   **else**
6        $M(t) = M(t-1) \cup \{(u_t, t)\}$ for $u_t \sim \text{unif}(\mathcal{F}_{c_t}(t))$.

8.C.1.1 Proof of Theorem 26

Theorem 26. *Let m be the unique solution of the differential inclusion*

$$\dot{m} \in F(m) := conv\left\{ f_{c,b_c}(m_c)e_c \; ; \; c \in \arg\max_{k \in [C]} f_{k,b_k}(m_k) \right\},$$

which is the convex hull of the mappings

$$f_{c,b_c}(x) = \sum_{d=1}^{D} (1 - e^{-a_{c,d}(b_c - x)}) \nu(d).$$

Then the matching built by ***Ex-ante Balance*** *satisfies for all $t \in [T]$ and $c \in \mathcal{C}$, with probability at least $1 - \frac{b\alpha}{n\epsilon^2}$,*

$$\left| \frac{M_c(t)}{n} - m_c(t/n) \right| \leq \min(\alpha, e^{L\alpha}/\sqrt{2L}) \sqrt{A_{\alpha,c}/n + \delta_c B_{\alpha,c} + \epsilon C_{\alpha,c}},$$

The different constants in are defined by $L = \max_{c \in [C]} \sum_{d=1}^{D} a_{c,d} \nu(d)$, $\delta_c = \frac{\sum_{d=1}^{D} \frac{a_{c,d}}{e} \nu(d)}{n}$, $K_\alpha = (c\alpha + \epsilon)e^{c\alpha}/c$, ϵ as defined in Lemma 35 and c in Lemma 31, $U_c = \sum_{d=1}^{D} (1 - e^{-a_{c,d}b_c}) \nu(d)$, $A_{\alpha,c} = U_c(U_c^2 + \frac{14U_c}{3} + 2K_\alpha)$, $B_{\alpha,c} = 2U_c^2 + 4L\delta_c + 12K_\alpha$, $C_{\alpha,c} = 2U_c^2 + 4L\epsilon + 8K_\alpha$.

The proof of Theorem 26 is structured as follows:

- We first characterize the drift of the process M_c.

- Next, we verify that M_c satisfies the assumptions required by Theorem 1 in [45].

- Finally, we define the associated differential inclusion and apply Theorem 4 from [45] to derive an explicit rate of convergence.

The following lemma computes the drift of the process M_c for $c \in [C]$, defined as the conditional expectation $\mathbb{E}[M_c(t+1) - M_c(t)|B, \mathbf{M}(t)]$, where $B = (b_1, \ldots, b_C)$ and $\mathbf{M}(t) = (M_1(t), \ldots, M_C(t))$.

Lemma 29. *For $c \in [C]$,*

$$\mathbb{E}[M_c(t+1) - M_c(t)|B, \mathbf{M}(t)] = H_{c,b_c,n}(M_c(t))\mathbb{1}_{\{\max_{k \in [C]} H_{k,b_k,n}(M_k(t)) = H_{c,b_c,n}(M_c(t))\}}$$

where $H_{c,b_c,n}(x) = \sum_{d=1}^{D} \left(1 - \left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - x}\right)\nu(d)$.

Proof. For $c \in [C]$, as defined previously, $M_c(t)$ is the number of matched nodes in the class c, for $t \in [T]$, $M_c(t)$ follows the dynamics,

$$M_c(t+1) = M_c(t) + \mathbb{1}_{\{\exists u \in \mathcal{N}_c(t) \text{ s.t } c_t^* = c \text{ and } m_u(t+1)=1\}}.$$

Let c_t^* denote the class selected by the `Ex-ante Balance` algorithm. We define the expected one-step change of the process $M_c(t)$, for each $c \in [C]$ and $t \in [T]$, as follows:

$$\mathbb{E}[M_c(t+1) - M_c(t)|n, \mathbf{M}(t)] \tag{8.18}$$
$$= \mathbb{P}(\exists u \in \mathcal{N}_c(t) \text{ s.t } c_t^* = c \text{ and } m_u(t+1) = 1|\mathbf{M}(t), B) \tag{8.19}$$
$$= \sum_{d=1}^{D} \mathbb{P}(\exists u \in \mathcal{N}_c(t) \text{ s.t } c_t^* = c \text{ and } m_u(t+1) = 1|\mathbf{M}(t), B, d(t+1) = d)\nu(d)$$
$$\tag{8.20}$$
$$= \sum_{d=1}^{D} \mathbb{P}(\exists u \in \mathcal{N}_c(t) \text{ s.t } m_u(t+1) = 1|\mathbf{M}(t), B, d(t+1) = d, c_t^* = c)\nu(d)$$
$$\mathbb{P}(c_t^* = c|\mathbf{M}(t), R, d(t+1) = d) \tag{8.21}$$
$$= H_{c,b_c,n}(M_c(t))\mathbb{1}_{\{\max_{k \in [C]} H_{k,b_k,n}(M_k(t)) = H_{c,b_c,n}(M_c(t))\}}. \tag{8.22}$$

where $H_{c,b_c,n}(x) = \sum_{d=1}^{D} \left(1 - \left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - x}\right)\nu(d)$. $\qquad \square$

The next lemma defines a martingale difference sequence based on the process M_c, and proves that its second moment is bounded.

Lemma 30. *Let $c \in [C]$ and $t \in [T]$. Define the process $Q_c(t+1) = M_c(t+1) - M_c(t) - \mathbb{E}[M_c(t+1) - M_c(t)|\mathbf{M}(t), B]$. Then $(Q_c(t))_{t \in [T]}$ is a martingale difference sequence with respect to the filtration generated by $\mathbf{M}(t)$. Moreover, there exists a constant $b > 0$ such that:*

$$\mathbb{E}[Q_c(t+1)|\mathbf{M}(t), B] = 0,$$

and

$$\mathbb{E}[|Q_c(t+1)|^2|\mathbf{M}(t), B] \leq b.$$

Proof. By direct computation, we obtain

$$\mathbb{E}[Q_c(t+1)|\mathbf{M}(t), B] = 0.$$

Furthermore, from the definition of the matching process, for all $c \in [C]$ we have

$$|M_c(t+1) - M_c(t)| \leq 1, \quad \forall t \in [T].$$

This implies that the second moment of $Q_c(t+1)$ is bounded. $\qquad\square$

The next result proves the first assumption of Theorem 1 in [45].

Lemma 31. *For $c \in [C]$ and $t \in [T]$, let $\mathcal{H}_{c,b_c,n}(y) = H_{c,b_c,n}(y)\mathbb{1}_{\{\max\limits_{k \in [C]} H_{k,b_k,n}(y) = H_{c,b_c,n}(y)\}}$, it satisfies,*

$$\forall y \leq nb_c, |\mathcal{H}_{c,b_c,n}(y)| \leq c(1 + |y|).$$

with $\alpha_c = \ln\left(1 - \frac{\min_{j \in [D]}(a_{c,j})}{n}\right), c = \max(|1 - e^{\alpha_c nb_c}|, |\alpha_c e^{\alpha_c nb_c}|).$

Proof.

$$
\begin{aligned}
|\mathcal{H}_{c,b_c,n}(y)| &\leq \left| \sum_{d=1}^{D} \left(1 - \left(1 - \frac{a_{c,d}}{n} \right)^{nb_c - y} \right) \nu(d) \right| \\
&= \left| 1 - \sum_{d=1}^{D} e^{\ln\left(1 - \frac{a_{c,d}}{n}\right)(nb_c - y)} \nu(d) \right| \\
&\leq \left| 1 - \sum_{d=1}^{D} e^{\ln\left(1 - \frac{\min_j(a_{c,j})}{n}\right)(nb_c - y)} \nu(d) \right| \\
&\leq \left| 1 - e^{\ln\left(1 - \frac{\min_j(a_{c,j})}{n}\right)(nb_c - y)} \right| \\
&\leq \left| 1 - e^{\alpha_c nb_c}(1 + \alpha_c y) \right| \\
&\leq \left| 1 - e^{\alpha_c nb_c} \right| + \left| \alpha_c e^{\alpha_c nb_c} y \right| \\
&\leq c(1 + |y|).
\end{aligned}
$$

with $\alpha_c = \ln\left(1 - \frac{\min_j(a_{c,j})}{n}\right), c = \max(|1 - e^{\alpha_c n b_c}|, |\alpha_c e^{\alpha_c n b_c}|)$. $\qquad\square$

The following technical lemma provides a bound on the distance between $H_{c,b_c,n}$ and its limit as n becomes large.

Lemma 32. *For $c \in [C]$,*

$$\left|\sum_{d=1}^{D}\left(1 - e^{-a_{c,d}(b_c - M_c(t)/n)}\right)\nu(d) - \left(1 - \left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - M_c(t)}\right)\nu(d)\right| \leq \sum_{d=1}^{D}\frac{a_{c,d}}{ne}\nu(d).$$

Proof. Let $A = \left|\sum_{d=1}^{D}\left(1 - e^{-a_{c,d}(b_c - M_c(t)/n)}\right)\nu(d) - \left(1 - \left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - M_c(t)}\right)\nu(d)\right|$,

$$A \leq \sum_{d=1}^{D}\left|\left(1 - e^{-a_{c,d}(b_c - M_c(t)/n)}\right)\nu(d) - \left(1 - \left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - M_c(t)}\right)\nu(d)\right|$$

$$\leq \sum_{d=1}^{D}\left|e^{-a_{c,d}(b_c - M_c(t)/n)}\nu(d) - \left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - M_c(t)}\nu(d)\right|$$

$$\leq \sum_{d=1}^{D}\frac{a_{c,d}}{ne}\nu(d) \qquad\qquad \text{(Lemma 27)}.$$

$\qquad\square$

The following lemma shows that the function f_{c,b_c}, defined as the limit of $H_{c,b_c,n}$ as $n \to \infty$ and given by $f_{c,b_c}(x) = \sum_{d=1}^{D}\left(1 - e^{-a_{c,d}(b_c - x)}\right)\nu(d)$, is L_c-Lipschitz continuous.

Lemma 33. *(Lipschitz condition) For $c \in [C]$, the function $f_{c,b_c}(x) = \sum_{d=1}^{D}(1 - e^{-a_{c,d}(b_c - x)})\nu(d)$ is Lipschitz continuous with constant $L_c = \sum_{d=1}^{D}a_{c,d}\nu(d)$.*

Proof. Let x, y such that $x \leq b_c$ and $y \leq b_c$ for all $c \in [C]$,

$$|f_{c,b_c}(x) - f_{c,b_c}(y)| = \left|\sum_{d=1}^{D}(e^{-a_{c,d}(b_c - y)} - e^{-a_{c,d}(b_c - x)})\nu(d)\right|$$

By mean value Theorem, there exists $\xi \in (x, y)$ such that we have,

$$|f_{c,b_c}(x) - f_{c,b_c}(y)| \leq |x - y|\sum_{d=1}^{D}e^{-a_{c,d}(b_c - \xi)}a_{c,d}\nu(d)$$

$$\leq |x - y|\sum_{d=1}^{D}a_{c,d}\nu(d).$$

$\qquad\square$

The two following lemma provides a technical bound essential for deriving the explicit rate of convergence in Theorem 26. For $t \in [T]$ let,

$$V_c(t) = \frac{1}{n} \sum_{k=0}^{t} Q_c(k+1).$$

and

$$\frac{M_c(t+1)}{n} = \frac{M_c(0)}{n} + \frac{1}{n} \sum_{l=0}^{t} \mathcal{H}_{c,b_c,n}(M_c(t)) + \frac{1}{n} \sum_{l=0}^{t} Q_c(l+1) \qquad (8.23)$$

$$= \frac{1}{n} \sum_{l=0}^{t} \mathcal{H}_{c,b_c,n}(M_c(t)) + \frac{1}{n} \sum_{l=0}^{t} Q_c(l+1). \qquad (8.24)$$

Lemma 34. *For all $T, n > 0$, and for all $\epsilon > 0$,*

$$\mathbb{P}\left(\sup_{0 \leq k \leq T} |V_c(k)| \geq \epsilon \right) \leq \frac{Tb}{n^2 \epsilon^2}.$$

Proof. Since $\mathbb{E}\left[Q_c(t+1)|\mathbf{M}(t), B\right] = 0$ and $\mathbb{E}\left[|Q_c(t+1)|^2|\mathbf{M}(t), B\right] \leq b$, we have $\mathbb{E}[|V_c(t)|^2] \leq \frac{tb}{n^2} \leq \frac{Tb}{n^2}$ for all $t \leq T$. Applying Kolmogorov's inequality (maximal inequality) for martingales to the martingale V leads to the bound of the lemma. $\square$

Lemma 35. *For $c \in [C]$, let M_c be defined as in Equation (8.23) with $|\mathcal{H}_{c,b_c,n}(y)| \leq c(1 + |y|)$. Let y denote the solution of the differential equation associated with F that is defined in Equation (8.26).*

If we denote by $\epsilon := \sup_{l \leq t} |V_c(l)|$, then

$$\max \left\{ \sup_{0 \leq t \leq T} |M_c(t)|, \sum_{0 \leq \tau \leq T/n} |m(\tau)| \right\} \leq K_\alpha.$$

with $K_\alpha = (c\alpha + \epsilon)\, e^{c\alpha}/c$ with c as defined in Lemma 31.

Proof. By definition of M_c in Equation (8.23) and *Lemma* 34, we have,

$$|M_c(t+1)/n| \leq \frac{1}{n} \sum_{l=0}^{t} c(1 + |M_c(l)|) + \epsilon$$

$$= \frac{tc}{n} + \epsilon + \frac{c}{n} \sum_{l=1}^{t} |M_c(l)|$$

$$\leq (\frac{Tc}{n} + \epsilon)e^{cT/n}/c.$$

The final inequality follows from the discrete Gronwall's lemma. Substituting $T = \alpha n$ then yields the desired result. $\square$

The next lemma proves that F defined in Theorem 26 is upper semicontinuous,

Lemma 36. *(Upper semicontinuous)* *Let* $f_{c,b_c} : \mathbb{R} \to \mathbb{R}$ *be continuous for each* $c \in [C]$, *and define the set-valued map* $F(m) = \text{conv}\left(f_{c,b_c}(m_c)e_c \mid c \in \arg\max_{j \in [C]} f_{j,b_j}(m_j)\right)$ *where* e_c *is the c-th standard basis vector in* $\mathbb{R}^C$. *Then* F *is upper semicontinuous as a set-valued map from* $\mathbb{R}^C$ *to subsets of* $\mathbb{R}^C$.

Proof. Let $(m^{(n)})_{n \in \mathbb{K}} \subset \mathbb{R}^C$ be a sequence such that $m^{(n)} \to m$ as $n \to \infty$, meaning:

$$\forall c \in [C], \quad m_c^{(n)} \to m_c.$$

Let $x_n \in F(m^{(n)})$ be such that $x_n \to x \in \mathbb{R}^C$. We aim to show that $x \in F(m)$.

Each $x_n \in F(m^{(n)})$ belongs to the convex hull:

$$F(m^{(n)}) = \text{conv}\left(f_{c,b_c}(m_c^{(n)})e_c \mid c \in \arg\max_{j \in [C]} f_{j,b_j}(m_j^{(n)})\right).$$

Because the index set $[C]$ is finite, there are only finitely many possible argmax sets. Thus, we may extract a subsequence (still denoted by n for simplicity) such that for some fixed $A \subseteq [C]$,

$$\arg\max_{j \in [C]} f_{j,b_j}(m_j^{(n)}) = A \quad \text{for all large } n.$$

Since each f_{j,b_j} is continuous and $m_j^{(n)} \to m_j$, we have $f_{j,b_j}(m_j^{(n)}) \to f_{j,b_j}(m_j)$ for all j, and hence:

$$\max_{j \in [C]} f_{j,b_j}(m_j^{(n)}) \to \max_{j \in [C]} f_j(m_j).$$

Therefore, for every $i \in A$,

$$f_{c,b_c}(m_c) = \max_{j \in [C]} f_{j,b_j}(m_j),$$

i.e., $A \subseteq \arg\max_{j \in [C]} f_{j,b_j}(m_j)$.

Now, each x_n is a convex combination of the vectors $f_{c,b_c}(m_c^{(n)})e_c$ with $c \in A$, and by continuity:

$$f_{c,b_c}(m_c^{(n)}) \to f_{c,b_c}(m_c), \quad \text{so} \quad f_{c,b_c}(m_c^{(n)})e_c \to f_{c,b_c}(m_c)e_c.$$

Thus, $x_n \to x$ implies that x lies in the convex hull of the limit points:

$$x \in \text{conv}\left(f_{c,b_c}(m_c)e_c \mid c \in A\right) \subseteq F(m),$$

since $A \subseteq \arg\max_{j \in [C]} f_{j,b_j}(m_j)$.

Hence, every limit point of a converging sequence (x_n) with $x_n \in F(m^{(n)})$ lies in $F(m)$, which proves that F is upper semicontinuous. $\square$

With all the preparatory results established — in particular, Lemmas 30 and 31, which shows that $M_c(t)$ satisfies the assumptions of Theorem 1 in [45] — we are now ready to prove Theorem 26.

Proof. For all $c \in [C]$ and $\tau \in [T/n]$, we consider the process $\tilde{M}_c(\tau)$ defined by,

$$\tilde{M}_c(\tau + 1/n) = \tilde{M}_c(\tau) + \frac{1}{n}\mathcal{H}_{c,b_c,n}(\tilde{M}_c(\tau)) + \tilde{Q}_c(\tau + 1/n). \qquad (8.25)$$

where $\tilde{Q}_c(\tau) = \frac{Q_c(\tau n)}{n}$ with Q_c as defined in Lemma 30, $\mathcal{H}_{c,b_c,n}$ as defined in Lemma 31, and let $\tilde{\mathbf{M}}(\tau) = (\tilde{M}_1(\tau), \dots, \tilde{M}_C(\tau))$.

When $n \to \infty$, and $t \in [T]$, the function $H_{c,b_c,n}$ converges to:

$$H_{c,b_c,n}(M_c(t)) \to \sum_{d=1}^{D} \left(1 - e^{-a_{c,d}(b_c - \frac{M_c(t)}{n})}\right)\nu(d) = f_{c,b_c}(M_c(t)/n).$$

Let $g(\tilde{\mathbf{M}}) = \left(f_{c,b_c}(\tilde{M}_c)\mathbf{1}\{\max_{k \in [C]} f_{k,b_k}(\tilde{M}_k) = f_{c,b_c}(\tilde{M}_c)\}\right)_{c \in [C]}$ be the drift vector. According to Lemma 31, each element of the drift vector satisfies the first assumption of Theorem 1 in [45]. Moreover, according to Lemma 30, $\tilde{Q}_c(\tau + 1/n)$ satisfies the second assumption of Theorem 1 in [45]. Since $\tilde{M}_c(0) = 0$, based on Theorem 1 in [45], for all $\tau \in [T/n]$, $\tilde{\mathbf{M}}(\tau)$ converges to $m(\tau)$, where m is the solution of the following differential inclusion:

$$\dot{m}(\tau) \in F(m(\tau)). \qquad (8.26)$$

where $F(m) = \text{conv}\left(f_{c,b_c}(m_c)e_c \mid c \in \arg\max_{j \in [C]} f_{j,b_j}(m_j)\right)$, and conv denotes the convex hull.

According to Lemma 36 and Lemma 31, the differential inclusion Equation (8.26) admits at least one solution m. To establish the uniqueness of this solution, it is sufficient to show that the set-valued map F is one-sided Lipschitz. By Lemma 33, each function f_{c,b_c} is L_c-Lipschitz continuous for all $c \in [C]$, and let $L = \max_{c \in [C]} L_c$. Let $s, s' \in \mathbb{R}^C$ and suppose $z \in F(s)$, $z' \in F(s')$. Then:

$$\langle z - z', s - s' \rangle = \sum_{i=1}^{C}(s_i - s'_i)\left(f_{i,b_i}(s_i) - f_{i,b_i}(s'_i)\right) \leq \sum_{i=1}^{C} L_i(s_i - s'_i)^2 \leq L\|s - s'\|^2.$$

Thus F is one-sided Lipschitz with constant L, which guarantees the uniqueness of the solution m to the differential inclusion Equation (8.26). To get the explicit rate of the convergence of $\tilde{M}_c$ to m_c, we use Theorem 4 of [45]. According to Lemma 30, for $t \in [T]$, $\mathbb{E}[|Q_c(t + 1)|^2|\mathbf{M}(t), B] \leq b$ and F is one sided Lipschitz with constant $L = \max_{c \in [C]} L_c$ where L_c is defined in Lemma 33. Thus according to theorem 4 in [45], taking $\delta_c = \sum_{d=1}^{D} \frac{a_{c,d}}{ne}\nu(d)$, K_α as defined in Lemma 35 and $U_c = \sup_{0 \leq t \leq T} f_{c,b_c}(M_c(t))$, taking $U_c = \sum_{d=1}^{D}(1 - e^{-a_{c,d}b_c})\nu(d)$ we define $A_{\alpha,c} =$

$U_c(U_c^2 + \frac{14U_c}{3} + 2K_\alpha), B_{\alpha,c} = 2U_c^2 + 4L\delta + 12K_\alpha, C_{\alpha,c} = 2U_c^2 + 4L\epsilon + 8K_\alpha.$ We have,

$$\mathbb{P}\left(\sup_{0 \leq t \leq T} \left|\frac{M_c(t)}{n} - m_c(t/n)\right| \geq \min(\alpha, e^{L\alpha}/\sqrt{2L})\sqrt{A_{\alpha,c}/n + \delta_c B_{\alpha,c} + \epsilon C_{\alpha,c}}\right) \leq \frac{bT}{n^2\epsilon^2}.$$

$\square$

8.C.2 Proof of Theorem 27

Theorem 27. *For any $c \in \mathcal{C}$, the function*

$$m_c^* : t \mapsto (b_c - \boldsymbol{\beta}_c^{(k_t)}) + \left(f_{c,\boldsymbol{\beta}^{(k_t)}}^{-1}(F_{k_t,\boldsymbol{\beta}^{(k_t)}}(\mu_{k_t,\boldsymbol{\beta}^{(k_t)}}(t - t_{k_t})))\right)_+ .$$

where $k_t = \max\{k \in [C] : t > t_k\}$, is the unique solution of the differential inclusion defined in Theorem 26.

We need a few lemmas before going into the proof of the Theorem 27.

Lemma 37. $\forall t \in \mathbb{R}_+, \forall c \in \mathcal{C},$

$$\mu_c(t) = \begin{cases} b_c - \boldsymbol{\beta}_c^{(k_t)} + f_{c,\boldsymbol{\beta}^{(k_t)}}^{-1}(F_{k_t,\boldsymbol{\beta}^{(k_t)}}(\mu_{k_t,\boldsymbol{\beta}^{(k_t)}}(t - t_{k_t}))) & \text{if } c \leq k_t, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. For $t \geq 0$ and $c > k_t$, $(b_c - \boldsymbol{\beta}_c^{(k_t)}) = 0$ by def of $\boldsymbol{\beta}_c^{(k_t)}$. Then

By definition of k_t, $t \in [t_{k_t}, t_{k_t+1})$.

$$\begin{aligned} t < t_{k_t+1} &\Leftrightarrow t < t_{k_t} + \mu_{k_t,\boldsymbol{\beta}^{(k_t)}}^{-1}\left(F_{k-1,\boldsymbol{\beta}^{(k_t)}}^{-1}\left(f_{k_t+1,\boldsymbol{\beta}^{(1)}}(0)\right)\right) \\ &\Leftrightarrow \mu_{k_t,\boldsymbol{\beta}^{(k_t)}}(t - t_{k_t}) < F_{k_t,\boldsymbol{\beta}^{(k_t)}}^{-1}\left(f_{k_t+1,\boldsymbol{\beta}^{(k_t)}}(0)\right) & (\mu_{k_t,\boldsymbol{\beta}^{(k_t)}} \text{ is increasing}) \\ &\Leftrightarrow F_{k_t,\boldsymbol{\beta}^{(k_t)}}(\mu_{k_t,\boldsymbol{\beta}^{(k_t)}}(t - t_{k_t})) > f_{k_t+1,\boldsymbol{\beta}^{(k_t)}}(0) & (F_{k_t,\boldsymbol{\beta}^{(k_t)}} \text{ is decreasing}) \\ &\Leftrightarrow \forall c > k_t, \ F_{k_t,\boldsymbol{\beta}^{(k_t)}}(\mu_{k_t,\boldsymbol{\beta}^{(k_t)}}(t - t_{k_t})) > f_{c,\boldsymbol{\beta}^{(k_t)}}(0) & (F_{k_t,\boldsymbol{\beta}^{(k_t)}} \text{ is decreasing}) \\ &\Leftrightarrow \forall c > k_t, \ f_{c,\boldsymbol{\beta}^{(k_t)}}^{-1}(F_{k_t,\boldsymbol{\beta}^{(k_t)}}(\mu_{k_t,\boldsymbol{\beta}^{(k_t)}}(t - t_{k_t}))) < 0 & (f_{c,\boldsymbol{\beta}^{(k_t)}}^{-1} \text{ is decreasing}). \end{aligned}$$

$\square$

Lemma 38. $\forall t \in \mathbb{R}_+, \ \sum_{c \in \mathcal{C}} \mu_c(t) = \|b - \boldsymbol{\beta}^{(k_t)}\|_1 + \mu_{k_t,\boldsymbol{\beta}^{(k_t)}}(t - t_{k_t}).$

Proof. By definition, for any $t \geq 0$, $\boldsymbol{\beta}_c^{(k_t)} \leq b_c$, so $\sum_{c \in \mathcal{C}}(b_c - \boldsymbol{\beta}_c^{(k_t)}) = \|b - \boldsymbol{\beta}^{(k_t)}\|_1.$

Let $k \in [C]$ be a phase.

$$\forall t \in [t_k, t_{k+1}), \quad \sum_{c \in \mathcal{C}} \left(f_{c,\boldsymbol{\beta}^{(k)}}^{-1} (F_{k,\boldsymbol{\beta}^{(k)}} (\mu_{k,\boldsymbol{\beta}^{(k)}} (t - t_k))) \right)_+ = \mu_{k,\boldsymbol{\beta}^{(k)}} (t - t_k)$$

$$\Leftrightarrow \forall t \in [0, t_{k+1} - t_k], \quad \sum_{c \leq k} f_{c,\boldsymbol{\beta}^{(k)}}^{-1} (F_{k,\boldsymbol{\beta}^{(k)}} (\mu_{k,\boldsymbol{\beta}^{(k)}} (t))) = \mu_{k,\boldsymbol{\beta}^{(k)}} (t)$$

$$\Leftrightarrow \forall s \in \left[\mu_{k,\boldsymbol{\beta}^{(k)}}^{-1} (0), \mu_{k,\boldsymbol{\beta}^{(k)}}^{-1} (t_{k+1} - t_k) \right], \quad \sum_{c \leq k} f_{c,\boldsymbol{\beta}^{(k)}}^{-1} (F_{k,\boldsymbol{\beta}^{(k)}} (s)) = s$$

$$\Leftrightarrow \forall u \in \left[F_{k,\boldsymbol{\beta}^{(k)}}^{-1} \left(\mu_{k,\boldsymbol{\beta}^{(k)}}^{-1} (t_{k+1} - t_k) \right), F_{k,\boldsymbol{\beta}^{(k)}}^{-1} \left(\mu_{k,\boldsymbol{\beta}^{(k)}}^{-1} (0) \right) \right], \quad \sum_{c \leq k} f_{c,\boldsymbol{\beta}^{(k)}}^{-1} (u) = F_{k,\boldsymbol{\beta}^{(k)}}^{-1} (u).$$

The last line is true by definition of $F_{k,\boldsymbol{\beta}^{(k)}}$. $\square$

Lemma 39. $\forall t \in \mathbb{R}_+, \forall c \in \mathcal{C}$

$$f_c(\mu_c(t)) = \begin{cases} F_{k_t,\boldsymbol{\beta}^{(k_t)}} (\mu_{k_t,\boldsymbol{\beta}^{(k_t)}} (t - t_{k_t})) & \text{if } c \leq k_t, \\ f_{c,nb_c}(0) & \text{otherwise.} \end{cases}$$

and $f_c(\mu_c(t)) \leq f_{k_t}(\mu_{k_t}(t))$.

Proof. For any $t \in \mathbb{R}_+$ and $c \in \mathcal{C}$,

$$\begin{aligned} f_c(\mu_c(t)) &= f_{c,b_c}(\mu_c(t)) \\ &= f_{c,\boldsymbol{\beta}_c^{(k_t)}} \left((\boldsymbol{\beta}_c^{(k_t)} - b_c + \mu_c(t)) \right) \\ &= f_{c,\boldsymbol{\beta}_c^{(k_t)}} \left(\left(f_{c,\boldsymbol{\beta}^{(k_t)}}^{-1} (F_{k_t,\boldsymbol{\beta}^{(k_t)}} (\mu_{k_t,\boldsymbol{\beta}^{(k_t)}} (t - t_{k_t}))) \right)_+ \right). \end{aligned}$$

Lemma 37 allows to conclude. $\square$

With all the preparatory lemmas established, we now proceed to the proof.

Proof. The result is proven by induction. For $k \in [C]$, the induction hypothesis is

$(\boldsymbol{A_k})$ $\forall t \leq t_k$, μ is such that $\dot{\mu}(t) \in F(\mu(t))$.

For $t \in [t_k, t_{k+1})$, we have $k = k_t$. Thus, restricted to the interval $[t_k, t_{k+1})$,

$$\dot{\mu} \in F(\mu) \Leftrightarrow \dot{\mu} \in \text{conv}\left(f_c(\mu_c)e_c \mid c \in \arg\max_{k \in [C]} f_k(\mu_k) \right) \tag{8.27}$$

$$\Leftrightarrow \dot{\mu} \in F_{k_t, \boldsymbol{\beta}^{(k_t)}}(\mu_{k_t, \boldsymbol{\beta}^{(k_t)}}(t - t_{k_t}))\text{conv}\left(e_c \mid c \in \arg\max_{k \in [C]} f_k(\mu_k) \right) \tag{8.28}$$

$$\Leftrightarrow \left(\sum_{c \le k} \dot{\mu}_c \le F_{k_t, \boldsymbol{\beta}^{(k_t)}}(\mu_{k_t, \boldsymbol{\beta}^{(k_t)}}(t - t_{k_t})) \right) \text{ and } (\forall c > k, \dot{\mu}_c = 0) \tag{8.29}$$

$$\Leftrightarrow \left(\dot{\mu}_{k_t, \boldsymbol{\beta}^{(k_t)}}(t - t_{k_t}) \le F_{k_t, \boldsymbol{\beta}^{(k_t)}}(\mu_{k_t, \boldsymbol{\beta}^{(k_t)}}(t - t_{k_t})) \right) \text{ and } (\forall c > k, \dot{\mu}_c = 0). \tag{8.30}$$

Going from (8.27) to (8.28) comes from Lemma 39. Going from (8.28) to (8.29) comes from the fact that the convex hull of a set of basis vectors is the L_1 ball restricted to the corresponding subspace. Going from (8.29) to (8.30) comes from applying Lemma 38. Equation (8.30) is TRUE by definition of $\mu_{k_t, \boldsymbol{\beta}^{(k_t)}}(t - t_{k_t})$ for the first term and by applying Lemma 37 for the second term. $\qquad\square$

8.C.3 Ex-post Balance

While the `Ex-ante Balance` policy improves upon the purely compatibility-driven `Myopic` approach by considering the probability of successful matches, it still relies on an expected availability model—it selects the class that likely has unmatched nodes, without verifying their presence. This can lead to wasted opportunities when the chosen class ends up being empty. To address this limitation, we introduce the `Ex-post Balance` policy, which takes an additional, more grounded step. Instead of only maximizing the expected match probability, `Ex-post Balance` ensures that the selected class actually contains at least one available node at the time of decision. This additional check prevents the algorithm from targeting unavailable options, making it more efficient. The formal procedure is detailed in Algorithm 17.

Algorithm 17: `Ex-post Balance`

Output: Updated matching $M(t)$
1 **for** $t \in [T]$ **do**
2   Choose $c_t = \arg\max_{c \in C(t)} \sum_{d=1}^{D} (1 - (1 - \frac{a_{c,j}}{n})^{nb_c - M_c(t)})\nu(d)$.
3   **if** $\mathcal{F}_{c_t}(t) = \varnothing$ **then**
4          $M(t) = M(t-1)$.
5   **else**
6          $M(t) = M(t-1) \cup \{(n_t, t)\}$ for $n_t \sim \text{unif}(\mathcal{F}_{c_t}(t))$.

The `Ex-post Balance` policy introduces an additional layer of selectivity compared to `Ex-ante Balance` by explicitly verifying the presence of available nodes in the selected class before committing to a match. While this refinement improves

matching efficiency, it also increases the non-smoothness of the system's dynamics. In particular, the policy induces more discontinuities—not only due to abrupt switches in the maximization rule, but also because the feasibility of a class now depends on the cardinality of its unmatched nodes at each step. As a result, the evolution of the matching process under `Ex-post Balance` cannot be captured by the same differential inclusion used for `Ex-ante Balance`. Instead, it follows a more constrained inclusion, where the feasible directions of evolution depend explicitly on whether a class still has unmatched capacity. The following theorem formalizes this behavior, showing that, with high probability, the normalized matching sizes converge to the solution of a new differential inclusion that incorporates both compatibility and real-time availability constraints.

Theorem 30. *For all $c \in [C]$, the matching size built by `Ex-post Balance` satisfies for all $t \in [T]$ with probability $1 - \frac{b\alpha}{n\epsilon}$,*

$$\left| \frac{M_c(t)}{n} - m_c(t/n) \right| \leq \min(\alpha, e^{L\alpha}/\sqrt{2L})\sqrt{A_{\alpha,c}/n + \delta_c B_{\alpha,c} + \epsilon C_{\alpha,c}}.$$

where m_c is the $c-$th coordinate of m the unique solution of the differential inclusion $\dot{m} \in G(m)$ and $G(m) = conv\left(f_{c,b_c}(m_c)e_c | c = \arg\max_{k \in [1,C]} f_{k,b_k}(m_k), b_c > m_c\right)$ with e_c a basis vector of $\mathbb{R}^C$, $L = \max_{c \in [1,C]} \sum_{d=1}^{D} a_{c,d}\nu(d)$, $\delta_c = \sum_{d=1}^{D} \frac{a_{c,d}}{ne}\nu(d)$, $K_\alpha = (c\alpha + \epsilon)e^{c\alpha}/c$ with ϵ as defined in Lemma 35 and c in Lemma 31, $U_c = \sum_{d=1}^{D}(1 - e^{-a_{c,d}b_c})\nu(d)$, $A_{\alpha,c} = U_c(U_c^2 + \frac{14U_c}{3} + 2K_\alpha)$, $B_{\alpha,c} = 2U_c^2 + 4L\delta_c + 12K_\alpha$, $C_{\alpha,c} = 2U_c^2 + 4L\epsilon + 8K_\alpha$.

As in Section 8.C.1, our objective here is to approximate the matching size M_c produced by `Ex-post Balance` for each class $c \in [C]$. Following the same approach as before, the first step involves computing the drift of the process M_c.

Lemma 40. *For $c \in [C]$,*

$$\mathbb{E}[M_c(t+1) - M_c(t)|B, \mathbf{M}(t)] = H_{c,b_c,n}(M_c(t))\mathbb{1}_{\{\max_{k \in [C]} H_{k,b_k,n}(M_k(t)) = H_{c,b_c,n}(M_c(t))\}}\mathbb{1}_{\{nb_c > M_c(t)\}}.$$

Proof.

$$\mathbb{E}[M_c(t+1) - M_c(t)|B, \mathbf{M}(t)] = \mathbb{P}(\exists u \in \mathcal{N}_c(t) \backslash \mathcal{M}_c(t)(t), c^* = c, m_u(t+1) = 1|B, \mathbf{M}(t))$$

$$= \sum_{d=1}^{D} \mathbb{P}(\exists u \in \mathcal{N}_c(t) \backslash \mathcal{M}_c(t), c^* = c, m_u(t+1) = 1|\mathbf{M}(t), B, d(t+1) = d)\nu(d)$$

$$= \sum_{d=1}^{D} \mathbb{P}(\exists u \in \mathcal{N}_c(t) \backslash \mathcal{M}_c(t), m_u(t+1) = 1|\mathbf{M}(t), B, d(t+1) = d, c^* = c)\nu(d)$$

$$\mathbb{P}(c^* = c|\mathbf{M}(t), B, d(t+1) = d)$$

$$= H_{c,b_c,n}(M_c(t))\mathbb{1}_{\{\max_{k \in [C]} H_{k,b_k,n}(M_k(t)) = H_{c,b_c,n}(M_c(t))\}}\mathbb{1}_{\{nb_c > M_c(t)\}}.$$

where $H_{c,b_c,n}(x) = \sum_{d=1}^{D}\left(1 - \left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - x}\right)\nu(d)$. $\square$

Proof. For all $c \in [C]$ and $\tau \in [T/n]$, we consider the process $\tilde{M}_c(\tau)$ defined by,

$$\tilde{M}_c(\tau + 1/n) = \tilde{M}_c(\tau) + \frac{1}{n}\mathcal{H}_{c,b_c,n}(\tilde{M}_c(\tau)) + \tilde{Q}_c(\tau + 1/n).$$

where $\tilde{Q}_c(\tau) = \frac{Q_c(\tau n)}{n}, Q_c(t+1) = M_c(t+1) - M_c(t) - \mathbb{E}[M_c(t+1) - M_c(t)|\mathbf{M}(t), B]$, $\mathcal{H}_{c,b_c,n}(y) = H_{c,b_c,n}(M_c(t))\mathbb{1}_{\{\max_{k \in [C]} H_{k,b_k,n}(M_k(t))=H_{c,b_c,n}(M_c(t))\}}\mathbb{1}_{\{nb_c > M_c(t)\}}$, and let $\tilde{\mathbf{M}}(\tau) = (\tilde{M}_1(\tau), \ldots, \tilde{M}_C(\tau))$.

When $n \to \infty$, and $t \in [T]$, the function $H_{c,b_c,n}(M_c(t))$ converges to:

$$H_{c,b_c,n}(M_c(t)) \to \sum_{d=1}^{D}\left(1 - e^{-a_{c,d}(b_c - \frac{M_c(t)}{n})}\right)\nu(d) = f_{c,b_c}(M_c(t)/n).$$

Let $g(\tilde{M}) = \left(f_{c,b_c}(\tilde{M}_c)\mathbb{1}_{\{\max_{k \in [C]} f_{k,b_k}(\tilde{M}_k)=f_{c,b_c}(\tilde{M}_c)\}}\mathbb{1}_{\{b_c > \tilde{M}_c\}}\right)_{i \in [C]}$ be the drift vector. According to Lemma 31, each element of the drift vector satisfies the first assumption of Theorem 1 in [45]. Moreover, according to Lemma 30, $\tilde{Q}_c(\tau + 1/n)$ satisfies the second assumption of Theorem 1 in [45]. Since $\tilde{M}_c(0) = 0$, based on Theorem 1 in [45], for all $\tau \in [T/n]$, $\tilde{M}(\tau)$ converges to $m(\tau)$, where m is the solution of the following differential inclusion:

$$\dot{m}(\tau) \in G(m(\tau)).$$

where $G(m) = \text{conv}\left(f_{c,b_c}(m_c)e_c \mid c \in \arg\max_{j \in [C]} f_{j,b_j}(m_j), b_c > m_c\right)$, and conv denotes the convex hull.

Following the same results as in Section 8.C.1, one can prove that G is upper semicontinuous and L-one-sided Lipschitz, where $L = \max_{c \in [C]} L_c$. This ensures uniqueness of the solution m. To get the explicit rate of the convergence of $\tilde{M}_c$ to m_c, we use Theorem 4 of [45]. According to Lemma 30, for $t \in [T]$, $\mathbb{E}[|Q_c(t+1)|^2|\mathbf{M}(t), B] \leq b$ and F is one sided Lipschitz with constant $L = \max_{c \in [C]} L_c$ where L_c is defined in Lemma 33. Thus according to theorem 4 in [45], taking $\delta_c = \sum_{d=1}^{D}\frac{a_{c,d}}{ne}\nu(d)$, K_α as defined in Lemma 35 and $U_c = \sup_{0 \leq t \leq T} f_{c,d_c}(M_c(t))$, taking $U_c = \sum_{d=1}^{D}(1 - e^{-a_{c,d}b_c})\nu(d)$ we define $A_{\alpha,c} = U_c(U_c^2 + \frac{14U_c}{3} + 2K_\alpha), B_{\alpha,c} = 2U_c^2 + 4L\delta + 12K_\alpha, C_{\alpha,c} = 2U_c^2 + 4L\epsilon + 8K_\alpha$. We have,

$$\mathbb{P}\left(\sup_{0 \leq t \leq T}\left|\frac{M_c(t)}{n} - m_c(t/n)\right| \geq \min(\alpha, e^{L\alpha}/\sqrt{2L})\sqrt{A_{\alpha,c}/n + \delta_c B_{\alpha,c} + \epsilon C_{\alpha,c}}\right) \leq \frac{bT}{n^2\epsilon}.$$

$\square$

8.C.4 Smoothbalance

Theorem 28. *With probability $1 - \mathcal{O}(n^{1/4} e^{-n^{1/4} \left(\sum_{d=1}^{D} a_{c,d} \right)^3})$, the number of matched nodes in the class $c \in [C]$ satisfies,*

$$M_c(T) = n z_c(T/n) + \mathcal{O}(n^{3/4}).$$

where z_c with $z_i(0) = 0$ and $0 \leq \tau \leq T/n$, is the solution of,

$$\dot{z}_c(\tau) = \sum_{d=1}^{D} \left(1 - e^{-a_{c,d}(b_c - z_c(\tau))} \right)^{1 + \frac{1}{\xi}} \frac{1}{\sum_{k=1}^{C} \left(1 - e^{-a_{k,d}(b_k - z_k(\tau))} \right)^{\frac{1}{\xi}}} \nu(d). \qquad (8.2)$$

To prove Theorem 28, the first step is to establish the one step change of the matching process M_i.

Lemma 41. *For $t \in [T]$, $c \in [C]$ and for any $R = (n_1, \ldots, n_C)$, the expectation of the one-step change of $M_c(t)$, when matching is constructed using the above policy is given by,*

$$\mathbb{E}[M_c(t+1) - M_c(t) | \mathbf{M}(t), R] = F_c(t, M_1(t), \ldots, M_C(t)).$$

where $F_c(t, M_1(t), \ldots, M_C(t))$ is defined by, $F_c(t, M_1(t), \ldots, M_C(t)) =$

$$\sum_{d=1}^{D} \left(1 - \left(1 - \frac{a_{c,d}}{n} \right)^{nb_c - M_c(t)} \right) \nu(d) \frac{\left(1 - \left(1 - \frac{a_{c,d}}{n} \right)^{nb_c - M_c(t)} \right)^{\frac{1}{\xi}}}{\sum_{k=1}^{C} \left(1 - \left(1 - \frac{a_{k,d}}{n} \right)^{nb_k - M_k(t)} \right)^{\frac{1}{\xi}}}.$$

Proof. Same proof as in Lemma 29, by replacing the indicator function with $q_{c,d(t)}$. $\qquad\square$

Lemma 42. *Let $0 \leq \tau \leq \frac{T}{n}$, $0 \leq z_c \leq 1$, and $Z_c(\tau) = \frac{M_c(n\tau)}{n}$, thus, it holds that for $c \in [C]$,*

- $|M_c(t+1) - M_c(t)| \leq 1$ *(Boundness hypothesis).*

- *The function $f_c(\tau, Z_1(\tau), \ldots, Z_C(\tau))$ defined as the limit of F_c when $n \to \infty$, is $L-$Lipschitz (Lipschitz hypothesis).*

- $\exists \delta > 0$ *such that $|\mathbb{E}[M_c(t+1) - M_c(t) | \mathbf{M}(t), R] - F_c(\tau, Z_1(\tau), \ldots, Z_C(\tau))| \leq \delta$ (Trend Hypothesis).*

Proof.

- Boundness hypothesis: $M_c(t)$ is the number of matched nodes at time t in the class $c \in [C]$. By definition of the matching process a node either gets matched or does not. Therefore, for $c \in [C]$ and $t \in [T]$

$$|M_c(t+1) - M_c(t)| \leq 1.$$

- Lipschitz hypothesis: in order to apply Wormald theorem, one need to prove that for any $c \in [C]$, $f_c(\tau, Z_1(\tau), \ldots, Z_C(\tau))$ is a L-Lipschitz function. First notice that f_c is the the sum of product of two functions, $g_j(\tau, Z_1(\tau), \ldots, Z_C(\tau)) = \left(1 - e^{-a_{c,j}(b_c - Z_c(\tau))}\right)^{1+\frac{1}{\xi}} \nu(d)$ and $p_j(\tau, z_1(\tau), \ldots, z_C(\tau)) = \frac{1}{\sum_{k=1}^{C}\left(1 - e^{-a_{k,j}(b_k - Z_k(\tau))}\right)^{\frac{1}{\xi}}}$. Notice that $|g_j| \leq \nu(d)$ and $|p_j| \leq 1$. The function $g_j(\tau, z_1(\tau), \ldots, z_C(\tau))$ is Lipschitz with constant $L_{1,j} = \left(1 + \frac{1}{\xi}\right) e^{a_{c,j}\epsilon}(1 - e^{a_{c,j}\epsilon})^{\frac{1}{\xi}}$ and the function $p_j(\tau, z_1(\tau), \ldots, z_C(\tau))$ is Lipschitz with constant $L_{2,j} = \sum_{k=1}^{C}\left(\frac{1}{\xi}\right) e^{a_{k,j}\epsilon}(1 - e^{a_{k,j}\epsilon})^{\frac{1}{\xi}-1}$ .Thus, the function f_c is Lipschitz with constant $L = \sum_{d=1}^{D} L_{1,j} + L_{2,j}\nu(d)$.

- Trend hypothesis:

$$|\mathbb{E}[S_c(t+1) - S_c(t)|S(t), n] - f_c(\tau, Z_1(\tau), \ldots, Z_C(\tau))|$$

$$\leq \left| \sum_{d=1}^{D} \nu(d) \left(e^{-a_{c,d}(b_c - Z_c(\tau))} - \left(1 - \frac{a_{c,d}}{n}\right)^{n(b_c - Z_c(\tau))} \right) \frac{\left(1 - \left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - Z_c(t)}\right)^{\frac{1}{\xi}}}{\sum_{k=1}^{C}\left(1 - \left(1 - \frac{a_{k,d}}{n}\right)^{nb_k - Z_k(t)}\right)^{\frac{1}{\xi}}} \right|$$

$$\leq \left| \sum_{d=1}^{D} \nu(d) \frac{a_{c,d}}{ne} \right|.$$

$\square$

Now, we can prove Theorem 28.

Proof. According to Lemma 42, The process $Z_c(\tau) = \frac{M_c(\tau n)}{n}$ satisfies the hypotheses of the Wormald theorem [98]. Thus taking $0 \leq z_c \leq 1$, and taking $\beta = 1$ (by definition of the matching process), L as defined in Lemma 42, $\lambda = n^{-1/3} \sum_{d=1}^{D} a_j \nu(d)$, then with probability $1 - \mathcal{O}(n^{1/4} e^{-n^{1/4}\left(\sum_{d=1}^{D} a_{c,d}\right)^3})$,

$$M_c(T) = n z_c(T/n) + \mathcal{O}(n^{3/4}).$$

Where z_c with $z_c(0) = 0$ and $0 \leq \tau \leq T/n$, is the solution of,

$$\dot{z}_c(\tau) = \sum_{d=1}^{D} \left(1 - e^{-a_{c,d}(b_c - z_c(\tau))}\right)^{1+\frac{1}{\xi}} \frac{1}{\sum_{k=1}^{C}\left(1 - e^{-a_{k,d}(b_k - z_k(\tau))}\right)^{\frac{1}{\xi}}} \nu(d). \tag{8.31}$$

$\square$

Solving the differential equation in Theorem 28 is challenging, one approach is to focus on the stationary solution of the differential equation as follows,

Lemma 24. *Equation* (8.3) *has a unique stable stationary solution denoted* $y_c^* = 0$.

Proof. Let $V(y_c) = -\sum_{d=1}^{D}\left(1 - e^{-a_{c,d}y_i}\right)^{1+\frac{1}{\xi}}\frac{1}{\sum_{k=1}^{C}\left(1-e^{-a_{k,d}y_k}\right)^{\frac{1}{\xi}}}\nu(d)$. $V(0) = 0$ and $V(b_c) \leq 0$, then a solution to $V(z_c^*) = 0$ exists. $V(z_c)$ is monotonic, thus the solution is unique. Let $\epsilon_c(t) : \mathbb{N} \to [0,1]$ be a perturbation of the stationary solution $y_c^* = 0$, thus Equation (8.3) becomes,

$$\dot{\epsilon}_c(\tau) = -\sum_{d=1}^{D}\left(1 - e^{-a_{c,d}\epsilon_i(\tau)}\right)^{1+\frac{1}{\xi}}\frac{1}{\sum_{k=1}^{C}\left(1 - e^{-a_{k,d}\epsilon_k(\tau)}\right)^{\frac{1}{\xi}}}\nu(d)$$

$$\dot{\epsilon}_c(\tau) \leq -\frac{1}{C}\sum_{d=1}^{D}\left(1 - e^{-a_{c,d}\epsilon_c(\tau)}\right)^{1+\frac{1}{\xi}}\nu(d)$$

$$\dot{\epsilon}_c(\tau) < \frac{1}{C}(\sum_{d=1}^{D}((1 - e^{-a_{c,d}})\epsilon_c(\tau))^{1+\frac{1}{\xi}}\nu(d)$$

$$\frac{\dot{\epsilon}_c(\tau)}{\epsilon_c(\tau)^{1+\frac{1}{\xi}}} \leq \frac{1}{C}(1 - e^{-a_{c,d}})^{1+\frac{1}{\xi}}\nu(d)$$

$$-\xi\epsilon_c(\tau)^{-\frac{1}{\xi}} + \xi\epsilon_c(0)^{-\frac{1}{\xi}} \leq \tau\frac{1}{C}\sum_{d=1}^{D}(1 - e^{-a_{c,d}})^{1+\frac{1}{\xi}}\nu(d)$$

$$\epsilon_c(\tau)^{-\frac{1}{\xi}} \geq -\frac{\tau}{\xi}\frac{1}{C}\sum_{d=1}^{D}(1 - e^{-a_{c,d}})^{1+\frac{1}{\xi}}\nu(d) + \epsilon_c(0)^{-\frac{1}{\xi}}$$

$$\epsilon_c(\tau) \leq \left(\frac{1}{-\frac{\tau}{\xi}\frac{1}{C}\sum_{d=1}^{D}(1 - e^{-a_{c,d}})^{1+\frac{1}{\xi}}\nu(d) + \epsilon_c(0)^{-\frac{1}{\xi}}}\right)^{\xi}.$$

Thus y_c^* is stable. □

8.D   Regret of ETC-Balance

In this section, we provide the proof of Theorem 29.

Theorem 29. *Let $R(T) = \sum_{i\in\mathcal{C}} M_i(T) - \hat{M}_i(T)$ denote the regret of* `ETC − balance`. *Suppose the exploration phase lasts for $T_{explore} = T^{\frac{q+3}{4}}$, for some $0 < q < 1$. Then the regret satisfies*

$$R(T) = \mathcal{O}(T^{\frac{q+3}{4}}).$$

The proof of Theorem 29 is structured in two main steps:

- We begin by establishing a concentration result for an estimator of $\left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - M_c(t)}$.

- Next, we decompose the regret and derive bounds for each resulting term.

Let $m', m \in [0, nb_c)$ and $\mathcal{V}_m = \{m' \in [0, nb_c) | 1/2 \leq \frac{nb_c - m'}{nb_c - m} \leq 2\}$, we consider a Bernoulli random variable $Y_{c,d,m'}(t)$ with parameter $1 - D_{c,d}(m') := 1 - (1 - a_{c,d}/n)^{nb_c - m'}$, whenever $c(t) = c$, $d(t) = d$, and $M_c(t) = m'$. Let the number of observations defined as $T_{c,d,m'} := \sum_{t=1}^{T} \mathbb{1}_{\{c(t)=c, d(t)=d, M_c(t)=m'\}}$ and $T_{total} = \sum_{m' \in \mathcal{V}_m} T_{c,d,m'}$. We consider the following estimator:

$$\Theta(m) := \frac{1}{T_{total}} \sum_{m' \in \mathcal{V}_m} \sum_{t=1}^{T} \mathbb{1}_{\{c(t)=c, d(t)=d, M_c(t)=m'\}} (1 - Y_{c,d,m'}(t)). \tag{8.32}$$

$$\mathbb{E}[\Theta(m)] = \frac{1}{T_{total}} \sum_{m' \in \mathcal{V}_m} T_{c,d,m'} D_{c,d}(m') \tag{8.33}$$

$$\mathbb{E}[\Theta(m)] = \frac{1}{T_{total}} \sum_{m' \in \mathcal{V}_m} T_{c,d,m'} D_{c,d}(m)^{\frac{nb_c - m'}{nb_c - m}} = g(D_{c,d}(m)). \tag{8.34}$$

with $D_{c,d}(m) = (1 - a_{c,d}/n)^{nb_c - m}$. The next lemma shows that g^{-1} is Lipschitz continuous.

Lemma 43. g^{-1} *is $2e^a$ Lipschitz.*

Proof. Let

$$g(x) = \frac{1}{T_{total}} \sum_{m' \in \mathcal{V}_m} T_{c,d,m'} x^{\frac{nb_c - m'}{nb_c - m}},$$

defined on the interval $x \in \left[(1 - \frac{a}{n})^{nb_c}, 1\right]$, where $a_{c,d} \leq a < n$. The function g is continuously differentiable and strictly increasing on this interval, as it is a finite sum of positive-coefficient power functions. Hence, g is invertible, and its inverse is differentiable with

$$\frac{dg^{-1}}{dy}(y) = \frac{1}{g'(g^{-1}(y))}.$$

We compute

$$g'(x) = \frac{1}{T_{total}} \sum_{m' \in \mathcal{V}_m} T_{c,d,m'} \cdot \frac{nb_c - m'}{nb_c - m} \cdot x^{\frac{nb_c - m'}{nb_c - m} - 1},$$

so that

$$\frac{dg^{-1}}{dy}(y) = \frac{1}{\frac{1}{T_{total}} \sum_{m' \in \mathcal{V}_m} T_{c,d,m'} \cdot \frac{nb_c - m'}{nb_c - m} \cdot (g^{-1}(y))^{\frac{nb_c - m'}{nb_c - m} - 1}}.$$

To bound this derivative, we lower bound $g'(x)$ over the domain. Let

$$\Delta := \min \left\{ 1, \left(1 - \frac{a}{n}\right)^{nb_c \cdot \max_{m' \in \mathcal{V}_m} \left(\frac{nb_c - m'}{nb_c - m} - 1\right)} \right\},$$

then for all $x \in \left[(1 - \frac{a}{n})^{nb_c}, 1\right]$,

$$g'(x) \geq \frac{\Delta}{T_{total}} \sum_{m' \in \mathcal{V}_m} T_{c,d,m'} \cdot \frac{nb_c - m'}{nb_c - m}.$$

This yields

$$\frac{\mathrm{d}g^{-1}}{\mathrm{d}y}(y) \leq \frac{1}{\frac{\Delta}{T_{total}} \sum_{m' \in \mathcal{V}_m} T_{c,d,m'} \cdot \frac{nb_c - m'}{nb_c - m}}.$$

Using the assumption $\mathcal{V}_m = \left\{ m' \in [0, nb_c) \,\middle|\, \frac{1}{2} \leq \frac{nb_c - m'}{nb_c - m} \leq 2 \right\}$, we get,

$$\frac{\mathrm{d}g^{-1}}{\mathrm{d}y}(y) \leq \frac{(1 - \frac{a}{n})^{-nb_c}}{1/2} = 2(1 - \frac{a}{n})^{-nb_c} \leq 2e^a,$$

so g^{-1} is $2e^a$-Lipschitz.

$\square$

The next result establishes a concentration result on an estimator of $\left(1 - \frac{a_{c,d}}{n}\right)^{nb_c - M_c(t)}$.

Lemma 44. *With probability* $1 - \delta$, $\hat{D}_{c,d}(m) = g^{-1}(\Theta(m))$ *satisfies,*

$$|\hat{D}_{c,d}(m) - D_{c,d}(m)| \leq 2e^a \sqrt{\frac{\log(2/\delta)}{2T_{total}}}.$$

Proof. Let $\Theta(m) = \frac{1}{T_{total}} \sum_{m' \in \mathcal{V}_m} \sum_{t=1}^{T} \mathbb{1}_{\{\{c(t)=c,\, d(t)=d,\, M_c(t)=m'\}\}}(1 - Y_{c,d,m'}(t))$, it is a sum of independent Bernoulli random variables, then using Hoeffding inequality, with probability $1 - \delta$, it satisfies,

$$|\Theta(m) - \mathbb{E}(\Theta(m))| \leq \sqrt{\frac{\log(2/\delta)}{2T_{total}}}.$$

As proved in Lemma 43, g^{-1} is $2e^a$ Lipschitz, $\hat{D}_{c,d}(m) = g^{-1}(\Theta(m))$ satisfies with probability $1 - \delta$,

$$|\hat{D}_{c,d}(m) - D_{c,d}(m)| \leq 2e^a \sqrt{\frac{\log(2/\delta)}{2T_{total}}}.$$

$\square$

Let $R(T)$ be the cumulative regret defined by,

$$R(T) = \sum_{c \in \mathcal{C}} M_c(T) - \hat{M}_c(T)$$

$$= R_{\text{explore}}(T_{\text{explore}}) + R_{\text{exploit}}(T_{\text{exploit}}).$$

We consider $T_{\text{explore}} = T^{\omega}$ with $0 < \omega < 1$ and $T_{\text{exploit}} = T - T_{\text{explore}}$, and M_c is the matching size in the class c created by `Ex-ante Balance` and $\hat{M}_c$ is the matching size in the class c created by $\mathtt{ETC-balance}$. Since $\mathtt{ETC-balance}$ may not be making optimal decision during exploration, we can bound $R_{\text{explore}}(T_{\text{explore}})$, as $R_{\text{explore}}(T_{\text{explore}}) \leq CT^{\omega}$. For exploitation phase, let m_c be the solution of differential inclusion defined in Theorem 26. $R_{\text{exploit}}(T_{\text{exploit}})$ satisfies,

$$R_{\text{exploit}}(T_{\text{exploit}}) = \sum_{c \in \mathcal{C}} \underbrace{\sum_{t=T_{\text{explore}}}^{T-1} \mathbb{1}_{\{\exists u \in \backslash c(t), c^*=c, m_u(t+1)=1\}}}_{M_c^{\text{exploit}}} - \underbrace{\sum_{t=T_{\text{explore}}}^{T-1} \mathbb{1}_{\{\exists u \in \backslash c(t), d^*=c, m_u(t+1)=1\}}}_{\hat{M}_c^{\text{exploit}}},$$

c^* is the class chosen by `Ex-ante Balance` and d^* is the class chosen by $\mathtt{ETC-balance}$.

$$|R_{\text{exploit}}(T_{\text{exploit}})|$$

$$\leq \sum_{c \in \mathcal{C}} |M_c^{\text{exploit}} - Nm_c(T/n) + Nm_c(T/n) - \hat{M}_c^{\text{exploit}}|$$

$$\leq \sum_{c \in \mathcal{C}} |M_c(T) - M_c(T_{\text{explore}}) - Nm_c(T/n) + Nm_c(T/n) - \hat{M}_c(T) + \hat{M}_c(T_{\text{explore}})|$$

$$\leq \sum_{c \in \mathcal{C}} \underbrace{|M_c(T) - Nm_c(T/n)|}_{w_1} + \underbrace{|\hat{M}_c(T_{\text{explore}}) - M_c(T_{\text{explore}})|}_{w_2} + \underbrace{|Nm_c(T/n) - \hat{M}_c(T)|}_{w_3}.$$

To bound w_2, we use the previous bound of CT^{ω}. To bound w_1, we leverage the result from Theorem 26. The next lemma, built upon Theorem 26, quantifies the discrepancy between M_c, the matching size produced by the `Ex-ante Balance` algorithm, and m_c, the solution to the differential inclusion described therein.

Lemma 45. *For $c \in \mathcal{C}$ and $0 < q < 1$,*

$$|M_c(T) - Nm_c(T/n)| \sim \mathcal{O}(T^{\frac{q+3}{4}}).$$

Proof. From Theorem 26, with probability $1 - \frac{b\alpha}{n\epsilon^2}$,

$$\left| \frac{M_c(T)}{n} - m_c(T/n) \right|$$

$$\leq \min(\alpha, e^{L\alpha}/\sqrt{2L})\sqrt{A_{\alpha,c}/n + \delta_c B_{\alpha,c} + \epsilon C_{\alpha,c}}.$$

Taking $\epsilon^2 = \frac{1}{n^{1-q}}$ with $0 < q < 1$, $\delta_c = \sum_{d=1}^{D} \frac{a_{i,j}}{ne} \nu(d)$, $K_\alpha = (c\alpha + \epsilon)e^{c\alpha}/c$ with c

defined in Lemma 31, $U_c = \sum_{d=1}^{D}(1 - e^{-a_{c,d}b_c})\nu(d) \leq 1$, $A_{\alpha,c} = U_c(U_c^2 + \frac{14U_c}{3} + 2K_\alpha) \leq (\frac{17}{3} + 2K_\alpha)$, $B_{\alpha,c} = 2U_c^2 + 4L\delta_c + 12K_\alpha \leq 2 + 4L\delta_c + 12K_\alpha$, $C_{\alpha,c} = 2U_c^2 + 4L\epsilon + 8K_\alpha \leq 2 + 4L\epsilon + 8K_\alpha$.

$$
\sqrt{A_{\alpha,c}/n + \delta_c B_{\alpha,c} + \epsilon C_{\alpha,c}}
$$
$$
\leq \sqrt{\frac{(\frac{17}{3} + 2K_\alpha)}{n} + \delta_c(2 + 4L\delta_c + 12K_\alpha) + \epsilon(2 + 4L\epsilon + 8K_\alpha)}
$$
$$
\leq \sqrt{\frac{17/3 + \alpha e^{c\alpha} + 2}{n} + \frac{e^{c\alpha}/c}{n^{\frac{3-q}{2}}} + \frac{4L}{n^2} + \frac{2 + \alpha e^{c\alpha}}{n^{\frac{1-q}{2}}} + \frac{4L + e^{\alpha c}/c}{n^{1-q}}}.
$$

Thus $\sqrt{A_{\alpha,c}/n + \delta_c B_{\alpha,c} + \epsilon C_{\alpha,c}} \sim \mathcal{O}(\frac{1}{n^{\frac{1-q}{4}}})$. Since $T = \alpha n$ with $\alpha > 1$, this leads to $|M_c(T) - m_c(T/n)| \sim \mathcal{O}(T^{\frac{q+3}{4}})$. $\qquad\qquad\square$

The next lemma is for bounding w_3,

Lemma 46.
$$
|Nm_c(T/n) - \hat{M}_c(T)| \sim \mathcal{O}(T^{\frac{q+3}{4}}).
$$

Proof. The goal here is to apply the differential inclusion approximation, to bound $|\hat{M}_c(T_{exploit}) - Nm_c(T_{exploit}/n)|$. As defined previously, $\hat{M}_c$ is the matching size in the class $c \in \mathcal{C}$, built by a policy that considers the estimator $\hat{D}_{c,d}(m)$. The process $\hat{M}_c$ satisfies then for $t \in [T_{explore} + 1, T]$,

$$
\hat{M}_c(t+1) = \hat{M}_c(t) + \mathbb{1}_{\{\exists u \in \backslash_c(t), c^* = c, m_u(t+1) = 1\}}.
$$

Here c^* is the class chosen by $\mathtt{ETC - balance}$. Let us compute the expected one step change of the process $\hat{M}_c$,

$$
\mathbb{E}[\hat{M}_c(t+1) - \hat{M}_c(t)|\hat{M}_c(t), B] = \hat{H}_{c,b_c,n}(\hat{M}_c(t))\mathbb{1}_{\{\max_{k \in [C]} \hat{H}_{k,b_k,n}(\hat{M}_k(t)) = \hat{H}_{c,b_c,n}(\hat{M}_c(t))\}}
$$
$$
= \hat{\mathcal{H}}_{c,b_c,n}(\hat{M}_c(t)).
$$

where $\hat{H}_{c,b_c,n}(\hat{M}_c(t)) = \sum_{d=1}^{D}\left(1 - \hat{D}_{c,d}(\hat{M}_c(t))\right)\nu(d)$. Let $\hat{Q}_c(t+1) = \hat{M}_c(t+1) - \hat{M}_c(t) - \mathbb{E}[\hat{M}_c(t+1) - \hat{M}_c(t)|B, \hat{M}_c(t)]$. Here Q_c is a martingale difference sequences that satisfies the same assumptions in Lemma 30. for $t \in [T_{explore} + 1, T]$

$$
\hat{M}_c(t+1) = \hat{M}_c(t) + \hat{\mathcal{H}}_{c,b_c,n}(\hat{M}_c(t)) + \hat{Q}_c(t+1) \tag{8.35}
$$
$$
= \hat{M}_c(t) + \mathcal{H}_{c,b_c,n}(\hat{M}_c(t)) + \Delta_{c,b_c,n}(\hat{M}_c(t)) + \hat{Q}_c(t+1). \tag{8.36}
$$

Note that the function $\mathcal{H}_{c,b_c,n}(\hat{M}_c(t))$ is the same as the one defined in Equation (8.25). The goal of the proof is to show that $\frac{1}{n}\sum_{l=1}^{t} \Delta_{c,b_c,n}(\hat{M}_c(l))$ converges to 0 with high probability. This is important because, according to the proofs of

Theorems 1 and 4 in [45], establishing this convergence implies that the process $\hat{M}_c$ converges to the same solution as the differential inclusion introduced in Theorem 26. This follows from the fact that we can write the evolution of $\hat{M}_c$ as in Equation (8.35) where $\hat{Q}_c$ is a martingale difference term and $\mathcal{H}_{c,b_c,n}(\hat{M}_c(t))$ is the drift of the process M_c defined in Equation (8.25). Thus, showing that the average of the perturbation term $\Delta_{c,b_c,n}(\hat{M}_c(t))$ vanishes ensures that $\hat{M}_c$ asymptotically follows the same differential inclusion introduced in Theorem 26.

We now turn our attention to analyzing $\Delta_{c,b_c,n}(\hat{M}_c(t))$,

$$\Delta_{c,b_c,n}(\hat{M}_c(t)) = \hat{H}_{c,b_c,n}(\hat{M}_c(t))\mathbb{1}_{\{\max_{k\in[C]} \hat{H}_{k,b_k,n}(\hat{M}_k(t))=\hat{H}_{c,b_c,n}(\hat{M}_c(t))\}}$$
$$- H_{c,b_c,n}(\hat{M}_c(t))\mathbb{1}_{\{\max_{k\in[C]} H_{k,b_k,n}(\hat{M}_k(t))=H_{c,b_c,n}(M_c(t))\}}$$

$$|\Delta_{c,b_c,n}(\hat{M}_c(t))| \le \left| \hat{H}_{c,b_c,n}(\hat{M}_c(t)) - H_{c,b_c,n}(\hat{M}_c(t)) \right|$$
$$+ H_{c,b_c,n}(\hat{M}_c(t)) \left| \mathbb{1}_{\{\max_{k\in[C]} \hat{H}_{k,b_k,n}(\hat{M}_k(t))=\hat{H}_{c,b_c,n}(\hat{M}_c(t))\}} - \mathbb{1}_{\{\max_{k\in[C]} H_{k,b_k,n}(\hat{M}_k(t))=H_{c,b_c,n}(M_c(t))\}} \right|$$

From the concentration inequality in Lemma 44, we have,

$$A = \left| \hat{H}_{c,b_c,n}(\hat{M}_c(t)) - H_{c,b_c,n}(\hat{M}_c(t)) \right| \le \sum_{d=1}^{D} |\hat{D}_{c,d}(\hat{M}_c(t)) - D_{c,d}(\hat{M}_c(t))|\nu(d) \quad (8.37)$$

with probability at least $1 - \delta$,

$$A \le 2e^a \sqrt{\frac{\log(2/\delta)}{2T_{total}}} \quad (8.38)$$

Now let us focus on $\left| \mathbb{1}_{\{\max_{k\in[C]} \hat{H}_{k,b_k,n}(\hat{M}_k(t))=\hat{H}_{c,b_c,n}(\hat{M}_c(t))\}} - \mathbb{1}_{\{\max_{k\in[C]} H_{k,b_k,n}(\hat{M}_k(t))=H_{c,b_c,n}(M_c(t))\}} \right|$
we need to bound the mismatch between the indicator functions. These indicators differ only when the maximum changes, i.e., when the "argmax" under $\hat{D}_{c,d}(\hat{M}_c(t))$ and $D_{c,d}(\hat{M}_c(t))$ do not agree. Suppose $H_{c,b_c,n}(\hat{M}_c(t))$ is the largest value among all $H_{k,b_k,n}(\hat{M}_k(t))$ by a margin $\gamma > 0$,

$$H_{c,b_c,n}(\hat{M}_c(t)) > H_{k,b_k,n}(\hat{M}_k(t)) + \psi \qquad \text{for } k \ne c.$$

To ensure that the estimator $\hat{D}_{c,d}$ does not flip the argmax, we want to control how much each function $H_{k,b_k,n}(\hat{M}_k(t))$ can change under small perturbations in $D_{c,d}$. Suppose that the maximizer changes for the function $\hat{H}$, this means that for some $j \ne c$,

$$\hat{H}_{j,b_j,n}(\hat{M}_j(t))(\hat{M}_j(t)) \ge \hat{H}_{c,b_c,n}(\hat{M}_c(t)).$$

Thus,

$$\hat{H}_{j,b_j,n}(\hat{M}_j(t)) - \hat{H}_{c,b_c,n}(\hat{M}_c(t)) = \hat{H}_{j,b_j,n}(\hat{M}_j(t)) - H_{j,b_j,n}(\hat{M}_j(t)) - \hat{H}_{c,b_c,n}(\hat{M}_c(t))$$
$$+ H_{c,b_c,n}(\hat{M}_c(t)) + H_{j,b_j,n}(\hat{M}_j(t)) - H_{c,b_c,n}(\hat{M}_c(t)).$$

$\hat{H}_{j,b_j,n}(\hat{M}_j(t)) \geq \hat{H}_{c,b_c,n}(\hat{M}_c(t))$ implies,

$$J = \hat{H}_{j,b_j,n}(\hat{M}_j(t)) - H_{j,b_j,n}(\hat{M}_j(t)) - \hat{H}_{c,b_c,n}(\hat{M}_c(t)) + H_{c,b_c,n}(\hat{M}_c(t))$$
$$J \geq H_{c,b_c,n}(\hat{M}_c(t)) - H_{j,b_j,n}(\hat{M}_j(t)) \geq \psi.$$

Applying the triangle inequality, we get,

$$B = |\hat{H}_{j,b_j,n}(\hat{M}_j(t)) - H_{j,b_j,n}(\hat{M}_j(t)) - \hat{H}_{c,b_c,n}(\hat{M}_c(t)) + H_{c,b_c,n}(\hat{M}_c(t))|$$
$$B \leq |\hat{H}_{j,b_j,n}(\hat{M}_j(t)) - H_{j,b_j,n}(\hat{M}_j(t))| + |\hat{H}_{c,b_c,n}(\hat{M}_c(t)) - H_{c,b_c,n}(\hat{M}_c(t))|.$$

Thus by the margin condition, we have

$$|\hat{H}_{j,b_j,n}(\hat{M}_j(t)) - H_{j,b_j,n}(\hat{M}_j(t))| + |\hat{H}_{c,b_c,n}(\hat{M}_c(t)) - H_{c,b_c,n}(\hat{M}_c(t))| \geq \psi$$
$$\Rightarrow \max\{|\hat{H}_{j,b_j,n}(\hat{M}_j(t)) - H_{j,b_j,n}(\hat{M}_j(t))| + |\hat{H}_{c,b_c,n}(\hat{M}_c(t)) - H_{c,b_c,n}(\hat{M}_c(t))|\} \geq \psi/2.$$

Let $\gamma_t = \left| \mathbb{1}_{\{\max_{k\in[C]} \hat{H}_{k,b_k,n}(\hat{M}_k(t))=\hat{H}_{c,b_c,n}(\hat{M}_c(t))\}} - \mathbb{1}_{\{\max_{k\in[C]} H_{k,b_k,n}(\hat{M}_k(t))=H_{c,b_c,n}(M_c(t))\}} \right|$, according to previous development, we deduce that,

$$\gamma_t \leq \mathbb{1}_{\{\exists k\in\mathcal{C}, |\hat{H}_{k,b_k,n}(\hat{M}_k(t))-H_{k,b_k,n}(\hat{M}_k(t))|\geq\psi/2\}}.$$

But from concentration, we have the following condition on γ,

$$\psi \geq 4e^a \sqrt{\frac{\log(2/\delta)}{2T_{total}}}.$$

Thus choosing δ such that this condition is satisfied, we ensure that with high probability $\frac{1}{n}\sum_{l=1}^{t}\Delta_{c,b_c,n}(\hat{M}_c(l))$ tends to 0. Having established this convergence, and noting that the process $\hat{M}_c$ can be represented in the form given in Equation (8.25). Thus according to the proof of Theorem 1 and 4 in [45], $\hat{M}_c$ converges to the solution of the same differential inclusion defined in Theorem 26. and we have $|\hat{M}_c(T) - Nm_c(T/n)| \sim \mathcal{O}(n^{\frac{q+3}{4}})$ for $0 < q < 1$.

$\square$

With all the previous result, $R(T) = R_{explore}(T_{explore}) + R_{exploit}(T_{exploit})$, we showed first that $R_{explore}(T_{explore}) \sim \mathcal{O}(T^\omega)$, and $R_{exploit}(T_{exploit}) \leq C(2T^{\frac{q+3}{4}} + T^\omega)$, thus choosing $\omega = \frac{q+3}{4}$, we get that $R(T) \sim \mathcal{O}(T^{\frac{q+3}{4}})$.

9

Optimizing the coalition gain in Online Auctions with Greedy Structured Bandits

Motivated by online display advertising, this work considers repeated second-price auctions, where agents sample their value from an unknown distribution with cumulative distribution function F. In each auction t, a decision-maker bound by limited observations selects n_t agents from a coalition of N to compete for a prize with p other agents, aiming to maximize the cumulative reward of the coalition across all auctions. The problem is framed as an N-armed structured bandit, each number of player sent being an arm n, with expected reward $r(n)$ fully characterized by F and $p + n$. We present two algorithms, Local-Greedy (`LG`) and Greedy-Grid (`GG`), both achieving *constant* problem-dependent regret. This relies on three key ingredients: **1.** an estimator of $r(n)$ from feedback collected from any arm k, **2.** concentration bounds of these estimates for k within an estimation neighborhood of n and **3.** the unimodality property of r under standard assumptions on F. Additionally, `GG` exhibits problem-independent guarantees on top of best problem-dependent guarantees. However, by avoiding to rely on confidence intervals, `LG` practically outperforms `GG`, as well as standard unimodal bandit algorithms such as `OSUB` or multi-armed bandit algorithms.

Contents

9

9.1 Introduction

The online display advertising has seen remarkable evolution in recent decades [49, 89, 99, 82]. Publishers, who are the suppliers of digital ad space on the internet, sell display spots for ads to advertisers through real-time bidding in spot auctions, with many of these auctions being conducted using first or second-price mechanisms [69]. Due to the technological complexity of online advertising, advertisers usually delegate the task of buying ad placements to demand-side platforms (DSP) that operate many advertising campaigns. This interaction between DSP and the publisher, can be simplified as the publisher acting as multiple ad auctions selling ad impressions (online displays), while the DSP acts as a *centralized coalition*: at each time step, it determines which campaign(s) from the coalition participate to the auction to maximize their total gain. The chosen campaign(s) then compete with others to secure impressions. The primary goal of advertising companies is then to maximize the cumulative utility: the total value of impressions won minus their costs. This raises a fundamental question: *how many ad campaigns should participate in the auction to optimize the overall utility?* In the *interim* setting, where the DSP observes current bidder values before deciding, it's known that only the highest value bidder should be sent. However, online privacy enhancements in browsers necessitate *ex-ante* decisions from DSPs [28], without exact value knowledge. Here, the problem becomes challenging: choosing a small number of campaigns can make it difficult to secure impressions, while securing the spot with a large number of bidders inevitably raises the price due to competition. In this paper, this problem is formalized and solved via novel Multi-Armed-Bandit (MAB) algorithms.

9.1.1 Problem statement

Consider a sequence of T ad impressions sold through *second price auctions* (see [69] for a survey). At auction $t \in [T]$, each participant (bidder) bids on the item based on its own (stochastic) value for the item. The highest bidder wins the item and pays a price equal to the second highest bid. The *decision maker* (the DSP) runs $N \in \mathbb{N}^*$ advertising campaigns forming a *coalition*. At time t, two groups of bidders participate: (1) $n_t \in [N]$ bidders from the coalition chosen by the decision maker *ex-ante* – without knowing the realization of the bidders' values – and (2) $p \in \mathbb{N}^*$ other bidders, that we call the *competition*. When a bidder from the coalition wins the auction, the decision maker observes the realized value for the winner (also called *winning bid*). In the rest of the paper, the following assumptions about the behavior of bidders is made.

Assumption 1. *All bidders are identical, their values are sampled i.i.d. from a distribution supported on $[0, 1]$ characterized by its cumulative distribution function (c.d.f.) F. All bidders bid their value.*

Assuming identical bidders with i.i.d values is a strong but widespread assumption in auction theory [69, 73], known as the symmetric bidders case. It is partic-

ularly relevant in online advertising, notably in homogeneous impression markets where advertisers compete for similar ad displays due to shared objectives, target demographics, or placement competition. The bounded support assumption is also standard, as letting an automated system bid arbitrarily large values is unrealistic. Finally, bidders bid their value as this is a weakly dominant strategy in this case. Lastly, assuming a known number of competitors p is frequently seen in auction models (see for instance [69] chapter 3.2.2). Under Assumption 1, the expected reward received by the decision maker at time t is given by $r(n_t)$, where r is the *expected reward function*, defined by

$$ r : n \in [N] \mapsto r(n) := \mathbb{E}_{\mathbf{v}=(v_i)_{i \in [n+p]} \sim F \times \cdots \times F} \left[(\mathbf{v}_{(1)} - \mathbf{v}_{(2)}) \mathbb{1} \left\{ \arg\max_{i \in [n+p]} v_i \in [n] \right\} \right]. \quad (9.1) $$

where $\mathbf{v}_{(1)}$ and $\mathbf{v}_{(2)}$ are respectively the first and second maximum of $\mathbf{v}$, and $[n]$ is used to abbreviate $\{1, \ldots, n\}$. The problem therefore reduces to a MAB where the decision maker chooses *arms* $n_1, \ldots, n_T \in [N]$ sequentially and aims to minimize its cumulative *expected regret* $\mathcal{R}(T)$ defined by

$$ \mathcal{R}(T) = \sum_{t \leq T} r(n^*) - r(n_t), \quad \text{with} \quad n^* = \arg\max_{n \in [N]} r(n), \quad (9.2) $$

given that privacy constraints from the browser [28] only let the decision maker observes (1) if the coalition won, (2) the realization of the maximum value when winning.

9.1.2 Related works

Following (9.2), the problem presented in this paper can be formulated as a Multi-Arm Bandits (MAB, see [72] for a survey). In MAB, a learner repeatedly selects from a set of actions, or "arms", each yielding a reward. The goal is to maximize total rewards by striking a balance between exploration (sampling various arms to learn their rewards) and exploitation (picking the arms with the highest anticipated rewards based on collected feedback). While the literature has known a significant development in the last years ([4, 27, 54], to name a few), the most popular approaches arguably remain *exponential weights algorithms* (EXP3, [12]) in adversarial settings, and *optimism in face of uncertainty* (UCB, [11]) when rewards are stochastic.

While UCB and EXP3 can both tackle the regret minimization problem presented here, they inevitably achieve sub-optimal performance due to not using the inherent *structure* of the expected reward function. Several types of structure have been explored in the bandit literature, some notable examples being linear bandits [2], Lipschitz bandits [75], or unimodal bandits [31, 86, 88]. The problem considered here is novel in the literature of structured bandits, arising from the observability restrictions coming with privacy-enhancing systems. Still, in the next section we show that unimodality – in this paper the fact that r admits only one local (hence global) maximum – is in many cases inherited from this stronger structure. A

Table 9.1: Comparison of regret guarantees for different algorithms

Algorithm	Regret upper bound
EXP3 [12]	$\mathcal{O}(\sqrt{NT})$
UCB1 [11]	$\mathcal{O}\left(\sum_{n\in[N]} \frac{\log(T)}{\Delta_n} \wedge \sqrt{NT}\right)$
OSUB [31]	$\mathcal{O}\left(\frac{\log(T)}{\Delta_{n^\star+1}} + \frac{\log(T)}{\Delta_{n^\star-1}} + \sum_{n\in[N]} \frac{\Delta_n \log\log(T)}{\Delta^2}\right)$
LG (this paper)	$\widetilde{\mathcal{O}}_N(\sum_{n\in[N]} \frac{\Delta_n}{\Delta^2})$
GG (this paper)	$\widetilde{\mathcal{O}}_N(\sum_{n\in\mathcal{B}^\star} \frac{1}{\Delta_n} + \sum_{n\in\mathcal{S}} \frac{\Delta_n}{\Delta^2}) \wedge \widetilde{\mathcal{O}}(\sqrt{(\log(N) + |\mathcal{B}^\star|)T})$

typical strategy to exploit unimodality - also used in this work - consists in playing a standard bandit policy (such as UCB) on a well chosen subset of arms (OSUB, [31]).

Last, the use of online learning algorithms to tackle repeated auction problems have been explored in various contexts ([83, 34, 13, 96, 84, 33]). However, none of these works approach the problem through the perspective of a coalition of bidders, and are thus not applicable to this setting.

9.1.3 Outline and contributions

Section 9.3 presents two novel bandit algorithms: LG (Local Greedy) which is inspired by OSUB[31], and GG (Greedy Grid) which combines Local Greedy and a successive elimination strategy. Theorem 32 and Theorem 33 provide upper bounds on the regret of LG and GG respectively, which are summarized in Table 9.1. Both algorithms achieve problem-dependent regret independent of T. However, their scaling differs: the regret of LG depends on the *worst local gap* $\Delta = \min_{n\in[N]} |r(n+1)-r(n)|$, while for GG it only depends on the gaps $\Delta_n = r(n^*)-r(n)$. Furthermore, w.h.p. GG only suffers regret for arms in a *reference grid* $\mathcal{S}$ containing $\mathcal{O}(\log(N))$ arms and in a *neighborhood* $\mathcal{B}^\star$ of the optimal arm. All these quantities, as well as the notation $\widetilde{\mathcal{O}}$ and $\widetilde{\mathcal{O}}_N$ (hiding logarithmic factors), are defined in Section 9.3. These regret upper bounds rely on three key ingredients presented in Section 9.2: (1) an estimator of $r(n)$ from feedback collected from any arm k (2) novel concentration bounds on these estimates for k within an estimation neighborhood of n (Theorem 31) and (3) the unimodality property of r under standard assumptions on F. Lastly, Section 9.D provides an experimental benchmark comparison of the performance of GG, LG and their competitors: LG has the lowest expected regret among the algorithms tested. Indeed, LG avoids the explicit use of the confidence bounds in the algorithm which makes it more practical, even though GG admits better theoretical guarantees.

9.2   Estimating the reward function from samples of powers of F

In this part, we put aside the sequential nature of the repeated auction setting that we introduced and consider the problem of estimating the expected reward as a function of the number of bidders, given a stream of collected data. We first present a formulation of the expected reward function in terms of powers of the c.d.f. F. Then, we leverage this formula to introduce *power estimates*, as a solution to estimate the expected reward of an arm $n \in [N]$ from samples collected from an arm $k \in [N]$. Lastly, we discuss the theoretical properties of these estimates, introducing upper and lower confidence bounds on the expected reward in Theorem 31.

9.2.1   Properties of the expected reward

The expected reward function r (Eq. (9.1)) can be expressed as a function of n, p and the c.d.f. F.

Lemma 47. *The expected reward function defined in Equation* (9.1) *satisfies,*

$$n \in [N] \mapsto r(n) = n \int_0^1 F^{p+n-1}(x) - F^{p+n}(x)\mathrm{d}x. \tag{9.3}$$

The proof can be found in Section 9.A.1 and is based on properties of order statistics.

This particular definition of $r(n)$, which is a product of n and a function that decreases with n, suggests that r could be unimodal for some choices of F. In the rest of the paper, we restrict ourselves to distributions that guarantees unimodal reward functions.

Assumption 2. *F and p are such that the reward function r in Equation* (9.3) *is unimodal*

As the next lemma shows, many classical distributions lead to unimodal rewards for all $p \in \mathbb{N}$.

Lemma 48. *Let F be the cumulative distribution function of a Bernoulli, truncated exponential or Complementary Beta distribution. Then, for any $p \in \mathbb{N}^*$, r in Equation* (9.3) *unimodal.*

The proof of Lemma 48 can be found in Section 9.A.2. Note that the Complementary Beta distributions [60], chosen for technical reasons, are similar to Beta distributions and any Beta distribution can be approached by a Complementary Beta. Furthermore, in Section 9.A.3 we present experiments suggesting that r is

unimodal for all $p \in \mathbb{N}^*$ if F is the c.d.f of Beta or Kumaraswamy distributions. However, we also show in Section 9.A.4 that this is not always the case, by providing a counter example. Nonetheless, we argue that (complementary) beta or truncated exponentials are flexible models for real world data, so Assumption 2 is reasonable in practice. We furthermore discuss in Section 9.3.2 the adaptation of our algorithms if this was not the case.

9.2.2 Estimation of powers of F

Consider the feedback $\overline{W}_k = (w_{k,1}, \ldots, w_{k,m_k})$ gathered after playing arm k and winning the auction m_k times. $\overline{W}_k$, represents the sequence of first values (value of the winning bid) which has been *collected by arm k*.

It is well known that the marginal distribution of any order statistic can be expressed as a function of the c.d.f. F (see Section 2.1 of [32]). The distribution of any element of $\overline{W}_k$ has cumulative distribution function $F_k : x \in [0,1] \to F^{k+p}(x)$, which clearly exhibits a one-to-one mapping between $F_k(x)$ and $F(x)$. Hence, given $\overline{W}_k$, for any $\ell \in \mathbb{N}$ we can estimate F^ℓ by

$$\widetilde{F}^\ell_{k+p} : x \mapsto (\widehat{F}_{k+p}(x))^{\frac{\ell}{k+p}}, \text{ where } \widehat{F}_{k+p} : x \mapsto \frac{1}{m_k} \sum_{j=1}^{m_k} \mathbb{1}\{w_{k,j} \leq x\} \text{ (emp. c.d.f. of } \overline{W}_k).$$

$$(9.4)$$

Estimation of r Consider any arm $n \in [N]$. Following Equation (9.3), it appears that estimating both F^{n+p} and F^{n+p-1} is sufficient to construct an estimate of $r(n)$. According to Equation (9.4), this can be done from samples originated from any arm $k \in [N]$, by using the *simple estimate*

$$\widehat{r}_k(n) = n \int_0^1 \left(\widetilde{F}^{n+p-1}_{k+p}(x) - \widetilde{F}^{n+p}_{k+p}(x) \right) \mathrm{d}x \ . \tag{9.5}$$

Furthermore, it also clear that any convex combination of estimates can become a new estimate, however in the rest of the paper we focus on simple estimates for simplicity.

Remark 5 (Adaptation to different feedback). *A similar procedure can be derived for a setting where the sequence of second prices would be observed instead. Indeed, their distribution would be $G_k : x \in [0,1] \mapsto (k+p)F(x)^{k+p-1} - (k+p-1)F(x)^{k+p}$, which can lead to a reward estimate similar to (9.5) by using a suitable inversion formula. The same can be said for the case where both first and second prices are observed, with additional complexity because the joint distribution should be considered since for each auction the first and second price are dependent variables.*

9.2.3   Concentration of estimates of the reward function

We now introduce the first theoretical contribution of this paper: confidence bounds on the deviations of an empirical estimate $\widehat{r}_k(n)$ w.r.t. the true expected reward $r(n)$.

Importance of (relatively) local estimation   In principle, (9.5) suggests that samples from any arm $k \in [N]$ can provide a simple estimate of the reward function of any other arm $n \in [N]$. However, we establish that the position of k w.r.t. n significantly impacts the concentration of $\widehat{r}_k(n)$. Intuitively, the ratio $(n + p)/(k + p)$ determines how the uncertainty on F^{k+p} propagates on the reward after performing the inversion to obtain an estimate of F^{n+p}. Indeed, considering any $i \in \mathbb{N}$, if for some $x \in [0, 1]$ the deviation $F(x)^i - \widehat{F}_i(x)$ is small then a first order approximation provides that

$$\forall j \in \mathbb{N} \; : \; (F(x)^i)^{\frac{j}{i}} - \widehat{F}_i(x)^{\frac{j}{i}} \approx (F(x)^i - \widehat{F}_i(x)) \times \frac{j}{i} F_i(x)^{\frac{j}{i}-1} \;. \tag{9.6}$$

Hence, a small error on $F(x)^i$ is multiplied by $\frac{j}{i} F_i(x)^{\frac{j}{i}-1}$ to obtain the resulting error on $F(x)^j$. For $j \geq i$ this term can be as large as j/i while for $j < i$ it can be arbitrarily large if $F^i(x)$ is very small. This observation motivates a restriction on the range of arms that can be used to estimate the reward of a given arm n, that we call its *estimation neighborhood*. We use the convention that arms smaller than 1 or greater than N exist but have not collected any sample and have a known reward of 0.

Definition 11 (Estimation neighborhood of an arm n).
Assume[1] that $p \geq 4$ Then, the estimation neighborhood of n is the range $\mathcal{V}(n) = [v_\ell(n), v_r(n)] = \big\{ k \in [N] : \; k + p \in \big[\frac{n+p}{2}, \frac{3}{2}(n + p - 1)\big] \big\}$. We call $v_\ell(n)$ and $v_r(n)$ respectively the furthest left and right neighbor of n.

Theorem 31 (Concentration of simple estimates). *Consider any $n \in [N]$ and $k \in \mathcal{V}(n)$. Let $\widehat{r}_k(n)$ be defined according to (9.5) from m_k samples collected by k. Then, there exists some constants $\beta_{k,n}$ (depending on n, k, p) and $\xi_{k,n,F}$ (additionally depending on F) such that, with probability $1 - \delta$,*

$$|\widehat{r}_k(n) - r(n)| \leq \beta_{k,n} \sqrt{\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}} + n \times \xi_{k,n,F} \left(\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}\right)^{\frac{n+p-1}{k+p}} \;. \tag{9.7}$$

Furthermore, the constants admit universal upper bounds for any n, k, p, F. For instance if $m_k \geq 4$ it holds that $\beta_{k,n} \leq 33$ and $\gamma_{k,n,F} \leq 100$.

Proof sketch (see Section 9.B for the detailed proof). The first ingredient consists in approximating the reward formulation (9.3) by a Riemann sum: for some step size

---

[1]This assumption simplifies the presentation but our theoretical results can easily be adapted for $p < 4$ if $n - 1$ and $n + 1$ are included by default in $\mathcal{V}(n)$.

$D^{-1} > 0$, it holds that $\widehat{r}_k(n) - r(n) = \frac{n}{D} \sum_{s=0}^{D-1} \mathcal{E}(x_s) + \mathrm{err}_D$, with $x_s = s/D$ for all $s \in \{0, \ldots, D-1\}$. In Lemma 50 we use elementary properties of F to show that the approximation error satisfies $\mathrm{err}_D \in [0, nD^{-1}]$. Next, we upper and lower bound $\mathcal{E}(x_s)$ with different concentration bounds according to the value of $F_{k,s} := F(x_s)^{k+p}$. More precisely, for any $\delta \in (0,1)$ the following bounds hold each with probability at least $1 - \delta$,

$$
\begin{cases}
|\widehat{F}_k(x_s) - F_{k,s}| \leq \sqrt{F_{k,s}} \times \sqrt{\frac{3\log(\frac{2}{\delta})}{m_k}} & \text{if } F_{k,s} \in I_0 := \left[\frac{3\log(2/\delta)}{m_k}, 1\right] & \text{(Chernoff)}, \\
\widehat{F}_k(x_s) \leq \frac{6\log(2/\delta)}{m_k} & \text{if } F_{k,s} \in I_1 := \left(\frac{\delta}{m_k}, \frac{3\log(2/\delta)}{m_k}\right) & \text{(Chernoff)}, \\
\widehat{F}_k(x_s) = 0 & \text{if } F_{k,s} \in I_2 := \left[0, \frac{\delta}{m_k}\right] & \text{(union bound)}.
\end{cases}
$$

These results are derived in Lemma 51 from a well-known multiplicative form of the Chernoff bound for Bernoulli random variables [51]. Then, the analysis consists in using the appropriate bound for each point $s \in \{0, \ldots, D-1\}$. The interval I_0 provides the first term in (9.7), which is dominant in terms of m_k, and we make $\beta_{k,n}$ fully independent of F by carefully using some properties of the reward function. The two remaining intervals I_1 and I_2 provide the second term in (9.7), and $\gamma_{k,n,F}$ depend on F through the boundaries of the interval I_1. The corresponding factor in $\xi_{k,n,F}$ can be bounded by 1 or estimated in practice (see Section 9.B.4). □

In Section 9.B, we give the expression of $\beta_{k,n}$ and $\xi_{k,n,F}$ and provide in (9.17) and (9.19) fully explicit upper and lower confidence bounds on $\widehat{r}_k(n)$, depending on all problem parameters, and that are much tighter than what the universal constants provided in the theorem suggest. These universal constants are purely indicative, in order to assess that $\beta_{k,n}$ and $\xi_{k,n,F}$ do not diverge for any value of the problem parameters. We now provide more high-level comments on the derivation of this result.

Discussion The proof of Theorem 31 is non-trivial, and the careful usage of the Chernoff bounds that we introduced is crucial to obtain tight bounds on $\widehat{r}_k(n)$ for two reasons. First, it seems necessary to concentrate estimates from arms $k > n$ (see the discussion below (9.6)), which are instrumental to the performance of the bandit algorithms presented in the next section. Secondly, by exhibiting powers of F, they make $\beta_{k,n}$ **not** increasing linearly in n, which is not easy to achieve. Indeed, it is clear from the analysis that this cost would be inevitable with standard Hoeffding bounds. However, completely avoiding n seems difficult in general, so our proof provides a way to mitigate its cost by multiplying it by a higher power of m_k^{-1}, at least $m_k^{-\frac{2}{3}}$ (if $k + p = \frac{3}{2}(n + p - 1)$). This is the theoretical motivation for the definition of $\mathcal{V}(n)$ (Definition 11): while $k + p = 2(n + p - 1)$ would lead to theoretically valid results, it would not ensure that the linear term in n is second-order in m_k.

We now conclude this section by exhibiting a condition on F that allows to reduce the scaling of the confidence bound in n to logarithmic terms.

Lemma 49 (Improved bound for Lipschitz quantile function). *Assume that $k \in \mathcal{V}(n)$ and F^{-1} is L-Lipschitz, then there exists an absolute constant ξ such that with probability $1 - \delta$ it holds that*

$$|\widehat{r}_k(n) - r(n)| \leq \beta_{k,n} \sqrt{\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}} + \xi L \log\left(\frac{4\lceil n\sqrt{m_k}\rceil}{\delta}\right) \left(\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}\right)^{\frac{n+p-1}{k+p}}.$$

(9.8)

This result is proved in Section 9.B.3, and shows that for some distributions (e.g. "close" to uniform) the confidence bounds converge relatively fast to standard sub-Gaussian type of bounds, even for very large n. Whether this result holds in general remains open.

9.3   Bandit algorithms

9.3.1   Bandit algorithms: Local-Greedy (`LG`) and Greedy-Grid (`GG`)

We now detail the two novel bandit algorithms proposed to tackle the problem presented in Section 9.1. Both rely on the use of simple estimates of $r(n)$ (see Section 9.2) by arms present in its *estimation neighborhood* $\mathcal{V}(n)$ (see Definition 11) and theoretically motivated by Theorem 31. In this section, for ease of exposition, we describe algorithms as if feedback was collected at every time steps. In Section 9.C.1, we show that the algorithms and their guarantees only require a slight adaptation when the feedback is collected only when the auction is won.

9.3.1.1   Local-Greedy

We first present Local-Greedy (`LG`), which is a natural adaptation of a standard policy in unimodal bandits, `OSUB` [31].The main idea of `OSUB` is to play `UCB` locally around a reference arm, and eventually reach the optimal arm $n^\star$ by gradually moving the reference arm in its direction. With `LG`, we adapt this principle to efficiently exploit the structure of the problem considered: at each round t, `LG` defines a reference arm ℓ_t, called *leader*, but plays *greedily* in the *neighborhood* $\mathcal{V}(\ell_t)$, based on simple power estimates computed with samples from ℓ_t only. In addition a *sampling requirement*, implemented by a parameter $\alpha \in (0,1)$, is used in order to ensure the good concentration of these estimates. We detail Local-Greedy in Algorithm 18 below.

Algorithm 18: Local Greedy (`LG`)

Input: exploration parameter α, neighborhoods $(\mathcal{V}(n))_{n \in [N]}$ (Definition 11)

1 Play $n_1 = 1$ and observe $w \sim F^{1+p}$;   ▷ `Initialization`

2 for $t \geq 2$ **do**

3   Set $\ell_t = n_{t-1}$, compute $(\widehat{r}_{\ell_t}(n))_{n \in \mathcal{V}(\ell_t)}$ (Eq.(9.5)) ; ▷ `Compute estimates`
         `from the leader`

4   **If** $m_t := |\{s \in [t-1], n_s = \ell_t\}| \leq \alpha t$: play $n_t = \ell_t$; ▷ `Linear sampling`
         `requirement`

5   **Else:** play $n_t \in \arg\max_{n \in \mathcal{V}(\ell_t)} \widehat{r}_{\ell_t}(n)$;   ▷ `Greedy play in` $\mathcal{V}(\ell_t)$

6   Observe $w \sim F^{n_t + p}$;   ▷ `Record feedback`

High-level properties of `LG`   First, using Greedy instead of `UCB` is only possible because of the structure of the problem: when ℓ_t is well-explored the estimates of arms in $\mathcal{V}(\ell_t)$ computed with samples from ℓ_t are sufficiently close to the true reward, so that *no exploration is needed*. The sampling requirement then guarantees that all greedy plays are made when ℓ_t is well explored.

A second property is that since $|\mathcal{V}(\ell_t)|$ grows with ℓ_t, a sequence of *locally optimal moves* (best play in a given neighborhood) allows to reach the optimal arm exponentially fast (in $\mathcal{O}(\log(N))$ steps), which is particularly interesting in practice if N is large. On the other hand, `LG` might suffer from the inherent drawback of any "local" policy: identifying a high-rewarding arm in a neighborhood can take a long time if the reward curve in this area is flat (depending on how small are the "local" gaps). This problem can be attenuated, but not solved, by adding an initial exploration phase. We propose Greedy-Grid, presented in the next section, as a way to fully address this issue.

Lastly, requiring only the computation of empirical reward estimates is a strength of Local-Greedy. Indeed, deriving tighter confidence bounds would improve its analysis, but not the practical implementation (and performance) of the algorithm.

9.3.1.2   Greedy-grid

The concept of Greedy-Grid is very intuitive: it plays a Local-Greedy strategy only if it can tell which segment of the reward function contains the best arm with high probability. To implement this idea, `GG` uses a Successive-Elimination procedure [39] on a *subset* of arms forming a *reference grid*, denoted by $\mathcal{S}$.

Reference grid   The grid $\mathcal{S}$ is designed so that two of its successive arms belong to their respective neighborhood (Definition 11), and can hence mutually estimate themselves and all arms in between (Theorem 31). In particular, the optimal arm can be well-estimated at least by its two closest neighbors on the grid, so its neighborhood can be "discovered" with high probability simply by sampling the points in the grid in a round-robin fashion for a sufficiently long time.

Following Definition 11, we construct $\mathcal{S} := \{s_i\}_{i\geq 1}$ recursively: $s_1 = 1$, and for $i \geq 2$ we set $s_{i+1} = \max\{s \geq s_i : s \in [N], s \in \mathcal{V}(s_i), s_i \in \mathcal{V}(s)\}$. We provide an illustrative example below.

Example 1.
For $N = 2000$ and $p = 100$ the grid is $\mathcal{S} = \{1, 50, 123, 233, 398, 645, 1016, 1572\}$.

Any arm $n \in [N]$ admits a left and right "neighbor in the grid", denoted respectively by $v_l^{\mathcal{S}}(n)$ and $v_r^{\mathcal{S}}(n)$ and defined by: $v_l^{\mathcal{S}}(n) = 0$ if $n < \min\mathcal{S}$, $v_r^{\mathcal{S}}(n) = N + 1$ if $n > \max\mathcal{S}$ and $(v_l^{\mathcal{S}}(n), v_r^{\mathcal{S}}(n)) = \arg\min_{(x,y)\in\mathcal{S}\setminus\{n\}:\, n\in[x,y]}(y - x)$ otherwise . We call the "bin" of arm n all arms between its left and right neighbors: $\mathcal{B}(n) = \{n \in [N], v_l^{\mathcal{S}}(n') < n < v_r^{\mathcal{S}}(n')\}$. For simplicity we use the notation $\mathcal{B}^\star = \mathcal{B}(n^\star)^2$.

Greedy-Grid   We provide the detailed implementation in Algorithm 19 below, and now describe the general principle of the algorithm. At each round, it operates in two steps. In the first step, it decides whether to play arms on the grid $\mathcal{S}$ (play the grid, to simplify), or to focus on a specific bin (and, as we will see, *play greedy*). This choice depends on an elimination procedure: an arm k in $\mathcal{S}$ should be *eliminated* for this round if their *upper confidence bound* (UCB) is smaller than the best *lower confidence bound* (LCB) among all other arms. Furthermore, if there exists an eliminated arm whose index is closer to the index i_t^* of the arm with the best LCB, then the unimodality assumption implies that k should also be eliminated. The set of arms not eliminated at t is called $\mathcal{C}_t$ in Algorithm 19.

To compute the UCB (U_n) and LCB (L_n) of an arm n, we elect a leader ℓ_n which is the arm in $[v_l^{\mathcal{S}}(n), v_r^{\mathcal{S}}(n)]$ that was played the most in the last t rounds and then compute the bounds based on ℓ_n, using Theorem 31. We show in the proof of Theorem 33 that this procedure ensures that a linear number of samples in t is used to compute the UCB and LCB of arms in $\mathcal{B}^\star$ with high probability.

If at least one arm is not eliminated ($\mathcal{C}_t$ is not empty), arms in $\mathcal{C}_t$ are played one after the other (Round Robin). If all arms in the grid are eliminated, GG plays greedily in the bin $B(i_t^*)$ of the arm with the highest LCB. The empirical reward of each arm $n \in B(i_t^*)$ is computed similarly as U_n and L_n using samples from the leader ℓ_n. GG then plays the best empirical arm αt times which is the same sampling requirement as LG.

The careful design of Greedy-Grid prevents the main theoretical drawback of Local-Greedy: since the algorithm has a very low probability to play in a sub-optimal bin, it almost never pays "local gaps" in a sub-optimal part of the reward function. However this guarantee comes at a cost: if $n^\star$ is not in the grid, it will never be played until the confidence intervals shrink "sufficiently" to eliminate the entire grid. Hence, GG might be more conservative than LG in practice, while offering better theoretical guarantees. We express this trade-off in the next section.

---

[2]We assume a unique optimal arm for simplicity, but the analysis holds if several successive arms are optimal.

Algorithm 19: Greedy Grid

Input: Grid $\mathcal{S}$, confidence levels $(\delta_t)_{t\in\mathbb{N}}$, sampling parameter α

1 Play $n_1 = \min \mathcal{S}$ and observe $w \sim F^{n_1+p}$

2 **for** $t \geq 2$ **do**

3      $\forall n \in [N]\ \ell_n = \arg\max_{k\in[v_l^S(n),v_r^S(n)]} \overbrace{|\{u \in [t-1], n_u = k\}|}^{m_k}$;   ▷ Elect leaders

4      $\forall n \in [N]$, $L_n = \widehat{L}_{\ell_n}(n,\delta_t)$ and $U_n = \widehat{U}_{\ell_n}(n,\delta_t)$;   ▷ Compute UCB (9.21), LCB (9.22)

5      $i_t^* = \arg\max_{n\in[N]} L_n$;   ▷ Compute best lower bound index

6      $\mathcal{C}_t = \{a \in \mathcal{S} : \forall s \in [N]$ s. t. $a \leq s \leq i_t^*$ or $a \geq s \geq i_t^*, U_s \geq L_{i_t^*}\}$;   ▷ Non-elim. grid arms

7   **if** $n_{t-1} \in B(i_t^*)$ *and* $m_{n_{t-1}} \leq \alpha t$ **then**   ▷ Ensure linear sampling for bin plays

8   Play $n_t = n_{t-1}$

9   **else**   ▷ Play grid if non-empty or greedy in the best LCB's bin

10   **If** $\mathcal{C}_t = \varnothing$: Play Round Robin on $\mathcal{C}_t$; **Else** play $\arg\max_{n\in B(i_t^*)} \hat{r}_{\ell_n}(n)$

11   Observe $w \sim F^{n_t+p}$;   ▷ Record feedback

9.3.2 Regret upper bounds

We now present the theoretical results obtained for the two algorithms presented in Section 9.3. We first establish the regret bounds and sketch their proofs, before discussing and comparing the results. We introduce some notation, that considerably simplifies the presentation of the results.

Notation: $\widetilde{\mathcal{O}}$ **and** $\widetilde{\mathcal{O}}_n$ For any $x > 0$, we use the notation $\widetilde{\mathcal{O}}(x)$ to describe a quantity that scales in x, up to logarithmic terms **in** x **and** N (hence the notation is linked to the problem). Furthermore, for $n \in [N]$ we also use $\widetilde{\mathcal{O}}_n$ as a shorthand notation for $\widetilde{\mathcal{O}}(\{n^6 \vee x\} \wedge n^2 x)$. This type of constants emerges from using (9.7) (Theorem 31) in the analysis. Indeed, we proved that the simple estimate of an arm n by an arm $k \in \mathcal{V}(n)$ admit sub-Gaussian ("square-root") confidence intervals, independent of n, when the sample size of k is larger than $\Omega(n^6)$.

Theorem 32 (Regret bound for Local-Greedy)**.** *Let* $\Delta := \min_{n\in[N-1]} |r(n+1)-r(n)|$ *(worst local gap). Under Assumption 2 and with* $\alpha = (\log_{3/2} N+1)^{-1}$*, the regret of* **LG** *is upper bounded by a* **problem-dependent constant***: there exists* $(C_n)_{n\in[N]\setminus\{n^\star\}}$*, each satisfying* $C_n = \widetilde{\mathcal{O}}_N\left(\frac{\Delta_n}{\Delta^2}\right)$*, such that* $\mathcal{R}_T \leq \sum_{n\in[N]\setminus n^\star} C_n$*.*

Additionally, if the arm set forms a single estimation neighborhood, that is $\forall n \in [N] : \mathcal{V}(n) \supset [N]$*, then each constant* C_n *can be refined to* $\widetilde{\mathcal{O}}_n(\Delta_n^{-1})$*, providing* $\mathcal{R}_T = \widetilde{\mathcal{O}}(\sqrt{NT})$*, which holds even when the reward function is not unimodal.*

Proof sketch (see Section 9.C.3 for the detailed proof). We start by the case where
the arm set forms a single neighborhood. Since LG is guaranteed that any arm it
selects will provide an estimate for all the other arms, this context is very similar
to a full information scenario. This explains why GG achieves both constant regret
depending on the gaps, and a gap-independent bound in $\sqrt{NT}$. Furthermore, the
hidden logarithmic constants come from carefully using Theorem 31 to separate the
linear term in n from the gaps when they are small.

The general case presents an additional complexity. Indeed, it is possible that
playing arm $n \neq n^\star$ is *locally optimal*, if n is the best arm in the neighborhood
of the current leader: playing n in that context would not be unlikely. To tackle
that scenario, we prove that pulling arm n at time t necessarily implies a *locally sub-
optimal play*, in some estimation neighborhood, at some point in the past (maximized
by the chosen value of α). We then show that this cannot happen after some
deterministic time w.h.p., leading to constant regret. However, since the sub-optimal
play might be any arm the constant now depends on the *worst local gap* Δ^2. □

Theorem 33 (Regret upper bound for Greedy-Grid)**.** *Suppose that GG is tuned with
confidence level $\delta_t = \frac{1}{N^2 t^3}$, and $\alpha = 1/4$. Then, for any $T \in \mathbb{N}$ it holds that*

$$\mathcal{R}_T = \widetilde{\mathcal{O}}_N \left(\sum_{n \in \mathcal{B}^\star} \frac{1}{\Delta_n} + \sum_{n \in \mathcal{S}} \frac{\log(T)}{\Delta_n} \wedge \Delta_n \left(\frac{\mathbb{1}\{n < n^\star\}}{\Delta_{v_l(n^\star)}^2} + \frac{\mathbb{1}\{n > n^\star\}}{\Delta_{v_r(n^\star)}^2} \right) \right) .$$

Additionally, it holds that $\mathcal{R}_T = \widetilde{\mathcal{O}} \left(\sqrt{(K + |\mathcal{B}^\star|)T} \right)$, for $K = \lfloor \log_{3/2}(N) \rfloor$.

Proof sketch (see Section 9.C.4 for the detailed proof). First we prove that, w.h.p.,
during a linear time range in t GG either played the grid or in $\mathcal{B}^\star$. Hence, arms
$n \in [N] \backslash \{\mathcal{S} \cup \mathcal{B}^\star\}$ are played a (universal!) constant number of times by GG in
expectation. Then, for $n \in \mathcal{S}$ the term in $\frac{\log(T)}{\Delta_n}$ comes from the standard analysis
of UCB [11]; while the constant bound comes from exploiting that after a constant
time n the LCB of $n^\star$ eliminates its neighboors w.h.p., and by extension the entire
grid. Finally, the constant bound $n \in \mathcal{B}^\star$ is derived similarly as the first bound of
Theorem 32. □

Discussion First, we show that being able to estimate $r(n)$ from the feedback
obtained after playing an arm k in its estimation neighborhood leads to a regret
independent of T for both LG and GG. For the former, the bound depends in general
on the worst *local gap* Δ, while for the latter only the actual gaps Δ_n (with $n^\star$)
are involved. This difference permits to obtain a problem-independent guarantee
for GG for any configuration of p and N. Furthermore, its scaling $\sqrt{K + |\mathcal{B}^*|} \leq$
$\sqrt{2n^* + \lfloor \log_{3/2}(N) \rfloor}$ can be much smaller than $\sqrt{N}$ if n^* is small.

Then, we would like to discuss the impact of the concentration bound presented
in Theorem 31 on the regret of both GG and LG. Indeed, a naive approach with

Hoeffding bounds would not allow to remove n from the first order term of the concentration bound, because of the multiplicative factor n in the definition of $r(n)$. A feature of our concentration bound is that the linear scaling in n does not appear in the first order term. Informally, this allows to exhibit terms of order $\widetilde{\mathcal{O}}_N(\Delta_n^{-1})$ in the regret analysis instead of $\widetilde{\mathcal{O}}(N^2\Delta_n^{-1})$, which can be significantly better for small gaps. A remark here is that the size of the grid in GG could be optimized as a larger grid makes the second order term in Theorem 31 smaller but is paid linearly in the regret.

We nevertheless highlight some potential for improvement in the analysis of LG. First, the local gaps Δ in the bound of LG could be replaced by (in spirit, referring to $\mathcal{S}$ for simplicity) $\min_{n \in [N]} |r(v_l^S(n)) - r(v_r^S(n))|$. It is clear though that this gap remains "local" and can be arbitrarily smaller than Δ_n for some arms $n \in [N]$, so the general interpretation of the results would be unchanged. Second, for simplicity, the analysis of LG was carried out using the constant upper bound of $\beta_{k,n}$ and $\xi_{k,n,F}$ but a tighter analysis could lead to a better dependency with respect to N.

We now justify the use of simple estimates in GG and LG. In practice, combining estimates would allow to use more samples for the estimation. However, this would make the algorithm slower, and we believe that the sampling requirement implemented in the algorithms makes the use of simple estimates efficient: potential uniform exploration in a neighborhood is replaced by a focus on a single arm, but the same quality of information is accrued. Furthermore, from a theoretical perspective union bounds over the samples collected by each arm might also cost a factor N in the analysis.

Lastly, while GG admits better theoretical guarantees, LG might be more appealing in practice because it does not require to explicitly compute confidence intervals. This means that the regret bounds provided for LG are conservative, and might be refined with tighter confidence bounds without changing the algorithm.

Adaptation for non-unimodal rewards While LG relies heavily on Assumption 2, GG can be readily adapted to handle non-unimodal reward functions. This is done by modifying the definition of the set of non-eliminated grid arms $\mathcal{C}_t$ to $\{s \in \mathcal{S}, U_s \geq L_{i_t^*}\}$ in Algorithm 19. In that case, the algorithm can no longer eliminate arms on the grid based on the elimination of other arms. This naturally induces that the number of plays of sub-optimal arms is no longer bounded by a constant. In Theorem 34 (see Appendix), we show that only the $\mathcal{O}(\log(T))$ term persists for $n \in \mathcal{S}$ in Theorem 33, while the problem-independent bound remains unchanged. Although we believe unimodality is necessary for achieving constant regret, this result demonstrates that, even without that assumption, GG can still provide the same logarithmic regret guarantees as UCB. However, it does so on a $|\mathcal{S}|$-armed bandit, rather than an N-armed bandits with $|\mathcal{S}| = \mathcal{O}(\log(N)) \ll N$ for large N.

Experimental results   In Section 9.D we present a benchmark of LG, GG, UCB, EXP3 and OSUB on synthetic data in terms of the expected regret $\mathcal{R}(T)$. This benchmark illustrates the strong performance of LG relative to the other approaches. Although GG offers more robust theoretical guarantees, particularly with sub-linear problem-independent bounds, LG proves to be more effective in practice. Several factors may explain this gap between theoretical guarantees and empirical performance. First, as discussed in the previous section, the worst-case local gap in the analysis of Local Greedy (Theorem 32) might be overly conservative. This worst-case scenario could occur under a combination of unfavorable conditions, such as poor initialization far from the optimal arm and a flat reward function, paired with bad luck in exploration. However, such a scenario is likely rare in practice and was not encountered in our experiments. Additionally, Local Greedy benefits from scenarios where it starts playing in the optimal neighborhood only after a few steps, a situation GG cannot exploit due to its need for sufficient statistical evidence to eliminate all suboptimal neighborhoods. While GG's caution leads to stronger theoretical guarantees, this comes at the cost of empirical performance. Moreover, GG's results are tied to the tightness of the confidence intervals in Theorem 32, a limitation that does not apply to LG. An interesting and challenging open problem remains whether LG can be modified to achieve the same theoretical guarantees as GG without sacrificing its performance. We leave this question for future work.

Conclusion

In this work, we consider a structured bandit problem where playing arm $n \in [N]$ gives a reward $r(n)$ determined by integers p, n and an unknown c.d.f F and yields an observation of a sample of F^{p+n} with probability $\frac{n}{n+p}$. The bandit problem studied in this work is structured since playing arm n gives a reward $r(n)$ determined by n, p and the unknown c.d.f F and with probability $\frac{n}{n+p}$ an observation of a sample of the distribution with c.d.f F^{n+p}.

While traditional bandit approaches give problem dependent bounds depending on T, algorithms GG and LG presented in this work have constant problem dependent bounds. Furthermore, GG and LG avoid a quadratic dependency in N for large T thanks to new concentration bounds introduced in Theorem 31. Overall, while GG has the best theoretical guarantees, LG has better constants and is therefore better suited for most practical problems (see the discussion at the end of Section 9.3 and experimental results in Section 9.D).

Whether an algorithm that has the theoretical guarantees of GG and the practical performance of LG can be designed is an interesting question. We believe that the main leverage to improve the practical performance of GG might be to derive tighter concentration bounds. Possible directions to improve over Theorem 31 might include: further refining the decomposition of the integral in (9.3) according to the value of F, further use "empirical" components (depending on estimates of F), or even using ideas from the proof of the DKW inequality [78] to avoid the union bounds

over the points of each interval in the decomposition. We leave these directions for future work.

To conclude, since in practice, a DSP can launch campaigns through multiple auctions, an interesting question is whether the current analysis could be extended to the case of A auctions where a play at time t is $(n_{a,t})_{a \in [A]}$ where $\sum_{a \in [A]} n_{a,t} = N$ and the reward is $\sum_{a \in [A]} r_a(n_{a,t})$ with r_a determined by integers p_a, $n_{a,t}$ and F_a in the same way that r depends on p, n_t and F. How to explore each auction in parallel in an efficient manner and how to handle the case where some auctions must be assigned zero players are then the main questions to solve.

9

Appendix 9

9.A   Properties of the expected reward function

In this appendix we prove the results presented in Section 9.2.1 of the paper, and discuss the shape of the expected reward.

9.A.1   Proof of Lemma 47

Lemma 47. *The expected reward function defined in Equation* (9.1) *satisfies,*

$$n \in [N] \mapsto r(n) = n \int_0^1 F^{p+n-1}(x) - F^{p+n}(x)\mathrm{d}x. \tag{9.3}$$

Proof. Given $\mathbf{v} = (v_i)_{i \in [n+p]} \sim F \times \cdots \times F$, we have

$$
\begin{aligned}
r(n) &= \mathbb{E}\left[(\mathbf{v}_{(1)} - \mathbf{v}_{(2)}) \mathbb{1}\left\{ \arg\max_{i \in [n+p]} v_i \in [n] \right\} \right] \\
&\overset{(i)}{=} \mathbb{E}\left[(\mathbf{v}_{(1)} - \mathbf{v}_{(2)}) \right] \mathbb{E}\left[\mathbb{1}\left\{ \arg\max_{i \in [n+p]} v_i \in [n] \right\} \right] \\
&\overset{(ii)}{=} \left(\mathbb{E}\left[\mathbf{v}_{(1)} \right] - \mathbb{E}\left[\mathbf{v}_{(2)} \right] \right) \times \frac{n}{n+p} \\
&= \frac{n}{n+p} \times \int_0^1 \mathbb{P}(\mathbf{v}_{(1)} > x) - \mathbb{P}(\mathbf{v}_{(2)} > x)\mathrm{d}x \\
&= \frac{n}{n+p} \times \int_0^1 \mathbb{P}(\mathbf{v}_{(2)} \leq x) - \mathbb{P}(\mathbf{v}_{(1)} \leq x)\mathrm{d}x \\
&\overset{(iii)}{=} \frac{n}{n+p} \times \int_0^1 ((n+p)F^{n+p-1}(x) - (n+p-1)F^{n+p}(x) - F^{n+p}(x))\mathrm{d}x \\
&= n \int_0^1 (F^{n+p-1}(x) - F^{n+p}(x))\mathrm{d}x \ .
\end{aligned}
$$

The first equality is the definition of $r(n)$ in Equation (9.1). Equality (i) follows by independence of the index of the maximum and the value of the maximum and second maximum. This is itself a consequence of the fact that the values are i.i.d.. Then equality (ii) follows since the distribution of the index of the maximum is uniform over $n + p$. This is also a consequence of the fact that the values are i.i.d. Lastly, equality (iii) follows from [32] (Equation 2.1.3) where for $k \in \{1, 2\}$, it is

shown that

$$\mathbb{P}(\mathbf{v}_{(k)} \leq x) = \sum_{i=n+p-k+1}^{n+p} \binom{n+p}{i} (1 - F(x))^{n+p-i} F(x)^i,$$

and the proof is concluded by substitution. □

9.A.2 Proof of Lemma 48

As a preliminary, we formally define the non-usual distributions considered in Lemma 48.

Truncated exponential distribution   Let $a > 0$ be some parameter. Then, we define a truncated exponential distribution of parameter a as the distribution with c.d.f. $F : x \mapsto \frac{1-e^{-ax}}{1-e^{-a}}$. Hence, $F(0) = 0$ and $F(1) = 1$, and the density of this distribution is the same as the density of the exponential distribution with same parameter on the segment $[0, 1]$, up to a normalization constant.

Complementary Beta distribution

Lemma 48. *Let F be the cumulative distribution function of a Bernoulli, truncated exponential or Complementary Beta distribution. Then, for any $p \in \mathbb{N}^*$, r in Equation (9.3) unimodal.*

Proof. We consider each family of distributions separately.

Bernoulli distributions   If F is the c.d.f. of $\mathcal{B}(q)$ (a Bernoulli distribution of parameter q), then $r(n)$ is equal to the probability that exactly one player from the coalition draws a value of 1, and every other player draw a value of 0. Hence, we obtain that $r(n) = nq(1 - q)^{n+p-1}$, which is trivially unimodal and maximized in $n^\star = \frac{-1}{\log(1-q)} \vee 1$, regardless of the size of the competition.

Truncated exponential distributions   Let $a > 0$ be the parameter of the distribution. Let $Q(x)$ be the inverse function of F (the quantile function), defined by $Q(x) = \frac{1}{a} \log \left(\frac{1}{1-x(1-e^{-a})} \right) = \frac{1}{a} \sum_{k=1}^{+\infty} \frac{x^k(1-e^{-a})^k}{k}$.

Let's denote by $q(x)$ the derivative of $Q(x)$, denoted by $q(x) = \sum_{k=0}^{+\infty} \lambda_k x^k$ where $\lambda_k = \frac{1}{a}(1 - e^{-a})^k$. Introducing theses functions allows us to rewrite $r(n)$ as follows,

$$
\begin{aligned}
r(n) &= n \int_0^1 F(v)^{p+n-1}(1 - F(v))\mathrm{d}v \\
&= n \int_0^1 x^{p+n-1}(1 - x)q(x)\mathrm{d}x \qquad \text{using } F(v) = x \\
&= n \int_0^1 x^{p+n-1}(1 - x)\left(\sum_{k=0}^{+\infty} \lambda_k x^k\right)\mathrm{d}x \\
&= \sum_{k=0}^{+\infty} \lambda_k \left(\frac{n}{p+n+k} - \frac{n}{p+n+k+1}\right) \\
&= \frac{n}{n+p}\lambda_0 + n \sum_{j=1}^{+\infty} \frac{1}{n+p+j}(\lambda_j - \lambda_{j-1}) \\
&= \lambda_0(1 - \frac{p}{n+p}) + \sum_{j=1}^{+\infty}(1 - \frac{p+j}{n+p+j})(\lambda_j - \lambda_{j-1}) \\
&= \lambda_0\left(-\frac{p}{n+p} + \sum_{j=1}^{+\infty} \underbrace{\frac{\lambda_{j-1} - \lambda_j}{\lambda_0}}_{\theta_j} \frac{p+j}{n+p+j}\right).
\end{aligned}
$$

where the last inequality follows since $\lim_{j\to\infty} \lambda_j = 0$. Remark that $\theta_j \geq 0$ since λ_j is decreasing and $\sum_{j=1}^{\infty} \theta_j = 1$.

The derivative of $r(n)$ is given by,

$$
\begin{aligned}
r'(n) &= \lambda_0\left(\frac{p}{(n+p)^2} - \sum_{j=1}^{+\infty} \theta_j \frac{p+j}{(n+p+j)^2}\right) \\
&= \lambda_0\left(\Theta_p(n) - \sum_{j=1}^{\infty} \theta_j \Theta_{p+j}(n)\right) \\
&= \lambda_0 \Theta_p(n)\left(1 - \sum_{j=1}^{\infty} \theta_j \Gamma_{p,p+j}(n)\right) \qquad \text{where } \Gamma_{p,p+j}(n) = \frac{\Theta_{p+j}(n)}{\Theta_p(n)}.
\end{aligned}
$$

the functions $\Gamma_{p,p+j}(n)$ are non-decreasing hence it is the same for their convex combination. As $\mathrm{sign}(r'(n)) = \mathrm{sign}(1 - \sum_{j=1}^{\infty} \theta_j \Gamma_{p,p+j}(n))$ it follows that $\mathrm{sign}(r'(n))$ is decreasing meaning $r(n)$ is unimodal.

Complementary beta distributions   Using the same change of variable as in the previous proof $(Y = F(X))$, we express the reward as follows,

$$
r(n) = n \times \mathbb{E}_{Y \sim Q}\left[Y^{n+p-1}(1 - Y)\right] ,
$$

where Q denotes the quantile function associated with c.d.f F (i.e. F^{-1}). By definition of F, Y follows a Beta distribution of parameters (a, b). We can thus compute the expected reward by using the explicit formula for moments of the Beta distribution,

$$\mathbb{E}_{X \sim \mathrm{B}(a,b)}[X^{n+p-1} - X^{n+p}] = \prod_{k=0}^{n+p-2} \frac{a+k}{a+b+k} - \prod_{k=0}^{n+p-1} \frac{a+k}{a+b+k}$$

$$= \prod_{k=0}^{n+p-2} \frac{a+k}{a+b+k} \times \left(1 - \frac{a+n+p-1}{a+b+n+p-1}\right)$$

$$= \prod_{k=0}^{n+p-2} \frac{a+k}{a+b+k} \times \frac{b}{a+b+n+p-1} .$$

Thanks to this expression, we prove the unimodality by analyzing the ratio $\frac{r(n+1)}{r(n)}$, that we first write as

$$\frac{r(n+1)}{r(n)} = \frac{n+1}{n} \times \frac{a+n+p-1}{a+b+n+p-1} \times \frac{a+b+n+p-1}{a+b+n+p}$$

$$= \frac{n+1}{n} \times \frac{a+n+p-1}{a+b+n+p} ,$$

and then obtain that this ratio is larger than 1 if and only if

$$(n+1)(a+n+p-1) \geq n(a+b+n+p) \iff n(a+n+p) + a + p - 1 \geq n(a+n+p) + bn$$

$$\iff n \geq \frac{a+p-1}{b} ,$$

which concludes the proof by showing the unimodality and expressing the value of the critical point.

$\square$

The proof of Lemma 48 highlights that the unimodality assumption is satisfied as soon as the quantile function, expressed as a power series, has its coefficients that slowly decrease (indeed, the k-th coefficient just needs to be smaller than $1 - \frac{1}{k}$ times the $k-1$-th one).

Similarly, the second proof technique highlights (up to standard algebraic manipulations) that unimodularity is guaranteed as soon as the function $n \mapsto 1 - E[X^{n+p-1}]/E[X^{n+p-2}]$ is log-concave.

9.A.3   Additional discussion on the unimodality of r

In this section, we plot the shape of $r(n)$ for some additional families of distribution that we conjecture to be unimodal from the plots.

Beta distribution   The following figure, illustrate the unimodal shape of $r(n)$ for different parameters for the Beta distribution and p.
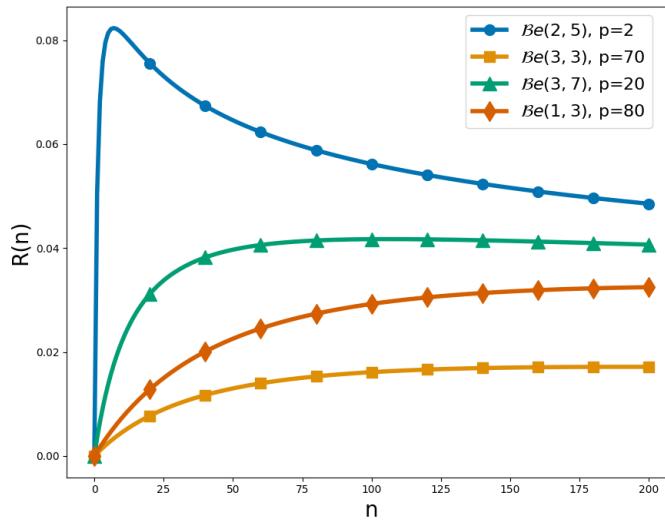

Figure 9.1: Shape of $r(n)$ when F is Beta

Kumaraswamy distribution   The cumulative distribution is defined by $F(x) = 1 - (1 - x^a)^b$ for some parameters (a, b) (we use the notation $K(a, b)$). The following figure, illustrate the unimodal shape of $r(n)$ for different parameters of $K(a, b)$ and p.
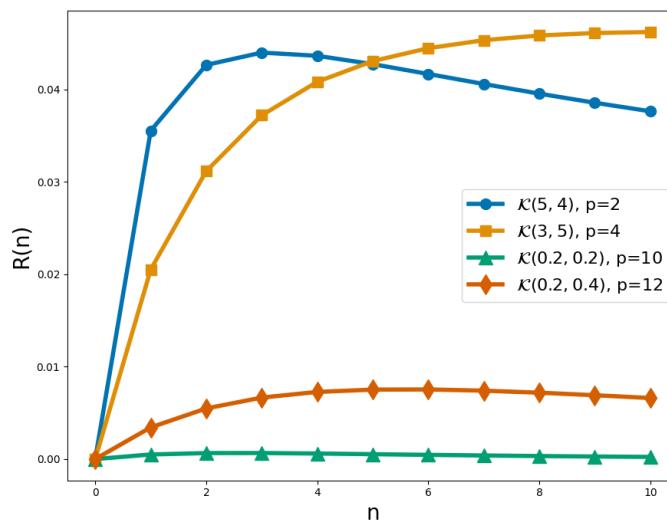

Figure 9.2: Shape of $r(n)$ when F is Kumaraswamy distribution

We now provide and discuss an example where Assumption 2 is not satisfied.

9.A.4 An example of distribution with non-unimodal rewards

Let us consider a discrete distribution supported on $\{0, 0.5, 1\}$. The counter-example emerges from putting all the probability mass in 0.5: let us consider a small $\epsilon > 0$, identify F with $\{\epsilon, 1 - \epsilon, 1\}$ and assume that there is no competition ($p = 0$). Then, we can verify with (9.3) that

$$r(n) = \frac{n}{2}(\epsilon^{n-1} + (1 - \epsilon)^{n-1} - (\epsilon^n + (1 - \epsilon)^n))$$
$$= \frac{n}{2}(\epsilon^{n-1}(1 - \epsilon) + \epsilon(1 - \epsilon)^{n-1}).$$

Consider $\epsilon = 0.15$, we get up to a precision 0.001 the values:

$$(r(n))_{n=1}^7 = (0.5, 0.255, 0.191, 0.190, 0.197, 0.200, 0.198)$$

showing that $r(n)$ is not unimodal in this case.

9.B Concentration bounds on simple reward estimates

9.B.1 Auxiliary results

Before presenting the proof of the theorem, we present two auxiliary results that are essential to its development.

9.B.1.1 Riemann sum approximation of the expected reward

The first result consists in upper bounding the deviation of a Riemann sum approximation of $r(n)$ (for some $n \in [N]$) with respect to its exact integral formulation. This result is also of practical interest, since it can prevent computing exact integrals at each step of the algorithms without altering their theoretical guarantees with an appropriate tuning of the approximation error.

Lemma 50 (Riemann sum approximation of $r(n)$). *Let $n \in [N]$, $D \in \mathbb{N}$, and define the grid $(x_s)_{s \in \{0, \dots, D-1\}} = \left\{0, \frac{1}{D}, \dots, \frac{D-1}{D}\right\}$. Then, the expected reward approximation*

$$\widetilde{r}(n) = n \times \frac{1}{D} \sum_{s=0}^{D-1} \left\{F(x_s)^{n+p-1} - F(x_s)^{n+p}\right\}.$$

satisfies

$$|r(n) - \widetilde{r}(n)| \leq \frac{n}{D} \ .$$

Proof. For any $j \in \mathbb{N}$ we consider

$$S_j = \frac{1}{D} \sum_{s=0}^{D-1} F^j(x_s) \quad \text{as an approximation of} \quad I_j = \int_0^1 F^j(x)\mathrm{d}x \ .$$

We recall that since F^j is a c.d.f., it is monotone, increasing and satisfies $F^j(0) = 0$ and $F^j(1) = 1$. This means that for any $s \in \{0, \dots, D-1\}$, it holds that $\forall x \in [x_s, x_{s+1}], F^j(x_s) \leq F^j(x) \leq F^j(x_{s+1})$. The linearity of the integral first provides that

$$I_j = \int_0^1 F^j(x)\mathrm{d}x = \sum_{s=0}^{D-1} \int_{\frac{s}{D}}^{\frac{s+1}{D}} F^j(x)\mathrm{d}x$$

Then, using this decomposition and the monotony of F^j we obtain that

$$S_j \leq \frac{1}{D} \sum_{s=0}^{D-1} F^j\left(\frac{s}{D}\right) \leq \int_0^1 F^j(x)\mathrm{d}x \leq \frac{1}{D} \sum_{s=0}^{D-1} F^j\left(\frac{s+1}{D}\right)$$

$$\leq \frac{1}{D} \sum_{s=0}^{D-1} F^j\left(\frac{s}{D}\right) + \frac{1}{D} \sum_{k=0}^{D-1}\left(F^j\left(\frac{s+1}{D}\right) - F^j\left(\frac{s}{D}\right)\right)$$

$$\leq \frac{1}{D} \sum_{s=0}^{D-1} F^j\left(\frac{s}{D}\right) + \frac{F^j(1) - F^j(0)}{D}$$

$$= S_j + \frac{1}{D} \ .$$

Therefore, we obtained that for any $j \in \mathbb{N}$ it holds that $S_j \leq I_j \leq S_j + \frac{1}{D}$. We conclude by using this result after splitting the reward as a difference of two integrals that can be expressed in this form, respectively with $j = n + p - 1$ and $j = n + p$. $\qquad\square$

9.B.1.2 Chernoff bounds for Bernoulli random variables

In the following lemma, we summarize the different concentration bounds that we use in the proof of Theorem 31.

Lemma 51 (Mutltiplicative Chernoff bounds). *Let $\widehat{\mu}_m$ be the empirical average of m i.i.d. Bernoulli random variables $X_1, \dots, X_m$ with expectation μ. Then, for any $\delta > 0$, each of the following bounds holds with probability at least $1 - \delta$,*

$$
\begin{cases}
|\widehat{\mu}_m - \mu| \leq \sqrt{\mu} \times \sqrt{\frac{3 \log(\frac{2}{\delta})}{m}} & \text{if } \mu \in I_0 := \left[\frac{3 \log(2/\delta)}{m}, 1 \right] , \\
\mu_m \leq \frac{6 \log(2/\delta)}{m} & \text{if } \mu \in I_1 := \left(\frac{\delta}{m}, \frac{3 \log(2/\delta)}{m} \right) , \\
\mu_m = 0 & \text{if } \mu \in I_2 := \left[0, \frac{\delta}{m} \right] .
\end{cases}
$$

Proof. We first tackle the case $\mu \in I_2$, where the bound is obtained by remarking that $\mathbb{P}(\mu_m > 0) \leq \mathbb{P}(\exists i \in [m] : X_i = 1) \leq m\mu \leq \delta$. The two other cases are obtained by using the multiplicative form of the well-known Chernoff bounds [51], that provide that

$$
\forall \gamma > 0, \quad \mathbb{P}(\widehat{\mu}_m \geq (1 + \gamma)\mu) \leq e^{-m\frac{\gamma^2 \mu}{2+\gamma}} \quad , \text{ and}
$$

$$
\forall \gamma \in [0, 1], \ \mathbb{P}(\widehat{\mu}_m \leq (1 - \gamma)\mu) \leq e^{-m\frac{\gamma^2 \mu}{2}} .
$$

When considering $\gamma \leq 1$ we can further write that

$$
\mathbb{P}(|\widehat{\mu}_m - \mu| \geq \gamma\mu) \leq 2e^{-m\frac{\gamma^2 \mu}{3}} .
$$

On the other hand, for $\gamma \geq 1$ the bound for the lower deviation is trivially 0 while the bound for the upper deviation can be written as follows,

$$
\gamma \geq 1 \Rightarrow \mathbb{P}(\widehat{\mu}_m \geq (1 + \gamma)\mu) \leq e^{-m\frac{\gamma^2 \mu}{2+\gamma}} \leq e^{-m\frac{\gamma^2 \mu}{3\gamma}} = e^{-m\mu\frac{\gamma}{3}} .
$$

Hence, the case separation between the intervals I_0 and I_1 simply consist in identifying the value μ for which a probability $1 - \delta$ can be obtained by setting an appropriate $\gamma \in [0, 1]$ or for $\gamma > 1$ in the above inequalities. More precisely, inverting the first bound provides $\gamma = \mu^{\frac{-1}{2}} \sqrt{\frac{3 \log(\frac{2}{\delta})}{m}}$, which is valid only if $\mu^{\frac{-1}{2}} \sqrt{\frac{3 \log(\frac{2}{\delta})}{m}} \leq 1 \Rightarrow \mu \geq \frac{3 \log(\frac{2}{\delta})}{m}$. This leads to the first confidence interval when $\mu \in I_0$. The same procedure for $\mu \in I_1$ leads to $\gamma = \mu^{-1} \frac{3 \log(\frac{2}{\delta})}{m}$, which provides the result stated in the lemma. This concludes the proof. $\square$

9.B.2   Proof of Theorem 31

Theorem 31 (Concentration of simple estimates)**.** *Consider any $n \in [N]$ and $k \in \mathcal{V}(n)$. Let $\widehat{r}_k(n)$ be defined according to (9.5) from m_k samples collected by k. Then, there exists some constants $\beta_{k,n}$ (depending on n, k, p) and $\xi_{k,n,F}$ (additionally depending on F) such that, with probability $1 - \delta$,*

$$
|\widehat{r}_k(n) - r(n)| \leq \beta_{k,n} \sqrt{\frac{\log\left(\frac{2\lceil n\sqrt{m_k} \rceil}{\delta} \right)}{m_k}} + n \times \xi_{k,n,F} \left(\frac{\log\left(\frac{2\lceil n\sqrt{m_k} \rceil}{\delta} \right)}{m_k} \right)^{\frac{n+p-1}{k+p}} . \quad (9.7)
$$

Furthermore, the constants admit universal upper bounds for any n, k, p, F. For instance if $m_k \geq 4$ it holds that $\beta_{k,n} \leq 33$ and $\gamma_{k,n,F} \leq 100$.

Proof. We build the proof from the Riemann sum approximation of the reward presented in Lemma 50 and the Chernoff bounds presented in Lemma 51. Defining some parameter $D \in \mathbb{N}$ that will be fixed later, we use the first result to consider the following approximation of the empirical reward estimate $\widehat{r}_k(n)$ by

$$\widetilde{r}_k(n) = \frac{1}{D} \sum_{s=0}^{D-1} \left(\widehat{F}_{k+p,m_k} \left(\frac{s}{D} \right)^{\frac{n+p-1}{k+p}} - \widehat{F}_{k+p,m_k} \left(\frac{s}{D} \right)^{\frac{n+p}{k+p}} \right) .$$

Thanks to Lemma 50, we know that $\widehat{r}_k(n) \in \left[\widetilde{r}_k(n) - \frac{n}{D}, \widetilde{r}_k(n) + \frac{n}{D} \right]$. For the rest of proof, we introduce the notation $F_s^j = F \left(\frac{s}{D} \right)^j$ for any $j \in [N]$, $\widehat{F}_s^{k+p} = \widehat{F}_{k+p,m_k} \left(\frac{s}{D} \right)$, and $\widehat{F}_{s,k,n} = \widehat{F}_{k+p,m_k} \left(\frac{s}{D} \right)^{\frac{n+p}{k+p}}$, so that

$$\widetilde{r}_k(n) := \frac{1}{D} \sum_{s=0}^{D-1} \left(\widehat{F}_{s,k,n-1} - \widehat{F}_{s,k,n} \right) ,$$

that we want to relate with

$$\widetilde{r}(n) := \frac{1}{D} \sum_{s=0}^{D-1} \left(F_s^{n+p-1} - F_s^{n+p} \right) .$$

We use that each variable $\widehat{F}_{s,k,n}$ can be expressed as the expectation of m_k i.i.d. Bernoulli random variables of expectation F_s^{k+p}, since $\widehat{F}_{k+p,m_k} = \frac{1}{m_k} \sum_{j=1}^{m_k} \mathbb{1}\{X_{k,j} \leq x\}$. Hence, we can use the confidence intervals providing by Lemma 51, according to the value of F_s^{k+p}. We define two critical values, corresponding to the switch between the different intervals I_0, I_1, I_2 in the lemma, and their closest upper point in the discretization grid. The first is

$$x_{0,k} = F^{-1} \left(\left(\frac{\delta}{m_k} \right)^{\frac{1}{k+p}} \right), \quad \text{and} \quad s_{0,k} = \lceil D x_{0,k} \rceil ,$$

and we recall that below $x_{0,k}$ it holds that $\widehat{F}_{k,i,n-1} = 0$ with probability larger than $1 - \delta$. Then, we define

$$x_{1,k} := F^{-1} \left(1 \wedge \left(4 \frac{\log \left(\frac{2}{\delta} \right)}{m_k} \right)^{\frac{1}{k+p}} \right), \quad \text{and} \quad s_{1,k} := \lceil D x_{1,k} \rceil .$$

We remark that we use a multiplicative factor 4 inside of F^{-1}, while Lemma 51 might suggest to use 3. We do that for technical reasons, that we will motivate at one stage of the proof. These terms depend both on the sample size m_k and the confidence level δ, but we omit them in the notation for simplicity. Then, for fixed values of these constants we decompose the estimator between the intervals

$I_0 = \{s_{1,k}, \ldots, D-1\}$, $I_1 = \{s_{0,k}, \ldots, s_{1,k}-1\}$, and $I_2 = \{0, \ldots, s_{0,k}-1\}$. Note that for the second interval to be non-empty it must hold that $s_{0,k} \leq s_{1,k}-1$, that we assume in the following, otherwise we can just remove this interval from the analysis.

For $s \geq s_{1,k}$, Lemma 51 guarantees that with probability larger than $1 - \delta$ it holds that

$$\widehat{F}_s^{k+p} \in \left[(1-\gamma_s)F_s^{k+p}, (1+\gamma_s)F_s^{k+p}\right] \quad \text{for } \gamma_s = F_s^{-\frac{k+p}{2}}\sqrt{3\frac{\log\left(\frac{2}{\delta}\right)}{m_k}},$$

while for $k \leq s_{1,k}-1$ we can use one of the two other bounds provided in the lemma. Using a union bound, all the confidence intervals hold simultaneously for the points in the sum and in $x_{1,k}$ with probability larger than $1 - (D+1)\delta$, which defines a "good" event

$$\mathcal{G} = \left\{\forall k \in I_2, \widehat{F}_s^{k+p} = 0, \quad \forall k \in I_1, \widehat{F}_s^{k+p} \leq 8\frac{\log\left(\frac{2}{\delta}\right)}{m_k}, \right.$$
$$\left. \forall k \in I_0, \widehat{F}_s^{k+p} \in \left[(1-\gamma_s)F_s^{k+p}, (1+\gamma_s)F_s^{k+p}\right]\right\}. \tag{9.9}$$

For the rest of the analysis, we assume that $\mathcal{G}$ holds. In particular, in that context there exists $D-s_{1,k}$ constants $(z_s)_{s \in \{s_{1,k}, \ldots, D-1\}}$ such that $\forall s \in I_0, \widehat{F}_s^{k+p} = (1+z_s)F_s^{k+p}$ and $z_s \in [-\gamma_s, \gamma_s]$. We now upper and lower bound $\widehat{r}_k(n)$ using these constants, first writing that under $\mathcal{G}$ it holds that

$$\widetilde{r}_k(n) = \frac{1}{D}\sum_{s=0}^{D-1}\left(\widehat{F}_{s,k,n-1} - \widehat{F}_{s,k,n}\right)$$

$$= \frac{1}{D}\sum_{s=s_{1,k}}^{D-1}\left(\widehat{F}_{s,k,n-1} - \widehat{F}_{s,k,n}\right) + \frac{1}{D}\sum_{s=0}^{s_{1,k}-1}\left(\widehat{F}_{s,k,n-1} - \widehat{F}_{s,k,n}\right)$$

$$= \frac{1}{D}\sum_{s=s_{1,k}}^{D-1}\left((1+z_s)^{\frac{n+p-1}{k+p}}F_s^{n+p-1} - (1+z_s)^{\frac{n+p}{k+p}}F_s^{n+p}\right) + \frac{1}{D}\sum_{s=s_{0,k}}^{s_{1,k}-1}\left(\widehat{F}_{s,k,n-1} - \widehat{F}_{s,k,n}\right),$$

where we used that all the terms are zero for indices smaller than $s_{0,k}$. We can thus express $\widehat{r}_k(n)$ as follows,

$$\widehat{r}_k(n) = \widetilde{r}(n) + n\mathcal{E}_0 + n\mathcal{E}_1,$$

with

$$\mathcal{E}_0 := \frac{1}{D} \sum_{s=s_{1,k}}^{D-1} ((1+z_s)^{\frac{n+p-1}{k+p}} - 1) F_s^{n+p-1} - \frac{1}{D} \sum_{s=s_{1,k}}^{D-1} ((1+z_s)^{\frac{n+p}{k+p}} - 1) F_s^{n+p} , \quad \text{and}$$

$$\mathcal{E}_1 := \frac{1}{D} \sum_{s=s_{0,k}}^{s_{1,k}-1} \left(\widehat{F}_{s,k,n-1} - \widehat{F}_{s,k,n} \right) - \sum_{s=0}^{s_{1,k}-1} \left(F_s^{n+p-1} - F_s^{n+p} \right) ,$$

so we can upper and lower bound $\widehat{r}_i(n)$ by upper and lower bounding $\mathcal{E}_0$ and $\mathcal{E}_1$ separately.

Bounding the individual terms of $\mathcal{E}_0$ For any $k \geq s_{1,k}$ we consider the term

$$\mathcal{E}_{0,s} := ((1+z_s)^{\frac{n+p-1}{k+p}} - 1) F_s^{n+p-1} - ((1+z_s)^{\frac{n+p}{k+p}} - 1) F_s^{n+p} .$$

We first re-arrange it in a more convenient way, remarking that

$$F_s^{n+p-1} = F_s^{n+p-1}(F_s + (1 - F_s)) = F_s^{n+p} + F_s^{n+p-1}(1 - F_s).$$

Using this result, we obtain that

$$\begin{aligned} \mathcal{E}_{0,s} &:= ((1+z_s)^{\frac{n+p-1}{k+p}} - 1) F_s^{n+p-1} - ((1+z_s)^{\frac{n+p}{k+p}} - 1) F_s^{n+p} \\ &= ((1+z_s)^{\frac{n+p-1}{k+p}} - 1) F_s^{n+p-1}(F_s + (1 - F_s)) - ((1+z_s)^{\frac{n+p}{k+p}} - 1) F_s^{n+p} \\ &= F_s^{n+p}((1+z_s)^{\frac{n+p-1}{k+p}} - 1) - (1+z_s)^{\frac{n+p}{k+p}} + 1) + F_s^{n+p-1}(1 - F_s)((1+z_s)^{\frac{n+p-1}{k+p}} - 1) , \end{aligned}$$

which simplifies to

$$\mathcal{E}_{0,s} = \underbrace{F_s^{n+p}((1+z_s)^{\frac{n+p-1}{k+p}} - (1+z_s)^{\frac{n+p}{k+p}})}_{\mathcal{E}_{0,s}^-} + \underbrace{F_s^{n+p-1}(1 - F_s)((1+z_s)^{\frac{n+p-1}{k+p}} - 1)}_{\mathcal{E}_{0,s}^+}$$

(9.10)

We remark that these two terms have opposite sign, $\mathcal{E}_{0,s}^+$ having the same sign as z_s. We first upper bound $\mathcal{E}_{0,s}$, starting with the case $z_s > 0$, for which it holds that

$$z_s \geq 0 \Rightarrow \mathcal{E}_{0,s} \leq \mathcal{E}_{0,s}^+ \leq \underbrace{((1+\gamma_s)^{\frac{n+p-1}{k+p}} - 1)}_{c_s} F_s^{n+p-1}(1 - F_s) .$$

The constant c_k is explicit from the definition of γ_s, and the bound holds for any $i \in [N + p]$ without restriction. However, if we only consider the case $\frac{n+p-1}{k+p} \leq 2$ then we can further write that

$$c_s \leq (1+\gamma_s)^2 - 1 \leq 2\gamma_s + \gamma_s^2 \leq 3\gamma_s ,$$

since $\gamma_s \leq 1$ for the values of s considered. Then, for $z_s \leq 0$ we use that

$$z_s \leq 0 \Rightarrow \mathcal{E}_{0,s} \leq \mathcal{E}_{0,s}^{-} \leq F_s^{n+p}((1+z_s)^{\frac{n+p-1}{k+p}} - (1+z_s)^{\frac{n+p}{k+p}})$$
$$= F_s^{n+p}(1+z_s)^{\frac{n+p-1}{k+p}}\left(1 - (1+z_s)^{\frac{1}{k+p}}\right) .$$

Using the notation $y_s = -z_s$ for convenience, we upper bound the last multiplicative term as follows,

$$(1-y_s)^{\frac{1}{k+p}} = e^{\frac{\log(1-y_s)}{k+p}} \geq 1 + \frac{\log(1-y_s)}{k+p}$$
$$= 1 - \frac{1}{k+p}\log\left(1 + \frac{y_s}{1-y_s}\right)$$
$$\geq 1 - \frac{1}{k+p} \times \frac{y_s}{1-y_s} ,$$

which leads to

$$y_s := -z_s \geq 0 \Rightarrow \mathcal{E}_{0,s} \leq \mathcal{E}_{0,s}^{-} \leq F_s^{n+p}(1-y_s)^{\frac{n+p-1}{k+p}}\frac{y_s}{(k+p)(1-y_s)}$$
$$= F_s^{n+p}(1-y_s)^{\frac{n+p-1}{k+p}-1}\frac{y_s}{k+p} .$$

We now remark that when $k + p \leq n + p - 1$ then the bound simply becomes $\mathcal{E}_{0,s}^{-} \leq F_s^{n+p} \times \frac{\gamma_s}{k+p}$. However, when $k + p > n + p - 1$ the upper bound is diverging when y_s gets close to 1. Since the upper bound is increasing in γ_s, and using that $n + p - 1 \geq \frac{2}{3}(k+p)$ we obtain that $\mathcal{E}_{0,s}^{-} \leq F_s^{n+p}\frac{\gamma_s}{i(1-\gamma_s)^{\frac{1}{3}}}$.

This is the motivation for calibrating the threshold $s_{1,k}$ so that $k \geq s_{1,k} \Rightarrow (1-\gamma_s)^{\frac{1}{3}} \geq \frac{1}{2}$, which is done by tuning $s_{1,k}$ so that $\gamma_{s_{1,k}} \leq \frac{7}{8}$ (hence the multiplicative 4 inside of F^{-1}).

We thus obtain that

$$z_s \leq 0, k \geq s_{1,k} \Rightarrow \mathcal{E}_{0,s} \leq 2F_s^{n+p}\frac{\gamma_s}{k+p} ,$$

which finally leads to

$$\mathcal{E}_{0,s} \leq \underbrace{3\gamma_s F_s^{n+p-1}(1-F_s)}_{\text{if } z_s \geq 0} \vee \underbrace{2\,F_s^{n+p}\frac{\gamma_s}{k+p}}_{\text{if } z_s \leq 0} . \qquad (9.11)$$

We now proceed to lower bound $\mathcal{E}_{0,s}$, using again Equation(9.10). The proof is

similar to the proof of the upper bound, for the case $z_s \geq 0$ we can write that

$$z_s \geq 0 \Rightarrow -\mathcal{E}_{0,s} \leq -\mathcal{E}_{0,s}^- \leq F_s^{n+p}(1+z_s)^{\frac{n+p-1}{k+p}}\left((1+z_s)^{\frac{1}{k+p}} - 1\right)$$

$$\leq F^{n+p}(1+\gamma_s)^{\frac{n+p-1}{k+p}}\frac{\gamma_s}{k+p}$$

$$\leq 2^{\frac{n+p-1}{k+p}}F^{n+p}\frac{\gamma_s}{k+p} \,,$$

using the concavity of $x \mapsto (1+x)^{\frac{1}{k+p}}$ and that $\gamma_s \leq 1$. Then, for the case $z_s \leq 0$ we use that

$$z_s \leq 0 \Rightarrow -\mathcal{E}_{0,s} \leq -\mathcal{E}_{0,s}^+ \leq F_s^{n+p-1}(1-F_s)\left(1 - (1+z_s)^{\frac{n+p-1}{k+p}}\right)$$

$$\leq F_s^{n+p-1}(1-F_s)\left(1 - (1-\gamma_s)^{\frac{n+p-1}{k+p}}\right).$$

If $\frac{n+p-1}{k+p} \leq 2$ we furthermore obtain that $(1-\gamma_s)^{\frac{n+p-1}{k+p}} \geq (1-\gamma_s)^2 \geq 1-2\gamma_s$, so

$$z_s \leq 0 \Rightarrow -\mathcal{E}_{0,s}^+ \leq 2\gamma_s F_s^{n+p-1}(1-F_s) \,.$$

Combining these results, we can lower bound $\mathcal{E}_{0,s}$ as follows,

$$\mathcal{E}_{0,s} \geq -\left\{\frac{2^{\frac{n+p-1}{k+p}}\gamma_s}{k+p}F_s^{n+p} \ \vee \ 2\gamma_s F_s^{n+p-1}(1-F_s)\right\} \,, \tag{9.12}$$

where the terms involved in this lower bound are analogous to the terms used in the upper bound up to some multiplicative constants.

Summary: bounds on $\mathcal{E}_0$ We start with the lower bound. Using Equation (9.12), we obtain that

$$n\mathcal{E}_0 \geq -\frac{n}{D}\sum_{s=s_{1,k}}^{D-1}\left\{\frac{2^{\frac{n+p-1}{k+p}}\gamma_s}{k+p}F_s^{n+p} \vee 2\gamma_s F_s^{n+p-1}(1-F_s)\right\}$$

$$\geq -\sqrt{\frac{3\log\left(\frac{2}{\delta}\right)}{m_k}} \times \left\{\frac{n}{D}\sum_{s=s_{1,k}}^{D-1}2^{\frac{n+p-1}{k+p}}\frac{F_s^{n+p-\frac{k+p}{2}}}{k+p} + \frac{n}{D}\sum_{s=s_{1,k}}^{D-1}2F_s^{n+p-1-\frac{k+p}{2}}(1-F_s)\right\}$$

The first sum can be trivially upper bounded by $2^{\frac{n+p-1}{k+p}}\frac{n}{k+p}$, which cannot be refined without more restrictive assumptions on F. For the second term, we use that $k+p \leq \frac{3}{2}(n+p)$ to exhibit the reward function associated with a number of players $n' := n - \frac{n(k+p)}{2(n+p)} \geq \frac{n}{4}$ and a competition of size $p' := p - \frac{p(k+p)}{2(n+p)} \geq \frac{p}{4}$.

$$\frac{n}{D} \sum_{s=s_{1,k}}^{D-1} F_s^{n+p-1-\frac{k+p}{2}}(1 - F_s) = n \times \frac{1}{D} \sum_{s=s_{1,k}}^{D-1} F_s^{n'+p'-1}(1 - F_s)$$

$$\leq n \times \int_0^1 F(x)^{\frac{n+p-1}{4}}(1 - F(x))\mathrm{d}x + \frac{n}{D}$$

$$= \frac{n}{n'} \times n' \int_0^1 F(x)^{n'+p'-1}(1 - F(x))\mathrm{d}x + \frac{n}{D}$$

$$\leq \frac{n}{n'+p'-1} + \frac{n}{D} = \frac{n}{n+p-\frac{k+p}{2}} + \frac{n}{D},$$

where we used that the reward is smaller than the probability that the coalition wins the auction, which is easily generalized even if $i/2$ is not integer.

We thus conclude the proof of the lower bound by writing that

$$n\mathcal{E}_0 \geq -4\left\{\left(\frac{n}{2(n+p)-(k+p)} + \frac{n}{2D}\right) + 2^{\frac{n+p-1}{k+p}-2} \times \frac{n}{k+p}\right\} \times \sqrt{\frac{3\log\left(\frac{2}{\delta}\right)}{m_k}},$$

$$(9.13)$$

where the worst scaling for the left-hand term in the maximum is attained in $k+p = 3\frac{n+p}{2}$ and provide $2\frac{n}{n+p}$, while for the right-hand term it is achieved in $k+p = \frac{n+p-1}{2}$ and provides $2\frac{n}{n+p-1}$.

As we already discussed, the upper bound can be expressed very similarly, remarking that the bound involving the terms γ_s/i have to be divided by two, and the other term have to be multiplied by $3/2$. We hence directly obtain that

$$n\mathcal{E}_0 \leq 6\left\{\left(\frac{n}{2(n+p)-(k+p)} + \frac{n}{2D}\right) + \frac{n}{3(k+p)}\right\} \times \sqrt{\frac{3\log\left(\frac{2}{\delta}\right)}{m_k}}. \qquad (9.14)$$

Bounds on $\mathcal{E}_1$ We start by upper bounding the second sum by 0. Under $\mathcal{G}$ we hence obtain that

$$n\mathcal{E}_1 \leq \frac{n}{D} \sum_{k\in I_1} (\widehat{F}_s^{k+p})^{\frac{n+p-1}{k+p}}$$

$$\leq \frac{n}{D} \sum_{k\in I_1} \left(8\frac{\log(2/\delta)}{m_k}\right)^{\frac{n+p-1}{k+p}}$$

$$\leq n\left(x_{1,k} - x_{0,k} + \frac{1}{D}\right)\left(8\frac{\log(2/\delta)}{m_k}\right)^{\frac{n+p-1}{k+p}},$$

which has a worst possible power of $2/3$ when $m_k \geq 8\log(2/\delta)$, corresponding to $k + p = \frac{3}{2}(n + p - 1)$. Replacing $x_{1,k}$ by its expression, we further obtain that

$$n\mathcal{E}_1 \leq n\left(8\frac{\log(2/\delta)}{m_k}\right)^{\frac{n+p-1}{k+p}} \times \left\{F^{-1}\left(1 \wedge \left(\frac{4\log\left(\frac{2}{\delta}\right)}{m_k}\right)^{\frac{1}{k+p}}\right) - F^{-1}\left(\left(\frac{\delta}{m_k}\right)^{\frac{1}{k+p}}\right) + \frac{1}{D}\right\}. \tag{9.15}$$

For the lower bound on $n\mathcal{E}_1^1$, we apply the exact same steps, remarking that the constant 8 at the very first step can be replaced by 4, since we now use the exact value of F^{k+p} in the upper bound. Furthermore, we have to remove the term $x_{0,k}$. We finally obtain

$$n\mathcal{E}_1 \geq -n\left(4\frac{\log(2/\delta)}{m_k}\right)^{\frac{n+p-1}{k+p}} \times \left\{F^{-1}\left(1 \wedge \left(\frac{4\log\left(\frac{2}{\delta}\right)}{m_k}\right)^{\frac{1}{k+p}}\right) + \frac{1}{D}\right\}. \tag{9.16}$$

Summary: bounds on $\widehat{r}_k(n)$ We conclude this proof by summarizing the results, and exhibiting the constants introduced in the theorem. First, by combining (9.13) and (9.16) we obtain the following lower bound,

$$\widehat{r}_k(n) \geq r(n) - \frac{n}{D} - n\left(4\frac{\log(2/\delta)}{m_k}\right)^{\frac{n+p-1}{k+p}} \times \left\{F^{-1}\left(1 \wedge \left(\frac{4\log\left(\frac{2}{\delta}\right)}{m_k}\right)^{\frac{1}{k+p}}\right) + \frac{1}{D}\right\}$$

$$- 4\left\{\left(\frac{n}{2(n+p)-(k+p)} + \frac{n}{2D}\right) + 2^{\frac{n+p-1}{k+p}-2} \times \frac{n}{k+p}\right\} \times \sqrt{\frac{3\log\left(\frac{2}{\delta}\right)}{m_k}}.$$

Then, by combining (9.14) and (9.15) we obtain the following upper bound,

$$\widehat{r}_k(n) \leq r(n) + \frac{n}{D} + 6\left\{\left(\frac{n}{2(n+p)-(k+p)} + \frac{n}{2D}\right) + \frac{n}{3(k+p)}\right\} \times \sqrt{\frac{3\log\left(\frac{2}{\delta}\right)}{m_k}}$$

$$+ n\left(8\frac{\log(2/\delta)}{m_k}\right)^{\frac{n+p-1}{k+p}} \times \left\{F^{-1}\left(1 \wedge \left(\frac{4\log\left(\frac{2}{\delta}\right)}{m_k}\right)^{\frac{1}{k+p}}\right) - F^{-1}\left(\left(\frac{\delta}{m_k}\right)^{\frac{1}{k+p}}\right) + \frac{1}{D}\right\}.$$

As a final step, we recall that in this proof $1 - \delta$ is the confidence level of the point estimate in each of the D points $(x_s)_{s \in \{0,...,D-1\}}$ and $x_{1,k}$. Hence, to obtain a confidence $1 - \delta$ on the full estimate $\widehat{r}_k(n)$ we need to multiply δ by $(D + 1)$ in the bounds presented above. As a final step, we choose $D + 1 = \lceil n\sqrt{m} \rceil$, so that the term $\frac{n}{D} \leq \frac{1}{\sqrt{m_k}}$ becomes a second order term in the bounds.

After replacing δ and D by the appropriate values, we hence obtain that

$$\widehat{r}_k(n) \geq r(n) - \frac{1}{\sqrt{m_k}} - \beta_{k,n}^- \sqrt{\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}} - n \times \xi_{k,n,F}^- \left(\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}\right)^{\frac{n+p-1}{k+p}}.$$

(9.17)

with

$$\beta_{k,n}^- = 4\sqrt{3}\left\{\left(\frac{n}{2(n+p)-(k+p)} + \frac{n}{2(\lceil n\sqrt{m_k}\rceil - 1)}\right) + 2^{\frac{n+p-1}{k+p}-2} \times \frac{n}{k+p}\right\}, \text{ and}$$

$$\xi_{k,n,F}^- = 4^{\frac{n+p-1}{k+p}}\left\{F^{-1}\left(1 \wedge \left(\frac{4\log\left(\frac{2}{\delta}\right)}{m_k}\right)^{\frac{1}{k+p}}\right) + \frac{1}{\lceil n\sqrt{m_k}\rceil - 1}\right\}.$$

(9.18)

Symmetrically, we obtain that

$$\widehat{r}_k(n) \leq r(n) + \frac{1}{\sqrt{m_k}} + \beta_{k,n}^+ \sqrt{\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}} + n \times \xi_{k,n,F}^+ \left(\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}\right)^{\frac{n+p-1}{k+p}}.$$

(9.19)

with

$$\beta_{k,n}^+ = 6\sqrt{3}\left\{\left(\frac{n}{2(n+p)-(k+p)} + \frac{n}{2(\lceil n\sqrt{m_k}\rceil - 1)}\right) + \frac{n}{3(k+p)}\right\}, \text{ and}$$

$$\xi_{k,n,F}^+ = 8^{\frac{n+p-1}{k+p}}\left\{F^{-1}\left(1 \wedge \left(\frac{4\log\left(\frac{2}{\delta}\right)}{m_k}\right)^{\frac{1}{k+p}}\right) - F^{-1}\left(\left(\frac{\delta}{m_k}\right)^{\frac{1}{k+p}}\right) + \frac{1}{\lceil n\sqrt{m_k}\rceil - 1}\right\}.$$

(9.20)

Hence, we obtain the statement of (9.7) by choosing $\beta_{k,n} = \beta_{k,n}^- \vee \beta_{k,n}^-$ and $\xi_{k,n,F} = \xi_{k,n,F}^+ \vee \xi_{k,n,F}^-$. Furthermore, it is clear from their expression and the constraint $k+p \in \left[\frac{n+p}{2}, 3\frac{n+p}{2}\right]$ that these two constants are bounded by by absolute constants. Their expression provided in the theorem comes from choosing the worst admissible value of k for each of their components. This concludes the proof of the theorem. $\quad\square$

Remark 6 (Improved constants for practical implementations). *We can further improve the constants of the bounds according to the position of i with respect to $n + p - 1$.*

- *In Equation (9.11) (upper bound): the constant 3 can be improved to 1 if $i \geq n+p-1$, while the constant 2 can be improved to $\frac{n+p-1}{k+p}$ if $i \leq n+p-1$.*

- *In Equation (9.12) (lower bound): the constant 2 on the right-hand side can be improved to 1 if $n+p-1 \leq i$.*

These improved constants translate easily to the upper and lower bounds presented in (9.13) *and* (9.14).

9.B.3   Proof of Lemma 49

In this part, we prove the tighter confidence bounds, assuming that the quantile function of the value distribution is Lipschitz.

Lemma 49 (Improved bound for Lipschitz quantile function)**.** *Assume that $k \in \mathcal{V}(n)$ and F^{-1} is L-Lipschitz, then there exists an absolute constant ξ such that with probability $1 - \delta$ it holds that*

$$|\widehat{r}_k(n) - r(n)| \leq \beta_{k,n} \sqrt{\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}} + \xi L \log\left(\frac{4\lceil n\sqrt{m_k}\rceil}{\delta}\right) \left(\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}\right)^{\frac{n+p-1}{k+p}}.$$

$$(9.8)$$

Proof. As the result indicates, the improvement comes from providing finer upper and lower bound on the term $n\mathcal{E}_1$ in the proof of Theorem 31. We can start the refined analysis from Equations (9.16) and (9.15).

Let us consider first the upper bound. In that case, the main ingredient comes from refining the upper bound of the difference $x_{1,k} - x_{0,k}$. To do that, we first provide an upper bound on $1 \wedge \left(\frac{4\log\left(\frac{2}{\delta}\right)}{m_k}\right)^{\frac{1}{k+p}} - \left(\frac{\delta}{m_k}\right)^{\frac{1}{k+p}}$. We recall that, as in the previous proof, this δ should be multiplied by $\lceil n\sqrt{m_k}\rceil$ at the end of the computation. We omit this term for now for simplicity. We consider a first case where the first term is equal to 1, which leads to

$$1 - \left(\frac{\delta}{m_k}\right)^{\frac{1}{k+p}} = 1 - e^{-\frac{1}{k+p}\log\left(\frac{m_k}{\delta}\right)}$$

$$\leq \frac{1}{k+p}\log\left(\frac{m_k}{\delta}\right)$$

$$\leq \frac{1}{k+p}\log\left(\frac{4\log(2/\delta)}{\delta}\right)$$

$$= \frac{1}{k+p} \times \left\{\log\left(\frac{4}{\delta}\right) + \log\log\left(\frac{2}{\delta}\right)\right\}.$$

For the alternative case we obtain a similar result,

$$\left(\frac{4\log\left(\frac{2}{\delta}\right)}{m_k}\right)^{\frac{1}{k+p}} - \left(\frac{\delta}{m_k}\right)^{\frac{1}{k+p}} = \left(\frac{4\log\left(\frac{2}{\delta}\right)}{m_k}\right)^{\frac{1}{k+p}}\left(1 - \left(\frac{\delta}{4\log(2/\delta)}\right)^{\frac{1}{k+p}}\right)$$

$$\leq \left(\frac{4\log\left(\frac{2}{\delta}\right)}{m_k}\right)^{\frac{1}{k+p}} \times \frac{1}{k+p} \times \left\{\log\left(\frac{4}{\delta}\right) + \log\log\left(\frac{2}{\delta}\right)\right\},$$

so the two upper bounds simplify to

$$\left\{ 1 \vee \left(\frac{4 \log \left(\frac{2}{\delta} \right)}{m_k} \right)^{\frac{1}{k+p}} \right\} \times \frac{1}{k+p} \times \left\{ \log \left(\frac{4}{\delta} \right) + \log \log \left(\frac{2}{\delta} \right) \right\}$$

Next, we use the assumption that F^{-1} is L-Lipschitz to obtain that

$$n(x_{1,k} - x_{0,k}) := n \left(F^{-1} \left(1 \wedge \left(\frac{4 \log \left(\frac{2}{\delta} \right)}{m_k} \right)^{\frac{1}{k+p}} \right) - F^{-1} \left(\left(\frac{\delta}{m_k} \right)^{\frac{1}{k+p}} \right) \right)$$

$$\leq \frac{Ln}{k+p} \times \left\{ \log \left(\frac{4}{\delta} \right) + \log \log \left(\frac{2}{\delta} \right) \right\}$$

$$\leq 4 \frac{Ln}{n+p} \times \log \left(\frac{4}{\delta} \right) \ .$$

By substituting δ by $\delta/(\lceil n\sqrt{m_k} \rceil)$ we obtain that the linear dependency in n obtained with the previous analysis is refined to a $\log \left(\frac{4 \lceil n \sqrt{m_k} \rceil}{\delta} \right)$ for the upper bound, which matches the result at this point.

We now consider the lower bound, and work on refining the upper bound of the term $-n\mathcal{E}_1$ in the proof of Theorem 31. To do that, we consider a new intermediary point $x'_{0,k} = F^{-1} \left(\frac{\delta}{m \times n^{\frac{k+p}{n+p-1}}} \right)$. For the rest of the proof we get back to the discretized formulation of the error (with generic step D^{-1}), and upper bound

$$-n\mathcal{E}_1 \leq \underbrace{\frac{n}{D} \sum_{k \in I_1} F_s^{n+p-1}}_{n\mathcal{E}_1^0} + \underbrace{\frac{n}{D} \sum_{k \in I_2} F_s^{n+p-1}}_{n\mathcal{E}_1^1} \ .$$

We remark that we can use the upper bound provided for $n(x_{1,k} - x_{0,k})$ to upper bound $n\mathcal{E}_1^1$, obtaining the same result up to some multiplicative constants. Hence, it remains to upper bound $\mathcal{E}_1^0$, for which additional steps are needed. However, we will simply use the exact same trick as before: we consider another sub-interval I'_2, for which this time it holds that $k \in I'_2 \Rightarrow F_s^{k+p} \leq \frac{\delta}{m \times n^{\frac{k+p}{n+p-1}}}$. Then, we have that

$$n\mathcal{E}_1^0 = \frac{n}{D} \sum_{k \in I'_2} F_s^{n+p-1} + \frac{n}{D} \sum_{k \in I_2 \setminus I'_2} F_s^{n+p-1}$$

$$\leq \frac{n}{D} \sum_{k \in I'_2} \frac{1}{n} \left(\frac{\delta}{m_k} \right)^{\frac{n+p-1}{k+p}} + \frac{n}{D} \sum_{k \in I_2 \setminus I'_2} \left(\frac{\delta}{m_k} \right)^{\frac{n+p-1}{k+p}}$$

$$\leq \frac{|I'_2|}{D} \left(\frac{\delta}{m_k} \right)^{\frac{n+p-1}{k+p}} + n \times \frac{|I_2| - |I'_2|}{D} \left(\frac{\delta}{m_k} \right)^{\frac{n+p-1}{k+p}} \ .$$

Just as before, we show that $\frac{|I_0| - |I_0'|}{D}$ cannot be too large if the quantile function is Lipschitz. More precisely, we obtain that

$$\left(\frac{\delta}{m_k}\right)^{\frac{1}{k+p}} - \left(\frac{\delta}{m_k n^{\frac{k+p}{n+p-1}}}\right)^{\frac{1}{k+p}} = \left(\frac{\delta}{m_k}\right)^{\frac{1}{k+p}} \frac{\log(n)}{n+p-1} ,$$

so that we can finally write that

$$n\mathcal{E}_1^0 \leq \left(\frac{\delta}{m_k}\right)^{\frac{n+p-1}{k+p}} + n \times \left(\left(\frac{\delta}{m_k}\right)^{\frac{1}{k+p}} \times \frac{\log(n)}{n+p-1} \times L + \frac{1}{D}\right) \times \left(\frac{\delta}{m_k}\right)^{\frac{n+p-1}{k+p}}$$

$$\leq \left(\frac{\delta}{m_k}\right)^{\frac{n+p-1}{k+p}} + n \times \left(\frac{\log(n)}{n+p-1} \times L + \frac{1}{n\sqrt{m_k}}\right) \times \left(\frac{\delta}{m_k}\right)^{\frac{n+p-1}{k+p}} ,$$

which is sufficient to conclude, since the multiplicative constants to $m_k^{-\frac{n+p-1}{k+p}}$ are clearly dominated by $\log\left(\frac{4\lceil n\sqrt{m_k}\rceil}{\delta}\right)$. $\qquad\square$

9.B.4 Empirical UCB and LCB

The UCB and LCB in Equation (9.17) and Equation (9.19) depend explicitly on the unknown F via $\xi_{k,n,F}^+$ and $\xi_{k,n,F}^-$ and therefore cannot be used in the implementation of GG.

Below, we give empirical UCB ($\widehat{U}_k(n,\delta)$) and LCB ($\widehat{L}_k(n,\delta)$) by replacing $\xi_{k,n,F}^+$ and $\xi_{k,n,F}^-$ by empirical estimates $\widehat{\xi}_{k,n}^+$ and $\widehat{\xi}_{k,n}^-$:

$$\widehat{U}_k(n,\delta) = \widehat{r}_k(n) + \frac{1}{\sqrt{m_k}} + \beta_{k,n}^- \sqrt{\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}} + n \times \widehat{\xi}_{k,n}^- \left(\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}\right)^{\frac{n+p-1}{k+p}}$$

$$\tag{9.21}$$

$$\widehat{L}_k(n,\delta) = \widehat{r}_k(n) - \frac{1}{\sqrt{m_k}} - \beta_{k,n}^+ \sqrt{\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}} - n \times \widehat{\xi}_{k,n}^+ \left(\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}\right)^{\frac{n+p-1}{k+p}}$$

$$\tag{9.22}$$

where $\beta_{k,n}^{-}$ and $\beta_{k,n}^{+}$ are defined in Equation (9.18) and Equation (9.20), and

$$\widehat{\xi}_{k,n}^{-} = 4^{\frac{n+p-1}{k+p}} \left\{ \frac{\hat{d}_{k+p} + 1}{\lceil n\sqrt{m_k}\rceil - 1} \right\},$$

$$\widehat{\xi}_{k,n}^{+} = 8^{\frac{n+p-1}{k+p}} \left\{ \frac{\hat{d}_{k+p} + 1}{\lceil n\sqrt{m_k}\rceil - 1} \right\}$$

for

$$\hat{d}_{k+p} = \inf\{d \in \{0, \dots, \lceil n\sqrt{m_k}\rceil - 1\}, \hat{F}_d^{k+p} \geq 8\frac{\log(2\lceil n\sqrt{m_k}\rceil/\delta)}{m_k}\}$$

if the infimum exists and $\hat{d}_{k+p} = 1$ otherwise.

It is a corollary of Theorem 31 that $\widehat{U}_k(n)$ and $\widehat{L}_k(n)$ are indeed high probability upper and lower bounds of the true reward:

Corollary 5 (Explicit upper and lower bounds). *It holds that*

$$\mathbb{P}(\widehat{L}_k(n, \delta) \leq r(n) \leq \widehat{U}_k(n, \delta)) \geq 1 - \delta$$

Proof. With $D = \lceil n\sqrt{m_k}\rceil$ and $x_{1,k} = F^{-1}\left(1 \wedge \left(\frac{4\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}\right)^{\frac{1}{k+p}}\right)$ the good event $\mathcal{G}$ defined in Equation (9.9) implies that with probability $1 - \delta$ the following event $\mathcal{H}$ holds:

$$\mathcal{H} = \left\{ \forall k \leq \lceil Dx_{1,k}\rceil, \widehat{F}_s^{k+p} \leq 8\frac{\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k} \right\}.$$

Under $\mathcal{H}$, and since by definition $\hat{F}_{\hat{d}_{k+p}}^{k+p} \geq 8\frac{\log(2\lceil n\sqrt{m_k}\rceil/\delta)}{m_k}$, it holds that

$$\frac{\hat{d}_{k+p}}{D} \geq x_{1,k} = F^{-1}\left(1 \wedge \left(\frac{4\log\left(\frac{2\lceil n\sqrt{m_k}\rceil}{\delta}\right)}{m_k}\right)^{\frac{1}{k+p}}\right)$$

This implies that $\widehat{\xi}_{k,n}^{-} \geq \xi_{k,n,F}^{-}$ and $\widehat{\xi}_{k,n}^{+} \geq \xi_{k,n,F}^{+}$.

Therefore, we can incorporate in the proof of Theorem 31 the fact that the good event $\mathcal{G}$ implies $\widehat{\xi}_{k,n}^{-} \geq \xi_{k,n,F}^{-}$ and $\widehat{\xi}_{k,n}^{+} \geq \xi_{k,n,F}^{+}$ and obtain the stated result.   □

9.C Regret analysis of Local-Greedy and Greedy-Grid

9.C.1 Clarification on the feedback received by the algorithms

In this section, we consider the case where a feedback (in the form of a sample from a power of F) is gathered only when an auction is won. If this is not the case, the decision-maker only knows that the coalition lost the auction. Therefore, if at time t, n_t agents are assigned to an auction and the auction is lost, it makes sense to continue assigning n_t agents to the auction at time $t+1$, in order to gather the information that the algorithm wanted to obtain. The meta algorithm called CoMAB for coalition multi-armed bandits described in Algorithm 20 implements this strategy.

Algorithm 20: CoMAB

Init: $\mathcal{J}_0 = \varnothing$, $m = 1$
Input: Algo

1 **for** $t = 1 \ldots T$ **do**
2   **if** $t > 1$ *and auction at* $t - 1$ *is not won* **then**
3   Play $n_t = n_{t-1}$;   ▷ `Play same arm until an auction is won`
4   **else**
5   Play $n_t = \mathrm{Algo}(\mathcal{J}_{m-1})$;   ▷ `When an auction is won play as`
        `prescribed by input Algo`
6   **if** *Auction is won* **then**
7   Observe w_{n_t} a sample from F^{n_t+p}
8   Set $\mathcal{J}_m = \mathcal{J}_{m-1} \cup \{(w_{n_t}, n_t, m)\}$; ▷ `Record feedback obtained when`
        `winning`
9   Update $m = m + 1$

Local-Greedy and Greedy-Grid are then defined as CoMAB applied on π_{LG} and π_{GG} for local greedy and greedy-grid respectively. These policies associate a play n_m to an history $\mathcal{J}_{m-1}$. In Algorithm 18 and Algorithm 19, the policies are called sequentially T times and feedback is observed after each request. This gives an implicit definition of the policies.

Lemma 52 expresses the regret of CoMAB in function of the behavior of any policy π when feedback is observed after each request.

Lemma 52 (Regret of CoMAB)**.** *Consider a policy π that associate to every $\mathcal{J}_{m-1}$ a play n_m^π. Define $\mathcal{J}_0 = \varnothing$ and $\mathcal{J}_m = \mathcal{J}_{m-1} \cup \{w_{n_m^\pi}, n_m^\pi, m\}$ where $w_{n_m^\pi}$ is a sample from a distribution with c.d.f $F^{n_m^\pi+p}$ and $n_m^\pi = \pi(\mathcal{J}_{m-1})$. Consider $m_n^\pi(m)$ the number of times π returns n after m calls of π.*

After T iterations, CoMAB based on π has regret:

$$\mathcal{R}_T \le \sum_{n=1}^{N} \mathbb{E}[m_n^\pi(T)]\frac{p+n}{n}(r(n^*) - r(n))$$

Proof. Call n_t the play chosen by CoMAB at time t,

$$\eta_t = \mathbb{1}\{\text{The auction is won at time } t\},$$

$$m_n(t) = |\{\rho \le t, n_\rho = n \text{ and } \eta_\rho = 1\}|$$

the number of times that n is played and the auction is won up to time t and

$$Z_{n,m}(t) = |\{\rho \le t, n_\rho = n \text{ and } m \le m_n(\rho) < m+1\}|$$

the number of times that n has been played between the m-th time n won an auction and the $m+1$-th time. Note that $m_n(t) \le m_n^\pi(t)$ since at time t, π has been called at most t times.

The regret of CoMAB satisfies:

$$
\begin{aligned}
\mathcal{R}_T &= \sum_{t=1}^{T} \mathbb{E}[r(n^*) - r(n_t)] \\
&= \sum_{n=1}^{N} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{n_t = n\}\right](r(n^*) - r(n)) \\
&= \sum_{n=1}^{N} \mathbb{E}\left[\sum_{m=1}^{m_n(T)} Z_{n,m}(T)\right](r(n^*) - r(n)) \\
&= \sum_{n=1}^{N} \mathbb{E}\left[\sum_{m=1}^{m_n(T)} Z_{n,m}(T)\right](r(n^*) - r(n)) \\
&\le \sum_{n=1}^{N} \mathbb{E}[m_n^\pi(T)]\frac{p+n}{n}(r(n^*) - r(n))
\end{aligned}
$$

where in the second to last inequality, we used the independence between $n_t = n$ and $Z_{n,m_t(n)}(T)$ as $n_t = n$ depends only on the history at times $t < m_t(n)$ while $Z_{n,m_t(n)}(T)$ depends only on times $t \ge m_t(n)$.

$\square$

9.C.2   Auxiliary result

Before proving the theorems, we present an auxiliary result from [14] that we to derive upper bounds that can be recovered explicit in the proof of the theorems.

Since the proof is simple, we recall it for completeness.

Lemma 53 (Lemma 4 from [14])**.** *For any $\zeta \geq 1$, the mapping*

$$f_\zeta : x \in [(\zeta + 2)^\zeta \vee 3, \infty) \mapsto \sup \left\{ t \in \mathbb{N} : \frac{t}{\log(t)^\zeta} \leq x \right\}$$

satisfies

$$f_\zeta(x) \leq (\zeta + 2)^\zeta \times \log(x)^\zeta x.$$

Proof. We start by remarking that the function $g(x) = \frac{x}{\log(x)^\zeta}$ is strictly increasing for all $x \geq e^\zeta$. Now, consider a value $s = Ax \log(x)^\zeta$ for some $A > 0$, such that $s \geq 3 \vee e^\zeta$. By the monotonicity of $\frac{t}{(\log t)^\zeta}$, we have that

$$t > s \Rightarrow \frac{t}{(\log(t)^\zeta} > \frac{s}{(\log(s)^\zeta} = x \times \frac{A \log(x)^\zeta}{(\log(A) + \log(x) + \zeta \log(\log(x)))^\zeta} \; .$$

Then, for $x \geq A \geq 3$, it holds that $\log(A) + \log(x) + \zeta \log(\log(x)) \leq (\zeta + 2) \log(x)$, so we can simply choose $A = (\zeta + 2)^\zeta$ to obtain the result.

All that is left is to verify that for this choice, $s = (\zeta + 2)^\zeta \times \log(x)^\zeta x \geq 3 \vee e^\zeta$, but this clearly holds for all $x \geq 3$ and $\zeta > 0$. $\qquad\square$

9.C.3   Proof of Theorem 32

Theorem 32 (Regret bound for Local-Greedy)**.** *Let $\Delta := \min_{n \in [N-1]} |r(n+1) - r(n)|$ (worst local gap). Under Assumption 2 and with $\alpha = (\log_{3/2} N + 1)^{-1}$, the regret of* `LG` *is upper bounded by a **problem-dependent constant**: there exists $(C_n)_{n \in [N] \backslash \{n^\star\}}$, each satisfying $C_n = \widetilde{\mathcal{O}}_N \left(\frac{\Delta_n}{\Delta^2} \right)$, such that $\mathcal{R}_T \leq \sum_{n \in [N] \backslash n^\star} C_n$.*

Additionally, if the arm set forms a single estimation neighborhood, that is $\forall n \in [N] : \mathcal{V}(n) \supset [N]$, then each constant C_n can be refined to $\widetilde{\mathcal{O}}_n (\Delta_n^{-1})$, providing $\mathcal{R}_T = \widetilde{\mathcal{O}}(\sqrt{NT})$, which holds even when the reward function is not unimodal.

Proof. First, we denote by $\widetilde{r}_t(n)$ the reward estimate used for arm n at time t, and by $\widehat{r}_{k,t}(n)$ its value when the arm used to compute the estimate is fixed to $k \in \mathcal{V}(n)$. The proofs rely on concentration bounds on $\widetilde{r}_t(n)$ derived from Theorem 31, with a confidence level δ_t that will be fixed later. However, we will use this result with extra care given that the identity of the arm k used to compute the estimate is a random variable, as well as its sample size $m_k(t)$. This issue is tackled with appropriate union bounds. Furthermore, in order to simplify the presentation we denote by $\mathcal{E}(m, \delta)$ the maximal diameter (as a function of k) of the confidence interval provided by Equation (9.7), defined by a number of plays m of the arm used to estimate, and by a confidence level δ. More precisely, with the notation of Theorem 31, for any $(k, n) \in [N]^2$ we write that $|\widehat{r}_{k,t}(n) - r(n)| \leq \mathcal{E}(m_k(t), \delta_t)$ with probability at least

$1 - \delta_t$. Furthermore, $\mathcal{E}(m_k(t), \delta_t)$ is increasing in δ_t and decreasing in $m_k(t)$. Finally, we use the notation $K = \lceil \log_{3/2}(N) \rceil$, so that $\alpha = \frac{1}{K+1}$.

We now prove the first statement of the theorem, by upper bounding the number of plays of each sub-optimal arm.

Single neighborhood Consider any sub-optimal arm n. The main ingredient of the proof is to tackle the forced sampling by using that if n is pulled at time t, then it is either pulled "on purpose" or due to forced sampling. However, if it is forced sampled then it must have been selected "on purpose" by being the best empirical arm in the neighborhood at some previous point in time. We hence consider the following good event

$$\mathcal{G}_t = \{\forall s \in \{t - \lfloor \alpha t \rfloor, \dots, t\}, \ (\forall k \in \mathcal{V}(n) : m_k(s) \geq \alpha t), \ |\widehat{r}_{k,s}(n) - r(n)| \leq \mathcal{E}(m_k(s), \delta_s)\} \ .$$

Using Theorem 31, $\mathcal{G}_t$ holds with probability at least $1 - |\mathcal{V}(n)| t^2 \delta_t$, where we used a crude union bound on the values of s, k and $m_k(t)$ (t could be replaced by $t - \lfloor \alpha \rfloor t \rfloor$). Using this result, we first upper bound the number of plays of n up to horizon T as follows,

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{n_t = n\}\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\exists s \in \{t - \lfloor \alpha t \rfloor, \dots, t\} : \widehat{r}_s(n) \geq \widehat{r}_s(n^\star)\}\right]$$

$$\leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\exists s \in \{t - \lfloor \alpha t \rfloor, \dots, t\} : \widehat{r}_s(n) \geq \widehat{r}_s(n^\star)\} \mathbb{1}\{\mathcal{G}_t\}\right] + \sum_{t=1}^{T} \mathbb{P}(\bar{\mathcal{G}}_t) \ .$$

As discussed above, the second term satisfies $\sum_{t=1}^{T} \mathbb{P}(\bar{\mathcal{G}}_t) \leq \sum_{t=1}^{+\infty} |\mathcal{V}(n)| t \delta_t$. In order to make it constant, we choose $\delta_t = \frac{1}{|\mathcal{V}(n)| t^4}$. For the first term, we use that $\forall s \in [\alpha t, t], \mathcal{E}(m_k(s), \delta_t) \leq \mathcal{E}(\alpha(1 - \alpha)t, \delta_t)$ and that n can be played only if the two confidence intervals overlap. Hence, we further obtain that

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{n_t = n\}\right] \leq \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{2\mathcal{E}(\alpha(1 - \alpha)t, \delta_t) \geq \Delta_n\}\right] + \sum_{t=1}^{+\infty} |\mathcal{V}(n)| t^2 \delta_{t - \lfloor \alpha t \rfloor}$$

$$= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{2\mathcal{E}(\alpha(1 - \alpha)t, \delta_t) \geq \Delta_n\}\right] + \sum_{t=1}^{+\infty} (1 - \alpha)^{-4} t^{-2}$$

$$= t_{\alpha,n} + \frac{\pi^2}{6(1 - \alpha)^4} \ ,$$

$$= t_{\alpha,n} + \frac{\pi^2}{6}\left(1 + \frac{1}{K}\right)^4 \ , \quad \text{since } \alpha = \frac{1}{K+1},$$

and for a deterministic constant $t_{\alpha,n}$ defined by

$$t_{\alpha,n} = \sup\{t \in \mathbb{N} : 2\mathcal{E}(\alpha(1 - \alpha)t, \delta_t) \geq \Delta_n\} \ .$$

We recall that $K = \lceil \log_{3/2}(N) \rceil$ is used to simplify the notation. The last step consists in upper bounding the value of $t_{\alpha,n}$ explicitly according to n and Δ_n. Considering some $t \geq 4 \vee n+1$, from Theorem 31 we know that there exist universal constants β and ξ, and we obtain that

$$
\begin{aligned}
\mathcal{E}(\alpha(1-\alpha)t, \delta_t) &\leq \beta \sqrt{\frac{\log\left(\frac{2\lceil n\sqrt{m_k(s)}\rceil}{\delta_t}\right)}{m_k(s)}} + n \times \xi \left(\frac{\log\left(\frac{2\lceil n\sqrt{m_k(s)}\rceil}{\delta_t}\right)}{m_k(s)}\right)^{\frac{2}{3}} \\
&\leq \beta \sqrt{\frac{\log\left(2t^4(n+1)\sqrt{t}|\mathcal{V}(n)|\right)}{\alpha(1-\alpha)t}} + n \times \xi \left(\frac{\log\left(2t^4(n+1)\sqrt{t}|\mathcal{V}(n)|\right)}{\alpha(1-\alpha)t}\right)^{\frac{2}{3}} \\
&\leq \beta \sqrt{\frac{\log\left(t^5(n+1)^2\right)}{\alpha(1-\alpha)t}} + n \times \xi \left(\frac{\log\left(t^5(n+1)^2\right)}{\alpha(1-\alpha)t}\right)^{\frac{2}{3}} \\
&\leq \beta \sqrt{\frac{7\log(t)}{\alpha(1-\alpha)t}} + n \times \xi \left(\frac{7\log(t)}{\alpha(1-\alpha)t}\right)^{\frac{2}{3}} .
\end{aligned}
$$

Since $\alpha = \frac{1}{K+1}$ it holds that $\alpha(1-\alpha) = \frac{1}{K+1}\frac{K}{K+1} \geq \frac{1}{2(K+1)}$. We then get that for some universal constants β' and ξ', it first holds that:

$$
\mathcal{E}(\alpha(1-\alpha)t, \delta_t) \leq \left(\beta'\sqrt{K+1} + n(K+1)^{\frac{2}{3}}\xi'\right)\sqrt{\frac{\log(t)}{t}} , \qquad (9.23)
$$

where we bounded $\left(\frac{\log(t)}{t}\right)^{\frac{2}{3}}$ by $\sqrt{\frac{\log(t)}{t}}$. Without this simplification, we also obtain that

$$
\mathcal{E}(\alpha(1-\alpha)t, \delta_t) \leq (K+1)^{\frac{2}{3}} \left\{\beta' + n\left(\frac{\log(t)}{t}\right)^{\frac{1}{6}}\xi'\right\}\sqrt{\frac{\log(t)}{t}} . \qquad (9.24)
$$

The different scaling proposed in the theorem then come from using Lemma 53 on (9.23) and (9.24), taking the minimum between the two (since both bounds are valid simultaneously), and for the latter splitting cases depending on $\frac{t}{\log(t)} \leq n^6$ being satisfied or not (taking this time the maximum between the two cases).

We provide the right-hand term of the result using (9.23), applying Lemma 53 with $\zeta = 1$ and

$$
x = \frac{4}{\Delta_n^2} \times \left(\beta'\sqrt{K+1} + n(K+1)^{\frac{2}{3}}\xi'\right)^2 ,
$$

which leads to $t_{\alpha,n} \leq 3x\log(x)$. This provides the term $\mathcal{O}\left(\frac{n^2}{\Delta_n^2}\right)$ of the result, and constants in the logarithmic terms can be recovered explicitly by recovering the values of β' and ξ'.

We then obtain the left-hand term of the result by considering (9.24). Lemma 53 first provides that $n\left(\frac{\log(t)}{t}\right)^{\frac{1}{6}} \leq 1$ for $t \geq 18n^6\log(n)$ (first bound). Still using

Lemma 53, this simplification permits to use $\zeta = 1$ and the threshold

$$y = \frac{4}{\Delta_n^2} \times \left(\beta' \sqrt{K+1} + (K+1)^{\frac{2}{3}} \xi' \right)^2 ,$$

and an upper bound of $t_{\alpha,n} \leq 3y \log(y)$, but only if this term is larger than $18n^6 \log(n)$. This provides the remaining terms of the bound, and again the logarithms can be easily recovered by computing β' and ξ'.

This concludes the proof for the problem-dependent in the favorable case where all arms are neighbors, remarking that these upper bounds just have to be multiplied by Δ_n and summed over $n \in [N]$ to convert into the regret bound. Furthermore, the problem-independent guarantee can be derived from taking the minimum between the bound and $\Delta_n T$, remarking that the worst case is $\Delta_n = T^{-1/2}$ if we omit the logarithms.

General case We now provide the regret bound for the general case, where at least some arms do not include $[N]$ in their neighborhood. We recall that two main ingredients of Local-Greedy are (1) that the arm n_t played in t is the best empirical arm in the neighborhood of $\mathcal{V}(n_{t-1})$, according to the simple estimates computed with samples from n_{t-1}, and (2) that $n_t = n_{t-1}$ if $m_{n_{t-1}}(t) < \alpha t$ (forced sampling). Hence, similarly to the previous proof we use that n_t is either pulled thanks to a "greedy play" or because of forced sampling. Furthermore, the two cases can be merged because forced sampling can only come after n_t being pulled because of a greedy play in the recent rounds. More precisely, we use that

$$\{n_t = n\} \subset \{\exists s \in [t - \lfloor \alpha t \rfloor] : n_s = n, m_s(\ell_s) \geq \lceil \alpha s \rceil\} . \tag{9.25}$$

This argument is at the core of our analysis, but before going further we need to introduce the notion of *locally optimal plays.*

Definition 12 (Locally optimal plays and optimal path)**.** *Given a reference arm ℓ, playing $n \in \mathcal{V}(\ell)$ is **locally optimal** if $n = \arg\max_{k \in \mathcal{V}(\ell)} r(k)$. In that case, n is the best neighbor of ℓ, and we use the notation $n = v^+(n)$.*

*Furthermore, a sequence of successive locally optimal plays is an **optimal path** towards $n^\star$. By construction of $\mathcal{V}$, an optimal path contains at most $K \coloneqq \mathcal{O}(\log(N))$ sub-optimal arms.*

The last fact presented in the definition is trivial: in the worst case the path start at one of the extremes of the interval $[N]$ and $n^\star$ is at the other. By design of $\mathcal{V}$ (Definition 11) we obtain that $n^\star$ is reached in $\lceil \log_{3/2}(N) \rceil$ steps at most. The rest of the proof is based on the idea that, when t is large enough, the algorithm starts following an optimal path with high probability, so a sub-optimal arm can be played only if it is located on an optimal path from another sub-optimal arm to $n^\star$. We formalize it with the following result.

Lemma 54 (Existence of a "recent" sub-optimal play). *For any time step $t \in [T]$ and arm $n \neq n^\star$, it holds that*

$$\{n_t = n\} \subset \mathcal{A}_t := \Big\{ \exists s \in \big[(1-\alpha)^K t, t\big], l \in [N], l' \in \mathcal{V}(l) : \widehat{r}_s(l) \leq \widehat{r}_s(l'),$$
$$m_s(l) \geq \alpha(1-\alpha)^K t \quad and \quad r(l) > r(l') \Big\} ,$$

Proof. Starting from (9.25), we first use that either $n \neq v^+(\ell_s)$, and in that case playing n is locally sub-optimal so this event belongs to $\mathcal{A}_t$, or $n = v^+(\ell_s)$. Let us now consider this second case: by definition of the leader, ℓ_s was played right before the sequence of forced plays of n started, which must have happened at least as recently as $t - \lfloor \alpha t \rfloor - 1$. From that point, the recursion pattern is clear: ℓ_s must have been selected in the last $\lfloor \alpha(s-1) \rfloor$, by either being a locally sub-optimal play or not. The first case is included in $\mathcal{A}_t$, why the second requires to add another step in the analysis. Furthermore, the arm used to estimate ℓ_s was itself samples at least proportionally to s. Using that this can happen K times, and that the worst number of steps to look into the past at each step is at most a fraction $(1-\alpha)$ of the number in the previous step, we finally obtain that there have been a sub-optimal greedy play in the last $\big[(1-\alpha)^K t, t\big]$ steps. $\square$

Before going further, we justify the tuning $\alpha = \frac{1}{K+1}$, by stating that it maximizes $\alpha(1-\alpha)^K$ (used later in the proof). At this step, we can simplify the notation by remarking that

$$(1-\alpha)^K = \left(\frac{K}{K+1} \right)^K \geq e^{-1} ,$$

hence we replace $(1-\alpha)^K$ by e^{-1} in the rest of the proof.

In words Lemma 54 states that, even if forced sampling slows down the ascension towards $n^\star$, since optimal paths contain at most K sub-optimal arm then $n^\star$ is relatively fast to reach from any arm in $[N]$. Next, we use this result in the regret analysis by considering its occurrences with the following event,

$$\mathcal{H}_t = \Big\{ \forall s \in \big[e^{-1}t, t\big], \forall n \in [N], \forall k \in \mathcal{V}(n) : m_k(s) \geq \frac{e^{-1}}{1+K}t,$$
$$|\widehat{r}_{k,s}(n) - r(n)| \leq \mathcal{E}(m_k(s), \delta_s) \Big\} .$$

Then, for any $t_K \in \mathbb{N}$ we can upper bound the number of plays of each sub-optimal arm $n \in [N]$ as follows,

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{n_t = n\} \right] \leq \mathbb{E}\left[\sum_{t \geq 1} \mathbb{1}\{\mathcal{A}_t\} \right]$$

$$\leq t_K + \mathbb{E}\left[\sum_{t \geq t_K} \mathbb{1}\{\mathcal{A}_t, \mathcal{H}_t\} \right] + \sum_{t \geq t_K} \mathbb{P}(\bar{\mathcal{H}}_t) ,$$

with the slight abuse of notation that $\mathcal{E}$ is now defined with the coalition size N and not the local value of n considered. The rest of the proof is analogous to the simple case, where all arms are in a single neighborhood. We first choose $\delta_t = \frac{1}{N^2 t^4}$, and obtain that $\sum_{t \geq t_K} \mathbb{P}(\bar{\mathcal{H}}_t) \leq \sum_{t=1}^{+\infty} \frac{1}{e^{-4} t^2} \leq \frac{\pi^2}{6 e^{-4}}$.

Next, we tune t_K large enough so that $\mathbb{E}\left[\sum_{t \geq t_K} \mathbb{1}\{\mathcal{A}_t, \mathcal{H}_t\}\right] = 0$. This can be done by choosing

$$ t_K = \sup\left\{ t \in \mathbb{N} : 2\mathcal{E}\left(\frac{e^{-1}}{1+K} t, \delta_t\right) \leq \Delta\right\} . $$

where $\Delta = \min_{n \in [N-1]}\{|r(n+1) - r(n)|\}$.

We then deduce the result by applying the exact same steps as for the upper bound of $t_{\alpha,n}$, by carefully replacing $\delta_t = \frac{1}{|\mathcal{V}(n)| t^4}$ by $\delta_t = \frac{1}{N^2 t^4}$, and $\alpha(1-\alpha)t$ by $\frac{e^{-1}}{1+K} t$. Lemma 53 then allows to easily obtain the desired scaling, and to make the upper bound explicit by substitution. Since K is logarithmic in N, it is clear that it only contributes to the bound only by logarithmic factors. $\qquad\square$

9.C.4   Proof of Theorem 33

Theorem 33 (Regret upper bound for Greedy-Grid). *Suppose that* GG *is tuned with confidence level* $\delta_t = \frac{1}{N^2 t^3}$, *and* $\alpha = 1/4$. *Then, for any* $T \in \mathbb{N}$ *it holds that*

$$ \mathcal{R}_T = \widetilde{\mathcal{O}}_N\left(\sum_{n \in \mathcal{B}^\star} \frac{1}{\Delta_n} + \sum_{n \in \mathcal{S}} \frac{\log(T)}{\Delta_n} \wedge \Delta_n\left(\frac{\mathbb{1}\{n < n^\star\}}{\Delta_{v_l(n^\star)}^2} + \frac{\mathbb{1}\{n > n^\star\}}{\Delta_{v_r(n^\star)}^2}\right)\right) . $$

Additionally, it holds that $\mathcal{R}_T = \widetilde{\mathcal{O}}\left(\sqrt{(K + |\mathcal{B}^\star|)T}\right)$, *for* $K = \lfloor \log_{3/2}(N) \rfloor$.

Proof. As the theorem suggests, we will use different arguments depending on the position of n with respect to the grid and the optimal bin $\mathcal{B}^\star$. Before that, we introduce the crucial result of this proof: for each time t large enough, thanks to the design of Greedy-Grid, all arms in optimal bin $\mathcal{B}^\star$ are estimated with a simple estimate computed with a *linear* number of samples in t.

Following the implementation of GG, at each time t and for each arm $n \in [N]$, a confidence interval $[\text{LCB}_t(n), \text{UCB}_t(n)]$ is computed so that $r(n) \in [\text{LCB}_t(n), \text{UCB}_t(n)]$ with probability at least $1 - \delta_t$. Similarly to the proof of Theorem 32, we consider a "good event" stating that all confidence intervals where valid on a given time range before t,

$$ \mathcal{G}_t = \left\{\forall s \in \left[\left\lfloor \frac{3t}{16}\right\rfloor, t\right], \forall n \in [N], \; r_t(n) \in [\text{LCB}_t(n), \text{UCB}_t(n)]\right\} . $$

It is clear that $\sum_{t=1}^{+\infty} \mathbb{P}(\bar{\mathcal{G}}_t) \leq \sum_{t=1}^{+\infty} N^2 t \delta_{\lceil \frac{3t}{16} \rceil}$, where the second union bound on N comes from considering the (random) identity of the arm whose samples are used to compute the interval. The following results proves that, under $\mathcal{G}_t$, there is at least one arm in the bin $\mathcal{B}^\star$ that was played a linear number of times in t.

Lemma 55 (Linear number of plays in $\mathcal{B}^\star$ under $\mathcal{G}_t$). *Under $\mathcal{G}_t$, there exists an arm $n \in \mathcal{B}^\star \cup \{v_\ell^{\mathcal{S}}(n^\star), v_r^{\mathcal{S}}(n^\star)\}$ satisfying $m_n(3t/4) \geq \frac{t}{4K} \wedge \frac{t}{8}$, where we call $K = |\mathcal{S}| = \lfloor \log_{3/2}(N) \rfloor$.*

Proof. $\mathcal{G}_t$ guarantees that any play during the interval $\left[\lfloor \frac{t}{4} \rfloor, t\right]$ (including those due to forced sampling), were decided with valid confidence intervals. Indeed, starting at $\frac{3t}{16}$ we are sure that all forced exploration launched before that time is completed in $t/4$. Furthermore, it is also direct from the design of the algorithms that, if $r_s(n) \in [\text{LCB}_s(n), \text{UCB}_s(n)]$ and Greedy-Grid is not forced to sample the previous arm it must hold that (1) it is playing the grid, or (2) it is playing an arm in $\mathcal{B}^\star$. Indeed, no arm from a sub-optimal bin would eliminate its best neighbor.

We consider two cases. First, if an arm $n \in \mathcal{B}^\star$ was played between rounds $\lfloor \frac{t}{4} \rfloor$ and $\lfloor \frac{t}{2} \rfloor$. In that case it has collected at least $\frac{t}{8}$ samples before t, thanks to forced sampling. In the alternative case, the grid was played between those these two rounds, which incurs $\frac{t}{4K}$ plays of $\arg\max_{s \in \mathcal{S}} r(s)$ since by $\mathcal{G}$, $\arg\max_{s \in \mathcal{S}} r(s)$ is not eliminated when the grid is played. Then, notice that $\arg\max_{s \in \mathcal{S}} r(s) \subset \{n^*, v_\ell^{\mathcal{S}}(n^\star), v_r^{\mathcal{S}}(n^\star)\}$. The result follows by combining the two cases.

$\square$

Without loss of generality, we assume in the following that $K \geq 2$ (if this is not the case, just replace K by $\max(K, 2)$). As a direct consequence of Lemma 55, using Theorem 31,under $\mathcal{G}_t$ there exists some constant β and ξ (coming from the bounds of the theorem multiplied by $2\sqrt{4} = 4$) such that the LCB of arm $n^\star$ satisfies

$$\forall s \in \left[\frac{3t}{4}, t\right] : \text{LCB}_s(n^\star) \geq r(n^\star) - \left\{ \beta \sqrt{K \frac{\log\left(\frac{2\lceil N\sqrt{t} \rceil}{\delta_t}\right)}{t}} + N \times \xi \left(K \frac{\log\left(\frac{2\lceil N\sqrt{t} \rceil}{\delta_t}\right)}{t}\right)^{\frac{2}{3}} \right\}.$$

$$(9.26)$$

To simplify the notation, we denote the right-hand term by $\mathcal{E}(t)$ in the rest of the proof. Using this result, we can now consider all the sub-cases presented in the theorem. We fix a sub-optimal arm n, and upper bound $\sum_{t=1}^T \mathbb{1}\{n_t = n, \mathcal{G}_t\}$ depending on the position of n with respect to the grid $\mathcal{S}$. Similarly to what we did in the proof of Theorem 32, we relate the pulls due to forced sampling to actual decisions by stating that, if $n_t = n$, then there exists a round s between $t - \lfloor t/4 \rfloor$ and t such that GG requested a pull of arm n from a "grid play" or a "greedy play".

Case 1: $n \in \mathcal{S}$. In that case if n is pulled then, since there is no forced sampling

for the arms of the grid it directly holds that

$$\mathtt{UCB}_t(n) \geq \max_{n' \in [N]} \mathtt{LCB}_t(n') \geq \mathtt{LCB}_t(n^\star) \geq r(n^\star) - \mathcal{E}(t) . \tag{9.27}$$

On the other hand, under $\mathcal{G}_t$ it holds that

$$\mathtt{UCB}_t(n) \leq r(n) + \underbrace{\left\{ \beta \sqrt{K \frac{\log\left(\frac{2\lceil N\sqrt{t}\rceil}{\delta_t}\right)}{m_n(t)}} + N \times \xi \left(K \frac{\log\left(\frac{2\lceil N\sqrt{t}\rceil}{\delta_t}\right)}{m_n(t)} \right)^{\frac{2}{3}} \right\}}_{\mathcal{E}'(t)},$$

so pulling arm n is possible only if $\mathcal{E}(t) + \mathcal{E}'(t, m_n(t)) \geq \Delta_n$, that we simplify to $2\mathcal{E}'(t, m_n(t)) \geq \Delta_n$ or $2\mathcal{E}(t) \geq \Delta_n$. Considering the second term leads to a constant problem-dependent bound, analogous to $t_{\alpha,n}$ in the proof of Theorem 32. Hence, we focus on the first term, that provide the $\log(T)$ bound.

This time, we don't know if $m_n(t)$ is large or not. This explains why we obtain a UCB-like ($\widetilde{\mathcal{O}}(\log(T))$) upper bound with this technique. We use that

$$t \leq T \Rightarrow \log\left(t\frac{2\lceil N\sqrt{t}\rceil}{\delta_t}\right) \leq \log\left(T\frac{2\lceil N\sqrt{T}\rceil}{\delta_T}\right) = \widetilde{\mathcal{O}}(\log(T)) .$$

Then, similarly to proof of Theorem 32, we mitigate the asymptotic scaling in N by noticing that if $m_n(t) = \Omega(N^6)$ then $\frac{N}{m_n(t)^{\frac{1}{6}}}$ simplifies. In that case, we obtain a sub-Gaussian confidence interval, and similarly to the analysis of UCB [11] we obtain that arm n is only pulled at most $\widetilde{\mathcal{O}}\left(\frac{\log(T)}{\Delta_n^2} \vee N^6\right)$ times with high probability. This is the first part of the result for $n \in \mathcal{S}$.

We then use another analysis to derive the constant problem-dependent bound. We remark that, when the confidence intervals are valid, the best arm between $v_\ell^{\mathcal{S}}(n^\star)$ and $v_r^{\mathcal{S}}(n^\star)$ can only be eliminated by an arm $i_t^\star \in \mathcal{B}^\star$. By design of the algorithm (exploiting the unimodality assumption), it furthermore holds that if $i_t^\star \in \mathcal{B}^\star$ and those two arms are eliminated then $\mathtt{GG}$ does not play on the grid. Hence, the constant bound in this case comes from upper bounding the time required for this event to happen under $\mathcal{G}_t$. If $v_\ell^{\mathcal{S}}(n^\star)$ is not eliminated it must hold that $\mathtt{UCB}_t(v_l^{\mathcal{S}}(n^\star)) \geq \mathtt{LCB}_t(n^\star)$ (if $n \leq n^\star$). Furthermore, Lemma 55 also guarantees that

$$\mathtt{UCB}_t(v_\ell(n^\star)) \leq r(v_\ell(n^\star)) + \mathcal{E}(t) ,$$

therefore the event that $v_\ell(n^\star)$ is not eliminated is only possible if $2\mathcal{E}(t) \geq \Delta_{v_\ell(n^\star)}^2$. We can then use Lemma 53, following the same as in the proof of Theorem 32 right after (9.23) and (9.24). When T is large enough, the derivation provides the scaling $\widetilde{\mathcal{O}}\left(\frac{1}{\Delta_{v_\ell^{\mathcal{S}}(n^\star)}^2}\right)$. We can then can follow the same steps for $v_r^{\mathcal{S}}(n^\star)$, and obtain

$\widetilde{\mathcal{O}}\left(\frac{1}{\Delta^2_{v^{\mathcal{S}}_r(n^\star)}}\right)$. Furthermore, it is clear by analogy with the proof of Theorem 32 that for small values of T the upper bounds have to be multiplied by a N^2 factor. Finally, we can remark that if $n < n^\star$ the first bound is used, while the second is used for $n > n^\star$. This concludes the derivation of the upper bound for $n \in \mathcal{S}$.

Case 2: $n \notin \mathcal{S}$, $\mathcal{B}(n) \neq \mathcal{B}^\star$. We prove that this case is actually impossible under the good event, which explains the surprising constant upper bound independent of any gap. Indeed, if $n \notin \mathcal{S}$ is played, then it must hold that its right and left neighbors in the grid are eliminated. Since $\mathcal{B}(n) \neq \mathcal{B}^\star$ then at least one of them has a reward at least as good as any arm in $\mathcal{B}(n)$. However, if playing arm n was possible under $\mathcal{G}_t$ it would hold that

$$
\exists \ell \in \mathcal{B}(n): \quad r(\ell) \geq \mathtt{LCB}_t(\ell) \\
> \max\{\mathtt{UCB}_t(v^{\mathcal{S}}_\ell(n)), \mathtt{UCB}_t(v^{\mathcal{S}}_r(n))\} \\
\geq \max\{r(v^{\mathcal{S}}_\ell(n)), r(v^{\mathcal{S}}_r(n))\} \geq r(\ell) ,
$$

which is a contradiction due to the strict inequality in the second line. Hence, the number of time such arm n is played is simply upper bounded by $\sum_{t=1}^{+\infty} \mathbb{P}(\bar{\mathcal{G}}_t)$, which is (by design) bounded by a universal constant.

Case 3: $n \notin \mathcal{S}$, $\mathcal{B}(n) = \mathcal{B}^\star$ We use that $n_t = n$ implies that $n_s = n$ *due to a greedy play* at some round $s \in [3t/4, t]$. Under $\mathcal{G}_t$, we can thus directly use (9.26), and obtain that if $2\mathcal{E}(t) \leq \Delta^2_n$ this event is not possible anymore. Using the same derivation as in the other cases (involving Lemma 53), we obtain the upper bound scaling in $\mathcal{O}\left(\frac{1}{\Delta^2_n}\right)$ for T large enough, and by $\mathcal{O}\left(\frac{N^2}{\Delta^2_n}\right)$ in general.

$\square$

9.C.5 Regret of Greedy-Grid adapted for non-unimodal rewards

In this section we develop the result presented in Section 9.3, regarding the adaptation of Greedy-Grid in the case when the reward function is no longer assumed to be unimodal. We recall that the adaptation consists in simplifying the definition of the set $\mathcal{C}_t$ in Algorithm 19 by

$$
\mathcal{C}_t = \{s \in \mathcal{S}, U_s \geq L_{i^*_t}\} .
$$

We call the resulting algorithm `GG-NU`, for *Greedy-Grid Non Unimodal*, to differentiate it from the original version of `GG` introduced in the paper. In the following, we formalize the upper bound of the regret of `GG-NU`, and discuss how this result is obtained by adapting the proof of Theorem 33 from the previous section.

Theorem 34 (Regret of `GG-NU`). *Suppose that* `GG` *is tuned with confidence level*

$\delta_t = \frac{1}{N^2 t^3}$, and $\alpha = 1/4$. Then, for any $T \in \mathbb{N}$ it holds that

$$\mathcal{R}_T = \widetilde{\mathcal{O}}_N \left(\sum_{n \in \mathcal{B}^\star} \frac{1}{\Delta_n} + \sum_{n \in \mathcal{S}} \frac{\log(T)}{\Delta_n} \right) .$$

Additionally, it holds that $\mathcal{R}_T = \widetilde{\mathcal{O}}\left(\sqrt{(K + |\mathcal{B}^\star|)T} \right)$, for $K = \lfloor \log_{3/2}(N) \rfloor$.

Proof. The proof follows the exact same steps as the proof of Theorem 33 presented in Section 9.C.4. To adapt the arguments to `GG-NU`, we first remark that it suffices to identify which part of the proof uses the definition of the set $\mathcal{C}_t$. We then find that this is the case when analyzing the *Case 1* in the proof, namely the regret caused by sub-optimal arms in the grid $|\mathcal{S}|$. More precisely, to obtain the bound of Theorem 33 for `GG` we provide two simultaneously valid upper bounds: a logarithmic ($\log(T)$) upper bound with a reasoning akin to the standard UCB analysis, and then a constant upper bound that carefully leverages the definition of $\mathcal{C}_t$. We easily verify that the steps for the logarithmic bound remain valid with the new definition of $\mathcal{C}_t$, while the second bound clearly does not hold. This completes the adaptation of Theorem 33 (for `GG`) into Theorem 34 (for `GG-NU`). □

9.D   Experiments

All the code for the experiments is written in Python. We use Matplotlib for plotting [57], Numpy [50] for numerical computing and Scipy [93] for scientific and technical computing. Scipy and Numpy are distributed under the BSD 3-Clause, and Matplotlib is distribution under BSD-style license. Theses licenses allow free use, modification, and distribution of the library. All the experiments were conducted on a single standard laptop, with an execution time shorter than 24 hours.

In a first simulation, we consider a coalition of size $N = 100$ and a competition of size $p = 4$. At each timestep t, the algorithm decide a number of bidders n_t to send to the auction and the values of all bidders (coalition and competition) $\mathbf{v} \in \{0, 1\}^{n_t + p}$ are sampled according to $\mathcal{B}(0.05)$. With probability $\frac{n_t}{n_t + p}$, the reward $\mathbf{v}_{(1)} - \mathbf{v}_{(2)}$ is received and $\mathbf{v}_{(1)}$ is observed. The (pseudo) regret at time t is then computed as the sum of reward obtained up to time t. The simulation above is repeated 20 times with random seeds and the mean value across seeds is reported as the expected regret $\mathcal{R}(t)$ in Figure 9.3. Error bars represent the first and the last decile.

In this simulation, the parameters are chosen to allow for having a significant number of players while keeping a gap Δ large enough (about 2×10^{-4}) to be able to observe logarithmic regrets for the baselines. `LG` practically outperforms other approaches by a large gap. The two algorithms that ignore the structure (`UCB` and `EXP3`) end up exhibiting a worse regret than `LG`, `OSUB` and `GG`, which is expected. However `GG` has a much higher regret than `LG` and only outperform `UCB` and `EXP3` for

horizons greater than 10^5, when it starts to eliminate points from the logarithmic grid $\mathcal{S}$. Indeed, due to the explicit use of concentration bounds in the algorithm, which multiplicative constants are not optimized for practical implementations, GG does not practically reach the constant regret regime in the horizon of these simulations.

To illustrate the practical performance of LG, we perform additional simulations which are identical to the first one except for the parameters N, p, and the distribution of value that are set according to Table 9.2. The results are plot in Figure 9.4 where it is shown that LG reaches the constant regret regime after only a couple hundreds or thousands time steps while the other algorithms are still in the transient linear regime.

Table 9.2: Configuration of additional experiments presented in Figure 9.4.

| Position	N	p	Value distribution	n^*	Δ
Top	5	2	$\mathcal{B}eta(a = 0.35, b = 0.63)$	3	5×10^{-5}
Middle	5	2	$Trunc.\ Exp(0, 1)$	3	4×10^{-4}
Bottom	20	4	$Trunc.\ Exp(0, 1)$	5	3×10^{-5}
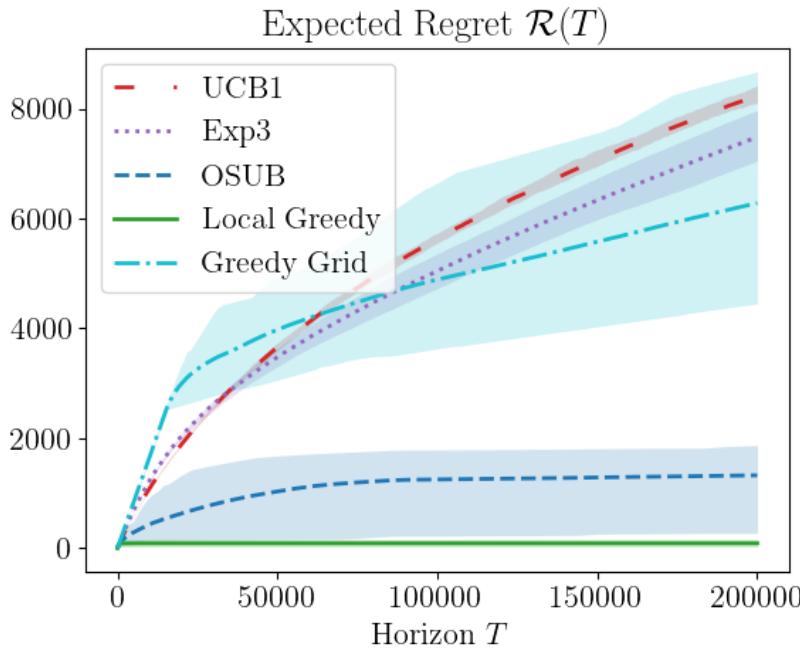
Figure 9.3: An empirical illustration of Table 9.1 with simulations in the following setting: values are distributed according to $\mathcal{B}(0.05)$, $N = 100$ and $p = 4$. We benchmark LG and GG (this paper), OSUB [31], UCB [11] and EXP3 [12] in terms of $\mathcal{R}(T)$ computed over 20 trajectories. Error bars represent the first and last decile.
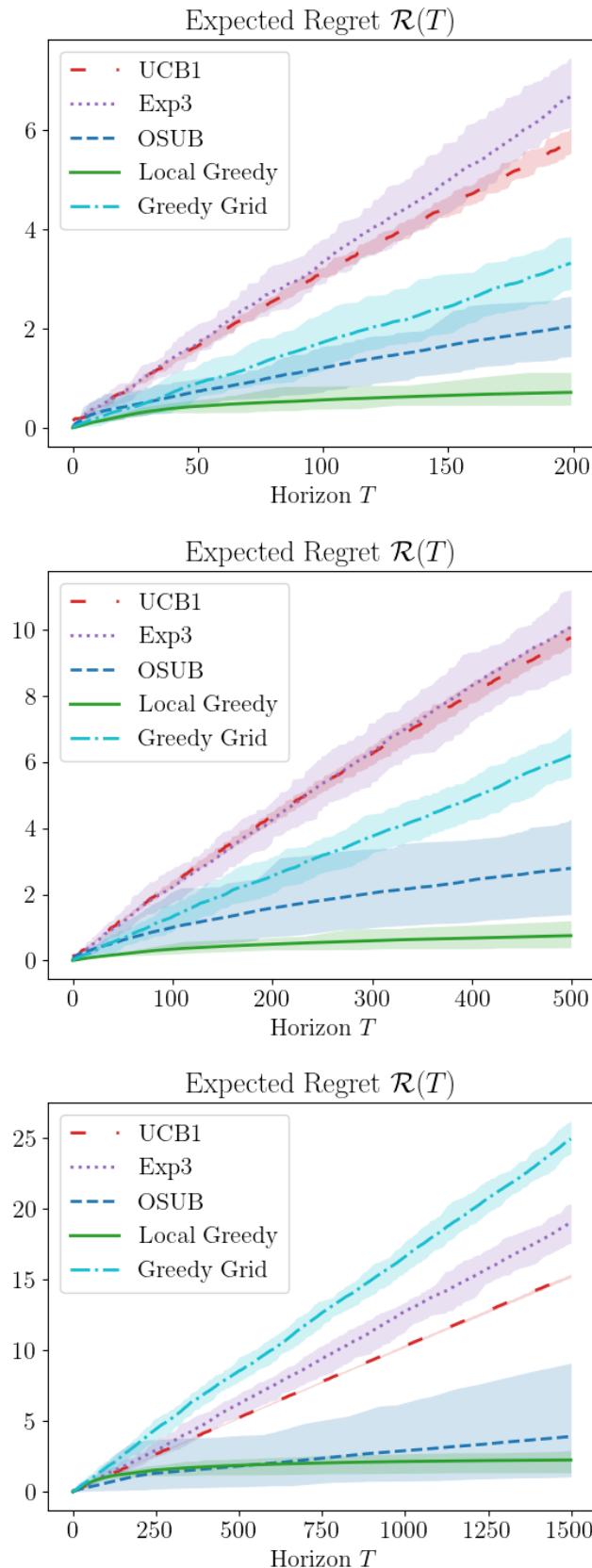
Figure 9.4: Illustration of additional experiments. Details of parameters are provided in Table 9.2.

9

10

General conclusion and
perspectives

This thesis explored three directions in online decision-making in algorithmic advertising. In the first part of this thesis, we looked at the online matching problem under dynamically evolving advertiser budgets. Starting from the traditional fixed-budget setting, we introduce a simple "refill" mechanism that periodically replenishes budgets and studied its impact on a Greedy matching algorithm in an Erdős–Rényi bipartite graph. Our analysis shows that, as the system scales the matching size is close from the solution of an associated system of ordinary differential equations, and under mild conditions the competitive ratio approaches 1. By studying the budget dynamics in this standard model, we gained clarity on how refills affect matching performance. A natural extension is to embed these dynamics into richer graph structures—stochastic block models, configuration models or geometric random graphs—to understand how topology and refill processes interact. Likewise, replacing our constant-rate Bernoulli refills with more realistic processes (nonuniform timing, revenue-dependent schedules, or bursty replenishments) would bring the model closer to real-world applications. From a technical standpoint, we proved only asymptotic stability of the ODE system (see Corollary 2); establishing exponential convergence remains an open challenge. Leveraging Lyapunov-based methods to strengthen our stability guarantees offers a promising direction for future work.

In the second part of this thesis, we shifted our focus from simplistic Erdős–Rényi graphs to more realistic community-structured networks by modeling user–advertiser interactions with a bipartite stochastic block model (SBM). We began by assuming that each ad campaign has a unit budget and that all SBM parameters are known. In this fully informed online setting, we showed that both the Myopic and

Balance policies admit fluid-limit characterizations: the matching size built by Myopic concentrates around the solution of an ordinary differential equation, while the matching size constructed by Balance policy converges to a solution of a differential inclusion due to a discontinuity induced by the policy. We then relaxed the knowledge of the parameters of the SBM assumption and frame the matching problem as a multi-armed bandit problem: affinities between user and campaign classes must be learned on the fly from binary match outcomes. Building on an Explore-Then-Commit algorithm combined with our Balance rule, we designed an algorithm that provably achieves sublinear regret. Natural extensions include reintroducing budget constraints—first as static per-campaign caps, then as dynamically evolving budgets under known SBM parameters—to assess how resource limits interact with community structure. In the learning regime, one can seek tighter regret bounds by replacing the simple ETC strategy with more sophisticated exploration policies (e.g. UCB or Thompson Sampling) tailored to the block-model context.

In the third part of the thesis, we turned our attention to a structured multi-armed bandit problem motivated by coalition bidding in repeated ad auctions. Here, the main challenge is to decide how many campaigns to send to an auction, to try to win the auction to secure online ad spots. We proposed two algorithms, Greedy-Grid and Local-Greedy , which leverage the problem's structural properties to achieve constant, time-independent regret. While Greedy-Grid delivers the strongest theoretical guarantee, Local-Greedy exhibits superior empirical performance and better constants in practice, by avoiding reliance on confidence intervals. This balance of theory and practice highlights the power of our new concentration tools, which allow us to leverage observations from "neighboring" arm groups with limited feedback. A natural extension, is to try to design a single policy that combines GG's optimal worst-case performances with LG's empirical efficiency—possibly through refined analysis or strategic integration of the two approaches. A broader direction is to extend this setting to multiple simultaneous auctions, where the decision-maker must allocate limited campaign resources across several markets. Key open questions include how to coordinate exploration across auctions, and handle heterogeneous value distributions.

Bibliography

[1] Anders Aamand, Justin Y Chen, and Piotr Indyk. (optimal) online bipartite matching with degree information. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[2] Yasin Abbasi-yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/e1d5be1c7f2f456670de3d53c7b54f4a-Paper.pdf.

[3] Emmanuel Abbe. Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018.

[4] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Proceedings of the 25th Annual Conference on Learning Theory*, 2012.

[5] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2008.

[6] Susanne Albers and Sebastian Schubert. Optimal Algorithms for Online b-Matching with Variable Vertex Capacities. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021)*, volume 207

of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[7] Susanne Albers and Sebastian Schubert. Tight bounds for online matching in bounded-degree graphs with vertex capacities. In *Embedded Systems and Applications*, 2022.

[8] Mohamed Habib Aliou Diallo Aoudi, Pascal Moyal, and Vincent Robin. Markovian online matching algorithms on large bipartite random graphs. *Methodology and Computing in Applied Probability*, 24(4):3195–3225, 2022. doi: 10.1007/s11009-022-09944-6.

[9] Mohamed Habib Aliou Diallo Aoudi, Pascal Moyal, and Vincent Robin. Large graph limits of local matching algorithms on configuration model graphs. *arXiv preprint arXiv:2410.18059*, 2024. arXiv:2410.18059.

[10] Jean-Pierre Aubin and Arrigo Cellina. *Differential Inclusions: Set-Valued Maps and Viability Theory*, volume 264 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, Berlin, 1984.

[11] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002. URL https://api.semanticscholar.org/CorpusID:207609497.

[12] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32 (1):48–77, 2002. doi: 10.1137/S0097539701398375. URL https://doi.org/10.1137/S0097539701398375.

[13] Moshe Babaioff, Yogeshwer Sharma, and Aleksandrs Slivkins. Characterizing truthful multi-armed bandit mechanisms: extended abstract. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, EC '09, page 79–88,

New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584584. doi: 10.1145/1566374.1566386. URL `https://doi.org/10.1145/1566374.1566386`.

[14] Dorian Baudry, Nadav Merlis, Mathieu Benjamin Molina, Hugo Richard, and Vianney Perchet. Multi-armed bandits with guaranteed revenue per arm. In *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, Proceedings of Machine Learning Research, 2024.

[15] Dorian Baudry, Hugo Richard, Maria Cherifa, Clément Calauzènes, and Vianney Perchet. Optimizing the coalition gain in online auctions with greedy structured bandits. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 19591–19635. Curran Associates, Inc., 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/22c799f287fd05e7174fd65a3ce134af-Paper-Conference.pdf`.

[16] Benjamin Birnbaum and Claire Mathieu. On-line bipartite matching made simple. *SIGACT News*, 39(1):80–87, mar 2008. ISSN 0163-5700.

[17] Béla Bollobás. *Random Graphs*. Cambridge University Press, 2001.

[18] Béla Bollobás and Graham Brightwell. The structure of random graph orders. *SIAM Journal on Discrete Mathematics*, 10(2):318–335, 1997. doi: 10.1137/S0895480194281215.

[19] Béla Bollobás. *Random Graphs*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2 edition, 2001.

[20] Charles Bordenave, Marc Lelarge, and Justin Salez. Matchings on infinite graphs. *Probability Theory and Related Fields*, 157(1–2):183–208, 2013. doi: 10.1007/s00440-012-0462-4.

[21] Stephen P. Borgatti and Martin G. Everett. Models of core/periphery structures. *Social Networks*, 21(4):375–395, 2000.

[22] Allan Borodin, Christodoulos Karavasilis, and Denis Pankratov. Greedy bipartite matching in random type poisson arrival model, 2018.

[23] Allan Borodin, Christodoulos Karavasilis, and Denis Pankratov. An experimental study of algorithms for online bipartite matching. *ACM J. Exp. Algorithmics*, 25, mar 2020. ISSN 1084-6654. doi: 10.1145/3379552.

[24] Allan Borodin, Calum MacRury, and Akash Rakheja. Bipartite stochastic matching: Online, random order, and i.i.d. models. 04 2020. doi: 10.48550/arXiv.2004.14304.

[25] Anna Brandenberger, Byron Chin, Nathan S. Sheffield, and Divya Shyamal. Matching Algorithms in the Sparse Stochastic Block Model. In Cécile Mailler and Sebastian Wild, editors, *35th International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms (AofA 2024)*, volume 302 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:21, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-329-4. doi: 10.4230/LIPIcs.AofA.2024.16.

[26] Brian Brubach, Karthik Abinav Sankararaman, Aravind Srinivasan, and Pan Xu. Online stochastic matching: New algorithms and bounds. *Algorithmica*, 82:2737 – 2783, 2016.

[27] Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, Gilles Stoltz, et al. Kullback–leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41(3):1516–1541, 2013.

[28] Priyanka Chatterjee and Itay Sharfi. Bidding and auction services in the privacy sandbox. `https://github.com/privacysandbox/`

`protected-auction-services-docs/blob/main/bidding_auction_`
`services_api.md`, 2024.

[29] Maria Cherifa, Clement Calauzenes, and Vianney Perchet. Dynamic online matching with budget refills. 05 2024. doi: 10.48550/arXiv.2405.09920.

[30] Maria Cherifa, Clément Calauzènes, and Vianney Perchet. Online matching on stochastic block model. June 2025. URL `https://hal.science/hal-05088588`. preprint.

[31] Richard Combes and Alexandre Proutière. Unimodal bandits: Regret lower bounds and optimal algorithms. *ArXiv*, abs/1405.5096, 2014. URL `https://api.semanticscholar.org/CorpusID:15210470`.

[32] H.A. David and H.N. Nagaraja. *Order Statistics*. Wiley Series in Probability and Statistics. Wiley, 2004. ISBN 9780471654018. URL `https://books.google.fr/books?id=bdhzFXg6xFkC`.

[33] Yuan Deng, Negin Golrezaei, Patrick Jaillet, Jason Cheuk Nam Liang, and Vahab S. Mirrokni. Multi-channel autobidding with budget and roi constraints. *ArXiv*, abs/2302.01523, 2023. URL `https://api.semanticscholar.org/CorpusID:256598278`.

[34] Nikhil R. Devanur and Sham M. Kakade. The price of truthfulness for pay-per-click auctions. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, EC '09, page 99–106, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584584. doi: 10.1145/1566374.1566388. URL `https://doi.org/10.1145/1566374.1566388`.

[35] Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized primal-dual analysis of ranking for online bipartite matching. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*,

SODA '13, page 101–107, USA, 2013. Society for Industrial and Applied Mathematics. ISBN 9781611972511.

[36] Rick Durrett. *Random Graph Dynamics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2006.

[37] Martin Dyer, Alan Frieze, and Boris Pittel. The average performance of the greedy matching algorithm. *The Annals of Applied Probability*, 3(2):526–552, 1993.

[38] Nathanael Enriquez, Gabriel Faraud, Laurent M'enard, and Nathan Noiry. Depth first exploration of a configuration model. *arXiv: Probability*, 2019.

[39] Eyal Even-Dar, Shie Mannor, Yishay Mansour, and Sridhar Mahadevan. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(6), 2006.

[40] Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 117–126, 2009. doi: 10.1109/FOCS. 2009.72.

[41] A. F. Filippov. *Differential Equations with Discontinuous Right-Hand Sides*, volume 18 of *Mathematics and Its Applications (Soviet Series)*. Kluwer Academic Publishers, Dordrecht, 1988. Translated from the Russian.

[42] Alan Frieze and Michał Karoński. *Introduction to Random Graphs*. Cambridge University Press, Cambridge, 2015. ISBN 9781107118508.

[43] Pierre Gaillard and Rémy Degenne. Stochastic bandit. 2019.

[44] David Gamarnik and David A. Goldberg. Randomized greedy algorithms for independent sets and matchings in regular graphs: Exact results and finite

girth corrections. *Combinatorics, Probability and Computing*, 19(1):61–85, 2010. doi: 10.1017/S0963548309990472.

[45] Nicolas Gast and Bruno Gaujal. Markov chains with discontinuous drifts have differential inclusions limits. Application to stochastic stability and mean field approximation. Research Report RR-7315, April 2011.

[46] Chris D. Godsil. Matchings and walks in graphs. *J. Graph Theory*, 5:285–297, 1981.

[47] Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, page 982–991, USA, 2008. Society for Industrial and Applied Mathematics.

[48] Edward F Grove, Ming-Yang Kao, P Krishnan, and Jeffrey Scott Vitter. Online perfect matching and mobile computing. In *Algorithms and Data Structures: 4th International Workshop, WADS'95 Kingston, Canada, August 16–18, 1995 Proceedings 4*, pages 194–205. Springer, 1995.

[49] Louisa Ha. Online advertising research in advertising journals: A review. *Journal of Current Issues and Research in Advertising*, 30, 05 2012. doi: 10.1080/10641734.2008.10505236.

[50] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585

(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

[51] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding*, pages 409–426, 1994.

[52] Remco van der Hofstad. *Random Graphs and Complex Networks*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2016.

[53] Paul W. Holland, Kathryn B. Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

[54] Junya Honda and Akimichi Takemura. Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards. *Journal of Machine Learning Research*, 16:3721–3756, 2015.

[55] Zhiyi Huang, Xinkai Shu, and Shuyi Yan. The power of multiple choices in online stochastic matching. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 91–103, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392648. doi: 10.1145/3519935.3520046.

[56] Zhiyi Huang, Zhihao Gavin Tang, and David Wajc. Online matching: A brief survey. *SIGecom Exch.*, 22(1):135–158, October 2024. doi: 10.1145/3699824.3699837.

[57] John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

[58] Patrick Jaillet and Xin Lu. Online stochastic matching: New algorithms with better bounds. *Mathematics of Operations Research*, 39(3):624–646, 2014. ISSN 0364765X, 15265471.

[59] Svante Janson, Tomasz Łuczak, and Andrzej Ruciński. *Random Graphs*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2011.

[60] M.C. Jones. The complementary beta distribution. *Journal of Statistical Planning and Inference*, 104(2):329–337, 2002. ISSN 0378-3758. doi: https://doi.org/10.1016/S0378-3758(01)00260-9. URL `https://www.sciencedirect.com/science/article/pii/S0378375801002609`.

[61] Bala Kalyanasundaram and Kirk Pruhs. Online weighted matching. *Journal of Algorithms*, 14(3):478–488, 1993.

[62] Bala Kalyanasundaram and Kirk R. Pruhs. An optimal deterministic algorithm for online b-matching. *Theoretical Computer Science*, 233(1):319–325, 2000.

[63] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC '90, page 352–358, New York, NY, USA, 1990. Association for Computing Machinery. ISBN 0897913612. doi: 10.1145/100216.100262.

[64] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC '90, page 352–358, New York, NY, USA, 1990. Association for Computing Machinery. ISBN 0897913612.

[65] Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

[66] Hassan K Khalil. *Nonlinear systems; 3rd ed.* Prentice-Hall, Upper Saddle River, NJ, 2002. The book can be consulted by contacting: PH-AID: Wallet, Lionel.

[67] Samir Khuller, Stephen G Mitchell, and Vijay V Vazirani. On-line algorithms for weighted bipartite matching and stable marriages. *Theoretical Computer Science*, 127(2):255–267, 1994.

[68] Minji Kim and Sewoong Oh. Stochastic blockmodel with cluster-dependent connection probabilities for modeling bipartite networks. *Journal of Machine Learning Research*, 21(234):1–43, 2020.

[69] Vijay Krishna. *Auction Theory*. Academic Press, 2009.

[70] Devadatta Kulkarni, Darrell Schmidt, and Sze-Kai Tsui. Eigenvalues of tridiagonal pseudo-toeplitz matrices. *Linear Algebra and its Applications*, 297(1): 63–80, 1999. ISSN 0024-3795. doi: https://doi.org/10.1016/S0024-3795(99) 00114-7.

[71] Markus Kunze. *Non-Smooth Dynamical Systems*, volume 1744 of *Lecture Notes in Mathematics*. Springer, Berlin, 2000. ISBN 978-3-540-67993-6. doi: 10. 1007/BFb0103843.

[72] Tor Lattimore and Csaba Szepesvari. *Bandit Algorithms*. 2019.

[73] Jonathan Levin. Auction theory. *Manuscript available at www. stanford. edu/jdlevin/Econ*, 20286, 2004.

[74] L. Lovász and M. D. Plummer. *Matching Theory*, volume 367. American Mathematical Society, 2009.

[75] Stefan Magureanu, Richard Combes, and Alexandre Proutiere. Lipschitz bandits: Regret lower bound and optimal algorithms. In Maria Florina Balcan, Vitaly Feldman, and Csaba Szepesvári, editors, *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pages 975–999, Barcelona, Spain, 13–15 Jun 2014.

[76] Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: An approach based on strongly factor-revealing lps. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 597–606, New York, NY, USA, 2011. Association for Computing Machinery.

[77] Vahideh H. Manshadi, Shayan Oveis Gharan, and Amin Saberi. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research*, 37(4):559–573, 2012. ISSN 0364765X, 15265471.

[78] P. Massart. The tight constant in the dvoretzky-kiefer-wolfowitz inequality. *The Annals of Probability*, 18(3):1269–1283, 1990. ISSN 00911798. URL `http://www.jstor.org/stable/2244426`.

[79] Andrew Mastin and Patrick Jaillet. Greedy online bipartite matching on random graphs. *ArXiv*, abs/1307.2536, 2013.

[80] Miller McPherson, Lynn Smith-Lovin, and James M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2001.

[81] Aranyak Mehta. Online matching and ad allocation. 8 (4):265–368, 2013.

[82] S. Muthukrishnan. Ad exchanges: Research issues. In *Workshop on Internet and Network Economics*, 2009. URL `https://api.semanticscholar.org/CorpusID:10046036`.

[83] Hamid Nazerzadeh, Amin Saberi, and Rakesh Vohra. Dynamic cost-per-action mechanisms and applications to online advertising. WWW '08, page 179–188, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605580852. doi: 10.1145/1367497.1367522. URL `https://doi.org/10.1145/1367497.1367522`.

[84] Thomas Nedelec, Clément Calauzènes, Noureddine El Karoui, and Vianney Perchet. 2022.

[85] Nathan Noiry, Vianney Perchet, and Flore Sentenac. Online matching in sparse random graphs: Non-asymptotic performances of greedy algorithm. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[86] Stefano Paladino, Francesco Trovò, Marcello Restelli, and Nicola Gatti. Unimodal thompson sampling for graph-structured arms. In Satinder Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. AAAI Press, 2017.

[87] Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951. doi: 10.1214/aoms/1177729586.

[88] Hassan Saber, Pierre M'enard, and Odalric-Ambrym Maillard. Forced-exploration free strategies for unimodal bandits. *ArXiv*, abs/2006.16569, 2020. URL https://api.semanticscholar.org/CorpusID:220265988.

[89] Amin S. Sayedi-Roshkhar. Real-time bidding in online display advertising. *Mark. Sci.*, 37:553–568, 2018. URL https://api.semanticscholar.org/CorpusID:52277027.

[90] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24. Springer Science & Business Media, 2003.

[91] Aleksandrs Slivkins. Introduction to multi-armed bandits. *Found. Trends Mach. Learn.*, 12(1–2):1–286, November 2019. ISSN 1935-8237. doi: 10.1561/2200000068.

[92] Nahuel Soprano-Loto, Matthieu Jonckheere, and Pascal Moyal. Online matching for the multiclass stochastic block model. *arXiv preprint arXiv:2303.15374*, 2023.

[93] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[94] Claude Viterbo. Systèmes dynamiques et équations différentielles, 2011.

[95] Lutz Warnke. On wormald's differential equation method. *ArXiv*, abs/1905.08928, 2019.

[96] Jonathan Weed, Vianney Perchet, and Philippe Rigollet. Online learning in repeated auctions. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1562–1583, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR. URL `https://proceedings.mlr.press/v49/weed16.html`.

[97] Nicholas C. Wormald. Differential equations for random processes and random graphs. *Annals of Applied Probability*, 5:1217–1235, 1995.

[98] Nicholas C. Wormald. The differential equation method for random graph processes and greedy algorithms. 1999.

[99] Yong Yuan, Juanjuan Li, and Rui Qin. A survey on real time bidding adver-
tising. *Proceedings of 2014 IEEE International Conference on Service Opera-
tions and Logistics, and Informatics, SOLI 2014*, pages 418–423, 11 2014. doi:
10.1109/SOLI.2014.6960761.

[100] Lenka Zdeborová and Marc Mézard. The number of matchings in random
graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(05):
P05003, may 2006. doi: 10.1088/1742-5468/2006/05/P05003.

[101] Lenka Zdeborová and Marc Mézard. The number of matchings in random
graphs. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(05):
P05003, may 2006. doi: 10.1088/1742-5468/2006/05/P05003.