université
PARIS-SACLAY

Sequential Resource Allocation for Network Diffusion Control

Thèse de doctorat de l'Université Paris-Saclay

Ecole Doctorale de Mathématique Hadamard (EDMH) n° 574
Spécialité de doctorat : Mathématiques aux interfaces
Unité de recherche : Centre Borelli (ENS Paris-Saclay), UMR 9010 CNRS
Référent : Ecole normale supérieure de Paris-Saclay

Thèse présentée et soutenue en visio-conférence totale à Fontenay-sous-bois, le 21 Janvier 2021, par

Mathilde FEKOM

Composition du jury :

Pierre-Yves Boëlle Professeur, Institut Pierre Louis d'Epidémiologie et de Santé Publique	Président
Theodoros Evgeniou Professeur, INSEAD	Examinateur
Nicole Immorlica Directeur de recherche, Microsoft Research	Rapportrice
Argyris Kalogeratos Chargé de recherche, ENS-Paris Saclay	Co-encadrant
Jean-Pierre Nadal Directeur de recherche, École Normale Supérieure	Rapporteur
Nicolas Vayatis PR, ENS Paris Saclay	Directeur de thèse
Elisabeta Vergu Directeur de recherche, INRA	Examinatrice

Thèse de doctorat

NNT: 2021UPASM008


école
normale
supérieure
paris—saclay


CENTRE
BORELLI


Fondation mathématique
FMJH
Jacques Hadamard

Abstract

The dynamic containment of an undesired network diffusion process, such as an epidemic, requires a decision maker (DM) to be able to respond to its evolution by taking the right control actions at the right moments. This task can be seen as managing the allocation of a limited amount of resources to the graph nodes, with the objective to reduce the effects of the process.

In this thesis we extend the *Dynamic Resource Allocation* (DRA) problem and propose a multi-round dynamic control framework, which we realize through two derived models: the *Restricted* and the *Sequential DRA* (RDRA, SDRA). Contrary to the standard full-information and full-access DRA considerations, the DM has limited information and access (altogether in the RDRA, sequentially in the SDRA) to a fraction of the nodes. The latter sequential aspect in the decision process offers a completely new perspective to the dynamic diffusion process control, making this work the first to cast the dynamic control problem as a series of specially designed *sequential selection processes*.

In the *Sequential Selection Problem* (SSP), *immediate* and *irrevocable* decisions need to be made by the DM as candidate items arrive randomly and get examined for one of the limited selection slots available. However, standard SSP variants, such as the very well-known *secretary problem*, begin with an empty selection set (*cold-start*) and perform the selection process once over a single candidate set (*single-round*). These two limitations are addressed in this thesis. First, we introduce the novel *Warm-starting SSP* setting that considers having at hand a *reference set*, which is a set of previously selected items, and tries to update optimally that set while examining the sequence of arriving candidates. The *Multi-round Sequential Selection Process*, the new online-within-online problem, is then introduced as a natural extension of the warm-starting selection.

Both *rank-based* and *score-based* objective functions over the final selection are considered. A cutoff-based approach is proposed for the former, while the optimal strategy based on dynamic thresholding is derived for the latter assuming that the score distribution is known. These strategies are then put in comparison for their efficiency in the traditional selection setting as well as in solving network control problems that motivated this thesis. The generality of the introduced models allow their application to a wide variety of fields and problems; for instance, reoccurring recruiting processes, management of resources (e.g. beds, staff) in healthcare units, as well as tackling difficult combinatorial problems under constrains, such as the *b-diversification* problem found in data-stream processing applications (e.g. in robotics or recommender systems).

Remerciements

Je tiens tout d'abord à remercier mes directeurs de thèse, Nicolas Vayatis et Argyris Kalogeratos, de m'avoir permis de faire une thèse au sein du Centre Borelli durant ces trois années, de m'avoir guidée et conseillée tout au long du parcours. Je pense en particulier à leur implication dans le formalisme de notre nouveau modèle, et aux nombreux centres d'intérêt scientifiques d'Argyris qui ont soulevé des questionnements intrigants et ont très souvent relancé ma motivation lorsque nécessaire.

Un grand merci aux membres du jury qui ont pris le temps de participer à ma soutenance, en particulier aux rapporteurs Jean-Pierre Nadal et Nicole Immorlica, dont les retours ont été largement bénéfiques à la qualité du manuscrit. Je pense aussi à leur enthousiasme et à leur bienveillance durant la présentation, ainsi qu'à celles d'Elisabeta Vergu, de Theodoros Evgeniou et de Pierre-Yves Boëlle, qui ont rendu l'expérience agréable et m'ont encouragée à poursuivre dans la recherche.

J'ai une petite pensée pour ceux avec qui j'ai collaboré ou échangé durant la thèse, entre autres Vianney Perchet, qui m'a aimablement accueillie aux group meetings du vendredi midi, pendant lesquels nous pouvions discuter de problématiques communes (et pas seulement culinaires). Collègues : Camille, Anton, Theos et Raphaël, c'était un plaisir de travailler avec vous sur ce projet commun qui, malgré la privation de sommeil, fut à la fois instructif et par moments, réellement amusant.

Je voudrais ensuite saluer mes camarades de labo, et plus particulièrement l'équipe de Cachan dont la fiabilité et la bonne entente ont toujours été vecteur de dynamisme. Nos débats du midi en salle ovale et nos pique-niques printaniers me manqueront ! Merci particulièrement à toi Alice, pour ton soutien et ta générosité (légendaire), et de manière plus générale de parvenir à me faire sentir toujours bien, entendue et comprise. Je voudrais aussi te remercier Matthieu, pour ton écoute (elle aussi légendaire), pour ton aide (administrative et bien plus), et globalement d'être véritablement une personne de confiance. Un grand merci à Firas pour ta bonne humeur, tes tentatives presque toujours réussies de blagues, à Myrto pour ta gentillesse et ta bienveillance, à Etienne d'être si drôle, à Dimitri d'avoir écouté mes délires sans jugement (Barbara), à Antoine, Théo, Pierre P., Pierre H., Batiste, Amir, Marie, Brian, Cédric, Charles, Julien et enfin à Asma, pour toutes les pauses, les rires, et les confidences. Je salue aussi celles et ceux qui ont partagé un bout de temps plus ou moins long avec moi durant cette période, notamment les post-docs, stagiaires, et les belles rencontres durant les conférences.

Il serait trop long d'écrire un paragraphe qui soit à l'image de la reconnaissance (et l'affection) que j'ai pour chaque membre de mon entourage proche, sans qui l'expérience aurait été bien plus pénible. Je me contenterai donc de dire que vous avez fait preuve d'une patience et d'un soutien infinis pendant ces 3 années, et m'avez montré (encore une fois) à quel point j'étais chanceuse. Un immense merci pour toutes les discussions, les encouragements, les fous rires, les cafés, les restau, les apéros, les sorties, et d'être tous aussi bienveillants, intéressants, fous, drôles, uniques.

Je souhaiterais tout de même finir par une mention spéciale pour ma famille et surtout pour mes parents, votre soutien inébranlable, vos encouragements et votre patience m'ont littéralement amenés jusqu'ici et la route n'a pas toujours été linéaire. Merci, vous êtes tout simplement les meilleurs du monde !

Contents

1 Introduction **1**

2 Related work **21**
2.1 Sequential Selection Problem 21
2.2 Epidemic control 22
2.3 Main SSP applications 23
2.3.1 Fields of application 23
2.3.2 Relevant algorithms from the literature 26

3 The Warm-starting Sequential Selection Problem **33**
3.1 Introduction 33
3.2 A general class of Sequential Selection Processes 34
3.2.1 Examples of well-known GSSPs 35
3.3 Introducing the warm-start 36
3.3.1 Description and rules of the game 36
3.3.2 Initial analysis 38
3.4 The multi-round extension 41
3.4.1 Principle 41
3.4.2 Sampling from the population 43
3.4.3 Estimating the quality 43
3.5 Conclusion 44

4 The Cutoff-based Cost Minimization algorithm **47**
4.1 Algorithm description 47
4.1.1 Cutoff-based strategy 47
4.1.2 Acceptance threshold 48
4.2 Optimal CCM 50
4.2.1 Optimal CCM for $q = 1/2$ 50
4.2.2 Optimal CCM for arbitrary q 54
4.2.3 Closed-form results for $b = 1$ 56
4.3 Adjusted policy: low failures-CCM 57
4.3.1 Algorithm description 57
4.3.2 Empirical study 59
4.4 Simulations in a multi-round setting 60
4.4.1 The impact of the cutoff value 60
4.4.2 Comparison with a few relevant algorithms from the literature 62
4.5 Conclusion 64
4.6 Appendix – Technical proofs 67

5 The Warm-starting Dynamic Thresholding algorithm **79**
5.1 Setting and rules of the game 79
5.2 Algorithm description 80
5.2.1 Threshold-based strategy 81

5.3 Full, partial, and no-information settings . . . 84
5.3.1 Some closed-form in full-information setting . . . 84
5.3.2 Rank-based strategy in no-information setting . . . 85
5.4 Simulations . . . 86
5.4.1 Simulations parameters . . . 86
5.4.2 Comparing score distributions . . . 86
5.4.3 Comparing information levels . . . 88
5.4.4 Large budget size . . . 89
5.5 Conclusion . . . 89
5.6 Appendix – Technical proofs . . . 89

6 Application to Epidemic Control **95**
6.1 Introduction . . . 95
6.2 Dynamic Resource Allocation strategies . . . 96
6.2.1 Network diffusion process . . . 96
6.2.2 Resource allocation for controlling a DP . . . 96
6.2.3 The dynamics of the SIS epidemic model with naïve/random allocation strategy . . . 98
6.2.4 Scoring function . . . 99
6.3 Sequential Dynamic Resource Allocation . . . 99
6.3.1 Restricted DRA . . . 99
6.3.2 Sequential DRA . . . 101
6.4 From DP control to a Multi-round Sequential Selection Process . . . 102
6.4.1 Link with the Sequential Selection Problem . . . 102
6.4.2 SSP algorithms for DP control . . . 104
6.5 Offline vs. Online . . . 105
6.6 Simulations . . . 107
6.6.1 Experimental setup . . . 107
6.6.2 Comparing online strategies . . . 109
6.6.3 RDRA with different scoring functions . . . 111
6.6.4 Sampling size . . . 113
6.6.5 Offline vs. online . . . 114
6.6.6 Allocation frequency . . . 115
6.7 Conclusion and discussion . . . 116

7 The stream-based *b*-diversification problem **121**
7.1 Introduction . . . 121
7.2 The MaxMin-diversification problem . . . 122
7.2.1 Description . . . 122
7.2.2 Mapping to a score-based Sequential Selection Problem . . . 123
7.3 Simulations . . . 125
7.3.1 Including a warm-start . . . 125
7.3.2 Simulations results . . . 127
7.4 Conclusion and discussion . . . 129

8 Conclusion **133**

Appendix A Personalised COVID-19 Isolation and Exit Policies **137**

Bibliography **155**

Chapter 1

Introduction (en français)

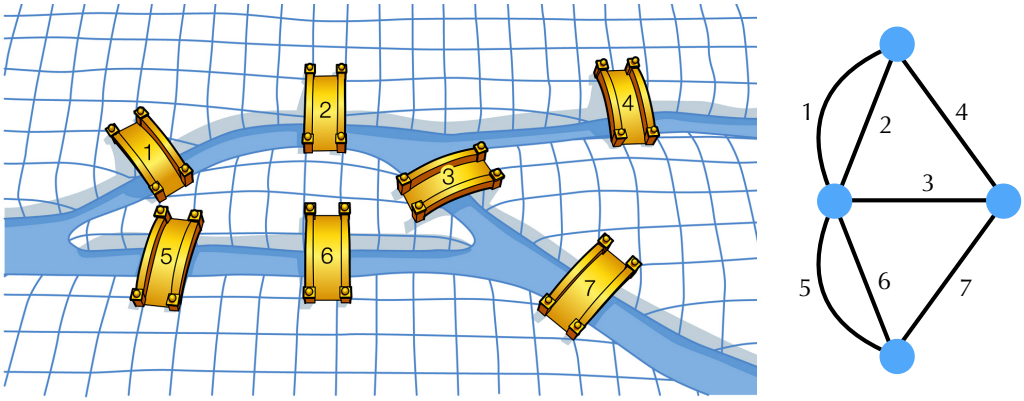


FIGURE 1.1: Représentation graphique (à gauche, ©Encyclopaedia Britannica, Inc.) et en graphe (à droite) du problème du pont de Könisberg. Est-il possible de traverser les sept ponts exactement une fois?

La première utilisation de graphe remonte à 1735, lorsque le mathématicien et médecin suisse Leonhard Euler a résolu le problème du pont de Königsberg. Ce puzzle mathématique, représenté sur la Fig. 1.1 met en scène sept ponts de chaque côté de deux îles de la vieille ville de Königsberg, en Prusse. Il s'agit de savoir s'il est possible de traverser la ville à pied de sorte que chaque pont soit traversé exactement une fois. Euler a montré qu'un tel chemin n'existait pas, et a ainsi obtenu le premier résultat de la théorie des graphes. En général, les graphiques s'avèrent plutôt utiles pour la modélisation de problèmes mathématiques, et sont devenus un outil largement utilisé et apprécié à cet égard au cours des dernières décennies. Pratiques pour simplifier les systèmes complexes et leurs interactions, les graphes sont des hôtes naturels aux processus de diffusion d'agent à agent, comme les épidémies [7] et la diffusion d'informations [62]. Bien que le comportement de ces processus constitue en soi un véritable domaine de recherche, un champ d'étude particulièrement intéressant concerne leur contrôle. Une *décideuse* (DM) est chargée de contrôler la propagation d'un processus de diffusion donné, au mieux de ses capacités et de sa puissance. Une poignée de solutions utiles et efficaces ont été apportées au fil des ans qui, cependant, donnent un pouvoir optimiste et parfois irréaliste au DM.

La principale ambition de cette thèse est de contribuer à rendre ces outils plus réalistes, et donc applicables à un plus large éventail de scénarios. Cela peut être tenté en assouplissant les hypothèses de départ faites sur les décisions de contrôle de la DM. Au cours de ce cheminement, nous avons rencontré des champs d'études

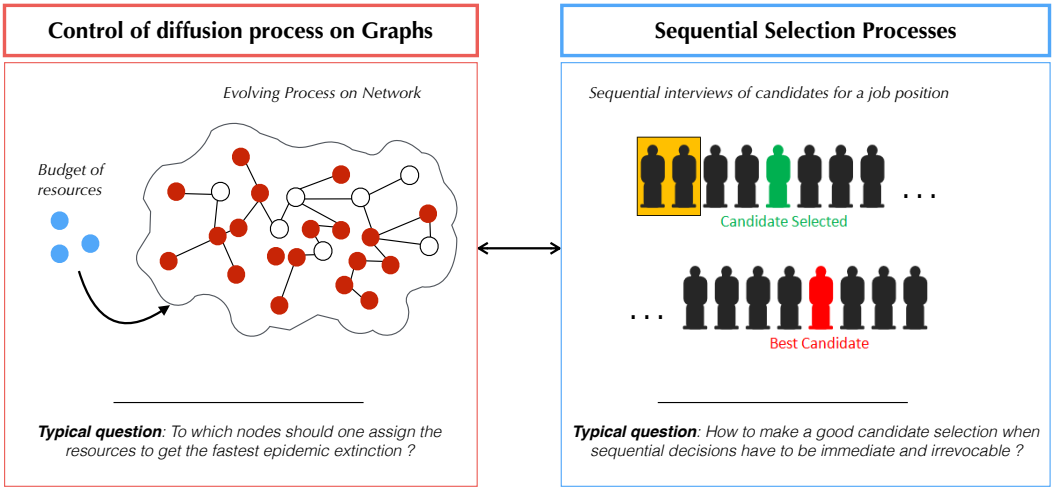


FIGURE 1.2: Schéma représentatif de l'idée clé de la thèse.

inexplorés dans des domaines de recherche très divers et inattendus, en particulier dans la prise de décision et la sélection en ligne, ce qui a permis d'élargir notre vision du problème initial. La Fig. 1.2 donne un premier aperçu de la connexion fondamentale établie entre les deux principaux domaines de cette thèse. D'un côté, il y a le problème du contrôle dynamique de processus de diffusion sur graphes et de l'autre, celui de la sélection séquentielle. En essayant de rapprocher ces deux mondes, nous avons découvert une lacune dans ce dernier, à savoir le manque d'outils nécessaires à l'utilisation de techniques de *Problème de Sélection Séquentielle* (SSP) dans le cadre du contrôle de diffusion. Nous avons essayé de combler ce manque en introduisant des concepts innovants qui ont ensuite été utilisés dans notre étude de cas principale. La sous-section suivante détaille l'exemple motivant avec une brève description de chaque domaine, tandis que la dernière sous-section de cette introduction résume les principales contributions de la thèse par chapitre.

Example de motivation

L'étude de cas originale qui nous intéresse met en scène une administratrice, ou décideuse, qui gère un petit budget de ressources en les allouant à un sous-groupe limité d'individus infectés dans le but de contrôler la propagation d'une épidémie non désirée. De manière plus significative, imaginez une unité de soins de santé dédiée au traitement des patients atteints d'une maladie contagieuse telle que la récente COVID-19 qui a eu un effet pandémique. L'unité ne dispose que de quelques lits disponibles pour couvrir les besoins des patients actuels et des personnes arrivant de manière aléatoire. Il est facile de voir que, dans ce scénario, la DM (un membre du personnel administratif, par exemple) décide quel(s) patient(s) doit/doivent recevoir un lit en priorité au fur et à mesure de leur arrivée et de leur départ, c'est-à-dire en temps réel. Il est évident que la DM ignore qui viendra plus tard et ne peut raisonnablement pas retirer un lit à un patient qui vient de lui être attribué. En général, l'aspect séquentiel de la stratégie de contrôle apparaît naturellement. C'est une conséquence de la nature stochastique de la demande sous-jacente de ressources, c'est-à-dire des arrivées séquentielles des individus demandant un traitement dans l'exemple précédent. Nous voulons donc un modèle qui permette l'allocation de

ressources en ligne, idéalement d'une manière qui interdise au DM d'avoir le pouvoir irréaliste de connaître le système ou d'agir sur lui.

Contrôle d'épidémies

Faisons un bref détour du côté de la théorie des graphes. Ici, les interactions entre agents sont généralement représentées par des arêtes (ou liens) qui relient des sommets (ou nœuds); l'ensemble forme ce que nous appelons un *graphe*, c'est-à-dire une structure mathématique qui modélise les relations par paires entre des entités (objets, agents, etc.). Selon la nature des interactions entre les nœuds, par exemple des contacts physiques, des amitiés virtuelles, un événement commun, etc., les arêtes peuvent être dirigées (influence dans un seul sens) ou non dirigées (influence dans les deux sens), pondérées (certaines interactions sont plus intenses que d'autres) ou non pondérées (toutes les interactions sont d'importance égale). Les nœuds eux-mêmes peuvent être dotés d'étiquettes qui font référence à leur appartenance à une classe. Un grand nombre d'extensions de la structure de base du graphe ont été développées et étudiées au fil du temps, c'est la raison pour laquelle elle constitue un outil si puissant pour modéliser des interactions de différentes natures. Par exemple, les agents peuvent très bien représenter des cellules du cerveau, des êtres humains, des gares, etc.

Les graphes sont également des hôtes propices aux processus de diffusion, c'est-à-dire aux phénomènes qui se propagent d'un nœud à un ou plusieurs de ses voisins. Là encore, la nature du processus de diffusion dicte ses caractéristiques; disons par exemple que nous voulons modéliser la propagation d'une rumeur, puisque l'agent n'est pas censé oublier l'information une fois qu'il l'a apprise (c'est-à-dire « désapprendre »), le processus de diffusion se décompose en trois états. Le premier lorsqu'il ignore la rumeur, le deuxième lorsqu'il l'apprend et peut la divulguer à ses voisins, et le troisième lorsqu'il la connaît mais la garde pour lui et ne participe donc pas à sa diffusion. Le modèle choisi pour ce scénario est similaire à celui utilisé pour modéliser la propagation d'une maladie infectieuse contre laquelle un individu infecté est définitivement immunisé une fois guéri, la rougeole par exemple. Il faut noter que d'autres maladies infectieuses, comme par exemple un rhume, peuvent ne nécessiter que deux états d'infection, car il n'y a pas de guérison perpétuelle. Ainsi, un nœud infecté peut passer de l'état infecté à l'état sain (avec des taux prédéfinis), et vice-versa.

Différentes actions et objectifs de contrôle peuvent être envisagés pour un processus de diffusion: maximiser la propagation, minimiser la propagation, trouver les « noeuds d'amorçage » avec lesquels commencer la diffusion, c'est-à-dire ceux qui, une fois infectés, permettent la diffusion la plus rapide, la plus large, ou inversement la plus lente ou la plus facile à contenir, etc.

Pour revenir à notre cas, le problème peut être énoncé en termes de réseau et de processus de diffusion comme suit: une DM, ou administratrice, est chargé d'allouer séquentiellement un nombre limité de ressources aux nœuds d'un réseau afin de réduire et/ou de contenir une propagation épidémique, sachant que ceux qui reçoivent les ressources bénéficient d'une probabilité de guérison accrue.

Récemment, la crise COVID-19 a mis en évidence certains problèmes de gestion quant à la manière de faire face à un tel processus de diffusion à l'échelle mondiale. En l'absence de vaccins, la solution la plus simple pour contrôler la propagation, qui

a en fait été adoptée par la majorité des pays touchés, a été d'imposer une réduction drastique des contacts au sein de la population avec un confinement obligatoire au niveau national. Il apparaît rapidement que la capacité en lits de réanimation de la plupart des pays serait largement dépassée si aucune mesure de confinement n'était prise. Dans un projet commun COVID-19 (Appendix A), nous proposons de diviser d'abord la population en deux catégories: les personnes qui auraient besoin d'un lit de réanimation en cas d'infection (moins d'1% de la population), et les autres, avant de quantifier l'impact de la qualité d'un tel classificateur sur les stratégies de sortie. Dans la même veine, nous avons étudié, dans un autre projet COVID- 19, la propagation de la maladie dans un hôpital disposant d'une unité dédiée COVID-19, et plus précisément comment la diffusion peut être contrôlée et contenue afin de minimiser le nombre d'unités, et donc d'individus, qui seraient touchés par la maladie. En effet, tout le personnel médical (médecins, infirmières, personnel administratif, etc.) qui a la possibilité de quitter l'hôpital, est un vecteur potentiel de virus pour le reste de la population.

Sélection séquentielle

Comme indiqué précédemment, la plupart des modèles d'allocation des ressources sur les graphe reposent sur le fait que la DM a un accès total aux nœuds du grpahe, à tout moment. Cependant, dans la réalité, il est parfois assez difficile (voire impossible) d'y parvenir. Malgré leur efficacité à cibler les nœuds pertinents auxquels assigner les ressources, ces modèles ne sont guère applicables à des cas réels, comme expliqué précédemment avec l'exemple de l'unité de soins de santé. Cette restriction nous a incité à chercher un moyen efficace de rendre ces modèles plus proches de situations réelles. Inspirés par un problème bien connu dans la communauté des SSPs, présenté comme le Problème de la Secrétaire en 1961 par Lindley [79], nous avons décidé de remédier au manque de réalisme des modèles de contrôle d'épidémies en concevant un modèle qui intègre un processus de décision séquentiel, permettant ainsi des actions de contrôle dynamiques.

Dans le problème initial des secrétaires, un nombre fixe et connu de candidats (ou secrétaires, à l'époque) sont interrogés les uns après les autres par une DM qui décide, immédiatement après l'entretien, soit d'engager le candidat, soit de le rejeter et de poursuivre les entretiens. Le processus s'achève lorsque le candidat est embauché. De nombreuses variantes de ce problème standard ont été développées au fil des ans [52], notamment une qui traite de plusieurs postes au lieu d'un seul, et une autre dont l'objectif est de minimiser la somme des rangs des candidats sélectionnés plutôt que d'essayer de maximiser la probabilité d'engager les meilleurs, comme c'était le cas dans la version originale. Notre idée initiale était de rechercher dans la littérature des SSPs le cadre le plus adapté au problème d'allocation de ressources décrit plus haut.

Nos recherches ont révélé deux différences notoires entre la configuration susmentionnée et celles pré-existantes. Premièrement, dans notre cas, l'allocation des ressources doit être dynamique, ce qui signifie qu'elle doit s'adapter aux changements de l'état infectieux des nœuds pour être aussi efficace que possible, alors que les SSPs classiques considèrent une seule séquence de candidats. Deuxièmement, lorsque l'allocation des ressources est remise en question, la DM peut avoir besoin de réaffecter les ressources qui sont déjà affectées à des nœuds à ce moment-là. En d'autres termes, la sélection initiale n'est pas vide, contrairement aux processus de

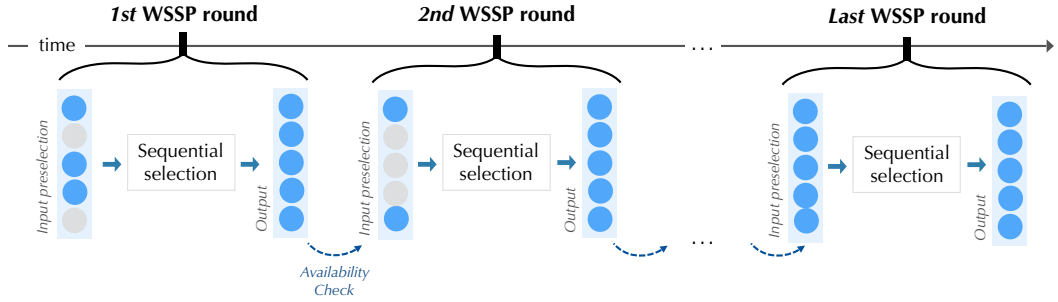


FIGURE 1.3: Concept clé d'un processus de sélection séquentielle à plusieurs tours (MSSP). Le résultat de chaque tour de sélection fournit la présélection d'entrée du tour suivant. Plus de détails dans Sec. 3.4.

sélection existants. Compte tenu de ces deux observations, nous avons décidé de formaliser un processus de sélection séquentielle qui tient compte des limites précédemment mentionnées et qui, par conséquent, s'applique naturellement à notre problème d'allocation dynamique de ressources contre la propagation d'une épidémie. Ce formalisme est présenté en détail au Chapitre 3.

Contribution par chapitre

Chapitre 2

L'état de l'art est organisé en trois sections. Les deux premières sections servent de revue de littérature approfondie pour les deux principaux domaines de cette thèse, à savoir les problèmes de sélection séquentielle et le contrôle d'épidémies. La dernière section commence par une liste non exhaustive des applications les plus courantes des SSPs, classées selon leur nature et les hypothèses qu'elles prennent en compte. Elle se poursuit par des descriptions détaillées de quelques algorithmes pertinents pour le SSP, classés par ordre chronologique.

Chapitre 3

L'objectif de ce chapitre est de planter le décor pour le reste du manuscrit. Il sert donc de toile de fond pour les chapitres qui suivent, et contient des explications détaillées du cadre construit au cours de ce doctorat. Les deux principales nouveautés en termes d'environnement de travail, c'est-à-dire le formalisme de démarrage à chaud et le formalisme de sélection séquentielle à tours multiples, y sont expliquées en détail. Un exemple de schéma de ces deux concepts clés est donné par la Fig. 1.6.

Le Chapitre 3 commence par une tentative d'unification des problèmes de sélection séquentielle avec une définition explicite, volontairement maintenue suffisamment large pour couvrir un large éventail de situations sans perdre en pertinence. Après la définition générique, l'étape suivante fut de se concentrer sur un cadre plus spécifique dans lequel la nécessité de commencer avec un ensemble de sélection vide est retirée. Elle partage également quelques hypothèses avec le cadre standard du problème des secrétaires, à savoir:

– il est purement basé sur le classement, c'est-à-dire sur des comparaisons d'éléments par paires et non sur des informations cardinales,

– les candidats à la sélection arrivent dans un ordre arbitraire et aléatoire,

– les décisions sont immédiates et irrévocables, et

– chaque poste doit être pourvu à la fin du processus.

La nouveauté de ce cadre nous a permis de fournir une première analyse réfléchie du mécanisme décrit, sous réserve d'hypothèses précises, justifiées en détail dans la section associée.

Le processus de sélection séquentielle à tours multiples (MSSP), est alors proposé comme une extension naturelle de la sélection de départ à chaud. Il consiste à répéter une seule sélection séquentielle, chaque fois avec des éléments différents à sélectionner, et son objectif global est de maintenir une bonne sélection dans le temps. Cette tâche est particulièrement difficile lorsque la valeur de chaque élément change d'un tour à l'autre d'une manière qui échappe au DM, et/ou lorsque, comme cela est décrit en détail avec quelques exemples dans ce chapitre, un nombre important d'éléments précédemment sélectionnés n'est plus disponible, ce qui ajoute apparemment une complication pour la DM qui doit les remplacer avant la fin de la séquence actuelle (tour).

Chapitre 4

Inspiré du problème standard de la secrétaire (SP), nous donnons dans le deuxième chapitre une première proposition algorithmique pour résoudre le problème de sélection séquentielle avec démarrage à chaud (WSSP), appelé Cutoff-based Cost Minimization (CCM). Comme dans [39], l'algorithme sépare la séquence entière en deux sous-séquences, la première est une phase d'apprentissage à partir de laquelle la DM recueille des informations précieuses sur l'ensemble de l'échantillon, bien qu'elle ne puisse faire aucune sélection avant le début de la deuxième phase. CCM est une version « démarrage à chaud » de l'algorithme standard du SP, qui utilise le temps d'apprentissage optimal (c'est-à-dire la valeur de coupure optimale) en fonction des principaux paramètres du problème. En effet, ayant à disposition un ensemble de référence d'une taille donnée et en supposant un algorithme basé sur la valeur de coupure, il n'est pas si aisé de déterminer si le temps d'apprentissage doit être plus long ou plus court. Supposons que la DM sache que ce dernier ensemble est « relativement bon », devrait-elle explorer davantage parce que le risque d'un mauvais résultat est faible? Ou, au contraire, devrait-elle explorer moins puisque l'ensemble fournit déjà une bonne « référence » au-dessus de laquelle des éléments pourraient être sélectionnés? La Sec. 4.2.2 fournit des réponses à la fois empiriques et analytiques à cette question, sous réserve d'hypothèses nécessaires.

Ce chapitre commence par une discussion sur l'algorithme que nous avons choisi, avant d'entrer dans les détails techniques de l'algorithme. À notre connaissance, à l'exception de l'algorithme standard SP, ou après avoir considéré des hypothèses spécifiques qui ne correspondent pas à notre configuration, aucun algorithme basé sur la coupure ne fournit de solution analytique pour la taille optimale de la phase d'apprentissage, et cela est dû à la complexité combinatoire du problème. Quelques papiers offrent une réponse empiriquement optimisée pour leur algorithme respectif, mais dans ce chapitre, nous fournissons une approximation des paramètres du problème. Plus précisément, nous approximons l'espérance des principaux paramètres du processus lors de l'utilisation de CCM, c'est-à-dire le seuil d'acceptation pour chaque élément, le nombre de sélections et la fonction de regret qui y est définie. Nous en

déduisons la valeur de coupure optimale compte tenu des principaux paramètres du problème, en commençant par le cas où la qualité de l'ensemble de sélection initiale est moyenne; nous proposons ensuite une *méthode de traduction* qui permet de déterminer la valeur de coupure optimale pour chaque valeur de cette qualité et met en évidence certains résultats intéressants. En outre, nous calculons le rapport concurrentiel de l'algorithme CCM et montrons qu'il se compare favorablement aux algorithmes existants, en particulier lorsque l'aspect de démarrage à chaud est plus important.

Une autre caractéristique intéressante du cadre proposé est que la DM est obligé de remplir tous les créneaux vides (ressources non affectées) avant la fin de la séquence, au lieu de devoir sélectionner au maximum un nombre donné d'éléments, ce qui est généralement le cas. Cela soulève une certaine interrogation sur la meilleure façon d'équilibrer la qualité des éléments sélectionnés et en même temps d'éviter d'accepter par défaut le(s) dernier(s) élément(s) de la séquence, quelle que soit leur qualité. À cette fin, nous proposons la variation lf-CCM (low failures-CCM) qui permet d'éviter ce phénomène en comparant la sélection de l'algorithme étape par étape à l'espérance à ce stade et de s'adapter en conséquence.

Enfin, le cadre du processus de sélection séquentielle à tours multiples introduit au Chapitre 3 (voir Sec. 3.4) est utilisé pour comparer les performances de différents algorithmes par le biais de simulations. Nous y montrons que le choix d'une valeur de coupure est très important dans ce cadre et doit donc être soigneusement ajusté.

Chapitre 5

Le Chapitre 5 propose un autre cadre pour le WSSP, dans lequel la DM peut observer les scores des éléments. Ces nouvelles informations modifient fondamentalement la stratégie optimale de sélection en ligne, qui doit maintenant être adaptée aux observations réelles faites. Afin de construire une stratégie de sélection optimale, une autre hypothèse est nécessaire, à savoir permettre à la DM d'apprendre la distribution des scores, ou même de la connaître à l'avance. Dans cette optique, la stratégie séquentielle optimale utilise le principe d'induction à rebours et la programmation dynamique pour générer un tableau des seuils d'acceptation optimaux, un pour chaque étape du processus.

La nouveauté ici est l'incorporation de l'aspect de démarrage à chaud dans la solution analytique présentée dans le chapitre. Nous avons également assoupli l'hypothèse selon laquelle la DM doit connaître la distribution en proposant la même méthode pour un contexte dans lequel l'information est partielle ou absente; une option particulièrement utile lorsque la distribution des scores évolue dans le temps, par exemple. Les performances des algorithmes sont, comme d'habitude, discutées par le biais de simulations multi-tours.

Chapitre 6

Le Chapitre 6 se concentre sur l'exemple de motivation discuté dans la section précédente, c'est-à-dire l'allocation séquentielle de ressources qui cherche à contenir un processus de diffusion tel qu'une épidémie. Ensuite, la discussion aborde les spécificités de cette première application avec des explications techniques sur l'environnement de travail: sur le graphe accueillant le processus de diffusion, le

type d'états infectieux considérés, et les actions et limites des stratégies de contrôle existantes. Dans une tentative de résolution analytique du problème, nous avons obtenu des équations qui régissent le processus stochastique à un niveau de contrôle grossier et nous avons réalisé que, même dans cette simple étude de cas, nous devons recourir à des approximations pour trouver une réponse sous forme finie.

La Sec. 6.3 est au cœur de cette thèse, elle présente l'idée conceptuelle et motivante à partir de laquelle le reste du manuscrit découle, dûment formalisée dans les chapitres précédents. En effet, l'hypothèse forte qui a été mise en place au cours de la dernière décennie, et par laquelle l'administratrice, qui alloue les ressources, a un accès total à la population, s'assouplit avec l'introduction de deux nouveaux modèles, rendant le problème plus réaliste et donc plus applicable à des cas réels (par exemple dans une unité de soins).

La section suivante se concentre sur la concrétisation du lien avec le SSP et met en œuvre l'algorithme proposé au Chapitre 4, ainsi que d'autres algorithmes adaptés au démarrage à chaud tirés de la littérature. Ensuite, la Sec. 6.5 étudie l'impact de la performance de toute stratégie de sélection séquentielle sur l'espérance du nombre d'individus infectés, au moyen d'une régression linéaire. Pour conclure, l'analyse de sensibilité menée sur différents types de graphes, de stratégies de contrôle et d'allocation, montre la robustesse de l'algorithme CCM proposé et la pertinence de ce nouveau modèle.

Chapitre 7

Ce dernier chapitre se concentre sur les deux défauts de l'application précédente, premièrement dû au fait qu'elle ne considère qu'une fonction objective additive, c'est-à-dire que la valeur de chaque élément est indépendante des autres éléments; et deuxièmement, qu'elle fonctionne sur des données purement générées pour les arrivées séquentielles.

Dans le Chapitre 7, l'accent est mis sur un exemple spécifique, le problème de la b-diversification, où le critère d'évaluation est non-additif, car la valeur de chaque élément dépend de ceux qui ont été sélectionnés auparavant. Cela soulève naturellement des interrogations algorithmiques, d'autant plus que les articles arrivent de manière séquentielle et ne peuvent être rappelés, ce qui exige une plus grande anticipation de la part de la DM. Plus précisément, dans le problème de la b-diversification, la sélection finale doit être aussi « diversifiée » que possible, c'est-à-dire que les éléments inclus doivent être éloignés les uns des autres, la notion de distance étant donnée par exemple par une distance euclidienne. Afin de remédier à la deuxième limitation, nous utilisons des données de environnementales réelles à grande dimension et comparons les performances de quelques algorithmes de l'état de l'art, décrits en détail au Chapter 2, en situations réelles.

Introduction (in english)

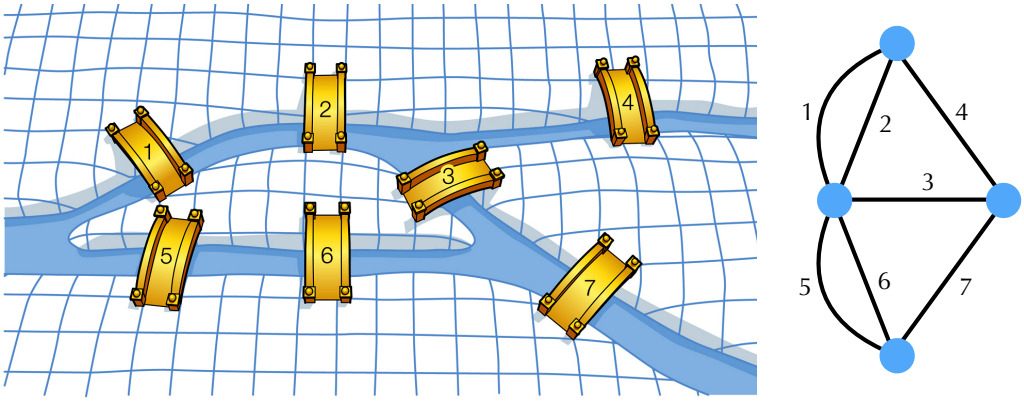


FIGURE 1.4: Graphical (left, ©Encyclopaedia Britannica, Inc.) and network (right) representations of the Könisberg bridge problem. Is it possible to cross all bridges exactly once?

The first use of graphs can be traced back to 1735, when the Swiss mathematician and physician Leonhard Euler solved the Königsberg bridge problem. This mathematical puzzle, depicted in Fig. 1.4, stages seven bridges on either sides of two islands of the old city of Königsberg, Prussia. The interrogation emerged of whether it was possible to walk through the town so that each bridge would be crossed exactly once. Euler showed that no such path existed, and by that he proved the first theorem in graph theory. In general, graphs turn out to be rather useful for mathematical problem modeling, and have grown to be a largely used and appreciated tool for that matter in the past decades. Helpful for simplifying complex systems and their interactions, graphs are natural hosts for agent-to-agent diffusion processes, such as epidemics [7] and information spread [62]. Although the behavior of those processes constitutes a proper research area by itself, a particularly interesting field of study concerns their control. A *decision maker (DM)* is in charge of controlling the propagation of a given diffusion process, to the best of her ability and power. A handful of useful and efficient solutions have been brought over the years that, however, give an optimistic and sometimes unrealistic power to the DM.

The main ambition of this thesis is to contribute towards making those tools more realistic, and therefore applicable to a wider range of scenarios. This can be attempted by relaxing the considered assumptions on the DM's control decisions. On this journey we encountered unexplored fields of study in quite diverse and unexpected areas of research, in particular in decision making and online selection, which in turn extended our vision over the original problem. A first glimpse at the prominent connection established between the two main fields of this PhD is displayed in Fig. 1.5. On one side there is the problem of dynamic control of diffusion processes in graphs and on the other that of sequential selection. By trying to bridge those two worlds, we discovered a shortcoming in the latter, namely a lack of available

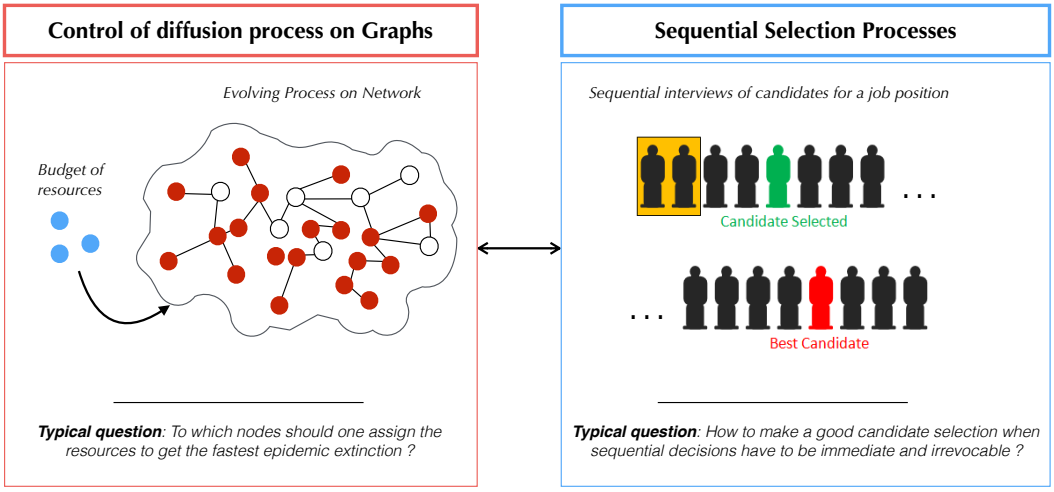


FIGURE 1.5: Representative scheme of the thesis key idea.

tools that are required for using *Sequential Selection Problem* (SSP) in the diffusion control setting, which we addressed with the introduction of innovative concepts that were then used in our main case study. The next subsection details the motivating example with a brief description of each field, while the last subsection of this introduction summarizes the main contributions per chapter of the thesis.

Motivating example

The original case study we are interested in features an administrator, or decision maker, who manages a small budget of resources by allocating them to a limited subgroup of infected individuals aiming to control the spread of an undesired epidemic. In a more meaningful way, picture a healthcare unit dedicated to treat patients of a spreading disease such as the recent COVID-19 that caused a pandemic effect. The unit only has a few available beds to cover the needs of both current patients and randomly incoming individuals. It is easy to see that, in this scenario, the DM (a member of the administrative staff, for instance) decides which patient(s) should receive a bed in priority as they come and go, i.e. in an online fashion. Obviously the DM ignores who will come later on and cannot reasonably withdraw a bed from a patient who was just assigned to it. Generally, the sequential aspect of the control strategy appears naturally. It is a consequence of the stochastic nature of the underlying demand for resource, i.e. the sequential arrivals of individuals asking for a treatment in the previous example. We therefore want a model that allows for online resource allocation, ideally in a way that forbids the DM from having unrealistic power to know the system or act on it.

Epidemic control

Let us make a brief detour through graph theory. There, interactions between agents are commonly represented by edges (or links) that connect vertices (or nodes); the whole forms what we call a *graph*, i.e. a mathematical structure that models pairwise relations between entities (objects, agents, etc.). According to the nature of the node

interactions, for instance physical contacts, virtual friendships, a common event, etc., edges can be directed (influence only in one way) or undirected (influence in both ways), weighted (some interactions are more intense than others) or unweighted (all interactions are of equal importance). The nodes themselves can be equipped with labels that refer to their membership to a class.

A large number of extensions of the basic network structure have been developed and studied over time, which is why it constitutes such a powerful tool to model interactions of various different natures. For instance, agents can very well represent brain cells, humain beings, railway stations, etc.

Graphs are also conducive hosts for a diffusion process, i.e. a phenomenon that propagates from a node to one or many of its neighbors. Here again, the diffusion process' nature dictates its characteristics; say for instance that we want to model the propagation of a rumor, since the agent is not supposed to forget the information once learned (i.e. to 'unlearn'), the diffusion process breaks down into three states. The first is when he is unaware of the rumor, the second st arts when he learns the rumor and may disclose it to its neighbors, and the third is when he knows the rumor but keeps it to himself and therefore does not participate in its diffusion. The chosen model for this scenario is similar to the one used to model the spread of an infectious disease from which an infected individual gets permanently immune once cured, e.g. measles. Observe that other infectious diseases, as for example a cold, might only need two infection states, as there is no perpetual cure. Thus, an infected node may go from the infected state to the healthy state (with pre-defined rates), and vice-versa.

Different control actions and objectives can be considered on a diffusion process: to maximize the spread, minimize the spread, find the 'seed nodes' to start the diffusion with, i.e. those that, once infected, permit the fastest diffusion, the widest, or reversely the slowest or easiest to contain, etc.

Going back to our case, the problem can be enunciated in terms of network and diffusion process as follow: a DM, or administrator, is in charge of sequentially allocating a limited number of resources to the nodes of a network in order to reduce and/or contain an epidemic spread, knowing that those who receive the resources enjoy an increased healing probability.

Recently, the COVID-19 crisis demonstrated some management issues on how to handle such a world-wide diffusion process. In the absence of vaccines, the easiest solution to control the spread that was in fact taken by the majority of the affected countries, was to enforce a drastic reduction of contacts among the population with a mandatory country-wise lock-down. It quickly appears that most countries' ICU bed capacity would be largely exceeded if no measure for containment was taken. In a shared COVID-19 project (Appendix A) we propose to first divide the population in two categories: the individuals that would require an ICU bed upon infection (less than 1% of the population), and the others, before quantifying the impact of the quality of such a classifier on the exit strategies. In the same vein, we investigated, in another COVID-19 project, the propagation of the disease in a hospital with a COVID-19 dedicated ward, and more precisely how the diffusion can be controlled and contained so as to minimize the number of wards, and hence of individuals, that would be affected by the disease. Indeed, all medical staff (doctors, nurses, administratives, etc.) who can leave the hospital, are potential virus vectors for the rest of the population.

Sequential selection

As mentioned before, most of the resource allocation models on graphs rely on the fact that the DM has total access to the graph nodes, at any point in time. However, in reality, this is sometimes rather difficult (even impossible). Despite their efficiency to target the relevant nodes to which the resources should be allocated, those models are therefore hardly implementable to real cases, as explained before with the example of the healthcare unit. This limitation encouraged us to look for an efficient way to make those models closer to real-life situations. Inspired by a well-known problem in the SSP community, introduced as the Secretary Problem in 1961 by Lindley [79], we decided to address the lack of realism of epidemic control models by designing a model combining a sequential decision process, and by that create the possibility for dynamic control actions.

In the original Secretary Problem, a fixed and known number of candidates (or secretaries, at that time) are interviewed one after the other by a DM who decides, immediately after the interview, either to hire the candidate or to reject him and pursue the interviews. The process ends when a candidate gets hired. Numerous variants of this standard problem have been developed over the years [52], in particular one that deals with multiple job positions instead of one, and another one in which the objective is to minimize the sum of the selected candidates ranks rather than trying to maximize the probability of hiring the best, as it was the case in the original version. Our initial idea was to look in the SSP literature for a setting most adapted to the resource allocation problem depicted earlier.

Our research revealed two notorious differences between the aforementioned configurations and the already existing settings. First, in our case the resource allocation must be dynamic, meaning that it should adapt to the changes in the infection state of the nodes to be as efficient as possible, while usual SSPs consider a single sequence of candidates. Second, when the resource allocation is questioned, the DM might need to reallocate even the resources that are already assigned to nodes by that time. In other words, the initial selection is not empty, contrary to existing selection processes. Bearing those two observations in mind, we decided to formalize a sequential selection process that addresses the aforementioned limitations and, consequently, applies naturally to our dynamic resource allocation problem, for cases such as controlling an epidemic spread. This formalism is presented in detail in Chapter 3.

Contribution per chapter

Chapter 2

The related work is organized in three sections. The two first sections serve as an extensive literature review for the two main fields of this thesis, i.e. Sequential Selection Problems and Epidemic control. The last section starts with a non-exhaustive list of the most common SSP applications, sorted by their nature and the assumptions they consider. It carries on with detailed descriptions of a few relevant SSP algorithms, listed chronologically.

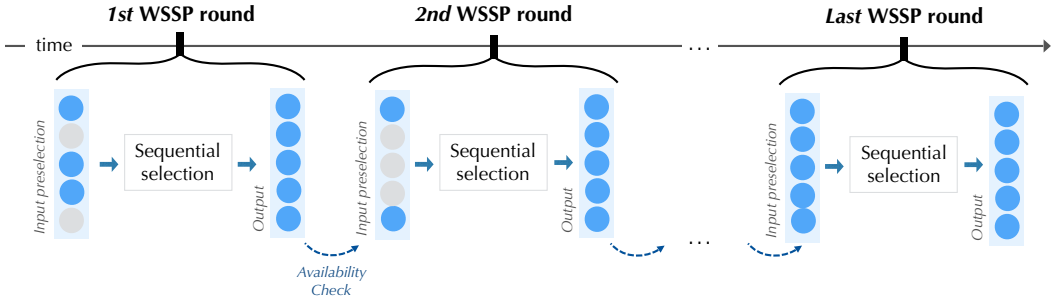


FIGURE 1.6: Key concept of a Multi-round Sequential Selection Process (MSSP). The output of each selection round provides the input preselection of the subsequent round. More details in Sec. 3.4.

Chapter 3

The objective of this chapter is to set the scene for the rest of the manuscript. Therefore it serves as a background for the chapters that follow, and contains detailed explanations of the framework built during this PhD. The main two novelties in terms of work environment, i.e. the warm-start and the multi-round sequential selection formalisms, are thoroughly explained there. An example scheme of those two key concepts is given in Fig. 1.6.

Chapter 3 starts with an attempt to unify *Sequential Selection Problems* (SSPs) with an explicit definition, voluntarily kept broad enough to cover a large range of settings without losing in relevance. Following the generic definition, the next step undertaken focused on a more specific setting where the necessity of the process to start with an empty selection set is withdrawn. It also shares a few assumptions with the standard Secretary Problem setting, namely:

– it is purely rank-based, i.e. based on pairwise item comparisons and not on cardinal information,

– candidates for selection arrive in arbitrary random order,

– decisions are immediate and irrevocable, and

– every job position should be filled at the end of the process.

The novelty of this setting empowered us to provide a thought-through initial analysis of the described mechanism, subject to precise hypotheses, justified in details in the associated section.

The Multi-round Sequential Selection Process (MSSP), i.e. the new online-within-online problem, is then proposed as a natural extension of the warm-starting selection. It iterates a single sequential selection, each time with different items to select from, and its overall goal is to keep a good selection set through time. This task is particularly difficult when the value of each item changes from a round to another in a way that is unknown to the DM, and/or when, as is described in detail with a few examples in this chapter, a significant number of previously selected items is no longer available, which apparently adds a complication for the DM who must replace them before the end of the current sequence (round).

Chapter 4

Inspired by the standard Secretary Problem (SP), in the second chapter we give a first algorithmic proposition for solving the Warm-start Sequential Selection Problem (WSSP), called *Cutoff-based Cost Minimization* (CCM). Similar to [39], it separates the entire sequence into two subsequences, the first is a learning phase from which the DM collects some valuable information about the whole sample, though unable to make any selection before the second phase begins. CCM is a warm-starting version of the standard SP algorithm, which uses the optimal learning time (i.e. the optimal cutoff value) according to the main parameters of the problem. In fact, having a reference set of a given size and assuming a cutoff-based algorithm, one may find it unclear whether the learning time should be longer or shorter. Say the DM knows that the latter set is 'relatively good', should she explore more because the risk of a bad outcome is low? Or, on the contrary, should she explore less since the set already provides a good 'reference' above which items could be selected? Sec. 4.2.2 provides both empirical and analytical answers to this question, under necessary assumptions.

This chapter starts with an argumentation on our chosen algorithm, before going into further algorithmic technical details. To the best of our knowledge, except for the standard SP algorithm, or after considering specific assumptions that do not match our configuration, no cutoff-based algorithm provides an analytical solution for the optimal size of the learning phase, and this is due to the combinatorial complexity of the problem. A few papers offer an empirically optimized answer for their respective algorithm, but in this chapter we provide an approximation of the parameters of the problem. More precisely, we approximate the expectation of the main parameters of the process when using CCM, i.e. the acceptance threshold for each item, the number of selections, and the regret function defined therein. From the latter, we infer the optimal cutoff given the main parameters of the problem, starting with the case where the quality of the initial selection set is average; thereafter we propose a */translation method* that permits to derive the optimal cutoff for every value of this quality and highlights some interesting results. In addition, we compute the CCM algorithm's competitive ratio and show that it compares favorably to existing algorithms, especially when the warm-starting aspect is more important.

Another interesting characteristic of the proposed setting is that the DM is forced to fill all empty slots (unassigned resources) before the end of the sequence, contrary to having to select *at most* a given number of items, which is usually the case. This raises some interrogation on how to best balance competitive selected items and at the same time avoid accepting by default the last item(s) of the sequence regardless their quality. To this end, we propose the *low failures-CCM* (lf-CCM) variation that prevents this phenomenon from occurring by comparing the algorithm step-wise selection to the expectation at that stage and adapt accordingly.

Finally, the Multi-round Sequential Selection Process setting introduced in Chapter 3 (see Sec. 3.4) is used for performance comparison through simulations. There, we show that the choice of a cutoff value matters a lot in a multi-round setting, and should thereby be carefully tuned.

Chapter 5

Chapter 5 proposes an alternative WSSP setting, in which the DM can observe the items' scores. This new information fundamentally changes the optimal online selection strategy, that should now be adapted to the actual real-valued observations made. In order to construct an optimal selection strategy, another assumption is necessary, namely to allow the DM to learn the score distribution, or even to know it beforehand. This in mind, the optimal sequential strategy uses backward induction and dynamic programming to generate a table of optimal acceptance thresholds, one for each step of the process.

The novelty here is the incorporation of the warm-starting aspect in the analytical solution presented in the chapter. We also relaxed the assumption which requires the DM to know the distribution by proposing the same method for both a partial and a no-information settings; a particularly useful option when the score distribution is changing through time, for instance. The algorithms performances are, as usual, discussed through multi-round simulations.

Chapter 6

Chapter 6 concentrates on the motivating example discussed in the previous section, i.e. the sequential allocation of resources that aims at containing a diffusion process such as an epidemic. Then, the discussion addresses the specifics of this first application with technical explanations on the work environment: on the graph hosting the diffusion process, the type of infection states considered, and the actions and limitations of existing control strategies. In an attempt to solve the problem analytically, we derived equations that rule the stochastic process at a coarse-grained level of control and realized that, even in this simple case study, we must resort to approximations to find a closed-form answer.

Sec. 6.3 is at the heart of this thesis, it presents the conceptual and motivating idea from which the rest of the manuscript arrows from, properly formalized by the previous chapters. Indeed, the strong assumption that has been in place in the last decade, and by which the administrator, who allocates resources, has total access to the population gets relaxed with the introduction of two new models, rendering the problem more realistic and thus more applicable to real cases (e.g. in a healthcare unit).

The next section concentrates on making the link with SSP more concrete, and implements the algorithm proposed in Chapter 4, along with other warm-start-adapted algorithms from the literature. Then, Sec. 6.5 studies the impact of the performance of any sequential selection strategy on the expected number of infected individuals, by means of linear regression. To conclude, the conducted sensitivity analysis on various types of graphs, control and allocation strategies, shows the robustness of the proposed CCM algorithm and relevancy of this new model.

Chapter 7

This last chapter focuses on the two shortcomings of the previous application, firstly with the fact that it solely considers a modular objective function, meaning where

each item value is independent from the other items; and secondly, that it works on purely generated data for the sequential arrivals.

In Chapter 7 an emphasis is given on a specific example, the b-diversification problem, where the evaluation criterion is non-modular, as every item's value depends on those selected before. This naturally raises algorithmic interrogations, especially as items arrive sequentially and cannot be called back, requiring more anticipation from the DM. More specifically, in the diversification problem, the final selection should be as 'diverse' as possible, i.e. the included items need to be far away from each other, where the notion of distance is given by, for instance, an euclidian distance. In order to address the second limitation, we play with real high-dimensional environmental data and compare the performance of some state-of-the-art algorithms, thoroughly described in Chapter 2, in real-life situations.

Chapter 2

Related work

2.1 Sequential Selection Problem

Various extensions of the basic Secretary Problem have been investigated; for non-exhaustive surveys see [9, 52, 47]. While in the orignal setting decisions are immediate and irrevocable, a few papers study the possibility for recall [90, 27], where past solicitations are allowed, yet successful or unsuccessful. Importantly, a change in the setting or in the objective function, changes also the optimal cutoff. As mentioned before, in the ordinal model, the DM has information solely on to the total order of the items seen so far. In addition to its robustness, this model allows for elegant algorithmic solutions, see for instance [39, 67, 8, 5]. In some scenarios, the DM can not only compute the *relative rank* of an interviewed candidate among those examined earlier, but also assess candidate's true *quality score*. This value, sometimes referred as score, can be thought of as a random variable associated with each candidate. In [13], candidates are drawn from a uniform distribution on $[0,1]$ but the DM can only rank candidates relatively to those she has seen before, i.e. scores are unobserved, and the objective is to maximize the expectation of the score of the selected candidate. They have shown that in this case, the optimal cutoff becomes $c^* = \sqrt{n} - 1$ where the square-root stands for either the ceil or the floor. On the other end, *Robbin's problem* [23] seeks to minimize the expectation of the rank of the selected candidate (note: low ranks are better). As of yet, no optimal solution has been found for this particular problem, except for extreme cases where the number of candidates is less or equal to $n = 4$ [31]. In the Gusein-Zade problem [56, 50, 91], the DM's objective is to select one of the r-best out of the sequence, instead of selecting only the best as in the original version.

Notable variants are those related to *multiple stopping*, or simply b-choice, where the DM has to select b candidates [67, 12, 76, 8, 21, 87]. In that case, the objective set function can be modular (i.e. equivalent to adding up the independent application of the function to the elements of the set), submodular [12, 44], or subject to matroid constraints [45]. Non-modularity introduces interesting set evaluation aspects, such as the complementarity or mutual-enhancement among the selected candidates. Moreover, [2] maps the secretary problem to a high-dimension online linear programming problem and proposes a near-optimal algorithm to solve it. Regarding modular objective functions, [8] studies the b-choice problem with the objective to maximize the sum of the scores of the selected candidates, when candidates arrive in a random order, without assuming prior knowledge of the score distribution. An interesting finding is that, when n tends to infinity, the optimal cutoff for this setting remains $c^* = \lfloor n/e \rfloor$.

In addition to the above, some selection strategies proposed are not cutoff-based. An experimental comparison of simple and intuitive heuristics is provided in [98]. Furthermore, the *Bruss' odds theorem* [22] introduces an algorithm where a candidate is said to be *interesting* when he is the best seen so far. In order to stop the process and select a currently interviewed interesting candidate, the DM needs to estimate the probability that he is the last interesting candidate, i.e. none better will appear later. The strategy adapts accordingly during the selection process.

Besides, a few papers study the human capacity to learn the right cutoff after reviewing multiple independent candidate sets [54, 14], while in [101] two distinct aims are considered: to maximize the probability of selecting the best, or to maximize the expected score of the selected candidate. The selection process is repeated and the DM tries to learn the candidates' score distribution, while outputting an independent selection after each round. That work concludes by stating that learning the score distribution does contribute to the efficiency of the selection only w.r.t. the second aim when considering an underlying uniform score distribution. In [87] a table is precomputed containing the expected 'reward' for different number of steps and different number of jobs. During the whole selection process, the DM refers to that table to take decisions.

A rather different scenario concerns a startup company (or a new ambitious business unit) which is initially funded by a handful of people but is about to grow larger. The so-called *hiring problem* [21] refers to the SSP that aims at driving the optimal growth of personnel using an adaptive selection threshold based on the already employed items. Among heuristics, such as hiring above the worst or the best current referents, hiring above the mean referent score shown to be the best performing strategy. Similar settings where a set of selected candidates increases through time are considered in [75, 58, 48], while [58] makes a thorough analysis of *hiring above the m-th best* strategies.

Another interesting setting allows for contracts to have a fixed duration, thus to be temporary, see [88], and is introduced as the *temp SP* in [48]. The improved algorithm presented in [66] generalizes towards general packing constraints and arbitrary hiring durations.

2.2 Epidemic control

Compartmental models have been extensively studied in epidemiology since early last century. In recent years, they have gained much wider attention due to their usually simple analytic formulations that can model modern problems related to information diffusion and social epidemics, e.g. rumor spreading [62] and other social contagions [59, 63]. Being able to control efficiently undesired diffusion processes (DPs) is very crucial for public health and security. Yet, it is a difficult problem that in fact gets instantly much more complicated the moment one starts including more realistic constraints or objectives. This explains why most studies of the literature, despite providing high-level insights about the phenomena, remain rather far from being applicable in practice. A source of limitations is the theoretical *interaction model* one considers, along with its network-wise abstraction level (e.g. macro- vs microscopic modeling), which may be over-simplistic for the analyzed phenomenon. Another source of shortcomings is the requirement for having information regarding the *system state*, such as the infection state of nodes or the network connectivity.

Finally, limitations come from the way a control model assumes it can intervene to the DP, in a static or dynamic fashion to the evolution of the process.

Dynamic models for allocating medical resources are subject to wide investigation [25, 80], for which [96] gives a convenient formalism with the introduction of the *Dynamic Resource Allocation* (DRA), a model for network control, originally developed for SIS-like processes [86] (a node is either infected, or healthy without permanent immunity). It distributes a limited budget of available treatment resources on infected nodes in order to speed-up their recovery. The term *dynamic* as opposed to *static* such as vaccinations [92], comes from the adaptive nature of the strategy; it is not fixed in advance but evolves according to the infection state and possibly the structural changes of the network. Since it is reasonable to assume that authorities take real-time actions, we essentially focus on DRA strategies. Quite different dynamic resource allocation strategies have been studied so far. One example is the use of contact tracing that targets individuals who may have been in contact with other infected individuals [18, 40], and is particularly effective when only a few infected nodes remain.

Another example of DRA are the so-called *score-based* strategies that introduce an elegant way, through a simple score value, of assessing the criticality of each node individually for the containment of the DP. Then, the administrator (or DM) only has to ensure that at each moment the resources will be spent on the infected nodes with the highest scores. Among the proposed options, a simple yet efficient local score is the *Largest Reduction in Infectious Edges* (LRIE) [96], which depends on the infection state of the neighbors, hence it needs to be updated regularly during the process. A second option, called *priority-planning* [97], computes offline a *priority-order* of the network nodes, by considering the *max-cut* of a given list. This order provides a fixed global score for all nodes, which can then be used by the administrator to perform DRA. Note that the principle behind the LRIE has been generalized to SIS-like models with competition [63]. In [82] scores are continuous (called *control signals*) and are derived for each node with the purpose of minimizing a loss function represented by a tradeoff between the cost of a treatment and the cost incurred by infected nodes. The term *dynamic* as opposed to *static* comes from the adaptive nature of the strategy; it is not fixed in advance but adapts to the infection state and possibly the structural changes of the network. Since it is reasonable to assume that authorities take real-time actions, we essentially focus on DRA strategies in this thesis.

2.3 Main SSP applications

2.3.1 Fields of application

Applications expand from merely conceptual settings to concrete case-studies, usually related to robotics, sometimes in biology or social contexts. We build the following non-exhaustive list by first differentiating single selection ($b=1$) to multiple selections ($b>1$). In fact, this distinction raises an interrogation on items' aggregation, i.e. whether the worth of an item (or instance) depends on the bulk of selected items or not. This leads to the second differentiation made in Sec. 2.3.1 with the modular and non-modular objective functions. For each example of application, the

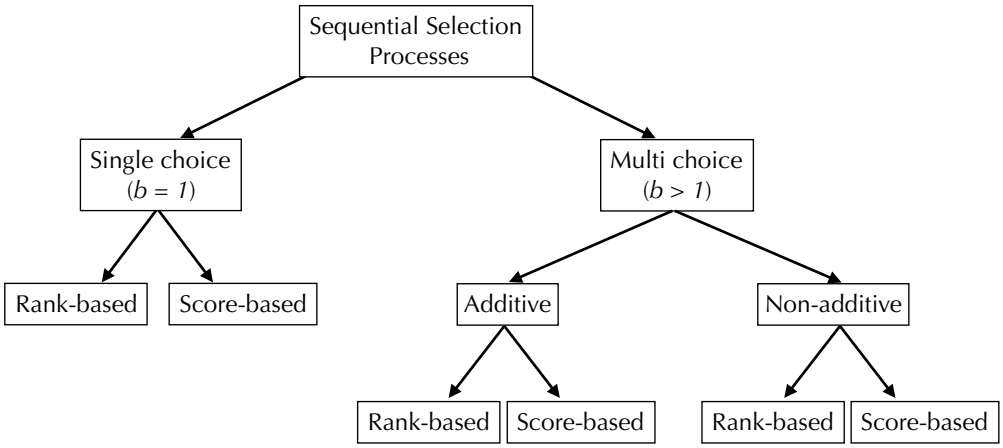


FIGURE 2.1: Organization chart of the main types of SSPs.

score distribution, the objective function, and the assumption(s) of the model are made explicit.

Single-choice SSP ($b = 1$)

In the simplest scenario, only one instance of the stream is selected, i.e. $b = 1$, which puts an end to the process since there is no point to continue examining candidates. In this case, the objective function belongs to either of the following categories:

■ *Rank-based (ordinal):* Each instance's evaluation is made by pairwise comparison with those seen before. In other words, the scores are considered 'unobserved' (i.e. nonexistent); therefore, the score distribution is unknown and non-definable.

Examples:

– *House hunting* problem: the DM sequentially visits houses (or apartments) to rent. In big cities, renting decisions should be made (almost) immediately if the future tenant wants to avoid that the place becomes unavailable, thereby choices are considered irrevocable. This particular problem often occurs in animal groups (bees, ants, etc.) that must choose the best nesting site [51], as well as its distributed version where multiple DMs examine the possible sites separately to find potential nest candidates, before reaching a collective decision [99].

– *Marriage* problem: defined as above, by replacing instances with potential partners and where there can be an increasing interview cost [11]. Note that, contrary to the previous definition, in some papers the Marriage problem refers to a matching problem between two groups of individuals and whose objective is to find a stable match.

– *Parking spot* problem: defined as above, by replacing instances with parking spots [71].

■ *Score-based:* Each instance's evaluation is made through the observation of a real-valued score that represents its worth. The score distribution may be known or unknown.

Examples:

– *Strategy-proof online single-item auction* [57]: An auctioneer has a single good to sell to a group of bidders, that dynamically arrive and depart. Each bidder (or instance, in our context) has its own evaluation of the good, given through a score. The problem is constructed so that each bidder's dominant strategy, i.e. a strategy that is better for himself regardless of how his opponents may play, is to truthfully give its valuable information. Here, the scores are independently drawn from a stationary distribution that is either known (full-information setting in Sec. 5.3 therein) or unknown (no-information case in Sec. 5.1).

Multiple selections SSP ($b > 1$)

Here, the DM selects exactly b instances. Once again, the objective function is divided into two subcategories, it is either *additive* or *non-additive* (subadditive or superadditive).

■ *Additive:* Each instance contributes in the exact same way in the final selection, regardless when it was accepted, or which instances have been selected so far. The scores of the selected instances merely add up to give the score of the whole solution. Formally, if $\phi : 2^{\Omega} \to \mathbb{R}$ is an additive set function over a finite set Ω, such that, $\forall B, A \subseteq \Omega$, then $\phi(A \cup B) = \phi(A) + \phi(B)$. Alternatively, a function is subadditive (respectively superadditive) if $\phi(A \cup B) \leq \phi(A) + \phi(B)$ (respectively $\phi(A \cup B) \geq \phi(A) + \phi(B)$).

Examples:

– *Rank-based Multi-choice Secretary Problem* [8, 5]: the DM is looking for candidates to fill b empty job positions. Sometimes, scores are computed in an evaluation grid (e.g. scores in $[0, 1]$), bounding the score density.

– *Health-care facilities* [53, 15]: in an emergency service the number of resources of a facility, such as beds or nurses, is limited and the demand is high as patients are arriving dynamically.

– *Epidemic control*: this is about the control of the spread of an infectious disease in an online fashion, by means of resource allocation. Again, the budget is very limited compared to the demand, for instance vaccines [108].

– *Strategy-proof online multiple-items auction* [67]: similar to the single-item auction but where the DM has b items to sell.

■ *Non-additive:* Each selected instance influences the value of the next instances to come, for example a submodular objective function implies 'diminishing returns' i.e. the larger the set of selected instances, the less an additional selected instance brings value. Note that selecting a new instance can also devalue the past selected items, as shown in the following example.

Examples:

– The *b-diversification* problem [109]: a robot (the DM) of limited space capacity goes along a pre-defined path and sequentially encounters environmental instances to select from, e.g. insects, fishes, or rocks, etc. The DM must select b instances, each represented by $d \in \mathbb{N}^*$ features, out of the stream in order to maximize the minimum 'distance' between the selected items (typically, an Euclidian distance)

that quantifies the diversity level of the selection set. See Chapter 7 for a detailed study.

– The *gaussian-process regression* problem [73, 83]: similar to the previous exemple, except that instances are now sensors that measure environmental characteristics of 2D locations, and the feature of interest (e.g., temperature or lightning) is not observed as online decisions are made; hence at each sensor the DM solely observes every other environmental feature, e.g. humidity, voltage, etc. The true value of the feature of interest of the selected sensors is only revealed ex-situ and the objective is therefore to learn the utility function that maps an instance to the scalar of interest, e.g. the temperature. Thus, the DM's goal is to select instances that maximize the mutual information gain, a function of the entropy of each element of the set.

– *Kidney exchange* [26]: a kidney exchange program enables matching of incompatible donor-patient pairs via swapping with other incompatible donor-patient pairs, sometimes through a chain of such pairs. The online version of this originally offline problem allows for pairs to arrive sequentially, instead of being all indexed in a pool.

– *Network design*: similarly to the previous setting, this example is related to bipartite matching. In particular, in internet-based systems that sell advertising [70], the two-partite sets are nodes that respectively represent the website slots (all available offline) and the sequentially arriving ads. Each incoming ad, when arriving, also reveals its existing weighted edges with every slot node. The DM must decide immediately whether to accept the ad, and if so, the slot node to match it with. The overall goal is to maximize the final weight of the selected edges.

2.3.2 Relevant algorithms from the literature

In an effort to later compare our algorithms with those already proposed in the literature, we list in this section a couple of them that originally assume a multi-choice SSP. In addition, for each of them we successively describe the nature of the objective function (additive or non-additive), the assumptions they require, the algorithmic procedure and finally the guarantees they provide, if any. Before starting, here is a quick reminder on two categories of online strategies:

- *Cutoff-based*: it takes as input a given *cutoff value* $c \in \mathbb{N}$; it rejects by default the first c incoming instances, called the *learning phase* and then selects an instance according to information gathered during the latter phase.
- *Threshold-based*: a particular case of *cutoff-based* strategies with $c = 0$. The j-th arriving instance is accepted if its score beats a specified *acceptance threshold*, denoted by $S_j^{\text{acc.}} \in \mathbb{R}_+$.

Also, recall that the rank-based (or ordinal) modeling approach assumes that decisions are based on the instances relative pair-wise comparisons only, rather than on their real-valued scores, the former allowing for elegant solutions and robust regardless the underlying score distribution. The rank-based setting can be interpreted with the implicit assumption that the DM cannot learn any valuable information from the observed scores (sort of 'dummy scores') or that the DM cannot even

observe those scores, and hence simply evaluates instances against each other (see Remark 3 in Chapter 3).

KLEINBERG (2005)

In the algorithm presented in [67], which we call KLEINBERG here for simplicity, the original objective function is additive and more precisely it is given by the sum of the scores of the selected instances. As every variant of the secretary problem, this assumes that instances scores arrive uniformly at random, such that no two instances have the same value. It is implicitly assumed that the only purpose of scores is to enable the comparison of instances, meaning that the evaluation is essentially rank-based. The guarantee of KLEINBERG algorithm is given through its competitive ratio, see Eq. 4.9, which is equal to $1-O(\sqrt{1/b})$. In addition, it is also claimed in the paper that the competitive ratio of any algorithm is $1-\Omega(\sqrt{1/b})$, which gives, for instance $1-\sqrt{1/5}\approx 0.553 \leq \mathbb{E}[\text{ALG}]/\text{OPT}$ for $b=5$ resources. The algorithm proceeds as follows: the DM draws a random value $m \sim \text{Binom}(n, 1/2)$ and recursively applies the standard SP algorithm to select up to $l=\lfloor b/2 \rfloor$ candidates among the first m candidates. After the m-th candidate, it selects every instance that exceeds the l-th best seen during the first phase until the b positions are filled.

OPTIMISTIC (2007)

The OPTIMISTIC algorithm [8] was originally conceived for the *knapsack Secretary Problem* in which every instance j has an associated weight $w(j)$ in addition to the standard value of its worth $v(j)$. The knapsack constraint over the set of selected instances S is such that $\sum_{j\in S} w(j) \leq W$, with W a given weight threshold, renders the objective function non-additive. However since we do not generally consider a knapsack constraint in this thesis, but an explicit set-size budget, the version of the OPTIMISTIC algorithm we are interested in is the one presented for the unweighted case, i.e. where $w(j)=1, \forall j$, in Sec. 3 therein. Therefore the setting degenerates to a additive objective function which is comparable to the rest of the methods presented in this thesis. Similarly to the setting of the KLEINBERG algorithm described above, each instance carries a score value, whose purpose is to allow for instances comparison and from which no meaningful learning can be attempted, i.e. the process intrinsically rank-based. The OPTIMISTIC guarantee is also presented through its competitive ratio, see Eq. 4.9, that leans towards $1/e$ as n tends to infinity; The algorithm is cutoff-based, and takes as a cutoff value $c=\lfloor n/e \rfloor$. The acceptance threshold $S_j^{\text{acc.}}$ is dynamic and depends on the instances' arrival time and potentially on the number of instances accepted so far. More specifically, the threshold is equal to the relative rank of the b-th best recorded during the learning phase when no instance has been accepted yet. An insightful comment on the OPTIMISTIC algorithm concerns its appreciated fast and easy implementation.

Hiring-above-the-mean (MEAN) (2009)

The objective function of the MEAN strategy [21] differs a bit form the other strategies presented so far. In the original setting, each incoming instance $j \leq n$ has a *quality score* $Q_j \sim \mathcal{U}(0,1)$; and the goal is to keep a good tradeoff between the quality and the speed of selection (here the motivation is recruiting). The worth of

each instance's being independent on those selected before, the objective function is additive. Let the average quality of $b+1$ employees be denoted b y A_b (it stars with $A_0 \in [0,1]$). To quantify the rate of convergence of the latter and the rate at which candidates are hired, the *gap* value $G_b = 1 - A_b$ is defined, which converges to 0 almost surely when b goes to infinity. Among other results, the authors prove that after $b^{3/2}$ candidate interviews, the expected value for the mean gap for the best b candidates is $O(\sqrt{1/b})$, which makes the strategy close to optimal given this particular evaluation criterion. The MEAN algorithm is a *threshold-based* strategy for which the dynamic acceptance threshold is the average score of the instances selected so far, which adapts to each new selection. We also consider the MEDIAN strategy presented in the same work where the acceptance threshold used is the median score of those selected so far. However, the MEAN algorithm is the only strategy out of those presented here that allows for a warm-start, hence why it is included in this list.

SUBMODULAR (2013)

The work of [12] studies a general setting where the objective function is submodular, i.e. the benefit of each instance selection is non-increasing as the set grows, see Definition 16 in Chapter 7.

This way, a wide range of settings are allowed, the standard sum of scores being the special case of the above statement with an equality. For the algorithm's sake, n is assumed to be a multiple of b, otherwise 'dummy' instances could be added at the end of the sequence without loss of generality. In addition to a competitive ratio of $\frac{1-1/e}{7}$, the paper provides a lower bound of $b(1-1/e)$ over the expected total number of selected instances, since here, the setting enables at most b instances to be selected, rather than exactly b as in [83]. The entire sequence of n candidates is divided into $\lfloor n/b \rfloor$ sub-sequences. In each of them, the standard SP is applied, i.e. $c = n/(be)$ instances are rejected by default at the beginning of each sub-sequence, and at most one instance is accepted in [12] in every of them.

SINGLE-REF (2019)

The original objective function is additive. Similarly to the KLEINBERG and OPTIMISTIC algorithms, it is assumed that decisions are based on the instances' total order only, rather than on their real-valued scores. Guarantees are, again, presented in the form of a competitive ratio that is shown to be optimal for $b = 1$ using the SINGLE-REF algorithm [5], and equal to 0.4119 for $b = 2$, and hence better than the standard $1/e$. In addition, numerical computations show the monotonic increase of the competitive ratio with the budget size b, reaching 0.6306 for $b = 20$, for instance, and greater than the 0.5 threshold from $b = 6$ and on. The SINGLE-REF algorithm is cutoff-based, and takes as input a cutoff value c and a fixed acceptance threshold $S_j^{\text{acc.}}$, both empirically optimized according to the number of resources b. For instance, when $b = 5$, then $c = 0.2525\, n$ and each candidate should beat the second best recorded during the learning phase, while when $b = 50$, then $c = 0.1536\, n$ and each candidate should beat the ninth best recorded during the learning phase. A large table of these optimized parameters is provided in the paper. The SINGLE-REF algorithm is attractive both analytically, by its plain combinatorial nature, and numerically due to the fixed acceptance threshold that allows an easy implementation.

Table of main notations for Chapter 3 and Chapter 4

Term/Symbol	Description
$\mathbb{1}\{\text{condition}\}$	indicator function that gives 1 if the condition is true, otherwise 0
$\mathbf{1}_{[k]}$	unit vector of length k
$\mathcal{P}_l(E)$	set of l-combinations of the finite set E, or simply $\mathcal{P}_l$ when $E = \{1,...,l\}$
$\mathcal{B}$	collection of information or items initially available to the DM
$b \in \mathbb{N}^*$	budget of resources: number of work positions to manage
$n \in \mathbb{N}^*$	number of candidates to appear
$R_N : \mathbb{R} \times \mathbb{R}^N \to \{1,...,N\}$	ranking function that gives to each element of a collection of N values its rank when compared to the other value
$\dot{\mathbf{S}} = (\dot{S}_{(1)},...,\dot{S}_{(b)}) \in \mathbb{R}^b$	scores of the reference set from best to worst consistent with the ordinal model
$\dot{\mathbf{X}} = (\dot{X}_{(1)},...,\dot{X}_{(b)}) \in \mathbb{R}^b$	absolute ranks of the reference set from best to worst
$\dot{\mathbf{A}}_0 = (\dot{A}_{0,(1)},...,\dot{A}_{0,(b)}) \in \{0,1\}^b$	availability status of the reference set
$r \in \mathbb{N}$	number of resignations
$\mathbf{S} = (S_1,...,S_n) \in \mathbb{R}^n$	candidates scores consistent with the ordinal model, in the order of appearance
$\mathbf{X} = (X_1,...,X_n) \in \mathbb{R}^n$	candidates absolute ranks in the order of appearance
$\mathbf{A} = (A_1,...,A_n) \in \{0,1\}^n$	a specific sequence of decisions
$\phi_{\mathcal{B}} : \mathbb{R} \times \mathcal{A} \to \mathbb{R}_+$	regret function
$\phi_{\text{off}} \in \mathbb{R}$	minimal obtainable cost by an offline oracle strategy
$q \in [0,1]$	estimate measure of the average quality of the reference set compared to the candidates
$\dot{\mathbf{X}}^{\text{avail}} \in \{\dot{X}_{(i)} : \dot{A}_{0,(i)} = 1, \forall i \leq b\}^{b-r}$	available referents absolute ranks
$\tilde{A}_j = \sum_{i=1}^{j} A_i$	number of new hires up to candidate j (included)
$c \in \mathbb{N}$, $c^* \in \mathbb{N}$	cutoff, optimal cutoff value when using CCM, i.e. integer that specifies the size of the learning phase
$\mathbf{S}_j^{\text{upd}} \in \{\dot{\mathbf{X}}, X_1,...,X_j\}^b$	updated reference set i.e. set of the b-best items known by the DM after j interviews ($\mathbf{S}^{\text{upd}}$ without subscript refers to $\mathbf{S}_c^{\text{upd}}$, i.e. the b-best known up to the end of the learning phase)
$S_j^{\text{acc.}} \in \mathbb{R}$	score-based acceptance threshold i.e. value to beat for candidate j in order to be accepted
$\gamma_0 \in \mathbb{R}_+$	expectation of the rank-based threshold at the beginning of a WSSP
$\gamma_j \in \mathbb{R}_+$	expectation of the rank-based threshold $R_{n+b}(S_j^{\text{acc.}}, \{\dot{\mathbf{S}}, \mathbf{S}\})$
$g_j(x) = \mathbb{P}(\tilde{A}_{j-1} < x) \in [0,1]$	prob. that less than x candidates have been accepted up to step $j-1$
$\equiv_{(\gamma_0)}$	settings of two instances are said to be γ_0-similar if their reference sets have the same ranks w.r.t. the rest of the sample
$f_j \in \{0,1\}$	event of accepting a last incoming candidate at step j by default (to fill empty job positions)
$\rho_f \in \mathbb{R}$	failure rate i.e. sum of the number of failures divided by the number of tests
$\hat{\mu}_j(c,r) \in \mathbb{R}$	expected number of accepted candidates at step j given that the total number of accepted candidates (i.e. at the end of the selection) is greater or equal to r
$Z \in \mathbb{R}^2$	area around the expectation of the number of accepted candidates inside which the threshold γ_j is identical to that of CCM
$w_j \in \mathbb{R}$	thickness of zone Z
$D_j \in \mathbb{R}$	parameter which determines how the threshold is updated when a point lies outside the zone
$k \in \mathbb{N}^*$	round number

TABLE 2.1: Index of main notations. Horizontal lines indicate groups of notations.

Chapter 3

The Warm-starting Sequential Selection Problem

3.1 Introduction

Since its introduction in the early 60's, the Secretary Problem (SP) [39, 52, 47] has been the most famous optimal stopping problem: n randomly incoming candidate secretaries are interviewed one after the other for a job position. In each interview, the *decision maker* (DM) acquires information about a candidate's competence, and by that compares him with the so far examined candidates. She can decide when to terminate the process by selecting the last candidate interviewed. Although she knows how many candidates will appear, the DM has no knowledge of who will come later on, yet her decisions should be *immediate* and *irrevocable* after each interview. This describes a *Sequential Selection Problem* (SSP[1]). The class of SSP problems is attractive for theoretical analysis and for practical use, due to its generality and evident relevance to online selection under realistic constraints. Same as in this work, SSPs are usually presented in the intuitive recruitment context.

The goal of the original problem is to select none but the best among the sequence of n candidates, while in each interview the DM only realizes the relative quality of the examined candidate, that is his relative rank. The algorithm proposed in [39], is a *cutoff-based* approach which comprises two phases: the *learning phase* where a number (referred to as *cutoff*) of candidates are automatically rejected, and the *selection phase* where the first candidate ranked above the best recorded during the first phase is hired (or the last one, by default). In essence, the former phase learns a threshold that is subsequently used in the latter to spot the first candidate to beat it. Significant advantages of any cutoff-based strategy is that they are intuitive and easy to implement. The length of each phase is determined by the a cutoff value and is subject to an *exploration-exploitation trade-off* that depends on the considered objective function to optimize. For instance, the optimal cutoff for maximizing the probability to find the best candidate is $c^* = n/e$ for large n and constant c/n ratio. Besides, when candidates have an underlying *quality score*, rather than a relative quality like his rank among the already examined candidates, one may be interested in maximizing the expectation of the score of the selected candidate. For this objective function, and under the assumption that i) scores are uniformly drawn from $[0,1]$, and ii) scores are unobserved i.e. the DM can only see the candidates relative ranks, the optimal cutoff is $c^* = \sqrt{n} - 1$ [13]. Note that the multi-choice

[1] Depending on the context, the last letter of the abbreviations SSP and the herein presented MSSP may refer to the respective selection '*Problems*' or the associated selection '*Processes*'.

problem where the DM can select more than one candidate is a natural extension of the above (see Sec. 2.1).

Motivation and contribution. Our motivation derives from real-world recruitment processes that take place in large organizations or companies whose aim is to dynamically adapt in their operating environments. Typically, an organization has already many referents and the DM has the challenging task to keep the personnel as competitive as possible at any moment in time. Moreover, the DM has to ensure that jobs are always assigned to referents. It is easy to see why such an organization requires constant recruitment processes in parallel to their operation cycle. This setting goes beyond the existing SSP models in literature that have two important limitations, namely they consider a *cold-start* initialization where there is no assignment of jobs at the beginning of the selection process and ii) they operate in a *single-round* where only one sequence of candidates is processed.

To address these issues, we introduce a new online *initialized* problem that we call *Warm-starting SSP* (WSSP): at the beginning of the selection, the DM has at hand a *reference set* of referents for whom she knows the status of *availability* (referents are allowed to quit their jobs just before the beginning of the interviews), and eventually the *average quality* w.r.t. the candidates. In order to address the second limitation, we propose a new *online-within-online* problem, the *Multi-round Sequential Selection Process* (MSSP). Essentially, each round of the latter can be seen as a single WSSP initialized with the items selected in the previous round; the overall goal being to keep a highly-skilled set through time. As for the technical contributions, in this chapter we analyze the Warm-starting SSP and derive analytical formulas for: i) the initialization, specifically the expected rank of the referents (available or not) and the minimal regret of an offline strategy.

Notations. A bold symbol denotes a vector, for instance, $\mathbf{A} = (A_1, ..., A_k) \in \mathbb{R}^k$, $\forall k \in \mathbb{N}^*$, in which with little abuse we omit the symbol of the transpose. The concatenation of matrices is denoted by $(\mathbf{A}, \mathbf{B})$, and $\mathbb{1}\{\cdot\}$ is the indicator function, which is 1 if the input condition is true, and otherwise 0.

3.2 A general class of Sequential Selection Processes

In a standard SSP, *candidates* for a job position arrive sequentially in random order. The qualitative skills of each candidate can be assessed independently on his arrival by the DM, allowing the relative ranking of the examined candidates against each other. According to this evaluation, the DM chooses who to hire in order to optimize a given *evaluation criterion*.

In the class of *Generalized Sequential Selection Processes* (GSSP) that follows, what we call as *Background* $\mathcal{B}$ is essentially the context in which the selection process takes place. It contains, among other elements, the set $\mathcal{A}$ of all possible actions the DM can take (e.g. hire, fire, add in queue, put on standby, etc.). A set of assumptions regarding the sequential arrivals and/or the decision process can be added to the following definition, according to the context.

Definition 1. *Generalized SSP (GSSP): Online selection process described by the following elements organized in several categories:*

1. *Background*

   $\mathcal{B}$*: collection of information known upfront by the DM, including the set* $\mathcal{A}$ *of all possible actions the DM can take (e.g. hire, fire, add in queue, put on standby, etc.).*

2. *Sequential Arrivals*
   - $\mathbf{S} = (S_j)_{j\geq 1}$*: sequence of candidate scores s.t.* $S_j \in \mathcal{S} \subset \mathbb{R}$*, drawn from distribution* f_j*,* $\forall j \geq 1$*.*

3. *Decision Process*
   - $\pi = (\pi_j)_{j\geq 1}$*: policy, i.e. sequence of mappings from observations to decisions;*
   - $\mathbf{A} = (A_j)_{j\geq 1}$ *: sequence of decisions regarding the candidates, according to the policy* π_j*, i.e.* $A_j = \pi_j(S_j \,|\, S_1, ..., S_{j-1}, A_1, ..., A_{j-1}) \in \mathcal{A}$*,* $\forall j$*.*

4. *Evaluation*
   - $\ell : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$*: loss function s.t.* $\ell(s,a)$ *is the loss for taking decision* a *right after observing* s*;*
   - $L(\mathbf{S}, \mathbf{A}) = \sum_{j\geq 1} \ell(S_j, A_j)$*: cumulative loss;*
   - *Let* P *be the distribution of* $(\mathbf{S}, \mathbf{A})$*. The evaluation criterion, called* regret*, is evaluated at the end of the process and defined as* $\Phi(\pi) = \mathbb{E}_P[\phi_{\mathcal{B}}(\mathbf{S}, \mathbf{A})]$*, where:*

$$\phi_{\mathcal{B}}(\mathbf{S}, \mathbf{A}) = |\, L(\mathbf{S}, \mathbf{A}) - \phi_{\text{off}} \,| \in \mathbb{R}_+, \tag{3.1}$$

   and ϕ_{off} *is a baseline value.*

With the high-level formalization of the GSSP class, we can summarize several well-known processes, such as the indicative ones mentioned below.

3.2.1 Examples of well-known GSSPs

■ Standard SP [39]: A GSSP setting with $\mathcal{B} = (b, n, \mathcal{A} = \{0,1\})$, where $b = 1$ is the number of job position, n is the finite number of candidates, and a candidate is either selected (hired, $A_j = 1$) or rejected ($A_j = 0$). It is assumed that decisions are immediate and irrevocable, that candidates arrive in a random order, and that their scores are unobserved so that the DM can only make pairwise comparison between candidates. This is equivalent to having relative ranks as observations, i.e. a triangular array $(X_{i,j})$ where $X_{i,j} \in \{1, ..., j\}$ is the relative rank of the i-th incoming candidate after having examined $j \geq i$ of them. The vector of absolute ranks, evaluated at the end, is given by $\mathbf{X} = (X_{1,n}, ..., X_{n,n}) \in \mathcal{P}_n$, where $\mathcal{P}_l$ is the set of all permutations of the elements of $\{1, ..., n\}$. The evaluation criterion to maximize is the probability to select the best candidate (the one with absolute rank 1 at the end of the selection), which can be expressed by $\ell(X_{j,j}, A_j) = \mathbb{1}\{X_{j,n} A_j = 1\}$, $\forall j \leq n$, therefore $\phi_{\text{off}} = 0$ and $\Phi = \mathbb{P}(\mathbf{X}^\mathsf{T}\mathbf{A} = 1)$.

■ Prophet inequality [74]: A GSSP setting with $\mathcal{B} = (b, n, \mathcal{A} = \{0,1\}, \mathbf{f})$, where the f_j's are known a priori but not the realizations X_j. It is assumed that decisions are immediate and irrevocable, and that scores are observed. Multiple objectives have been investigated, in [65] it is to maximize the scores of the selected items, i.e. $\ell(S_j, A_j) = S_j A_j$ and $\Phi = \mathbb{E}[\mathbf{S}^\mathsf{T}\mathbf{A}]$, while in [68] the selected items are subject to a matroid constraint. Note that scores are not necessarily i.i.d.; in some settings an adversary can choose the candidates' arrival order.

■ Hiring problem [21]: A multi-choice GSSP setting where, by respecting the trade-off between the rate of hires and the quality of the hired candidates, the objective is to grow the company as much as possible while keeping maximal the average score of the employees. The recruitment process has infinite horizon. Therefore we have $\mathcal{B}=(b,n,\mathcal{A}=\{0,1\})$, where $b\to\infty$ and $n\to\infty$. Again, t is assumed that decisions are immediate and irrevocable, and that observations are i.i.d. scores drawn from a uniform distribution, i.e. $S_j\sim\mathcal{U}(0,1)$. Using the above formalism, the cumulative loss after b hires is given by $L_b(\mathbf{S},\mathbf{A})=-\frac{1}{b}\sum_{j\geq 1}\ell(S_j,A_j)$, where $\ell(S_j,A_j)=S_jA_j$, and the objective is for the expected regret from Eq. 7.7, with $\phi_{\text{off}}=1$, to convergence towards zero w.r.t. b as fast as possible.

Remark 1. *The number of job positions b and of candidates n are usually included in the background $\mathcal{B}$; (i.e. known by DM) however variants of the standard SP [102, 72] may involve an unknown number of candidates. Note that $b>1$ is sometimes referred to as a multi-choice or multi-stopping problem.*

Remark 2. *In most GSSP settings, the loss suffered at each decision is the score of an accepted item, i.e. $\ell(S_j,A_j)=\pm S_jA_j$, with a positive (resp. negative) sign if the goal is to minimize (resp. maximize) the sum of scores. When the goal is to maximize the loss (i.e. the payoff in this case), the regret is called* reward. *The evaluation criterion is further detailed into two cases: 1) the 'no regret' case, where the DM merely tries to optimize its selection i.e. for $\phi_{\text{off}}=0$, and 2) the 'with regret' case, where the online selection is to be compared to the best associated offline strategy π_{off}, where the DM knows the entire sequence of candidates beforehand, in this case $\phi_{\text{off}}=\min_{\mathbf{O}\in\pi_{\text{off}}}\phi(\mathbf{S},\mathbf{O})$ (or $\phi_{\text{off}}=\max_{\mathbf{O}\in\pi_{\text{off}}}\phi(\mathbf{S},\mathbf{O})$ when the goal is to maximize the sum of the scores).*

3.3 Introducing the warm-start

3.3.1 Description and rules of the game

The *Warm-starting SSP* (WSSP) is a particular GSSP instance that overcomes the limitations of standard cold-starting SSP frameworks. Its characteristics is to start with a set of items at hand, called *reference set* and composed of *referents*, each of them having also a status of *availability*. The total number of job positions determines the size of the reference set. Items can therefore be of two types, candidate or referent. The value of each item is given by a real-valued score, however it is not observed by the DM that can only make pairwise comparisons between items. Although the referent's availability status can be broad (e.g. on vacation, sick leave, resigned, etc.), we only allow resignations, i.e. a referent is *unavailable* if he resigned (leaving his position empty) and *available* otherwise (in other words, he is preselected). In this paper, we work under the simple assumption that resignations are independent. The DM therefore seeks highly-skilled candidates to 1) fill up empty positions and 2) replace non-competitive available referents; by respecting the following specific constraints.

Assumption 1. *On the sequence of arriving candidates:*

1.A) Candidates arrive in arbitrary random order.

1.B) Scores are not observed, the DM can only make pairwise comparisons between items.

Assumption 2. *On the decision policy:*

2.A) The availability status is known upfront, and fixed throughout the process.

2.B) Decisions are immediate and irrevocable.

2.C) Every position must be filled at the end of the process (by a candidate or an available referent).

Formal definition. We add a dot on top of a variable to refer explicitly to the reference set, e.g. $\dot{\mathbf{S}} = (\dot{S}_{(1)}, \dot{S}_{(2)}, ..., \dot{S}_{(b)}) \in \mathbb{R}^b$ gives the value represented by the variable $\mathbf{S}$ (here, scores) of the referents in descending order: the best, the second best, etc.

Definition 2. *The ranking function* $R_N : \mathbb{R} \times \mathbb{R}^N \to \{1, ..., N\}$ *gives to each element of a collection of* N *values its rank when compared to the other values. Let* $\mathbf{\Sigma}$ *a finite number set, then* $\forall s \in \mathbf{\Sigma}:\ R_N(s, \mathbf{\Sigma}) = \sum_{i=1}^{N} \mathbb{1}\{\Sigma_i \leq s\}$.

Definition 3. *Warm-starting SSP (WSSP): A particular GSSP with the following characteristics:*

1. *Background*

   $\mathcal{B} = (n, b, \mathcal{A}, \dot{\mathbf{S}}, \dot{\mathbf{A}}_0)$, *where the included elements are:*
   - $n \in \mathbb{N}^*$*: finite number of candidates to appear;*
   - $b \in \mathbb{N}^*$*: number of job positions s.t.* $b \leq n$*;*
   - $\mathcal{A} = \{0, 1\}$*: the set of possible actions the DM can take, respectively reject or hire;*
   - $\dot{\mathbf{S}} = (\dot{S}_{(1)}, ..., \dot{S}_{(b)}) \in \mathbb{R}^b$*: reference set scores (sorted from best to worst for convenience);*
   - $\dot{\mathbf{A}}_0 = (\dot{A}_{0,(1)}, ..., \dot{A}_{0,(b)}) \in \{0, 1\}^b$*: availability status of the reference set s.t.* $\dot{A}_{0,(i)} =$ 1 *if the* i*-th best referent is available.*

2. *Sequential Arrivals and 3. Decision Process as in Definition 1*

4. *Rank-based evaluation*

   The following simplified notation for the absolute ranks is written $R(s) = R_{b+n}(s, (\dot{\mathbf{S}}, \mathbf{S}))$.
   - $\dot{\mathbf{X}} = \big(\dot{X}_{(1)} = R(\dot{S}_{(1)}), ..., \dot{X}_{(b)} = R(\dot{S}_{(b)})\big) \in \mathbb{R}^b$*: referents' absolute ranks,*
   - $\mathbf{X} = \big(X_1 = R(S_1), ..., X_n = R(S_n)\big) \in \mathbb{R}^n$*: candidates' absolute ranks,*
   - *Let* P *be the distribution of* $(\mathbf{X}, \mathbf{A})$*. The evaluation criterion, called* regret*, is evaluated at the end of the process and defined as* $\Phi(\pi) = \mathbb{E}_P[\phi_{\mathcal{B}}(\mathbf{X}, \mathbf{A})]$*, with:*

$$\phi_{\mathcal{B}}(\mathbf{X}, \mathbf{A}) = \left(\dot{\mathbf{X}}^\intercal \dot{\mathbf{A}}_n + \mathbf{X}^\intercal \mathbf{A}\right) - \min_{(\dot{\mathbf{O}}_n, \mathbf{O}) \in \pi_{\text{off},\mathcal{B}}} \left(\dot{\mathbf{X}}^\intercal \dot{\mathbf{O}}_n + \mathbf{X}^\intercal \mathbf{O}\right) \in \mathbb{R}_+, \quad (3.2)$$

   where $\pi_{\text{off},\mathcal{B}} = \left\{(\dot{\mathbf{O}}_n, \mathbf{O}) \in \{0, 1\}^{b+n} : \|(\dot{\mathbf{O}}_n, \mathbf{O})\|_1 = b\right\}$ *and* $\dot{\mathbf{A}}_n \in \{0, 1\}^b$ *is the hiring decisions of the referents after* n *interviews of candidates.*

The first term in Eq. 3.2 is the sum of the ranks of the items to which jobs have been assigned at the end of the selection. The second term is the minimal regret achievable by an offline oracle strategy that, knowing the ranks, would select the best out of the available referents[2] (i.e. for some i: $\dot{A}_{0,(i)} = 1$) and the candidates, see Eq. 3.6.

Remark 3. *In this work we make no assumptions at all about the source and nature of the scores (see Assumption 1.B)). This is why we consider unobserved scores and adopt a rank-based criterion to assess the selection strategy, which is a standard approach in nonparametric statistics [30]. This choice is motivated by real-life scenarios where observations might be noisy, the items score distribution changing, or non-standard [109, 29].*

Example. A DM manages $b = 3$ job positions and is about to interview $n = 20$ candidates, one after the other. Before these interviews, only the best referent resigns ($r = 1$), and by that leaves one position empty, while the two others stay at their positions, i.e. $\dot{\mathbf{A}}_0 = (0,1,1)$. Upon arriving, the first candidate is compared to the referents; say he is better than all of them, i.e. has a relative rank of 1. The DM decides to hire him, i.e. $A_1 = 1$, and this candidate fills up the initially empty position. Then, after some candidate rejections, the DM decides to hire the 15-th incoming candidate ($j = 15$), because he is better than the worst available referent, i.e. has a relative rank of 3. It follows $A_{15} = 1$ and $\dot{A}_{15,(2)} = 0$ since the second best referent just got fired. The process goes on until the end of the sequence is reached, or when all available referents have been replaced.

3.3.2 Initial analysis

Defining the quality

A natural question that arises from the existence of the reference set concerns the 'value' (or quality) of the referents compared to the candidates next to come. How 'good' is our initial set with respect to the arriving candidates? Besides, a notion of 'good' should also be defined. We address the latter interrogation by introducing the 'goodness' of $\dot{\mathbf{X}}$ for $\mathbf{X}$, called *quality of the reference set* and denoted as q (see Definition 4). This parameter quantifies how the reference set ranks on average compared to the candidates. Herein, we suppose that this parameter is provided in advance to the DM, learned or estimated.

Definition 4. *True rank-based relative quality of reference set (q): For a WSSP, q is the average normalized rank of the b items of the reference set compared to the n candidates:*

$$q := 1 - \frac{\frac{1}{b}\|\dot{\mathbf{X}}\|_1 - 1}{n+b-1}, \tag{3.3}$$

where $\dot{\mathbf{X}} = (\dot{X}_{(1)} = R(\dot{S}_{(1)}), ..., \dot{X}_{(b)} = R(\dot{S}_{(b)}))$ are the referents absolute ranks, $q \in]0,1[$, with $q \to 1$ as the reference set gets better skilled and $q = 1/2$ corresponds to the medium quality s.t. $\frac{1}{b}\|\dot{\mathbf{X}}\|_1 = \frac{1}{2}(n+b+1)$.

Other options to define the quality have been envisioned, as for instance, the following.

[2] The unavailable referents are unreachable, even for the oracle.

- The DM knows the absolute ranks of each referent, i.e. the vector $\mathbf{q} = (q_{(1)}, ..., q_{(b)}) \in]0,1[$, where $q_{(i)} = 1 - \frac{\dot{X}_{(i)}-1}{n}$ is the quality of the i-th best referent. This alternative is the one with the most information in this rank-based setting, the DM knows, beforehand, how every referent ranks individually compared the the candidates to be interviewed. This strong assumption makes it difficult to apply in most situations, hence why we chose to discard it in this study.

- The DM knows the rank of the worse referent, i.e. $q_{(b)} = 1 - \frac{\dot{X}_{(b)}-1}{n}$. Here, the assumption is that the worse candidate makes for the quality of the overall reference set, which is a rather hard constraint, and might be unfortunate, if, for instance, he is an outlier compared to the other referents.

- Although unobserved, the scores could also dictate the quality; i.e. the DM knows the average score-based $q = \frac{\frac{1}{b}\|\dot{\mathbf{S}}\|_1 - S_{\min}}{S_{\max} - S_{\min}}$. The DM would take advantage of this definition of the quality solely if the evaluation criterion was defined through the items' scores, and not the absolute ranks as is the case here.

Offline analysis

Initialization and resignations. This analysis concerns the initialization of the process, i.e. before the arrival of candidates, and is independent on the chosen strategy. Although the status of referent's availability can be broad (e.g. on vacation, sick leave, resigned, etc.), we only allow resignations, i.e. a referent is *unavailable* if he resigned and *available* otherwise.

Definition 5. *Resignation, resignation number $r \in \mathbb{N}^*$: Event by which a former employee leaves his job position, occurring prior to the selection process. The DM knows which employee are unavailable, i.e. who resigned and thus cannot be called back, before the interviews start. The number of resignations is given by $r := b - \|\dot{\mathbf{A}}_0\|_1 \in \mathbb{N}$.*

In this paper, we work under the simple assumption that resignations are independent. The DM has information about the average quality of the referents, but we are particularly interested in the available ones, i.e. those with ranks $\dot{\mathbf{X}}^{\text{avail}} = (\dot{X}_{(i)})_{i \in I}$, where $I = \{i : \dot{A}_{0,(i)} = 1\}$. These preselected referents might end up, if competitive enough, in the final selection.

Proposition 1. *Let a given WSSP starting with $r \leq b$ resignations. The expectation of the rank of the l-th item from the available reference set $\dot{\mathbf{X}}^{\text{avail}}$ is given by:*

$$\mathbb{E}[\dot{X}^{\text{avail}}_{(l)}] = \frac{\gamma_0(b+1)l}{b(b-r+1)}, \ \forall l \leq b-r \tag{3.4}$$

$$\textit{where} \quad \gamma_0 := \mathbb{E}[\dot{X}_{(b)}] = (1-q)\frac{2b(n+b-1)}{b+1} + \frac{2b}{b+1}, \tag{3.5}$$

s.t. γ_0 is the expectation of the b-th item from the reference set $\dot{\mathbf{X}}$, and a function of the relative quality q of the reference set.

Proof. Eq. 3.5 derives from the definition of the quality in Definition 4, and uses the fact that $\mathbb{E}[\dot{X}_{(i)}] = \frac{\mathbb{E}[\dot{X}_{(b)}]}{n} i = \frac{\gamma_0}{n} i, \ \forall i \leq b$. The best available referent, i.e. with rank

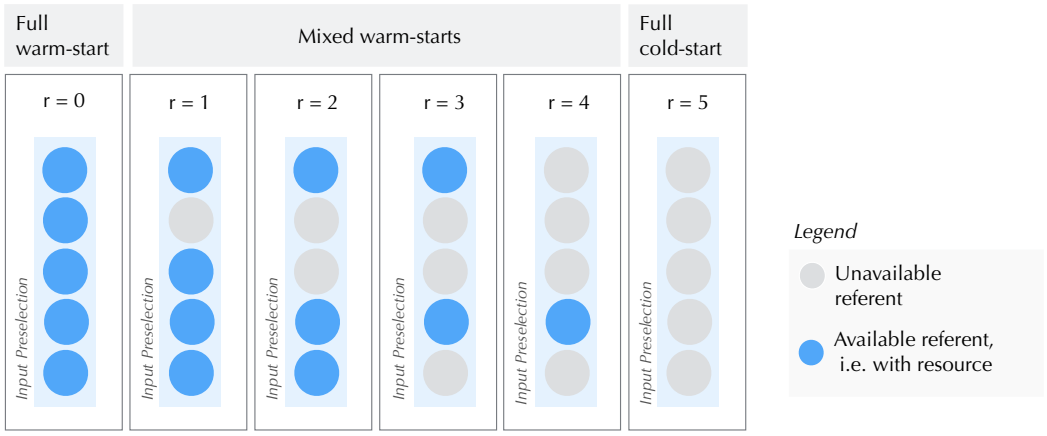


FIGURE 3.1: Example scheme of different initial states for a reference set of size $b=5$, with different resignations $r \in \{0,...,b\}$. Mixed warm-starts differ from full state, in that when $r \neq \{0,b\}$ there is an additional variable on which referent(s) is (are) unavailable.

$\dot{X}^{\text{avail}}_{(1)}$ is therefore expected to have a rank at best γ_0/b and at worst $\gamma_0(r+1)/b$. He has an expected rank of γ_0/b iff the available item(s) are any of the $b-1$ below him in the ranking, i.e. with a probability $\binom{b-1}{r}/\binom{b}{r}$. Then, he has an expected rank of $2\gamma_0/b$ iff the best referent resigned and the $r-1$ other unavailable referents are any of the $b-2$ below him in the ranking, i.e. with probability $\binom{b-2}{r-1}/\binom{b}{r}$. Finally:

$$\mathbb{E}[\dot{X}^{\text{avail}}_{(1)}] = \sum_{i=1}^{r+1} \mathbb{P}\left(\dot{X}^{\text{avail}}_{(1)} = \mathbb{E}[\dot{X}_{(i)}]\right) \mathbb{E}[\dot{X}_{(i)}]$$

$$\mathbb{E}[\dot{X}^{\text{avail}}_{(1)}] = \sum_{i=1}^{r+1} \frac{\binom{b-i}{r+1-i}}{\binom{b}{r}} \frac{i\gamma_0}{b} = \frac{\gamma_0}{b\binom{b}{r}} \sum_{i=1}^{r+1} \binom{b-i}{r+1-i} i,$$

from the multiset relation $\sum_{i=0}^{n} \binom{m+i-1}{i} = \binom{n+m}{n}$ we obtain:

$$\mathbb{E}[\dot{X}^{\text{avail}}_{(1)}] = \frac{\gamma_0}{b\binom{b}{r}} \left((r+1)\binom{b+1}{r+1} - (b-r)\binom{b+1}{r} \right) = \frac{\gamma_0(b+1)}{b(b-r+1)}.$$

Using $\mathbb{E}[\dot{X}^{\text{avail}}_{(l)}] = \mathbb{E}[\dot{X}^{\text{avail}}_{(1)}]l$, $\forall l \in \{1,...,b-r\}$, we obtain $\mathbb{E}[\dot{X}^{\text{avail}}_{(l)}] = \frac{\gamma_0(b+1)l}{b(b-r+1)}$. □

Offline selection. It is desirable for any online algorithm to perform as close as possible to the optimal *offline case* where the DM knows the b-best items and can directly select them. Hence, we want our strategy to converge towards the offline case and have ϕ as small as possible. The offline output $\phi_{\text{off}} \in \mathbb{R}_+$ is given by Definition 3 as:

$$\phi_{\text{off}} := \min_{(\dot{\mathbf{O}}_n, \mathbf{O}) \in \pi_{\text{off},\mathcal{B}}} \left(\dot{\mathbf{X}}^{\mathsf{T}} \dot{\mathbf{O}}_n + \mathbf{X}^{\mathsf{T}} \mathbf{O} \right), \tag{3.6}$$

where $\pi_{\text{off},\mathcal{B}} = \left\{ (\dot{\mathbf{O}}_n, \mathbf{O}) \in \{0,1\}^{b+n} : \|(\dot{\mathbf{O}}_n, \mathbf{O})^{\mathsf{T}}\|_1 = b \right\}$.

Proposition 2. *In the WSSP context, the expected minimal regret an offline algorithm can achieve, by selecting the b-best out of the $n+b-r$ candidates and available referents, is:*

$$\mathbb{E}[\phi_{\text{off}}] = \frac{b(b+1)}{2} + \frac{rb^2(\gamma_0+r)}{2\gamma_0^2}, \tag{3.7}$$

where γ_0 is given in Proposition 1.

The first term of Eq. 3.7 accounts for the standard average offline regret, i.e. the sum of the b-best ranks, while the second represents the increase due to potentially unavailable items from the b-best.

Proof. Proposition 2. We begin by deriving the variable η that gives the expected number of unavailable referents that belong to the b-best. We know that the available referents ranks are included in the overall referent ranks, i.e. $\dot{X}^{\text{avail}}_{(l)} \in \dot{\mathbf{X}}$, $\forall l \leq b-r$, hence if we write by $\mathcal{L} = \left\{ l \in \{1,...,b\} \mid \dot{X}_{(l)} \notin \dot{\mathbf{X}}^{\text{avail}} \right\}$ the set of referents indices that resigned, we get:

$$\begin{aligned}
\eta = \mathbb{E}\left[\sum_{l\in\mathcal{L}} \mathbb{1}\{\dot{X}_{(l)} \leq b\}\right] = \sum_{l=1}^{b} \mathbb{P}(l \in \mathcal{L})\mathbb{P}(\dot{X}_{(l)} \leq b) &= \frac{r}{b}\sum_{l=1}^{b} \mathbb{P}(\dot{X}_{(l)} \leq b) \\
&= \frac{r}{b}\sum_{l=1}^{b} \mathbb{1}\left\{\frac{\gamma_0 l}{b} \leq b\right\} \\
\Leftrightarrow\ \eta &= \frac{r\,b}{\gamma_0}.
\end{aligned}$$

In $(\dot{\mathbf{X}}, \mathbf{X}) \in \mathcal{P}_{n+b}$, candidates and reference set are ranked jointly, regardless if the referents resigned or not. The optimal regret is defined as the average sum of the b-best available ranks. If one of the unavailable referents is among the b-best, his rank is replaced by the next best available rank (same for multiple unavailable referents), which increases the expected offline regret. Formally:

$$\begin{aligned}
\mathbb{E}[\phi_{\text{off}}] = \frac{b-\eta}{b}\sum_{m=1}^{b} m + \sum_{m=b+1}^{b+\eta} m = (b-\eta)\frac{b+1}{2} + \eta b + \frac{\eta(\eta+1)}{2} \\
= \frac{b(b+1+\eta+\eta^2/b)}{2} \\
\Leftrightarrow\quad \mathbb{E}[\phi_{\text{off}}] = \frac{b(b+1)}{2} + \frac{rb^2(\gamma_0+r)}{2\gamma_0^2}.
\end{aligned}$$

□

3.4 The multi-round extension

3.4.1 Principle

We build upon the WSSP, and introduce the *Multi-round Sequential Selection Process* (MSSP). This is a new selection framework whose novelty lies in that a selection process takes place in multiple successive *rounds* and the global objective function is optimized in each round given the output of the previous one. The *environment* of the problem is set to be on a large population $\mathcal{C}$ of job-seekers. Namely, each round is a WSSP Definition 3 instance on a *sample* of candidates of the population with, as reference set, the final selection of the previous round, to which is individually applied a given *probability to resign*. In the multi-round setting, resignations are defined through this probability to resign $\mathrm{P}^r_{k,i} \in [0,1]$ that is the probability for node

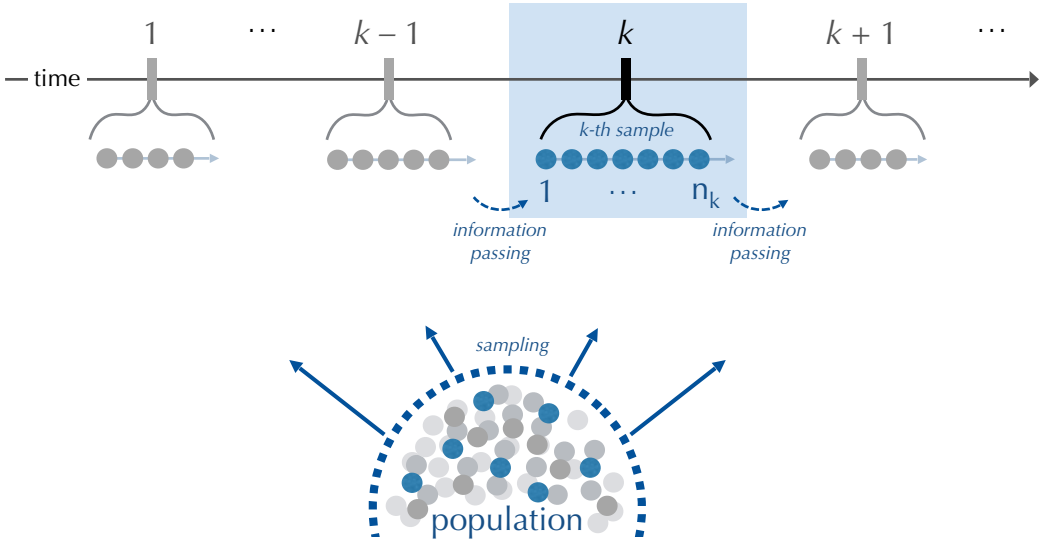


FIGURE 3.2: The timeline of a Multi-round Sequential Selection Process (MSSP). A new round of sequential selection is launched when a new sample of candidates appears. The output of each selection round provides prior knowledge to its next round.

$i \in \mathcal{C}$ that belongs to the final selection of round $k-1$, $k \geq 1$ to become unavailable at the subsequent round k.

A non-exhaustive catalogue of useful forms taken by the resignation process is given thereafter.

- Score-dependent: $\mathrm{P}^r_{k,i} \propto \frac{e^{S_{i,t}}}{\sum_i e^{S_{i,t}}}$, $\forall i \in \mathcal{C}$, $\forall k \leq K$, i.e. candidates are more likely to resign as they are more competent. This case might happen in a process that encourages the highest scores to become unavailable. In the context of maintenance, for instance, incoming candidates represent tasks to be complete, and their score reflects the importance of achieving this task. At each round, the task with the highest scores are therefore more likely to be unavailable, i.e. to be fixed.

- Time-dependent: $\mathrm{P}^r_{k,i} \propto \frac{k}{K}$, $\forall i \in \mathcal{C}$, $\forall k \leq K$, i.e. candidates are more likely to resign as time goes by.

- Uniform: $\mathrm{P}^r_{k,i} = \alpha \in [0,1]$, $\forall i \in \mathcal{C}$, $\forall k \leq K$, i.e. the probability to resign is uniform among the b items of the reference set.

Fig. 3.2 illustrates this selection framework. In addition to the resignations, the MSSP allows for alterations of the score across the population of potential candidates between two subsequent rounds implying i) changes in the shape of the score distribution that leave the population ranking intact, ii) changes in the ranking whereas the score distribution remains intact, iii) a combination of the previous.

Remark 4. *When facing the issue of selecting b instances from a large data-stream it might be efficient to split the data-stream into b warm-start rounds, so that each round start with some valuable information of the previous rounds' selection. This is especially useful when the objective function is non-modular, i.e. when every item's worth depends on the instances selected before (see Chapter 7).*

3.4.2 Sampling from the population

The MSSP requires further assumptions: i) the environment is considered to be fixed during each WSSP round, such that $\mathcal{S}_k \in \mathbb{R}^N$ is the random variable that gives the population's scores at round k, ii) candidates for each round are obtained by sampling from the population according to a given sampling function. The process may have an arbitrary number of WSSP rounds. Therefore, the challenge for the DM is to improve, or at least adapt, the personnel in the course of the multi-round process: at the end of any round that is to have selected the b-best items she could have chosen under the above assumptions and while respecting all the management constraints described for a single round. We use the notations introduced in Sec. 3.3 and add a subscript k at each variable to refer to a precise round k, for instance n_k is the number of candidates at round k.

Since we want to give the reference set a meaningful contribution, i.e. even when all of them resigned at round k, we impose Assumption 3 that guarantees that their absolute ranks do not change drastically from a round to another.

Assumption 3. *If $\mathcal{X}_{k,i}$ is the absolute rank of individual i among the population at round k, then:*

$$\exists \varepsilon \text{ s.t. } \max_i(|\mathcal{X}_{k-1,i} - \mathcal{X}_{k,i}|) < |\mathcal{C}|\varepsilon, \ \forall k \geq 1, \ \forall i \in \{1, ..., |\mathcal{C}|\}, \tag{3.8}$$

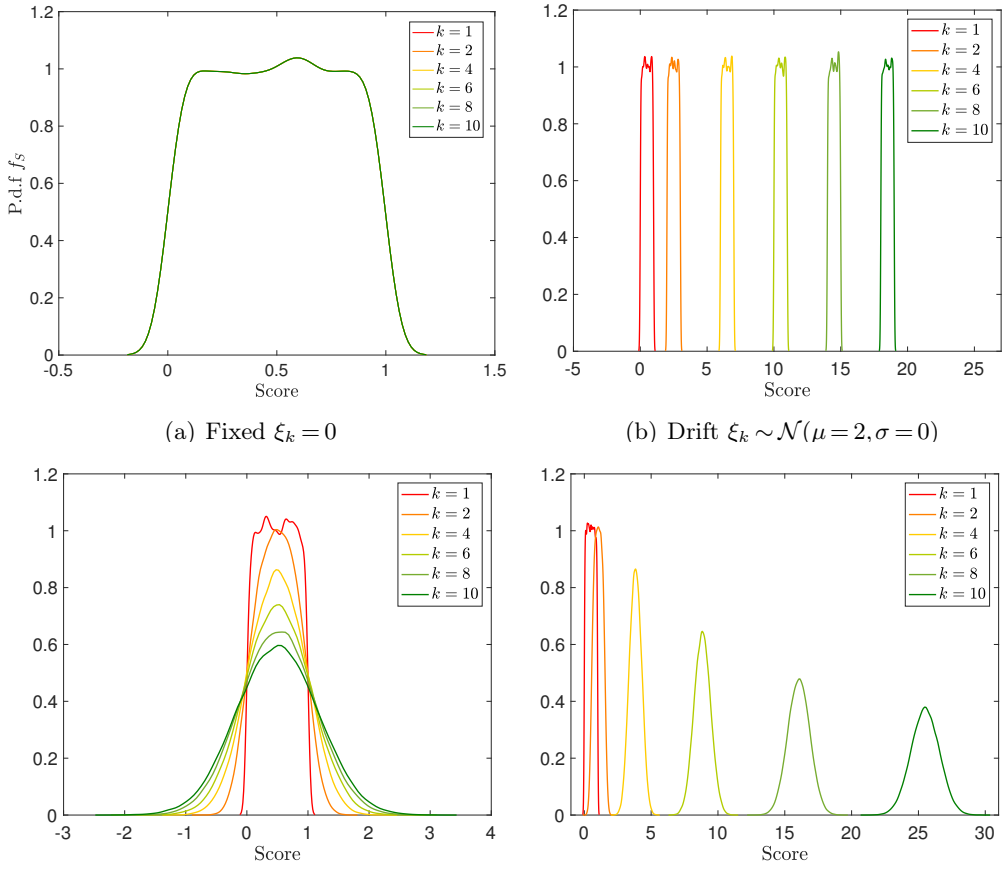
where $\varepsilon \ll 1$, and regardless of the underlying score distribution. $|\mathcal{X}_{k-1,i} - \mathcal{X}_{k,i}|$ expresses how valuable the prior knowledge carried by the reference set is for round k (the smaller the better).

Example. In Fig. 3.3, are displayed some examples of score distributions, fixed for Fig. 3.3(a) and changing for all others, i.e. the interesting case where the population's scores are evolving from a round to another. This situation may occur when, for instance, the population gains skill in a particular domain, leading to a shift in the score distribution; or when the best-skilled job-seekers become even better, while the worst become worse. In general, the employed selection strategy has to be robust to such modifications. For this example, we work under Assumption 3 and with the following choices: $S_{1,j} \sim U(0,1)$, i.e. scores are initially uniformly distributed between 0 and 1, and $S_{k,j} = S_{k-1,j} + \xi_k$ with $\xi_k \sim \mathcal{N}(\mu_k, \sigma_k)$ a gaussian noise. The most commun changes are the drift Fig. 3.3(b), i.e. when the mean of the overall population shifts, for instance when the entire population gains (or lose) skills as group; the deflection Fig. 3.3(c), i.e. when the difference in skills increases among the population; and a combination of the above Fig. 3.3(d) when both phenomena are observed.

3.4.3 Estimating the quality

In the MSSP, the reference set for round k, is composed of the items that were previously selected in round $k-1$, some of which may not be currently available. Set $\hat{q_k}$ to be the estimate of the true quality of the reference set q_k. Simple estimations can be computed by, for instance:

– comparing the reference set of round k to the candidates of round $k-1$, $k-2$, etc., just before beginning that selection process;



(a) Fixed $\xi_k = 0$

(b) Drift $\xi_k \sim \mathcal{N}(\mu = 2, \sigma = 0)$

(c) Deflection $\xi_i \sim \mathcal{N}(\mu = 0, \sigma = 0.1)$

(d) Drift+deflection $\xi_i \sim \mathcal{N}(\mu_k, \sigma_k)$ s.t. $\mu_{k+1} = \mu_k + 1/T$, $\mu_0 = 0$, and $\sigma_{k+1} = \sigma_k + 0.5/K$, $\sigma_0 = 0.1$, where $K = 10$ is the number of rounds.

FIGURE 3.3: Example of evolution of the score distribution at different rounds represented by different colors $k = \{1, 2, 4, 6, 8, 10\}$

– adjusting the estimation of the quality w.r.t. the current candidates as they appear;

– a combination of the above.

The investigation of this quantity is out of the scope of this work, hence we assume that $\hat{q}_k \approx q_k$. Nevertheless, we believe that it could be promising to adopt advanced statistical machine learning methods that would focus on the estimation of q_k in the multi-round setting.

3.5 Conclusion

We start this manuscript with an attempt to unify the many existing definitions of Sequential Selection Problem, by introducing the formal GSSP that allows for a wide range of settings, as shown with a few examples. Secondly, we use this generalized

notations to build the Warm-starting SSP, a novel setting that addresses the cold-start limitations of standard settings, and for which we provide an offline analysis of the initial state. Finally, we extend this warm-start option to a multi-round context, where each round constitues a separate WSSP that starts with at hand, i.e. as a warm-start, the final selection of the previous round. The warm-start aspect introduced in the generalized SSP (GSSP) raised some algorithmic questions regarding the online decisions, particularly in a multi-round setting where an ideal algorithm would maintain an optimal selection in the course of multiple rounds, rather than at the scale of a single selection round. For this reason, in the next chapter we propose a novel algorithm, inspired by the standard Secretary Problem algorithm, fully suited to any degree of warm-start considered. In addition, the algorithm is robust to changes in the score distribution, as it was built by considering the rank-based evaluation criterion proposed in Eq. 3.2.

Chapter 4

The Cutoff-based Cost Minimization algorithm

This section presents our novel algorithm for the WSSP problem described in Chapter 3, the *Cutoff-based Cost Minimization* (CCM) strategy. It takes as input a *cutoff value* $c \in \mathbb{N}$ that represents the size of the *learning phase*, i.e. the number of candidates to reject by default, and from which the DM learns valuable information about the overall sample. Note that this strategy merely constitutes a proposed response to the WSSP in its most generic form. Thus, its main goal is to perform sufficiently regardless of the problem parameters. An interested reader might find an algorithm that is closer to the optimal case, although the latter remains unknown to this day [23], provided that they consider extreme study cases.

4.1 Algorithm description

4.1.1 Cutoff-based strategy

Inspired by the original Secretary Problem algorithm, we develop the cutoff-based Cutoff-based Cost Minimization strategy, see Alg. 1. Our motivation for constructing a cutoff-based strategy relies on the following facts: i) the DM should somehow define a value above which a candidate might be accepted, value that needs to be consistent with the candidate sample (and not necessarily with the reference set), hence the need to explore before making any decision, ii) in a finite-horizon settings with limited and constrained budget, the DM should not rush into hiring since decisions are irrevocable, and iii) the learning phase's size should be fined-tuned to the reference set quality.

The CCM strategy is the natural warm-start adaptation of the main sequential selection algorithms in a rank-based setting, also referred to as *ordinal* (in contrast to *cardinal*), for which the DM can only make pairwise comparison between items. Such algorithms typically define two key parameters, 1) the *cutoff value* $c \in \mathbb{N}$ that represents the size of the *learning phase*, i.e. the number of candidates to be rejected by default, and 2) the *acceptance threshold*, or simply *threshold*, learned during the former phase, and that indicates the value to exceed for selection. In the classical Secretary Problem algorithm [39], both of these parameters are fixed, and with the objective to maximize the probability of getting the best candidate, and knowing the number of candidates $n \in \mathbb{N}^*$, the algorithm is optimal. However, in a multi-choice setting ($b > 1$) with the objective to minimize the sum of the selected items

ranks, the strategy should be slightly adjusted. For instance, in [5] the cutoff value is not fixed, but tailored to the budget b, while the threshold remains the same for each candidate. Another no-less efficient approach, described in [8], set a fixed cutoff (same as in the classical Secretary Problem) but rather considers a dynamic candidate-wise threshold. In the CCM algorithm, the cutoff value is fine-tuned to the problem's parameters, while the acceptance threshold dynamic aspect evolves according to the warm-start level of the setting, i.e. according to the resignation number r (see Definition 6). By that, it constitutes a fitted combination of well-functioning algorithms to the novel warm-start setting. The cutoff value being one of the key parameter of CCM algorithm, the policy is written $\pi(c)$ and the regret $\phi(c)$.

4.1.2 Acceptance threshold

Derived from the learning phase, the CCM policy dictates a set of *threshold values* specific to each job position (i.e. specific to each referent that filled them) that candidates need to exceed to be accepted.

In practice, during the selection phase, the initial acceptance threshold for the candidates is set to be the score of the b-th best up to the end of the learning phase. This set, called *updated reference set*, is defined as $\mathbf{S}_c^{\text{upd}} = (S_{(i),c}^{\text{upd}})_{i\leq b}$ where each term belongs to the concatenation of both the referents and the rejected candidates, i.e. the c first candidates, hence $S_{(i),c}^{\text{upd}} \in (\dot{\mathbf{S}}, S_1, ..., S_c)$ s.t. $S_{(1),c}^{\text{upd}} > ... > S_{(b),c}^{\text{upd}}$, where the available referent scores are denoted by $\dot{\mathbf{S}}^{\text{avail}} = (\dot{S}_{(i)})_{i\in I}$, with $I = \{i : \dot{A}_{0,(i)} = 1\}$. The acceptance threshold should be adapted to the scores of the available referents, so that no position gets filled by a worse item. Note that, candidates are rejected by default during the learning phase, hence the acceptance threshold is defined only during the selection phase. In the rest of the paper, the number of candidates accepted up to step j (included) is denoted by $\tilde{A}_j = \sum_{i=1}^{j} A_i$. Under these conditions, the acceptance threshold is defined as follows.

Definition 6. *Step-specific acceptance threshold ($S_j^{\text{acc.}}$): Score value to beat at step $j > c$ of the WSSP when the CCM policy is applied with cutoff value c:*

$$S_j^{\text{acc.}}(c) := \begin{cases} S_{(b),c}^{\text{upd}} & \tilde{A}_j < r + \sum_{j=1}^{c} \mathbb{1}\{S_j \geq S_{(b),c}^{\text{upd}}\}; \\ \dot{S}_{((b-l))}^{\text{avail}} & \textit{otherwise}, \end{cases} \tag{4.1}$$

The second term in the condition is the number of candidates from the learning phase that have been added in the updated reference set.

Remark 5. *Recall that the DM does not observe item scores (Assumption 1), hence in the 'practical threshold', scores represent the DM's pairwise comparison between items seen so far.*

Following the definition of the acceptance threshold, the decision variable is therefore given by:

$$A_j = \mathbb{1}\{j > c\}\, \mathbb{1}\left\{\tilde{A}_j < b\right\}\, \mathbb{1}\{S_j \geq S_j^{\text{acc.}}\}, \tag{4.2}$$

where the second indicator function ensures that no more than b items can be selected. Understandably, job positions are sequentially reassigned by ascending rank order of the remaining referents. The CCM algorithm is fully described in Alg. 1, and an additional visual example is displayed in Fig. 4.1.

Algorithm 1 The Cutoff-based Cost Minimization policy

Input: the number of b jobs, the number of candidates n , the number of resignations r, the reference set scores from best to worst $\dot{\mathbf{S}} = (\dot{S}_{(1)}, ..., \dot{S}_{(b)})$, the initial vector of reference set availability $\dot{\mathbf{A}}_0 = (\dot{A}_{0,(1)}, ..., \dot{A}_{0,(b)})$, and the cutoff value c.

Output: the set of final job assignment $(\dot{\mathbf{A}}_n, \mathbf{A})$

■ ***Learning phase***

$\mathbf{A}_{1,...,c} \leftarrow 0$ // reject by default all c first candidates

$\mathbf{S}_c^{\text{upd}} \leftarrow \text{top_of_rank}(b, (\dot{\mathbf{S}}, S_1, ..., S_c))$ // b-best from $\dot{\mathbf{S}}$ and $(S_1, ..., S_c)$, in descending value order

$n_{\text{rej}} \leftarrow \sum_{j=1}^{c} \mathbb{1}\{S_j > S_{(b),c}^{\text{upd}}\}$ // the nb. of candidates among the c first that beat the... // ...threshold, i.e. here, the worst of the updated reference set

$\dot{\mathbf{S}}^{\text{avail}} \leftarrow (\dot{S}_{(i)})_{i \in I}$ where $I = \{i : \dot{A}_{j,(i)} = 1\}_{1 \le i \le b}$ // initialize the selection with the available... // ...reference set

$l \leftarrow 0$ // the nb. of jobs assigned so far in the selection

■ ***Selection phase***

for $j = c+1$ to n **do**
  if $l < n_{\text{rej}} + r$ **then** // set the acceptance threshold for the j-th candidate (see Definition 6)
    $S_j^{\text{acc.}} = S_{(b),c}^{\text{upd}}$
  else $S_j^{\text{acc.}} = \dot{S}_{((b-l))}^{\text{avail}}$
  end if
  if $l < b$ and $(S_j > S_j^{\text{acc.}}$ or $j - l = n - r + 1)$ **then**
    $A_j \leftarrow 1$
    if $l \ge r$ **then**
      $\dot{A}_{j,(b-l)} \leftarrow 0$ // remove job from a referent
    end if
    $l \leftarrow l + 1$
  else $A_j \leftarrow 0$
  end if
end for

Example. Consider the example of Fig. 4.1 with $n = 8$ candidates (top line) and $b = 4$ job positions to manage. An item with a job position is represented by a blue circle and a rejected candidate by a red circle. Say the learning phase is of size 2, i.e. $c = 2$. The scheme reads from left ($j = 0$) to right ($j = n = 8$), i.e. each column lists the items seen so far ordered by absolute ranks (rank 1 is the best). Initially, the candidates with absolute ranks $\dot{\mathbf{X}}^{\text{avail}} = (2, 8, 10)$ are available, whereas the referent with rank 4 resigned. Note that, in terms of ranks, the condition $S_j > S_j^{\text{acc.}}$, corresponds to the rank-based condition $X_j < R(S_j^{\text{acc.}})$. Recall that the DM proceeds by pairwise comparison, i.e. instead of observing the absolute ranks $(2, 8, 10)$ she initially observes relative ranks $(1, 3, 4)$, and update her knowledge with each incoming candidate. According to the CCM algorithm, the updated reference set is the set of the $b = 4$ best items seen at the end the learning phase ($j = c = 2$). Therefore the absolute ranks of the updated reference set are $R(\mathbf{S}_2^{\text{upd}}) = (2, 3, 4, 8)$, and hence the initial rank-based acceptance threshold for the third arriving candidate is $R(S_3^{\text{acc.}}) = 8$. Again, the DM does not know this absolute rank at this stage, all she knows is that a candidate gets accepted if he compares favorably to the item associated with absolute ranks 8.

The first candidate of the selection phase (with absolute rank 9) is worse than the item associated with absolute ranks 8: he is rejected, i.e. $A_3 = 0$. The following candidate with absolute rank $X_4 = 7$ beats the acceptance threshold this time, and takes the job that used to be assigned to the referent with rank 4, who initially resigned. The 5-th candidate is also accepted since $X_5 = 6$, and takes the job from the referent with rank 10. When the 7-th candidate gets accepted, the rank-based

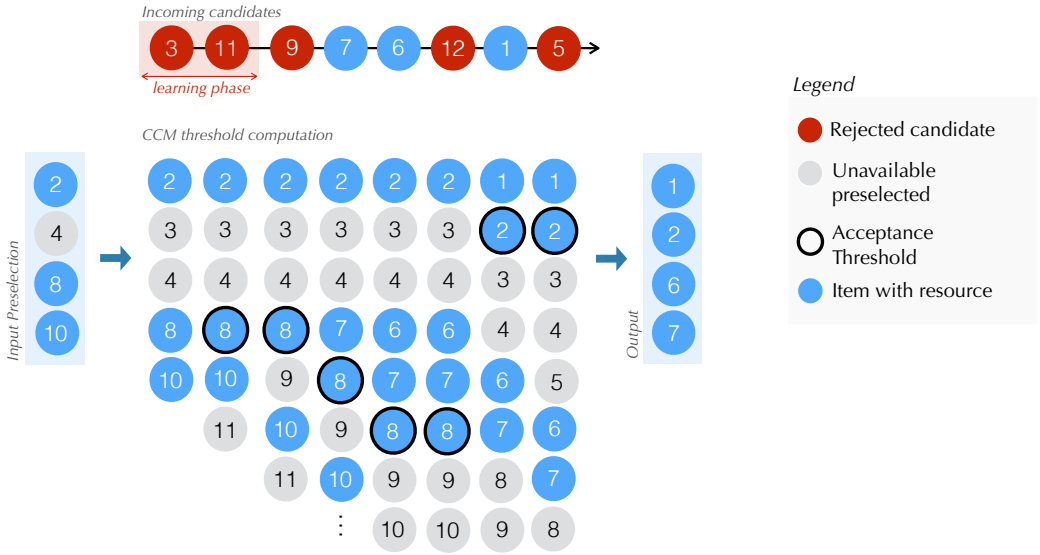


FIGURE 4.1: A step-by-step deployment of the CCM policy for the WSSP. The indicated candidate ranks refer to the true ranking of all twelve items, referents and candidates. The top line gives the sequence of incoming candidates. The scheme reads from left ($j=0$) to right ($j=n=8$), i.e. each column lists the items seen so far order by absolute ranks (rank 1 being the best).

acceptance threshold becomes $R(S_8^{\text{acc.}})=2$, which is the rank of the worst (and here, only) available referent that has not been replaced yet. In fact, it would be clearly suboptimal to leave the acceptance threshold to the item associated to rank 8, since it would mean accepting the last candidate with rank 5 and therefore replace referent with rank 2. In the end, items with absolute ranks $(1,2,6,7)$ have a job positions, and only at this stage can the DM knows how well she performed, since relative and absolute ranks are the same.

Remark 6. *Due to the finite horizon, the DM might select candidates* by necessity*, regardless their quality. This may occur in order to prevent having vacant positions in the output when the very end of the sequence is reached.*

4.2 Optimal CCM

In this section we provide an approximation of the optimal parameters of the CCM algorithm, without claiming that it is the best achievable, but with strong empirical guarantees that it is very close to the true parameters. The optimal CCM uses the $c=c^*$ that minimizes the expected regret.

4.2.1 Optimal CCM for $q=1/2$

Analytical study

Let us now focus on the CCM algorithm and first consider that, on average, referents have a medium quality i.e. $q=1/2$. Indeed, the analytical computation of the main variables of the problem is more challenging when $q\neq 1/2$. However, in Sec. 4.2.2)

we present the *translation method* that 'translates' any setting of arbitrary q to one where $q=1/2$, for which we have the following analytical results.In the analysis, the rank-based acceptance threshold $R(S_j^{\text{acc.}})$ is replaced by its expectation, $\gamma_j := \mathbb{E}[R(S_j^{\text{acc.}})]$. All proofs of this subsection that are not in the main text can be found in Sec. Technical proofs at the end of this chapter.

Lemma 1. *Let a WSSP with n candidates, and a reference set of size b. Using Eq. 4.2, a candidate is accepted if his rank beats the rank-based threshold, $\gamma_j = \mathbb{E}[R(S_j^{\text{acc.}})]$, and less than b candidates have been accepted. The probability for the number of accepted candidates at step j to be smaller than b, $g_{c,j}(b) := \mathbb{P}(\tilde{A}_{j-1} < b)$, is such that:*

$$\left| \mathbb{P}(\tilde{A}_{j-1} < b) - \sum_{i=0}^{b-1} \frac{\lambda_{j-1}^i e^{-\lambda_{j-1}}}{i!} \right| < 2 \sum_{i=1}^{j} p_i^2, \quad \forall\, c+1 \le j \le n, \tag{4.3}$$

where $p_j := \mathbb{P}(X_j < \gamma_j) = \frac{\gamma_j - 1}{n+b}$ and $\lambda_{j-1} = \sum_{i=c+1}^{j-1} p_i$. The proof uses the fact that, the random variable $\tilde{Z}_j$ in $\tilde{A}_j = \min(\tilde{Z}_j, b)$ follows approximately a Poisson distribution with parameter λ_j, and the bound is a straightforward application of [60].

Theorem 1. *Applying the CCM algorithm with parameter c as cutoff value, given that r referents resigned, and using Lemma 1, the WSSP exhibits the following features:*

- *The expected rank-based acceptance threshold for candidate j is given by $\gamma_j := \mathbb{E}[R(S_j^{\text{acc.}})]$ s.t.:*

$$\gamma_j = \gamma g_{c,j}(\Delta) + \frac{\gamma_0(b+1)}{b(b-r+1)} \left(b - \sum_{i=1}^{j-1} \frac{\gamma_i - 1}{n+b} g_{c,i}(b) \right) (1 - g_{c,j}(\Delta)), \tag{4.4}$$

  where $\gamma := \mathbb{E}[R(S_{(b),c}^{\text{upd}})] = \frac{b(b+n)}{b+c}$, $\Delta = r + c\frac{\gamma-1}{n+b}$ and γ_0 is given in Proposition 1.

- *The expected number of new hires at the end of the selection $\tilde{A}_n \le b$:*

$$\mathbb{E}[\tilde{A}_n] = \sum_{j=1}^{n} \frac{\gamma_j - 1}{n+b} g_{c,j}(b) \tag{4.5}$$

- *The expected regret function to minimize, i.e. expected average rank of the selected items:*

$$\mathbb{E}[\phi(c)] = \frac{1}{(n+b)} \sum_{j=c+1}^{n} g_{c,j}(b) \frac{\gamma_j(\gamma_j - 1)}{2} + \frac{\gamma_0(b+1)}{2b(b-r+1)} (b - \mathbb{E}[\tilde{A}_n])(b+1-\mathbb{E}[\tilde{A}_n]) - \mathbb{E}[\phi_{\text{off}}], \tag{4.6}$$

  where $\mathbb{E}[\phi_{\text{off}}]$ is the expected minimal offline loss defined in Proposition 2.

Eq. 4.6 holds a good approximation of the expected regret of WSSP when $q=1/2$, see Fig. 4.2. Recall that we want to find the optimal cutoff value $c^* = \underset{c}{\text{argmin}}\ \mathbb{E}[\phi(c)]$ which is equivalent to finding c^* s.t. $\frac{\partial}{\partial c}\mathbb{E}[\phi(c)]|_{c=c^*} = 0$. Unfortunately, this equation is analytically intractable unless approximations or restrictive assumptions are made,

however we can easily spot c^* numerically by tracking the lowest point of the curve $\mathbb{E}[\phi(c)]$ using Eq. 4.6, $\forall b, \forall n$, and store the results in $c^*(b,r,n)$.

Remark 7. *Note that in practice $\tilde{A}_n$ is actually equal to $\max(\tilde{A}_n, r)$ to avoid empty positions at the end of the selection. An approximation of $\mathbb{E}[\max(\tilde{A}_n, r)]$ can be found in the Appendix, as well as an empirical verification.*

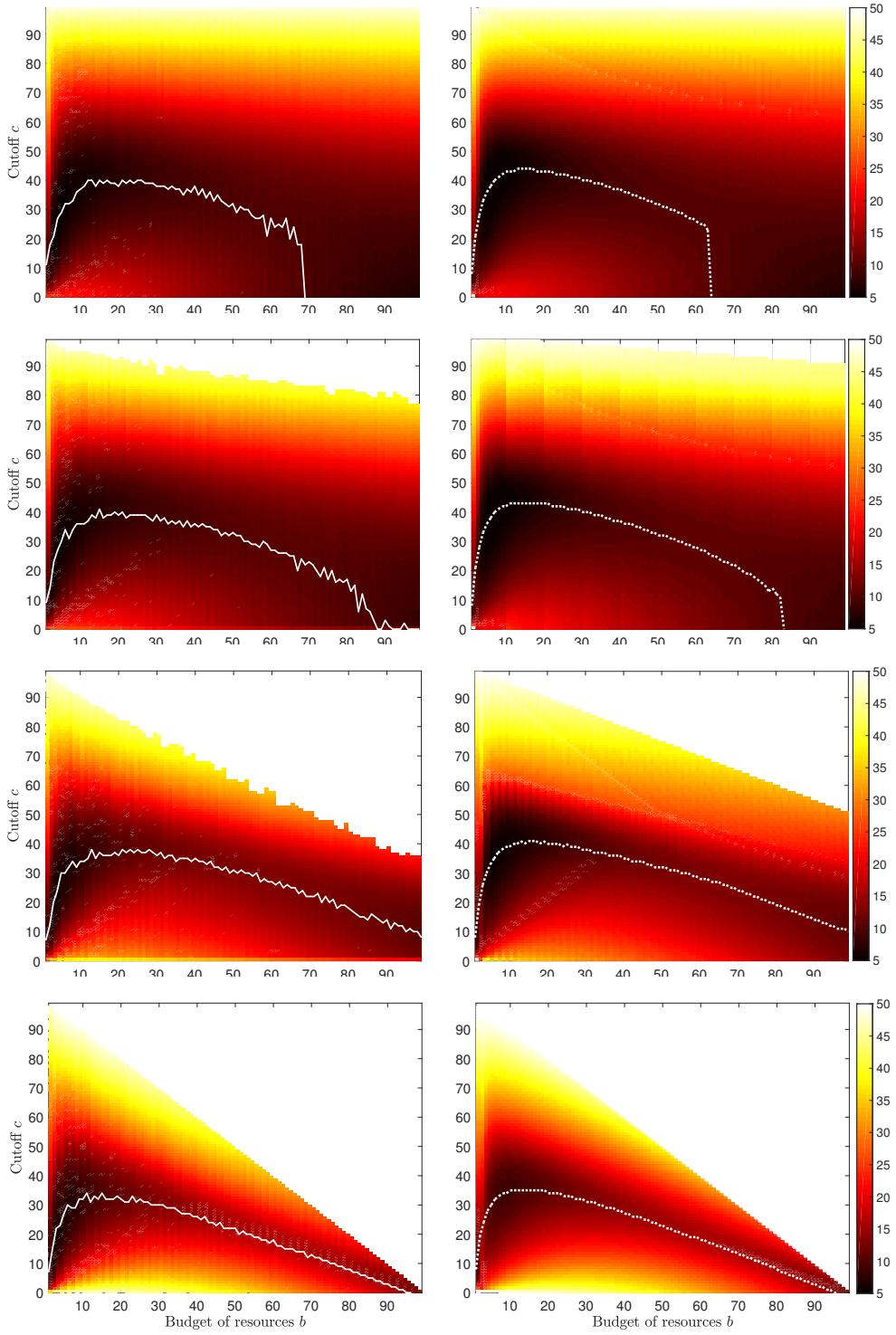
Example. Imagine a WSSP instance where $n = 100$ candidates are going to be sequentially interviewed. The DM handles $b = 5$ job positions, each of them already filled by available referents (i.e. $r = 0$) of average quality $q = 0.75$ w.r.t. to the candidates next to come. Using Theorem 1, the length of the learning phase is $c^* = \lfloor 38.36 \rfloor = 38$, the expected average rank of the selected items is $\mathbb{E}[\phi(c^*)]/5 = 2.60$, and the expected number of accepted candidates is $\mathbb{E}[\tilde{A}_n] = 0.997$. The latter is low since the initial quality is quite good, hence the DM expect to fire only his worse referent (available referent). Now, with the same setup of WSSP parameters except with full resignations, i.e. $r = b$, we get $c^* = \lfloor 28.27 \rfloor = 28$, $\mathbb{E}[\phi(c^*)]/5 = 3.40$ and $\mathbb{E}[\tilde{A}_n] = 5$, which is coherent with the fact that all positions are initially empty. The length of the learning phase is reduced compared that of the previous example, implying a less competitive acceptance threshold. Justifiably, the DM is less demanding on the quality of the accepted items, to avoid the risk of having to select last incoming candidates by default, called a *failure* (see Sec. 4.3).

Empirical verification

In order to guarantee the accuracy of our analytical approximation $\mathbb{E}[\phi(c)]$ in Eq. 4.6, we simulate each WSSP scenario 10000 times, i.e. for a fixed number of candidates $n = 100$ and a fixed reference set quality $q = 1/2$. The top row of Fig. 4.2 displays a heatmap of the average empirical regret (simulated) w.r.t. the number of jobs b (x-axis) and the value of the cutoff c (y-axis). The white plain line in each heatmap follows the path of the lowest simulated value of the heatmap, referred to as $c^*_{\text{sim}}(b) = c^*_{\text{sim}}$. These plots should be put in comparison with those in the bottom row which show the heatmaps of the expected regret according to our analysis. The white dashed line follows again the path of the lowest heatmap value, which we denote as $c^*(b) = c^*$. From Fig. 4.2, it becomes clear that the law of large number complies with the lemmas and propositions of Sec. 4.2 which are consistent in these experiments. The results are then stored in a table of optimal cutoffs, $c^*(b,r,n)$ for $q = 1/2$, of dimension 3 each respectively corresponding to the number job positions, resignations and candidates. An overview is presented in Tab. 4.1.

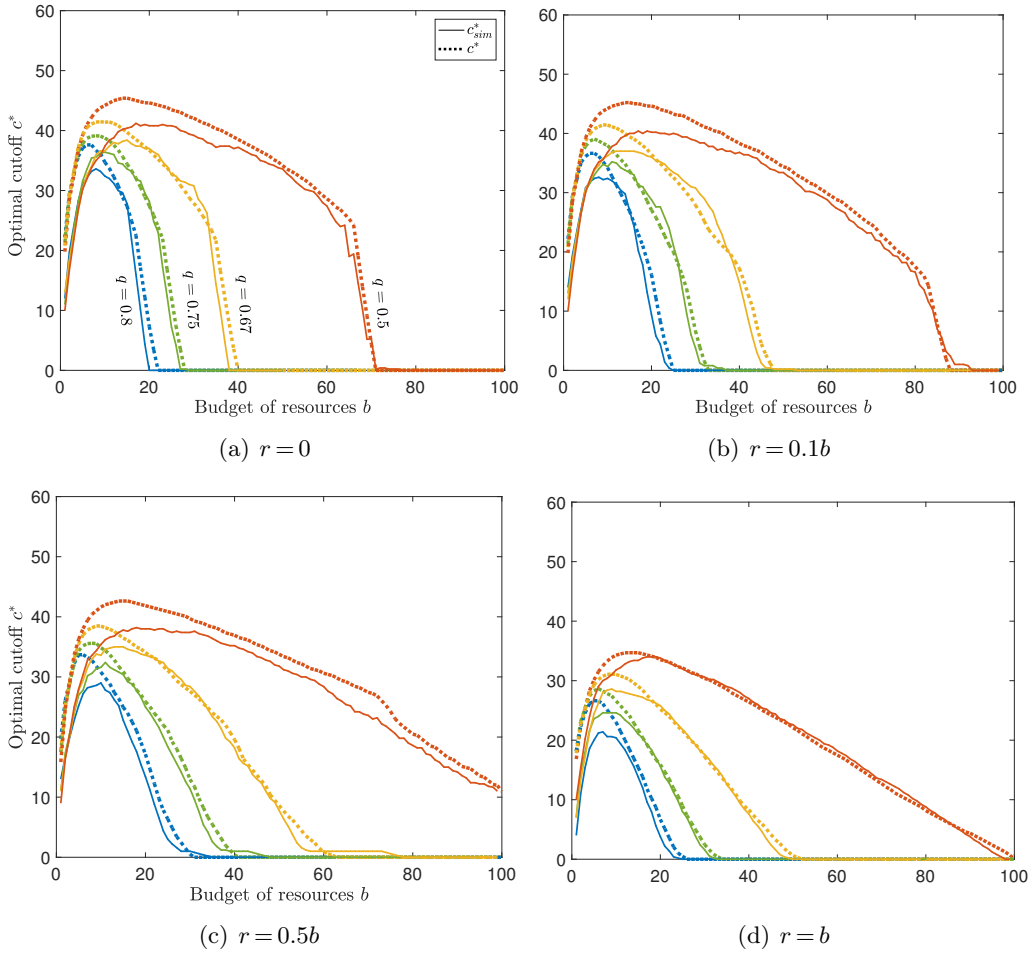
	$b=1$	$b=5$	$b=10$	$b=15$	$b=20$	$b=25$	$b=30$	$b=35$	$b=40$	$b=45$	$b=50$
$r=0$	19	39	44	45	45	44	42	41	39	36	33
$r=0.25b$	15	38	43	44	43	42	41	40	39	37	35
$r=0.50b$	14	36	41	42	42	41	39	38	36	35	33
$r=0.75b$	14	34	39	39	38	37	36	34	32	30	28
$r=b$	13	30	34	34	33	32	30	28	26	23	21

TABLE 4.1: Optimal cutoff $c^*(b,r)$ for $n = 100$ candidates and an initial quality $q = 1/2$.



(a) Empirical results (b) Analytical results

FIGURE 4.2: Comparative heatmaps for the average empirical regret (left column) and the expected regret (right column) derived from Theorem 1 over different resignation numbers $r = \{0, 0.1b, 0.5b, b\}$ from top to bottom line, and all for reference set quality $q = 1/2$. In each case, the heatmap of the regret is presented over the parametrization of the cutoff value c and the budget of resources (jobs), b.



(a) $r = 0$ (b) $r = 0.1b$

(c) $r = 0.5b$ (d) $r = b$

FIGURE 4.3: The optimal cutoff w.r.t. the number of jobs b (x-axis), according to the simulations in plain lines and to our analytical approximation (see Eq. 3.2) in dashed lines for different values of the relative quality of the reference set $q = \{\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}\}$, for $n = 100$ candidates.

4.2.2 Optimal CCM for arbitrary q

The translation method

In Sec. 4.2, we derived an analytical expression for $\mathbb{E}[\phi(c)]$ given a relative quality of the reference set $q = 1/2$. However, when $q \neq 1/2$, the analytical computation of the WSSP's main variables is highly complex. We introduce a rather simple trick to efficiently overcome this difficulty. More specifically, we provide a way translate any setting of arbitrary q to a γ_0-*similar setting* where the quality of the reference set is set to be $q = 1/2$ and for which we can use the results presented in Sec. 4.2. We introduce a notion of *similarity* between two different settings' reference set γ_0 (see Definition 7) and come up with what we call as the *translation method* described below (see Proposition 3).

Definition 7. *γ_0-similarity: Suppose each WSSP instance, denoted by $WSSP_x$, starts with b_x jobs positions filled with the available referents $\dot{\mathbf{X}}_x^{\text{avail}}$, and thereafter interviews n_x candidates using CCM (see Alg. 1) with the optimal cutoff value c_x^*.*

Algorithm 2 Translation method between two WSSP settings

Input: the WSSP setting of interest, WSSP_t (subscript t for '*target*' and, below, s for 'source'), and its main parameters: n_t candidates, b_t jobs, a reference set of a relative rank-based quality q_t from which r_t resigned, and the vector $c^*(b_t, n)$ with the optimal cutoffs for any sequence length n, for $q = 1/2$, as described in Theorem 1.
Output: the optimal cutoff c_t^* for the WSSP_t setting.

■ ***Find a γ_0-similar setting to WSSP_t, let that be WSSP_s***

1: Require $b_s = b_t = b$, as in Definition 7
2: Impose $q_s = 1/2$
3: Compute $n_s = \lfloor (n_t + b - 1)\frac{1-q_t}{1-q_s} - b + 1 \rfloor$, as suggested by Proposition 3

■ ***Translate the setting WSSP_s to WSSP_t***

4: Find the best cutoff value $c_s^* = c^*(b, n_s)$ from the input vector
5: Compute $c_t^* = \lfloor c_s^* \frac{n_t+b}{n_s+b} \rfloor$, according to Proposition 3

Then, the settings of two instances, WSSP_x and WSSP_y, are said to be γ_0-similar if their reference sets (even those unavailable) have the same ranks w.r.t. the rest of the sample, regardless n_x and n_y:

$$\text{WSSP}_x \equiv_{(\gamma_0)} \text{WSSP}_y \quad \text{if} \quad \begin{cases} b_x = b_y, \\ r_x = r_y, \\ \dot{\mathbf{X}}_x = \dot{\mathbf{X}}_y \Rightarrow \gamma_{0;x} = \gamma_{0;y}. \end{cases} \tag{4.7}$$

Proposition 3. *Translation method: If $\text{WSSP}_x \equiv_{(\gamma_0)} \text{WSSP}_y$, then:*

$$c_y^* = c_x^* \frac{n_y + b}{n_x + b} \quad \text{and} \quad n_x = (n_y + b - 1)\frac{1 - q_y}{1 - q_x} - b + 1. \tag{4.8}$$

Alg. 2 describes the overall translation algorithm. The plots in Fig. 4.3 indicate the large agreement between the optimal cutoffs computed by our analytical translation method and the cutoff empirically computed through simulations.

Examples. Let us illustrate the translation method with two examples.
■ Imagine the DM deals with WSSP_t, where no referent resigned, with $n_t = 100$, $b_t = 15$, $q_t = 0.8$, and she is interested in knowing c_t^*. One possible γ_0-similar setting, WSSP_s, has the following features $q_s = 1/2$ and $b_s = b_t = b = 15$. Using Proposition 3 we get $n_s = \lfloor (n_t + b - 1)\frac{1-q_t}{1-q_s} - b + 1 \rfloor = \lfloor \frac{114 \cdot 0.2}{0.5} - 14 \rfloor = 31$; then using Theorem 1 we compute c_s^* numerically for $n_s = 31$ (which is feasible as long as $q_s = 1/2$) and get $c_s^* = 9$. Finally we obtain $c_t^* = \lfloor c_s^* \frac{n_t+b}{n_s+b} \rfloor = 22$; the DM rejects the first $\frac{c_t^*}{n_t}$ candidates, that is 22% of the total sample, before starting to select.

■ Another example is made using a WSSP_t setting where $n_t = 100$, $b_t = 1$, $r_{,1} = 0$, and $q_t = 0.7$. A γ_0-similar setting, WSSP_s, is when $q_s = 1/2$, and $b_s = b_t = b = 1$. Using Proposition 3 we get $n_s = \lfloor \frac{100 \cdot 0.3}{0.5} \rfloor = 60$; then using our numerical result we get $c_s^* = 7$. Finally, $c_t^* = \lfloor c_s^* \frac{n_t+b}{n_s+b} \rfloor = 11$. In Sec. 4.2.2 we show that $c^* = \sqrt{\frac{n+1}{2\left(1-q+\frac{1}{n}\right)}} - 1$, $\forall q$, which in this case, reads $\sqrt{\frac{n_t+1}{2\left(1-q_t+\frac{1}{n_t}\right)}} - 1 = 11$. This is consistent with the result of the presented translation method between WSSP settings.

Empirical study. For a fixed quality q, it is worth pointing out that $c^*(b)$ is not a monotonic function but rather has two distinct regimes indicated by the $\text{sign}(\frac{\partial c^*}{\partial b})$.

This can be better explained as the following trade-off. Suppose fixed n and r (see Fig. 4.3) and that we start with $b=1$: increasing b would mean more jobs to assign, hence, the DM should very quickly (w.r.t. budget increase) increase the length of the rejection phase to make sure that she learns sufficiently before taking the many decisions (regime $\frac{\partial c^*}{\partial b} \geq 0$). From a point and further, though, increasing b would also mean a) to have a less competitive threshold (which depends on the quality of the worst current referents), b) that the whole process becomes less selective as less and less candidates need to be rejected, and c) to have a higher expected number of resignations (if $r>0$), which makes the exploration for the DM less safe. Hence, the DM should start shortening her learning phase (regime $\frac{\partial c^*}{\partial b} < 0$). The optimal cutoff values get lower as the number of resignations r increases (see the curves across the plots of Fig. 4.3), as well as with the decrease of reference set quality q (see the compared curves in each plot of Fig. 4.3).

Relation between optimal CCM and q

Competitive ratio

As stated earlier, the optimal online selection strategy in the rank-based setting remains – to this day – unknown for a reasonable number of candidates to evaluate. In such cases, any proposed online strategy is rather compared to the offline strategy that selects the b-best available items. A standard approach in the SSP literature consists of computing the *competitive ratio* α that quantifies the distance between an online algorithm's payoff ALG and that of the associated offline algorithm OPT, s.t.:

$$\mathbb{E}[\text{ALG}] \geq \alpha \, \text{OPT}. \tag{4.9}$$

Proposition 4. *In the full warm-start setting, i.e. for $r=0$ resignations and a preselection of size b with average rank γ_0, the competitive ratio α of the CCM algorithm is given by:*

$$\alpha = \frac{c^*}{n+b} \ln\left(\frac{n+b}{c^*}\right)\left(1-\frac{b^2}{2\gamma_0}\right)+\frac{b^2}{2\gamma_0}, \tag{4.10}$$

and as such, $\alpha \geq 1/e$. This means that the algorithm is at least $1/e$-competitive.

Observe that, as γ_0 gets higher, which means that the referents' quality deteriorates, the standard cold-start setting is retrieved, as well as the $1/e$ competitive ratio (retrieved by replacing c^* by $(n+b)/e$).

4.2.3 Closed-form results for $b=1$

Imagine the following setup of WSSP parameters: the DM handles a single job position ($b=1$) already occupied by a referent ($r=0$), with a given rank $\gamma_0 \in \{1,...,n+1\}$ compared to candidates next to come, and that is known upfront. From Definition 4, the DM also knows his quality $q = 1-\frac{\gamma_0-1}{n} \in]0,1[$. She then deploys the aforementioned CCM algorithm and by that must decide the length of the learning size, i.e. the value of c. In this case, the regret in Eq. 3.2 becomes $\phi(c) = \dot{X}_{(1)}\dot{A}_{n,(1)} + \mathbf{X}^\intercal \mathbf{A} - 1$; the last term being the best rank an offline strategy could achieve, here the rank 1. Results from Theorem 1 are adapted to this case, and lead to the following Theorem 2.

Theorem 2. *Let one job position be initially occupied by a referent of given rank $\gamma_0 \in \{1, ...n\}$ compared to the $n \in \mathbb{N}^*$ candidates next to come. Applying the CCM policy with a learning phase of size $c \leq n$, the WSSP exhibits the following features:*

- *Expected rank of the best item (referent or candidate) seen up to the end of the learning phase:*
$$\gamma := \mathbb{E}[R(S^{\text{upd}}_{(1),c})] = \min\left(\gamma_0, \frac{n+1}{c+1}\right). \tag{4.11}$$

- *Expected number of new hires at the end of the selection $\tilde{A}_n$:*
$$\mathbb{E}[\tilde{A}_n] = 1 - \left(1 - \frac{\gamma - 1}{n+1}\right)^{n-c}. \tag{4.12}$$

- *Expected regret, i.e. the expected rank of the item that gets the job position:*
$$\mathbb{E}[\phi(c)] = \frac{\gamma}{2} - \frac{\gamma}{2}\left(1 - \frac{\gamma - 1}{n}\right)^{n-c} + \gamma_0 \frac{c}{n} - 1. \tag{4.13}$$

- *Optimal cutoff value that minimizes the expected regret:*
$$c^* = \begin{cases} \sqrt{\frac{n+1}{2\left(\frac{\gamma_0}{n}\right)}} - 1 & \gamma_0 \geq 3; \\ 0 & \gamma_0 = \{1, 2\}. \end{cases} \tag{4.14}$$

  i.e. $c^* = \underset{c}{argmin}\ \mathbb{E}[\phi(c)]$.

Note that, with $q = 1/2$ and $n \gg 1$ this gives $c^* = O(\sqrt{n})$, see [13]. The proof of the theorem is detailed in Sec. Technical proofs at the end of this chapter. Also, when $\gamma_0 = \{1, 2\}$ the problem is trivial, either $\gamma_0 = 1$ and the DM already has the best item at hand or $\gamma_0 = 2$ and the DM has the second best, hence should hire any candidate that beats him to get the lowest possible regret.

Example. Let $n = 100$ be the number of candidates, and a referent that occupies the job position with rank $\gamma_0 = 25$. First, his quality is of $q = 1 - \frac{25-1}{n} = 0.76$, which is quite better than average. Second, we compute the optimal cutoff by $c^* = \left\lfloor \sqrt{\frac{100+1}{2\left(\frac{25}{100}\right)}} - 1 \right\rfloor = \lfloor 13.21 \rfloor = 13$; hence the DM should reject the first 13 candidates before starting to select. Following the CCM policy's procedure, the initial acceptance threshold is set to be $\lfloor \min(25, \frac{101}{14}) \rfloor = 7$; i.e. no candidate with absolute rank worse than 7 should be accepted during the process. Using those results, the regret is expected to be $\frac{\gamma}{2} + \gamma_0 \frac{c}{n} - 1 = \frac{7}{2} + 24\frac{13}{100} - 1 = 5$ which is far lower than the initial rank 25 of the referent.

4.3 Adjusted policy: low failures-CCM

4.3.1 Algorithm description

In real-life scenarios the proportion of referents that resign compared to those who stay is often relatively small; therefore, in the presented recruitment context, the

more relevant results of this work concern situations where the number of resignations is small (e.g. $2r \leq b$). However when the latter is quite high (i.e. most job positions are empty), the DM might have to accept last arriving candidate (s) in order to fill vacant positions, this event is called a *failure* (described in Definition 8) and is similar to hiring random candidate (s) which ends up increasing the regret.

Definition 8. *Failure and failure rate (f_j): A failure at step j is the event of accepting a last incoming candidate by default (to fill empty job positions) whose score did not beat its associated threshold $S_j^{\text{acc.}}$, i.e. $f_j = \mathbb{1}\{j - \tilde{A}_{j-1} = n - r + 1\}\mathbb{1}\{S_j < S_j^{\text{acc.}}\}$. The failure rate ρ_f is defined as the sum of the number of failures divided by the number of tests.*

Simulations show that in some settings the failure rate is indeed significant, for instance it reaches $\rho_f = 58\%$ for $b = 20$, $r = b$, and $q = 0.81$. This phenomenon appears due to the high quality of the updated reference set, i.e. the threshold becomes too competitive and hence difficult to beat for most candidates. Our idea to mitigate this effect is to estimate the expected number of accepted candidates at step j, denoted by $\hat{\mu}_j(c,r)$, given that the total number of accepted candidates (i.e. at the end of the selection) is greater or equal to the number of resignations r; formally that is: $\hat{\mu}_j(c,r) := \mathbb{E}[\tilde{A}_j \mid \tilde{A}_n \geq r]$, $\forall j$, i.e. there is no failure (see Proposition 5).

Proposition 5. *The expectation of the number of candidates accepted at step j given that there is no failure is given by:*

$$\mathbb{E}[\tilde{A}_j \mid \tilde{A}_n \geq r] = \frac{1}{1 - g_{c,n+1}(r)} \left(b(1 - g_{c,j+1}(b)) + \sum_{k=0}^{b-1} k(1 - g_{j,n+1}(r-k)) \frac{\lambda_j^k e^{-\lambda_j}}{(k-1)!} \right), \tag{4.15}$$

where $\lambda_j = \sum_{i=c+1}^{j} \frac{\gamma_i - 1}{n+b}$ and $g_{i,j}(x) := \mathbb{P}(\tilde{A}_{j-1} - \tilde{A}_i < x)$ (see Lemma 1).

Proposition 6. *In the special case where $b = 1$ and $r = b = 1$, Proposition 5 becomes:*

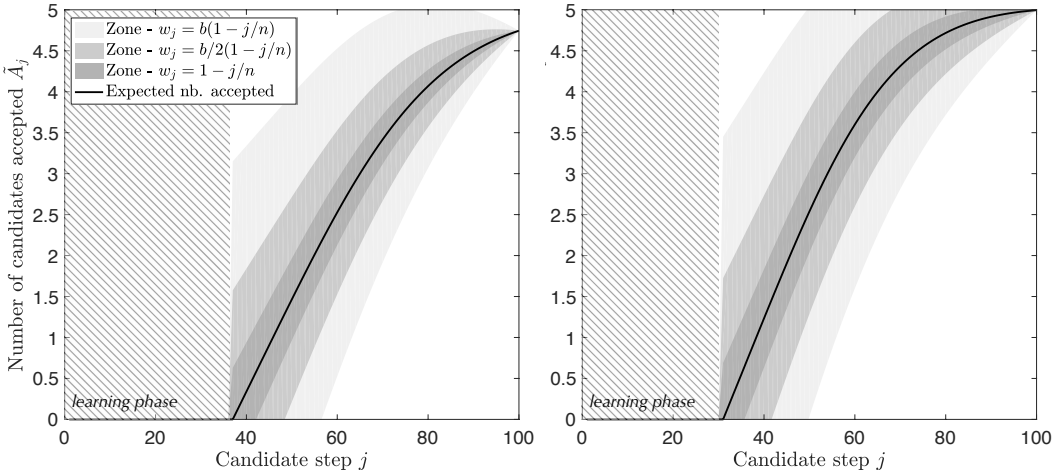
$$\mathbb{E}[\tilde{A}_j \mid \tilde{A}_n = 1] := \hat{\mu}_j = \frac{1 - (1 - \frac{\gamma-1}{n})^{j-c}}{1 - (1 - \frac{\gamma-1}{n})^{n-c}}, \tag{4.16}$$

$$\mathbb{V}[\tilde{A}_j \mid \tilde{A}_n = 1] = \hat{\mu}_j - \hat{\mu}_j^2; \tag{4.17}$$

where $\gamma = \min\left(\gamma_0, \frac{n+1}{c+1}\right)$.

We use Proposition 5 to compute $\hat{\mu}_j(c^*, r)$, $\forall j \in \{1, ..., n\}$, and compare it to the current number of accepted candidates at step j (included), denoted by $\tilde{A}_j$. From this comparison we introduce the notion of *zone* in Definition 9, that we use to adjust the threshold. In Fig. 4.4, that zone is shaded in gray, from light to dark, according to the width of the zone.

Definition 9. *Zone (Z): Interval around the expectation of the number of accepted candidates inside which the threshold γ_j is identical to that of CCM. It is defined between the two curves $\hat{\mu}_j + w_j$ and $\hat{\mu}_j - w_j$ where $w_j = w(j)$ is the function that defines the zone's thickness at step j.*



(a) $r = 0.5b$ (b) $r = b$

FIGURE 4.4: Expectation of the number of candidates accepted so far for $n = 100$ candidates given that no candidate was accepted by default (black line). Number of jobs is $b = 5$ and the reference set quality is $q = 0.5$. The zone (see Definition 9) is gray-shaded with, from light to dark, $w_j = \{b\left(1-\frac{j}{n}\right),\ \frac{b}{2}\left(1-\frac{j}{n}\right),\ 1-\frac{j}{n}\}$.

The threshold of this adjusted algorithm *low failures-CCM* (lf-CCM) is defined as:

$$S^{\hat{\text{acc.}}}{}_j(c) := \begin{cases} S^{\text{upd}}_{(b),c}, & \hat{\mu}_{j-1} - w_{j-1} \leq \tilde{A}_{j-1} \leq \hat{\mu}_{j-1} + w_{j-1} \text{ (in the zone)}, \\ S^{\text{upd}}_{(m+D^+_{j-1}),j} & \tilde{A}_{j-1} < \hat{\mu}_{j-1} - w_{j-1} \text{ (below the zone)}, \\ S^{\text{upd}}_{(m-D^-_{j-1}),j} & \tilde{A}_{j-1} > \hat{\mu}_{j-1} + w_{j-1} \text{ (above the zone)}; \end{cases} \tag{4.18}$$

where $S^{\text{upd}}_{(i),j} \in (\dot{\mathbf{S}}, S_1, ..., S_j)$ is the score of the i-th best seen out of the reference set and up to the j-th candidate s.t. $S^{\text{upd}}_{(1),j} > ... > S^{\text{upd}}_{(b),j}$, and $m \leq b$ is s.t. $S^{\text{upd}}_{(m),j} = S^{\text{upd}}_{(b),c}$. The D^+_{j-1} and D^-_{j-1} functions define how the threshold will change provided that a point $(j, \tilde{A}_j)$ is outside the zone Z. More precisely, when that point lies in the zone, the threshold is constant and equal to $S^{\text{upd}}_{(b),c}$. When it is below (resp., above) the zone, then the threshold for the next candidate is reduced (resp., increased) by $\lfloor D^+_j \rfloor$ (resp., $\lfloor D^-_j \rfloor$) positions from the former, as many times as needed until the point is inside the zone again, and the threshold goes back to the original one (i.e. $S^{\text{upd}}_{(b),c}$) for the next candidate. Finally, $d_j = D^+_j - D^+_{j-1} = D^-_j - D^-_{j-1}$ is the increment in the position each time a point has been above (resp. below) the zone in a row. For simplicity, we assume that an optimal cutoff value c^* for CCM is also an optimal cutoff value for lf-CCM.

4.3.2 Empirical study

Several points can be made from the comparison of CCM and low failures-CCM. Firstly, adjusting the threshold using lf-CCM reduces effectively the number of failures, as well as the average regret (see Tab. 4.2), and hence lf-CCM is indeed more robust to resignations. Concerning the function w_j that gives the thickness of the zone in lf-CCM, after a few trials we set it to the simple enough and efficient (compared to more complex options): $w_j = \frac{b}{2}\left(1-\frac{j}{n}\right)$.

Strategy	**Prob. to resign** r/b	**Failure rate (%)** ρ_f	**Average regret** $\langle\phi\rangle(c^*)$
CCM	0.5	41.4%	6.05
	1	57.8%	7.55
lf-CCM $w_j = \frac{b}{2}\left(1-\frac{j}{n}\right)$, $d_j = 1$	0.5	1.7%	3.38
	1	4.5%	4.02
lf-CCM $w_j = \frac{b}{2}\left(1-\frac{j}{n}\right)$, $d_j = e^{(j-v_1)/v_2}$	0.5	**0.80%**	**3.15**
	1	**2.70%**	**3.67**

TABLE 4.2: We work with $b = 5$ jobs and a quality of reference set $q = 0.9$. For lf-CCM with exponential increment, after empirical tuning, we use $v_1 = 82$ and $v_2 = 14$.

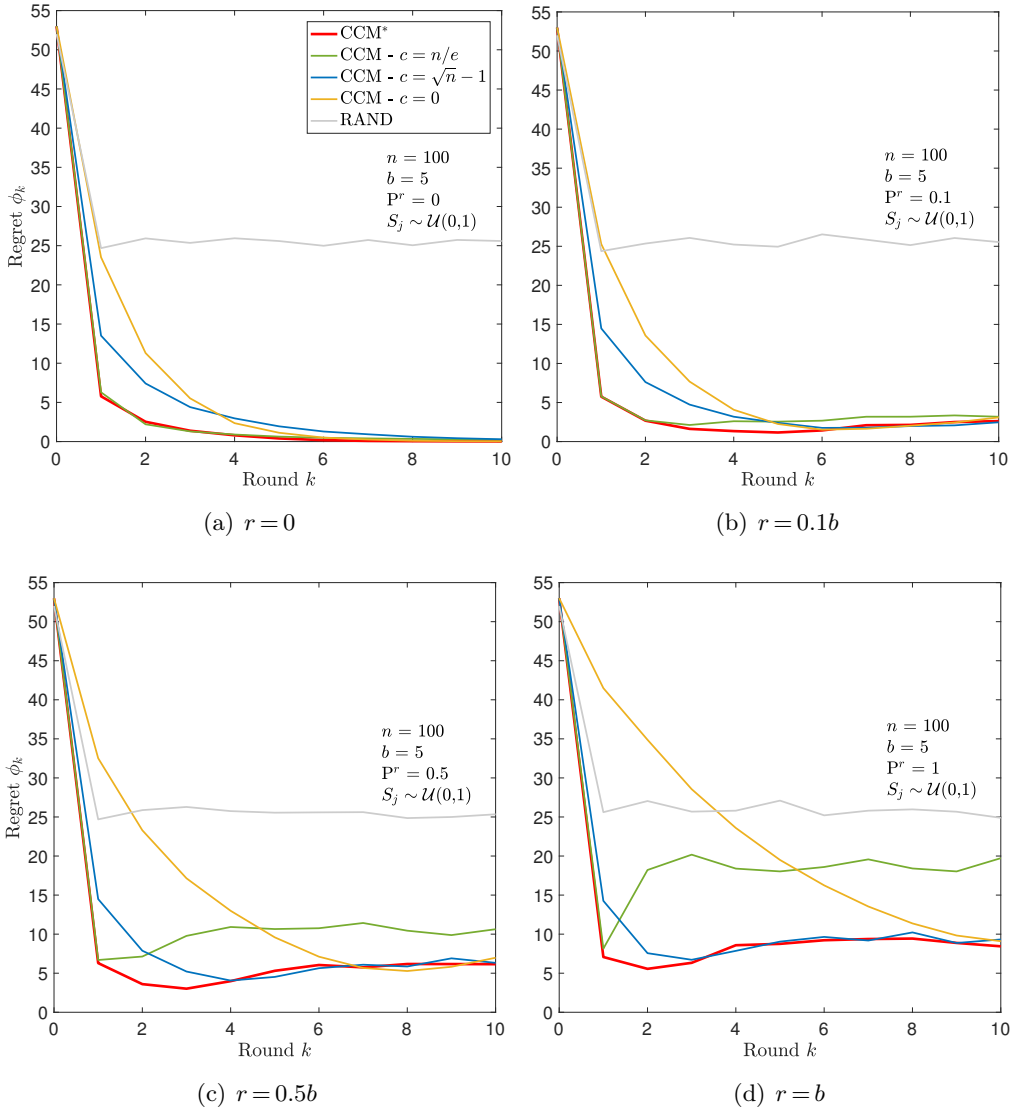
Finally, for D_j, the other parameter of lf-CCM that stipulates how the threshold is changed when a point $(j, \tilde{A}_j)$ is outside the zone, we simulated using a simple function that decreases (respectively increases) the threshold by one position at a time, namely $d_j = 1$, $\forall j \in \{1, ..., n\}$, hence $D_j^+ = \lambda_j^+ - j + 1$. Having a constant increment might not be the best option because it does not quantify how many selection steps (i.e. candidate) are left, therefore we switch for an exponential increment $d_j = e^{(j-v_1)/v_2}$, with v_1 and v_2 empirically computed constants. Its is indeed more efficient at lowering both the regret and the failure rate ρ_f (see Tab. 4.2).

4.4 Simulations in a multi-round setting

4.4.1 The impact of the cutoff value

In the previous section we created the Cutoff-based Cost Minimization that aims at selecting good candidates in a single-round horizon. In this section, we intend to plug it in the multi-round setting (MSSP) introduced in Chapter 3 in order to iteratively improve the DM's selection. For the simulations of this section we use the following parametrization. Firstly, each multi-round simulation considers a population of $|\mathcal{C}| = 1000$ items and for all rounds we set the number of candidates to $n = 100$. Secondly, the resignation probability $\mathrm{P}_k^r = \mathrm{P}^r \in [0, 1]$ is considered to be known in advance by the DM, and is kept constant for every round k and equal for all referents.

Cutoff-choice, number of jobs and score distribution. Fig. 4.5 ($b = 5$) and Fig. 4.6, ($b = 50$) display the average rank-based regret ϕ_k w.r.t. the round number k for different resignation probabilities. We note CCM* the Cutoff-based Cost Minimization algorithm with optimal size of learning phase $c = c^*$. We first observe that, regardless the resignation probability, our proposed cutoff c^* (red curves) outperforms other alternatives heuristics such as the case $c = 0$, hence the importance of tuning the cutoff parameter c. As presented, MSSP allows for referents to quit their job at the beginning of a round, with probability P^r. Notice that, the cutoff $c = n/e$ is a decent alternative to $c = c^*$ when $\mathrm{P}^r = 0$ (see Fig. 4.5(a)), although failing at reducing the regret when $\mathrm{P}^r = 1$ (see Fig. 4.5(c)). Many resignations can occur when the environment changes abruptly (e.g. company's future, changes in the job market, etc.), or when the time-interval between two subsequent rounds is very long and more referents may happen to resign.



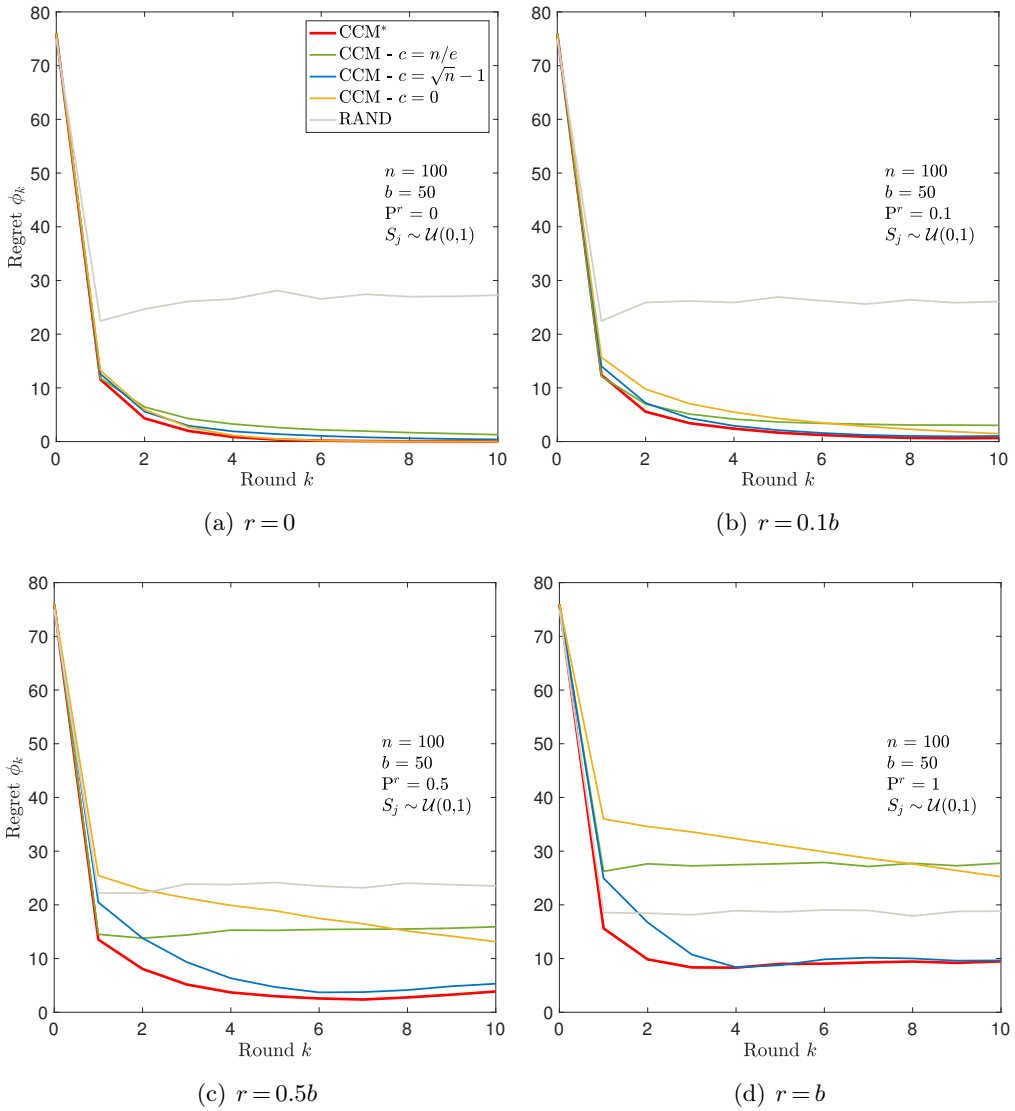
(a) $r = 0$ (b) $r = 0.1b$

(c) $r = 0.5b$ (d) $r = b$

FIGURE 4.5: Average regret ϕ_k w.r.t. the round number k, for $n = 100$ candidates, $b = 5$ jobs positions, and the score distribution is $S_{i,k} \sim \mathcal{U}(0,1)$. Each subfigure corresponds to a different number of resignations $r = \{0, 0.1b, 0.5b, b\}$.

In Fig. 4.7, we use another underlying score distribution for the candidates, namely an exponential distribution Exp(1). We still display the rank-based regret, which shows that the CCM has consistent behavior, and is robust to a change in the considered score distribution.

Resignations. Another observation on Fig. 4.5, Fig. 4.6 and Fig. 4.7 is that CCM seems to struggle to make the regret converge towards zero, and as stated in Sec. 4.3 this effect is a consequence of being forced to select the last candidate(s) in order to assign all vacant jobs (i.e. failure), hence the low failures-CCM. A comparison of CCM and lf-CCM can be found in Fig. 4.8 and illustrates the fact that lf-CCM is more efficient at improving the selection through rounds than CCM, although it requires more adaptation from the DM.



(a) $r = 0$ (b) $r = 0.1b$

(c) $r = 0.5b$ (d) $r = b$

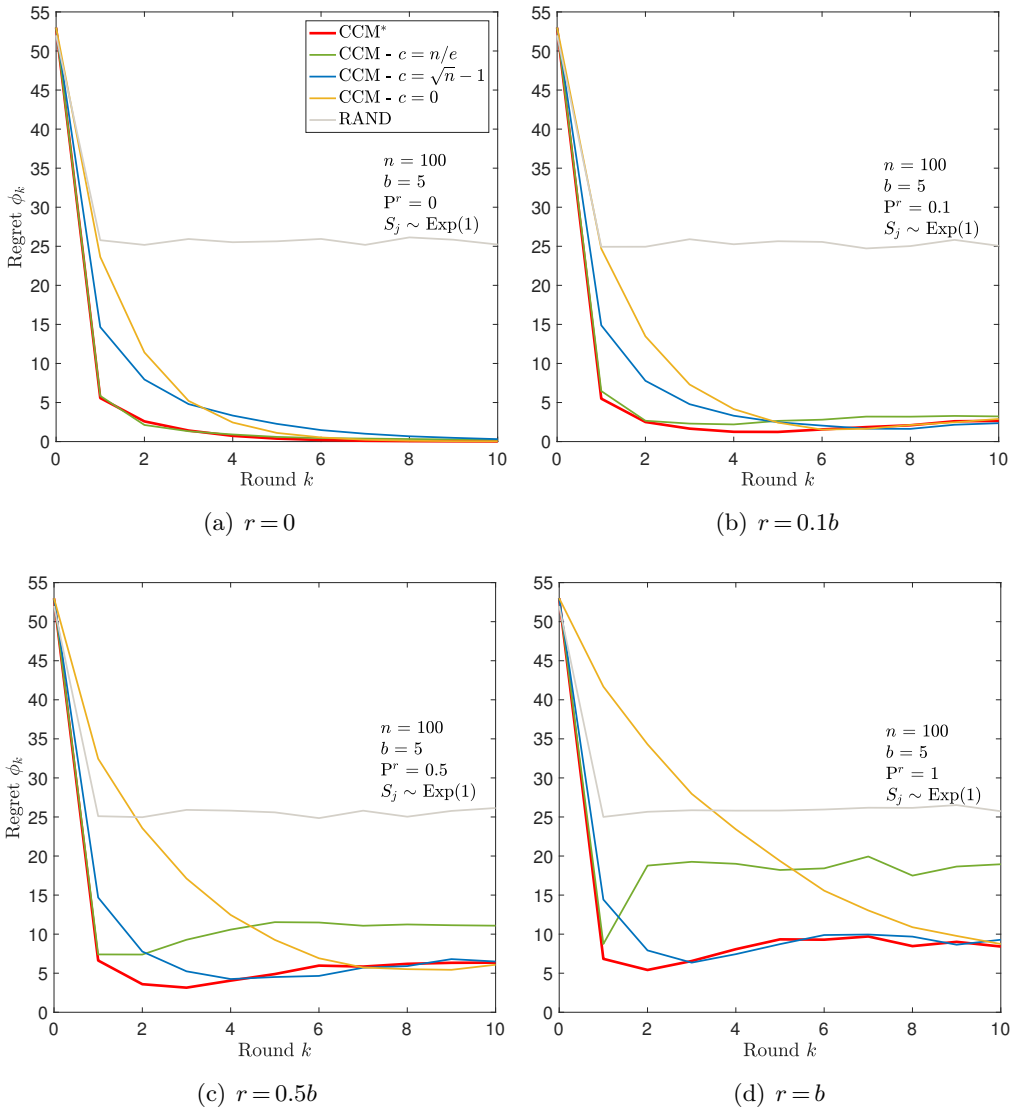
FIGURE 4.6: Average regret ϕ_k w.r.t. the round number k, for $n = 100$ candidates, $b = 50$ jobs positions, and the score distribution is $S_{i,k} \sim \mathcal{U}(0,1)$. Each subfigure corresponds to a different number of resignations $r = \{0, 0.1b, 0.5b, b\}$.

4.4.2 Comparison with a few relevant algorithms from the literature

In Sec. 2.3.2 we presented a few relevant algorithms that are adapted to the multi-choice problem. Here, those algorithms are implemented in a MSSP setting, and compared against the CCM* algorithm. In addition and when the context allows it, the parameters involved are translated according to the notations introduced in this chapter.

– KLEINBERG (2005): The acceptance threshold for the j-th arriving candidate is s.t. $\tau_j = S^{\text{upd}}_{(l),m}$, $\forall j > m$; where $m \sim \text{Binom}(n, 1/2)$ and $l = \lfloor b/2 \rfloor$.

– OPTIMISTIC (2007): The algorithm is cutoff-based, and takes as a cutoff value $c = \lfloor n/e \rfloor$. The acceptance threshold τ_j is dynamic and is equal to the relative rank of the b-th best recorded during the learning phase when no candidate has been accepted yet, i.e. $\tau_j = S^{\text{upd}}_{(b),c}$ for $\tilde{A}_j = 0$, the $b-1$-th best when one candidate



(a) $r = 0$ (b) $r = 0.1b$

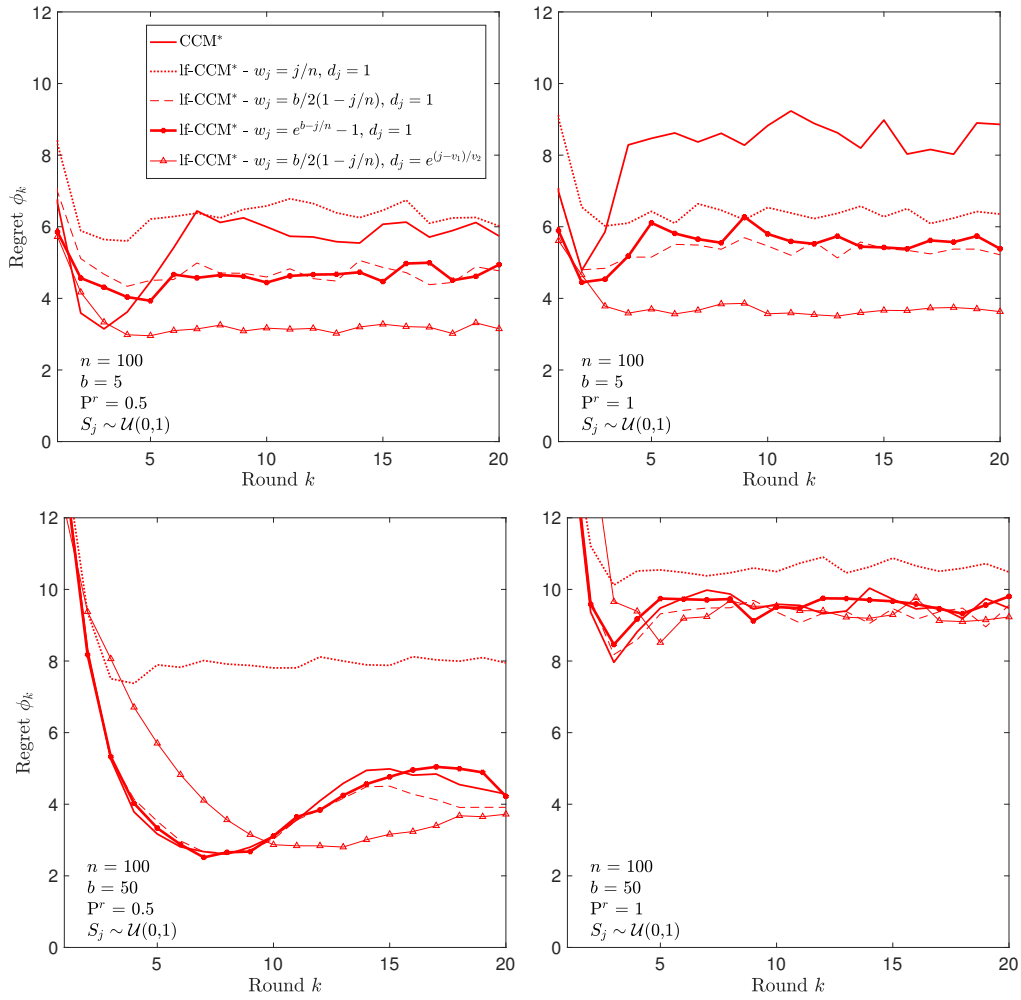
(c) $r = 0.5b$ (d) $r = b$

FIGURE 4.7: Average regret ϕ_k w.r.t. the round number k, for $n = 100$ candidates, $b = 5$ jobs positions, and the score distribution is $S_{i,k} \sim \text{Exp}(1)$. Each subfigure corresponds to a different number of resignations $r = \{0, 0.1b, 0.5b, b\}$.

has been accepted, i.e. $\tau_j = S^{\text{upd}}_{(b-1),c}$ for $\tilde{A}_j = 1$, etc. up to the last candidate accepted.

– SINGLE-REF (2019): The algorithm is cutoff-based, and takes as inputs a cutoff value c and a fixed acceptance threshold τ_j, both empirically optimized according to the number of resources b. For instance, when $b = 5$, $c = 0.2525\, n$ and $\tau_j = S^{\text{upd}}_{(2),c}$, i.e. each candidate should beat the second best recorded during the learning phase, and when $b = 50$, $c = 0.1536\, n$, $\tau_j = S^{\text{upd}}_{(9),c}$. A wide table of these optimized parameters is provided in the paper.

Simulation results. Fig. 4.9 displays the average regret (rank-based) of each algorithm using two different underlying score distributions, namely uniform, i.e. $S_{j,k} \sim \mathcal{U}(0,1), \forall j, \forall k$ (left column) and Gaussian, i.e. $S_{j,k} \sim N(0,1), \forall j, \forall k$ (right column). We also add Fig. 4.10 for a score-based version of the regret for the same simulations. The CCM* algorithm (red curve) outperforms other existing strategies



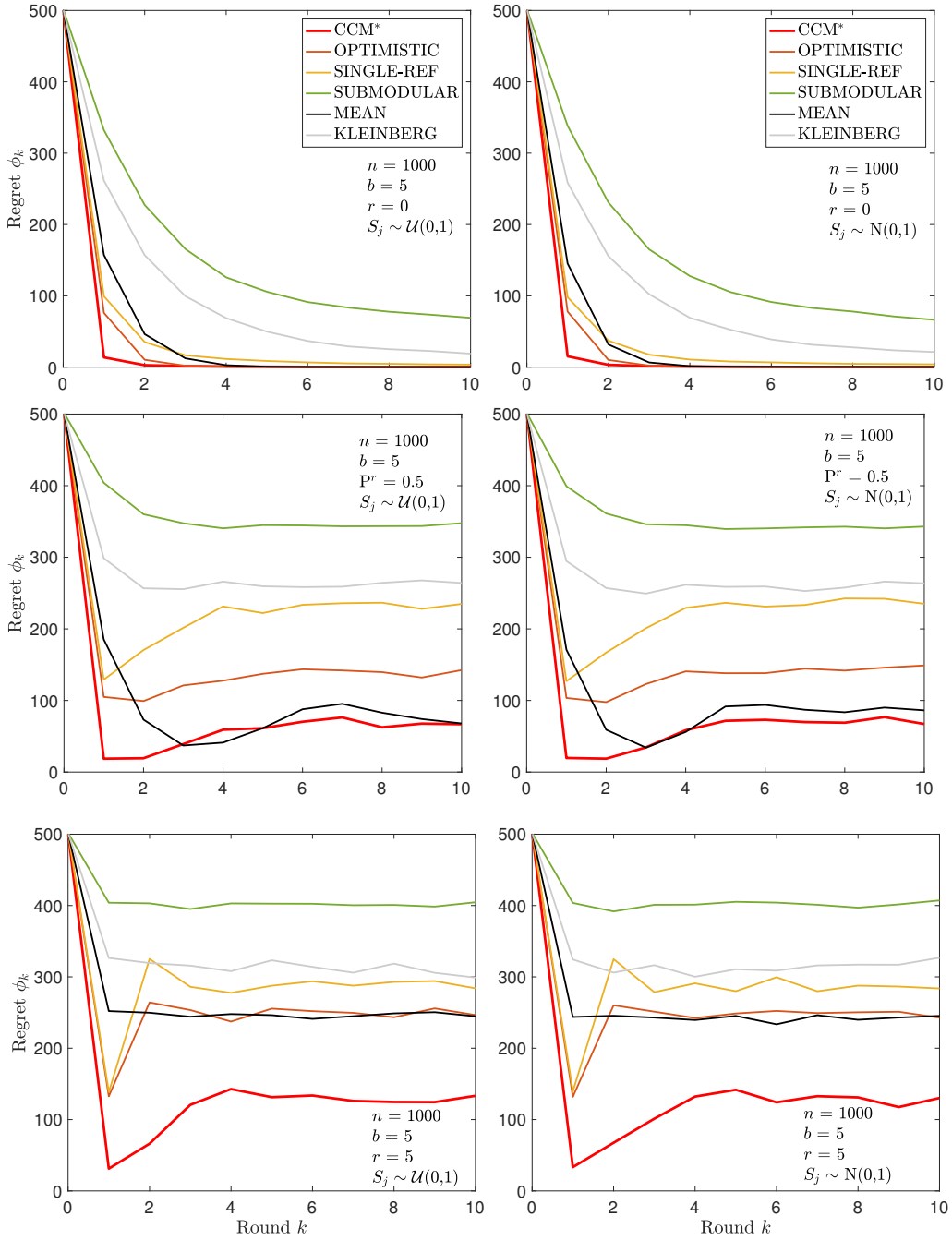
(c) $P^r = 0.5$ (d) $P^r = 1$

FIGURE 4.8: Average regret ϕ_k w.r.t. the round number k, for $n = 100$ candidates and $b = 5$ job positions. The score distribution is $S_{i,k} \sim \mathrm{U}(0,1)$. The plain red curve uses CCM when all others use lf-CCM. The parameters for the curve with triangle markers are $v_1 = 82$ and $v_2 = 14$.

in the literature in this multi-round setting. Note that, the OPTIMISTIC strategy (for which $c = n/e$) is a decent alternative to CCM* when $r = 0$, i.e. $P^r = 0$ (see Fig. 4.9(a)), although failing at reducing the regret when $r = b$, i.e. $P^r = 1$ (see Fig. 4.9(c)). As a final remark, observe that in this multi-round setting a dynamic threshold, as in OPTIMISTIC (orange curve), always performs better than the fixed threshold used in SINGLE-REF (yellow curve).

4.5 Conclusion

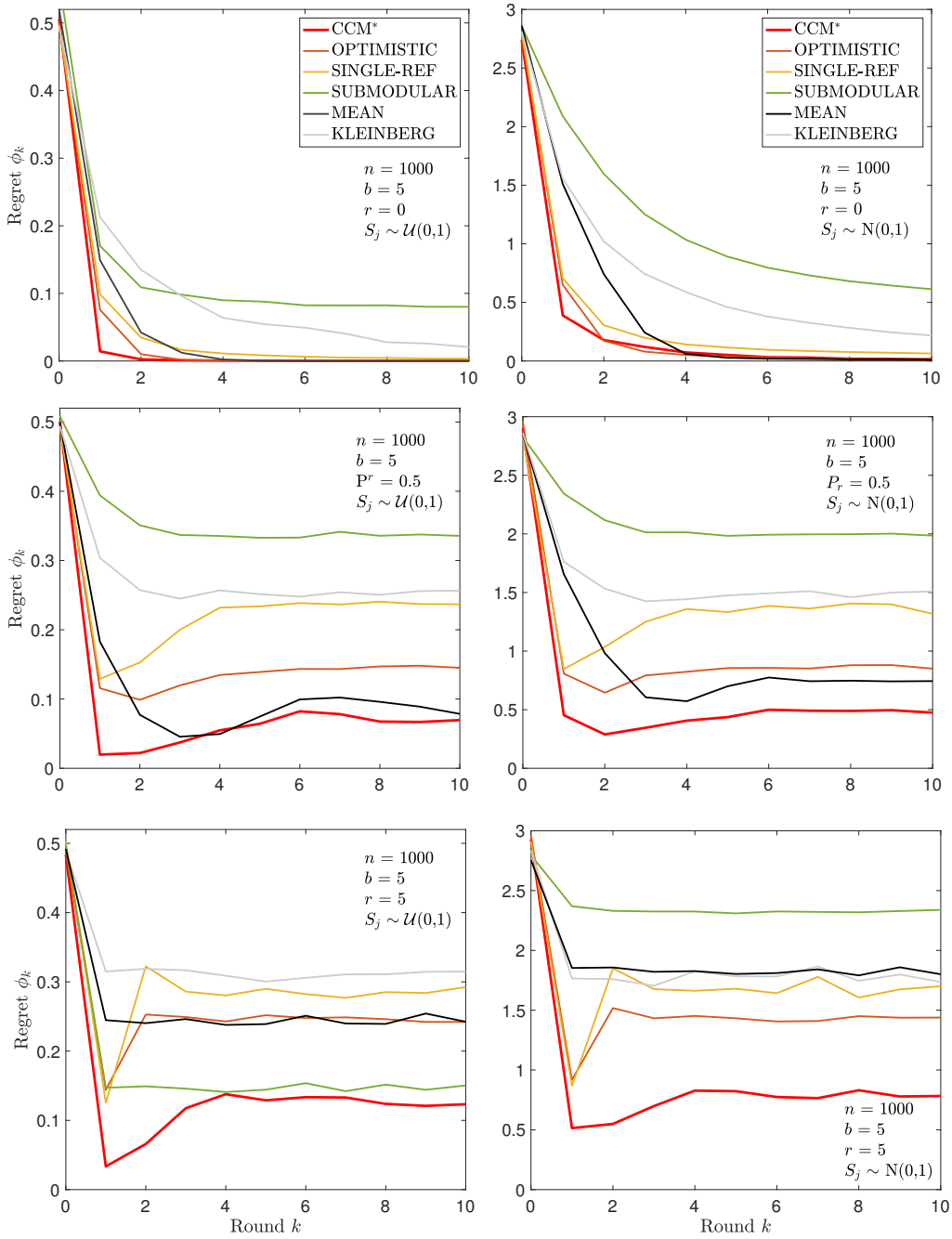
Following the well-known SP, we developed a *cutoff-based* strategy, the Cutoff-based Cost Minimization (and the low failures-CCM improved version), composed of a learning phase and a selection phase. Setting the optimal length of the former according to the number of initially empty jobs is an intriguing question for which we brought interesting and not always straight to see results. The rank-based regret function that we used in the WSSP setting enables our algorithm to be efficient for



(a) Uniform(0, 1) score distribution (b) Normal(0, 1) score distribution

FIGURE 4.9: Average rank-based regret ϕ_k w.r.t. the round number k for a budget of $b = 5$ jobs and $n = 1000$ candidates. A different resignation probability is used in each row, namely: $\mathrm{P}^r = 0$ $(r = 0)$(top row), $\mathrm{P}^r = 0.5$ (middle row) and $\mathrm{P}^r = 1$ $(r = b = 5)$ (bottom row).

arbitrary candidate scores. We approximate analytically this objective function by deriving the expectation of the main parameters of the problem (e.g. the acceptance threshold, the number of accepted candidates, the regret, etc.). Secondly, we implemented CCM in a multi-round framework (MSSP). Such a process was motivated by the natural needs of real-world recruitment processes that are regularly trying to improve the personnel of an organization or a company.



(a) Uniform$(0, 1)$ score distribution (b) Normal$(0, 1)$ score distribution

FIGURE 4.10: Average score-based regret ϕ_k w.r.t. the round number k for a budget of $b = 5$ jobs and $n = 1000$ candidates. A different resignation probability is used in each row, namely: $\mathrm{P}^r = 0$ $(r = 0)$(top row), $\mathrm{P}^r = 0.5$ (middle row) and $\mathrm{P}^r = 1$ $(r = b = 5)$ (bottom row).

The conducted simulations are consistent with our analytical work and demonstrated that CCM is efficient in reducing the regret at the course of the multi-round process while being robust to scores, resignations, or changes in the number of jobs.

4.6 Appendix – Technical proofs

Proof of Lemma 1. Set $p_j = \mathbb{P}(X_j < \gamma_j)$, $Z_j \sim \text{Bernoulli}(j-c, p_j)$, and $\tilde{Z}_j = \sum_{i=c+1}^{j} Z_i$. Thus:

$$\mathbb{P}(\tilde{A}_{j-1} < b) = \mathbb{P}\left(\min\left(\sum_{i=c+1}^{j-1} Z_i, b\right) < b\right) = \mathbb{P}\left(\sum_{i=c+1}^{j-1} Z_i < b\right) = \mathbb{P}(\tilde{Z}_{j-1} < b). \tag{4.19}$$

We have $q = 1/2$, hence $(\dot{\mathbf{X}}, \mathbf{X})$ is uniformly distributed in $\{1, ..., n+b\}$. Therefore, $p_j = \frac{\gamma_j - 1}{n+b}$ and since $j \leq n$, $j \to \infty \Rightarrow n \to \infty$ thus, $\lim_{j\to\infty} \sum_{i=c+1}^{j} (\frac{\gamma_i - 1}{n+b})^2 = 0$. Therefore $\lim_{j\to\infty} \sum_{i=c+1}^{j} p_i^2 = 0$, in other words, the more candidates there are, the smaller the probability for each of them to be accepted. Set $\sigma_j^2 = \sum_{i=c+1}^{j} p_i^2$ and $\lambda_j = p_{c+1} + ... + p_j$. From Le Cam's theorem [78, 100] we have $\sum_{k=0}^{\infty} \left| \mathbb{P}(\tilde{Z}_j = k) - \frac{\lambda_j^k e^{-\lambda_j}}{k!} \right| < 2 \sum_{i=1}^{j} p_i^2$,

i.e. $\tilde{Z}_j$ follows approximately a Poisson distribution with parameter λ_j, and since $\lim_{j\to\infty} \sigma_j^2 = 0$, the bound gets tighter as $j \to \infty$.

□

Proof of Theorem 1. We handle separately:

- First, we investigate the rank-based expected threshold to beat for the first incoming candidate just after the learning phase, $\gamma := \mathbb{E}[S^{\text{upd}}_{(b),c}]$. The proof is done by backward induction. We first consider the case where the number of rejected candidates c is s.t. $c = n$; the updated reference set is composed of the b-best items of $(\dot{\mathbf{X}}, \mathbf{X})$ since every candidate has been rejected and their scores are stored in the updated reference set. Thus $\gamma(c = n) = b$. Let us go one step ahead and consider the case where $c = n - 1$, which implies that $\gamma = b$ if the candidate that has not been examined is not among the b-best items, and $b+1$ if he is. Hence:

  $$\gamma(c = n-1) = b\frac{c}{b+c} + (b+1)\frac{b}{b+c},$$

  and one more step ahead:

  $$\gamma(c = n-2) = b\left(\frac{c}{b+c}\right)^2 + 2(b+1)\frac{b}{b+c}\frac{c}{b+c} + (b+2)\left(\frac{b}{b+c}\right)^2,$$

By recursion, we get:

$$\begin{aligned}\gamma(c) &= \sum_{m=0}^{n-c}\binom{n-c}{m}\left(\frac{c}{b+c}\right)^{n-c-m}\left(\frac{b}{b+c}\right)^{m}(b+m)\\ &= \frac{1}{(b+c)^{n-c}}\sum_{m=0}^{n-c}\binom{n-c}{m}c^{n-c-m}b^{m}(b+m)\\ &= \frac{b}{(b+c)^{n-c}}\sum_{m=0}^{n-c}\binom{n-c}{m}c^{n-c-m}b^{m}+\frac{1}{(b+c)^{n-c}}\sum_{m=0}^{n-c}m\binom{n-c}{m}c^{n-c-m}b^{m}\\ &= \frac{b}{(b+c)^{n-c}}(b+c)^{n-c}+\frac{b(n-c)}{(b+c)^{n-c}}(b+c)^{n-c-1}\\ \Leftrightarrow\quad \gamma(c) &= \frac{b(n+b)}{b+c}.\end{aligned}$$

The acceptance threshold is replaced by its expectation, in particular $\gamma := \mathbb{E}[S^{\text{upd}}_{(b),c}]$. Hence $\delta = r+\sum_{j=1}^{c}\mathbb{1}\{X_j < S^{\text{upd}}_{(b),c}\}$ becomes $\mathbb{E}[r+\sum_{j=1}^{c}\mathbb{1}\{X_j<\gamma\}]$, i.e. $\Delta := \mathbb{E}[\delta] = r+\sum_{j=1}^{c}\mathbb{P}(X_j<\gamma) = r+c\frac{\gamma-1}{n+b}$. Then, the evolving threshold is $\gamma_j = \gamma\mathbb{E}[\mathbb{1}\{\tilde{A}_{j-1}<\Delta\}]+\mathbb{E}[\dot{X}^{\text{avail}}_{(b-\tilde{A}_{j-1})}\mathbb{1}\{\tilde{A}_{j-1}\geq\Delta\}]$. In order to use the fact that $\mathbb{E}[\dot{X}^{\text{avail}}_{(l)}] = \mathbb{E}[\dot{X}^{\text{avail}}_{(1)}]l$, $\forall l\in\{1,...,b-r\}$, in the proof we approximate $\dot{X}^{\text{avail}}_{(b-\tilde{A}_{j-1})}$ by $\dot{X}^{\text{avail}}_{(b-\mathbb{E}[\tilde{A}_{j-1}])}$ by considering that $\tilde{A}_j$ has a small variance, which is given by σ^2_{j-1}. Therefore:

$$\begin{aligned}\gamma_j &= \gamma\mathbb{E}[\mathbb{1}\{\tilde{A}_{j-1}<\Delta\}]+\mathbb{E}[\dot{X}^{\text{avail}}_{(b-\mathbb{E}[\tilde{A}_{j-1}])}]\mathbb{E}[\mathbb{1}\{\tilde{A}_{j-1}\geq\Delta\}]+o(\sigma^2_{j-1})\\ &= \gamma\mathbb{P}(\tilde{A}_{j-1}<\Delta)+\frac{\gamma_0(b+1)}{b(b-r+1)}(b-\mathbb{E}[\tilde{A}_{j-1}])\mathbb{P}(\tilde{A}_{j-1}\geq\Delta)+o(\sigma^2_{j-1})\end{aligned}$$

We have $\mathbb{E}[\tilde{A}_j] = \sum_{i=c+1}^{j}\mathbb{E}[A_i] = \sum_{i=c+1}^{j}\mathbb{P}(A_i=1) = \sum_{i=c+1}^{j}\mathbb{P}(X_i<\gamma_i)\mathbb{P}(\tilde{A}_{i-1}<b)$; hence:

$$\gamma_j = \gamma g_{c,j}(\Delta)+\frac{\gamma_0(b+1)}{b(b-r+1)}\left(b-\sum_{i=1}^{j-1}\frac{\gamma_i-1}{n+b}g_{c,i}(b)\right)(1-g_{c,j}(\Delta))+o(\sigma^2_{j-1}).$$

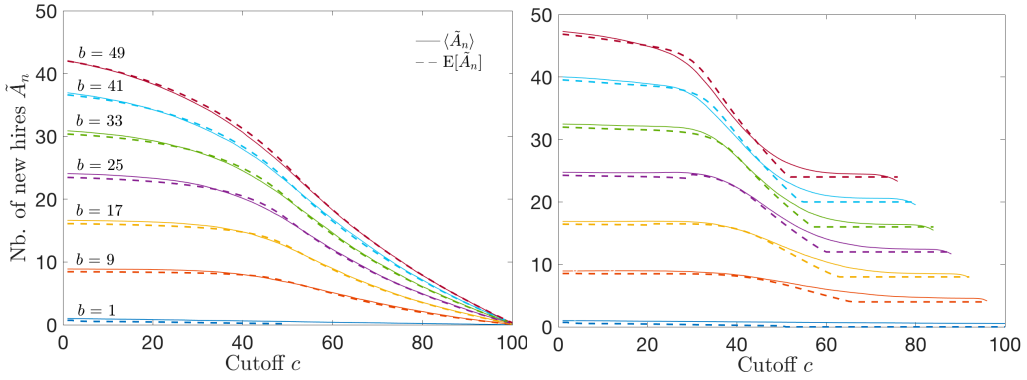
where $g_{c,j}(x) := \mathbb{P}(\tilde{A}_{j-1}<x)$ is computed using Lemma 1.

- The score-based acceptance threshold τ_j is replaced by its expected rank-based equivalent, γ_j, hence Eq. 4.2 becomes $A_j = \mathbb{1}\{j>c\}\mathbb{1}\{\tilde{A}_{j-1}<b\}\mathbb{1}\{X_j<\gamma_j\}$, hence:

$$\begin{aligned}&\mathbb{E}[\tilde{A}_n] := \sum_{j=1}^{n}\mathbb{E}[A_j] = \sum_{j=1}^{n}\mathbb{P}(A_j=1) = \sum_{j=1}^{n}\mathbb{P}(X_j<\gamma_j)\mathbb{P}(\tilde{A}_{j-1}<b)\\ \Leftrightarrow\quad &\mathbb{E}[\tilde{A}_n] = \sum_{j=1}^{n}\frac{\gamma_j-1}{n+b}g_{c,j}(b).\end{aligned}$$

Recall that from the definition the second bullet point, $\tilde{A}_n = \max(\tilde{A}_n, r)$, thus in practice we use the following approximation $\mathbb{E}[\tilde{A}_n] = \mathbb{E}[\tilde{A}_n\mathbb{1}\{\tilde{A}_n\geq r\}]+r\mathbb{E}[\mathbb{1}\{\tilde{A}_n<r\}] \approx \mathbb{E}[\tilde{A}_n](1-g_{c,n+1}(r))+rg_{c,n+1}(r)$, where $g_{c,n+1}(r)$ is defined in Lemma 1, see the empirical verification in Fig. 4.11.

- Recall the definition of the regret $\phi := \dot{\mathbf{X}}^{\mathsf{T}}\dot{\mathbf{A}}_n + \mathbf{X}^{\mathsf{T}}\mathbf{A} - \phi_{\text{off}}$. Set $\phi_1 = \dot{\mathbf{X}}^{\mathsf{T}}\dot{\mathbf{A}}_n$ and



(a) $r = 0$ (b) $r = 0.5b$

FIGURE 4.11: Dotted lines show display the analytical number of new hires $\mathbb{E}[\tilde{A}_n]$ given in Eq. 4.5, while plain lines are the respective empirical $\tilde{A}_n$, i.e. the average number of accepted candidates in the simulations. Different colors depict scenarios with number of jobs being, from bottom to top line, $b = \{1, 9, 17, 25, 33, 41, 49\}$.

$\phi_2 = \mathbf{X}^\mathsf{T}\mathbf{A}$ that give respectively the reference set and the candidates contribution to the regret. We start with the candidates, $\phi_2 = \sum_{j=1}^{n} X_j A_j$. Its expectation is given by $\mathbb{E}[\phi_2] = \mathbb{E}[\sum_{j=1}^{n} X_j A_j]$. We use the fact that $A_j = 0, \ \forall j \leq c$:

$$\mathbb{E}[\phi_2] = \sum_{j=c+1}^{n} \sum_{m=1}^{n+b} \sum_{a=\{0,1\}} \mathbb{P}(X_j = m, A_j = a) a m$$

$$\mathbb{E}[\phi_2] = \sum_{j=c+1}^{n} \sum_{m=1}^{n+b} \mathbb{P}(A_j = 1 \mid X_j = m) \mathbb{P}(X_j = m) m$$

A candidate with rank higher than the threshold γ_j is rejected, hence:

$$\mathbb{E}[\phi_2] = \sum_{j=c+1}^{n} \sum_{m=1}^{\gamma_j - 1} \mathbb{P}(A_j = 1 \mid X_j = m) \mathbb{P}(X_j = m)\, m.$$

A candidate with rank lower than the threshold is accepted if there were less than b candidates accepted before him. Moreover, we use the fact that $\mathbb{P}(X_j = m) = \frac{1}{n+b}$ to write:

$$\mathbb{E}[\phi_2] = \sum_{j=c+1}^{n} \sum_{m=1}^{\gamma_j - 1} \mathbb{P}(\tilde{A}_{j-1} < b) \frac{m}{n+b} = \sum_{j=c+1}^{n} \sum_{m=1}^{\gamma_j - 1} g_{c,j}(b) \frac{m}{n+b}$$

$$\Leftrightarrow \mathbb{E}[\phi_2] = \frac{1}{n+b} \sum_{j=c+1}^{n} g_{c,j}(b) \frac{\gamma_j(\gamma_j - 1)}{2}.$$

Following up with the reference set contribution, the regret associated with the available referents that were not fired at the end of the selection is given by:

$$\phi_1 = \sum_{l=1}^{b} \dot{X}_{(l)}^{\text{avail}} \mathbb{1}\{l \leq b - \tilde{A}_n\} = \sum_{l=1}^{b-\tilde{A}_n} \dot{X}_{(l)}^{\text{avail}}.$$

We suppose that variables $\tilde{A}_n$ and $\dot{X}_{(l)}$ are independent $\forall l$, which is a reasonable assumption since we consider a reference set with medium quality, i.e. medium average rank, and we use $\mathbb{E}[\dot{X}_{(l)}^{\text{avail}}] = \frac{\gamma_0(b+1)l}{b(b-r+1)}$ (see Proposition 1):

$$\mathbb{E}[\phi_1] = \sum_{l=1}^{b-\mathbb{E}[\tilde{A}_n]} \frac{\gamma_0(b+1)l}{b(b-r+1)} = \frac{\gamma_0(b+1)}{b(b-r+1)} \sum_{l=1}^{b-\mathbb{E}[\tilde{A}_n]} l$$

$$\Leftrightarrow \quad \mathbb{E}[\phi_1] = \frac{\gamma_0(b+1)}{2b(b-r+1)}(b-\mathbb{E}[\tilde{A}_n])(b+1-\mathbb{E}[\tilde{A}_n]).$$

Finally the expected regret is given by:

$$\mathbb{E}[\phi] := \mathbb{E}[\phi_1] + \mathbb{E}[\phi_2] - \mathbb{E}[\phi_{\text{off}}]$$

$$\mathbb{E}[\phi] := \frac{\gamma_0(b+1)}{2b(b-r+1)}(b-\mathbb{E}[\tilde{A}_n])(b+1-\mathbb{E}[\tilde{A}_n]) + \frac{1}{n+b}\sum_{j=c+1}^{n} g_{c,j}(b)\frac{\gamma_j(\gamma_j-1)}{2} - \mathbb{E}[\phi_{\text{off}}],$$

which concludes the proof.

Practical implementation. As mentioned before, in the implementation last accepted candidates are included in the following way: if at the end of the sample the number of accepted candidates (i.e. $\tilde{A}_n$) is smaller than the number of resignations r, we add 'extra' candidates of medium rank and rescale the expectation of the regret. Therefore, the regret rescale from $\phi = \phi_1 + \phi_2 - \phi_{\text{off}}$ to:

$$\phi = \phi_1 + \mathbb{1}\{\tilde{A}_n \geq r\}\phi_2 + \mathbb{1}\{\tilde{A}_n < r\}\frac{n}{n-r+\mathbb{E}[\tilde{A}_n]}\phi_3 - \phi_{\text{off}};$$

where a failure is given by the event $\{\tilde{A}_n < r\}$, and where ϕ_3 is given by $\phi_3 = \phi_2 + \frac{n+b+1}{2}$, i.e. it gives the regret associated with candidates accepted if there was at least one failure at the end of the selection, the last term being the average rank of a candidate accepted by default. Hence, in practice we have:

$$\mathbb{E}[\phi] = \mathbb{E}[\phi_1] + \mathbb{P}(\tilde{A}_n \geq r)\mathbb{E}[\phi_2] + \mathbb{P}(\tilde{A}_n < r)\frac{n}{n-r+\mathbb{E}[\tilde{A}_n]}\left(\mathbb{E}[\phi_2] + \frac{n+b+1}{2}\right) - \mathbb{E}[\phi_{\text{off}}]$$

$$\mathbb{E}[\phi] = \mathbb{E}[\phi_1] + (1-g_{n+1}(r))\,\mathbb{E}[\phi_2] + g_{n+1}(r)\frac{n}{n-r+\mathbb{E}[\tilde{A}_n]}\left(\mathbb{E}[\phi_2] + \frac{n+b+1}{2}\right) - \mathbb{E}[\phi_{\text{off}}],$$

where $g_{n+1}(x) := \mathbb{P}(\tilde{A}_n < x)$ is given in Lemma 1.

□

Proof of Theorem 2. As in the proof of Theorem 1, we handle separately.

- The expected acceptance threshold when $q = 1/2$ is given in Eq. 4.4 by:

$$\gamma := \mathbb{E}[R(S_{(b),c}^{\text{upd}})] = \frac{b(b+n)}{b+c}.$$

When the referent has a rank of γ_0 (which is known), $q \neq 1/2$, in fact $q = 1 - \frac{\gamma_0 - 1}{n}$, therefore, for $b = 1$ we get:

$$\gamma := \mathbb{E}[R(S^{\text{upd}}_{(1),c})] = \min\left(\gamma_0, \frac{n+1}{c+1}\right).$$

- Recall the formula for the regret:

$$\phi = \left(\dot{\mathbf{X}}^\intercal \dot{\mathbf{A}}_n + \mathbf{X}^\intercal \mathbf{A}\right) - \min_{(\dot{\mathbf{O}}_n, \mathbf{O}) \in \pi_{\text{off},\mathcal{B}}} \left(\dot{\mathbf{X}}^\intercal \dot{\mathbf{O}}_n + \mathbf{X}^\intercal \mathbf{O}\right) \in \mathbb{R}_+,$$

where $\pi_{\text{off},\mathcal{B}} = \left\{(\dot{\mathbf{O}}_n, \mathbf{O}) \in \{0,1\}^{n+b} : (\dot{\mathbf{O}}_n, \mathbf{O})^\intercal \mathbf{1}_{[n+b]} = b\right\}$. Here, since there is only one job position to manage, it is either occupied by a candidate, or by the initial referent. Moreover, the offline term that accounts for the outcome of an offline strategy that would see every item altogether as a batch simply becomes the best rank, i.e. 1. In the end, we get:

$$\phi = \sum_{j=1}^{n} X_j A_j + \gamma_0 \mathbb{1}\{\sum_{j=1}^{n} A_j = 0\} - 1,$$

First, we want to evaluate the right-hand term, i.e. the probability of accepting no candidate and keeping the preselected node. The important observation to make when using the CCM strategy is that the event of not accepting any candidate is the same as the event of having the best candidate among the c first. Indeed, if the best individual (i.e. the one with rank equal to 1) is among the first c candidates, he gets rejected, and becomes the acceptance threshold to beat according to CCM. As he is the best, none of the next incoming candidates is accepted, and the preselected individual keeps the job. Now, if the best candidate is not among the rejected but the second-best is, the latter becomes the acceptance threshold, and eventually the best candidate is accepted. Using similar reasoning for the third, fourth, fifth-best and so on, we find that if the best candidate is not among the c first candidates, at least one candidate is accepted. Hence we write the expectation of the regret as:

$$\begin{aligned}
\mathbb{E}[\phi(c)] &= \sum_{j=1}^{n} \mathbb{E}[X_j A_j] + \gamma_0 \mathbb{P}(\text{“the best candidate is among the } c \text{ first”}) - 1 \\
\mathbb{E}[\phi(c)] &= \sum_{j=1}^{n} \sum_{m=1}^{n+1} \mathbb{P}(X_j = m, A_j = 1) m + \gamma_0 \sum_{i=1}^{c} \mathbb{P}(X_j = 1) - 1 \\
\mathbb{E}[\phi(c)] &= \sum_{j=1}^{n} \sum_{m=1}^{n+1} \mathbb{P}(A_j = 1 \mid X_j = m) \mathbb{P}(X_j = m) m + \gamma_0 \sum_{i=1}^{c} \mathbb{P}(X_j = 1) - 1.
\end{aligned}$$

We set $\tilde{A}_{j-1} = \sum_{i=1}^{j-1} A_i$, and $\mathbb{P}(X_j = m) = \frac{1}{n}\mathbb{1}\{\gamma_0 \neq m\}, \forall m \in \{1, ..., n+1\}$ since ranks are uniformly distributed among candidates. Note that the probability of accepting a candidate given that his rank is worse than that of the threshold γ is equal to zero. Otherwise, the probability for the j-th candidate to be accepted is equal to the probability of not finding a suitable candidate before him (i.e. $A_j = \mathbb{1}\{X_j < \gamma\}\mathbb{1}\{\tilde{A}_{j-1} = 0\}, \forall j > c$). Therefore:

$$\mathbb{E}[\phi(c)] = \sum_{j=c+1}^{n} \sum_{m=1}^{\gamma-1} \mathbb{P}(\tilde{A}_{j-1} = 0)\frac{1}{n}m + \gamma_0 c\left(\frac{1}{n}\mathbb{1}\{\gamma_0 \neq 1\}\right) - 1$$

$$\mathbb{E}[\phi(c)] = \frac{1}{n}\frac{\gamma(\gamma-1)}{2} \sum_{j=c+1}^{n} \mathbb{P}(\tilde{A}_{j-1} = 0) + \gamma_0 c\left(\frac{1}{n}\mathbb{1}\{\gamma_0 \neq 1\}\right) - 1$$

No candidate is accepted up to step j if each of them is below the acceptance threshold, i.e.:

$$\mathbb{E}[\phi(c)] = \frac{1}{n}\frac{\gamma(\gamma-1)}{2} \sum_{j=c+1}^{n} \prod_{i=c+1}^{j-1} \mathbb{P}(X_j \geq \gamma) + \gamma_0 c\left(\frac{1}{n}\mathbb{1}\{\gamma_0 \neq 1\}\right) - 1$$

$$\mathbb{E}[\phi(c)] = \frac{1}{n}\frac{\gamma(\gamma-1)}{2} \sum_{j=c+1}^{n} \prod_{i=c+1}^{j-1} \left(1 - \frac{1}{n}(\gamma-1)\right) + \gamma_0 c\left(\frac{1}{n}\mathbb{1}\{\gamma_0 \neq 1\}\right) - 1.$$

Set $z = \frac{1}{n}(\gamma - 1)$, s.t. $0 \leq z \leq 1$:

$$\mathbb{E}[\phi(c)] = \frac{1}{n}\frac{\gamma(\gamma-1)}{2} \sum_{j=c+1}^{n} (1-z)^{j-c-1} + \gamma_0 c\left(\frac{1}{n}\mathbb{1}\{\gamma_0 \neq 1\}\right) - 1$$

$$\mathbb{E}[\phi(c)] = \frac{1}{n}\frac{\gamma(\gamma-1)}{2}\frac{1-(1-z)^{n-c}}{1-(1-z)} + \gamma_0 c\left(\frac{1}{n}\mathbb{1}\{\gamma_0 \neq 1\}\right) - 1$$

$$\mathbb{E}[\phi(c)] = \frac{\gamma}{2}\left(1-(1-z)^{n-c}\right) + \gamma_0 c\left(\frac{1}{n}\mathbb{1}\{\gamma_0 \neq 1\}\right) - 1.$$

Hence, in the non-trivial case $\gamma_0 \neq 1$:

$$\mathbb{E}[\phi(c)] = \frac{\gamma}{2} - \frac{\gamma}{2}\left(1 - \frac{\gamma-1}{n}\right)^{n-c} + \frac{\gamma_0}{n}c - 1.$$

- We now want to find the optimal cutoff c^* s.t. $\frac{\partial \mathbb{E}[\phi(c)]}{\partial c} = 0\,|_{c=c^*}$. Starting with the expression of the expected regret:

We distinguish two cases, either $\gamma_0 < \frac{n+1}{c+1}$ or $\gamma_0 \geq \frac{n+1}{c+1}$. Since $\gamma = \min(\gamma_0, \frac{n+1}{c+1})$, starting with the former gives $\gamma = \gamma_0$, and therefore:

$$\mathbb{E}[\phi(c)] = \frac{\gamma_0}{2} - \frac{\gamma_0}{2}\left(1 - \frac{\gamma_0 - 1}{n}\right)^{n-c} + \frac{\gamma_0}{n}c - 1.$$

The latter equation is always minimized at $c = c^* = 0$.

Let us then focus on the second, more interesting, case where $\gamma_0 \geq \frac{n+1}{c+1}$. In this case $\gamma = \frac{n+1}{c+1}$, hence:

$$\mathbb{E}[\phi(c)] = \frac{n+1}{2(c+1)} - \frac{n+1}{2(c+1)}\left(1 - \frac{\frac{n+1}{c+1}-1}{n}\right)^{O(n)} + \frac{\gamma_0}{n}c - 1$$
$$\mathbb{E}[\phi(c)] = \frac{n+1}{2(c+1)} - \frac{n+1}{2(c+1)}\left(1 - \frac{n-c}{n(c+1)}\right)^{O(n)} + \frac{\gamma_0}{n}c - 1$$
$$\mathbb{E}[\phi(c)] = \frac{n+1}{2(c+1)} - \frac{n+1}{2(c+1)}e^{O(n)\ln\left(1-\frac{n-c}{n(c+1)}\right)} + \frac{\gamma_0}{n}c - 1$$
$$\mathbb{E}[\phi(c)] = \frac{n+1}{2(c+1)} - \frac{n+1}{2(c+1)}e^{O(n)\ln\left(1-\frac{1-\frac{c}{n}}{c+1}\right)} + \frac{\gamma_0}{n}c - 1$$
$$\mathbb{E}[\phi(c)] = \frac{n+1}{2(c+1)} - \frac{n+1}{2(c+1)}e^{O(n)\ln\left(1-O\left(\frac{1}{c}\right)\right)} + \frac{\gamma_0}{n}c - 1$$

Using $\ln(1-x) < -x,\ \forall\, 0 < x \leq 1$, we get:

$$\mathbb{E}[\phi(c)] = \frac{n+1}{2(c+1)} - \frac{n+1}{2(c+1)}e^{O\left(\frac{n}{c}\right)} + \frac{\gamma_0}{n}c - 1$$

We have $\frac{n+1}{2(c+1)}e^{O\left(\frac{n}{c}\right)} \leq \frac{n}{c}e^{O\left(\frac{n}{c}\right)}$, and $xe^{-x} \leq 1\ \forall x \in \mathbb{R}$, thus $\frac{n+1}{2(c+1)}e^{O\left(\frac{n}{c}\right)} = O(1)$. Finally, we write:

$$\mathbb{E}[\phi(c)] = \frac{n+1}{2(c+1)} + \frac{\gamma_0}{n}c - 1 + O(1).$$

Deriving the expected regret, we get:

$$\frac{\partial \mathbb{E}[\phi(c)]}{\partial c} = -\frac{n+1}{2(c^*+1)^2} + \frac{\gamma_0}{n}.$$

Therefore:

$$\begin{aligned}
\frac{\partial \mathbb{E}[\phi(c)]}{\partial c} = 0 \ &\Rightarrow\ \frac{n+1}{2(c^*+1)^2} = \frac{\gamma_0}{n} \\
&\Rightarrow\ c^* = \sqrt{\frac{n+1}{2\left(\frac{\gamma_0}{n}\right)}} - 1 \\
&\Leftrightarrow\ c^* = \sqrt{\frac{n+1}{2\left(1-q+\frac{1}{n}\right)}} - 1.
\end{aligned}$$

□

Proof of Proposition 3. If two settings are γ_0-similar, then the optimal proportion of candidates to reject compared to the total number of items (reference set and candidates) is the same for both settings which explains the first term of the proposition. Using the definition of the relative quality of the reference set for two equivalent

settings (see Definitions 4, 7), we get:

$$q_x = 1 - \frac{\frac{1}{b}||\dot{\mathbf{X}}_x||_1 - 1}{n_x + b - 1} = 1 - \frac{\frac{1}{b}||\dot{\mathbf{X}}_y||_1 - 1}{n_x + b - 1} \quad \Leftrightarrow \quad 1 - q_x = \frac{(1-q_y)(n_y + b - 1)}{n_x + b - 1}.$$
$$\Leftrightarrow \quad n_x = (n_y + b - 1)\frac{1-q_y}{1-q_x} - b + 1.$$

□

Proof of Proposition 4. Let us replace each item absolute rank by its actual value to match the definition of the competitive ratio. Thus, a rank $X = m$ is replaced by its value $v_m = n + b - m$ so that the best (worst) item with rank $X = 1$ ($X = n + b$) has the highest (lowest) value $v_1 = n + b - 1$ ($v_{n+b} = 0$). Using our formalism, we have $\mathbb{E}[n + b - (\phi + \phi_{\text{off}})] \geq \alpha\,\mathbb{E}[(n + b - \phi_{\text{off}})]$. Now when $r = 0$, the CCM acceptance threshold reduces to that of the OPTIMISTIC algorithm proposed in [8]. In [5] (Lemma 2.1 therein) it is shown that, using the OPTIMISTIC algorithm in a cold-start setting, it holds $\alpha = \frac{1}{b}\sum_{m=1}^{b} p_m$, where p_m is the probability of accepting rank m. In our case, either a referent (they are all available) or a candidate has the rank m, namely $p_m := \mathbb{P}(\dot{A}_{0,(l_{[m]})} = 1) + \mathbb{P}(A_{j_{[m]}} = 1)$, where $l_{[m]}$ and $j_{[m]}$ are the indices respectively referring to the referent with rank m or the candidate with rank m, i.e. $\dot{X}_{(l_{[m]})} = m$ or $X_{j_{[m]}} = m$. Then, the first term gives $\mathbb{P}(\dot{A}_{0,(l_{[m]})} = 1) = \sum_{l=1}^{b} \mathbb{P}(\dot{X}_{(l)} = m) = \left(\frac{\gamma_0(l+1)}{b} - \frac{\gamma_0(l-1)}{b}\right)^{-1} = \frac{b^2}{2\gamma_0}$ (see proof of Proposition 1), while the second term gives, $\mathbb{P}(A_{j_{[m]}} = 1) = \mathbb{P}(A_{j_{[m]}} = 1 | \dot{X}_{(1)} \geq b)\mathbb{P}(\dot{X}_{(1)} \geq b) + \mathbb{P}(A_{j[m]} = 1 | \dot{X}_{(1)} < b)\mathbb{P}(\dot{X}_{(1)} < b) \geq \mathbb{P}(A_{j_{[m]}} = 1 | \dot{X}_{(1)} \geq b)\mathbb{P}(\dot{X}_{(1)} \geq b)$, where $\mathbb{P}(\dot{X}_{(1)} \geq b) = (1 - \frac{b}{2\gamma_0/b})$. Observe that the probability to accept a candidate with rank $m \leq b$ given that the best referent has a rank higher than b is necessarily larger than that of accepting a candidate with rank $m \leq b$ when considering a cold-start with $n+b$ candidates, which is, from [8] (Lemma 2 therein), larger than $\frac{c}{n}\ln(\frac{n}{c})$. Finally, we have $\alpha = \left(\frac{b^2}{2\gamma_0} + \frac{c^*}{n}\ln(\frac{n}{c^*})(1 - \frac{b^2}{2\gamma_0})\right)$. □

Proof of Proposition 5. Set $\tilde{Z}_j = \sum_{i=c+1}^{j} Z_i$ where $Z_j \sim \text{Bernouilli}(j-c, p_j)$ and set $\lambda_j = \sum_{i=c+1}^{j} p_i,\ \forall j$. We have $\mathbb{E}[\tilde{A}_j \mid \tilde{A}_n \geq r] = \sum_{k=0}^{b} k\mathbb{P}(\tilde{A}_j = k \mid \tilde{A}_n \geq r)$. Hence:

$$\begin{aligned}
\mathbb{E}[\tilde{A}_j \mid \tilde{A}_n \geq r] &= \sum_{k=0}^{b} k \frac{\mathbb{P}\big(\min(\tilde{Z}_j, b) = k, \min(\tilde{Z}_n, b) \geq r)\big)}{\mathbb{P}(\tilde{A}_n \geq r)} \\
&= \frac{1}{\mathbb{P}(\tilde{A}_n \geq r)} \sum_{k=0}^{b} k\mathbb{P}\Big(\big(\{\tilde{Z}_j < b, \tilde{Z}_j = k\} \cup \{\tilde{Z}_j \geq b, b = k\}\big) \\
&\cap \Big(\{\tilde{Z}_n < b, \tilde{Z}_n \geq r\} \cup \{\tilde{Z}_n \geq b, b \geq r\}\Big)\Big) \\
&= \frac{1}{\mathbb{P}(\tilde{A}_n \geq r)} \sum_{k=0}^{b-1} k\mathbb{P}\Big(\{\tilde{Z}_j = k\} \cap \Big(\{\tilde{Z}_n < b, \tilde{Z}_n \geq r\} \cup \{\tilde{Z}_n \geq b\}\Big)\Big) \\
&+ \frac{b}{\mathbb{P}(\tilde{A}_n \geq r)} \mathbb{P}\Big(\big(\{\tilde{Z}_j < b, \tilde{Z}_j = b\}) \cup \{\tilde{Z}_j \geq b\}\big) \cap \Big(\{\tilde{Z}_n < b, \tilde{Z}_n \geq r\} \cup \{\tilde{Z}_n \geq b\}\Big)\Big) \\
&= \frac{1}{\mathbb{P}(\tilde{A}_n \geq r)} \sum_{k=0}^{b-1} k\mathbb{P}\Big(\{\tilde{Z}_j = k\} \cap \{\tilde{Z}_n \geq r\}\Big) + \frac{b}{\mathbb{P}(\tilde{A}_n \geq r)} \mathbb{P}\Big(\{\tilde{Z}_j \geq b\} \cap \{\tilde{Z}_n \geq r\}\Big) \\
&= \frac{1}{\mathbb{P}(\tilde{A}_n \geq r)} \sum_{k=0}^{b-1} k\mathbb{P}(\tilde{Z}_n \geq r | \tilde{Z}_j = k)\mathbb{P}(\tilde{Z}_j = k) + \frac{b\mathbb{P}(\tilde{Z}_j \geq b)}{\mathbb{P}(\tilde{A}_n \geq r)} \\
&= \frac{1}{\mathbb{P}(\tilde{A}_n \geq r)} \sum_{k=0}^{b-1} k\mathbb{P}(\tilde{Z}_n - \tilde{Z}_j \geq r - k)\mathbb{P}(\tilde{Z}_j = k) + \frac{b\mathbb{P}(\tilde{Z}_j \geq b)}{\mathbb{P}(\tilde{A}_n \geq r)} \\
\mathbb{E}[\tilde{A}_j \mid \tilde{A}_n \geq r] &= \frac{1}{1 - g_{c,n+1}(r)} \left(b(1 - g_{c,j+1}(b)) + \sum_{k=0}^{b-1} k(1 - g_{j,n+1}(r-k))\mathbb{P}(\tilde{Z}_j = k) \right).
\end{aligned}$$

From Le Cam's theorem we have $\sum_{m=0}^{\infty} |\mathbb{P}(\tilde{Z}_j = m) - \frac{\lambda_j^m e^{-\lambda_j}}{m!}| < 2\sigma_j^2$, and since $\lim_{j\to\infty} \sigma_j^2 = 0$, $\tilde{Z}_j$ tends a Poisson distribution with parameter λ_j, see proof of Lemma 1, hence:

$$= \frac{1}{1 - g_{c,n+1}(r)} \left(b(1 - g_{c,j+1}(b)) + \sum_{k=0}^{b-1} k(1 - g_{j,n+1}(r-k)) \frac{\lambda_j^k e^{-\lambda_j}}{(k-1)!} \right).$$

□

Proof of Proposition 6. Let us start by assessing what a failure is in this case. Since $b = 1$, a failure means that, while evaluating the last candidate, none has been accepted, i.e. $\tilde{A}_n = 0$, and therefore 'no failure' is the complementary event that happens with a probabilty $1 - \mathbb{P}(\tilde{A}_n = 0)$. Hence, the expected number of accepted given

that there were no failure is given by $\mathbb{E}[\tilde{A}_j|\text{ no failure}] := \mathbb{E}[\tilde{A}_j|\tilde{A}_n = 1]$:

$$\begin{aligned}
\mathbb{E}[\tilde{A}_j|\tilde{A}_n = 1] = \sum_{a=\{0,1\}} a\,\mathbb{P}(\tilde{A}_j = a \mid \text{no failure}) &= \frac{\mathbb{P}(\tilde{A}_j = 1, \tilde{A}_n = 1)}{\mathbb{P}(\text{no failure})} \\
&= \frac{\mathbb{P}(\tilde{A}_j = 1)}{1 - \mathbb{P}(\tilde{A}_n = 0)} \\
&= \frac{1 - \prod_{i=c+1}^{j} \mathbb{P}(X_i \geq \gamma)}{1 - \prod_{j=c+1}^{n} \mathbb{P}(X_j \geq \gamma)} \\
&= \frac{1 - \prod_{i=c+1}^{j} \left(1 - \frac{\gamma-1}{n}\right)}{1 - \prod_{j=c+1}^{n} \left(1 - \frac{\gamma-1}{n}\right)} \\
\Leftrightarrow\ \mathbb{E}[\tilde{A}_j|\tilde{A}_n = 1] &= \frac{1 - \left(1 - \frac{\gamma-1}{n}\right)^{j-c}}{1 - \left(1 - \frac{\gamma-1}{n}\right)^{n-c}}
\end{aligned}$$

□

Chapter 5

The Warm-starting Dynamic Thresholding algorithm

In Chapter 4, the proposed *Cutoff-based Cost Minimization* algorithm makes no assumption on the distribution of candidates' scores. It splits the process into two phases: the *learning phase* in which the DM learns an acceptance threshold from candidates that are getting rejected by default, and thereafter, the *selection phase* which uses that threshold to accept or reject from the rest candidates. Despite its biased rule, which is due to the disregard of the early-arriving candidates, the strategy can be efficient and robust to score changes, provided an appropriately set learning phase size.

Important to note, when the score distribution is known, i.e. in a *full-information* case, the cutoff-based strategies are not optimal and policies derived from dynamic programming are more suitable. In Nikolaev and Sofronov [87], dynamic programming is employed to compute the optimal stopping rule for each candidate, i.e. the score threshold to beat in order to be hired. Every candidate's threshold depends on his *arrival time* and on the number of jobs positions left to be filled.

In this chapter, we propose the *Warm-starting Dynamic Thresholding* (WDT) algorithm that attributes to each incoming candidate a threshold value to beat according to: i) its arrival time, ii) the current number of empty job positions, and iii) the current number of positions occupied by initial employees (i.e. available employees that can be replaced). The threshold value to beat is computed by means of dynamic programming, adapted to the warm-starting scenario. The algorithm is easy to implement and gives each candidate a chance to be hired. WDT's downside lies in the assumption that the score distribution is known. We relax this requirement by first considering the *partial information* case where the DM only knows the nature of the distribution, and then the *no-information* case where the threshold is purely rank-based.

5.1 Setting and rules of the game

As described in Sec. 3.3.1, $r \leq b$ positions are initially empty as some employees have become permanently unavailable, while the rest $b-r$ are initially occupied by existing employees that form a set called *preselection*. Available and unavailable employees constitute the *reference set* for the DM. For notation simplicity, and because in this strategy we get no use of the scores of the unavailable employees, we consider only the preselection scores given by $\dot{\mathbf{S}} = (\dot{S}_{(1)}, \dot{S}_{(2)}, ..., \dot{S}_{(b-r)})$. For convenience, the

latter vector is considered to be sorted in decreasing order, i.e. $\dot{S}_{(1)}$ and $\dot{S}_{(b-r)}$ are respectively the best and the worst scores of the preselected employees. Once the j-th candidate arrives, the DM observes his score $S_j \in \mathbb{R}$, which, like all scores, is assumed to follow a known distribution f_S. The rules of the game that characterize this setting are similar to those of the rank-based setting (Assumption 1 and Assumption 2), except for the observations that are now observed.

R1. Each decision (i.e. hire or not) shall be immediate and irrevocable, which means that every position can be assigned (or reassigned) at most once throughout the selection process.

R2. The DM must at least fill the empty job positions by hiring r candidates.

R3. Once no empty position is left, the DM can still hire an incoming candidate by removing one of the employees of the preselection from his position (only fire on hire).

The following definition is a score-based adaptation of the Warm-starting SSP (WSSP) (see Sec. 3.3), where here the DM initially knows also the distribution from which candidates' scores are drawn.

Definition 10. *A particular* distribution-aware *GSSP, with the following characteristics:*

1. *Background*

   $\mathcal{B} = (n, b, f_S, \dot{\mathbf{S}})$*, where the included elements are:*
   - $n \in \mathbb{N}^*$*: finite number of candidates to appear;*
   - $b \in \mathbb{N}^*$*,* $b \leq n$*: number of resources;*
   - f_S*: the distribution generating candidates' scores;*
   - $\dot{\mathbf{S}} = (\dot{S}_{(1)}, ..., \dot{S}_{(b-r)}) \in \mathbb{R}^{b-r}$*;* $\dot{S}_{(j)} \sim f_S$*,* $\forall j$*: scores of the preselection.*

2. *Sequential arrivals and 3. Decision Process as in Definition 1 where* $S_j \sim f_S$*,* $\forall j$*;*

4. *Score-based evaluation*

   Let P *be the distribution of* $(\mathbf{S}, \mathbf{A})$*. The evaluation criterion, called* reward*, is evaluated at the end of the process and defined as* $\Phi(\pi) = \mathbb{E}_P[\phi_{\mathcal{B}}(\mathbf{S}, \mathbf{A})]$*, with:*

   The reward *is evaluated at the end of the process by:*

$$\phi_{\mathcal{B}} = \dot{\mathbf{S}}^\mathsf{T} \dot{\mathbf{A}}_n + \mathbf{S}^\mathsf{T} \mathbf{A} \in \mathbb{R}_+, \tag{5.1}$$

   where $\dot{A}_{n,(i)} \in \{0, 1\}$ *indicating if the* i*-th preselected employee kept his position after* n *candidate interviews.*

Note that, since the DM is forced to fill the b positions by the end of the process, it holds: $\|\dot{\mathbf{A}}_n\|_1 + \|\mathbf{A}\|_1 = b$.

5.2 Algorithm description

Here, we present the *Warm-starting Dynamic Thresholding* (WDT) method to solve optimally the distribution-aware WSSP of Definition 10. Without loss of generality,

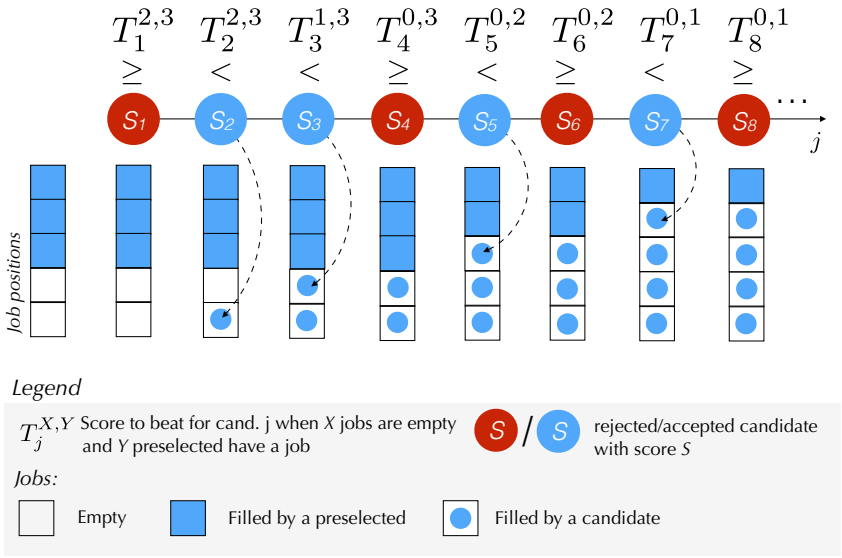


FIGURE 5.1: Demonstration of the WDT algorithm. The score of an incoming candidate, S_j, is compared to an associated acceptance threshold $T_j^{X,Y}$, where X and Y are respectively the current number of empty positions and that of positions occupied by preselected employees. Accepted candidates (blue circle) first fill empty positions and then, if they are competitive enough, take a preselected employee's positions (e.g. the 5-th candidate).

we consider non-negative i.i.d. scores $S_j \geq 0, \forall j$, and that the best- (resp. worst-) skilled individual has the highest (resp. lowest) score.

5.2.1 Threshold-based strategy

The idea behind the strategy is to find the optimal *acceptance threshold* $S^{\text{acc.}}j \in \mathbb{R}$ that the j-th arriving candidate should beat in order to be hired, $\forall j \in \{1,...,n\}$; see the example of Fig. 5.1. Note that, in order to be optimal, this threshold must depend on the state of the ongoing selection process. Therefore, we write $T_j = T_j^{X_j,Y_j}$, where $X_j \in \{0,...,r\}$ positions are still empty and $Y_j \in \{0,...,b-r\}$ jobs are still occupied by preselected employees, i.e. after $j-1 \in \mathbb{N}^*$ interviews and while the j-th candidate is being interviewed. Using the notations of Definition 10 we get:

$$X_j := \max\left(r - \sum_{i=1}^{j-1} A_i,\ 0\right) \quad \text{and} \quad Y_j := \sum_{i=1}^{r} \dot{A}_{j,(i)}. \tag{5.2}$$

To simplify our notations, we omit the dependency of X_j and Y_j to j, and we simply write X and Y to refer to their value at the implied step j of the selection process.

Value function

The fundamental question remains how to compute the thresholds optimally. The solution can be found via dynamic programming where the *value function*, which is the expected value of the regret here, is computed for each possible scenario. Let $V_j^{X,Y} \in \mathbb{R}$ be this value function when $j-1 \in \mathbb{N}^*$ candidates have been interviewed so far.

Before going into the method's details, we can deduce the following remarks from the rules that constrain the process:

Remark 8. *The second rule, R2, implies that if $X \geq n-j$, then $V_j^{X,Y} = 0, \forall Y$. Thus, the last incoming candidates might be accepted by default, and this leads to $X = 0$ when $j = n$.*

Remark 9. *R3 implies that if $X \neq 0$, then $Y = b-r, \forall j$.*

Remark 10. *From R2 and R3 we deduce that X and Y are non-increasing with j: throughout the process the number of job positions assigned to candidates cannot decrease, while those for the preselected employees cannot increase.*

To compute $V_j^{X,Y}$ we work by means of backward induction. First, we consider the extreme case that n candidates are automatically rejected, and compute the value function $V_n^{r,b-r}$. Here, either $X = 0$ and $r = 0$, i.e. each position is occupied by a preselected employee, or $V_n^{r,b-r} = 0$ because one or more positions are empty. The induction then goes to the first non-zero value function, which is when $n-r$ candidates get automatically rejected: there, the only option is to hire the r last candidates to arrive (see Remark 8), hence:

$$V_{n-r+1}^{r,b-r} = \mathbb{E}\left[\sum_{j=n-r+1}^{n} S_j + \sum_{i=1}^{b-r} \dot{S}_{(i)}\right] = r\mu + \sum_{i=1}^{b-r} \dot{S}_{(i)}, \tag{5.3}$$

where μ is the mean of the known score distribution f_S.

One step back, $V_{n-r}^{r,b-r}$ is evaluated by accounting every option: either the $(n-r)$-th candidate gets rejected and the last r candidates get hired, leading to the reward of $V_{n-r+1}^{r,b-r}$ (Eq. 5.3), or the $(n-r)$-th candidate gets hired and in this case we should, again, present two options. The DM can lower by one either the number of empty job positions, or the number of preselected employees that will keep their positions at the end of the selection. This reasoning is generalized to the following recurrent value functions $\forall j \in \{1, ...n\}$:

$$V_j^{X,Y} = \mathbb{E}\left[\max\left(V_{j+1}^{X,Y},\ S_j + \max(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1})\right)\right]. \tag{5.4}$$

The first term in the outer $\max(\cdot)$ corresponds to the option of rejecting the j-th candidate, while the second term is the option of accepting him. Two observations can be made: i) if the goal is to minimize instead of maximize the objective, then $\max(\cdot)$ should be replaced by $\min(\cdot)$ functions; ii) results of [87], and specifically their Theorem 3 in Sec. 3, can be retrieved by setting $Y = 0$ that implies $V_{j+1}^{X,Y-1} = 0$.

Recurrence relation

Thanks to our backward induction formulation, we enunciate a generic formula for the recurrence relation of the value function.

Proposition 7. *Let a WSSP with a population of i.i.d. scores $S \in [\alpha, \beta]$, each drawn from a known distribution f_S with cumulative distribution function $F_S(x) = \int_\alpha^x f_S(y)dy$. Having processed $j-1$ interviews, $X \in \{0, ..., r\}$ job positions are empty*

Algorithm 3 Warm-starting Dynamic Thresholding (WDT)

Input: the evaluation table of the value function $V_j^{X,Y}$ where b, r, and n are the numbers of resp. job positions, initially empty among them, and sequentially incoming candidates; $\dot{\mathbf{A}}_0 = (\dot{A}_{0,(1)},...,\dot{A}_{0,(b-r)}) = (1,...,1)$ is the initial status of the preselected employees.
Output: the set of final job assignments $\dot{\mathbf{A}}_n \in \{0,1\}^{b-r}$, and $\mathbf{A} \in \{0,1\}^n$.

$X \leftarrow r$ // nb. of empty jobs
$Y \leftarrow b-r$ // nb. of jobs occupied by preselected employees
for $j = 1$ to n **do**
  $T_j \leftarrow V_{j+1}^{X,Y} - \max(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1})$ // see Proposition 8
  if $S_j > T_j$ **then**
    $A_j \leftarrow 1$ // accept candidate
    if $X > 0$ **then**
      $X \leftarrow \max(X-1, 0)$ // fill one empty position
    else
      $\dot{A}_{j,(Y)} \leftarrow 0$ // remove job from a preselected employee
      $Y \leftarrow Y - 1$
    end if
  else
    $A_j \leftarrow 0$
  end if
end for

and $Y \in \{0,...,b-r\}$ *are occupied by preselected employees with scores* $\dot{\mathbf{S}} = (\dot{S}_{(1)},...,\dot{S}_{(b-r)})$. *Then, the recurrence relation of Eq. 5.4 becomes:*

$$V_j^{X,Y} - V_{j+1}^{X,Y} = Z_{j+1}^{X,Y}\big(F_S(Z_{j+1}^{X,Y}) - 1\big) + \int_{Z_{j+1}^{X,Y}}^{\beta} s f_S(s) ds, \tag{5.5}$$

where $Z_{j+1}^{X,Y} := V_{j+1}^{X,Y} - \max(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1})$, *with initial condition given in Eq. 5.3.*

The proof can be found in the Appendix 5.6

Optimal threshold. Evidently, the j-th candidate must be accepted if the value function associated to the option of accepting him is larger than that of rejecting him. According to Eq. 5.4, the latter amounts to choosing the optimal threshold defined in the following proposition that maximize the expectation of the reward.

Proposition 8. *Let a WSSP where* $j-1$ *interviews have been processed,* $X \in \{0,...,r\}$ *job positions are empty, and* $Y \in \{0,...,b-r\}$ *are occupied by preselected employees. The optimal acceptance threshold for candidate* j *is defined as:*

$$T_j^{X,Y} = V_{j+1}^{X,Y} - \max(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1}). \tag{5.6}$$

Essentially, the j*-th candidate is accepted if his score beats the corresponding threshold, i.e.* $A_j = \mathbb{1}\{S_j > T_j^{X,Y}\}$.

The WDT procedure works with the optimal threshold described above and is described in Alg. 3.

Y	X	$V_1^{X,Y}$	$V_2^{X,Y}$	$V_3^{X,Y}$	$V_4^{X,Y}$	$V_5^{X,Y}$	$V_6^{X,Y}$	$V_7^{X,Y}$
0	1	0.893	0.886	0.879	0.871	0.861	0.850	0.836
	2	1.719	1.702	1.683	1.661	1.636	1.606	1.571
1	0	0.907	0.902	0.897	0.891	0.885	0.877	0.869
	1	1.756	1.742	1.729	1.712	1.694	1.673	1.650
	2	2.547	2.523	2.496	2.465	2.431	2.391	2.345

Y	X	$V_8^{X,Y}$	$V_9^{X,Y}$	$V_{10}^{X,Y}$	$V_{11}^{X,Y}$	$V_{12}^{X,Y}$	$V_{13}^{X,Y}$	$V_{14}^{X,Y}$
0	1	0.823	0.800	0.775	0.741	0.768	0.732	0.682
	2	1.529	1.476	1.409	1.320	1.195	1.000	0.000
1	0	0.859	0.847	0.833	0.816	0.795	0.979	0.979
	1	1.621	1.588	1.547	1.495	1.428	1.333	1.182
	2	2.290	2.224	2.142	2.036	1.894	1.682	0.000

TABLE 5.1: Evaluation of the $V_j^{X,Y}$ function at each step $j=1,...,14$. The distribution of the scores is $S_j \sim \mathcal{U}(0,1)$, and there is $b-r=3-2=1$ initially non-empty job position occupied by an employee with score $\dot{S}_{(1)}=0.682$.

5.3 Full, partial, and no-information settings

5.3.1 Some closed-form in full-information setting

For some specific score distributions, the cumulative and probability density functions are easily computed, and allow the recurrence relation to get simplified. In the following proposition, scores are drawn from an exponential distribution.

Proposition 9. *Set $S_j \sim Exp(\lambda), \forall j \leq n$. Then, Proposition 7 becomes:*

$$V_j^{X,Y} = V_{j+1}^{X,Y} + \frac{e^{-\lambda Z_{j+1}^{X,Y}}}{\lambda}, \tag{5.7}$$

where $Z_{j+1}^{X,Y} = V_{j+1}^{X,Y} - \max(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1})$.

Consider now a setting where scores are drawn uniformly from (α, β).

Proposition 10. *Set $S_j \sim \mathcal{U}(\alpha,\beta), \forall j \leq n$. Then, Proposition 7 becomes:*

$$V_j^{X,Y} - V_{j+1}^{X,Y} = \frac{{Z_{j+1}^{X,Y}}^2 - 2\alpha Z_{j+1}^{X,Y} + \beta^2}{2(\beta-\alpha)} - Z_{j+1}^{X,Y}, \tag{5.8}$$

where $Z_{j+1}^{X,Y} = V_{j+1}^{X,Y} - \max(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1})$.

Example. Set $\alpha=0$ and $\beta=1$, $b=3$, $r=2$, and $n=14$. Tab. 5.1 displays the evaluation of the value function $V_j^{X,Y}$ at each step of the selection process. Recall that, at the respective step j, $X \in \{1,...,b\}$ is the number of empty job positions, and $Y \in \{0,1\}$ is the DM's hiring decision. Now, imagine the following sequence of scores: $\mathbf{S} = (0.498, 0.858, 0.749, 0.398, ...)$ and $\dot{\mathbf{S}} = (0.682)$ is the score of the preselected employee. We are determining the acceptance threshold sequentially. Note

that, since $r \neq 0$, $V_j^{0,0} = 0$. For the first incoming candidate, $j = 1$, $r = 2$ job positions are empty, i.e. $X = 2$ and $b - r = 1$ position is occupied by a preselected employee, i.e. $Y = 1$. The first candidate is rejected since $T_1^{2,1} = V_2^{2,1} - \max(V_2^{1,1}, V_2^{2,0}) = 2.523 - \max(1.742, 1.702) = 0.781 > 0.498$. The second threshold reads $T_2^{2,1} = V_3^{2,1} - \max(V_3^{1,1}, V_3^{2,0}) = 2.496 - \max(1.729, 1.683) = 0.767 < 0.858$, which allows the acceptance of the second candidate. The following threshold is then, $T_3^{1,1} = 0.832$, thus the third candidate is rejected. The process continues this way until the sequence is finished.

5.3.2 Rank-based strategy in no-information setting

When the score distribution is either unknown (no-information case), does not exhibit a closed-form cumulative density function, or scores are simply not observed, the DM can only rely on a relative evaluation of the candidates. In practice, she can assign a relative rank to each incoming candidate by comparing him to those already examined (let the best one be ranked first).

In that case, the acceptance threshold is *rank-based*, hence $T_j^{X,Y}$ stands for the *absolute rank* (which cannot be known, though) that the j-th candidate needs to exceed to get selected. Conveniently, the absolute ranks of a set follow a discrete uniform distribution that exhibits a closed-form description. Then, the threshold value for the j-th candidate is computed using Proposition 8 and, following the same reasoning as in Proposition 10, we get the following simplified expression:

$$V_j^{X,Y} = V_{j+1}^{X,Y} - \frac{{Z_{j+1}^{X,Y}}^2 - Z_{j+1}^{X,Y}}{2(n+b-r)}, \tag{5.9}$$

where $Z_{j+1}^{X,Y} = V_{j+1}^{X,Y} - \min(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1})$.

As mentioned, the DM cannot know the absolute rank of a candidate before finishing all interviews. She can still, though, estimate it knowing his *relative rank* and by taking into account the proportion of candidates that has already been examined. More precisely, the j-th candidate has relative rank denoted by $Z_j^{\text{rel}} \in \mathbb{N}^*$ after the examination of $j + b - r$ individuals (including the preselected employees), and the absolute rank denoted by Z_j^{abs} that he would have after the examination of $n+b-r$ individuals. Hence, we set $Z_j^{\text{rel}} = \frac{j+b-r}{n+b-r} Z_j^{\text{abs}}$ and, thereby, the practical threshold of relative rank that the j-th candidate must exceed to be accepted is:

$$\tau_j^{X,Y} = \frac{j+b-r}{n+b-r} T_j^{X,Y}. \tag{5.10}$$

Example. Set $b = 3$, $r = 2$, and $n = 14$. Tab. 5.2 displays the evaluation of the value function $V_j^{X,Y}$ at each step of the selection process. Consider a setting where the DM does not observe each item's s cores, but can only proceed to pairwise comparison, and for which the best (respectively worse) item has an absolute rank 1 (respectively $n + b - r = 16$). Suppose that the two available employees have average rank of $\dot{S}_{(1)} = \dot{S}_{(2)} = (n + b - r + 1)/2 = 8.5$. While arriving, the first candidate $j = 1$ is ranked compared to those two; and should have a relative rank lower than $\tau_1^{1,2} = \frac{1+3-1}{14+1-1} (V_1^{1,2} - \min(V_2^{0,2}, V_2^{1,1})) = \frac{3}{14}(9.387 - 5.554) = 0.821$ to get accepted, which is not possible, therefore he is automatically rejected. Similarly,

X	Y	$V_1^{X,Y}$	$V_2^{X,Y}$	$V_3^{X,Y}$	$V_4^{X,Y}$	$V_5^{X,Y}$	$V_6^{X,Y}$	$V_7^{X,Y}$
0	1	2.261	2.361	2.476	2.606	2.758	2.935	3.146
	2	5.558	5.825	6.127	6.474	6.875	7.344	7.903
1	0	2.243	2.341	2.453	2.580	2.727	2.899	3.103
	1	5.318	5.554	5.820	6.121	6.466	6.865	7.332
	2	9.387	9.823	10.31	10.87	11.51	12.24	13.10

X	Y	$V_8^{X,Y}$	$V_9^{X,Y}$	$V_{10}^{X,Y}$	$V_{11}^{X,Y}$	$V_{12}^{X,Y}$	$V_{13}^{X,Y}$	$V_{14}^{X,Y}$
0	1	3.402	3.717	4.119	4.649	5.388	6.508	8.500
	2	8.579	9.415	10.48	11.90	13.89	17.00	25.50
1	0	3.349	3.651	4.034	4.535	5.225	6.250	8.000
	1	7.885	8.554	9.378	10.42	11.80	13.69	16.50
	2	14.12	15.35	16.87	18.80	21.36	25.00	33.50

TABLE 5.2: Evaluation of the $V_j^{X,Y}$ function at each step $j=1,...,14$. The distribution of the scores is $S_j \sim \mathcal{U}[1, n+b-r]$, and there are $b-r=3-1=2$ initially non-empty job position occupied by 2 employees of average rank $\dot{S}_{(1)} = \dot{S}_{(2)} = (n+b-r+1)/2 = 8.5$.

the second candidate should have a relative rank lower than $\tau_2^{1,2} = \frac{4}{14}\, 4.003 = 1.143$, therefore, he is accepted if he is better than the first candidate, and both of the preselected employees (i.e. $Z_2^{\text{rel}} = 1$), and rejected otherwise. The process goes on until the sequence is finished.

5.4 Simulations

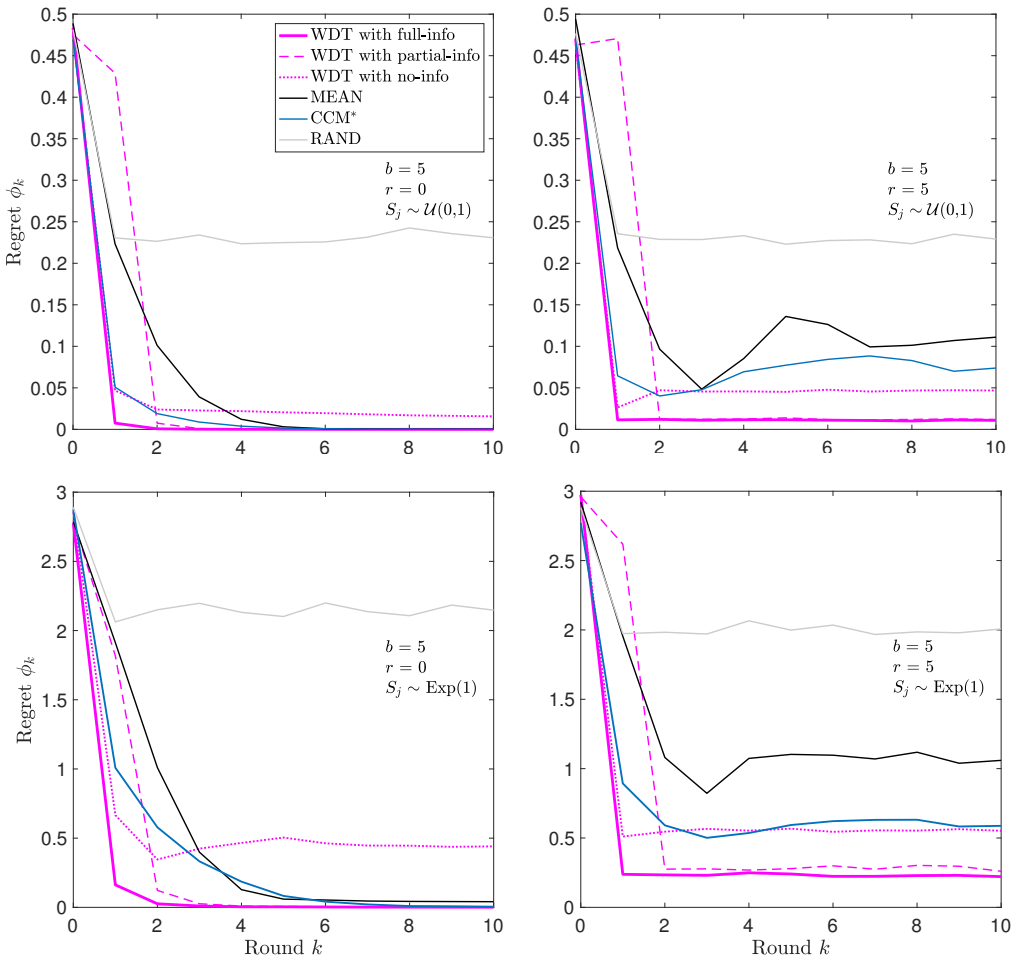
5.4.1 Simulations parameters

The WSSP setting is similar to that of Sec. 4.4 i.e. it takes as input the reference set (containing available and unavailable employees) and the sequence of candidates, and outputs a selection of size b. As stated before, a very interesting feature of this configuration is that it enables multi-round applications where the output selection of a round can be fed as input for the consecutive round.

In the simulations, a population of $N = 10000$ job-seekers is considered, and $n = 100$ interviews take place in each of the $K = 10$ considered rounds. The candidates' scores are drawn from a given distribution and remain fixed during the process. Again, the preselected employees of the first round are chosen uniformly at random from the population, and hence, carry an average quality score. We desire to compare our online strategy, the WDT, to the best an offline strategy achieves, i.e. in the case that the DM could examine the candidates altogether as a batch. Therefore, instead of the reward, in the figures we plot the regret defined as $\phi_k = |\phi_{\text{off},k} - \phi_k|, \forall k \leq K$, where $\phi_{\text{off},k}$ and ϕ_k are respectively the offline and online reward.

5.4.2 Comparing score distributions

Fig. 5.2 displays the average regret ϕ_k in different settings, namely $S_{i,k} \sim \mathcal{U}(0,1)$ (top row) and $S_{i,k} \sim \text{Exp}(1), \forall i,k$ (bottom row). Let us start by focusing on the plain line curves, one of which is the RAND baseline (grey line) that decides for the hires



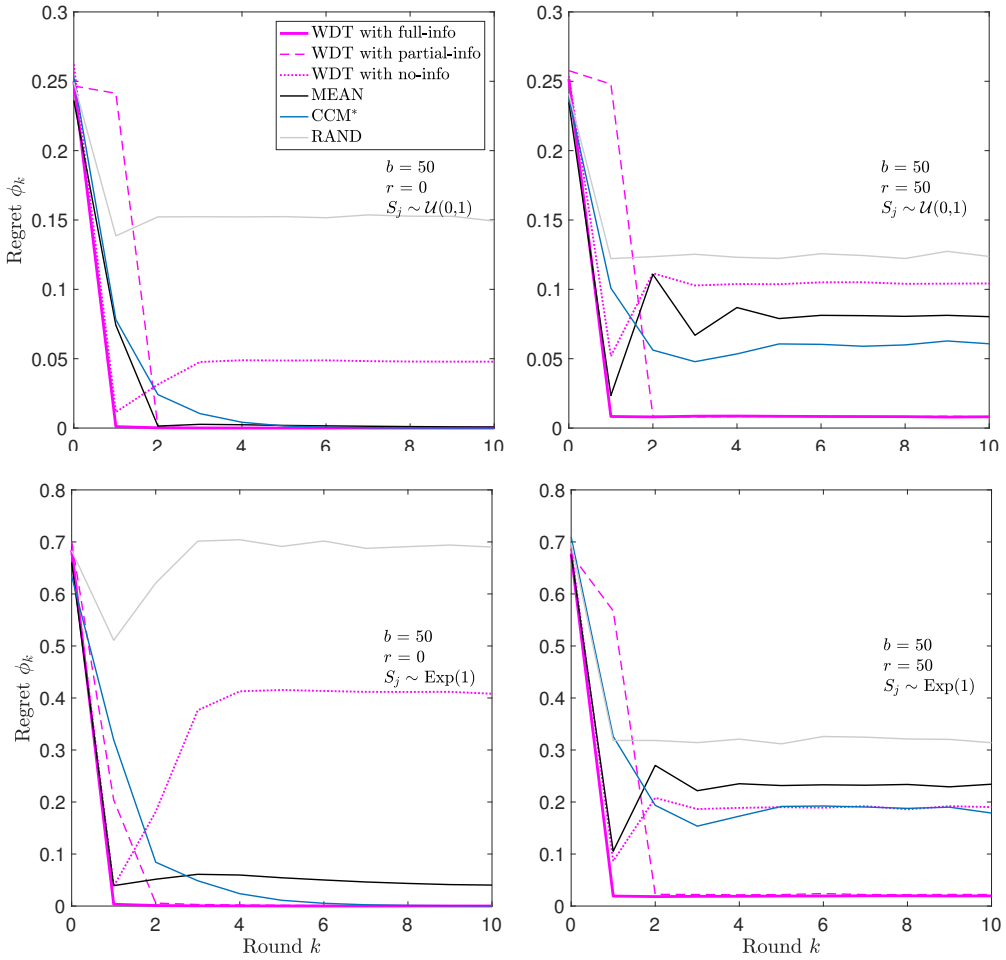
(c) no resignations ($r = 0$) (d) max resignations ($r = b$)

FIGURE 5.2: Average regret ϕ_k versus the round number k. The number of job positions is $b = 5$ for $n = 100$ candidates per round. Score distribution: uniform, $S_{i,k} \sim \mathcal{U}(0,1)$ (top row); exponential, $S_{i,k} \sim \text{Exp}(1), \forall i, \forall k$ (bottom row). The number of initially empty job positions is $r = 0$ (left) and $r = b = 5$ (right). Magenta lines use WDT in full-information (plain line), partial-information (dashed line) and no-information (dotted line) settings.

at random. A first straightforward observation is that the average regret has similar inefficient behavior for both distributions. In the following description, we therefore focus on the uniform distribution.

Secondly, the subfigures on the left assume $r = 0$, hence the process always starts with b empty positions, whereas on the right it is assumed $r = b$, thus the process starts with b positions occupied by preselected employees. In the first case, since the employees do not quit their position in-between two subsequent rounds, the DM cannot deteriorate the selection, and might even improve the set of employees through time by replacing initial employees with more skilled candidates. The regret naturally goes to zero, and it does go faster for the proposed WDT than for instance the MEAN (see Sec. 2.3.2) or CCM* (see Sec. 4.2) strategies. In the second case ($r = 0$), the regret does not necessarily converge towards zero; neither MEAN nor CCM* manage to keep a low average regret. This phenomenon is explained as follows: progressively, the average score of the employees of the previous round gets

so competitive for the incoming candidates that it forces the DM to select the last-arriving ones by default to prevent ending up with having empty positions after all the interviews (see the rule R2 in Sec. 6.2).



(c) no resignations ($r = 0$) (d) max resignations ($r = b = 50$)

FIGURE 5.3: Average regret ϕ_k w.r.t. the round number k. The number of job positions is $b = 50$ for $n = 100$ candidates per round. The score distribution is uniform s.t. $S_{i,k} \sim \mathcal{U}(0,1) \forall i, \forall k$. The number of initial job positions is $r = 0$ (left) and $r = b = 50$ (right). Magenta lines use WDT in full-information (plain), partial-information (dash) and no-information (dot).

5.4.3 Comparing information levels

In Fig. 5.2, we also show the performance of the algorithms for cases with different levels of DM's knowledge for the score distribution f_S. In the *full-information* case (plain line), f_S is perfectly known. In the *partial information* case (dashed line), DM knows only the shape of the f_S (e.g. uniform, normal, exponential, etc.) and needs to learn its parameters (e.g. lower and upper bounds, mean, etc.). Finally, the *no-information* case (dotted line) is when the DM does not hold any information about f_S (e.g. the shape, as discussed in the example of Sec. 5.3.1), or not even a way to compute an absolute score per candidate. In that case, the DM should rather rely on relative ranks that are re-assessed after the examination of each individual (see Sec. 5.3.1).

We observe that, provided the DM knows the shape of the score distribution (i.e. partial-information), the learning of its parameters throughout the rounds is relatively fast, and the regret quickly converges to that of full-information (plain lines). Here, the strategy is much slower in converging towards the full-information case, which is achieved for $r=0$ after approximately 40 rounds. The slow convergence is due to the DM's inability to estimate each candidate's absolute rank before having examined all other candidates (see Eq. 5.10).

5.4.4 Large budget size

In Fig. 5.3 are displayed the average regret when the number of job positions is quite large compared to the number of incoming candidates, i.e. $b=50$ positions for $n=100$ candidates. Although the regret behaves similarly as for $b=5$, each algorithm goes faster towards a small regret when $r=0$, that is because the sequential selection problem is less difficult when the DM has a large budget compared to the number of candidates to interview. Regardless, the strategy WDT still manages to maintain a good selection through rounds.

5.5 Conclusion

In this chapter we presented a new algorithm, called *Warm-starting Dynamic Thresholding* (WDT), for the Warm-starting SSP (WSSP), considering the case where the incoming candidates have scores following a known distribution. The proposed algorithm is based on a dynamic programming approach and achieves optimal threshold estimation at each step of the sequence of interviewed candidates. Experiments have been performed in the multi-round setting, which is interesting for real-world reoccurring recruitment processes. WDT demonstrated a clearly better performance than existing algorithms, regardless the number of initially empty job positions. We additionally proposed a rank-based dynamic programming alternative that can go beyond the need of knowing perfectly the distribution that generates the scores, yet, resulting in satisfying outcomes.

After a analytical study of the optimal algorithm when candidate scores are observed, in the next chapter we investigate the first application of such warm-starting processes, i.e. the Sequential allocation of resources for epidemic control.

5.6 Appendix – Technical proofs

For a fixed distribution $S_j \in [\alpha,\beta]$, $\forall j \leq n$, and $\mu = \mathbb{E}[S_j]$.

Proposition 11. *Let a WSSP with a population of i.i.d. scores $S \in [\alpha,\beta]$, each following a given distribution f_S. Set $F_S(x) = \int_\alpha^x f_S(x)$ to be its cumulative distribution function. The situation is as follows, $j-1$ interviews have been processed, $X \in \{0,...,r\}$ job positions are empty, and $Y \in \{0,...,b-r\}$ are occupied by preselected employees. The recurrence relation in Eq. 5.4 becomes:*

$$V_j^{X,Y} - V_{j+1}^{X,Y} = Z_{j+1}^{X,Y}\left(F_S(Z_{j+1}^{X,Y}) - 1\right) + \int_{Z_{j+1}^{X,Y}}^{\beta} s f_S(s) ds,$$

where $Z_{j+1}^{X,Y} := V_{j+1}^{X,Y} - \max(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1})$.

Proof. Let each candidate have an i.i.d. score $S_j \in \mathbb{R}, \forall j \leq n$. The objective function to maximize is the sum of the scores of the individuals with a job position at the end of the selection.

We have:

$$\begin{aligned}
V_j^{X,Y} &= \mathbb{E}[\max\left(V_{j+1}^{X,Y}, S_j + \max(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1})\right)] \\
V_j^{X,Y} &= V_{j+1}^{X,Y}\,\mathbb{P}\big(V_{j+1}^{X,Y} - \max(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1}) \geq S_j\big) \\
&\quad + \max(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1})\mathbb{P}\big(V_{j+1}^{X,Y} - \max(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1}) < S_j\big) \\
&\quad + \int_{V_{j+1}^{X,Y} - \max(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1})}^{\beta} s f_S(s) ds
\end{aligned}$$

Set $Z_{j+1}^{X,Y} := V_{j+1}^{X,Y} - \max(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1})$, and $F_S(x) = \int_\alpha^x f_S(x)$ the cumulative density function:

$$\begin{aligned}
V_j^{X,Y} &= V_{j+1}^{X,Y} F_S(Z_{j+1}^{X,Y}) + \max(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1})(1 - F_S(Z_{j+1}^{X,Y})) + \int_{Z_{j+1}^{X,Y}}^{\beta} s f_S(s) ds \\
V_j^{X,Y} &= V_{j+1}^{X,Y} - Z_{j+1}^{X,Y} + Z_{j+1}^{X,Y} F_S(Z_{j+1}^{X,Y}) + \int_{Z_{j+1}^{X,Y}}^{\beta} s f_S(s) ds \\
V_j^{X,Y} &= V_{j+1}^{X,Y} + Z_{j+1}^{X,Y}\left(F_S(Z_{j+1}^{X,Y}) - 1\right) + \int_{Z_{j+1}^{X,Y}}^{\beta} s f_S(s) ds.
\end{aligned}$$

□

Recurrence relation for a uniform distribution $S_j \sim U(\alpha, \beta), \forall j \leq n, \mu = (\alpha+\beta)/2$.

Proposition 12. *Set $S_j \sim \mathcal{U}(\alpha, \beta), \forall j \leq n$. Proposition 11 becomes:*

$$V_j^{X,Y} - V_{j+1}^{X,Y} = \frac{{Z_{j+1}^{X,Y}}^2 - 2\alpha Z_{j+1}^{X,Y} + \beta^2}{2(\beta - \alpha)} - Z_{j+1}^{X,Y},$$

where $Z_{j+1}^{X,Y} = V_{j+1}^{X,Y} - \max(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1})$.

Proof. We have $F_S(s) = \frac{s-\alpha}{\beta-\alpha}$ and $f_S(s) = \frac{1}{\beta-\alpha}$:

$$V_j^{X,Y} = V_{j+1}^{X,Y} + Z_{j+1}^{X,Y}\left(\frac{Z_{j+1}^{X,Y}-\alpha}{\beta-\alpha} - 1\right) + \frac{1}{\beta-\alpha}\int_{Z_{j+1}^{X,Y}}^{\beta} s ds$$

$$V_j^{X,Y} = V_{j+1}^{X,Y} - Z_{j+1}^{X,Y} + \left(\frac{{Z_{j+1}^{X,Y}}^2 - \alpha Z_{j+1}^{X,Y}}{\beta-\alpha}\right) + \frac{\beta^2 - {Z_{j+1}^{X,Y}}^2}{2(\beta-\alpha)}$$

$$V_j^{X,Y} = V_{j+1}^{X,Y} - Z_{j+1}^{X,Y} + \frac{{Z_{j+1}^{X,Y}}^2}{2(\beta-\alpha)} + \frac{\beta^2 - 2\alpha Z_{j+1}^{X,Y}}{2(\beta-\alpha)}$$

$$V_j^{X,Y} = V_{j+1}^{X,Y} - Z_{j+1}^{X,Y} + \frac{{Z_{j+1}^{X,Y}}^2 - 2\alpha Z_{j+1}^{X,Y} + \beta^2}{2(\beta-\alpha)}.$$

□

Recurrence relation for a discrete uniform distribution $S_j \sim U\{\alpha,\beta\}$, $\forall j \leq n$, $\mu = (\alpha+\beta)/2$. The objective is to minimize the sum of the scores, that can be seen as ranks.

$$V_{n-r+1}^{r,b-r} = \mathbb{E}[\sum_{j=n-r+1}^{n} S_j + \sum_{i=1}^{b-r} \dot{S}_{(i)}] = r\mu + \sum_{i=1}^{b-r} \mathbb{E}[\dot{S}_{(i)}]$$

Note that here, $Z_{j+1}^{X,Y} = V_{j+1}^{X,Y} - \min(V_{j+1}^{X-1,Y}, V_{j+1}^{X,Y-1})$. Hence:

$$V_j^{X,Y} = V_{j+1}^{X,Y}(1 - F_S(Z_{j+1}^{X,Y})) + (V_{j+1}^{X,Y} - Z_{j+1}^{X,Y})F_S(Z_{j+1}^{X,Y}) + f_S\frac{Z_{j+1}^{X,Y}(Z_{j+1}^{X,Y}+1)}{2}$$

$$V_j^{X,Y} = V_{j+1}^{X,Y} - Z_{j+1}^{X,Y}F_S(Z_{j+1}^{X,Y}) + f_S\frac{{Z_{j+1}^{X,Y}}^2 + Z_{j+1}^{X,Y}}{2}.$$

More specifically, for ranks between $\alpha = 1$ and $\beta = n+b$, we have $F_S(k) = \frac{k}{n+b}$, and $f = \frac{1}{n+b}$, thus:

$$V_j^{X,Y} = V_{j+1}^{X,Y} - \frac{{Z_{j+1}^{X,Y}}^2}{n+b} + \frac{1}{n+b}\frac{{Z_{j+1}^{X,Y}}^2 + Z_{j+1}^{X,Y}}{2}$$

$$V_j^{X,Y} = V_{j+1}^{X,Y} - \frac{{Z_{j+1}^{X,Y}}^2 - Z_{j+1}^{X,Y}}{2(n+b)}.$$

Recurrence relation for an exponential distribution $S_j \sim Exp(\lambda)$, $\forall j \leq n$, $\mu = 1/\lambda$.

We have, $\alpha = 0$, $\beta = +\infty$, $F_S(s) = 1 - e^{-\lambda s}$ and the truncated expectation is s.t.:

$$\int_{Z_{j+1}^{X,Y}}^{\beta} s f_S(s) ds = \int_{Z_{j+1}^{X,Y}}^{+\infty} s\lambda e^{-\lambda s}$$
$$\int_{Z_{j+1}^{X,Y}}^{\beta} s f_S(s) ds = Z_{j+1}^{X,Y} e^{-\lambda Z_{j+1}^{X,Y}} + \frac{e^{-\lambda Z_{j+1}^{X,Y}}}{\lambda}.$$

Hence:

$$V_j^{X,Y} = V_{j+1}^{X,Y} - Z_{j+1}^{X,Y} e^{-\lambda Z_{j+1}^{X,Y}} + Z_{j+1}^{X,Y} e^{-\lambda Z_{j+1}^{X,Y}} + \frac{e^{-\lambda Z_{j+1}^{X,Y}}}{\lambda}$$
$$V_j^{X,Y} = V_{j+1}^{X,Y} + \frac{e^{-\lambda Z_{j+1}^{X,Y}}}{\lambda}.$$

Table of notations for Chapter 6

Symbol	Description
$\mathbb{1}\{\text{condition}\}$	indicator function that gives 1 if the condition is true, otherwise 0
$\mathcal{G}, \mathcal{V}, N, \mathcal{E}, E$	network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ of $N = \lvert\mathcal{V}\rvert$ nodes and $E = \lvert\mathcal{E}\rvert$ edges, where $\mathcal{V}$, $\mathcal{E}$ are the sets of nodes and edges
$\mathbf{M} \in \mathbb{R}^{N \times N}$	network's adjacency matrix s.t. $M_{ij} \neq 0$ if node i is linked with an edge to node j
$\mathbf{X}_t \in \{0,1\}^N$	infection state vector s.t. $X_{i,t} = 1$ if node i is infected at time t
$N_t^I \in \mathbb{N}^*$	number of infected nodes at time t
$b \in \mathbb{N}^*$	budget of control actions, i.e. resources to allocate to nodes
$\mathbf{R}_t \in \{0,1\}^N$	resource vector at time t s.t. $R_{i,t} = 1$ if a treatment is allocated to node i at time t subject to $\lvert\lvert\mathbf{A}\rvert\rvert = b$
β, δ, ρ	parameters of the DP
$s : \mathcal{V} \to \mathbb{R}$	scoring function s.t. $S_{i,t} \in \mathbb{R}$ is the score of node i at time t
$T \in \mathbb{R}_+$	time-horizon
$A_N(T) \in \mathbb{R}_+$	percentage of infected nodes from time $t = 0$ to time $t = T$
$\dot{\mathbf{C}}_t \subset \mathcal{V}$	treated nodes at time t, i.e. $R_{\dot{C}_i,t} = 1, \forall i, \forall t$
$k \leq K$	round index s.t. t_k is the time at which the k-th round occurs, and K is the total number of rounds
$\pi(\mathcal{I}, \mathcal{A}) \in \{0,1\}^n$	control strategy for which the DM has information on the nodes of the set $\mathcal{I}$, and access to the nodes of the set $\mathcal{A}$
$n \leq N, \mathbf{C} \subset \mathcal{V}$	node sample of size n at time t s.t. C_1 is the first incoming candidate and C_n the last of a round
$\Lambda(c; n, x) \in \{0,1\}$	probability of observing sample c of size n for a given infection state x
$\phi \in \mathbb{R}_+$	cost function to minimize
$\mathbf{R}^{\text{off}} \in \{0,1\}^N$	allocation vector that minimizes the cost for full access and info on the sample i.e. $\mathcal{I} = \mathcal{A} = \mathbf{C}$, subject to $\lvert\lvert\mathbf{R}^{\text{off}}\rvert\rvert = b$
$\dot{\mathbf{S}}, \mathbf{S}^{\mathbf{C}} \subset \mathbf{S} = s(\mathcal{V})$	scores of the treated nodes, and of the candidate nodes respectively
$\Delta N_t^I \in \mathbb{R}_+$	difference between the expected number of infected nodes using an online strategy and the corresponding offline strategy
$e \in \mathbb{R}$	online error, i.e. the expected sum of half false negatives (FN) and false positives (FP) compared to an offline strategy
$t_{\text{exct.}}$	first time at which every nodes are healthy
$\alpha \in [0,1]$	fraction of the infected nodes that compose the sample
$c \in \mathbb{N}$	integer that specifies when to stop rejecting candidates called cutoff using the CCM strategy
f_S	distribution of the network nodes scores

TABLE 5.3: Index of main notations.

Chapter 6

Application to Epidemic Control

6.1 Introduction

The motivation of our work is to bring the score-based DRA modeling closer to reality. In the majority of real-life scenarios, authorities have access to limited information regarding the network state, and can reach a limited part of the population to apply control actions (e.g. deliver treatments). Even more importantly, the decision making process is essentially a sequence of time-sensitive decisions over choices that appear and remain available to the administrator only for short time, also with little or no margin of revocation. An intuitive paradigm to consider is how a healthcare unit works: patients arrive one-by-one seeking for care, and online decisions try to assign the limited available resources (e.g. medical experts, beds, treatments) to the most important medical cases [15, 61, 53]. By establishing a link between the DRA problem and the sequential decision making literature, our work offers a completely new perceptive to dynamic DP control. Among the existing *Sequential Selection Problems* (SSP) that have been widely studied, the most well-known is the Secretary Problem [47]. Our aim, however, is to propose a concordant match to the DP control setting described above.

Concerning our technical contribution, we first present the *Restricted DRA* (RDRA) model, in which each time the administrator can decide the reallocation of the resources only among a random sample of currently reachable nodes. On top of that, we next propose the special case of *Sequential DRA* (SDRA) where the latter sample of nodes is provided with a random *arrival order*, forcing the administrator to decide for the resource reallocation sequentially according to the characteristics of the incoming nodes. We believe that the major achievement of our modeling is that it manages to create a new playground where SSP algorithms can be incorporated to the DP control, and this way makes control strategies more applicable in real conditions. The implementation of existing online algorithms such as MEAN [21] or CCM (see Chapter 4) leads to SDRA strategies that manage to reduce the DP in a comparable fashion to the unrestricted DRA.

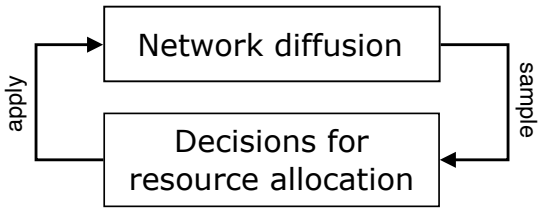


FIGURE 6.1: Scheme of the network diffusion control problem.

6.2 Dynamic Resource Allocation strategies

6.2.1 Network diffusion process

The interactions among a population of N individuals are modeled by a fixed network represented by a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ of $|\mathcal{V}|=N$ nodes and $|\mathcal{E}|=E$ edges. The graph structure is arbitrary; the reader might picture it as being a directed or non-directed, weighted or unweighted graph, etc. Each entry M_{ij} of the graph's adjacency matrix $\mathbf{M}\in\mathbb{R}^{N\times N}$, expresses the possible influence of node i on node j, s.t. $M_{ij}\neq 0$ if node i is linked with an edge to node j and may have an impact on him, and vice-versa.

The graph hosts an *agent-based diffusion process* (DP) that spreads from one node to another. Conceptually, the sequential epidemic control framework that we present, can apply to arbitrary DPs (even non-compartmental diffusion models), provided that there is a means to assess the criticality of the graph elements (e.g. nodes) for reducing the epidemic process. To simplify the presentation, we use a setting for which such criticality assessment has been made possible in the literature. We suppose that the DP in place is a continuous-time Markov process [86], so that at each time instance $t\in\mathbb{R}_+$ there can be at most one event of node state change in the network. More specifically, in this work we consider an SIS-like recurrent epidemic, where nodes are either healthy ('S': susceptible to infection), or infected ('I'). The infection spreads from any infected node to its reachable healthy neighbors. Nodes are equipped with self-recovery without ever achieving permanent immunity. The infection state of the network is denoted by $\mathbf{X}_t=(X_{1,t},...,X_{N,t})^\intercal\in\{0,1\}^N$, s.t. $X_{i,t}=1$ if node $i\in\mathcal{V}$ is infected and 0 otherwise. In the rest of the paper, $\bar{X}_{i,t}=1-X_{i,t}$, and $N_t^I=\sum_{i=1}^N X_{i,t}$ stands for the number of infected nodes in the network at time t.

Remark 11. *A similar study can be done with epidemics that exhibits more than two infection states, for instance the SIR-like epidemic, in which nodes are healthy, infected, or recovered ('R'), and in the latter nodes are perpetually cured, or dead.*

6.2.2 Resource allocation for controlling a DP

A DM has the mission to reduce the DP by managing a fixed budget of $b\in\mathbb{N}^*$ resources that help the receiving nodes leaning towards the healthy state. The resources are regarded as being reusable, non-storable through time, and non-cumulable at nodes (i.e. at most one on each node). Resources might serve as treatments, doctors, nurses, beds, etc. in an emergency service, in which allocation decisions have to be made on-the-go. Minimizing the total number of resources required for removing a particularly viral epidemic from a network [28], or investigating the link between the number of resources and the expected time of extinction [35, 36] are the main subject of a couple of papers. However, here we assume that, despite $b\ll n$, distributing resources thoughtfully can achieve the extinction of the epidemic in a finite time horizon. In other words, there exists an allocation strategy that manages to extinguish the epidemic with a given set of parameters and hence the budget size is not the main focus here.

At each time instance $t\in\mathbb{R}_+$, the *Dynamic Resource Allocation* (DRA) [96, 97] dynamically determines the resource allocation vector $\mathbf{R}_t=(R_{1,t},...,R_{N,t})^\intercal\in\{0,1\}^N$ where $R_{i,t}=1$ if a treatment is allocated to node i at time t and 0 otherwise; subject

to $\sum_i R_{i,t} \leq b$. The node state transitions of the *homogeneous SIS model under control* are given by:

$$\begin{aligned} &\text{(healthy to infected)} \quad X_{i,t}: 0 \to 1 \text{ at rate } \beta \textstyle\sum_j M_{ij} X_{j,t}; \\ &\text{(infected to healthy)} \quad X_{i,t}: 1 \to 0 \text{ at rate } \delta + \rho R_{i,t}, \end{aligned} \tag{6.1}$$

where β is the contribution of any edge to the infection rate from an infected towards a healthy node, δ is the self-recovery rate of each infected node, and ρ is the contribution of a received treatment (if $R_{i,t} = 1$) to the node recovery rate. If we let λ_i be the transition rate for node i in Eq. 6.1, then the mechanism generating DP events (infection or recovery) is a Poisson process and the probability distribution of the time intervals between events is given by:

$$p(\Delta t, \lambda) = \lambda e^{-\lambda \Delta t}, \ \forall \Delta t \in \mathbb{R}_+, \tag{6.2}$$

where $\lambda = \sum_{i=1}^{N} \lambda_i$ is the sum of all node transition rates. Then, each node $i \in \mathcal{V}$ can be the transitioning node with probability equal to λ_i / λ. Using the formalism of [7], we write as $p_{vu}(\Delta t) := \mathbb{P}(N^I_{t+\Delta t} = v \,|\, N^I_t = u)$ the probability of going from u to v number of infected nodes in a time interval Δt, and we get:

$$p_{vu}(\Delta t) = \begin{cases} b(u)\Delta t + o(\Delta t) & v = u+1 \\ d(u)\Delta t + o(\Delta t) & v = u-1 \\ 1 - (b(u) + d(u))\,\Delta t + o(\Delta t) & v = u \\ o(\Delta t) & v \notin \{u+1, u, u-1\}, \end{cases} \tag{6.3}$$

$$\text{where } b(u) := b(N^I_t = u) = \beta \textstyle\sum_{ij} M_{ij} \bar{X}_{i,t} X_{j,t}, \tag{6.4}$$

$$d(u) := d(N^I_t = u) = \textstyle\sum_i (\delta + \rho R_{i,t}) X_{i,t}. \tag{6.5}$$

Note that, the resource allocation vector is constant within a time-interval Δt. For continuous-time, $\Delta t \to 0$, and by using the Markov property $\mathbb{P}(N^I_{t+\Delta t} \mid N^I_0, ..., N^I_t) = \mathbb{P}(N^I_{t+\Delta t} | N^I_t)$, the forward Kolmogorov differential equations are found for the probability of having n infected nodes at time t, denoted by $p_u(t)$:

$$\frac{dp_u(t)}{dt} = p_{u-1} b(u-1) + p_{u+1} d(u+1) - p_u (b(u) + d(u)), \tag{6.6}$$

for $u = 1, 2, ..., N$, and $dp_0/dt = p_1 d(1)$. The evolution of the stochastic variables of the problem are then derived from this standard equation. In particular, by multiplying by n and summing over n, the equation that rules the evolution of the epidemic becomes:

$$\frac{d\mathbb{E}[N^I_t]}{dt} = \beta\, \mathbb{E}[\mathbf{X}^\mathsf{T} \mathbf{M} \bar{\mathbf{X}}] - \delta \mathbb{E}[N^I_t] - \rho \mathbb{E}[\mathbf{X}^\mathsf{T} \mathbf{R}], \tag{6.7}$$

where $\mathbb{E}[\mathbf{X}^\mathsf{T} \mathbf{R}] = \min(b, N^I_t)$. In an attempt to solve this equation analytically in the following section we consider the random assignment of the budget among the infected nodes as resource allocation strategy.

6.2.3 The dynamics of the SIS epidemic model with naïve/random allocation strategy

As presented before, the random variable of interest in the stochastic SIS epidemic model is the number of infected nodes at time t, N_t^I, more precisely its expectation $\mathbb{E}[N_t^I]$ for which we attempt to derive a closed-form equation. Let us start by considering the following assumptions.

1. The graph is a random Erdös-Rényi (ER) where every node has approximately the same degree, i.e. $\bar{k} \approx k_i := \sum_{j=1}^{N} M_{ij}$, $\forall i \in \mathcal{V}$.
2. The allocation of resources is coarse-grained, i.e. we apply the RAND strategy for which the b nodes to receive a treatment are chosen at random among the population of infected nodes.
3. When $b > N_t^I$, the few remaining infected nodes can 'accumulate' more than one resource to increase their recovery rate until $N_t^I = 0$.

Note that those assumptions are usually not verified in real cases, but are taken as a simple starting point for the analysis. By respecting the above constraints, we multiply Eq. 6.6 by u, sum over u, and obtain:

$$\begin{aligned}
\frac{d\mathbb{E}[N_t^I]}{dt} &= \beta\, \mathbb{E}[\mathbf{X}^\intercal \mathbf{M} \bar{\mathbf{X}}] - \delta \mathbb{E}[N_t^I] - \rho \mathbb{E}[\mathbf{X}^\intercal \mathbf{R}] \\
\frac{d\mathbb{E}[N_t^I]}{dt} &= \beta \mathbb{E}[\textstyle\sum_i \sum_j M_{ij} X_{j,t} \bar{X}_{i,t}] - \mathbb{E}[\sum_i (\delta + \rho R_{i,t}) X_{i,t}] \\
\frac{d\mathbb{E}[N_t^I]}{dt} &= \beta\, \mathbb{E}[\textstyle\sum_i \bar{k} \frac{N_t^I}{N} \bar{X}_{i,t}] - \delta \mathbb{E}[N_t^I] - \rho b \\
\frac{d\mathbb{E}[N_t^I]}{dt} &= \frac{\beta \bar{k}}{N}\, \mathbb{E}[N_t^I (N - N_t^I)] - \delta \mathbb{E}[N_t^I] - \rho b,
\end{aligned}$$

that finally leads to the evolution of the first-order moment:

$$\frac{d\mathbb{E}[N_t^I]}{dt} = (\beta \bar{k} - \delta)\, \mathbb{E}[N_t^I] - \frac{\beta \bar{k}}{N}\, \mathbb{E}[(N_t^I)^2] - \rho b. \tag{6.8}$$

We now multiply Eq. 6.6 by u^2, sum over u, and similarly obtain the dynamic equation for the second-order moment:

$$\begin{aligned}
\frac{d\mathbb{E}[(N_t^I)^2]}{dt} &= (\beta \bar{k} + \delta - 2\rho b)\, \mathbb{E}[N_t^I] - 2\frac{\beta \bar{k}}{N}\, \mathbb{E}[(N_t^I)^3] \\
&\quad + \left(2(\beta \bar{k} - \delta) - \frac{\beta \bar{k}}{N}\right) \mathbb{E}[(N_t^I)^2] + \rho b.
\end{aligned} \tag{6.9}$$

It is easy to check that every moment equation depends on a higher order moment, and thus the closed-form equation for the evolution of the expected number of infected nodes cannot be computed, even in the simple coarse-grained allocation case. In order to get some results, a first option is to use a moment closure technique to approximate the highest-order term, as in [81]. The two most common approximations are the following:

$$\text{Normal: } \mathbb{E}[(N_t^I)^3] = 3\mathbb{E}[(N_t^I)^2]\mathbb{E}[(N_t^I)] - 2(\mathbb{E}[(N_t^I)])^3 \tag{6.10}$$

$$\text{Lognormal: } \mathbb{E}[(N_t^I)^3] = \left(\frac{\mathbb{E}[(N_t^I)^2]}{\mathbb{E}[(N_t^I)]}\right)^3. \tag{6.11}$$

A second option is to bound the expected number of infected nodes by using the fact that $\mathbb{E}[(N_t^I)^2] \geq \mathbb{E}[N_t^I]^2$, and thus that the mean of the stochastic SIS epidemic model is less than that of the deterministic solution.

The conclusion is that, even in this simple case the closed-form equation of the $\mathbb{E}[N_t^I]$ cannot be retrieved, unless restrictive assumptions are introduced. This thesis primarily aims at showing the practical consequences of choosing different allocation strategies on arbitrary networks, and therefore we make no assumptions on the structure of the graph or on the expected evolution of the spread.

6.2.4 Scoring function

The *score-based DRA* assumes that there exists a scoring function $s : \mathcal{V} \to \mathbb{R}$ that returns a score $S_{i,t} \in \mathbb{R}$ for each node i at time t, according to the mission, and the nodes with the highest scores are those to receive the resources. This class of strategies depends on the size of the available budget of resources and the efficiency of each of them, as well on the ability of the scoring function in assessing correctly the criticality of nodes.

The evaluation of the performance of a score-based strategy at time horizon $T \in \mathbb{R}_+$ is the expected area under the curve (AUC) of the percentage of infected nodes w.r.t. time:

$$A_N(T) = \int_0^T \frac{\mathbb{E}[N_t^I]}{N} dt \ \in \mathbb{R}_+, \tag{6.12}$$

By making this choice we acknowledge that the $\mathbb{E}[A_N(T)]$ is more characteristic for the effect of a strategy than, for example, the expected extinction time, $\mathbb{E}[t_{\text{exct.}}]$.

6.3 Sequential Dynamic Resource Allocation

The standard DRA strategies are build on a strong assumption whereby the DM always has full information about the process and full access to the network, which is apparently far from being realistic in most practical cases. To reduce this distance, we introduce two models in the following sections. Their generality stems from the fact that they assume the scoring function s to be a 'black-box' (hence out of our main research focus here) that is appropriate for the studied network process. In this context, this means that it is efficient in determining the criticality of an infected node when asked. It then remains to the algorithmic part of a strategy to take the decisions of resource reallocations to nodes that would be as much as possible valued by the function s.

6.3.1 Restricted DRA

In the *Restricted DRA* (RDRA) model, only a fraction of nodes are reachable at each moment. Let $\mathcal{I}$ be the set of nodes for which we have information and $\mathcal{A}$ the set of accessible nodes, with $\mathcal{I}, \mathcal{A} \subseteq \mathcal{V}$. We put forward two reasonable assumptions: A1) the accessible nodes are always included in the set of nodes for which we have information, i.e. $\mathcal{A} \subseteq \mathcal{I}$, and A2) at any time $t \in \mathbb{R}_+$, the treated nodes, $\dot{\mathbf{C}}_t = \{i \in \mathcal{V} :$

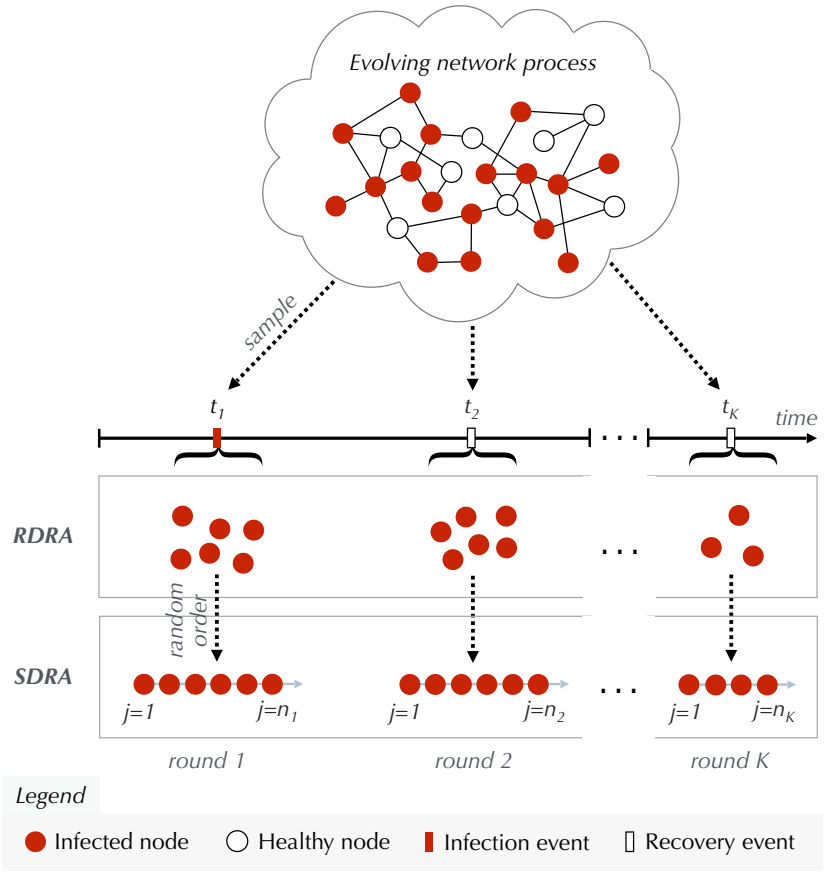


FIGURE 6.2: The sequential evaluation of candidates in the SDRA model.

$R_{i,t}=1\}$ are accessible. The first notion to define for specifying the RDRA model is that of a *round*.

Definition 11. *Round k is a discrete event of reviewing and revising the allocation of resources on the network. The series of rounds is defined by the sequence of time instances $(t_k)\in\mathbb{R}_+^K$, where $t_0=0$, $t_K=T$, and K is the total number of rounds.*

Remark 12. *A round-triggering process is considered to invoke the revision of the resource allocation. In this thesis we restrict ourselves to a passive process, i.e. the $(t_k)'s$ are not decided or known to the DM in advance. However, an active process can be an interesting addition to the RDRA strategy.*

Generally, two subsequent rounds can be arbitrarily distant in time; i.e.

$$t_k=t_{k-1}+\delta t,\ \ \forall k\leq K,\ \forall\delta t>0. \tag{6.13}$$

During that time interval the allocation budget remains fixed and hence can become suboptimal or even 'wasted' when resources stay on healthy nodes. This thesis though, mainly deals with the impact of the allocation strategy on the evolution of the spread. Hence the need to separate the effect of distant subsequent rounds from the effect of a poor choice of resource assignment by a bad allocation strategy. Here, more specifically, a new round is triggered whenever there is a change in the infection state of the network, and thus, at most one node can recover or get infected between two subsequent rounds. Formally, this is described by the following condition on

the allocation events:

$$t_k = t_{k-1} + \delta t^*, \ \forall k \leq K, \tag{6.14}$$

where $\delta t^* := \min(\delta t \,|\, \|\mathbf{X}_{t+\delta t} - \mathbf{X}_t\|_2 = 1)$. This assumption gets relaxed in Sec. 6.6.6 where more than one event can separate two rounds.

Since rounds are essentially a measure of time, each variable can be defined round-wise, for instance we write $\mathbf{X}_k$ for the infection state at round k, i.e. at the time instance t_k.

Definition 12. *The Restricted DRA (RDRA) strategy $\pi_k(\mathcal{I}, \mathcal{A})$ is a DRA strategy parametrized by the number of resources $b \in \mathbb{N}^*$ and a scoring function $s : \mathcal{V} \to \mathbb{R}$. At any moment during a round k, the DM has information about the nodes of the set $\mathcal{I}$ and simultaneous access to the nodes of the set $\mathcal{A}$, in addition to those currently treated, $\dot{\mathbf{C}}_k$, more specifically: $\dot{\mathbf{C}}_k \subseteq \mathcal{A}$. The strategy outputs a resource allocation vector, i.e. $\pi_k = \mathbf{R}_k, \forall k \leq K$.*

We consider as the default RDRA setting when $\mathcal{I} = \mathcal{A} = \mathbf{C}_k$, where $\mathbf{C}_k$ is called *sample* and is defined below. Choosing to define $\mathcal{A}$ or $\mathcal{I}$ otherwise, results in special RDRA cases, while choosing $\mathcal{I} = \mathcal{A} = \mathcal{V}$ degenerates to standard DRA.

Definition 13. *Node sample $\mathbf{C}_k$ is the set of accessible infected nodes at time t_k, $\mathbf{C}_k = (C_{1,k}, ..., C_{n_k,k}) \subset \mathcal{V}$. Its size $n_k = f(\mathbf{X}_k) \in \mathbb{N}^*$ is given by $f : \{0,1\}^N \to \mathbb{N}^*$, a function of the infection state. The probability of observing a sample $c \subset \mathcal{V}$, given its size $n \in \mathbb{N}^*$ and the network state $x \in \{0,1\}^N$, is $\Lambda_k(c; n, x) = \mathbb{P}(\mathbf{C}_k = c \,|\, |\mathbf{C}_k| = n, \mathbf{X}_k = x)$. In short we write $\mathbf{C}_k \sim \Lambda_k(n, \mathbf{X}_k)$.*

Typically, the number of accessible nodes is proportional to the number of infected nodes in the graph, i.e. $n_k = \lfloor \alpha \sum_i X_{i,k} \rfloor$, $\alpha \in [0,1]$. Similarly to the round triggering process, although the size of the sample is known by the DM, it it still not of her choice but it is generated. In a crisis scenario for instance, more candidate nodes might ask for healthcare as the public awareness arises from watching the media. Take as examples these two simple sampling functions, whereas more complicated ones can be considered:

- $\mathbb{P}(i \in \mathbf{C}_k) = \frac{n_k}{N} \mathbb{1}\{X_{i,k} = 1\}$, i.e. the sampling is uniform among the population of infected nodes;
- $\mathbb{P}(i \in \mathbf{C}_k) = \frac{e^{S_{i,k}}}{\sum_i e^{S_{i,k}}} \mathbb{1}\{X_{i,k} = 1\}$, i.e. nodes with high scores have a highest probability to be sampled.

The RDRA's assumption for simultaneous access to all the nodes of a sample in a round remains far from being realistic. We next refine the access constraints and present a model that processes the sample sequentially.

6.3.2 Sequential DRA

The *Sequential DRA* (SDRA) model does not reassign altogether the b treatments as DRA and RDRA strategies do. Here, the round is further divided into n_k time intervals and the reallocation is performed sequentially at the time-scale of the round duration. In each time interval one candidate of the sample is examined for getting a treatment. Let the discrete index $j \in \{1, ..., n_k\}$ characterize the sequential arrival order of candidates, e.g. $j = 1$ and $j = n_k$ are respectively the first and last candidates

Algorithm 4 DP control with Restricted and Sequential DRA

Input: N: population size; b: budget of resources; $\mathbf{X}_0$: initial infection state; $\mathbf{p}(x)$: transition probability from state x to every other state; f: function that gives the number of accessible nodes; Λ: p.d.f.of the sample; π: Restricted DRA strategy; *isSequential*: specifies if the strategy is RDRA (false) or SDRA (true).
Output: $\mathbf{X}$: final network state, $\mathbf{R}$: final allocation of the resources

$\mathbf{X} \leftarrow \mathbf{X}_0$ // initialize infection state
$\mathbf{R}(\text{randp}(b,N)) \leftarrow 1$ // initialize resource allocation
while $\text{sum}(\mathbf{X}) \neq 0$ **do**
  $n \leftarrow f(\mathbf{X})$ // compute the number of accessible nodes
  $\dot{\mathbf{C}} \leftarrow \text{find}(\mathbf{R} = 1)$ // currently treated nodes
  $\mathbf{C} \sim \Lambda(n, \mathbf{X})$ // generate a node sample
  if *isSequential* == true **then**
    for $j = 1...n$ **do** // loop of a selection round
      $\mathbf{R} \leftarrow \pi(\dot{\mathbf{C}}, C_j)$ // update resource allocation sequentially
    end for
  else
    $\mathbf{R} \leftarrow \pi(\dot{\mathbf{C}}, \mathbf{C})$ // update resource allocation altogether
  end if
  $\mathbf{X} \leftarrow \mathbf{p}(\mathbf{X})$ // update infection state
end while
return $\mathbf{X}$, $\mathbf{R}$

of round k. Since the administrator gains access sequentially to candidates, the problem variables can depend on the index $j \in \{1, ..., n_k\}$.

Definition 14. *The* Sequential DRA *(SDRA)* $\pi_k(\mathcal{I}, \mathcal{A})$ *is the RDRA strategy defined by the sequence* $\pi_k(\mathcal{I}, \mathcal{A}) = (\pi_{k,1}(\mathcal{I}_1, \mathcal{A}_1), ..., \pi_{k,n_k}(\mathcal{I}_{n_k}, \mathcal{A}_{n_k})) \in \{0,1\}^{n_k}$ *where* $\mathcal{I}_j = (C_1, ..., C_j)$ *and* $\mathcal{A}_j = C_j$, $\forall j \leq n_k$, *providing a uniformly random arrival order of the nodes of the sample.*

The way the RDRA and SDRA models operate is described in Alg. 4, and an deployment example is depicted in Fig. 6.2.

6.4 From DP control to a Multi-round Sequential Selection Process

6.4.1 Link with the Sequential Selection Problem

To the best of our knowledge, this work is the first to cast the dynamic DP control as a problem where decisions are taken as in a Sequential Selection Problem (SSP).

We map the problem of DP control with SDRA to a succession of separate *Warm-starting SSPs* (WSSPs) which were introduced in Definition 3. For convenience, in these notations we drop the round subscript k within each WSSP, e.g. $\mathbf{C}_k$ becomes $\mathbf{C}$.

Definition 15. *A particular* distribution-aware *GSSP (see Definition 15), with the following characteristics:*

1. *Background*

   $\mathcal{B} = (n, b, s, \dot{\mathbf{C}})$, *where the included elements are:*
   – $n \in \mathbb{N}^*$: *number of candidates to come,*

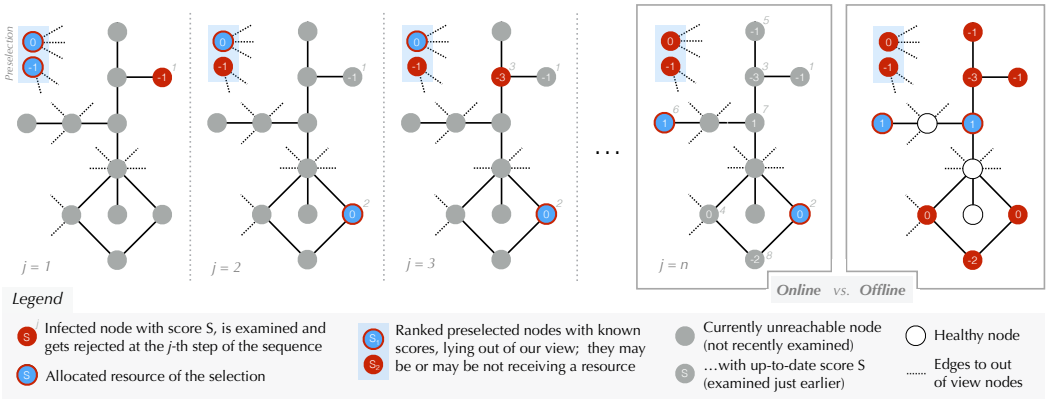


FIGURE 6.3: Example of the 3 first steps of an SDRA round. Candidates, from a sample of $n=8$ infected nodes, appear sequentially w.r.t. j and possible reallocations are decided immediately. For instance, at step $j=2$, the examined candidate receives a resource that is withdrawn from the second preselected node (lies out of the main graph view). The two rightmost figures compare the result of the online selection to that of the offline selection used by RDRA.

- $b \in \mathbb{N}^*$: *fixed budget of resources,*
- $s : \mathcal{V} \to \mathbb{R}$: *scoring function s.t.* $\mathbf{S} = s(\mathcal{V}) \in \mathbb{R}^N$ *is the score vector of the entire population.*
- $\dot{\mathbf{C}} = (\dot{C}_1, ..., \dot{C}_b) \subset \mathcal{V}$: *the subset of the population, called* preselection*, to which resources are initially allocated when a round begins, i.e.* $R_{\dot{C}_i} = 1, \forall i \leq b$.

2. *Sequential arrivals*
   - $(C_1, ..., C_n) \in \mathcal{P}_n(\mathcal{V} \backslash \dot{\mathbf{C}})$: *sequence of randomly incoming candidates for receiving a resource, where* $\mathcal{P}_l(E)$ *denotes the set of* l*-combinations of some finite set* E
3. *Decision Process*
   - $(R_{C_1}, ..., R_{C_n}) \in \{0,1\}^n$: *sequence of resource allocation decisions taken; giving a resource to a candidate immediately withdraws it from a preselected node (recovered or not), i.e.* $R_{C_j} = 1 \Rightarrow \exists i \leq b$ *s.t.* $R_{\dot{C}_i} = 0$.
4. *Score-based evaluation*

   Let P *be the distribution of* $(\mathbf{S}, \mathbf{R})$*. The evaluation criterion, called* regret*, is evaluated at the end of the process and defined as* $\Phi(\pi) = \mathbb{E}_P[\phi_{\mathcal{B}}(\mathbf{S}, \mathbf{R})]$*, with:*

   The reward *is evaluated at the end of the process by:*

$$\phi_{\mathcal{B}} = \mathbf{S}^\mathsf{T}\mathbf{R}^{\text{off}} - \mathbf{S}^\mathsf{T}\mathbf{R} \in \mathbb{R}_+, \tag{6.15}$$

   where $\mathbf{R}^{\text{off}} = \text{argmax}_{\mathbf{R}}\{\mathbf{S}^\mathsf{T}\mathbf{R} \mid R_i = 0, \forall i \notin \mathcal{A}, |\mathbf{R}| = b\}$.

The first term of the cost function in Eq. 6.15 defines the highest achievable score, while the second one gives the score achieved by the taken sequential decisions. As the sequence of incoming candidates is a random variable, $\mathbb{E}[\phi_{\mathcal{B}}/b]$ is the objective function to minimize.

Some observations have to be made concerning our specific SDRA-to-WSSP mapping: 1) It translates the objective of the DP control (i.e. to minimize the percentage of infected nodes through time) into an SSP objective (i.e. to minimize the expected

cost function of the selected items), hence, $\eta_t = \mathbb{E}[\frac{1}{N}\sum_i X_{i,t}]$ is closely related to $\mathbb{E}[\phi_{\mathcal{B}}/b]$. 2) During each WSSP instance, the administrator does not need to know anything about the infection state of the network, and merely selects online. 3) The final selection of a round may not exactly constitute the preselection of the following one, specifically in case of recoveries of nodes that will not be any more part of the preselection. Their resources are returned and become available for the candidates of the round. 4) If the administrator has still unassigned resources while reaching the end of the sequence of a round (for the triggering function we considered, this can be at most one – see Remark 12 and below it), then they are by default given to the very last candidates to appear.

6.4.2 SSP algorithms for DP control

In Sec. 6.4.1 we argued about plugging online algorithms from the SSP literature into the SDRA model to sequentially control DPs. Here we focus on the algorithmic part, in particular on two classes of online strategies:

- *Cutoff-based*: such a strategy takes as input a *cutoff value* $c \in \mathbb{N}$; it first rejects by default the first c incoming candidates, in a *learning phase*, and then selects a candidate according to information collected during the first phase.
- *Threshold-based*: a particular case of *cutoff-based* strategies with $c = 0$. A candidate is accepted if his score beats a specified *acceptance threshold*.

We chose an indicative algorithm from each class: the *Cutoff-based Cost Minimization* (CCM) (see Chapter 4) and the *Hiring-Above-the-Mean* (MEAN) [21] (see Sec. **??**), whose objectives are to minimize the expected sum of the ranks (or respectively, the expected sum of scores) of the selected nodes at the end of a round.

Cutoff-based Cost Minimization. The CCM algorithm is thoroughly explained in Chapter 4 and is, for convenience, briefly explained again here. It takes as input a measure of *quality* $q \in]0,1[$ of the preselected nodes w.r.t. the sample. This measure indicates how worthy the current resource allocation is. A high quality suggests that the currently treated nodes are among the network's most critical nodes for the epidemic spreading. For this strategy, the administrator needs to value the selection in terms of ranks instead of scores.

Let $\dot{\mathbf{S}} = (s(\dot{C}_1), ..., s(\dot{C}_b)) \subset \mathbf{S}$ be the scores of the candidates and $\mathbf{S^C} = (s(c_1), ..., s(C_n)) \subset \mathbf{S}$ the scores of the preselected nodes. Using Definition 2, the rank-based cost function is defined as $\phi^{\sigma}_{\mathcal{B},k} = \sum_{S \in \mathcal{S}} R_{n+b}(S, (\dot{\mathbf{S}}, \mathbf{S^C}))$, where $\mathcal{S} = \{S_j \in \mathbf{S} | R_j = 1\}_{1 \le j \le N}$. In this thesis, we set $q_k = \phi^{\sigma}_{\mathcal{B},k-1}/b, \forall k \le K$, where $\phi_{\mathcal{B},0} = 0.5$. This way, the quality of the preselected nodes in the k-th round is simply the sum of their ranks w.r.t. to the sample of the previous round, when they were selected. This implies our assumption that the item ranks are rather constant between two subsequent rounds. Then, the table $c^*(b,n,q) \in \mathbb{N}^{b \times n}$ is computed by tracking the lowest point of the expected rank-based cost provided in Chapter 4 (see Sec. 4.2).

The algorithm proceeds as follows. The b-best scores recorded during the learning phase of size c^* are stored ordered in a *reference set*. Then, the acceptance threshold starts at the worse score of the set and moves up each time a candidate is accepted, pointing at the next higher score of a non-resigned employee. The process terminates when the end of the sequence is reached. For simplicity, we refer to the CCM strategy with $c = c^*(b,n,q)$ as CCM*. Note that the rank-based evaluation is particularly

suited for the DP control, as nodes' criticality scores are most likely changing through time.

Hiring-above-the-mean. MEAN strategy [21] considers the average score of the preselection as acceptance threshold, which is updated after each new selection. In the original setting, each incoming candidate $j \leq n$ has a *quality score* $Q_j \sim \mathcal{U}(0,1)$; and the goal is to keep a good trade-off between the quality of the selection and speed of hires. Let the average quality of $b+1$ employees be denoted as A_b (it starts with $A_0 \in [0,1]$). To quantify the rate of convergence of the latter and the rate at which candidates are hired, they define a *gap* $G_b = 1 - A_b$, which converges to 0 almost surely when b goes to infinity. Among other results, they proved that after $b^{3/2}$ candidate interviews, the expected value for the mean gap for the best b candidates is $O(1/\sqrt{b})$, which makes the strategy close to optimal given this particular evaluation criterion. Despite being more intuitive and easier to implement than CCM, this strategy reaches its limit when the preselection is of poor quality with respect to the sample (and probably also with all the population of care-seekers). We can also consider another strategy presented in [21], where the acceptance threshold used is their median score, MEDIAN.

6.5 Offline vs. Online

In our DP context, a strategy is called *offline* when it deterministically selects the b-best reachable nodes and immediately assigns resources to them. As explained earlier, the notion of 'best' is given by the 'black box' scoring function $s : \mathcal{V} \to \mathbb{R}$, which prioritizes nodes independently based on their criticality for the spread. On the other side, an *online* strategy can only examine the candidate nodes one-by-one (see Definition 14).

Before going further, let us clarify that for simplicity we add the superscript 'off' to refer to the offline strategy associated to the online strategy $\mathbf{\Pi}_k(\mathcal{C})$ that is implicitly used. Also, the offline strategy defined in this way is only an indicator of which nodes would have been selected by the oracle.

The main issue that arises from the link with SSP concerns the relationship between the selection performance of an online strategy and the expected number of infected nodes. This can be rather measured as a difference in the effect at the epidemic compared to the corresponding offline strategy, namely $\Delta N_t^I = \mathbb{E}[N_t^I - N_t^{I,\text{off}}]$. The performance of an online selection strategy is quantified by its expected sum of false negatives (FN) and false positives (FP) in the sequence, hence we define the *online selection error* (or just *error*): $e_k = \frac{1}{2}\mathbb{E}[|\mathbf{R}_k - \mathbf{R}_k{}^{\text{off}}|]$. e_k is evaluated among the subsequent rounds, right after a round's selection is finalized. An *online strategy with guarantee* can therefore be defined by providing a bound on either:

i) the expected number of errors over all rounds, s.t.:

$$\frac{1}{b}\sum_{k=1}^{K} e_k \leq M_K, \text{ with } M_K \in \mathbb{R}_+, \tag{6.16}$$

ii) or the expected cost at any k-th round, $\phi_{\mathcal{B},k}$ (see Eq. 6.15), written as $\phi_k = \mathbf{S}_k \cdot (\mathbf{R}_k{}^{\text{off}} - \mathbf{R}_k)$, s.t.:

$$\mathbb{E}[\phi_k] \leq L_k, \ \forall k \leq K, \text{ with } L_k \in \mathbb{R}_+. \tag{6.17}$$

This second bound is stronger as it is given round-wise. However, it requires a certain knowledge of the score distribution. Going back to the DP, a thorough numerical analysis lead us to the formulation of the following linear regression assumption.

Assumption 4. *The expected difference in the percentage of infected nodes between an online and an offline selection strategy through time is an affine function of the online selection error though rounds, which is formulated as:*

$$\int_0^T \frac{\Delta N_t^I}{N} dt = c_1 \left(\sum_{k=1}^{K} \frac{e_k}{b} \right) + c_2, \tag{6.18}$$

where $c_1 \geq 0 \geq c_2$ are constants that depend on the sampling size α (see Sec. 6.3.1), and the epidemic parameters of the problem.

Suppose that an online strategy provides a bound on its expected number of errors over the rounds and validates Eq. 6.16. Under the Assumption 4, one can deduce the following bound on the AUC of the percentage of infected nodes:

$$\int_0^T \frac{\Delta N_t^I}{N} dt \leq c_1 M_K + c_2. \tag{6.19}$$

Suppose now that an online strategy provides a bound on the cost at round k and validates Eq. 6.17. Observe that $\mathbb{E}[\phi_k] = \mathbb{E}[\mathbf{S}_k \cdot (\mathbf{R}_k{}^{\text{off}} - \mathbf{R}_k)] \geq \mathbb{E}[S_{\min,k} \mathbf{1}_k \cdot (\mathbf{R}_k{}^{\text{off}} - \mathbf{R}_k)] = 2\mathbb{E}[S_{\min,k}]\mathbb{E}[e_k]$, where $S_{\min,k}$ is the minimum score over all candidates of round k. Thus, Eq. 6.18 in Assumption 4 becomes:

$$\int_0^T \frac{\Delta N_t^I}{N} dt \leq c_1 \left(\sum_{k=1}^{K} \frac{L_k}{2b\mathbb{E}[S_{\min,k}]} \right) + c_2. \tag{6.20}$$

A short investigation of the role of the constants c_1 and c_2 can be found in the simulations.

Example: online vs. offline selection – Fig. 6.3 displays an example of an online selection round where two resources are initially assigned to the nodes of the preselection (top-left in each subfigure). Suppose that the online strategy gives a resource to incoming nodes with score higher than the average score of the preselection, here with scores $\{0, -1\}$ (the higher, the more critical they are). Scores of other infected nodes appear when their turn in the sequence arrives and each of them gets accessed. The first candidate ($j = 1$) is not selected since his score of -1 does not beat the threshold of $\frac{1}{2}(-1+0) = -0.5$. The second candidate ($j = 2$), though, gets the resource unit from the worse preselected node. The new score threshold to beat is set to 0. The process continues, up to the last candidate ($j = n$). The final resource allocation of an offline selection strategy is also shown (rightmost subfigure). Here, the cost function for the online case is $\frac{1}{b}\phi_{\mathcal{B}} = (1+1) - (1+0) = 1$, where the first term is the highest achievable average score of the selection (i.e. the offline score), and the second term is what the online strategy achieved. Regardless the scoring function, an efficient SDRA strategy (online) should be as close as possible to the associated RDRA strategy (offline) in terms of $A_N(T)$ (see Eq. 6.12).

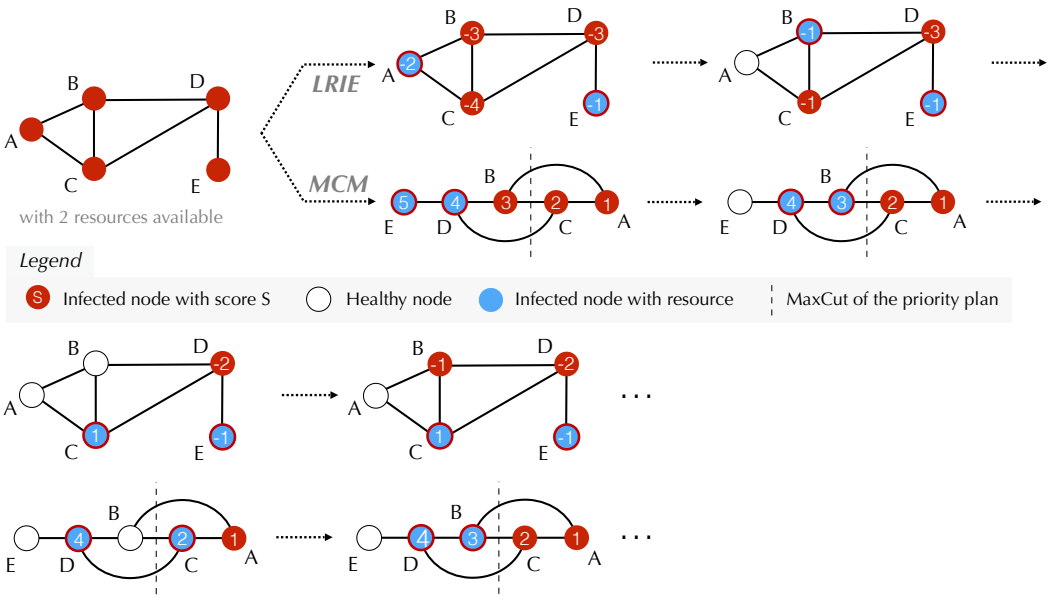


FIGURE 6.4: Example of the evolution of the network infection state when considering the dynamic scoring function LRIE and the static priority plan of MCM.

6.6 Simulations

6.6.1 Experimental setup

Network. The interactions among a population of N individuals are modeled by a fixed, symmetric (undirected), and unweighted network with adjacency matrix $\mathbf{M} \in \{0,1\}^{N \times N}$ where $M_{ij} = 1$ only if nodes i and j are linked with an edge. The connectivity structure is generated according to a *scale-free* (SF), a *small-world* (SW), a *community-structured* (CS) network model, or a real network of Facebook user-user friendships.

- In the SF type, the node degree distribution follows a power law, hence few nodes are *hubs* and have much more edges than the rest. We use the Barabási-Albert *preferential attachment* model [10] that starts with two connected nodes and, thereafter, connects each new node to $m \in \mathbb{N}^*$ existing nodes, randomly chosen with probability equal to their normalized degree at that moment.
- In the SW type, nodes are reachable to each other through short paths. We use the Watts-Strogatz model [107]: it starts by arranging the N nodes on a ring lattice, each connected to $m \in \mathbb{N}^*$ neighbors, $m/2$ on each side. Then, with a fixed probability $p \in [0,1]$ for each edge, it decides to rewire it to a uniformly chosen node of the network.
- In the CS type, nodes are grouped into sets that are densely connected internally and sparsely between groups. We use an hierarchical Erdös-Rényi model, where the probability $p_l \in [0,1]$ for creating each edge reduces at each level l as we move up in the hierarchy. At the lowest level there are 12 groups of 100 nodes, $N = 1200$ nodes overall (see Fig. 6.7).

For the two first types of generated networks, we use a small population size of $N = 100$ individuals, which however is sufficient for our demonstration. Furthermore,

by rescaling the epidemic parameters, the same phenomena can be reproduced in larger networks. The model parameters to generate each network, are mentioned explicitly in the associated figures.

Scoring function. In this section, we briefly expose a series of known scoring functions that we used in our experiments:

- *Random* (RAND): selects nodes uniformly at random, among the infected nodes.
- *Largest Reduction in Spectral Radius (LRSR)*: selects each time the infected nodes that lead to the largest drop in the first eigenvalue of the network's adjacency matrix [103].
- *Largest Reduction in Infectious Edges* (LRIE) [96]: selects the infected nodes that minimize the number of infectious edges that connect an infected and a healthy node. The associated score is the difference between the number of healthy and infected neighbors of each node:

$$S_{i,t} = \sum_j \left(M_{ij} \bar{X}_{j,t} - M_{ji} X_{j,t} \right), \tag{6.21}$$

  recall that $\bar{X}_{i,t} = 1 - X_{i,t}$, $\forall j,t$. This scoring function is derived from the minimization of the second order derivative of the expected number of infected nodes w.r.t. time: $\frac{d^2 \mathbb{E}[N_t^I]}{dt^2}$. LRIE is greedy and dynamic, since node scores change each time the infection state and/or the network structure changes.
- *Max-Cut Minimization* (MCM) [97]: A *priority planning* strategy computes a *healing plan* which is an ordering of the nodes that accounts for their criticality w.r.t. the epidemic spread, and that is given prior to the beginning of the diffusion. It is thereby a static scoring function. The resources are given to the first b nodes of the plan, and once those nodes get cured, the resources are reallocated to the next nodes of the plan. The MCM finds a node priority-order with as low as possible max-cut (see Definition 6.22). For a network with adjacency matrix $\mathbf{M} = (M_{ij}) \in \{0,1\}^{N \times N}$, the max-cut of a given priority-order ℓ is defined as:

$$CUT(\ell) = \max_{c=1,\dots,N} \sum_{i,j} M_{ij} \mathbb{1}_{\ell(i)<c<\ell(j)}, \tag{6.22}$$

  where $\ell(i)$ is the order of node i in the plan. Finally, the scoring functions of the infected nodes are given by:

$$S_i = N + 1 - \ell(i), \tag{6.23}$$

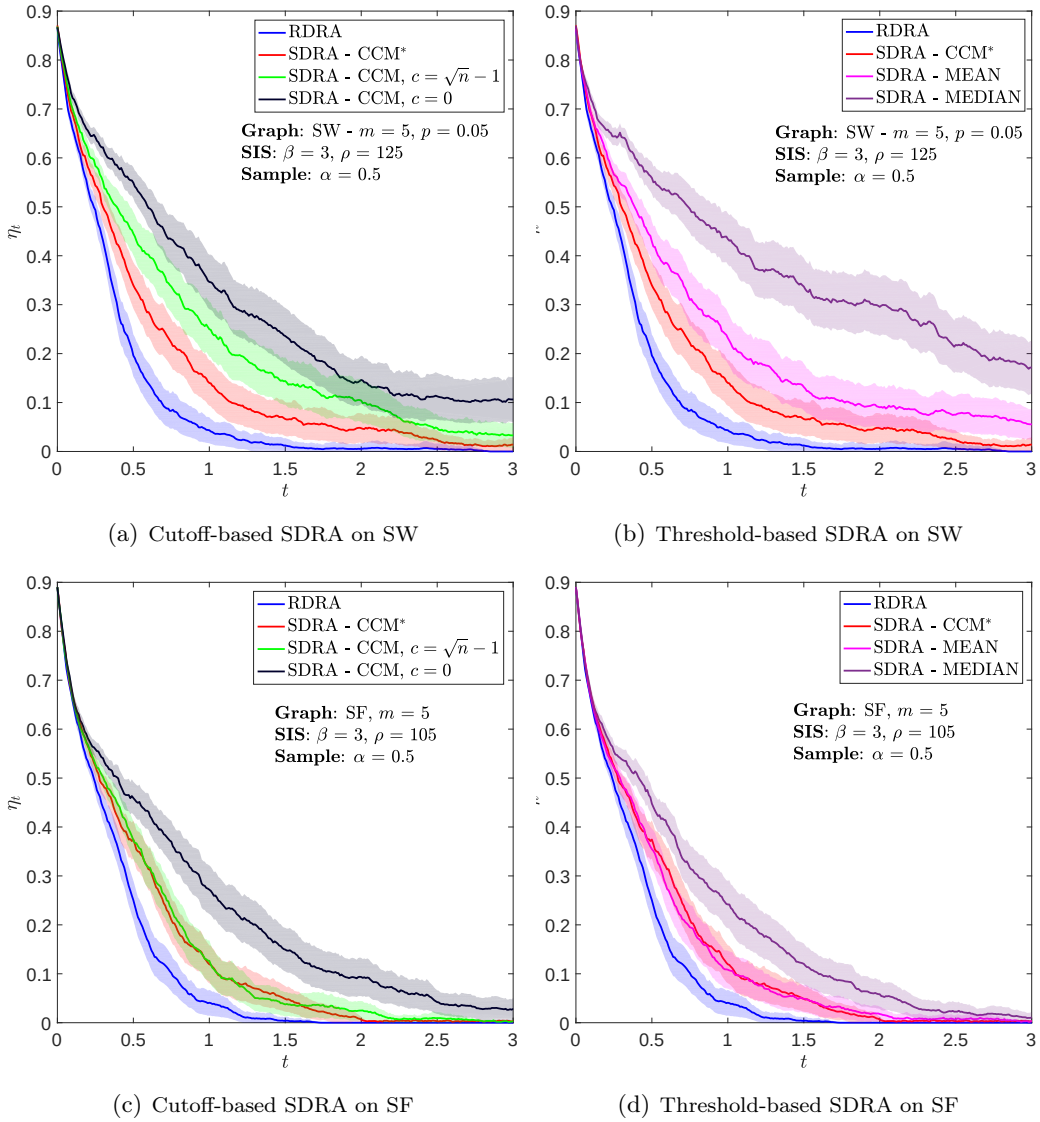
  i.e. priority is given to nodes at the beginning of the ordering. Using such a strategy, and under some specific assumptions, a bound over the expected extinction time can be retrieved, defined as $t_{\text{exct.}} = \min(t, \mathbf{X}_t = 0) \in \mathbb{R}_+$.

Example – In Fig. 6.4 is displayed the early evolution of the diffusion over the graph $\mathcal{G}$ starting with full infection, i.e. $\mathbf{X}_0 = (1,...,1)^\mathsf{T}$. The administrator manages dynamically $b = 2$ resources (blue nodes). Following the LRIE strategy, node scores have negative initial values due to the full infection, which however increase as more nodes recover. We can see that the two highest LRIE scores are spread across the network, as in each score computation only local information is taken into account.

On the other side, following the MCM strategy implies to first compute the priority-order $\ell = \{E, D, B, C, A\}$ that minimizes the max-cut, which is $CUT(\ell) = 3$ (between nodes B and C) in this case, and provides fixed node scores.

Diffusion process and score-based DRA. The diffusion process we simulate in our experiments is as described in Sec. 6.2.2 and with a fixed budget of $b = 5$ resources. In this SIS formulation we have dropped the self-recovery (i.e. $\delta = 0$ in Eq. 6.1) in order to emphasize the role of the decisions taken by the compared strategies. Concerning the scoring function, in the simulations we use both a static and a dynamic scoring function, namely MCM and LRIE.

6.6.2 Comparing online strategies



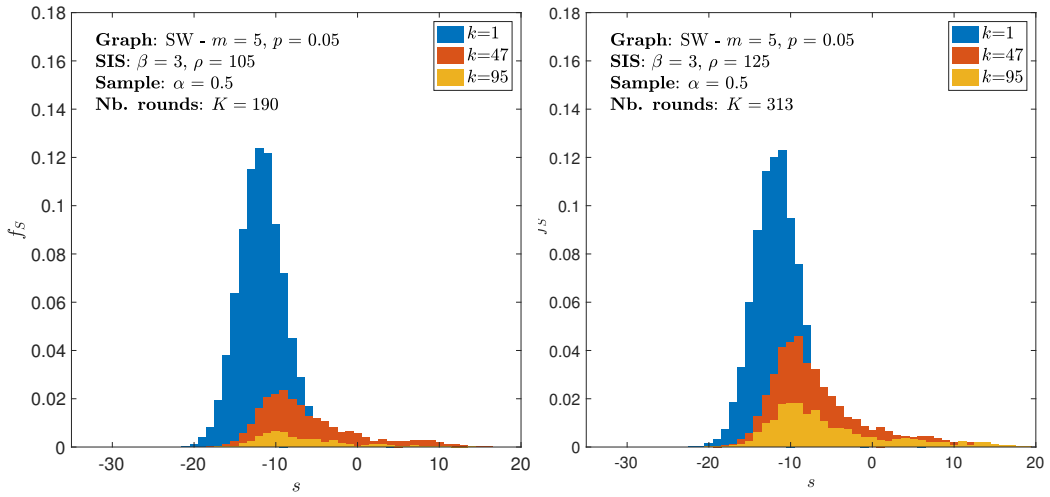
(a) Cutoff-based SDRA on SW

(b) Threshold-based SDRA on SW

(c) Cutoff-based SDRA on SF

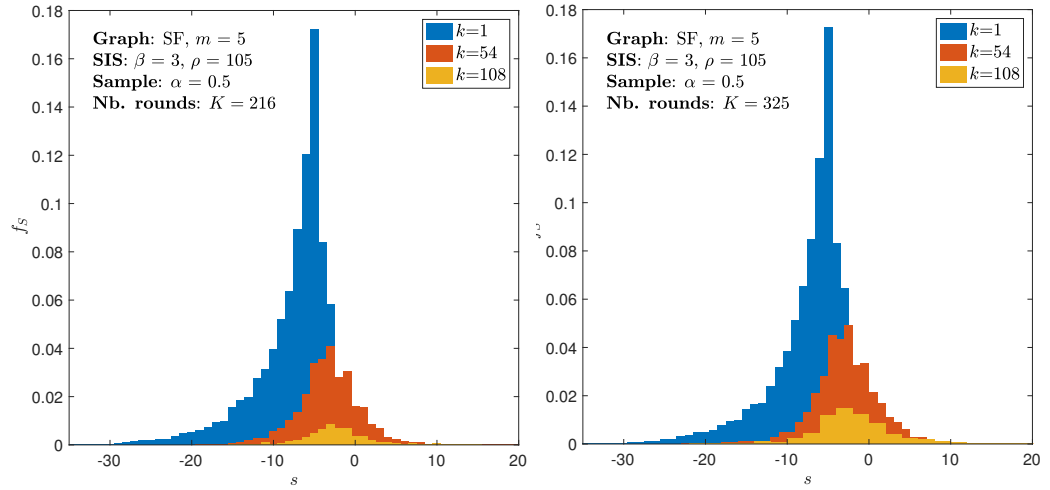
(d) Threshold-based SDRA on SF

FIGURE 6.5: Comparison of cutoff-based and threshold-based SDRA strategies, in terms of the average percentage of infected nodes η_t through time, for SW (top row) and for SF (bottom row) networks. The RDRA is shown for reference; for the same reason, the proposed SDRA-CCM is also repeated in the right subfigure of each row.



(a) RDRA on SW

(b) SDRA, CCM* on SW



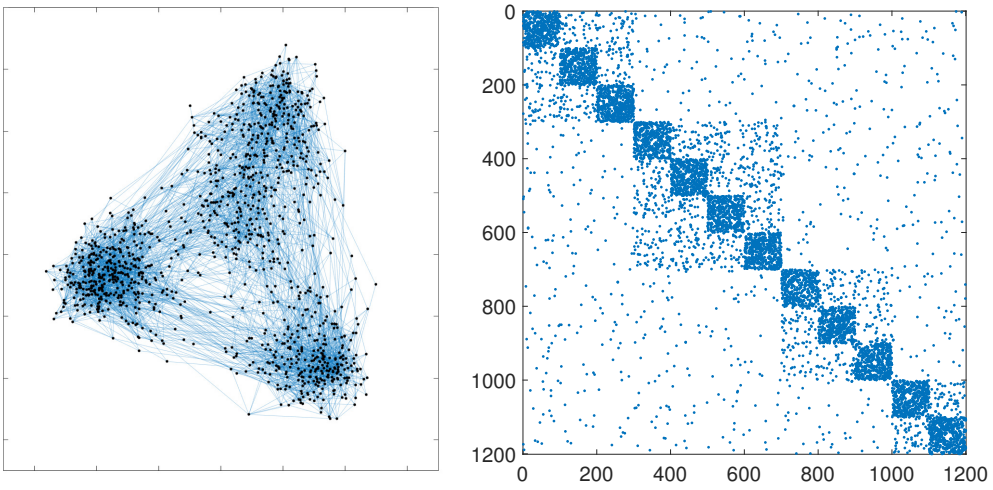
(c) RDRA on SF

(d) SDRA, CCM* on SF

FIGURE 6.6: The empirical p.d.f. f_S of the node scores, at different different rounds k through the multi-round process, for SW (top row) and for SF (bottom row) networks. The results refer to the application of RDRA or SDRA (the CCM*) strategies with the LRIE scoring function (initialization with the same infection level in each row).

In the empirical study, our aim is to compare the performance of several DRA strategies that follow Alg. 4. The offline selection strategy, which picks the reachable candidates with the highest LRIE (resp. MCM) scores, is always plotted with a dark blue (resp. light blue) curve as reference (see Figs. 6.5, 6.8, 6.9). Other colors imply the sequential allocation of resources. At each round, a fraction $\alpha \in [0,1]$ of the infected nodes $n_t = \lfloor \alpha \sum_i X_{i,t} \rfloor$ is uniformly sampled and become accessible to the administrator.

Fig. 6.5 displays the evolution of the average percentage of infected nodes with time, using the compared DRA strategies on two network types. The SW type at the top row, where on the left appears the cutoff-based CCM strategy with various cutoffs, and on the right the MEAN and MEDIAN variations of the threshold-based strategy. In both subfigures, CCM* is clearly the best performing approach. Here, MEAN is a lot better than MEDIAN, however, on an SF network (bottom row), the curves appear to be closer together and they do not differ in performance.



(a) Network (b) Block-diagonal adjacency matrix

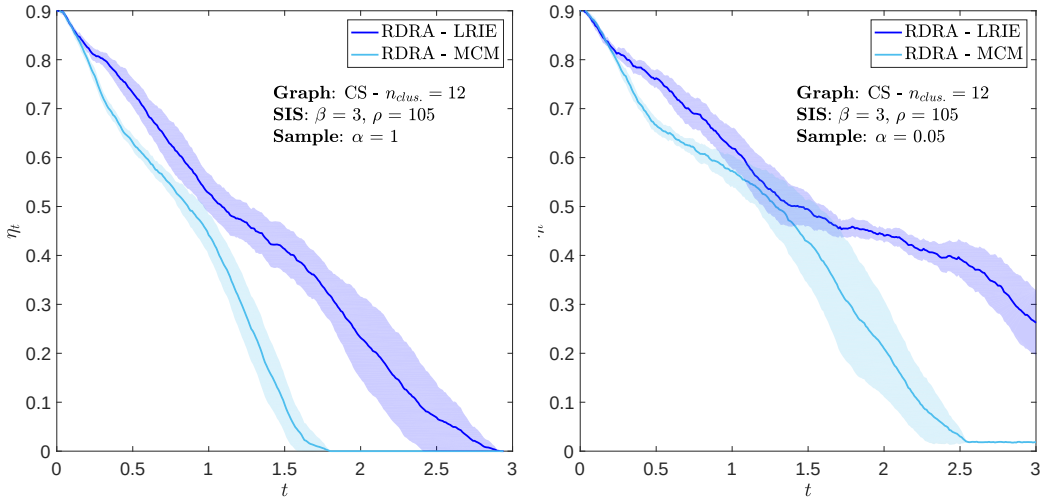
FIGURE 6.7: Example of community-structured (CS) network with $N = 1200$ nodes and 12 groups.

To further investigate the behavior of the strategies, we plot in Fig. 6.6 the score distribution f_S (here from LRIE) for the infected nodes of the network at three different rounds in the course of a multi-round process. The top and bottom rows refer respectively to SW and SF networks. Fig. 6.6(a) and Fig. 6.6(c) show the f_S obtained using a RDRA strategy (blue curves in Fig. 6.5), while for Fig. 6.6(b) and Fig. 6.6(d) the sequential strategy used is the CCM* (red curve in Fig. 6.5). Starting from nearly identical f_S per row at $k = 1$ (same initial infection level), we observe that throughout the rounds the difference between the distributions of the RDRA and SDRA strategies is larger for SW networks. This is as expected since in that example the strategies have larger difference in performance. Moreover, for SF networks, the f_S leans towards a Gaussian-like shape, which explains why MEAN and MEDIAN behave similarly, contrary to the more skewed shape obtained for a SW network.

It is easy to see in these simulations that the network structure plays a crucial role in the epidemic spread and sets the difficulty level to any strategy that tries to contain it. Also, the highly evolving shape of the score distribution throughout the process (rounds) illustrates the challenges that SDRA strategies need to address in order to be sufficiently effective.

6.6.3 RDRA with different scoring functions

CS network. So far we have used the LRIE scoring function for the prioritization of infected nodes. Here, we compare the behavior of the RDRA strategy when using either one of the dynamic LRIE or the static MCM functions. In Fig. 6.8, the percentage of infected nodes w.r.t. time is displayed for a CS network that exhibits a hierarchical community structure, see [6]. Due to the large variation of edge density that such networks exhibit, they usually require less resources than a graph with the same number of edges but without community structure; in this case $b = 17$ resources are enough for $N = 1200$ nodes. We notice that RDRA performs better with the



(a) $\alpha = 1$ (b) $\alpha = 0.05$

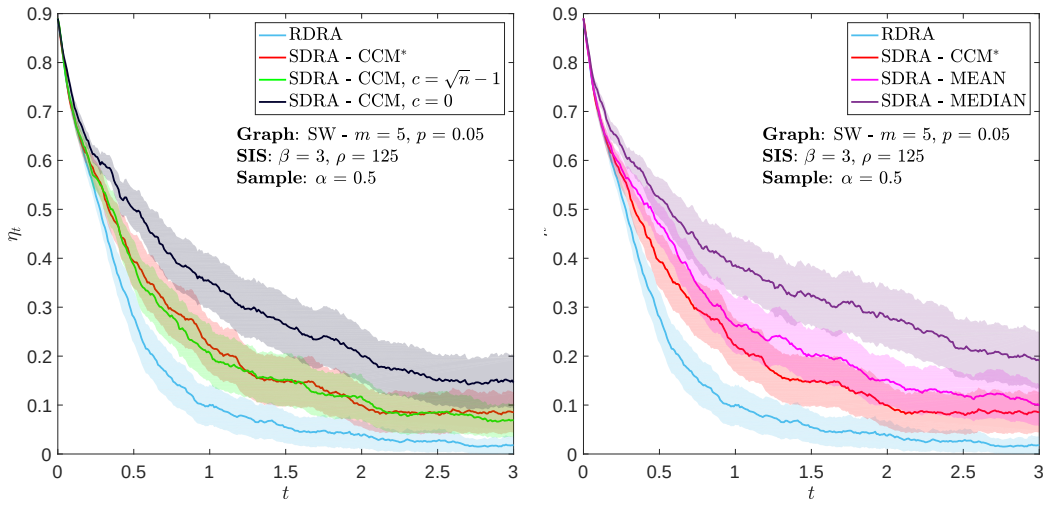
FIGURE 6.8: Comparison of the performance of the RDRA with LRIE (dark blue) and MCM (light blue). Average percentage of infected nodes w.r.t. time t, η_t for a CS network of $N = 1200$ nodes, 4 high-level groups, and 12 low-level groups, displayed in Fig. 6.7. The number of resources is $b = 17$.

MCM scoring function than with the LRIE, as it is more efficient at targeting the critical nodes. Fig. 6.8 shows the impact of the sampling size (left: $\alpha = 1$; right: $\alpha = 0.05$). Despite the significant difference between the two sampling sizes, the efficiency is only slightly reduced from one to the other.

Overall, in practice our framework seems to be able to translate the quality of a scoring function to better performance in the constrained RDRA setting, and its applicability remains high even when the sampling size is quite small. The community graph used for simulations in Fig. 6.8 is displayed in Fig. 6.7, showing a clear hierarchical structure with three levels of point density.

SW and SF networks. In Fig. 6.9, the simulations of Sec. 6.6.2 are repeated, changing only the scoring function which now is MCM (RDRA appears in both sides for reference). The two scoring functions, LRIE and MCM, give similar results on SW and on SF networks. The noticeable difference concerns the SDRA strategy for which the sequential CCM* (red curve) is almost identical to the CCM strategy with $c = \sqrt{n} - 1$ (green curve). In [13], the latter strategy is shown to be optimal for node scores drawn from a uniform distribution (but unobserved by the administrator who only makes pairwise comparisons). In order to verify this assertion, we displayed in Fig. 6.10 the score distribution f_S of the nodes w.r.t. the different rounds k. This clearly seems to be more uniform-like than the Gaussian-like shape obtained with LRIE, which explains well why the red and green curves are very close. Another observation is the fact that nodes with the highest scores are treated before scores with lowest scores (see orange bar plot), although some of them are still infected (the tail of the orange bar plot) since re-infection might occur at the beginning of the priority-order, especially in a graph without community structure. The results on a SF network are very similar to Fig. 6.5(c) and Fig. 6.5(d) and are therefore omitted due to space constraints.

Real networks. In the last simulations, in Fig. 6.11, we use a real network that



(a) Cutoff-based SDRA on SW

(b) Threshold-based SDRA on SW

FIGURE 6.9: Comparison of cutoff-based and threshold-based SDRA strategies. The Restricted DRA is shown in light blue for reference; for the same reason, the proposed SDRA-CCM is also repeated in the right figure. Average percentage of infected nodes η_t through time for SW network.

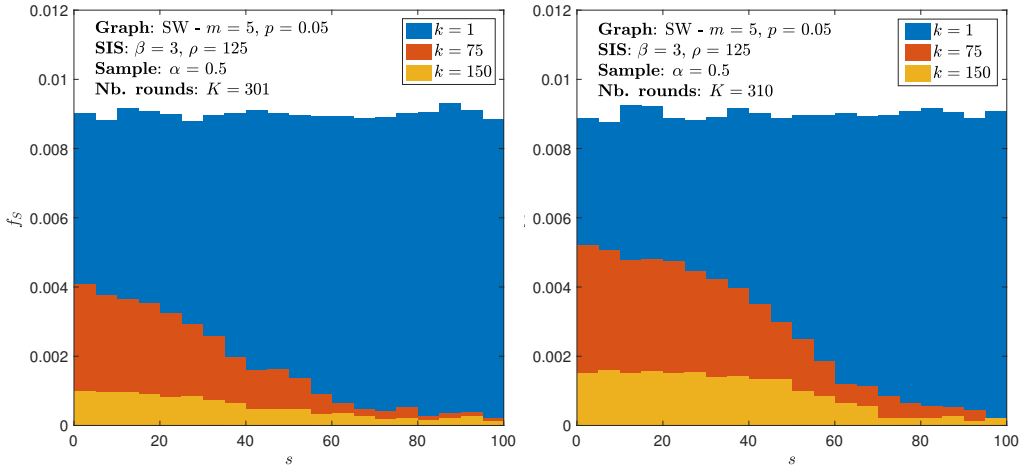
contains Facebook user-user friendships[1], which is larger than the synthetic datasets used. A node is a user and an edge indicates a friendship between two users. Note that from Sec. 4.4 in Chapter 4, using CCM with $c = \lfloor n/e \rfloor$ is a decent alternative to CCM* when few or no node recovered; it is therefore used in this simulation (red curve). We deduce, from the positions of nodes on the circle of the layout, that any node is easily reachable through a small number of jumps from another node, which is characteristic of a SW network. For the simulations, we use the MCM scoring function, slightly better than the LRIE. Despite an initial number infected nodes of 20%, with only $b = 16$ resources for $N = 2888$ nodes the epidemic exploses when the allocation strategy is not proper, e.g. when using the MEDIAN strategy (purple curve).

The simulations of Fig. 6.12 are similar to those of Fig. 6.11 but focusing on the LRIE instead of the MCM scoring function. On the left, RDRA strategies using both scoring functions are compared, while on the right a comparison of different sequential strategies using LRIE is presented. Contrary to Fig. 6.11 with MCM, the sequential strategies here fail to reduce the epidemic spread when using the LRIE scores.

6.6.4 Sampling size

As described, the sampling is performed on the infected nodes and so far we used an arbitrary fixed ratio α. To analyze the impact of the sampling size on the efficiency of the CCM* strategy, we plot in Fig. 6.13 the average percentage of infected nodes w.r.t. time for various sampling ratios. We observe that the SDRA is less sensitive to the sampling size on SF networks (right) than on SW networks (left). Furthermore, regardless the network structure, increasing the sampling size does not improve linearly the efficiency of the algorithm. In Fig. 6.14 is displayed the AUC of Eq. 6.12 for two different number of average neighbors $\bar{k} = \{2, 10\}$, and on each figure, for

[1] Available at: http://konect.uni-koblenz.de/networks/



(a) RDRA on SW

(b) SDRA, CCM* on SW

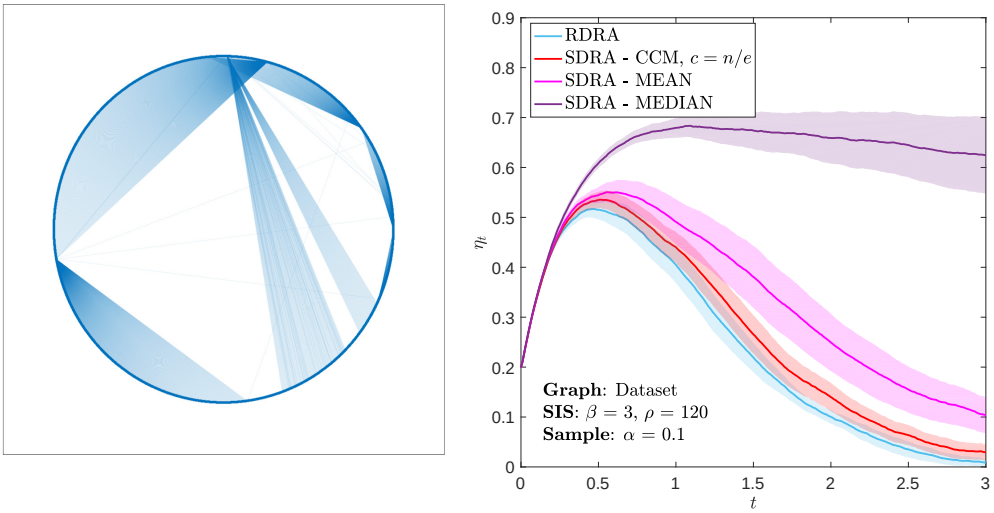
FIGURE 6.10: The empirical p.d.f. f_S of the node scores, at different different rounds k through the multi-round process, for a SW network. The results refer to the application of RDRA or SDRA (the CCM*) strategies with the MCM scoring function (initialization with the same infection level)).

three different types of networks. Clearly, the difference between the network types is more evident when the edge density of the network increases (right-hand side), and the AUC is smaller for the SW type. This can be explained by the fact that in this type of graphs, increasing the edge density merely reinforces the connectivity within the hubs, i.e. aggregate of nodes, and hence assigning resources to those critical nodes allows to reduce efficiently the spread. Note that, in this work we consider a passive sampling that is purely random and does not follow any strategy, however we might envision an adversarial sampling made by a malicious agent that adapts to the administrator's strategy. In this case, the worst case scenario plays an important role in the strategy's performance, using the CCM strategy for instance, it occurs when the b-best candidates are the first incoming, since they are very likely to be rejected by default.

6.6.5 Offline vs. online

In this section we investigate the linear regression hypothesis stipulated in Eq. 6.18 of Sec. 6.5 by simulating the epidemic spread for different sequential selection strategies. We then compare, for a fixed time horizon T, the expected percentage of infected nodes through time, $A_{\Delta N}(T) = \int_0^T \frac{\Delta N_t^I}{N} dt$, and the expected number of errors through rounds, $A_e(T) = \sum_{k=1}^K \frac{e_k}{b}$, as displayed in Fig. 6.15. The lines represent four different sampling sizes $\alpha = \{1, 0.5, 0.4, 0.2\}$ from left to right.

Example – Imagine a scenario with $b = 5$ treatments, $N = 100$ nodes, and that the administrator examines sequentially half of them, i.e. $\alpha = 0.5$ (see Fig. 6.15(a)). As expected, the result of each strategy, i.e. a 2-d point, lies on a line with slope coefficient $c_1 = 0.714$ and intercept $c_2 = -52.14$; therefore we get $A_{\Delta N}(T) \leq 0.714 \cdot M_K - 52.14$, which empirically verifies Assumption 4 (see Tab. 6.1 in the Appendix for more examples). After only 5-6 rounds, the CCM* algorithm makes no more than 1.5 errors on average for a fixed number of candidates $n_{\max} = \alpha N = 50$ and $b = 5$ resources, which gives $M_K \leq \frac{1.5}{5} K = 0.3 \cdot 301 = 90.3$; hence we get $A_{\Delta N}(T) \leq 0.714 \cdot 90.3 - 52.14 = 12.33$ that should be compared with the maximum value over



(a) User-user friendships dataset (b) RDRA and SDRA performance

FIGURE 6.11: The average percentage of infected nodes w.r.t. time t, η_t (right), on a real data network of $N = 2888$ nodes from Facebook user-user friendships (left) using the MCM scoring function. The light blue curve displays the associated non-sequential RDRA strategy. The number of resources is $b = 16$.

301 rounds, that is $A_{\Delta N}^{\text{MAX}}(T) = 301$. Therefore, the error ratio is less than $\frac{A_{\Delta N}(T)}{A_{\Delta N}^{\text{MAX}}(T)} = 4.1\%$ of the worse case scenario.

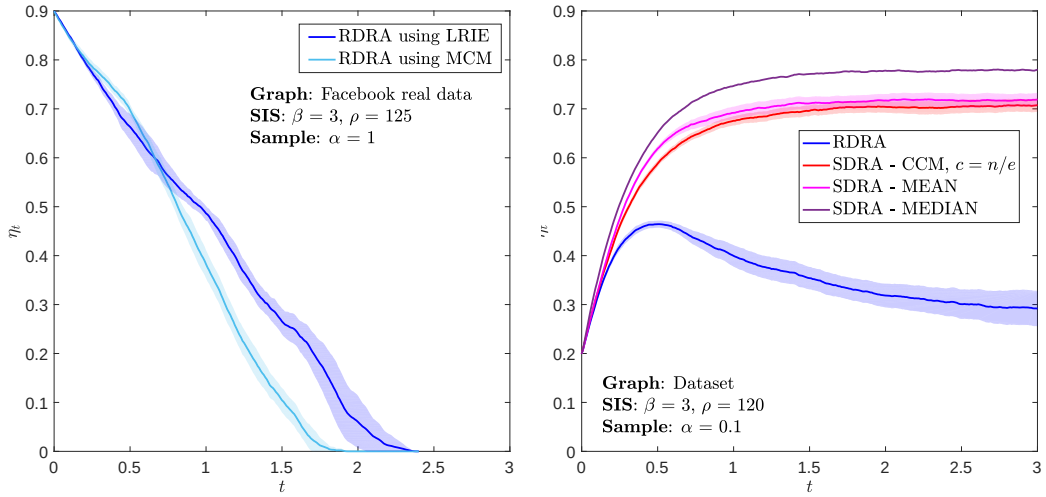
6.6.6 Allocation frequency

In Fig. 6.16 is displayed the average percentage of infected nodes for four different allocation frequencies, the plain blue curve being the usual reference curve for which the allocation matches exactly the events (infection or recovery) frequency, i.e. $\delta t = \delta t^*$, see Eq. 6.14. The parameters are the following: the scoring function used is the LRIE, the sampling size is $\alpha = 1$ and, the budget size is $b = 5$ resources and the number of nodes is $N = 100$. Note that, to simulate a continuous time process, we use Eq. 6.2 for generating the time intervals between two events, and thereby δt^* depends on the parameters of the problem. As an indication, using the parameters mentioned above and the diffusion process parameters shown in the figure, the average δt^* is almost constant through time and equal to $\langle \delta t^* \rangle = 1,4.10^{-3}$.

For the dotted, dashed and star lines, the allocation interval time is respectively set to $\delta t = \{3.10^{-3}, 4.10^{-3}, 5.10^{-3}\}$ in Eq. 6.13. More intuitively, the number of events that are 'skipped', i.e. the average number of events in between two allocation times (or rounds), from using those frequencies are shown in Fig. 6.16(b). The first remark

	$b/N = 0.05$		$b/N = 0.06$	
	c_1	c_2	c_1	c_2
$\alpha = 1$	0.68	-39	0.584	-18.7
$\alpha = 0.5$	0.714	-52.4	0.579	-19.2
$\alpha = 0.2$	1.03	-153	0.58	- 25.4

TABLE 6.1: Values of constants c_1 and c_2 for $\beta = 3$, $\rho = 125$ for a SW graph with $p = 0.05$ and $m = 5$.



(a) LRIE vs. MCM scoring function

(b) LRIE scoring function

FIGURE 6.12: The average percentage of infected nodes w.r.t. time t, η_t, on a real data network of $N = 2888$ nodes from Facebook user-user friendships. The number of resources is $b = 16$.

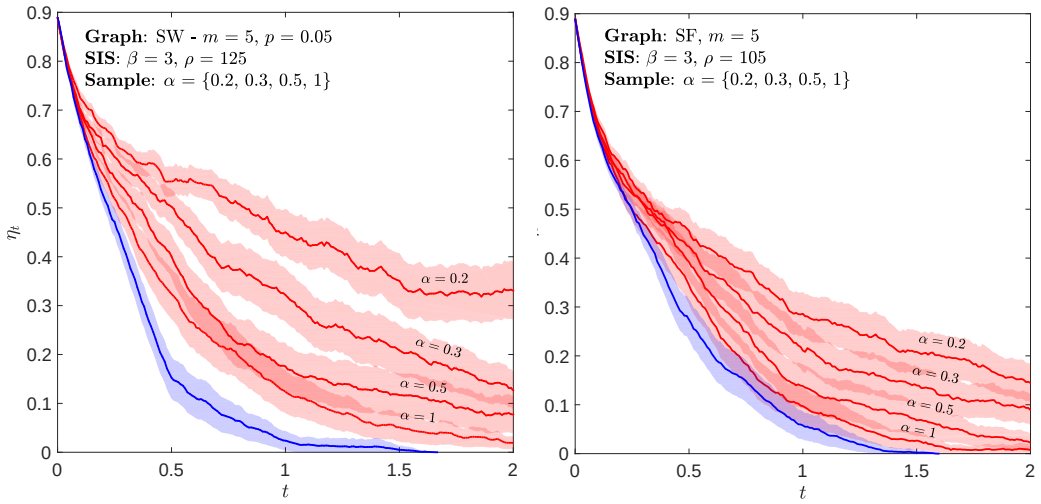
is that for those 3 frequencies the number of events skipped is on average small compared to the $N = 100$ nodes of the network. However, the impact on the percentage of infected nodes is huge, as seen in Fig. 6.16(a). This phenomenon is explained by the following reasoning; if the event that happens is a recovery, then it can be only from the resource allocation since there is no self-recovery in the simulations, and thus if this event is 'skipped', one of the very few resource is assigned to an healthy node, which is equivalent to working with $b-1$ resources.

The parameters used in Fig. 6.17 are similar to those used for the dashed line in Fig. 6.16, i.e. with, among others, $\delta t = 3.10^{-3}$. As usual, the curves that represent performances of the RDRA using LRIE (left-hand side) and using MCM (right-hand side) are respectively drawn in dark and light blue. Along with those, the SDRA performances are also shown for the best-performing online selection strategies by red (CCM), magenta (MEAN) and purple (MEDIAN) curves. Not surprisingly, the MEAN strategy works a bit better with a more meaningful and dynamic scoring function as LRIE compared to a static one as MCM.

Here it is clear that allowing a re-allocation of the resources every time there is a change in the state of the network is essential for the epidemic spread.

6.7 Conclusion and discussion

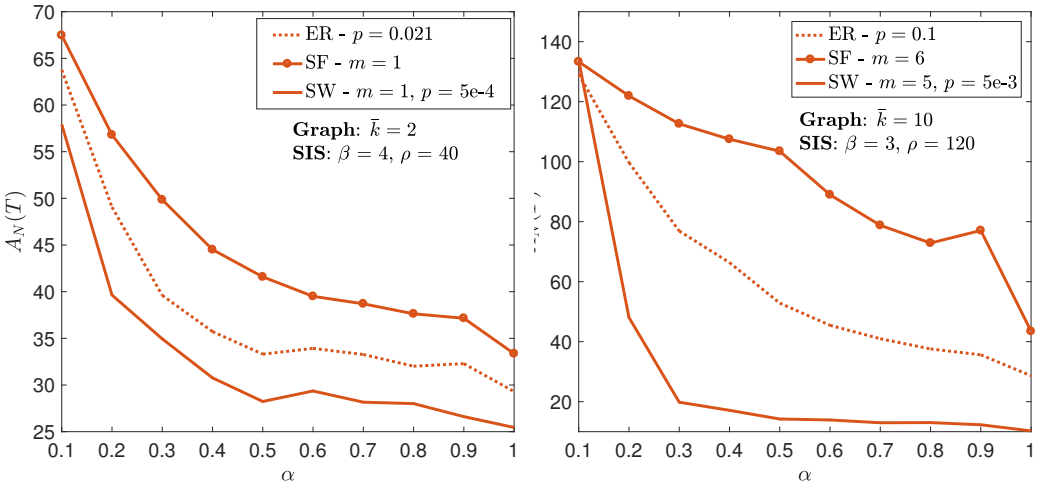
This chapter aimed towards bringing Dynamic Resource Allocation (DRA) strategies closer to meeting real-life constraints. We revisited their strong assumption that the administrator has full information and access to all network nodes, at any moment decisions takes place: anytime needed, she can instantaneously reallocate resources to any nodes indicated by a criticality scoring function. We significantly relaxed this assumption by first introducing the *Restricted DRA* model, where only a sample of nodes becomes accessible at each round of decisions. Inspired by the way decisions are taken while care-seekers arrive at a healthcare unit, we next proposed the *Sequential DRA* model that limits further the control strategy so as to have only



(a) SW network (b) SF network

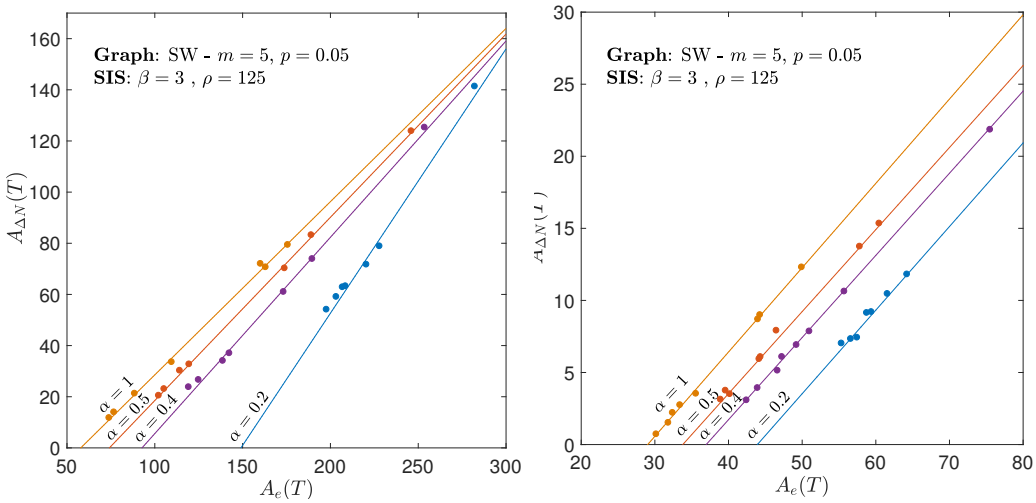
FIGURE 6.13: The average percentage of network infection though time using the sequential CCM* strategy (red lines), for various fixed sampling sizes. The blue curves display the associated non-sequential RDRA strategy with full access to nodes at each round (i.e. $\alpha = 1$).

sequential access to the sample of nodes at each round. This setting offers a completely new perspective for dynamic control: the administrator examines the nodes one-by-one and decides *immediately* and *irrevocably* whether to reallocate resources or not. This online problem is put in relation with Sequential Selection Processes (see Chapter 3) where efficient algorithms have been presented for the selection of items from a sequence for which little or no information is available in advance. Finally, according to our simulations on SIS epidemics, where we compared the performance of several variants of the above DP control models, we conclude that the cutoff-based CCM* is a very promising approach for the setting of sequential DP control.



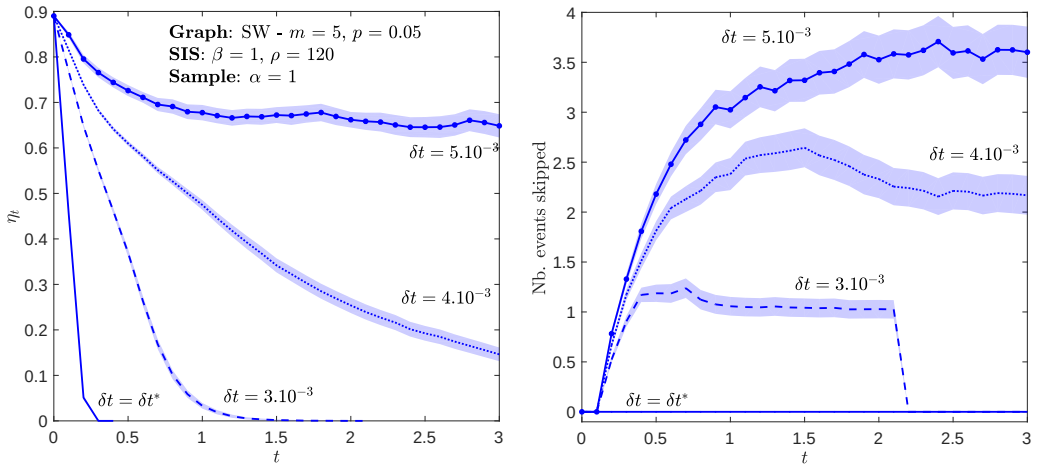
(a) $\bar{k} = 2$ (b) $\bar{k} = 10$

FIGURE 6.14: $A_N(T)$ w.r.t. α, i.e. the ratio of infected nodes that get accessible at each round, with $b = 5$ resources and $N = 100$ nodes. Dark, medium and light orange respectively stand for an Erdös-Rényi (ER), a scale-free (SF), and a small-world (SW) network, each of them having an average degree of $\bar{k}$.



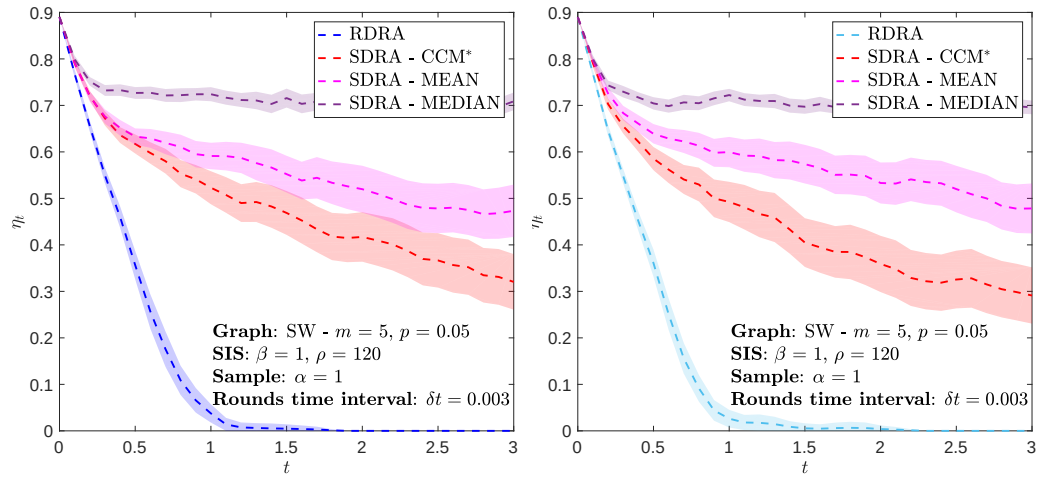
(a) $b/N = 0.05$ (b) $b/N = 0.06$

FIGURE 6.15: At fixed time horizon T, the plots show the correlation between the expected difference in percentage of infected through time using an online or the corresponding offline strategy, $A_{\Delta N}(T)$ (y-axis) and the expected number of errors through time, $A_e(T)$ (x-axis).



(a) Percentage of infected (b) Number of events skipped

FIGURE 6.16: The left and right figures represent respectively the average percentage of network infection and the average number of events in between two re-allocation of resources through time using different allocation interval time, namely $\delta t = \{3.10^{-3}, 4.10^{-3}, 5.10^{-3}\}$, and where the average time between two events is $\langle \delta t^* \rangle = 1, 4.10^{-3}$. All lines are offline RDRA strategies using the LRIE scoring function.



(a) LRIE scoring function (b) MCM scoring function

FIGURE 6.17: The left and right figures represent the average percentage of network infection through time using respectively the LRIE and MCM scoring functions. The time interval for resource allocations is $\delta t = \{3.10^{-3}\}$, with an average time between two events is $\langle \delta t^* \rangle = 1, 4.10^{-3}$ (similar to the dashed lines in Fig. 6.16).

Chapter 7

The stream-based b-diversification problem

7.1 Introduction

Collecting and analyzing large amounts of data has become increasingly common and helpful for understanding complex and various systems. In particular, *data streams* [], i.e. sequential arrival of data *instances*, are by nature difficult to handle, especially when they have great length and/or frequency. The online nature of the problem imposes new challenges to the static data preprocessing and data mining methods. Most common difficulties include working with time and memory limitations, dynamically adapt learning algorithms, or dealing with non-stationary data streams that are the results of *concept drift* [93], i.e. time-dependent properties of the incoming data. Most of the static preprocessing methods assume that either the entire training set is available as a batch, or that data statistical characteristics are not evolving over time, and sometimes both. Translating directly those static techniques for data streams is therefore not clear.

In this chapter, we are interested in the online selection of a fixed and limited number, still called *budget*, of such instances, which were referred to as *candidates* in Chapters 3-5 and as *nodes* in Chapter 6. Here again, the stream is passively observed by the DM as opposed to an active instance generation. Typically, less than 5% of the instances are selected, due to storage constraints. By definition, a selection from a stream is *irrevocable* since instances cannot be called back. For instance in the online single-item auction which is a prevalent example of such problems and which is described in Sec. 2.3.1, an auctioneer has to sequentially accept or decline bids without possibility for recall.

The selection is subject to an *evaluation criterion*, i.e. a chosen function to optimize throughout the process, hoping to be as good as possible at the end of it. Usually, the utility is an additive function of the selected items, for instance the sum of the selected items' individual value, as was the case in the previous chapters. To be more specific, the value of every item was given through ranks in Chapter 4 and through scores in Chapter 5. In this chapter however, we focus on a more complex setting for which the evaluation function is non-additive, meaning that every selected item's utility depends on the previously selected items. As a direct application, we explore the widespread *b-diversification problem* (also known as b-dispersion problem [24, 3]), formally introduced in a stream-based fashion in [109], and where the DM wants to maximize the minimum distance between all selected items. The notion of distance is inherent to each situation, and assumed known in advance by the DM,

the more common being the euclidian distance. Although situations with limited physical or memory space are frequently encountered, as in robotics, in geology, etc., few algorithms propose an easy to implement solution to this problem.

In this chapter, we study the stream-based b-diversification problem, and for that, we translate the problem of selecting items from a data stream with a non-additive objective function, into a score-based problem for which we can apply algorithms from the literature. For the mapping to be efficient, we thus need to choose carefully the scoring function to apply to every instance so that it represents adequately its relevance in the final selection. Furthermore, we try the general idea introduced in Chapter 3 to split the entire data stream into successive warm-starting rounds, this way the algorithm can adjust to already selected instances and in fact this approach seems to outperform the direct approach.

7.2 The MaxMin-diversification problem

7.2.1 Description

As mentioned before, to complete the SSP study proposed in this thesis the emphasis is made on a problem with a non-additive objective function. A widespread example of such non-additive function is the submodular function, see definition below.

Definition 16. *Submodular function: Set function* $\phi : 2^{\Omega} \to \mathbb{R}$ *over a finite set* Ω*, such that,* $\forall B, A \subseteq \Omega$ *with* $B \subseteq A$ *and* $a \subset \Omega \backslash B$*, the following condition is satisfied:*

$$\phi(A \cup \{a\}) - \phi(A) \leq \phi(B \cup \{a\}) - \phi(B). \tag{7.1}$$

In other words, the difference in the incremental value of the function that the addition of a single element makes to an input set decreases as the size of the input set increases.

The application that particularly stroke our interest is the *b-diversification* problem, also known as *result diversification*. A DM must select out of a large pool of instances, a typically very small subset of them that are, altogether, as diverse as possible. This way, the proposed set contains a wide range of options and is particularly useful when handling users-query results [105], for instance. In [38] the notion of diversity itself, that either represents *dissimilarity*, *novelty* or *semantic coverage*, and more particularly the quantification of diversity is subject to close investigation.

In the *Max-Sum diversification* problem the objective is to maximize the sum of the distance between the selected instances, while in the *Max-Min diversification* problem the objective is to maximize the minimum distance between the selected instances. Here, we chose to explore the latter as it ensures a stronger selection output, in the sense that no two instances will be too 'close' if the selection strategy performs well. This batch diversification problem has been proven NP-hard, however a few papers propose efficient approximation algorithms [41].

Formally proposed in [109], the *online* version of the above problem requires more adaptation from the DM, as instances are presented in a sequential fashion, rendering decisions irrevocable. The DM's initial knowledge only concerns the budget, i.e. the number of instances to select, and the number of instances that compose the stream.

The formal definition of the problem is as follows: out of a stream of $N \in \mathbb{N}^*$ instances $\mathbf{X} \in \mathbb{R}^{d \times N}$ presented one by one, a DM must select $b \in \mathbb{N}^*$ of them (typically, $b \ll N$) in order to maximize the minimum distance between any two items of the selected set $\dot{\mathbf{X}}_b \subseteq \mathbf{X}$. Let us first define the minimum distance between every pair of instances that respectively belong to two sets $\mathbf{U}$ and $\mathbf{V}$ by:

$$D(\mathbf{U}, \mathbf{V}) = \min_{u \in \mathbf{U}, v \in \mathbf{V}, u \neq v} \text{Dist}(x_i, x_j), \tag{7.2}$$

where $\text{Dist}(x_i, x_j) \in \mathbb{R}_+$ is the distance between instances x_i and $x - j$. When both instances belong to the same set, i.e. when $\mathbf{U} = \mathbf{V} = \dot{\mathbf{X}}_b$, the minimum distance between every instance pair is written as:

$$D(\dot{\mathbf{X}}_b) = \min_{x_i, x_j \in \dot{\mathbf{X}}_b, i \neq j} \text{Dist}(x_i, x_j) \tag{7.3}$$

Finally, the b-size best selection set is formally defined as:

$$\dot{\mathbf{X}}_b^* = \underset{\dot{\mathbf{X}}_b \subseteq \mathbf{X}, |\dot{\mathbf{X}}_b| = b}{\text{argmax}} \; D(\dot{\mathbf{X}}_b), \tag{7.4}$$

and the maximum distance associated to this set becomes $D^* = D(\dot{\mathbf{X}}_b^*)$.

The notion of distance is given according to the nature of the instances and the objective function, typically an Euclidian distance is considered.

7.2.2 Mapping to a score-based Sequential Selection Problem

The first step that we take to investigate online selection from a stream is to translate this type of process into a Generalized SSP (GSSP) (see Definition 1 in Chapter 3), which is easier to handle by applying the algorithms used in the literature. The principle of the mapping is to associate each instance to a *score* that translates its worth for the DM in a way that represents how valuable the instance is for the selection at a given moment

More formally, the scoring function $s : \mathbb{R}^d \to \mathbb{R}_+$ maps an instance to a real-valued score. By that, we allow the DM to use any score-based or rank-based sequential selection method. However, in the large majority of cases, the score distribution is unknown a priori, and is likely to change depending on the choice of scoring function. In this particular case, there is no 'correct way' to compute each instance's score, since, as stated before, the b-diversification problem is NP-hard, even in an offline fashion; i.e. there is no single score that reflects the inherent value of an instance.

The goal here is therefore to propose a step-wise scoring function that is appropriate to every instance. The score of the instance arriving at step j, $S_{j,k} \in \mathbb{R}_+$, should depend on the $k \leq b$ instances accepted so far, written $\dot{\mathbf{X}}_k = \{\dot{x}_1, ..., \dot{x}_k\} \subseteq \mathbf{X}$, which have been selected at positions $\{m_1, ..., m_k\}$ of the sequence (hence, $\dot{x}_l = x_{m_l}, \forall l \leq k$). We propose the following score for the j-th arriving instance:

$$S_{j,k} = D\left(\dot{\mathbf{X}}_k, \{x_j\}\right) \in \mathbb{R}_+, \forall j \leq N, \forall k \leq b, \tag{7.5}$$

which computes the minimum distance between this instance and every other previously selected instances.

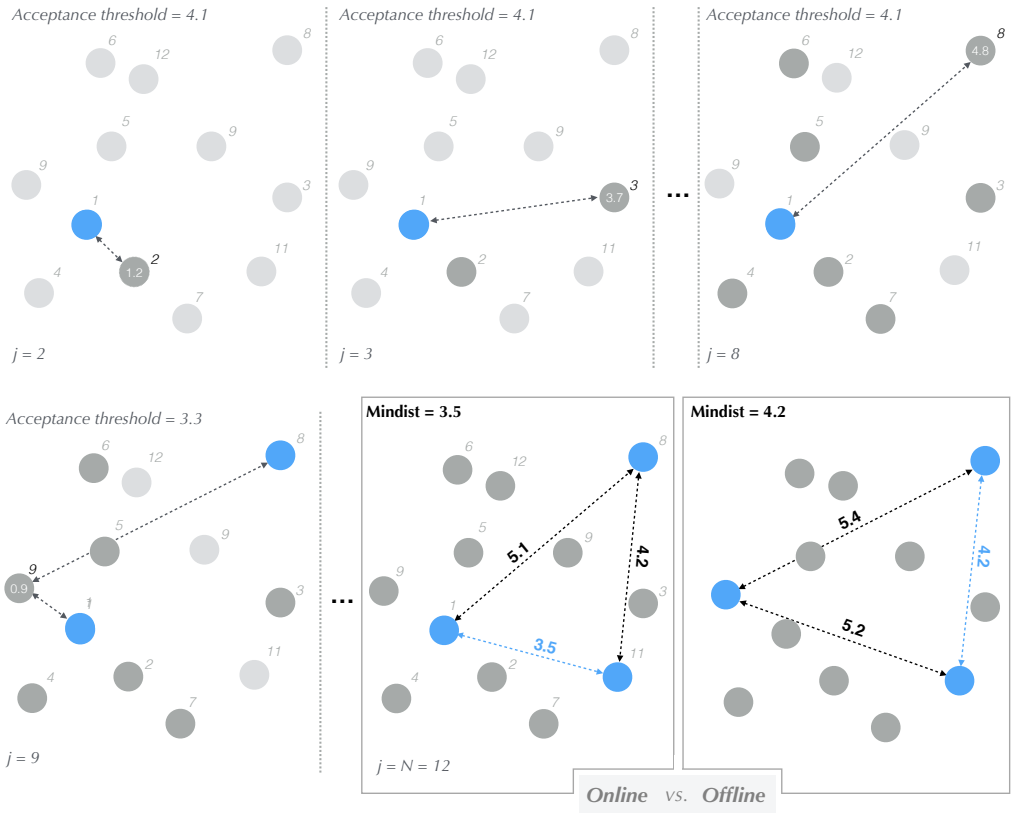


FIGURE 7.1: Example scheme of some steps of the irrevocable-choice b-diversification problem. Here $b=3$ instances must be selected out of $N=12$. The first instance x_1 ($j=1$) is accepted by default, then possible selections are decided immediately according to the instance score (inside the grey circle), i.e. their minimum distance w.r.t. to the already accepted instance(s).

Note that, using the above definition, the score of instances is defined using the current set of the k selected instances $\dot{\mathbf{X}}_k$, and hence, the scores of those selected instances are being defined separately. Otherwise, the l-th selected instance would have a score of $S_{m_l,k} = D(\dot{\mathbf{X}}_k, \{x_{m_l}\}) = 0$ since $x_{m_l} \subseteq \dot{\mathbf{X}}_k, \forall l \leq k$. Since it is meaningless to assign a score to an instance when none has been selected yet, we assume that, regardless the strategy used, the first incoming instance is always selected, i.e. $m_1 = 1$. This assumption is also considered in [109] (see Sec. 4.1.1) and the conducted empirical study therein concludes that, as long as b is reasonably large, the first instance selected has a marginal effect on the overall quality of the selection when using their algorithm.

Then, in order to avoid null scores when $k > 1$, we propose the following to define at any time, the score of the l-th already selected instance:

$$\dot{S}_{m_l,k} = D\left(\dot{\mathbf{X}}_k \backslash x_{m_l}, \{x_{m_l}\}\right) \in \mathbb{R}_+, \forall l \leq k, \forall k \leq b. \tag{7.6}$$

The sequence of decisions $\mathbf{A} = (A_j)_{j \geq 1}$ regarding the instances is such that $A_j = 1$ if the j-th instance is accepted and 0 otherwise, in particular $A_{m_1} = ... = A_{m_k} = 1$. An algorithm's performance is given through the following criterion (the *reward*)

evaluated at the end of the process and defined as $\mathbb{E}[\phi(\mathbf{S},\mathbf{A})]$, where:

$$\phi(\mathbf{S},\mathbf{A}) = \min_{1\leq j\leq N}(S_{j,b}\, A_j) = \min_{1\leq l\leq b} \dot{S}_{m_l,b} := D(\dot{\mathbf{X}}_b) \in \mathbb{R}_+, \tag{7.7}$$

with $\mathbf{S} = (S_{j,b})_{j\leq N}$ the vector of instances scores at the end of the selection. The DM's goal is therefore to maximize the $\mathbb{E}[\phi(\mathbf{S},\mathbf{A})]$.

Example. The scheme of an example is shown in Fig. 7.1, where the DM has to select $b=3$ instances out of $N=12$ sequentially incoming instances. Say that the algorithm chosen for this example starts by selecting the first instance (hence $m_1=1$). The score of the j-th arriving instance is the distance between that and the initially selected instance. The acceptance threshold γ_j represents the minimum score that the j-th instance should have in order to be selected. The algorithm selects the $j=8$-th incoming instance since its distance to the first selected instance is $\text{Dist}(x_1,x_8)=4.8>\gamma_8=4.1$. This selection leads to a change in the acceptance threshold for the next step ($j=9$), where $\gamma_9=3.3$ as the DM decides to relax the acceptance threshold. Finally, the last instance selected is the $j=11$-th one. At the end, the selected instances are $m_1=1$, $m_2=8$ and $m_3=11$, which gives a final reward of $\phi=\min(3.5,3.5,4.2)=3.5$. This can be put in comparison with the reward of an offline strategy (it has access to the batch of all instances from the beginning) that is worth $\phi_{\text{off}}=4.2$ in this example.

7.3 Simulations

7.3.1 Including a warm-start

Let us now see the precise implementation of two of the algorithms described in Chapter 2 that need careful adjustments to include the possibility of having a warm-start.

▪ **Hiring-above-the-mean (MEAN)**. Recall that the MEAN strategy needs the score of at least one selected instance in order to operate, which is not obvious at the very start of the stream, in fact $S_{m_1,0}$ from Eq. 7.6 is not defined. To implement the algorithm in this situation we then have two options, either i) the second instance is selected as well by default, i.e. $m_2=2$, and then the score of both two first instances becomes their distance, i.e. $S_{m_1,2}=S_{m_2,2}=\text{Dist}(x_1,x_2)$; or ii) the second instance is rejected, i.e. $m_2\neq 2$, but we keep as score of the selected instance the distance between the two first instances, i.e. $S_{m_1,1}=\text{Dist}(x_1,x_2)$, until another instance is selected and then the score of the first instance is updated and becomes the distance between that and the latter, i.e. $S_{m_1,2}=\text{Dist}(x_1,x_{m_2})$. This second option is the one implemented in the simulations. From $k>2$ and on, an instance is accepted if its score beats the average score of the selected instances, therefore the acceptance threshold to beat for an instance when $k\leq b$ of them have been selected so far is given by $\gamma=\frac{1}{k}\sum_{l=1}^{k} S_{m_l,k}$.

▪ **Cutoff-based Cost Minimization (CCM)**. Algorithms that consider a preselection might also work in a *warm-starting* setting, i.e. a setting where the decisions are made according to the instances that have been selected so far (see Chapter 3). Hence, for those algorithms, as the CCM, we use the following method, discussed in Sec. 3.4:

– split the GSSP of $N \in \mathbb{N}$ instances into $K \in \mathbb{N}$ subsequences, i.e. *rounds* of size $n = \lfloor N/K \rfloor$ (without loss of generality, we assume that k divides perfectly N), see Fig. 3.2.

– Each round is initialized with information from the previous one, and thus constitues a Warm-starting SSP (WSSP), see Chapter 3. For convenience, we take $K = b$, i.e. during each round exactly one instance is selected, therefore the preselection set at round k, i.e. the set of selected items so far in the previous rounds, is given by $\dot{\mathbf{X}}_k = \{\dot{x}_1, ..., \dot{x}_k\} \subseteq \mathbf{X}$. The final selection is $\dot{\mathbf{X}} = \{\dot{x}_1, ..., \dot{x}_b\} \in \mathbb{R}^{d \times b}$.

In addition, this strategy takes as input the *quality* of the selection set when k instances have been selected so far. This is a relative measure of how good one of the selected instances is compared to what remains to be seen from the sequence. It is denoted $q_k \in]0,1[$, where $q_k \to 1$ represents a good selection set. This notion needs to be redefined in this context. Intuitively, in this diversification problem we can tell that a single selected instance cannot have a 'quality' by itself, since it essentially needs to be compared to other selected instances. Henceforth, the first selected instance should not be considered 'good' or 'bad', and the quality is rather set to the moderate $q_1 = 1/2$. Originally the quality was defined as a function of the rank of the selected instances compared to the following instances. Making the analogy with Definition 4, we get:

Definition 17. *Rank-based relative quality of the preselection set at the beginning of round k (q_k): when $k-1$ instances have been selected so far, the average rank of the selected set compared to the n instances:*

$$q_k := 1 - \frac{\frac{1}{k-1}\sum_{l=1}^{k-1} R(\dot{S}_{m_l,k-1}) - 1}{n}, \tag{7.8}$$

where $R(\dot{S}_{m_l,k-1}) := R_n\left(\dot{S}_{m_l,k-1}, \mathbf{\Sigma} = \{S_{n(k-2)+1,k-1}, ..., S_{n(k-1),k-1}\}\right)$ is the rank of the l-th selected instance, computed during the k-th round, compared to the instances scores that arrive at that round. The initial quality is set to $q_1 = 1/2$.

The simulations from the following section are implemented with the low-failure version of the CCM (lf-CCM) for which the acceptance threshold is given in Eq. 4.18 with $b = 1$, $\hat{\mu}_j$ and $w_j := \sqrt{\mathbb{V}[\tilde{A}_j \mid \tilde{A}_n = 1]}$ are determined in Proposition 6 of Chapter 4, and using $c = \lfloor \sqrt{n} \rfloor$ [13].

At every round of the lf-CCM algorithm, the entire learning phase requires $\mathcal{O}(c\log(c)d)$ time, i.e. $\mathcal{O}(\sqrt{n}\log(\sqrt{n})d)$ operations, as it only sorts the $c = \lfloor \sqrt{n} \rfloor$ first instances. It takes $\mathcal{O}((n-c)d)$, i.e. $\mathcal{O}((n-\sqrt{n})d)$ time to go through the selection phase, i.e. to compare each d-dimensional instance to a fixed scalar score, without any adaptation from the DM. Replacing a fixed acceptance threshold with a dynamic one requires to compare every new instance with the sorted vector of already seen scores, hence it needs $\mathcal{O}((n-c)d\log(n))$, i.e. $\mathcal{O}((n-\sqrt{n})d\log(n))$ operations. The entire time complexity of the lf-CCM algorithm is therefore $\mathcal{O}((N - \sqrt{Nb}/2)d\log(N/b))$, or more simply $\mathcal{O}(Nd\log(N))$. Note that this is significantly lower than the $\mathcal{O}(N^2(d+1) + Nd(b+1))$ complexity of the DYN-SIMPLEK algorithm built for the b-diversification problem [109].

■ **Dynamic simplek (DYN-SIMPLEK)**. The algorithm proposed in [109] which was built exclusively for the stream-based b-diversification problem. It relies on a

heavy offline computation to derive an approximation of the acceptance threshold be set by an offline strategy and hence, it does not necessarily constitutes an appropriate 'online' strategy for other practical cases where the DM should take quick decisions.

The so-called *simplek* algorithm [109] operates as follows: the DM systematically rejects by default all first $c = \lfloor N/2 \rfloor$ instances and from them computes an acceptance threshold γ for the instances that will follow. This acceptance threshold is found by binary search, it is set to the average between a lower bound –which is at first the minimum distance among the batch of rejected instances– and an upper bound –which is initially the maximum distance among the batch of rejected instances. If the number of instances which are more than (or exactly) γ away is larger (respectively smaller) than b, the acceptance threshold becomes the upper (respectively lower) bound, until exactly b instances are γ away. According to the authors, this step takes $\mathcal{O}(cd)$ to store the c instances, then $\mathcal{O}(N^2 d)$ time to compute the threshold. Here we denote this threshold by $\gamma_{\text{off}} \in \mathbb{R}_+$ since it requires a rather costly offline computation to have been made. In the selection phase that follows, the first arriving instance is always selected, i.e. $m_1 = \lfloor N/2 \rfloor$. Next, every instance whose distance with the latter is greater or equal to the acceptance threshold γ_{off} gets selected until b instances are selected.

In order to avoid a high failure rate, which is approximately $\rho_f = 35\%$ using this static acceptance threshold (see Sec 4.2.1 in the original paper), the authors propose the *dynamic simplek* (DYN-SIMPLEK). In this adapted version, the probability distribution of the distance between future and selected instances, noted f, gets approximated by using past rejected instances as estimates for future instances. Note, though, that this action is quite computationally heavy and considerably slows down the online decision process. In the selection phase, the time complexity is $\mathcal{O}(Nd)$ for every selection, i.e. $\mathcal{O}(Ndb)$ in total, and $\mathcal{O}(N)$ for every other rejected instance, i.e. $\mathcal{O}((N - N/2 - b)N)$ overall. Finally, the entire algorithm time complexity is $\mathcal{O}(N^2(d+1/2) + (d/2 + bd - b)N)$.

The dynamic threshold γ_j at the current step j is set such that the area below the pdf curve between that and to the right is $\frac{b-k}{N-j}$, where k is the number of instances selected so far; i.e. $\int_{\gamma_j}^{+\infty} f(x)dx = \frac{b-k}{N-j}$. Finally, the new acceptance threshold is set to the minimum between that and γ_{off} at each step j. This manipulation allows to significantly reduce the failure rate, as shown in the next simulations.

7.3.2 Simulations results

Generated data

Inspired by the simulations of [109], we start with data from $N = 5000$ z-normalized random-walks. The walks are reshuffled to produce 100 streams of N walks of the original dataset ordered randomly.

In Fig. 7.2 is displayed the minimum distance between the set of $b = 10$ selected instances using the different strategies described in Sec. 2.3.2, averaged over the 100 tests. The red line represents the median of each strategy, the blue box the quantiles, the whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually with red dots.

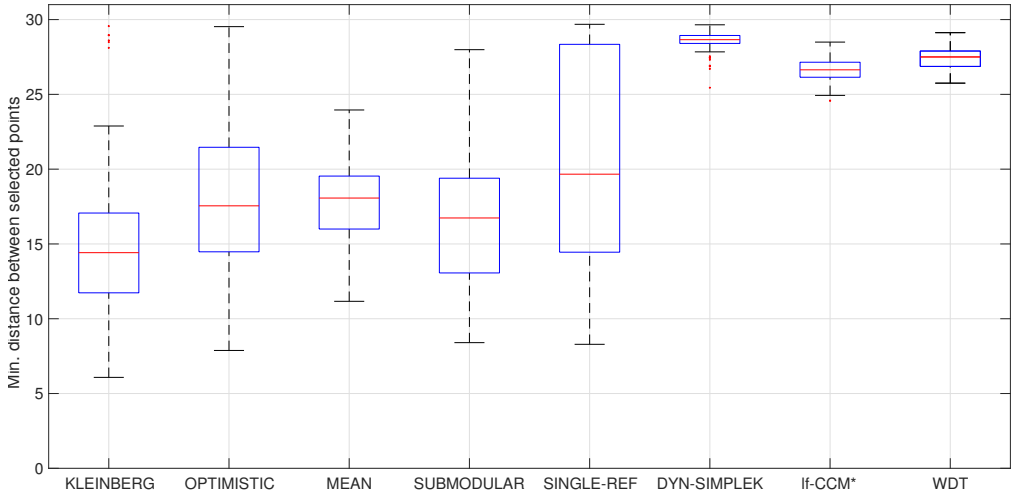


FIGURE 7.2: Minimum distance between the $b = 10$ selected instances on generated random walks. The red line gives each strategy's median, blue boxes are the quantiles and the outliers are represented with red dots. The lf-CCM* and the WDT algorithms operate in b rounds.

Strategy	**Failure rates (%)**			
	Random-walks	Insects	Fishes	Rocks
KLEINBERG	94.0	92.4	73.8	88.2
OPTIMISTIC	87.0	83.4	59.8	62.6
MEAN	0.0	0.0	1.0	4.2
SUBMODULAR	100	96.2	77.8	87.6
SINGLE-REF	63.0	56.2	48.4	53.2
Dynamic simplek	0.0	1.4	1.8	6.4
lf-CCM	0.0	4.8	5.7	74.2
WDT (b rounds)	0.0	0.6	1.4	27.4
WDT (1 round)	100	100	61.0	97.0

TABLE 7.1: Table of failure rates ρ_f in % for different strategies (rows) and different datasets (columns).

A striking result from the random walks simulations concerns the failure rates associated to these simulations and stored in Tab. 7.1 (see the Random-walks column). As the ratio between the budget of instances to select and the length of the sequence is rather small ($b/N = 2.10^{-3}$), the chances that the algorithm fail to fill all $b = 10$ empty slots should be, intuitively, quite low. The failure rates for the KLEINBERG, OPTIMISTIC, and SUBMODULAR strategies are however surprisingly high (all very close to 100%), and lead to a poor quality of the resulting selected sets. More specifically, they achieve a median of approximately $D \approx 16$, when the median of a strategy with a slightly better failure rate of 63% (see SINGLE-REF in Tab. 7.1) already climbs to $D \approx 20$ (see SINGLE-REF in Fig. 7.2).

A second observation concerns the MEAN strategy, which always has the lowest failure rate (see MEAN in Tab. 7.1), yet does not manage to output a high-quality selection set (the median is at $D \approx 17$). This is due to the algorithm procedure; the x_1 is selected, x_2 is rejected and the score of x_1 becomes its distance to x_2, then subsequent instances are selected if their score is higher than the score of x_1, hence the acceptance threshold is initially rather low.

Finally, the difference between a strategy that operates in a single round and one that operates in b rounds by separating the entire sequence into b subsequences in which a single instance is selected each time. A straightforward example is the WDT algorithm that we purposely applied in both situations (see the two last rows in Tab. 7.1). Note that the WDT algorithm is applied in the no-information setting, i.e. it is the rank-based version of the algorithm (see Sec. 5.3.2 in Chapter 5) and is therefore very fast to generate and to apply since it is not based on score values, thus no p.d.f. has to be computed during the process. The failure rate of the WDT applied in b rounds is significantly lower than that of a single round, which confirms our intuition that separating a long sequence into b warm-starting subsequences helps improving the final selection.

Real dataset

Fig. 7.3(a) displays the minimum distance between the set of $b = 8$ selected instances of dimension $d = 256$ out of a stream of $N = 1200$ insects, using the same strategies as in Fig. 7.2, averaged over 500 tests, this time using real-data provided again in [109].

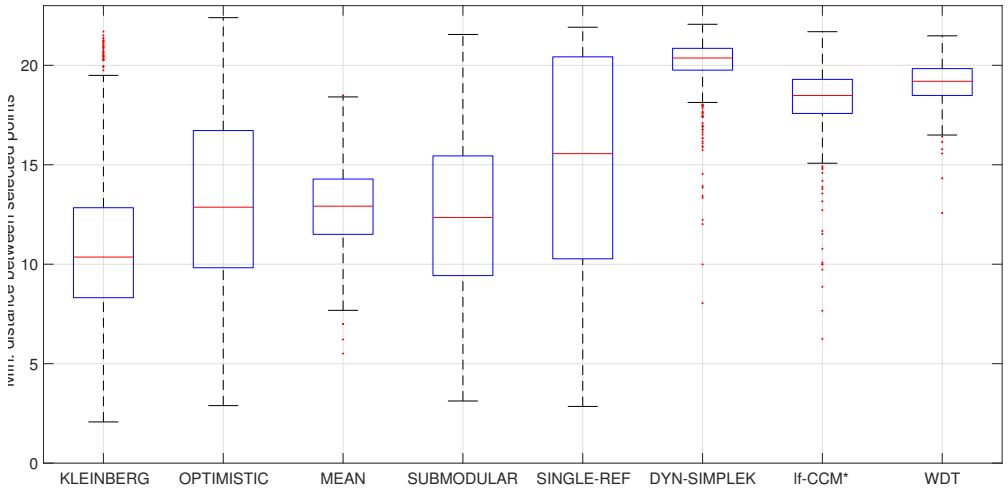
Fig. 7.3(b) and Fig. 7.3(c) uses real-data but from fishes and rocks respectively. In the provided dataset, each row corresponds to an insect/fish/rock encountered on the path, while each column gives the numerical value of every of the d features.

Unsurprisingly, the results on real-data comply with those on generated random walks, i.e. the KLEINBERG, OPTIMISTIC, and SUBMODULAR strategies have very high failure rates, and therefore perform poorly; while the lf-CCM algorithm remains a suited alternative to the DYN-SIMPLEK. Moreover, the difference in achievement gets smaller as the number of instances to select from decline. Observe also that the algorithm which seems to have a low failure rate while maintaining a good minimum distance is the b-rounds WDT algorithm (in a no-information setting) for most datasets. However, the DYN-SIMPLEK is the best performing algorithm on these sets of data, despite being rather computationally heavy which renders it less practical than other faster algorithms, e.g. the SINGLE-REF, the lf-CCM or the rank-based WDT algorithm.

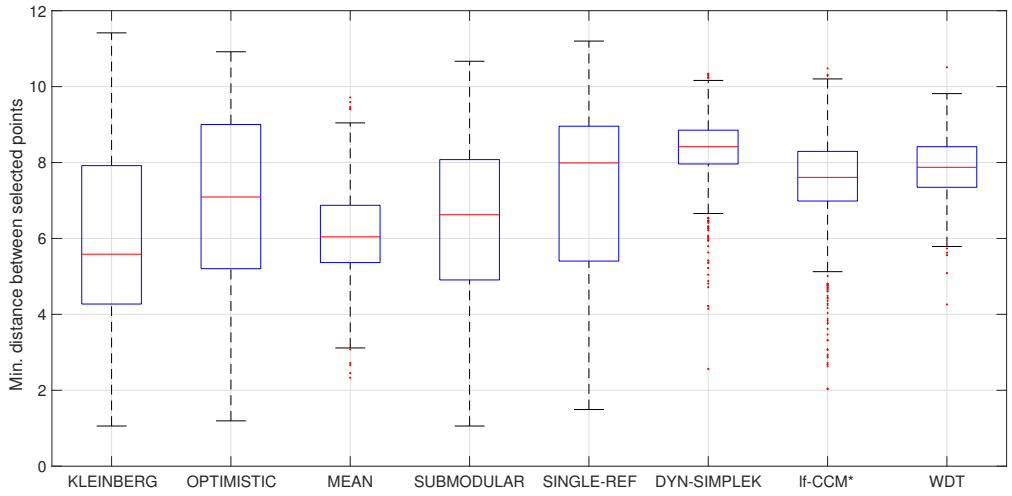
7.4 Conclusion and discussion

This chapter investigated a non-additive multi-choice Sequential Selection Problem application, namely the stream-based b-diversification problem. In this case, the DM's goal is to have, at the end, the most diverse selection set at hand. There are several ways to define the set diversity, in this chapter we set that to be measured as the minimum distance between elements of a set. Also as distance function at element-level we regarded the standard Euclidian distance. After a thorough description of this problem, we propose to translate it to a score-based SSP, for which we can apply already existing algorithms that are described in previous chapters.

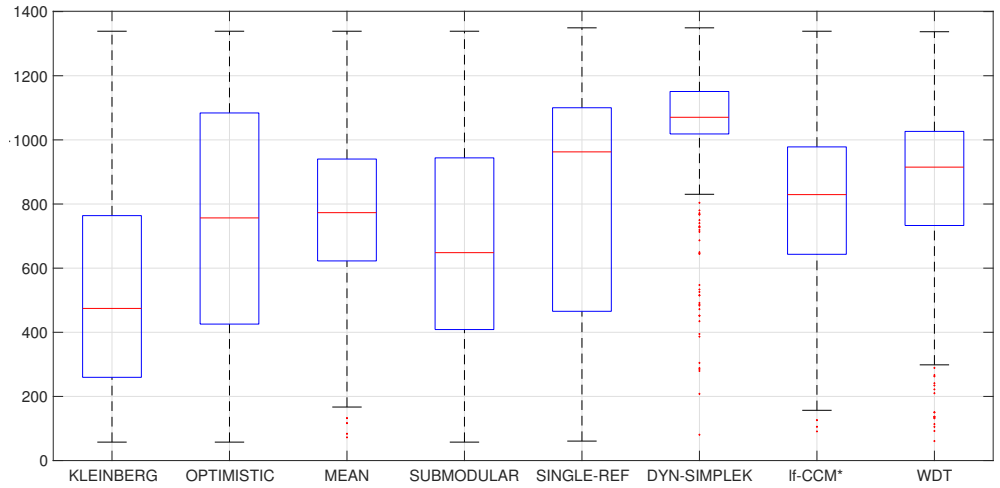
Secondly, the aforementioned algorithms' performance are compared through simulations based on real datasets and their respective limitations are highlighted. The lf-CCM and WDT algorithms clearly present the better trade-offs between time complexity and efficiency at maximizing the objective function.



(a) Data from a sequence of $N = 1200$ insects with $b = 8$ resources



(b) Data from a sequence of $N = 200$ fishes with $b = 4$ resources



(c) Data from a sequence of $N = 70$ rocks with $b = 5$ resources

FIGURE 7.3: Minimum distance between the selected instances. Red is the median and blue box the quantiles. The whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually with red dots. The lf-CCM * and the WDT algorithms operate in b rounds.

Chapter 8

Conclusion

This manuscript aimed at filling one of the existing gap between real-life control of diffusion processes and their current modeling. To fulfill this objective, we first needed to fill a lack in the field of sequential selection problems, with the introduction of the Warm-starting SSP, a novel setting that addresses the cold-start limitations of standard settings, and for which we provide an offline analysis of the initial state in Chapter 3.

Then, we extend this warm-start option to a multi-round context, where each round constitues a separate WSSP that starts with at hand, i.e. as a warm-start, the final selection of the previous round. The warm-start aspect introduced in the generalized SSP raised some algorithmic questions regarding the online decisions, particularly in a multi-round setting where an ideal algorithm would maintain an optimal selection in the course of multiple rounds, rather than at the scale of a single selection round.

The Cutoff-based Cost Minimization algorithm is next proposed as an answer to this new setting in Chapter 4. Inspired by the standard Secretary Problem algorithm, i.e. composed of a learning phase and a selection phase, CCM is fully suited to any degree of warm-start considered. The rank-based regret function that we used in the WSSP setting enables our algorithm to be efficient for arbitrary candidate scores. We approximate analytically this objective function by deriving the expectation of the main parameters of the problem (e.g. the acceptance threshold, the number of accepted candidates, the regret, etc.). In the end, we implemented CCM in a multi-round framework (MSSP). Such a process was motivated by the natural needs of real-world recruitment processes that are regularly trying to improve the personnel of an organization or a company. The conducted simulations are consistent with our analytical work and demonstrated that CCM is efficient in reducing the regret at the course of the multi-round process while being robust to scores, resignations, or changes in the number of jobs.

In the subsequent chapter, the score distribution is now assumed to be known by the DM. This assumption forces the DM to reconsider her selection strategy accordingly. A new algorithm is therefore presented in Chapter 5, called Warm-starting Dynamic Thresholding, and which applies to a score-aware WSSP setting, i.e. considering the case where the incoming candidates have scores following a known distribution. The proposed algorithm is based on a dynamic programming approach and achieves optimal threshold estimation at each step of the sequence of interviewed candidates. Experiments have been performed in the multi-round setting, which is interesting for real-world reoccurring recruitment processes. WDT demonstrated a clearly better performance than existing algorithms, regardless the number of initially empty job positions. We additionally proposed a rank-based dynamic programming alternative

that can go beyond the need of knowing perfectly the distribution that generates the scores, yet, resulting in satisfying outcomes.

Chapter 6 goes back to the motivating example, i.e. bringing Dynamic Resource Allocation strategies closer to meeting real-life constraints. We revisited their strong assumption that the DM has full information and access to all network nodes, at any moment decisions takes place: anytime needed, she can instantaneously reallocate resources to any nodes indicated by a criticality scoring function. We significantly relaxed this assumption by first introducing the Restricted DRA model, where only a sample of nodes becomes accessible at each round of decisions. Inspired by the way decisions are taken while care-seekers arrive at a healthcare unit, we next proposed the Sequential DRA model that limits further the control strategy so as to have only sequential access to the sample of nodes at each round. This setting offers a completely new perspective for dynamic control: the DM examines the nodes one-by-one and decides immediately and irrevocably whether to reallocate resources or not. This online problem is then put in relation with previous chapters and particularly with the CCM* algorithm presented in Chapter 4 that is implemented in this epidemic control case. Obviously, the Multi-round Sequential Selection Process is naturally fitting for handling this new setting. Finally, according to our simulations on SIS epidemics, where we compared the performance of several variants of the above DP control models, we conclude that the cutoff-based CCM* is a very promising approach for the setting of sequential DP control.

Finally, we investigated a non-additive multi-choice Sequential Selection Problem application, namely the stream-based b-diversification problem in which the DM's goal is to have, in the end, the most diverse selection set at hand. After a thorough description of this problem we compare the performances of a few relevant algorithms from the literature through simulations based on real-data and highlight their respective limitations.

Appendix A

Personalised COVID-19 Isolation and Exit Policies Based on Machine Learning Predictions

Authors: Theodoros Evgeniou, Mathilde Fekom, Anton Ovchinnikov, Raphael Porcher, Camille Pouchol, and Nicolas Vayatis.

Problem definition: In mid-2020, following social distancing measures due to COVID-19, governments consider “exit", “deconfinement", and the like strategies to relax the lock-downs. We study how that might be done by utilising machine learning predictions of clinical risk.

Academic / Practical Relevance: Pandemics present existential threats to humanity, much like climate change or nuclear proliferation do. Battling such threats requires innovation, and we explore the extent to which the main innovation of the last decades – machine learning and artificial intelligence – could help.

Methodology: We extend a standard susceptible-exposed-infected-removed (SEIR) model to account for personalised predictions of clinical “severity" risk, defined by the risk of an individual needing intensive care (ICU) if infected. We then simulate differential exit policies using COVID-19 data and estimates for France as of early May 2020.

Results: Simulations and sensitivity analyses indicate that an exit policy considering clinical risk predictions starting on May 11, the date chosen by the French government, could enable to immediately relax isolation restrictions for millions of the lowest-risk population, and consequently relax the restrictions on the remaining population months faster – while abiding to the ICU capacity at all times. Exit policies without risk predictions would exceed the ICU capacity by a multiple, or they should isolate a substantial portion of population for over a year to not overwhelm the medical system at one of the waves of the infection.

Managerial Implications: Governments need to develop policies and invest in infrastructure to implement personalised isolation and exit policies based on machine learning predictions at scale. This involves health data policies to train predictive models and apply them to all residents, policies to support targeted resource allocation to maintain strict isolation for high risk individuals, and the likes.

Introduction

As of mid-2020, many countries have adopted non-pharmaceutical interventions, such as isolation restrictions, or "lock-down" [69], to control the spread of COVID-19. Epidemic models have been used to inform such policies [49, 46]. Governments now consider relaxing these restrictions, but how to best do so is unclear. Some approaches rely on immunity tests [89] or on test and tracing technologies [106]. We studied a different approach that, instead of using ("ex post") immunity or diagnostic tests, utilises ("ex ante") predictive technologies, such as machine learning, which have been proven successful in other contexts. This approach can complement those relying on immunity and diagnostic tests, or be considered independently.

The type of personalised isolation, or "confinement", and exit, or "deconfinement", policy that we study is as follows. First, for each individual one predicts his or her clinical risk score (e.g., the probability of experiencing symptoms severe enough to require an ICU bed). Second, those with predicted scores above a certain threshold are classified as "severe," or "high risk," and the remainder are classified as "mild," or "low risk." Third, those classified into the high risk group are subject to stricter isolation and protection ("confined"), while those in the low-risk group to a softer one, or none at all ("released"). In practice, this can be achieved by targeted allocation of resources (e.g., providing masks and other personal protective equipment (PPE), dedicated health support, free delivery of groceries and other necessities, etc.) to the high-risk group, targeted government communication that differentiates between high and low risk individuals, and other targeted policies.

The studied policies rely on the observation that per current estimates, the vast majority of the population, >99% in France [95], do not experience severe symptoms needing an ICU, if infected by Sars-Cov-2. Hence, should one be able to (i) determine who the ∼1% of severe cases are, and (ii) perfectly and temporarily isolate and protect them, the remainder would be able to return to a more-or-less normal life. In such an ideal scenario, many low risk people would get infected and would infect others, but none infected would have severe symptoms – as those would have already been correctly identified and perfectly protected. The medical system would not be overwhelmed, no one will die, and the society and economy would avoid a major shock from the indiscriminatory lock-down that was implemented in many countries.

In practice, the effectiveness of such a personalised policy would depend on two critical imperfections: (i) risk prediction models might occasionally make mistakes, e.g., false positive and false negative errors, and (ii) isolation would also be imperfect, as for example high-risk individuals, who should be isolated, may occasionally contact those infected (e.g., due to PPE shortages or non-compliance), and the low-risk individuals, who would be able to continue normal life, may not do so (e.g., due to fear).

We studied how these two imperfections impact the effectiveness of the aforementioned personalised pandemic isolation and exit policies. In order to do so, we extended a standard epidemic model, namely a version of SEIR [77], to incorporate personalised predictions of severity risk, defined as requiring an ICU bed if infected (other definitions can be similarly used). Using simulations, we investigated how prediction models for patient severity may inform policy in two scenarios. First, when there is an ongoing outbreak as it was the case in France on the 17th of March, 2020, when lock-down, or confinement, started. Second, when the outbreak

has been curbed and progressive loosening of isolation policies (exit, or deconfinement) may take place, as was the case in France starting from the 11th of May, 2020.

We did not attempt building a risk-model of our own. Rather, given the existing research indicating differential impact of COVID-19 across patients, for example depending on age, body mass index, hypertension, diabetes and other factors [55], and the emerging risk models, such as referenced in [16] and [104], we assumed hypothetical risk prediction models. We then studied the sensitivity of the simulation results with respect to model discrimination akin to the models already developed.

To populate our simulation models, we used available COVID-19 estimates and data from France as of May 2020 [32]. At the end of lockdown on May 11, there were about 2750 ICU beds occupied by people with COVID-19, with a peak at 7 148, compared to the total roughly 10 000 capacity recently reached by the French health system. We used current estimates with a reproduction number value of $\mathcal{R}_0 = 2.9$ prior to lock-down, and 1.5 million people who had been immune or infected when it started in France on March 17, 2020 [95]. Uncertainty was analysed using Approximate Bayesian Computation.

Our simulations lead to the following main observations and the corresponding implications:

1. Isolation and exit policies, when based on risk-model predictions, could be substantially faster and safer. Utilising realistic parameter values and a high-quality risk model at the upper-end of [16], simulations indicated that the complete COVID-19 exit could be undertaken in 6 months, with only 30% of population being under strict isolation for longer than 3 months – all without overwhelming the medical system and exceeding the ICU capacity at any point. In contrast, without such a model, simulations indicated that the complete exit would take 17 months[1] and 40% of the population would be subject to strict isolation for over a year, or the ICU capacity will be exceeded by over four times.
   Implication: Governments should invest in individual health data infrastructure to make such models implementable at scale. That is, infrastructure to not only collect and analyse data on a few thousand people who exhibited symptoms and went to hospitals, but also the individual medical data on the entire population to obtain health risk predictions for all residents; see [42] for further discussion on the resultant data policies, privacy and other related issues.

2. Even moderate-quality risk models could already bring non-insignificant results – that is, relaxing isolation for millions of people and months faster, while abiding to the existing constraints on medical resources.
   Implication: For immediate COVID-19 action focus on the "minimal viable product" data and models that can be used at scale. For example, age, body mass index, and hypertension and diabetes data – all of which can be assessed at a nearby pharmacy for all people within weeks, can already be used with a risk model such as in [16] to inform policies that could be relevant for practice.

[1] Such excessively long and massive lock-downs in the absence of a reasonable risk-model may, in fact, be an under-estimate, e.g., [1] propose isolating all seniors for 18 months, and even that hinges on the assumption that a vaccine will be developed by then.

3. Personalised policies based on risk-model predictions are highly sensitive to the protection level of confined people. Interestingly, the impact of protection level of deconfined people on the simulated outcomes depends on the risk-model quality. With a high-quality risk model, the optimal policy builds herd immunity, if policy choices allow, which can be done faster when deconfined people are less protected.

   Implication: Personalise resource allocation to protect the confined, predicted high-risk, people: distribute them masks and other PPE, supply with food and other necessities for free, prioritise testing those in contact with them, etc.

4. Lastly, whether an individual is classified as high- vs low-risk changes over depending on the state of the epidemic. The proposed personalised policy combines the epidemic progression with the data-science principles, and optimally adjusts the high- vs/ low-risk classification threshold so as to ensure safe and fast deconfinement.

   Implication: A careful communication strategy is needed to convey such personalised policies to the public.

This paper is organised as follows. First, because multiple research articles on COVID-19 are published daily, we do not present a formal literature review; instead some of us maintain an open-access repository of relevant literature with brief discussions here: https://github.com/MyrtoLimnios/covid19-biblio. Most of that literature is from the medical and related sciences, but there is a growing body of work in operations management too. [64] provides an excellent review. [17] and [37] are the working papers on COVID-19 that we are aware of; more will likely come up soon.

Second, because our findings are based on numerical simulations, the code is available via GitHub at: https://reine.cmla.ens-cachan.fr/boulant/seair, and the detailed description of the algorithmic details for how our model is implemented in the code is provided in [19]. This forms a "code appendix" to our paper.

Lastly, to facilitate dissemination of our results for the general public, we created a non-coding demo "simulator" for the differential policies that we study here: https://ipolcore.ipol.im/demo/clientApp/demo.html?id=305. It is pre-populated with the default parameters for France used in our simulations, but one can change the parameters and simulate the pandemic exit policies for other counties/regions given their respective situations.

Model

At a high level our model is a combination of a rather standard epidemiological model (compartmentalised SEAIR) with an equally standard machine learning risk model (binary classifier).

Extended SEAIR Model.

SEAIR is a deterministic model of ordinary differential equations (ODEs), which considers 5 categories of individuals of the following types:

- Susceptibles: those who have never contracted the disease;
- LatEnt, who have contracted the disease, but are asymptomatic and not yet contagious;
- Asymptomatic, who have contracted the disease, and are asymptomatic and contagious;
- Infectious, who have contracted the disease, and are symptomatic and contagious;
- Recovered, who have had the disease and have become immune to it.

We extend this base-model in two ways. First, we add an explicit compartment for people in ICU. Second, we split each compartment of the S, E, A, I, R categories into four sub-categories, with the ICU compartment split in two. Together with a compartment for people who died from the disease, they add up to a total of 23 compartments. The four subcategories correspond to the so-called "confusion matrix" of the machine learning risk prediction model (see Table A.1): (i) True Positives, who would actually experience severe symptoms upon infection needing and ICU bed, and classified as high-risk and hence confined, (ii) False Negatives, who would actually experience severe symptoms upon infection needing and ICU bed, but classified low-risk and hence released, False Positives, who would actually experience only mild symptoms upon infection not needing an ICU bed, but classified high-risk and hence confined, and True Negatives, who would actually experience only mild symptoms upon infection not needing an ICU bed, and classified low-risk and hence released. Each individual falls into exactly one of these four groups, but one's membership in a group is determined endogenously by our model as is explained in the next section.

Because this notation is critically important going forward, we reiterate that the policy we study confines those who are predicted to have high-risk, and releases those who are predicted to have low-risk. As any model, our (assumed) prediction model makes mistakes, and therefore both the confined and the released groups contain a mix of actually severe- and actually mild-symptom individuals. The sub- and super-scripts $j = \{s, m\}$, for (actually) "severe" vs "mild", and $i = \{c, r\}$, for "confined" (i.e., predicted severe) vs "released" (i.e., predicted mild), designate the sub-categories in each compartment. For example, $S_s^{(r)}$ refers to susceptibles who are released, but will get severe symptoms when infected; $S^{(r)}$ refers to all released, and so on.

We use $\rho \in [0, 1]$ as the control parameter of our policy, denoting the proportion of individuals who should be released, i.e., those classified by the risk model as the low-risk group (correctly or incorrectly) and therefore subject to low isolation restrictions. We initially consider a single-release policy (in which we optimise over a scalar ρ), and then extend the analyses to multiple release epochs, in which we optimise over a vector $\{\rho_0, \rho_1, ...\}$ released at times $\{0, t_1, ...\}$.

The impact of the differentiated isolation restrictions on people's behaviour is captured by two behavioural parameters: δ_r, for the group with low isolation restrictions, i.e., released, and δ_c, for the group with high isolation restrictions, i.e., confined; $0 \leq \delta_r < \delta_c \leq 1$. These parameters capture a level of "protection," which may aggregate several factors such as respiratory and hand hygiene, how much a person has lowered the number of exits from home and social interactions, etc.

Note, that how individuals in group $i = \{r, c\}$ reduce their chances of contracting the disease not only depends on δ_r and δ_c, but also on the proportion of people in each group, ρ and $1-\rho$. The so-called "contact rates," c_r and c_c for the released and confined groups – important parameters in "standard" epidemiological models satisfy $1 - c_i = (1 - \delta_i) \times ((1-\delta_r)\rho + (1-\delta_c)(1-\rho)), \quad i \in \{r, c\}$. These parameters have been used in the literature modelling the current lock-down, e.g., [34, 32]. Our approach is fully aligned with those, as we endogenise contact rates per the preceding equation.

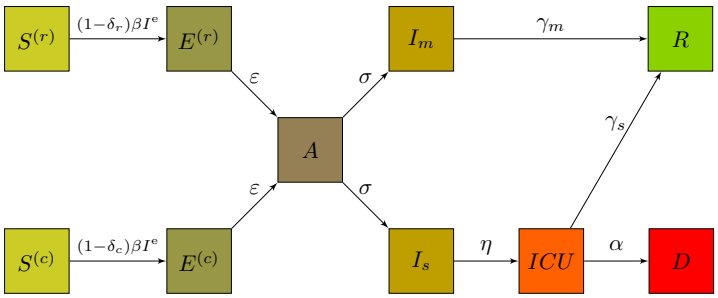


FIGURE A.1: Simplified schematic of the risk-extended SEAIR model, showing rates of passage from the different compartments. Displayed compartments are the released susceptibles, $S^{(r)}$, the confined susceptibles, $S^{(c)}$, the released latent, $E^{(r)}$, the confined latent, $E^{(c)}$ the asymptomatic, A, the infectious with severe symptoms requiring ICU, I_s, the infectious with milder symptoms, I_m, the people in Intensive Care Unit, ICU, as well as those who died from the disease, D, and those who recovered and are immune, R. All parameters may be found in Table A.2.

Figure A.1 presents the simplified schematic of the SEAIR side of our model, which corresponds to the following set of ODEs (reduced from the full set of 23 equations for ease of exposition[2]); all parameters are in Table A.2:

$$
\begin{aligned}
\dot{S}^{(r)} &= -(1-\delta_r)\beta I^e S^{(r)} \\
\dot{E}^{(r)} &= (1-\delta_r)\beta I^e S^{(r)} - \varepsilon E^{(r)} \\
\dot{S}^{(c)} &= -(1-\delta_c)\beta I^e S^{(c)} \\
\dot{E}^{(c)} &= (1-\delta_c)\beta I^e S^{(c)} - \varepsilon E^{(c)} \\
\dot{A} &= \varepsilon E - \sigma A \\
\dot{I}_m &= \sigma A_m - \gamma_m I_m \\
\dot{I}_s &= \sigma A_s - \eta I_s \\
\dot{U} &= \eta I_s - (\gamma_s + \alpha) U \\
\dot{R} &= \gamma_m I_m + \gamma_s I_s \\
\dot{D} &= \alpha U.
\end{aligned}
$$

Here the effective number of contagious people is $I^e = (1-\delta_r)(A^{(r)} + I^{(r)}) + (1-\delta_c)(A^{(c)} + I^{(c)})$, and, given the so-called *basic reproduction number* $\mathcal{R}_0$, the transmission rate is $\beta = \frac{\mathcal{R}_0}{N_0} \times \frac{1}{p(\sigma^{-1}+\gamma_m^{-1})+(1-p)(\sigma^{-1}+\eta^{-1})}$, which is derived by stability analysis as in [34], using the so-called next generation matrix method [33].

[2]The full set of ODEs and the algorithmic details of their solution are presented in the "code appendix" [19].

	Actual, s	Actual, m	
Model, $s \to$ confine	True Positives = $(1-p) \times \int_T^1 f_s(\omega)$	False Positives = $p \times \int_T^1 f_m(\omega)$	$1-\rho$
Model, $m \to$ release	False Negatives = $(1-p) \times \int_0^T f_s(\omega)$	True Negatives = $p \times \int_0^T f_m(\omega)$	ρ
	$1-p$	p	

TABLE A.1: Confusion matrix of the risk model. f_s, f_m denote class-conditional predictive distributions; T is the classification threshold, p is the proportion of people with mild symptoms in the population, and ρ is the proportion of the released population – the decision variable in our model.

Risk Prediction / Classification Model.

Our model requires identifying individuals at highest risk of severity and correspondingly advising them to remain in strict isolation, while relaxing isolation restrictions for individuals at lower risk. Such identification is done in two steps following the common data science and machine learning approach. At step one, a risk-"score" is obtained for each individual with a logistic regression, random forest, gradient boosting, used in [16], or the like model. A standard metric to assess the discriminating power of such models is the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve [43]. At step two, individuals with risk scores above a certain threshold, T, are classified as high-risk and are confined, and the rest are classified as low risk and are released. T is determined endogenously so that the the proportion of the released population equals ρ, which is, recall, the decision variable in our model.

As mentioned in the introduction, we do not attempt training a risk-model of our own. Training such models requires access to non-trivial personalised data, and many research teams have better access to such data than we do. The goal of this paper is to evaluate the efficacy of the personalised pandemic management assuming such a model. Therefore, we create hypothetical risk models, which discriminating powers (measured by the AUC) bracket existing models in the literature, e.g., [16].

To create a hypothetical risk model, let f_s and f_m denote the so-called "predictive distributions" – the PDFs of the proportions of people having the severe and mild symptoms, respectively, as predicted by the model. Together with the threshold, T, and the proportion of the population with mild symptoms in the population, p, f_s, f_m define the model's confusion matrix per Table A.1.

As is evident from Table A.1, given ρ, p and "the model," i.e., f_s, f_m, the corresponding classification threshold T^* is selected such that:

$$(1-p) \times \int_0^{T^*} f_s(\omega) + p \times \int_0^{T^*} f_m(\omega) = \rho. \tag{A.1}$$

Let q^{FP} and q^{FN} denote the model's false-positive and false-negative error rates; i.e., $q^{FP} = \int_{T^*}^1 f_m(\omega)$ and $q^{FN} = \int_0^{T^*} f_s(\omega)$; for notational convenience we omit the dependency of qs on T^*, and through that, on ρ. Then (A.1) is equivalent to:

$$(1-p) \times q^{FN} + p \times (1-q^{FP}) = \rho, \tag{A.2}$$

which highlights a key relationship of (any) classification model that we exploit. Selecting a small ρ results in a small T^* and, thus, also a small q^{FN}. Relatively few people will be released, but very few of the released would be, by mistake, actually severe. Increasing ρ would not only increase T^*, hence releasing more people, but it would also increase q^{FN}, making a disproportional impact on the number of people who would require an ICU. An increase in q^{FN} will be smaller for a higher quality, i.e., higher AUC, model than for a lower-quality one. This is because $AUC = \int_0^1 (1-q^{FN})dq^{FP}$ and thus a higher AUC implies, ceteris paribus, lower q^{FN} at a given T.

Another important observation from Table A.1 is that the released individuals consist of two groups: True Negatives and False Negatives. The latter will experience severe symptoms upon infection, requiring ICU beds. However, they are not the only group which will require ICU: because the confinement of the non-released is imperfect ($\delta_c < 1$), some of the True Positives will also get infected (and will require ICUs). Setting a smaller ρ, decreases the number of FNs, but increases the number of TPs, thus leading to a non-trivial relationship between selecting ρ, the resultant threshold T^*, and ICU demand. This relationship, further, depends on the model's quality (AUC).

The goal is to select ρ so that, given the model's quality, confine only a few people, yet not to violate the ICU bed capacity due to the errors made by the model in identifying such people and due to imperfect confinement of the rest.

Connecting Risk and SEIR Models.

The risk-model connects with the SEAIR model as follows. For $Q \in \{S, E, A, I, U, R\}$ and a policy ρ, we re-scale the initial risk-independent epidemiologic conditions Q_0 to account for the distribution of people in the four groups:

$$\begin{aligned} Q_m^{(r)}(0) &\leftarrow p\,(1-q^{FP})Q_0, \qquad & Q_m^{(c)}(0) &\leftarrow p\,q^{FP}Q_0, \\ Q_s^{(r)}(0) &\leftarrow (1-p)\,q^{FN}Q_0, \qquad & Q_s^{(c)}(0) &\leftarrow (1-p)\,(1-q^{FN})Q_0. \end{aligned}$$

This results in the starting "day 0" conditions for the 23 compartments in the extended SEAIR model, which all depend on ρ:

$$\begin{aligned} &(S_m^{(r)}, E_m^{(r)}, A_m^{(r)}, I_m^{(r)}, R_m^{(r)}, S_m^{(c)}, E_m^{(c)}, A_m^{(c)}, I_m^{(c)}, R_m^{(c)}, S_s^{(c)}, E_s^{(c)}, A_s^{(c)}, I_s^{(c)}, U^{(c)}, R_s^{(c)}, S_s^{(r)}, \\ &...E_s^{(r)}, A_s^{(r)}, I_s^{(r)}, U^{(r)}, R_s^{(r)}, D)(0). \end{aligned} \tag{A.3}$$

As we investigate policies changing over time, we also update the numbers of people in each compartment when the decision-maker increases ρ from some value ρ_{old} to ρ_{new}, with corresponding false positive and false negatives rates q_{old}^{FP}, q_{old}^{FN} and q_{new}^{FP}, q_{new}^{FN}, respectively:

$$\begin{aligned} Q_m^{(r)} &\leftarrow Q_m^{(r)} + \frac{q_{\text{old}}^{FP} - q_{\text{new}}^{FP}}{q_{\text{old}}^{FP}} Q_m^{(c)}, \qquad & Q_m^{(c)} &\leftarrow Q_m^{(c)} - \frac{q_{\text{old}}^{FP} - q_{\text{new}}^{FP}}{q_{\text{old}}^{FP}} Q_m^{(c)}, \\ Q_s^{(r)} &\leftarrow Q_s^{(r)} + \frac{q_{\text{new}}^{FN} - q_{\text{old}}^{FN}}{1 - q_{\text{old}}^{FN}} Q_s^{(c)}, \qquad & Q_s^{(c)} &\leftarrow Q_s^{(c)} - \frac{q_{\text{new}}^{FN} - q_{\text{old}}^{FN}}{1 - q_{\text{old}}^{FN}} Q_s^{(c)}. \end{aligned}$$

Symbol	Description	Value(s)	Reference
N_0	total initial number of people in the population	6.710^7	
S_0	total initial number of infected people in the population	computed	
E_0	total initial number of exposed people in the population	case-dependent	estimated
A_0	total initial number of asymptomatic people in the population	case-dependent	estimated
I_0	total initial number of infected people in the population	case-dependent	estimated
U_0	total initial number of people in ICU	case-dependent	known/estimated
R_0	total initial number of immune people in the population	case-dependent	[95]/estimated
I_{max}	hospital capacity for COVID-19 ICU beds	7 250	assumed
p	proportion with mild symptoms (prior)	$0.9932\,[0.9891-0.9961]$	[95]
β	transmission rate	computed	
$\mathcal{R}_0$	basic reproduction number	2.9	[95]
ε	waiting rate to viral shedding	$1/3.7$ day^{-1}	[32]
σ	waiting rate to symptom onset	$1/1.5$ day^{-1}	[32]
η	waiting rate from symptom onset to ICU	$1/7$ day^{-1}	[95]
γ_m	recovery rate from mild symptoms	$1/2.3$ day^{-1}	[32]
γ_s	recovery rate for people in ICU	$1/17$ day^{-1}	[95]
α	mortality rate for people in ICU	$1/11.7$ day^{-1}	[95]

TABLE A.2: Simulation parameters used with relevant 95% confidence intervals.

Summary of Key Parameters and Data.

All parameters used in simulations are listed in Table A.2. Parameters for the initial conditions S_0, E_0, A_0, I_0, U_0 and R_0 depend on the investigated scenario: "day 0" is either set on March 17, 2020 – the first day of country-wide lock-down in France, or "day 0" is set on May 11, 2020 – the beginning day for the lock-down exit.

The initial number of susceptibles S_0 is computed as $S_0 = N_0 - (E_0 + A_0 + I_0 + R_0 + U_0)$.

The class-conditional predictive distributions are modelled as Beta distributions: $f_s \sim \text{Beta}(a_s, b_s)$ and $f_m \sim \text{Beta}(a_m, b_m)$. In simulated scenarios, the *no model* refers to $a_m = b_m = a_s = b_s = 1$, and otherwise we fix $b_s = a_m = 2$ and vary[3] $a_s = b_m$. The *low AUC model* refers to $a_s = b_m = 3$ (AUC∼75%) and the *high AUC model* refers to $a_s = b_m = 5$ (AUC∼96%). These parameters are selected so as to bracket the "high" (AUC∼82%) and "low" (AUC∼93%) models from [16]. For sensitivity analyses we explore the range from $a_s = b_m = 2.5$ (AUC∼65%) to $a_s = b_m = 6.5$ (AUC∼99%), further bracketing the range of models that are likely to be available in practice for a disease like COVID-19.

Estimation of missing parameters and construction of confidence intervals.

Most parameters are not known with certainty. To that end, we focused on the key parameters driving the simulation results: the fraction p of individuals with mild symptoms if infected, the reduction of contact rates during lock-down c, and the numbers of people exposed/asymptomatic/infected at the beginning of lock-down, on March 17.

We used Bayesian methods to estimate the joint distribution of these parameters by comparing model predictions to the actual data for ICU occupancy from March 17 to May 11, 2020 obtained from the official portal of the French government: `https://dashboard.covid19.data.gouv.fr`. A noteworthy complication in using the standard Bayesian methods is that the likelihood function for the resultant prediction errors is unknown and likely highly non-trivial as the errors are not independent over time

[3]The symmetric ROC models that we considered are certainly not the only possible approach. While working on this paper we explored multiple non-symmetric settings, and observed no qualitative impact on the simulation results.

(intuitively, if one SIR-like curve is higher than another early in a time horizon, it must get lower at a later time, etc.)

To deal with this complication we utilised Approximate Bayesian Computation method (ABC) [85], which has been specifically designed for such situations. The ABC method was implemented with the root mean standard error as a distance function [20], with a maximum error set at 1 000 beds over the 55 data points, which corresponds to an "acceptance rate" of about 10%. We assumed the prior distributions to be independent with the following choices: Beta distribution for p with parameters $(2265, 15.6)$, fitting the mean and 95% confidence interval observed in [95], uniform distribution for the total number of people exposed/asymptomatic/infected on March 17 with range between 0.5 and 1.4 million (recall that 1.5 million of people were estimated to be exposed/asymptomatic/infected/immune at that time), and uniform distribution for c between 65% and 75%, which is consistent with [95].

The number of samples for the prior distribution for p and the initial number of exposed/asymptomatic/infected at the beginning of lock-down $E_0+A_0+I_0$ was set at $n=10\ 000$ (resp. 100 000 for robustness when computing means).

This led to around 1 000 (resp. 10 000) posterior samples as the acceptance rate was at 10%. In order to reduce the parameter space, we have estimated the total number of people exposed/asymptomatic/infected and inferred the number in each state by using the fractions of the mean time spent in each category in the majority population, i.e. that of people with mild symptoms. More precisely, this corresponds to setting $x := (1+\varepsilon\left(\sigma^{-1}+\gamma_m^{-1}\right))^{-1}$ and then

$$E_0 \leftarrow x(E_0+A_0+I_0), \qquad A_0 \leftarrow \frac{\varepsilon}{\sigma}x(E_0+A_0+I_0), \qquad I_0 \leftarrow \frac{\varepsilon}{\gamma_m}x(E_0+A_0+I_0).$$

The mean posterior values were found to be $p\sim 0.993$ and $E_0+A_0+I_0\sim 1200000$ (for March 17). The mean posterior value for c was found to be $c\sim 69.2\%$. Note that the latter value is unused in our numerical experiments, since we investigate scenarios with differentiated isolation policies.

We remark that by modelling uncertainty in p, we implicitly introduced uncertainty in the risk model as the error rates q^{FN}, q^{FP} solve equation (A.2) where p is a parameter. We acknowledge that other sources of uncertainty in risk models could also exist[4].

Simulations with 95% confidence intervals.

In all figures showing the evolution of the number of people in ICU (see Figure A.2), the initial condition $E_0+A_0+I_0$ and the proportion of people p not requiring ICU

[4] Uncertainty in risk models, i.e., in risk predictions and even in the classification threshold, could be due sampling bias (e.g. clinical trial data used are not reflecting population data or electronic medical records failing to account for the part of the population who was never admitted to the hospital) or methodological bias (model misspecification, suboptimal machine learning/statistical method used). To provide statistical guarantees on the risk estimators, it may be necessary to compute confidence bands on the estimated ROC curve which will then lead to explicit confidence bands on the risk model. Typical approaches to derive confidence bands are to perform error propagation on distribution parameters (in a parametric framework), or to generate several ROC curves and Precision-Recall curves through resampling and provide some bootstrap estimate of the confidence band [84]. Resampling strategies may include label flipping (prediction uncertainty), sample perturbation or shifting (sampling bias), etc. Lastly, the quality of a prediction model was assessed by AUC; other measures also exist, [94].

were sampled according to their posterior distribution. The mean curve of Figure A.2 is obtained by taking the average of all the sampled curves, while subsequent 95% confidence intervals were derived by removing the 2.5% and 2.5% upper and lower values for the computed number of ICU beds at each time.

Simulations with grid searches.

As some numerical experiments (see Figures A.3, A.4 and Table A.3) require grid searches, we did not sample according to the posterior distribution for each scenario, but rather computed mean values in order to ease the computational burden.

- March 17, 2020: simulations are operationalised as follows:
  - the initial number of utilised ICU beds is known $I_0 \sim 700$, and the estimate for the total number $E_0 + A_0 + I_0 + R_0 \sim 1.5.10^6$ is also available from [95];
  - therefore, we took the average over the posterior and obtained $E_0 + A_0 + I_0 \sim 1200000$, resuting in $R_0 \sim 300000$ people;
  - similarly, taking the mean along posterior samples resulted in the estimate of $p \sim 0.993$;
  - the ODE system was then integrated up until 200 days past March 17, 2020.
- May 11, 2020: simulations are operationalised as follows:
  - sampling according to the posterior for p and $E_0 + A_0 + I_0$, and integrating the ODE system from March 17 to May 11, we obtained a sample of initial conditions for May 11, of which we took the averages to obtain estimates for all parameters,
  - the ODE system was then integrated up until 200 days past May 11, 2020.

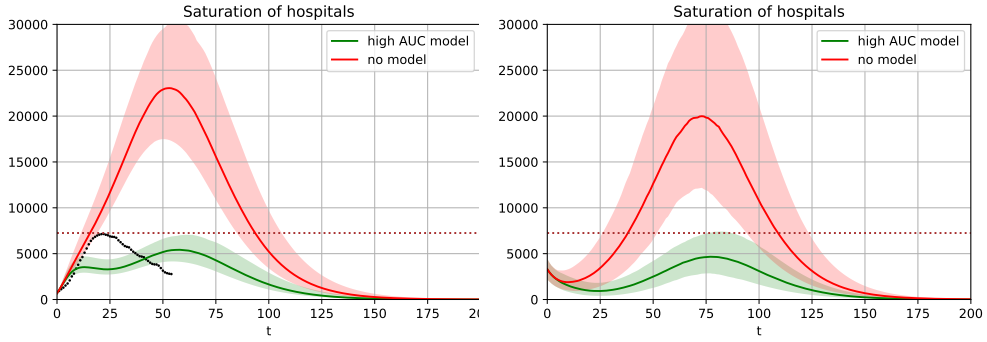
Results

We present the results of our simulations in three steps. First, we consider a partial exit problem with a single release. Second, we explore the sensitivity of the single release problem. Third, we consider the complete exit problem over multiple release epochs, as well as discuss its sensitivity.

Partial exit with a single release.

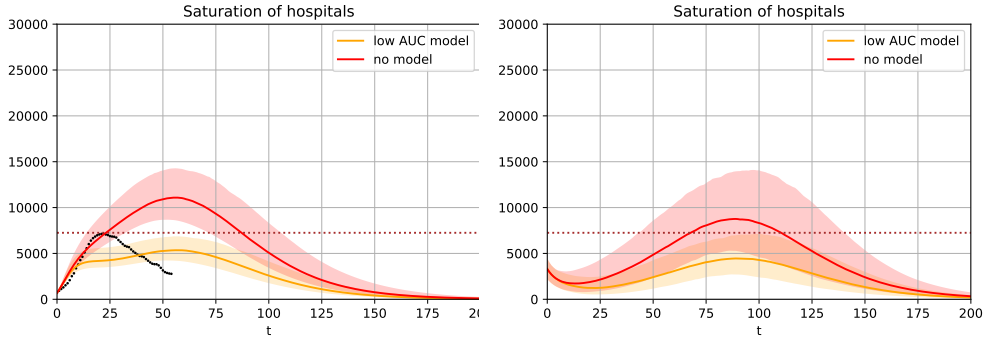
Figure A.2 displays the number of individuals requiring an ICU bed w.r.t. time t. The March 17 scenario is in the left column, the May 11 scenario is in the right. Two risk models are considered: a "high" AUC$\sim$95.99% (top row) and a low AUC$\sim$ 75.71% (bottom row), bracketing the performance of initial risk models developed for COVID-19 [16, 104].

In each plot, ρ represents the maximal percentage of the population that is released, i.e., submitted to lighter restrictions, which is assumed to correspond to $\delta_r = 0.1$, in



(a) $\rho = 62\%$

(b) $\rho = 70\%$



(c) $\rho = 53\%$

(d) $\rho = 60\%$

FIGURE A.2: Number of individuals requiring an ICU bed w.r.t. time t (days). Left column starts on March 17 (the day of the initial lock-down in France), right column starts on May 11 (the day when lock-down ends). The dotted line on the left column shows the actual data for France from March 17 to May 11. Top row uses a risk prediction model with high AUC$\sim 95.99\%$, bottom row uses a risk prediction model with low AUC$\sim 75.71\%$. Without a model, $\rho = 45\%$ for March 17, and $\rho = 55\%$ for May 11.

such a way that the 95% confidence interval of the number of individuals requiring an ICU bed when using the risk prediction model (green and orange curves) remains below the maximum number of ICU beds assumed (7 250). In these first simulations, the rest of the population is confined with stricter restrictions, $\delta_c = 0.9$. Finally, the red curves show the number of individuals requiring an ICU bed w.r.t. time if the same ρ of population is released, but selected at random without any risk prediction model.

Figure A.2 shows that a high-AUC model (green curve) allows for having 62% released (corresponding to a decrease of social interaction by 46%) from March 17 on, while a low-AUC model (orange curve) enables only 53%. In France, with a population of 67 million, these percentage differences correspond to ~ 6 million people. Perhaps more importantly, without a model, $\rho = 45\%$ for March 17 – a 17% and 8% difference, respectively, or some 5 to 11 million people.

Plots for lock-down exit strategies (May 11) investigate the effect of the same risk prediction accuracies. All differences (62% vs 53% vs 45% for March 17, 70% vs 60% vs 55% for May 11) are statistically significant at the 5% level. Lastly, without a risk prediction model, the ICU beds demand greatly exceeds the current capacity at either ρ.

Sensitivity analyses of the single release policy.

Figure A.3 presents sensitivity analyses of the difference between the maximal percentage of people which may be released without exceeding ICU capacity for several risk-prediction models, relative to the maximal percentage, but with no risk-prediction model. The results are shown for both the March 17 and the May 11 scenarios. Sensitivity is tested with respect to the discrimination performance of the risk prediction models (AUC) and the degree of isolation of the confined population (δ_c). We also alter the degree of isolation for the released population (δ_r) across different plots.

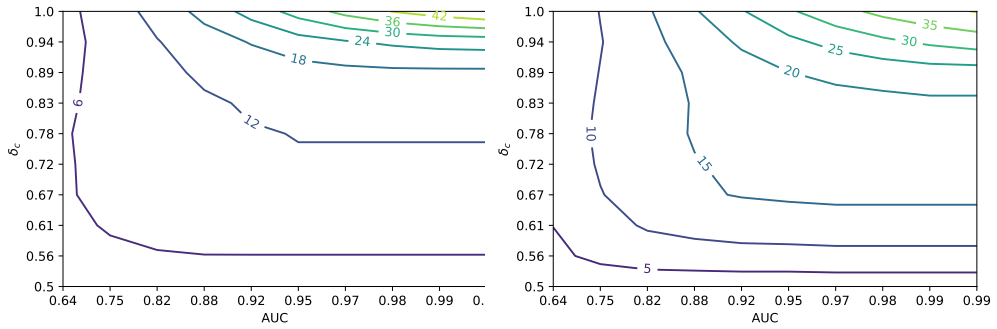
As expected, the higher the discrimination of the prediction model, the bigger the difference. However, the degree of isolation has different impact depending on who is considered: for the confined population the stricter the isolation (the higher is δ_c) the larger the impact of the risk prediction model. But for the released population ($\delta_r = 0.1$ or 0.2 in Figure A.3), the results are more intricate. It is often also better to isolate more strictly, except when the risk prediction model is of very high quality and the confined people are in very strict isolation. In those situations the optimal ρ is large enough to achieve "herd immunity" ($1-\frac{1}{\mathcal{R}_0} \approx 65\%$) which can be done faster if the released population is less protected. The main implication of these analyses is that it important to both assume in models and encourage in practice the stricter isolation practices for the high risk population, for example by focusing distribution of PPEs and other resources, strictly isolating nursing homes, etc., as this can not only protect the high-risk group better, but also allow for more efficient / faster exit from the pandemic for the rest.

Figure A.4 shows sensitivity analyses regarding the percentage p of the population with severe symptoms upon infection (which we decrease from the posterior average ~ 0.993 to 0.98) and the ICU capacity (which we increase from 7 500 to 15 000). Other parameters are same as in Figure A.3.

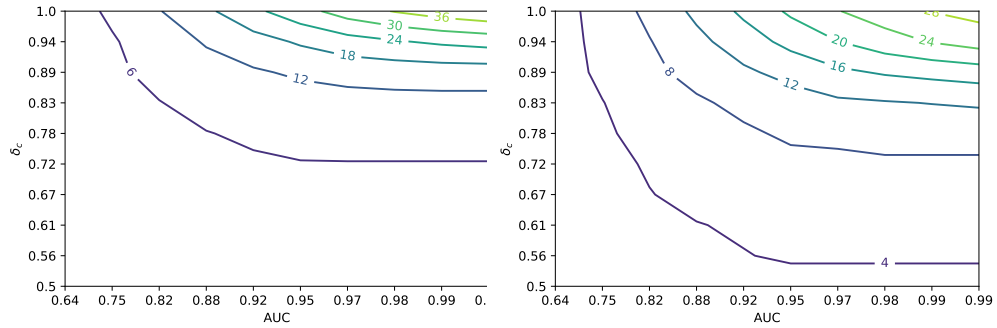
First, as expected, the lower p the less the impact of a risk prediction model keeping the risk-model AUC constant: given the limited ICU – and possibly other – resources, a smaller p allows for a smaller range of percentages of the population being released, making all differences between policies smaller in absolute terms. Second, when we compare Figures A.3 (a) and (c) with Figures A.4 (a) and (c), where the only difference is in the total ICU capacity, we see that the more ICUs available the larger the impact of using personalised policies, keeping everything else constant. More available resources allow for a larger range of percentage of released people making the differences between policies – risk based vs not – larger in absolute terms. Note that in all cases a risk prediction model approach allows for confining fewer people: this is consistent with value of information related arguments, as any test provides information which can be beneficial assuming everything else (including behavioural aspects) kept constant.

Complete exit with multiple release epochs

Finally, we explored the implications of our simulation for gradual exit strategies. To consider practical and realistic scenarios we solved the resultant optimisation dynamic program allowing releases every 30 days at multiples of 5% of the population, while ensuring that the maximum number of utilised ICU beds was at most 5 250. We found that implementing the full-blown confidence interval analyses as in



(a) March 17, $\delta_r = 0.1$ (b) March 17, $\delta_r = 0.2$



(c) May 11, $\delta_r = 0.1$ (d) May 11, $\delta_r = 0.2$

FIGURE A.3: Difference in maximum possible percentage of people in low isolation without hospital saturation. Maximal number of people in the low isolation group without exceeding the limit of 7 250 beds, with a margin of 2 000 beds as imposed by typical 95% confidence intervals, compared to the case of not using a risk prediction model. Plotted as a function of the AUC of a risk prediction model and the protection level δ_c for people recommended to be in isolation with stricter restrictions. $p \sim 0.993$ for all figures.

Figure A.2 was computationally intractable in the dynamic program setting. But observing that the maximum confidence range in Figure A.2 was ~ 2000 beds, we reduced the ICU capacity from the "base-case" of 7 250 to 5 250 to account for uncertainty in ICU demand.

	Four epochs, March 17		
	High AUC model	Low AUC model	No model
$\delta_c = 0.9$	6	>12	>12
$\delta_c = 0.8$	8	>12	>12
$\delta_c = 0.7$	10	>12	>12

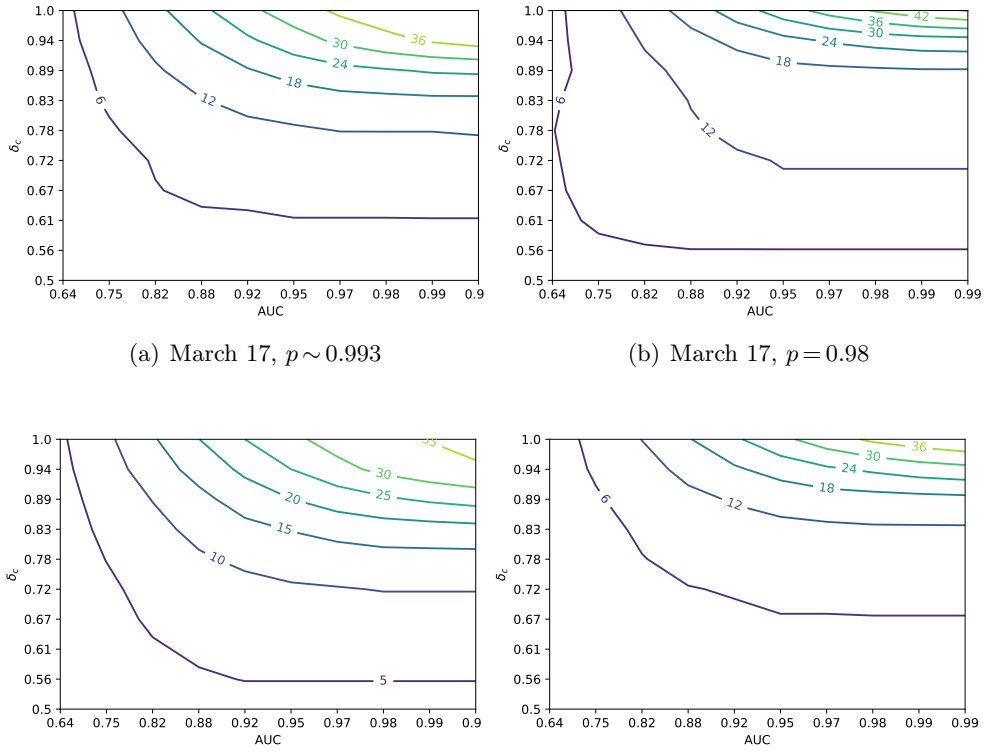
	Three epochs, March 17		
	High AUC model	Low AUC model	No model
$\delta_c = 0.9$	7	>12	>12
$\delta_c = 0.8$	11	>12	>12
$\delta_c = 0.7$	>12	>12	>12

	Four epochs, May 11		
	High AUC model	Low AUC model	No model
$\delta_c = 0.9$	6	11	>12
$\delta_c = 0.8$	7	11	>12
$\delta_c = 0.7$	9	11	>12

	Three epochs, May 11		
	High AUC model	Low AUC model	No model
$\delta_c = 0.9$	6	>12	>12
$\delta_c = 0.8$	8	>12	>12
$\delta_c = 0.7$	12	>12	>12

TABLE A.3: Minimal time (in months) required for all people to exit isolation, starting from March 17 or May 11, depending on δ_c, model quality and the number of epochs of gradual deconfinement. $\delta_r = 0.1$

Table A.3 shows the total number of months to release the entire population for different scenarios, keeping all other parameters constant: no model, low-AUC model,



(a) March 17, $p \sim 0.993$

(b) March 17, $p = 0.98$

(c) May 11, $p \sim 0.993$

(d) May 11, $p = 0.98$

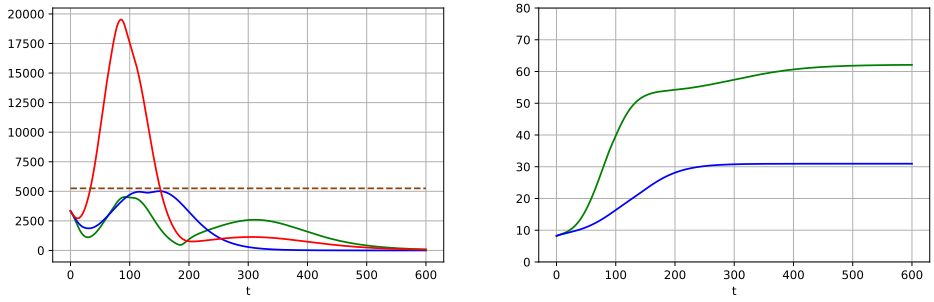
FIGURE A.4: Difference in maximum possible percentage of people in low isolation without hospital saturation. Maximal number of people in the low isolation group without exceeding the limit of 15 000 beds, with a margin of 2 000 beds as imposed by typical 95% confidence intervals, compared to the case of not using a risk prediction model. Plotted as a function of the AUC of a risk prediction model and the protection level δ_c for people recommended to be in isolation with stricter restrictions. $\delta_r = 0.1$ for all figures.

high-AUC model, and for three different values of δ_c. We considered only gradual releases in 3 or 4 epochs – the ICU system was overwhelmed when using only 2 epochs for most simulations. The main insight is that using no risk model would require more than a year in all scenarios, while exit with risk-based models would lead to relaxing restrictions for the entire population as fast as over 6 months.

Figure A.5 shows example gradual policies for May 11 corresponding to the four epochs from Table A.3, and assuming $\delta_c = 0.9$ as in Figure A.2. The insights complement those for single release policies: with risk-prediction models. A smaller percentage of the population may need to be confined; consequently, one could also reach the moment when isolation measures could be lifted sooner.

For example, using the high-AUC model and without exceeding the ICU capacity at any point, 65% of the lowest-risk population could be released on May 11 ("day 0"), followed by another 5% on July 10 ("day 60"), and yet another 20% on August 9 ("day 90"), and finally releasing the remaining 10% on November 7 ("day 180"). The resultant ICU demand is shown as a green line on Figure A.5a.

Implementing the same exit schedule without a model would lead to ICU demand of nearly 20 000 beds (red line). In contrast, a capacity-abiding exit strategy without a model (blue line) would require 17 months to reach full deconfinement. Only 50%



(a) ICU demand

(b) Percent Immune

FIGURE A.5: Examples of gradual schedules of relaxing isolation restrictions with and without model-based risk predictions. High AUC model (green) and no model (red) $\rho = [0.65, 0.7, 0.9, 1]$ and $\mathbf{t} = [60, 90, 180, 600]$, no model (blue): $\rho = [0.5, 0.55, 0.6, 1]$ and $\mathbf{t} = [30, 120, 510, 600]$. Vectors $\mathbf{t} = [t_1, t_2]$ give the population release schedules $\rho = [\rho_1, \rho_2]$ as follows: ρ_1.100% of the population is released on day 0, then $(\rho_2 - \rho_1)$.100% are released on day t_1, etc.

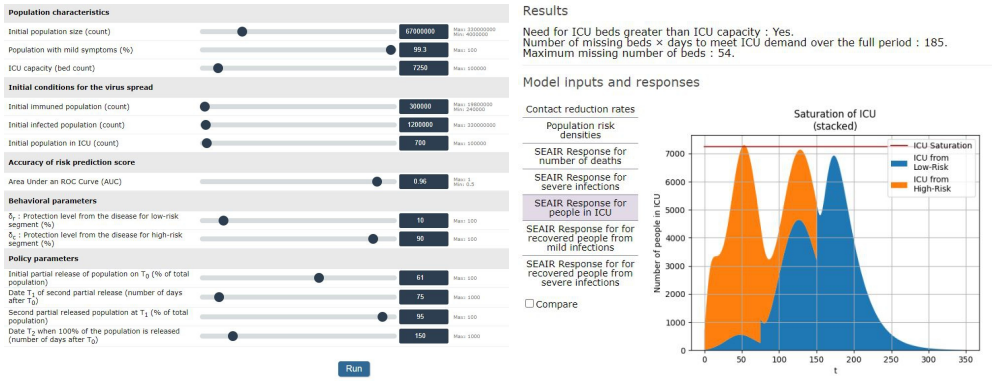
of population could be in low isolation on "day 0" (May 11), another 5% on "day 30" (June 10), additional 5% on "day 120" (September 8), and the last 40% only on "day 510" (October 3, 2021), or 11 months later than the similar risk-model-based strategy. Such an extended isolation would also apply to many more people: 10% with the model versus 40% without; for France this means the additional ∼20 million people in isolation for the additional 11 months.

For both scenarios, Figure A.5b shows the percentage of the population that becomes immune over time. Because the model-based policy releases larger portion of the low-risk population and does so faster, it approaches herd immunity, allowing for the ultimate protection against the disease. In contrast, herd immunity is not achieved by a policy without the risk-model: the disease is suppressed, but could explode again. In other words, assuming everything else constant, our simulations indicated that, using risk prediction models, isolation restriction may be relaxed faster and safer.

Discussion

Data-driven prediction models, which made large impacts in many areas the past decades, can enable, among others, personalisation of policies for managing epidemic outbreaks. We studied how prediction models for the severity of symptoms upon infection could be used in epidemic simulations to study the effect of non-pharmaceutical policies, particularly isolation restrictions, during an outbreak. We used COVID-19 data from France as of early May 2020 as an example, and provided sensitivity analyses to understand how different parameters could impact pandemic isolation and exit policies.

Simulations indicated that considering differential relaxation of isolation restrictions depending on predicted severity risk can decrease the immediate percentage of the population in France under stricter isolation by $\sim 5 - 20\%$ relative to not using such risk predictions, and fasten the complete exit by several months, thus directly impacting lives of millions of people.



(a) Parameters (b) Results

FIGURE A.6: Screenshots of the demo simulator available here: https://ipolcore.ipol.im/demo/clientApp/demo.html?id=305.

We made our simulation engine available to a broad, non-technical audience via an interactive demo that is available at: https://ipolcore.ipol.im/demo/clientApp/demo.html?id=305 and is illustrated on Figure A.6. The demo is pre-populated with the model parameters for France that we used in this paper, but one can input the parameters for other countries or regions and experiment with the envisioned policy parameters. We hope this demo facilitates the dissemination of our work and raises awareness of the promise of personalised data-driven policies for pandemic management.

Sensitivity analyses showed that the qualitative insights from our simulations are robust to changes in risk prediction accuracy, percentage of severe-if-infected cases in the population, availability of resources (such as ICUs), and social distancing. Benefits increased when risk prediction accuracy increased, percentage of severe-if-infected cases in the population decreased, availability of resources (such as ICUs) increased, and the isolation of high-risk individuals increased. All results were developed using hypothetical risk prediction models for COVID-19, with discrimination ranges in line with early indications from initial models developed as of early May 2020 such as [16, 104].

The proposed approach can also be adopted for other epidemic models, and personalisation can further be explored using this approach for policies other than isolation restrictions. Moreover, predicted risk based isolation restrictions can be combined with other policies such as test-based ones, e.g., [89, 106], possibly also using other relevant prediction models, to limit the impact of outbreaks such as COVID-19. Finally, the same analysis can be done focusing on mortality or other outcomes instead of ICU demand that we focused on in this paper.

Several caveats should be noted. First, epidemic models – and the conclusions they may support – rely on a number of parameters, for example virus incubation and recovery times and the basic reproduction number $\mathcal{R}_0$, while the effects of policies also depend on healthcare system factors such as the availability of relevant resources (e.g., trained personnel). Second, these parameters are uncertain and evolve dynamically [4]; the resultant policies are therefore contingent. Observing an ICU demand that is closer to an upper boundary of the confidence interval may require the next wave to be delayed or involve a smaller release percentage than out current simulations built from day zero suggest. Third, policy decisions

require careful context-specific robustness analysis; however, using risk prediction models can at worst make no significant difference while at best improve policies by a significant margin, fixing all other conditions. Fourth, risk prediction models cannot be used when, or for people, for whom the necessary data is unavailable. In this case, simple models (e.g., based only on age and some reliable chronic disease data) may need to be used, which may limit the benefits of the approach. Finally, risk-predictions based policies using epidemic simulations should be developed taking into account behavioural aspects that may both help and hurt any model predictions and policy actions; ethical issues, fear, widespread non-compliance to isolation measures, and the likes.

In conclusion, our simulations show that combining prediction models using data science and machine learning principles with epidemiological models may improve outbreak management policies. Governments should thus make substantial investments in the data infrastructure necessary to implement such models at scale, and consider their predictions when developing pandemic isolation (confinement) and exit (deconfinement) policies.

Bibliography

[1] Daron Acemoglu et al. "A Multi-Risk SIR Model with Optimally Targeted Lockdown". In: Working Paper Series 27102 (2020).

[2] S. Agrawal, Z. Wang, and Y. Ye. "A Dynamic Near-Optimal Algorithm for Online Linear Programming". In: *Operations Research* 62.4 (2014), pp. 876–890.

[3] Toshihiro Akagi et al. "Exact Algorithms for the Max-Min Dispersion Problem". In: *Frontiers in Algorithmics*. Ed. by Jianer Chen and Pinyan Lu. 2018.

[4] Andres Alban et al. "ICU Capacity Management During the COVID-19 Pandemic Using a Stochastic Process Simulation". In: *SSRN Electronic Journal* (Jan. 2020).

[5] Susanne Albers and Leon Ladewig. "New Results for the k-Secretary Problem". In: *30th International Symposium on Algorithms and Computation (ISAAC 2019)*. Ed. by Pinyan Lu and Guochuan Zhang. Vol. 149. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, 18:1–18:19.

[6] David Allen et al. "Hierarchical Random Graphs for Networks with Weighted Edges and Multiple Edge Attributes". In: *International Conference on Data Mining (DMIN)*. Jan. 2011.

[7] L.J.S. Allen. "An Introduction to Stochastic Epidemic Models". In: *Brauer F., van den Driessche P., Wu J. (eds) Mathematical Epidemiology. Lecture Notes in Mathematics, vol 1945. Springer, Berlin, Heidelberg.* (2008).

[8] M. Babaioff et al. "A Knapsack Secretary Problem with Applications". In: *APPROX-RANDOM*. 2007.

[9] M. Babaioff et al. "Online Auctions and Generalized Secretary Problems". In: *SIGecom Exchanges*. Vol. 7. ACM, 2008, pp. 1–11.

[10] Albert-László Barabási and Réka Albert. "Emergence of Scaling in Random Networks". In: *Science* 286.5439 (1999), pp. 509–512.

[11] R. Bartoszyński and Z. Govindarajulu. "The Secretary Problem with Interview Cost". In: *Sankhyā: The Indian Journal of Statistics, Series B (1960-2002)* 40.1/2 (1978), pp. 11–28.

[12] M. Bateni, M. Hajiahghayi, and M. Zadimoghaddam. "Submodular secretary problem and extension". In: *ACM Transactions on Algorithms*. Vol. 9. 2013.

[13] J.N. Bearden. "A new secretary problem with rank-based selection and cardinal payoffs". In: *Journal of Mathematical Psychology*. Vol. 50. 2006, pp. 58–59.

[14] J.N. Bearden, A. Rapoport, and R.O. Murphy. "Experimental studies of sequential selection and assignment with relative ranks". In: *Journal of Behavioral Decision Making*. Vol. 19. 2006, pp. 229–250.

[15] R. Bekker, G. Koole, and D. Roubos. "Flexible bed allocations for hospital wards". In: *Health Care Management Science* 20.4 (2017), pp. 453–466.

[16] Dimitris Bertsimas. *Mortality Risk Calculatory*. 2020. URL: https://www.covidanalytics.io/calculator.

[17] John R Birge, Ozan Candogan, and Yiding Feng. "Controlling Epidemic Spread: Reducing Economic Losses with Targeted Closures". In: *University of Chicago, Becker Friedman Institute for Economics Working Paper* 2020-57 (2020).

[18] Christian Borgs et al. "How to distribute antidote to control epidemics". In: *Random Structures and Algorithms* 37.2 (2010), pp. 204–222.

[19] Olivier Boulant et al. *An SEAIR model with personalised risk prediction scores and application to the Covid-19 epidemic*. 2020. URL: https://www.ipol.im/pub/pre/305/.

[20] Tom Britton. "Stochastic epidemic models: a survey". In: *Mathematical biosciences* 225.1 (2010), pp. 24–35.

[21] A.Z. Broder et al. "The Hiring Problem and Lake Wobegon Strategies". In: *SIAM Journal on Computing*. Vol. 39. 2009, pp. 1223–1255.

[22] F.T. Bruss. "Sum the odds to one and stop". In: *The Annals of Probability*. Vol. 26. 2000, pp. 1384–1391.

[23] T.F. Bruss. "What is known about Robbins problem?" In: *Applied Probability*. Vol. 42. 2005, pp. 108–120.

[24] R. Chandrasekaran and A. Daughety. "Location on Tree Networks: P-Centre and n-Dispersion Problems". In: *Mathematics of Operations Research* 6.1 (1981), pp. 50–57.

[25] D. Chen et al. "A dynamic vaccination strategy to suppress the recurrent epidemic outbreaks". In: *Chaos, Solitons & Fractals* 113 (2018), pp. 108–114.

[26] D. Chisca et al. "From Offline to Online Kidney Exchange Optimization". In: *IEEE International Conference on Tools with Artificial Intelligence* (2018), pp. 587–591.

[27] K. I. Choe and D. S. Bai. "A secretary problem with backward solicitation and uncertain employment". In: *Journal of Applied Probability* 20.4 (1983), pp. 891–896.

[28] Reuven Cohen, Shlomo Havlin, and Daniel Ben-Avraham. "Efficient immunization strategies for computer networks and populations". In: *Physical review letters* 91.24 (2003), p. 247901.

[29] Jnaneshwar Das et al. "Data-driven robotic sampling for marine ecosystem monitoring". English (US). In: *International Journal of Robotics Research* 34.12 (2015), pp. 1435–1452.

[30] H. A. David and H. N. Nagaraja. "Order Statistics". In: *Encyclopedia of Statistical Sciences*. American Cancer Society, 2004.

[31] Rémi Dendievel and Yvik Swan. "One step futher: an explicit solution to Robbins' problem when $n = 4$". In: *arXiv preprint* (2016).

[32] Laura Di Domenico et al. "Expected impact of lockdown in Île-de-France and possible exit strategies". In: *medRxiv* (2020).

[33] Odo Diekmann, JAP Heesterbeek, and Michael G Roberts. "The construction of next-generation matrices for compartmental epidemic models". In: *Journal of the Royal Society Interface* 7.47 (2010), pp. 873–885.

[34] Ramses Djidjou-Demasse et al. "Optimal COVID-19 epidemic control until vaccine deployment". In: *medRxiv* (2020).

[35] Kimon Drakopoulos, Asuman Ozdaglar, and John N Tsitsiklis. "An efficient curing policy for epidemics on graphs". In: *IEEE Transactions on Network Science and Engineering* 1.2 (2014), pp. 67–75.

[36] Kimon Drakopoulos, Asuman Ozdaglar, and John N Tsitsiklis. "When is a network epidemic hard to eliminate?" In: *Mathematics of Operations Research* 42.1 (2017), pp. 1–14.

[37] Kimon Drakopoulos and Ramandeep S Randhawa. *Why perfect tests may not be worth waiting for: Information as a commodity.* Tech. rep. Working paper, SSRN, 2020., 2020.

[38] Marina Drosou and Evaggelia Pitoura. "DisC Diversity: Result Diversification Based on Dissimilarity and Coverage". In: *Proc. VLDB Endow.* 6.1 (Nov. 2012), pp. 13–24. ISSN: 2150-8097.

[39] E.B. Dynkin. "The optimum choice of the instant for stopping a Markov process". In: *Sov. Math. Dokl.* 1963.

[40] Ken TD Eames and Matt J. Keeling. "Contact tracing and disease control". In: *Proc. of the Royal Society of London. Series B: Biological Sciences* (2003), pp. 2565–2571.

[41] Erhan Erkut, Yilmaz Ülküsal, and Oktay Yenicerioğlu. "A comparison of p-dispersion heuristics". In: *Computers & operations research* 21.10 (1994), pp. 1103–1113.

[42] Theodoros Evgeniou, David Hardoon R., and Anton Ovchinnikov. "Leveraging AI to Battle This Pandemic ? And The Next One". In: *Harvard Business Review* (Apr. 2020).

[43] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006). ROC Analysis in Pattern Recognition, pp. 861 –874.

[44] M. Feldman and R. Izsak. "Building a Good Team: Secretary Problems and the Supermodular Degree". In: *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*. 2017, pp. 1651–1670.

[45] M. Feldman, O. Svensson, and R. Zenklusen. "A Simple O(log log(rank))-Competitive Algorithm for the Matroid Secretary Problem". In: *Mathematics of Operations Research* 43.2 (2018), pp. 638–650.

[46] Neil Ferguson et al. *Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand*. 2020.

[47] T.S. Ferguson. "Who Solved the Secretary Problem?" In: *Statistical Science*. 1989.

[48] A. Fiat et al. "The Temp Secretary Problem". In: *European Symposium on Algorithms*. Ed. by Nikhil Bansal and Irene Finocchi. 2015.

[49] Seth Flaxman et al. *Report 13: Estimating the number of infections and the impact of non-pharmaceutical interventions on COVID-19 in 11 European countries*. 2020.

[50] Arthur Q. Frank and Stephen M. Samuels. "On an optimal stopping problem of Gusein-Zade". In: *Stochastic Processes and their Applications* 10.3 (1980), pp. 299 –311.

[51] Nigel R Franks et al. "Decision making by small and large house-hunting ant colonies: one size fits all". In: *Animal behaviour* 72.3 (2006), pp. 611–616.

[52] P.R. Freeman. "The secretary Problem and Its Extensions: a Review". In: *International Statistical Review*. 1983.

[53] A. Gnanlet and W. G. Gilland. "Sequential and Simultaneous Decision Making for Optimizing Health Care Resource Flexibilities". In: *Decision Sciences* 40.2 (2009), pp. 295–326.

[54] Daniel G Goldstein et al. "Learning in the repeated secretary problem". In: *arXiv preprint* (2017).

[55] W. Guan et al. "Clinical Characteristics of Coronavirus Disease 2019 in China". In: *New England Journal of Medicine* (2020).

[56] SM Gusein-Zade. "The problem of choice and the optimal stopping rule for a sequence of independent trials". In: *Theory of Probability & Its Applications* 11.3 (1966), pp. 472–476.

[57] Mohammad Taghi Hajiaghayi, Robert Kleinberg, and David C Parkes. "Adaptive limited-supply online auctions". In: *Proceedings of the ACM conference on Electronic commerce*. 2004, pp. 71–80.

[58] A. Helmi, C. Martínez, and A. Panholzer. "Analysis of the Strategy "Hiring Above the m-th Best Candidate"". In: *Algorithmica*. Vol. 70. 2014, pp. 267–300.

[59] A. L. Hill et al. "Infectious disease modeling of social contagion in networks". In: *PLoS Computational Biology* 6.11 (2010), e1000968.

[60] Joseph L Hodges and Lucien Le Cam. "The Poisson approximation to the Poisson binomial distribution". In: *The Annals of Mathematical Statistics* 31.3 (1960), pp. 737–740.

[61] S. M. Kabene et al. "The importance of human resources management in health care: a global context". In: *Human Resources for Health* 4.1 (2006), p. 20.

[62] A. Kalogeratos et al. "Information Diffusion and Rumor Spreading". In: *Cooperative and Graph Signal Processing*. Ed. by P. M. Djuric and C. Richard. Elsevier, 2018. Chap. 24, pp. 651–678.

[63] Argyris Kalogeratos and Stefano Sarao Mannelli. "Winning the competition: enhancing counter-contagion in SIS-like epidemic processes". In: *arXiv preprint* (2020).

[64] Edward H Kaplan. "COVID-19 Scratch Models To Support Local Decisions". In: *Forthcoming, Manufacturing & Services Operations Management.* (2020).

[65] D. P. Kennedy. "Optimal Stopping of Independent Random Variables and Maximizing Prophets". In: *The Annals of Probability* 13.2 (1985), pp. 566–571.

[66] T. Kesselheim and A. Tönnis. "Think Eternally: Improved Algorithms for the Temp Secretary Problem and Extensions". In: *European Symposium on Algorithms*. 2016, 54:1–54:17.

[67] R. Kleinberg. "A Multiple-choice Secretary Algorithm with Applications to Online Auctions". In: *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*. 2005, pp. 630–631.

[68] Robert Kleinberg and Seth Matthew Weinberg. "Matroid prophet inequalities". In: *Proceedings of the ACM symposium on Theory of computing*. 2012, pp. 123–136.

[69] Joel R Koo et al. "Interventions to mitigate early spread of SARS-CoV-2 in Singapore: a modelling study". In: *The Lancet Infectious Diseases* (2020).

[70] Nitish Korula and Martin Pál. "Algorithms for secretary problems on graphs and hypergraphs". In: *International Colloquium on Automata, Languages, and Programming*. Springer. 2009, pp. 508–520.

[71] P L Krapivsky and S Redner. "Where should you park your car? The $\frac{1}{2}$ rule". In: *Journal of Statistical Mechanics: Theory and Experiment* 2020.7 (2020), p. 073404.

[72] A. Krasnosielska-Kobos. "Multiple-stopping problems with random horizon". In: *Optimization* 64.7 (2015), pp. 1625–1645.

[73] Andreas Krause, Ajit Singh, and Carlos Guestrin. "Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies". In: *Journal of Machine Learning Research* 9 (2008).

[74] U. Krengel and L. Sucheston. "Semiamarts and finite values". In: *American Mathematical Society*. 1977.

[75] A.M. Krieger, M. Pollak, and E. Samuel-Cahn. "Beat the Mean: Sequential Selection by Better than Average Rules". In: *Journal of Applied Probability* 45 (2008), pp. 244–259.

[76] A.M. Krieger, M. Pollak, and E. Samuel-Cahn. "Extrem(ly) mean(ingful): Sequential formation of a quality group". In: *Annals of Applied Probability*. Vol. 20. 2010, pp. 2261–2294.

[77] Adam J Kucharski et al. "Early dynamics of transmission and control of COVID-19: a mathematical modelling study". In: *The Lancet Infectious Diseases* (2020).

[78] L. Le Cam. "An approximation theorem for the Poisson binomial distribution." In: *Pacific Journal of Mathematics* 10.4 (1960), pp. 1181–1197.

[79] D.V. Lindley. "Dynamic programming and decision theory". In: *Applied Statistics*. Vol. 101. 1961, pp. 39–51.

[80] M. Liu and J. Liang. "Dynamic optimization model for allocating medical resources in epidemic controlling". In: *Journal of Industrial Engineering and Management* 6.1 (2013), pp. 73–88.

[81] Alun Lloyd. "Estimating variability in models for recurrent epidemics: Assessing the use of moment closure techniques". In: *Theoretical population biology* 65 (Mar. 2004), pp. 49–65.

[82] Lars Lorch et al. "Stochastic optimal control of epidemic processes in networks". In: *arXiv preprint* (2018).

[83] Wenhao Luo, Changjoo Nam, and Katia Sycara. "Online decision making for stream-based robotic sampling via submodular optimization". In: *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. Nov. 2017, pp. 118–123.

[84] Sofus A Macskassy, Foster Provost, and Saharon Rosset. "ROC confidence bands: An empirical evaluation". In: *Proceedings of the International Conference on Machine learning*. 2005, pp. 537–544.

[85] Paul Marjoram et al. "Markov chain Monte Carlo without likelihoods". In: *Proceedings of the National Academy of Sciences* 100.26 (2003), pp. 15324–15328.

[86] P. V. Mieghem, J. Omic, and R. Kooij. "Virus spread in networks". In: *IEEE/ACM Transactions on Networking* 17.1 (2009), pp. 1–14.

[87] M.L. Nikolaev and G.Y. Sofronov. "A multiple optimal stopping rule for sums of independent random variables". In: *Diskr. Mat.* Vol. 19. 2007, pp. 42–51.

[88] Charles E. M. Pearce, Krzysztof Szajowski, and Mitsushi Tamaki. "Duration problem with multiple exchanges". In: *Numerical Algebra, Control & Optimization* 2 (2012), p. 333.

[89] Anna Petherick. "Developing antibody tests for SARS-CoV-2". In: *The Lancet, World Report* 395 (2020).

[90] Joseph D. Petruccelli. "Full-Information Best-Choice Problems with Recall of Observations and Uncertainty of Selection Depending on the Observation". In: *Advances in Applied Probability* 14.2 (1982), pp. 340–358.

[91] Zdzisław Porosiński. "On optimal choosing of one of the k best objects". In: *Statistics & Probability Letters* 65.4 (2003), pp. 419 –432.

[92] Victor M Preciado et al. "Optimal vaccine allocation to control epidemic outbreaks in arbitrary networks". In: *IEEE Conference on Decision and Control*. IEEE. 2013, pp. 7486–7491.

[93] Sergio Ramírez-Gallego et al. "A survey on data preprocessing for data stream mining: Current status and future directions". In: *Neurocomputing* 239 (2017), pp. 39–57.

[94] Mark D. Reid and Robert C. Williamson. "Information, Divergence and Risk for Binary Experiments". In: *Journal of Machine Learning Research* 12.22 (2011), pp. 731–817.

[95] Henrik Salje et al. "Estimating the burden of SARS-CoV-2 in France". In: *Science* (2020).

[96] K. Scaman, A. Kalogeratos, and N. Vayatis. "A Greedy Approach for Dynamic Control of Diffusion Processes in Networks". In: *IEEE International Conference on Tools with Artificial Intelligence*. 2015, pp. 652–659.

[97] K. Scaman, A. Kalogeratos, and N. Vayatis. "Suppressing Epidemics in Networks using Priority Planning". In: *IEEE Transactions on Network Science and Engineering* 3.4 (2016), pp. 271–285.

[98] D. Seale and A. Rapoport. "Sequential Decision Making with Relative Ranks: An Experimental Investigation of the 'Secretary Problem'". In: *Organizational Behavior and Human Decision Processes*. Vol. 69. 1997, pp. 221–236.

[99] Thomas D. Seeley, P. Kirk Visscher, and Kevin M. Passino. "Group Decision Making in Honey Bee Swarms: When 10,000 bees go house hunting, how do they cooperatively choose their new nesting site?" In: *American Scientist* 94.3 (2006), pp. 220–229.

[100] J.M. Steele. "Le Cam's Inequality and Poisson Approximations". In: *The American Mathematical Monthly* 101.1 (1994), pp. 48–54.

[101] J. Stewart. "Optimal selection from a random sequence with learning of the underlying distribution". In: *Journal of the American Statistical Association*. Vol. 73. 1978, pp. 775–780.

[102] T.J. Stewart. "The Secretary Problem with an Unknown Number of Options". In: *Operations Research* 29.1 (1981), pp. 130–145.

[103] Hanghang Tong et al. "Gelling, and melting, large graphs by edge manipulation". In: *Proceedings of the ACM CIKM* (2012), pp. 245–254.

[104] Samuel G Urwin, Gavinder Kandola, and Sara Graziadio. *What prognostic clinical risk prediction scores for COVID-19 are currently available for use in the community setting?* 2020.

[105] Erik Vee et al. "Efficient computation of diverse query results". In: *2008 IEEE 24th International Conference on Data Engineering*. IEEE. 2008, pp. 228–236.

[106] C. Jason Wang, Chun Y. Ng, and Robert H. Brook. "Response to COVID-19 in Taiwan: Big Data Analytics, New Technology, and Proactive Testing". In: *Journal of the American Medical Association* 323.14 (Apr. 2020), pp. 1341–1342.

[107] D. J. Watts and S. H. Strogatz. "Emergence of Scaling in Random Networks". In: *Nature* 393 (1998), 440 EP.

[108] Justin Zhan et al. "Vaccination allocation in large dynamic networks". In: *Journal of Big Data* 4.1 (2017), p. 2.

[109] Yan Zhu and Eamonn J. Keogh. "Irrevocable-choice algorithms for sampling from a stream". In: *Data Mining and Knowledge Discovery* 30 (2016), pp. 998–1023.

université PARIS-SACLAY

ÉCOLE DOCTORALE
de mathématiques
Hadamard (EDMH)

Title: Sequential Resource Allocation for Network Diffusion Control

Keywords: Epidemic control, Sequential Selection Problem, Optimal stopping

Abstract: In this thesis we extend the Dynamic Resource Allocation (DRA) problem and propose a multi-round dynamic control framework, which we realize through the derived Sequential DRA model (SDRA). Contrary to the standard full-information and full-access DRA considerations, at each intervention round, the DM gains information and access only a fraction of the nodes, in a sequential fashion.

Standard SSP variants, such as the very well-known secretary problem, begin with an empty selection set (cold-start) and perform the selection process once over a single candidate set (single-round). These two limitations are addressed in this thesis. First, we introduce the novel Warm-starting SSP setting that considers having at hand a reference set, which is a set of previously selected items, and tries to update optimally that set while examining the sequence of arriving candidates. The Multi-round Sequential Selection Process, the new online-within-online problem, is then introduced as a natural extension of the warm-starting selection.

Both rank-based and score-based objective functions over the final selection are considered. A cutoff-based approach is proposed for the former, while the optimal strategy based on dynamic thresholding is derived for the latter assuming that the score distribution is known. These strategies are then put in comparison for their efficiency in the traditional selection setting as well as in solving network control problems that motivated this thesis. The generality of the introduced models allow their application to a wide variety of fields and problems; for instance, reoccurring recruiting processes or management of resources (e.g. beds, staff) in healthcare units.

Titre: Contrôle de processus de diffusion sur graphe avec Allocation Séquentielle de Ressources

Mots-clés: Contrôle d'épidémies, Problème de Sélection Séquentielle, Arrêt optimal

Résumé: Dans cette thèse, nous étendons le problème de l'allocation dynamique de ressources (DRA) et proposons un cadre de contrôle dynamique multi-tours, que nous réalisons grâce au modèle dérivé de DRA Séquentiel (SDRA). Contrairement aux considérations standard d'information et d'accès complets à l?environnement, à chaque cycle d'intervention, la DM n?a d'accès qu'à une fraction des nœuds, de manière séquentielle.

Les variantes standard de Processus de Sélection Séquentielle (SSP), telles que le très connu problème de la secrétaire, commencent avec un ensemble de sélection vide (démarrage à froid) et effectuent le processus de sélection une fois sur un seul ensemble de candidats (un seul tour). Ces deux limites sont abordées dans la présente thèse. Tout d'abord, nous introduisons le nouveau paramètre de démarrage à chaud dans lequel la DM a un ensemble de référence à portée de main, à savoir un ensemble d'éléments préalablement sélectionnés. Le but est de mettre à jour cet ensemble de façon optimale tout en examinant la séquence de candidats qui arrivent. Le processus de sélection séquentielle à plusieurs tours est ensuite proposé comme extension naturelle de la sélection avec démarrage à chaud.

Des fonctions d'objectif basées sur le rang et le score lors de la sélection finale sont prises en compte. Une approche basée sur le seuil est proposée pour la première, tandis que la stratégie optimale basée sur le seuillage dynamique est obtenue pour la seconde, en supposant que la distribution de scores est connue. Ces stratégies sont ensuite mises en comparaison dans le cadre de la sélection traditionnelle ainsi que pour la résolution de problèmes de contrôle sur graphe. La généralité des modèles introduits permet leur application à une grande variété de domaines et de problèmes; par exemple, les processus de recrutement récurrents ou la gestion des ressources (lits, personnel, etc.) dans les unités de soins.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France